

OpenPR

0.0

Generated by Doxygen 1.7.1

Mon Apr 25 2011 18:27:05

Contents

1	Main Page	1
2	Todo List	3
3	Namespace Index	5
3.1	Namespace List	5
4	Class Index	7
4.1	Class Hierarchy	7
5	Class Index	9
5.1	Class List	9
6	File Index	11
6.1	File List	11
7	Namespace Documentation	13
7.1	openpr Namespace Reference	13
7.1.1	Detailed Description	14
7.1.2	Typedef Documentation	14
7.1.2.1	bmNameToTypeMap	14
7.1.2.2	string	14
7.1.3	Enumeration Type Documentation	14
7.1.3.1	eMode	14
7.1.4	Function Documentation	14
7.1.4.1	cArchitectureNameConst	14
7.1.4.2	cEdaNameConst	14
7.1.4.3	cXilinxNameConst	14
7.2	openpr::bitstream Namespace Reference	14
7.2.1	Enumeration Type Documentation	15

7.2.1.1	tile_types	15
7.3	openpr::netlist Namespace Reference	16
7.3.1	Typedef Documentation	16
7.3.1.1	tokenizer	16
7.3.2	Function Documentation	16
7.3.2.1	hash_value	16
7.3.2.2	hash_value	16
8	Class Documentation	17
8.1	openpr::AntiCoreBase Class Reference	17
8.1.1	Detailed Description	19
8.1.2	Member Typedef Documentation	19
8.1.2.1	tileToSiteMap	19
8.1.3	Constructor & Destructor Documentation	19
8.1.3.1	AntiCoreBase	19
8.1.3.2	AntiCoreBase	20
8.1.3.3	~AntiCoreBase	20
8.1.4	Member Function Documentation	20
8.1.4.1	allocateMask	20
8.1.4.2	blockRoutes	21
8.1.4.3	blockRoutes	21
8.1.4.4	blockSites	21
8.1.4.5	blockTileRoutes	22
8.1.4.6	blockTileRoutesPartial	22
8.1.4.7	buildSiteMap	23
8.1.4.8	buildValidBoundaries	23
8.1.4.9	checkEfficacy	23
8.1.4.10	dumpMask	24
8.1.4.11	expandRegion	24
8.1.4.12	expandRegionToINT	24
8.1.4.13	exportPipFromArc	24
8.1.4.14	genMacroPlacement	24
8.1.4.15	genPlaceConstraints	25
8.1.4.16	genProhibitConstraints	25
8.1.4.17	genProhibitConstraints	25
8.1.4.18	getRegionTiles	25
8.1.4.19	getRegionVertices	25

8.1.4.20	getSiteType	26
8.1.4.21	importXDL	26
8.1.4.22	inRegion	26
8.1.4.23	mergeClockTree	26
8.1.4.24	placeMacro	27
8.1.4.25	retrieveDynamicRegion	27
8.1.4.26	setMode	28
8.1.4.27	setRegionVertices	28
8.1.4.28	setupRouteBlocker	28
8.1.4.29	shrinkRegion	29
8.1.4.30	siteNameToTileIndex	29
8.1.4.31	updateRegion	29
8.1.4.32	updateRegionExpand	30
8.1.4.33	validateRegion	30
8.1.5	Member Data Documentation	30
8.1.5.1	blockingNet	30
8.1.5.2	busMacroMap	31
8.1.5.3	currentMode	31
8.1.5.4	endTile	31
8.1.5.5	macroTiles	31
8.1.5.6	macroWidth	31
8.1.5.7	mask	31
8.1.5.8	mDB	31
8.1.5.9	mSegments	31
8.1.5.10	mTiles	31
8.1.5.11	placedXDLInput	32
8.1.5.12	prohibitedSites	32
8.1.5.13	sinks_buf	32
8.1.5.14	siteMap	32
8.1.5.15	sources_buf	32
8.1.5.16	startTile	32
8.1.5.17	validBoundaries	32
8.1.5.18	wires_buf	32
8.1.5.19	xMax	33
8.1.5.20	xMin	33
8.1.5.21	yMax	33

8.1.5.22	yMin	33
8.2	openpr::AntiCoreV4 Class Reference	33
8.2.1	Detailed Description	34
8.2.2	Constructor & Destructor Documentation	34
8.2.2.1	AntiCoreV4	34
8.2.2.2	AntiCoreV4	34
8.2.3	Member Function Documentation	35
8.2.3.1	genMacroPlacement	35
8.2.3.2	placeMacro	35
8.2.4	Member Data Documentation	36
8.2.4.1	tilesPerRegion	36
8.3	openpr::AntiCoreV5 Class Reference	36
8.3.1	Detailed Description	37
8.3.2	Constructor & Destructor Documentation	37
8.3.2.1	AntiCoreV5	37
8.3.2.2	AntiCoreV5	37
8.3.3	Member Function Documentation	38
8.3.3.1	genMacroPlacement	38
8.3.3.2	placeMacro	38
8.3.4	Member Data Documentation	39
8.3.4.1	tilesPerRegion	39
8.4	openpr::bitstream::architecture Class Reference	39
8.4.1	Detailed Description	39
8.4.2	Friends And Related Function Documentation	40
8.4.2.1	device	40
8.4.2.2	virtex4	40
8.4.2.3	virtex5	40
8.4.3	Member Data Documentation	40
8.4.3.1	frame_height	40
8.4.3.2	frame_words	40
8.4.3.3	tile_frames	40
8.5	openpr::bitstream::bitstream Class Reference	40
8.5.1	Detailed Description	42
8.5.2	Constructor & Destructor Documentation	42
8.5.2.1	bitstream	42
8.5.2.2	bitstream	43

8.5.2.3	~bitstream	43
8.5.3	Member Function Documentation	43
8.5.3.1	buildGCLKItoaMap	43
8.5.3.2	buildItoaMap	44
8.5.3.3	buildPartial	44
8.5.3.4	buildXDLName	45
8.5.3.5	expect	45
8.5.3.6	expect	45
8.5.3.7	expect	46
8.5.3.8	farToStruct	46
8.5.3.9	loadFile	46
8.5.3.10	mapBitstream	46
8.5.3.11	mapBRAM	47
8.5.3.12	readHeader	48
8.5.3.13	readPackets	48
8.5.3.14	readXilinxString	48
8.5.3.15	structToFar	49
8.5.3.16	write	49
8.5.3.17	write	49
8.5.3.18	write	49
8.5.3.19	writeBitstream	49
8.5.3.20	writeFrames	49
8.5.3.21	writeHeader	50
8.5.3.22	writePackets	50
8.5.3.23	writePacketsPartial	50
8.5.3.24	writeXilinxString	51
8.5.4	Member Data Documentation	51
8.5.4.1	bitstreamFile	51
8.5.4.2	bitstreamLength	51
8.5.4.3	bitstreamWordCount	51
8.5.4.4	cfgMemoryStart	51
8.5.4.5	changedFrames	51
8.5.4.6	designDate	51
8.5.4.7	designName	51
8.5.4.8	designTime	51
8.5.4.9	deviceName	51

8.5.4.10	findexToFaddr	52
8.5.4.11	frame_array	52
8.5.4.12	frameBitmap	52
8.5.4.13	frameECC	52
8.5.4.14	isPartial	52
8.5.4.15	mFrameData	52
8.5.4.16	my_dev	52
8.5.4.17	num_frames	52
8.5.4.18	tile_map	52
8.5.4.19	tileMap	52
8.6	openpr::bitstream::device Class Reference	53
8.6.1	Detailed Description	54
8.6.2	Constructor & Destructor Documentation	55
8.6.2.1	device	55
8.6.2.2	~device	55
8.6.3	Member Function Documentation	55
8.6.3.1	bramcoord_to_major	55
8.6.3.2	build_xdl_layout	56
8.6.3.3	chip_height	56
8.6.3.4	chip_width	56
8.6.3.5	frame_offset	56
8.6.3.6	get_addressable_blk_types	56
8.6.3.7	get_blk_type	56
8.6.3.8	get_blk_type	57
8.6.3.9	get_cfg_size	57
8.6.3.10	get_cfg_size	57
8.6.3.11	get_chip_id	57
8.6.3.12	get_frame_words	57
8.6.3.13	get_gclk_index	58
8.6.3.14	get_name	58
8.6.3.15	get_num_rows	58
8.6.3.16	get_row_height	58
8.6.3.17	get_row_width	58
8.6.3.18	get_tile_frames	58
8.6.3.19	get_tile_frames	58
8.6.3.20	get_tile_type	59

8.6.3.21	tile_offset	59
8.6.3.22	tilecoord_to_major	59
8.6.4	Member Data Documentation	60
8.6.4.1	block_type	60
8.6.4.2	clb_slices	60
8.6.4.3	frame_height	60
8.6.4.4	frame_words	60
8.6.4.5	gclk_index	60
8.6.4.6	id	60
8.6.4.7	logic_table	60
8.6.4.8	name	60
8.6.4.9	num_blk_types	61
8.6.4.10	num_cols	61
8.6.4.11	num_rows	61
8.6.4.12	routing_table	61
8.6.4.13	row_layout	61
8.6.4.14	row_width	61
8.6.4.15	tile_frames	61
8.6.4.16	tile_width	61
8.6.4.17	xdl_layout	61
8.7	openpr::netlist::eq_net Struct Reference	61
8.7.1	Detailed Description	62
8.7.2	Member Function Documentation	62
8.7.2.1	operator()	62
8.8	openpr::netlist::eq_pip Struct Reference	62
8.8.1	Detailed Description	62
8.8.2	Member Function Documentation	62
8.8.2.1	operator()	62
8.9	openpr::netlist::eq_point Struct Reference	62
8.9.1	Detailed Description	63
8.9.2	Member Function Documentation	63
8.9.2.1	operator()	63
8.10	openpr::netlist::eq_segment Struct Reference	63
8.10.1	Detailed Description	63
8.10.2	Member Function Documentation	63
8.10.2.1	operator()	63

8.11	<code>openpr::bitstream::frame_addr</code> Struct Reference	63
8.11.1	Detailed Description	64
8.11.2	Constructor & Destructor Documentation	64
8.11.2.1	<code>frame_addr</code>	64
8.11.2.2	<code>frame_addr</code>	64
8.11.3	Member Function Documentation	64
8.11.3.1	<code>str</code>	64
8.11.4	Member Data Documentation	65
8.11.4.1	<code>col</code>	65
8.11.4.2	<code>mna</code>	65
8.11.4.3	<code>row</code>	65
8.11.4.4	<code>tb</code>	65
8.11.4.5	<code>type</code>	65
8.12	<code>openpr::netlist::hash_net</code> Struct Reference	65
8.12.1	Detailed Description	65
8.12.2	Member Function Documentation	65
8.12.2.1	<code>operator()</code>	65
8.13	<code>openpr::netlist::hash_pip</code> Struct Reference	66
8.13.1	Detailed Description	66
8.13.2	Member Function Documentation	66
8.13.2.1	<code>operator()</code>	66
8.14	<code>openpr::netlist::hash_point</code> Struct Reference	66
8.14.1	Detailed Description	66
8.14.2	Member Function Documentation	66
8.14.2.1	<code>operator()</code>	66
8.15	<code>openpr::netlist::hash_segment</code> Struct Reference	66
8.15.1	Detailed Description	67
8.15.2	Member Function Documentation	67
8.15.2.1	<code>operator()</code>	67
8.16	<code>openpr::netlist::InPin</code> Class Reference	67
8.16.1	Detailed Description	68
8.16.2	Constructor & Destructor Documentation	68
8.16.2.1	<code>InPin</code>	68
8.16.2.2	<code>~InPin</code>	68
8.16.3	Member Function Documentation	68
8.16.3.1	<code>operator!=</code>	68

8.16.3.2	operator!=	68
8.16.3.3	operator()	69
8.16.3.4	operator()	69
8.16.3.5	operator<	69
8.16.3.6	operator<	69
8.16.3.7	operator=	69
8.16.3.8	operator=	69
8.16.3.9	operator==	69
8.16.3.10	operator==	69
8.16.3.11	operator>	69
8.16.3.12	operator>	70
8.16.3.13	printData	70
8.16.3.14	printData	70
8.16.4	Member Data Documentation	70
8.16.4.1	inpin	70
8.16.4.2	location	70
8.17	openpr::netlist::Net Class Reference	70
8.17.1	Detailed Description	72
8.17.2	Constructor & Destructor Documentation	72
8.17.2.1	Net	72
8.17.2.2	~Net	72
8.17.2.3	Net	72
8.17.2.4	Net	72
8.17.3	Member Function Documentation	72
8.17.3.1	addCfg	72
8.17.3.2	addInPin	72
8.17.3.3	addOutPin	73
8.17.3.4	clearInPins	73
8.17.3.5	hash_value	73
8.17.3.6	hash_value	73
8.17.3.7	insertPip	73
8.17.3.8	mergePips	73
8.17.3.9	operator!=	73
8.17.3.10	operator!=	73
8.17.3.11	operator()	73
8.17.3.12	operator()	73

8.17.3.13	operator<	73
8.17.3.14	operator<	74
8.17.3.15	operator=	74
8.17.3.16	operator=	74
8.17.3.17	operator==	74
8.17.3.18	operator==	74
8.17.3.19	operator>	74
8.17.3.20	operator>	74
8.17.3.21	printData	74
8.17.3.22	printData	75
8.17.3.23	remotePin	75
8.17.4	Member Data Documentation	75
8.17.4.1	cfg	75
8.17.4.2	configure	75
8.17.4.3	inPins	75
8.17.4.4	name	75
8.17.4.5	outpin	75
8.17.4.6	outpinLetter	75
8.17.4.7	outPins	76
8.17.4.8	pips	76
8.18	openpr::netlist::NetList Class Reference	76
8.18.1	Detailed Description	77
8.18.2	Constructor & Destructor Documentation	77
8.18.2.1	NetList	77
8.18.2.2	NetList	77
8.18.2.3	NetList	77
8.18.2.4	~NetList	78
8.18.3	Member Function Documentation	78
8.18.3.1	findNet	78
8.18.3.2	findNet	78
8.18.3.3	findNet	78
8.18.3.4	findPin	79
8.18.3.5	findPip	79
8.18.3.6	getNetList	79
8.18.3.7	getPipToNet	79
8.18.3.8	insertNet	79

8.18.3.9	insertPip	79
8.18.3.10	netParser	79
8.18.3.11	printData	79
8.18.3.12	printData	80
8.18.3.13	removePip	80
8.18.3.14	topLevelParser	80
8.18.4	Member Data Documentation	81
8.18.4.1	mDB	81
8.18.4.2	netList	81
8.18.4.3	netToPip	81
8.18.4.4	outputXDL	81
8.18.4.5	pipToNet	81
8.18.4.6	pointToPip	81
8.18.4.7	segmentToNet	81
8.19	openpr::openPR Class Reference	81
8.19.1	Detailed Description	83
8.19.2	Constructor & Destructor Documentation	83
8.19.2.1	openPR	83
8.19.2.2	~openPR	84
8.19.3	Member Function Documentation	84
8.19.3.1	buildBlockingNet	84
8.19.3.2	buildRelativePaths	84
8.19.3.3	genLockConstraints	84
8.19.3.4	genPartialBitstream	84
8.19.3.5	genPassThroughScripts	85
8.19.3.6	genPlaceConstraints	85
8.19.3.7	mergeClockTree	86
8.19.3.8	placeMacros	86
8.19.3.9	routeBlocker	87
8.19.3.10	serialize	87
8.19.3.11	setupAntiCore	87
8.19.3.12	setupDynamicRegion	88
8.19.3.13	siteBlocker	88
8.19.4	Friends And Related Function Documentation	88
8.19.4.1	boost::serialization::access	88
8.19.5	Member Data Documentation	88

8.19.5.1	anticore	88
8.19.5.2	blockedXdlPath	88
8.19.5.3	buildPath	89
8.19.5.4	busMacroNames	89
8.19.5.5	busMacroPath	89
8.19.5.6	busMacroPrefix	89
8.19.5.7	busWidth	89
8.19.5.8	clkNetNames	89
8.19.5.9	db	89
8.19.5.10	designName	89
8.19.5.11	deviceName	89
8.19.5.12	dynamicAGName	89
8.19.5.13	fullBsPath	90
8.19.5.14	fullUcfPath	90
8.19.5.15	fullXdlPath	90
8.19.5.16	isPartial	90
8.19.5.17	l_xMax	90
8.19.5.18	l_xMin	90
8.19.5.19	l_yMax	90
8.19.5.20	l_yMin	90
8.19.5.21	mergedXdlPath	90
8.19.5.22	partialBsPath	90
8.19.5.23	partialPath	91
8.19.5.24	passThroughNet2	91
8.19.5.25	passThroughNetName	91
8.19.5.26	pcfPath	91
8.19.5.27	placedXdlPath	91
8.19.5.28	projectPath	91
8.19.5.29	regionDefined	91
8.19.5.30	routedXdlPath	91
8.19.5.31	routePTScriptPath	91
8.19.5.32	staticPath	91
8.19.5.33	staticPlacedXDLPath	92
8.19.5.34	ucfPath	92
8.19.5.35	unroutePTScriptPath	92
8.19.5.36	xMax	92

8.19.5.37	xMin	92
8.19.5.38	yMax	92
8.19.5.39	yMin	92
8.20	openpr::OpenPRTree Class Reference	92
8.20.1	Detailed Description	93
8.20.2	Constructor & Destructor Documentation	93
8.20.2.1	OpenPRTree	93
8.20.3	Member Function Documentation	93
8.20.3.1	databasePath	93
8.20.3.2	edaPath	93
8.20.3.3	executablePath	93
8.20.3.4	logPath	93
8.20.3.5	relativePath	94
8.20.3.6	workingPath	94
8.20.4	Member Data Documentation	94
8.20.4.1	sDatabasePath	94
8.20.4.2	sEdaPath	94
8.20.4.3	sExecutablePath	94
8.20.4.4	sLogPath	94
8.20.4.5	sRelativePath	94
8.20.4.6	sWorkingPath	94
8.21	openpr::netlist::OutPin Class Reference	94
8.21.1	Detailed Description	95
8.21.2	Constructor & Destructor Documentation	96
8.21.2.1	OutPin	96
8.21.2.2	OutPin	96
8.21.2.3	~OutPin	96
8.21.3	Member Function Documentation	96
8.21.3.1	operator!=	96
8.21.3.2	operator!=	96
8.21.3.3	operator()	96
8.21.3.4	operator()	96
8.21.3.5	operator<	96
8.21.3.6	operator<	96
8.21.3.7	operator=	97
8.21.3.8	operator=	97

8.21.3.9	operator==	97
8.21.3.10	operator==	97
8.21.3.11	operator>	97
8.21.3.12	operator>	97
8.21.3.13	printData	97
8.21.3.14	printData	97
8.21.4	Member Data Documentation	98
8.21.4.1	location	98
8.21.4.2	outpin	98
8.22	openpr::netlist::Pin Class Reference	98
8.22.1	Detailed Description	98
8.22.2	Constructor & Destructor Documentation	99
8.22.2.1	Pin	99
8.22.2.2	~Pin	99
8.22.3	Member Function Documentation	99
8.22.3.1	operator!=	99
8.22.3.2	operator!=	99
8.22.3.3	operator()	99
8.22.3.4	operator()	99
8.22.3.5	operator<	99
8.22.3.6	operator<	99
8.22.3.7	operator=	99
8.22.3.8	operator=	99
8.22.3.9	operator==	100
8.22.3.10	operator==	100
8.22.3.11	operator>	100
8.22.3.12	operator>	100
8.22.3.13	printData	100
8.22.3.14	printData	100
8.22.4	Friends And Related Function Documentation	100
8.22.4.1	hash_value	100
8.22.4.2	hash_value	100
8.23	openpr::netlist::Pip Class Reference	100
8.23.1	Detailed Description	102
8.23.2	Constructor & Destructor Documentation	102
8.23.2.1	Pip	102

8.23.2.2	Pip	102
8.23.2.3	Pip	103
8.23.2.4	Pip	103
8.23.2.5	~Pip	104
8.23.3	Member Function Documentation	104
8.23.3.1	generateFullStream	104
8.23.3.2	generateLocation	104
8.23.3.3	getDestination	104
8.23.3.4	getSinkStr	104
8.23.3.5	getSource	104
8.23.3.6	getSourceStr	104
8.23.3.7	getTileStr	105
8.23.3.8	operator!=	105
8.23.3.9	operator!=	105
8.23.3.10	operator()	105
8.23.3.11	operator()	105
8.23.3.12	operator()	105
8.23.3.13	operator()	105
8.23.3.14	operator<	105
8.23.3.15	operator<	105
8.23.3.16	operator=	106
8.23.3.17	operator=	106
8.23.3.18	operator==	106
8.23.3.19	operator==	106
8.23.3.20	operator>	106
8.23.3.21	operator>	106
8.23.3.22	parseLocation	106
8.23.3.23	printData	106
8.23.3.24	printData	107
8.23.4	Member Data Documentation	107
8.23.4.1	destination	107
8.23.4.2	location	107
8.23.4.3	mDB	107
8.23.4.4	source	107
8.23.4.5	type	107
8.23.4.6	wholeData	108

8.23.4.7	xLoc	108
8.23.4.8	yLoc	108
8.24	openpr::netlist::Point Class Reference	108
8.24.1	Detailed Description	109
8.24.2	Constructor & Destructor Documentation	109
8.24.2.1	Point	109
8.24.2.2	Point	109
8.24.2.3	~Point	109
8.24.3	Member Function Documentation	110
8.24.3.1	getSegmentIndex	110
8.24.3.2	getSource	110
8.24.3.3	getSourceDestination	110
8.24.3.4	operator!=	110
8.24.3.5	operator!=	110
8.24.3.6	operator()	110
8.24.3.7	operator()	111
8.24.3.8	operator<	111
8.24.3.9	operator<	111
8.24.3.10	operator==	111
8.24.3.11	operator==	111
8.24.3.12	operator>	111
8.24.3.13	operator>	112
8.24.3.14	setIndices	112
8.24.4	Member Data Documentation	112
8.24.4.1	segmentIndex	112
8.24.4.2	source	112
8.24.4.3	sourceDestination	112
8.24.4.4	tile	112
8.25	openpr::prohibitRange Struct Reference	112
8.25.1	Detailed Description	113
8.25.2	Constructor & Destructor Documentation	113
8.25.2.1	prohibitRange	113
8.25.2.2	prohibitRange	113
8.25.2.3	prohibitRange	114
8.25.3	Member Function Documentation	114
8.25.3.1	operator<	114

8.25.3.2	operator>	114
8.25.4	Member Data Documentation	114
8.25.4.1	maxSite	114
8.25.4.2	minSite	114
8.26	openpr::bitstream::tile_coord Struct Reference	115
8.26.1	Detailed Description	115
8.26.2	Constructor & Destructor Documentation	115
8.26.2.1	tile_coord	115
8.26.2.2	tile_coord	115
8.26.3	Member Function Documentation	115
8.26.3.1	operator==	115
8.26.3.2	set	115
8.26.4	Friends And Related Function Documentation	116
8.26.4.1	hash_value	116
8.26.5	Member Data Documentation	116
8.26.5.1	x	116
8.26.5.2	y	116
8.27	openpr::bitstream::tile_data Struct Reference	116
8.27.1	Detailed Description	117
8.27.2	Constructor & Destructor Documentation	117
8.27.2.1	tile_data	117
8.27.3	Member Function Documentation	117
8.27.3.1	print	117
8.27.4	Member Data Documentation	118
8.27.4.1	byte_off	118
8.27.4.2	coord	118
8.27.4.3	far	118
8.27.4.4	first_frame	118
8.27.4.5	frame_num	118
8.27.4.6	name	118
8.27.4.7	num_frames	118
8.28	openpr::bitstream::v4_bitstream Class Reference	118
8.28.1	Detailed Description	120
8.28.2	Member Enumeration Documentation	120
8.28.2.1	ECommand	120
8.28.2.2	EMasks	121

8.28.2.3	EOpcode	121
8.28.2.4	EPacketType	122
8.28.2.5	ERegister	122
8.28.2.6	EShiftFAR	122
8.28.2.7	EShifts	123
8.28.2.8	EWords	123
8.28.3	Constructor & Destructor Documentation	123
8.28.3.1	v4_bitstream	123
8.28.3.2	v4_bitstream	123
8.28.4	Member Function Documentation	123
8.28.4.1	farToStruct	123
8.28.4.2	readPackets	124
8.28.4.3	reverseFrameBits	124
8.28.4.4	structToFar	124
8.28.4.5	unmangleTilePair	124
8.28.4.6	writeFrameData	125
8.28.4.7	writePacketHeader	125
8.28.4.8	writePackets	125
8.28.4.9	writePacketsPartial	125
8.28.4.10	writePartialFrames	126
8.28.5	Member Data Documentation	126
8.28.5.1	mRegister	126
8.28.5.2	sCommandName	126
8.28.5.3	sOpcodeName	126
8.28.5.4	sRegisterName	126
8.28.5.5	sTypeName	127
8.28.5.6	top	127
8.29	openpr::bitstream::v5_bitstream Class Reference	127
8.29.1	Detailed Description	129
8.29.2	Member Enumeration Documentation	129
8.29.2.1	ECommand	129
8.29.2.2	EMasks	129
8.29.2.3	EOpcode	130
8.29.2.4	EPacketType	130
8.29.2.5	ERegister	130
8.29.2.6	EShiftFAR	131

8.29.2.7	EShifts	131
8.29.2.8	EWords	131
8.29.3	Constructor & Destructor Documentation	132
8.29.3.1	v5_bitstream	132
8.29.3.2	v5_bitstream	132
8.29.4	Member Function Documentation	132
8.29.4.1	farToStruct	132
8.29.4.2	readPackets	132
8.29.4.3	structToFar	133
8.29.4.4	writeFrameData	133
8.29.4.5	writePacketHeader	133
8.29.4.6	writePackets	134
8.29.4.7	writePacketsPartial	135
8.29.4.8	writePartialFrames	135
8.29.5	Member Data Documentation	136
8.29.5.1	mRegister	136
8.29.5.2	sCommandName	136
8.29.5.3	sOpcodeName	136
8.29.5.4	sRegisterName	136
8.29.5.5	sTypeName	136
8.30	openpr::bitstream::virtex4 Class Reference	137
8.30.1	Detailed Description	138
8.30.2	Member Enumeration Documentation	138
8.30.2.1	"@7	138
8.30.3	Constructor & Destructor Documentation	138
8.30.3.1	virtex4	138
8.30.3.2	~virtex4	138
8.30.4	Member Function Documentation	139
8.30.4.1	get_addressable_blk_types	139
8.30.4.2	tile_offset	139
8.30.5	Member Data Documentation	139
8.30.5.1	virtex4_block_type	139
8.30.5.2	virtex4_logic_table	139
8.30.5.3	virtex4_routing_table	140
8.30.5.4	virtex4_tile_frames	140
8.31	openpr::bitstream::virtex5 Class Reference	140

8.31.1	Detailed Description	142
8.31.2	Member Enumeration Documentation	142
8.31.2.1	"@8	142
8.31.3	Constructor & Destructor Documentation	142
8.31.3.1	virtex5	142
8.31.3.2	~virtex5	142
8.31.4	Member Function Documentation	142
8.31.4.1	get_addressable_blk_types	142
8.31.4.2	tile_offset	143
8.31.5	Member Data Documentation	143
8.31.5.1	virtex5_block_type	143
8.31.5.2	virtex5_logic_table	143
8.31.5.3	virtex5_routing_table	143
8.31.5.4	virtex5_tile_frames	143
8.32	openpr::bitstream::xc4vfx60 Class Reference	144
8.32.1	Detailed Description	145
8.32.2	Member Enumeration Documentation	145
8.32.2.1	"@2	145
8.32.3	Constructor & Destructor Documentation	145
8.32.3.1	xc4vfx60	145
8.32.4	Member Data Documentation	146
8.32.4.1	xc4vfx60_id	146
8.32.4.2	xc4vfx60_name	146
8.32.4.3	xc4vfx60_row_layout	146
8.33	openpr::bitstream::xc4vlx15 Class Reference	146
8.33.1	Detailed Description	148
8.33.2	Member Enumeration Documentation	148
8.33.2.1	"@0	148
8.33.3	Constructor & Destructor Documentation	148
8.33.3.1	xc4vlx15	148
8.33.4	Member Data Documentation	148
8.33.4.1	xc4vlx15_id	148
8.33.4.2	xc4vlx15_name	148
8.33.4.3	xc4vlx15_row_layout	148
8.34	openpr::bitstream::xc4vlx60 Class Reference	149
8.34.1	Detailed Description	150

8.34.2	Member Enumeration Documentation	150
8.34.2.1	"@1	150
8.34.3	Constructor & Destructor Documentation	150
8.34.3.1	xc4vlx60	150
8.34.4	Member Data Documentation	150
8.34.4.1	xc4vlx60_id	150
8.34.4.2	xc4vlx60_name	150
8.34.4.3	xc4vlx60_row_layout	151
8.35	openpr::bitstream::xc5vlx110t Class Reference	151
8.35.1	Detailed Description	152
8.35.2	Member Enumeration Documentation	152
8.35.2.1	"@6	152
8.35.3	Constructor & Destructor Documentation	152
8.35.3.1	xc5vlx110t	152
8.35.4	Member Data Documentation	153
8.35.4.1	xc5vlx110t_id	153
8.35.4.2	xc5vlx110t_name	153
8.35.4.3	xc5vlx110t_row_layout	153
8.36	openpr::bitstream::xc5vlx50 Class Reference	153
8.36.1	Detailed Description	155
8.36.2	Member Enumeration Documentation	155
8.36.2.1	"@3	155
8.36.3	Constructor & Destructor Documentation	155
8.36.3.1	xc5vlx50	155
8.36.4	Member Data Documentation	155
8.36.4.1	xc5vlx50_id	155
8.36.4.2	xc5vlx50_name	155
8.36.4.3	xc5vlx50_row_layout	155
8.37	openpr::bitstream::xc5vlx50t Class Reference	156
8.37.1	Detailed Description	157
8.37.2	Member Enumeration Documentation	157
8.37.2.1	"@4	157
8.37.3	Constructor & Destructor Documentation	157
8.37.3.1	xc5vlx50t	157
8.37.4	Member Data Documentation	157
8.37.4.1	xc5vlx50t_id	157

8.37.4.2	xc5vlx50t_name	157
8.37.4.3	xc5vlx50t_row_layout	158
8.38	openpr::bitstream::xc5vsx95t Class Reference	158
8.38.1	Detailed Description	159
8.38.2	Member Enumeration Documentation	159
8.38.2.1	"@5	159
8.38.3	Constructor & Destructor Documentation	159
8.38.3.1	xc5vsx95t	159
8.38.4	Member Data Documentation	160
8.38.4.1	xc5vsx95t_id	160
8.38.4.2	xc5vsx95t_name	160
8.38.4.3	xc5vsx95t_row_layout	160
9	File Documentation	161
9.1	openpr/AntiCore.cpp File Reference	161
9.1.1	Function Documentation	162
9.1.1.1	main	162
9.2	openpr/anticore/AntiCoreBase.cpp File Reference	162
9.3	openpr/anticore/AntiCoreBase.hpp File Reference	163
9.4	openpr/anticore/AntiCoreV4.cpp File Reference	165
9.5	openpr/anticore/AntiCoreV4.hpp File Reference	165
9.6	openpr/anticore/AntiCoreV5.cpp File Reference	166
9.7	openpr/anticore/AntiCoreV5.hpp File Reference	167
9.8	openpr/anticore/OpenPR.cpp File Reference	167
9.9	openpr/anticore/OpenPR.hpp File Reference	168
9.10	openpr/anticore/OpenPRTree.cpp File Reference	169
9.11	openpr/anticore/OpenPRTree.hpp File Reference	170
9.12	openpr/anticore/ProhibitRange.hpp File Reference	171
9.13	openpr/bitstream/architecture.h File Reference	172
9.14	openpr/bitstream/bitstream.cpp File Reference	173
9.15	openpr/bitstream/bitstream.h File Reference	174
9.16	openpr/bitstream/device.cpp File Reference	175
9.17	openpr/bitstream/device.h File Reference	176
9.18	openpr/bitstream/tile.h File Reference	178
9.19	openpr/bitstream/v4_bitstream.cpp File Reference	178
9.20	openpr/bitstream/v4_bitstream.h File Reference	179
9.21	openpr/bitstream/v4_devices.cpp File Reference	180

9.22	openpr/bitstream/v4_devices.h File Reference	181
9.23	openpr/bitstream/v5_bitstream.cpp File Reference	183
9.24	openpr/bitstream/v5_bitstream.h File Reference	183
9.25	openpr/bitstream/v5_devices.cpp File Reference	184
9.26	openpr/bitstream/v5_devices.h File Reference	185
9.27	openpr/bitstream/virtex4.cpp File Reference	187
9.28	openpr/bitstream/virtex4.h File Reference	187
9.29	openpr/bitstream/virtex5.cpp File Reference	189
9.30	openpr/bitstream/virtex5.h File Reference	189
9.31	openpr/netlist/HashStructs.h File Reference	191
9.32	openpr/netlist/InPin.cpp File Reference	192
9.33	openpr/netlist/InPin.h File Reference	192
9.34	openpr/netlist/Net.cpp File Reference	193
9.35	openpr/netlist/Net.h File Reference	194
9.36	openpr/netlist/NetHashStruct.h File Reference	196
9.37	openpr/netlist/NetList.cpp File Reference	196
9.38	openpr/netlist/NetList.h File Reference	197
9.39	openpr/netlist/OutPin.cpp File Reference	198
9.40	openpr/netlist/OutPin.h File Reference	199
9.41	openpr/netlist/Pin.cpp File Reference	200
9.42	openpr/netlist/Pin.h File Reference	200
9.43	openpr/netlist/Pip.cpp File Reference	202
9.44	openpr/netlist/Pip.h File Reference	202
9.45	openpr/netlist/Point.cpp File Reference	203
9.46	openpr/netlist/Point.h File Reference	204

Chapter 1

Main Page

The bitLib bitstream library provides access to XDL configuration settings at the bitstream level. Currently the library supports Xilinx Virtex4 (routing and CLB logic) and Virtex5 (routing) devices. For information on how to use the library, see the `bit_test.cc` and `v5_bit_test.cc` examples, as well as the bitstream class documentation.

Chapter 2

Todo List

Member `openpr::AntiCoreBase::allocateMask()` Replace integer matrix with `dynamic_bitset`.

Replace integer matrix with `dynamic_bitset`.

Member `openpr::bitstream::device::bramcoord_to_major(int x, int y)` Change this so it isn't a dirty hack

Member `openpr::bitstream::v4_bitstream::writePackets(std::fstream &outStream)` test whether this actually works

Member `openpr::bitstream::v4_bitstream::writePacketsPartial(std::fstream &outStream)` get actual partial bitstream sequence and write this function

Member `openpr::bitstream::virtex4::~~virtex4()` Remove if not used

Member `openpr::bitstream::virtex5::~~virtex5()` Remove if not used

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

openpr	13
openpr::bitstream	14
openpr::netlist	16

Chapter 4

Class Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

openpr::AntiCoreBase	17
openpr::AntiCoreV4	33
openpr::AntiCoreV5	36
openpr::bitstream::architecture	39
openpr::bitstream::bitstream	40
openpr::bitstream::v4_bitstream	118
openpr::bitstream::v5_bitstream	127
openpr::bitstream::device	53
openpr::bitstream::virtex4	137
openpr::bitstream::xc4vfx60	144
openpr::bitstream::xc4vlx15	146
openpr::bitstream::xc4vlx60	149
openpr::bitstream::virtex5	140
openpr::bitstream::xc5vlx110t	151
openpr::bitstream::xc5vlx50	153
openpr::bitstream::xc5vlx50t	156
openpr::bitstream::xc5vsx95t	158
openpr::netlist::eq_net	61
openpr::netlist::eq_pip	62
openpr::netlist::eq_point	62
openpr::netlist::eq_segment	63
openpr::bitstream::frame_addr	63
openpr::netlist::hash_net	65
openpr::netlist::hash_pip	66
openpr::netlist::hash_point	66
openpr::netlist::hash_segment	66
openpr::netlist::Net	70
openpr::netlist::NetList	76
openpr::openPR	81
openpr::OpenPRTree	92
openpr::netlist::Pin	98
openpr::netlist::InPin	67

openpr::netlist::OutPin	94
openpr::netlist::Pip	100
openpr::netlist::Point	108
openpr::prohibitRange	112
openpr::bitstream::tile_coord	115
openpr::bitstream::tile_data	116

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

openpr::AntiCoreBase	17
openpr::AntiCoreV4	33
openpr::AntiCoreV5	36
openpr::bitstream::architecture	39
openpr::bitstream::bitstream	40
openpr::bitstream::device	53
openpr::netlist::eq_net	61
openpr::netlist::eq_pip	62
openpr::netlist::eq_point	62
openpr::netlist::eq_segment	63
openpr::bitstream::frame_addr	63
openpr::netlist::hash_net	65
openpr::netlist::hash_pip	66
openpr::netlist::hash_point	66
openpr::netlist::hash_segment	66
openpr::netlist::InPin	67
openpr::netlist::Net	70
openpr::netlist::NetList	76
openpr::openPR	81
openpr::OpenPRTree	92
openpr::netlist::OutPin	94
openpr::netlist::Pin	98
openpr::netlist::Pip	100
openpr::netlist::Point	108
openpr::prohibitRange	112
openpr::bitstream::tile_coord	115
openpr::bitstream::tile_data	116
openpr::bitstream::v4_bitstream	118
openpr::bitstream::v5_bitstream	127
openpr::bitstream::virtex4	137
openpr::bitstream::virtex5	140
openpr::bitstream::xc4vfx60	144
openpr::bitstream::xc4vlx15	146

openpr::bitstream::xc4vlx60	149
openpr::bitstream::xc5vlx110t	151
openpr::bitstream::xc5vlx50	153
openpr::bitstream::xc5vlx50t	156
openpr::bitstream::xc5vsx95t	158

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

openpr/ AntiCore.cpp	161
openpr/anticore/ AntiCoreBase.cpp	162
openpr/anticore/ AntiCoreBase.hpp	163
openpr/anticore/ AntiCoreV4.cpp	165
openpr/anticore/ AntiCoreV4.hpp	165
openpr/anticore/ AntiCoreV5.cpp	166
openpr/anticore/ AntiCoreV5.hpp	167
openpr/anticore/ OpenPR.cpp	167
openpr/anticore/ OpenPR.hpp	168
openpr/anticore/ OpenPRTree.cpp	169
openpr/anticore/ OpenPRTree.hpp	170
openpr/anticore/ ProhibitRange.hpp	171
openpr/bitstream/ architecture.h	172
openpr/bitstream/ bitstream.cpp	173
openpr/bitstream/ bitstream.h	174
openpr/bitstream/ device.cpp	175
openpr/bitstream/ device.h	176
openpr/bitstream/ tile.h	178
openpr/bitstream/ v4_bitstream.cpp	178
openpr/bitstream/ v4_bitstream.h	179
openpr/bitstream/ v4_devices.cpp	180
openpr/bitstream/ v4_devices.h	181
openpr/bitstream/ v5_bitstream.cpp	183
openpr/bitstream/ v5_bitstream.h	183
openpr/bitstream/ v5_devices.cpp	184
openpr/bitstream/ v5_devices.h	185
openpr/bitstream/ virtex4.cpp	187
openpr/bitstream/ virtex4.h	187
openpr/bitstream/ virtex5.cpp	189
openpr/bitstream/ virtex5.h	189
openpr/netlist/ HashStructs.h	191
openpr/netlist/ InPin.cpp	192
openpr/netlist/ InPin.h	192

openpr/netlist/ Net.cpp	193
openpr/netlist/ Net.h	194
openpr/netlist/ NetHashStruct.h	196
openpr/netlist/ NetList.cpp	196
openpr/netlist/ NetList.h	197
openpr/netlist/ OutPin.cpp	198
openpr/netlist/ OutPin.h	199
openpr/netlist/ Pin.cpp	200
openpr/netlist/ Pin.h	200
openpr/netlist/ Pip.cpp	202
openpr/netlist/ Pip.h	202
openpr/netlist/ Point.cpp	203
openpr/netlist/ Point.h	204

Chapter 7

Namespace Documentation

7.1 openpr Namespace Reference

Namespaces

- namespace [bitstream](#)
- namespace [netlist](#)

Classes

- class [AntiCoreBase](#)
- class [AntiCoreV4](#)
- class [AntiCoreV5](#)
- class [openPR](#)
- class [OpenPRTree](#)
- struct [prohibitRange](#)

Typedefs

- typedef std::string [string](#)
- typedef boost::unordered_map< [string](#), [string](#) > [bmNameToTypeMap](#)

Enumerations

- enum [eMode](#) { [eStatic](#), [ePartial](#) }

Functions

- const std::string [cEdaNameConst](#) ("eda")
- const std::string [cArchitectureNameConst](#) ("architecture")
- const std::string [cXilinxNameConst](#) ("xilinx")

7.1.1 Detailed Description

XC4VLX15 Part

This constructor takes in a single string that is in the format of "pip CLBLM_X36Y59 CLBLM_X36Y59 -> CLBLM_X36Y59 ,"

Parameters

input the input string

7.1.2 Typedef Documentation

7.1.2.1 typedef boost::unordered_map<string, string> openpr::bmNameToTypeMap

Stores relationship between the name of a bus macro to its type

Definition at line 41 of file AntiCoreBase.hpp.

7.1.2.2 typedef std::string openpr::string

Definition at line 35 of file AntiCoreBase.hpp.

7.1.3 Enumeration Type Documentation

7.1.3.1 enum openpr::eMode

Whether object is in Static or Partial mode determines behavior.

Enumerator:

eStatic

ePartial

Definition at line 37 of file AntiCoreBase.hpp.

7.1.4 Function Documentation

7.1.4.1 const std::string openpr::cArchitectureNameConst ("architecture")

7.1.4.2 const std::string openpr::cEdaNameConst ("eda")

7.1.4.3 const std::string openpr::cXilinxNameConst ("xilinx")

7.2 openpr::bitstream Namespace Reference

Classes

- struct [frame_addr](#)
- class [architecture](#)
- class [bitstream](#)

- class [device](#)
- struct [tile_coord](#)
- struct [tile_data](#)
- class [v4_bitstream](#)
- class [xc4v1x15](#)
- class [xc4v1x60](#)
- class [xc4vfx60](#)
- class [v5_bitstream](#)
- class [xc5v1x50](#)
- class [xc5v1x50t](#)
- class [xc5vsx95t](#)
- class [xc5v1x110t](#)
- class [virtex4](#)
- class [virtex5](#)

Enumerations

- enum [tile_types](#) {
 [IOB](#), [GCLK](#), [CLB](#), [DSP48](#),
 [BRAM](#), [BRAM_INT](#), [TRANSCV](#), [PAD](#),
 [NUM_TILE_TYPES](#), [MULTIPLE](#), [INVALID](#) }

7.2.1 Enumeration Type Documentation

7.2.1.1 enum openpr::bitstream::tile_types

Enumerator:

IOB

GCLK

CLB

DSP48

BRAM

BRAM_INT

TRANSCV

PAD

NUM_TILE_TYPES

MULTIPLE

INVALID

Definition at line 24 of file architecture.h.

7.3 openpr::netlist Namespace Reference

Classes

- struct [hash_pip](#)
- struct [eq_pip](#)
- struct [hash_point](#)
- struct [eq_point](#)
- struct [hash_segment](#)
- struct [eq_segment](#)
- class [InPin](#)
- class [Net](#)
- struct [hash_net](#)
- struct [eq_net](#)
- class [NetList](#)
- class [OutPin](#)
- class [Pin](#)
- class [Pip](#)
- class [Point](#)

Typedefs

- typedef boost::tokenizer< boost::char_separator< char > > [tokenizer](#)

Functions

- std::size_t [hash_value](#) ([Pin](#) &pin)
- std::size_t [hash_value](#) ([Pin](#) *pin)

7.3.1 Typedef Documentation

7.3.1.1 typedef boost::tokenizer<boost::char_separator<char> > openpr::netlist::tokenizer

Definition at line 33 of file Pin.h.

7.3.2 Function Documentation

7.3.2.1 std::size_t openpr::netlist::hash_value ([Pin](#) & *pin*)

Definition at line 30 of file Pin.cpp.

7.3.2.2 std::size_t openpr::netlist::hash_value ([Pin](#) * *pin*)

Definition at line 34 of file Pin.cpp.

Chapter 8

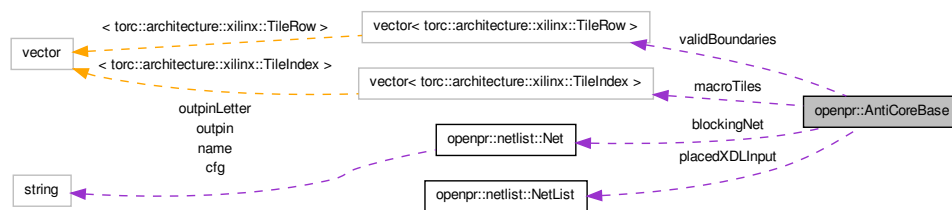
Class Documentation

8.1 openpr::AntiCoreBase Class Reference

#include <AntiCoreBase.hpp>

Inherited by [openpr::AntiCoreV4](#), and [openpr::AntiCoreV5](#).

Collaboration diagram for openpr::AntiCoreBase:



Public Member Functions

- [AntiCoreBase](#) (torc::architecture::DDB &inDB)
- [AntiCoreBase](#) (torc::architecture::DDB &inDB, [string](#) sliceA, [string](#) sliceB)
- void [allocateMask](#) ()
- void [setMode](#) ([eMode](#) newMode)
- void [buildValidBoundaries](#) (const int tilesPerRegion)
- std::string [exportPipFromArc](#) (torc::architecture::Tilewire source, torc::architecture::Tilewire sink, torc::architecture::DDB &mDB)
- void [buildSiteMap](#) ()
- void [importXDL](#) (void)
- void [getRegionVertices](#) (torc::architecture::xilinx::TileCol &_xMin, torc::architecture::xilinx::TileCol &_xMax, torc::architecture::xilinx::TileRow &_yMin, torc::architecture::xilinx::TileRow &_yMax)
- void [setRegionVertices](#) (torc::architecture::xilinx::TileCol _xMin, torc::architecture::xilinx::TileCol _xMax, torc::architecture::xilinx::TileRow _yMin, torc::architecture::xilinx::TileRow _yMax)

- void [updateRegion](#) (string siteA, string siteB)
- void [updateRegionExpand](#) (string siteA, string siteB)
- void [expandRegionToINT](#) ()
- void [expandRegion](#) (int increment)
- void [shrinkRegion](#) (int decrement)
- bool [validateRegion](#) ()
- void [blockSites](#) (void)
- void [genProhibitConstraints](#) ()
- bool [retrieveDynamicRegion](#) (fstream &constraintsFile, string dynamicAGName)
- void [genProhibitConstraints](#) (fstream &ucfFile)
- virtual [bmNameToTypeMap](#) & [genMacroPlacement](#) (int busWidth, fstream &ucfFile, string bus-MacroPrefix)=0
- void [genPlaceConstraints](#) (int busWidth, fstream &ucfFile)
- virtual string [placeMacro](#) (torc::architecture::xilinx::TileIndex ti, bool minOrMax)=0
- string [getSiteType](#) (string site)
- bool [inRegion](#) (torc::architecture::xilinx::TileIndex queryTileIndex)
- torc::architecture::xilinx::TileIndex [siteNameToTileIndex](#) (string siteName)
- void [setupRouteBlocker](#) ()
- void [dumpMask](#) (void)
- vector< string > [getRegionTiles](#) ()
- void [blockTileRoutes](#) (torc::architecture::xilinx::TileRow r, torc::architecture::xilinx::TileCol c)
- void [blockTileRoutesPartial](#) (torc::architecture::xilinx::TileRow r, torc::architecture::xilinx::TileCol c)
- void [blockRoutes](#) ()
- boost::uint32_t [checkEfficacy](#) (torc::architecture::xilinx::TileRow r, torc::architecture::xilinx::TileCol c, boost::uint32_t &noSourceCount)
- void [blockRoutes](#) (string placedXDLPath, string blockedXDLPath, string blockingNetName)
- bool [mergeClockTree](#) (string staticFullXdlPath, string partialPlacedXdlPath, string partialMergedXdlPath, vector< string > clockNetNames)
- ~AntiCoreBase ()

Protected Types

- typedef std::multimap< torc::architecture::xilinx::TileIndex, torc::architecture::Sites::Site > [tile-ToSiteMap](#)

Protected Attributes

- torc::architecture::DDB & [mDB](#)
- const torc::architecture::Tiles & [mTiles](#)
- const torc::architecture::Segments & [mSegments](#)
- torc::architecture::TilewireVector [wires_buf](#)
- torc::architecture::TilewireVector [sinks_buf](#)
- torc::architecture::TilewireVector [sources_buf](#)
- torc::architecture::TileInfo [startTile](#)
- torc::architecture::TileInfo [endTile](#)
- torc::architecture::xilinx::TileCol [xMin](#)
- torc::architecture::xilinx::TileCol [xMax](#)
- torc::architecture::xilinx::TileRow [yMin](#)
- torc::architecture::xilinx::TileRow [yMax](#)

- [eMode](#) [currentMode](#)
- `map< std::string, openpr::prohibitRange > prohibitedSites`
- `std::multimap< torc::architecture::xilinx::TileIndex, torc::architecture::Sites::Site > siteMap`
- `vector< torc::architecture::xilinx::TileIndex > macroTiles`
- `const int macroWidth`
- `bmNameToTypeMap busMacroMap`
- `openpr::netlist::NetList * placedXDLInput`
- `openpr::netlist::Net * blockingNet`
- `int ** mask`
- `vector< torc::architecture::xilinx::TileRow > validBoundaries`

8.1.1 Detailed Description

Definition at line 43 of file `AntiCoreBase.hpp`.

8.1.2 Member Typedef Documentation

8.1.2.1 `typedef std::multimap<torc::architecture::xilinx::TileIndex, torc::architecture::Sites::Site> openpr::AntiCoreBase::tileToSiteMap [protected]`

Maps between Tile indices and the sites within that tile.

Definition at line 75 of file `AntiCoreBase.hpp`.

8.1.3 Constructor & Destructor Documentation

8.1.3.1 `openpr::AntiCoreBase::AntiCoreBase (torc::architecture::DDB & inDB)`

construct Anticore object.

Parameters

inDB CDB database.

Definition at line 25 of file `AntiCoreBase.cpp`.

Here is the call graph for this function:



8.1.3.2 openpr::AntiCoreBase::AntiCoreBase (torc::architecture::DDB & inDB, string sliceA, string sliceB)

Construct an AntiCore object, and define boundaries of AntiCore based upon sliceA and sliceB tile coordinates.

Parameters

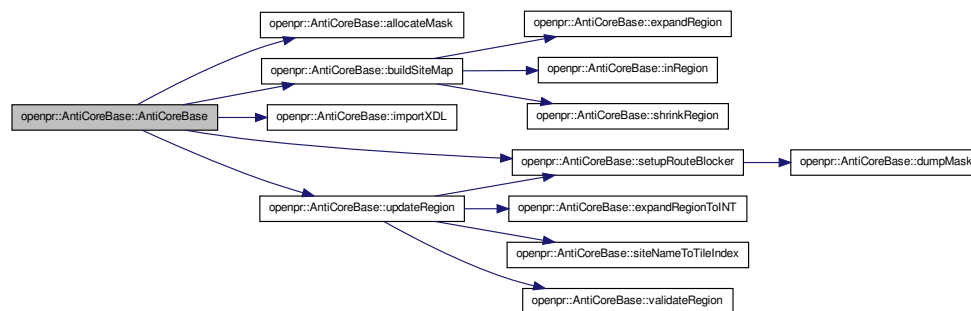
inDB CDB enabled database for access to browser data.

sliceA String representing bottom corner of AntiCore region.

sliceB String representing top corner of AntiCore region.

Definition at line 41 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.3.3 openpr::AntiCoreBase::~~AntiCoreBase ()

Definition at line 1186 of file AntiCoreBase.cpp.

8.1.4 Member Function Documentation

8.1.4.1 void openpr::AntiCoreBase::allocateMask ()

Allocate memory for bitmask.

Todo

Replace integer matrix with dynamic_bitset.

Allocate memory for bitmask.

Todo

Replace integer matrix with dynamic_bitset.

Definition at line 54 of file AntiCoreBase.cpp.

8.1.4.2 void openpr::AntiCoreBase::blockRoutes (string *placedXDLPath*, string *blockedXDLPath*, string *blockingNetName*)

Block routing access to the tiles that surround the region.

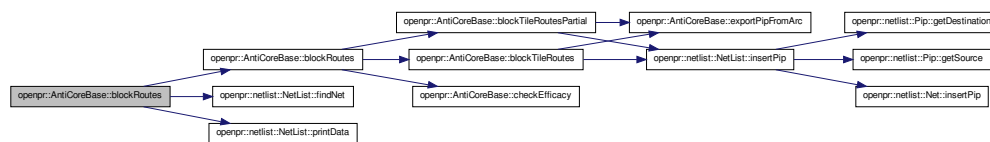
Parameters

placedXDL Pointer to a NetList object that represents the XDL file.

blockingNet name of the net that blocking routes should be added to.

Definition at line 1117 of file AntiCoreBase.cpp.

Here is the call graph for this function:

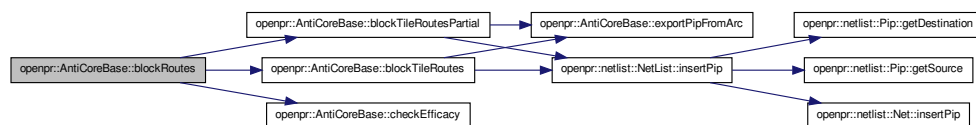


8.1.4.3 void openpr::AntiCoreBase::blockRoutes ()

Block routing access to the tiles that surround the region.

Definition at line 988 of file AntiCoreBase.cpp.

Here is the call graph for this function:

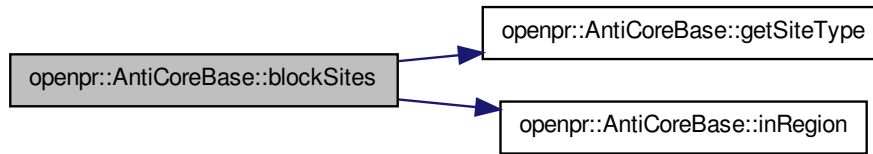


8.1.4.4 void openpr::AntiCoreBase::blockSites (void)

Generate PROHIBIT constraints that can be copied into the ucf file.

Definition at line 392 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.5 void openpr::AntiCoreBase::blockTileRoutes (torc::architecture::xilinx::TileRow *r*, torc::architecture::xilinx::TileCol *c*)

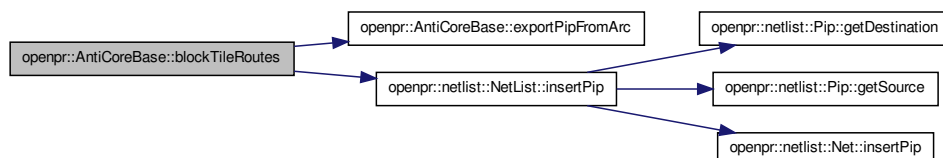
Block routing access to the to the tile specified by *r*,*c*.

Parameters

- r* Row of tile to be blocked.
- c* Column of tile to be blocked.

Definition at line 770 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.6 void openpr::AntiCoreBase::blockTileRoutesPartial (torc::architecture::xilinx::TileRow *r*, torc::architecture::xilinx::TileCol *c*)

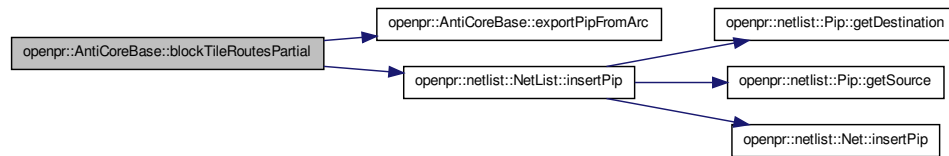
Block routing access to the to the tile specified by *r*,*c*.

Parameters

- r* Row of tile to be blocked.
- c* Column of tile to be blocked.

Definition at line 879 of file AntiCoreBase.cpp.

Here is the call graph for this function:

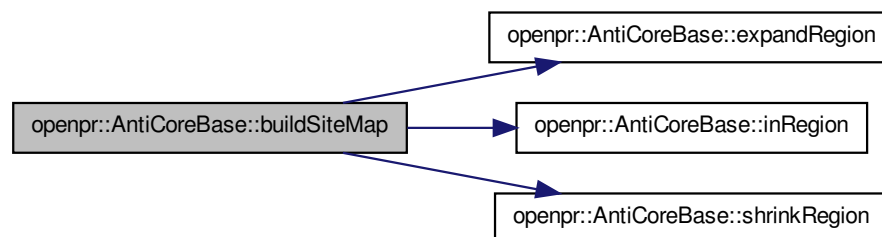


8.1.4.7 void openpr::AntiCoreBase::buildSiteMap ()

Build map between tile index and sites contained within.

Definition at line 104 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.8 void openpr::AntiCoreBase::buildValidBoundaries (const int *tilesPerRegion*)

Build vector which stores all valid boundary rows for this device.

Parameters

tilesPerRegion Number of tile rows per clock region.

Definition at line 73 of file AntiCoreBase.cpp.

8.1.4.9 boost::uint32_t openpr::AntiCoreBase::checkEfficacy (torc::architecture::xilinx::TileRow *r*, torc::architecture::xilinx::TileCol *c*, boost::uint32_t & *noSourceCount*)

Check how effectively par was blocked from entering the region. There are two important measurements here, first is how many region-crossing segments are left or "unused segments", this is the return value from this function. Another important measurement is whether these segments were unused because of a bug,

or because no suitable sources were available. If there were no sources available, then we have effectively blocked the segment even if it is unused!

Parameters

- r* Row of tile to be analysed.
- c* Column of tile to be analysed.
- noSourceCount* Keep track of how many segments are unused because the sources have been exhausted.

Definition at line 1038 of file AntiCoreBase.cpp.

8.1.4.10 void openpr::AntiCoreBase::dumpMask (void)

Dump the bitmask as ASCII art using 1s and 0s.

Definition at line 718 of file AntiCoreBase.cpp.

8.1.4.11 void openpr::AntiCoreBase::expandRegion (int increment)

Expand the region by the specified number of tiles.

Parameters

- increment* The number of tiles the region needs to be expanded by.

Definition at line 298 of file AntiCoreBase.cpp.

8.1.4.12 void openpr::AntiCoreBase::expandRegionToINT ()

Expand the region so that the first column includes the interconnect tiles.

Definition at line 264 of file AntiCoreBase.cpp.

8.1.4.13 std::string openpr::AntiCoreBase::exportPipFromArc (torc::architecture::Tilewire source, torc::architecture::Tilewire sink, torc::architecture::DDB & mDB)

Definition at line 90 of file AntiCoreBase.cpp.

8.1.4.14 virtual bmNameToTypeMap& openpr::AntiCoreBase::genMacroPlacement (int busWidth, fstream & ucFile, string busMacroPrefix) [pure virtual]

Generate placement of bus macros, and return map representing relationship of busmacro name to type.

Parameters

- busWidth* The width of bus entering the region.
- ucFile* fstream for output of constraints
- busMacroPrefix* String containing the prefix of the busMacro name.

Implemented in [openpr::AntiCoreV4](#), and [openpr::AntiCoreV5](#).

8.1.4.15 void openpr::AntiCoreBase::genPlaceConstraints (int *busWidth*, fstream & *ucfFile*)

Generate placement constraints to block site placement within the region.

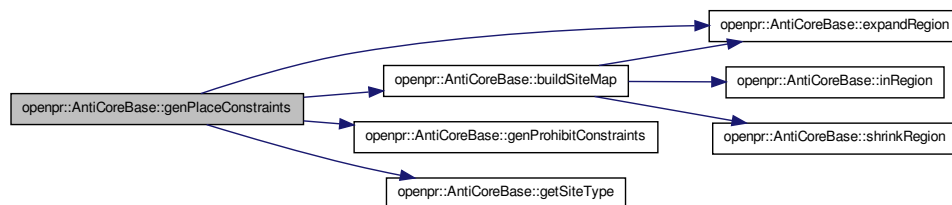
Parameters

busWidth The width of bus entering the region.

ucfFile fstream for output of constraints

Definition at line 527 of file AntiCoreBase.cpp.

Here is the call graph for this function:

**8.1.4.16 void openpr::AntiCoreBase::genProhibitConstraints ()**

Write the necessary Prohibit constraints to stdout

Definition at line 444 of file AntiCoreBase.cpp.

8.1.4.17 void openpr::AntiCoreBase::genProhibitConstraints (fstream & *ucfFile*)

Write the necessary Prohibit constraints to a specified ucf file.

Parameters

ucfFile fstream object representing the ucf file

Definition at line 502 of file AntiCoreBase.cpp.

8.1.4.18 vector< string > openpr::AntiCoreBase::getRegionTiles ()

Return a vector of tilenames for all tiles that exist within the region.

Definition at line 735 of file AntiCoreBase.cpp.

8.1.4.19 void openpr::AntiCoreBase::getRegionVertices (torc::architecture::xilinx::TileCol & *_xMin*, torc::architecture::xilinx::TileCol & *_xMax*, torc::architecture::xilinx::TileRow & *_yMin*, torc::architecture::xilinx::TileRow & *_yMax*)

Return values of four region vertices by reference.

Parameters

- _xMin* Minimum x-coordinate.
- _xMax* Maximum x-coordinate.
- _yMin* Minimum y-coordinate.
- _yMax* Maximum y-coordinate.

Definition at line 168 of file AntiCoreBase.cpp.

8.1.4.20 string openpr::AntiCoreBase::getSiteType (string *site*)

Return site type (CLB, IOB, etc.).

Parameters

- site* Site to be queried.

Definition at line 618 of file AntiCoreBase.cpp.

8.1.4.21 void openpr::AntiCoreBase::importXDL (void)

Import an XDL file and reserve all currently used pips.

Definition at line 135 of file AntiCoreBase.cpp.

8.1.4.22 bool openpr::AntiCoreBase::inRegion (torc::architecture::xilinx::TileIndex *queryTileIndex*)

Determine whether tile is within the dynamic region. This function is used by the blockSites function currently.

Parameters

- queryTileIndex* Tile index to be queried.

Definition at line 629 of file AntiCoreBase.cpp.

8.1.4.23 bool openpr::AntiCoreBase::mergeClockTree (string *staticFullXdlPath*, string *partialPlacedXdlPath*, string *partialMergedXdlPath*, vector< string > *clockNetNames*)

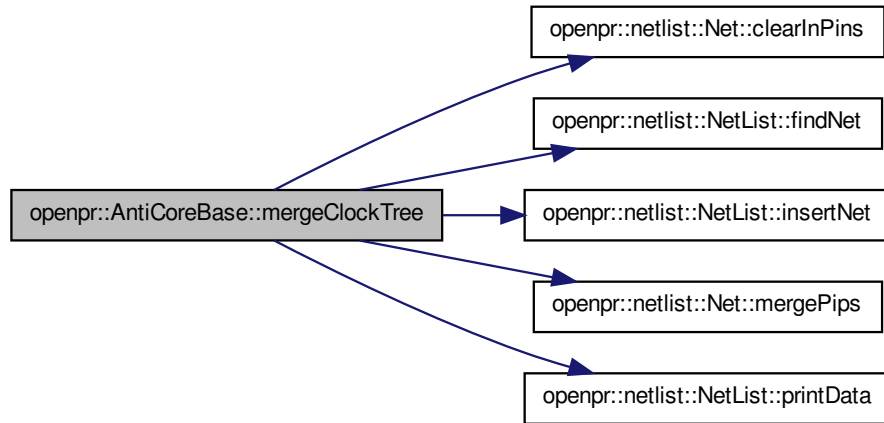
Merge the clock tree net from the static design into the partial design's xdl file.

Parameters

- staticFullXdlPath* string storing full path to static xdl file.
- partialPlacedXdlPath* string storing full path to partial design's placed xdl file.
- partialMergedXdlPath* string storing full path to desired output file.
- clockNetName* Name of clock net in both design files...these MUST be the same (for now).

Definition at line 1152 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.24 `virtual string openpr::AntiCoreBase::placeMacro (torc::architecture::xilinx::TileIndex ti, bool minOrMax) [pure virtual]`

Given a specific tile, find a SLICE to place the macro in.

Parameters

ti `torc::architecture::xilinx::TileIndex` of tile to look in.

minOrMax Indicates whether macro should be placed at most extreme slice or least extreme slice in tile.

Implemented in [openpr::AntiCoreV4](#), and [openpr::AntiCoreV5](#).

8.1.4.25 `bool openpr::AntiCoreBase::retrieveDynamicRegion (fstream & constraintsFile, string dynamicAGName)`

Parse the specified UCF file and update the region to reflect the AREA_GROUP constraints found within.

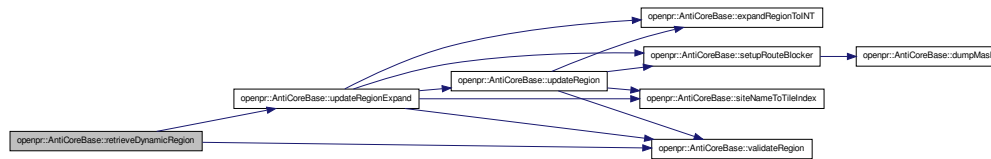
Parameters

ucfFile Path/filename of UCF File.

dynamicAGName the name of the PR region in the UCF file.

Definition at line 467 of file `AntiCoreBase.cpp`.

Here is the call graph for this function:



8.1.4.26 void openpr::AntiCoreBase::setMode (eMode *newMode*)

Set the mode to determine behavior of anticore.

Parameters

newMode enum value of mode to switch to.

Definition at line 66 of file AntiCoreBase.cpp.

8.1.4.27 void openpr::AntiCoreBase::setRegionVertices (torc::architecture::xilinx::TileCol *_xMin*, torc::architecture::xilinx::TileCol *_xMax*, torc::architecture::xilinx::TileRow *_yMin*, torc::architecture::xilinx::TileRow *_yMax*)

Set values of four region vertices.

Parameters

_xMin Minimum x-coordinate.

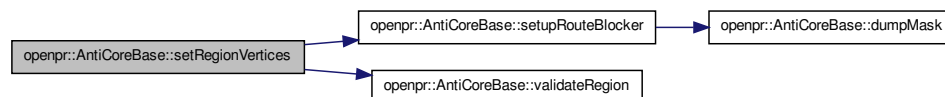
_xMax Maximum x-coordinate.

_yMin Minimum y-coordinate.

_yMax Maximum y-coordinate.

Definition at line 184 of file AntiCoreBase.cpp.

Here is the call graph for this function:

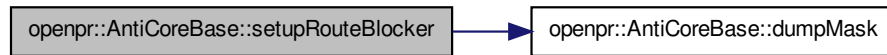


8.1.4.28 void openpr::AntiCoreBase::setupRouteBlocker ()

Setup bitmask.

Definition at line 689 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.29 void openpr::AntiCoreBase::shrinkRegion (int *decrement*)

Shrink the region by the specified number of tiles.

Parameters

decrement The number of tiles the region needs to be shrunk by.

Definition at line 331 of file AntiCoreBase.cpp.

8.1.4.30 torc::architecture::xilinx::TileIndex openpr::AntiCoreBase::siteNameToTileIndex (string *siteName*)

Translate site name to tile index.

Parameters

siteName Name of site to be translated.

Definition at line 661 of file AntiCoreBase.cpp.

8.1.4.31 void openpr::AntiCoreBase::updateRegion (string *siteA*, string *siteB*)

Update boundaries of Anticore region based upon siteA and siteB tile coordinates.

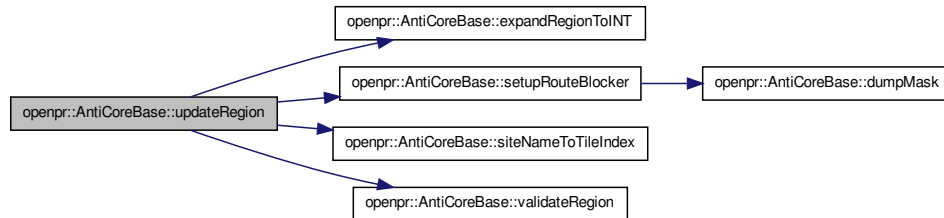
Parameters

siteA String representing bottom corner of AntiCore region.

siteB String representing top corner of AntiCore region.

Definition at line 201 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.32 void openpr::AntiCoreBase::updateRegionExpand (string *siteA*, string *siteB*)

Update boundaries of Anticore region based upon sliceA and sliceB tile coordinates.

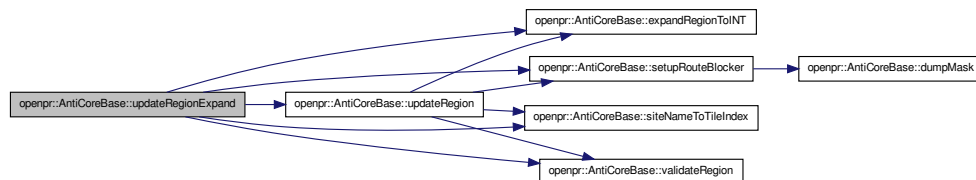
Parameters

siteA String representing bottom corner of AntiCore region.

siteB String representing top corner of AntiCore region.

Definition at line 226 of file AntiCoreBase.cpp.

Here is the call graph for this function:



8.1.4.33 bool openpr::AntiCoreBase::validateRegion ()

Ensure that region spans multiples of clock regions.

Definition at line 363 of file AntiCoreBase.cpp.

8.1.5 Member Data Documentation

8.1.5.1 openpr::netlist::Net* openpr::AntiCoreBase::blockingNet [protected]

Pointer to Net object representing the net on which we will insert blocking pips.

Definition at line 88 of file AntiCoreBase.hpp.

8.1.5.2 bmNameToTypeMap openpr::AntiCoreBase::busMacroMap [protected]

Maps between the uniquely generated name of the bus macro and the type of bus macro.

Definition at line 83 of file AntiCoreBase.hpp.

8.1.5.3 eMode openpr::AntiCoreBase::currentMode [protected]

Stores the current operating mode of the AntiCore

Definition at line 69 of file AntiCoreBase.hpp.

8.1.5.4 torc::architecture::TileInfo openpr::AntiCoreBase::endTile [protected]

Definition at line 61 of file AntiCoreBase.hpp.

8.1.5.5 vector<torc::architecture::xilinx::TileIndex> openpr::AntiCoreBase::macroTiles [protected]

A Vector of the tiles that contain bus macros. Currently this only knows about macros generated by Anti-Core code.

Definition at line 79 of file AntiCoreBase.hpp.

8.1.5.6 const int openpr::AntiCoreBase::macroWidth [protected]

Current macros have a fixed width of 8, perhaps this will change in the future.

Definition at line 81 of file AntiCoreBase.hpp.

8.1.5.7 int openpr::AntiCoreBase::mask [protected]**

Mask representing which tiles are within the region and which are without.

Definition at line 91 of file AntiCoreBase.hpp.

8.1.5.8 torc::architecture::DDB& openpr::AntiCoreBase::mDB [protected]

database reference

Definition at line 46 of file AntiCoreBase.hpp.

8.1.5.9 const torc::architecture::Segments& openpr::AntiCoreBase::mSegments [protected]

segment information reference

Definition at line 50 of file AntiCoreBase.hpp.

8.1.5.10 const torc::architecture::Tiles& openpr::AntiCoreBase::mTiles [protected]

tile and tile type information reference

Definition at line 48 of file AntiCoreBase.hpp.

8.1.5.11 openpr::netlist::NetList* openpr::AntiCoreBase::placedXDLInput [protected]

Pointer to NetList object representing a post-place XDL netlist to be route blocked.

Definition at line 86 of file AntiCoreBase.hpp.

8.1.5.12 map<std::string, openpr::prohibitRange> openpr::AntiCoreBase::prohibitedSites [protected]

Maps between a site type CLB, IOB, etc. and a range of those sites to be prohibited.

Definition at line 72 of file AntiCoreBase.hpp.

8.1.5.13 torc::architecture::TilewireVector openpr::AntiCoreBase::sinks_buf [protected]

scratch sink buffer

Definition at line 55 of file AntiCoreBase.hpp.

8.1.5.14 std::multimap<torc::architecture::xilinx::TileIndex, torc::architecture::Sites::Site> openpr::AntiCoreBase::siteMap [protected]

Definition at line 77 of file AntiCoreBase.hpp.

8.1.5.15 torc::architecture::TilewireVector openpr::AntiCoreBase::sources_buf [protected]

scratch source buffer

Definition at line 57 of file AntiCoreBase.hpp.

8.1.5.16 torc::architecture::TileInfo openpr::AntiCoreBase::startTile [protected]

Two tile info objects store x,y coordinates that define AntiCore rectangle.

Definition at line 60 of file AntiCoreBase.hpp.

8.1.5.17 vector<torc::architecture::xilinx::TileRow> openpr::AntiCoreBase::validBoundaries [protected]

Vector storing valid row indices for region boundaries.

Definition at line 94 of file AntiCoreBase.hpp.

8.1.5.18 torc::architecture::TilewireVector openpr::AntiCoreBase::wires_buf [protected]

scratch segment buffer

Definition at line 53 of file AntiCoreBase.hpp.

8.1.5.19 `torc::architecture::xilinx::TileCol` `openpr::AntiCoreBase::xMax` `[protected]`

Definition at line 65 of file AntiCoreBase.hpp.

8.1.5.20 `torc::architecture::xilinx::TileCol` `openpr::AntiCoreBase::xMin` `[protected]`

Local variables to integrate with torc's col and row

Definition at line 64 of file AntiCoreBase.hpp.

8.1.5.21 `torc::architecture::xilinx::TileRow` `openpr::AntiCoreBase::yMax` `[protected]`

Definition at line 67 of file AntiCoreBase.hpp.

8.1.5.22 `torc::architecture::xilinx::TileRow` `openpr::AntiCoreBase::yMin` `[protected]`

Definition at line 66 of file AntiCoreBase.hpp.

The documentation for this class was generated from the following files:

- openpr/anticore/AntiCoreBase.hpp
- openpr/anticore/AntiCoreBase.cpp

8.2 openpr::AntiCoreV4 Class Reference

```
#include <AntiCoreV4.hpp>
```

Inherits [openpr::AntiCoreBase](#).

Collaboration diagram for openpr::AntiCoreV4:



Public Member Functions

- [AntiCoreV4](#) (`torc::architecture::DDB &inDB`)
- [AntiCoreV4](#) (`torc::architecture::DDB &inDB`, `string siteA`, `string siteB`)
- virtual `bmNameToTypeMap` & `genMacroPlacement` (`int busWidth`, `fstream &ucfFile`, `string bus-MacroPrefix`)
- virtual `string placeMacro` (`torc::architecture::xilinx::TileIndex ti`, `bool minOrMax`)

Private Attributes

- const int [tilesPerRegion](#)

8.2.1 Detailed Description

Definition at line 18 of file AntiCoreV4.hpp.

8.2.2 Constructor & Destructor Documentation

8.2.2.1 `openpr::AntiCoreV4::AntiCoreV4 (torc::architecture::DDB & inDB)`

construct Anticore object.

Parameters

inDB torc::architecture::DDB database.

Definition at line 24 of file AntiCoreV4.cpp.

Here is the call graph for this function:



8.2.2.2 `openpr::AntiCoreV4::AntiCoreV4 (torc::architecture::DDB & inDB, string siteA, string siteB)`

Construct an AntiCore object, and define boundaries of AntiCore based upon sliceA and sliceB tile coordinates.

Parameters

inDB torc::architecture::DDB enabled database for access to browser data.

sliceA String representing bottom corner of AntiCore region.

sliceB String representing top corner of AntiCore region.

Definition at line 36 of file AntiCoreV4.cpp.

Here is the call graph for this function:



8.2.3 Member Function Documentation

8.2.3.1 bmNameToTypeMap & openpr::AntiCoreV4::genMacroPlacement (int *busWidth*, fstream & *ucfFile*, string *busMacroPrefix*) [virtual]

Generate placement of bus macros, and return map representing relationship of busmacro name to type.

Parameters

busWidth The width of bus entering the region.

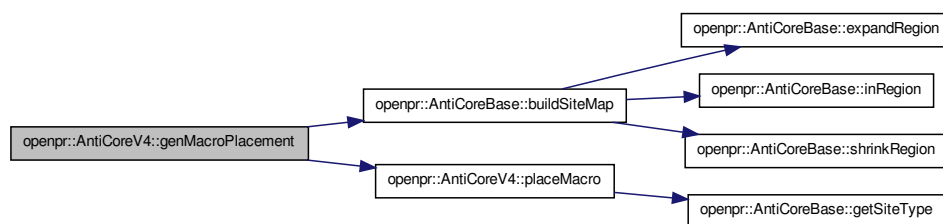
ucfFile fstream for output of constraints

busMacroPrefix String containing the prefix of teh busMacro name.

Implements [openpr::AntiCoreBase](#).

Definition at line 47 of file AntiCoreV4.cpp.

Here is the call graph for this function:



8.2.3.2 string openpr::AntiCoreV4::placeMacro (torc::architecture::xilinx::TileIndex *ti*, bool *minOrMax*) [virtual]

Given a specific tile, find a SLICE to place the macro in.

Parameters

ti torc::architecture::xilinx::TileIndex of tile to look in.

minOrMax Indicates whether macro should be placed at most extreme slice or least extreme slice in tile.

Implements [openpr::AntiCoreBase](#).

Definition at line 115 of file AntiCoreV4.cpp.

Here is the call graph for this function:



8.2.4 Member Data Documentation

8.2.4.1 const int openpr::AntiCoreV4::tilesPerRegion [private]

Definition at line 20 of file AntiCoreV4.hpp.

The documentation for this class was generated from the following files:

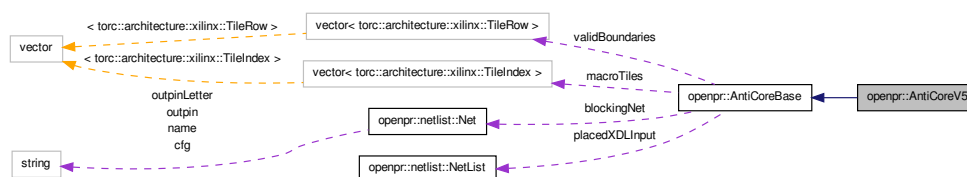
- openpr/anticore/[AntiCoreV4.hpp](#)
- openpr/anticore/[AntiCoreV4.cpp](#)

8.3 openpr::AntiCoreV5 Class Reference

```
#include <AntiCoreV5.hpp>
```

Inherits [openpr::AntiCoreBase](#).

Collaboration diagram for openpr::AntiCoreV5:



Public Member Functions

- [AntiCoreV5](#) (torc::architecture::DDB &inDB)
- [AntiCoreV5](#) (torc::architecture::DDB &inDB, [string](#) siteA, [string](#) siteB)
- virtual [bmNameToTypeMap](#) & [genMacroPlacement](#) (int busWidth, fstream &ucfFile, [string](#) bus-MacroPrefix)

Private Member Functions

- virtual [string placeMacro](#) (torc::architecture::xilinx::TileIndex ti, bool minOrMax)

Private Attributes

- const int [tilesPerRegion](#)

8.3.1 Detailed Description

Definition at line 18 of file AntiCoreV5.hpp.

8.3.2 Constructor & Destructor Documentation

8.3.2.1 openpr::AntiCoreV5::AntiCoreV5 (torc::architecture::DDB & *inDB*)

construct Anticore object.

Parameters

inDB CDB database.

Definition at line 58 of file AntiCoreV5.cpp.

Here is the call graph for this function:



8.3.2.2 openpr::AntiCoreV5::AntiCoreV5 (torc::architecture::DDB & *inDB*, string *sliceA*, string *sliceB*)

Construct an AntiCore object, and define boundaries of AntiCore based upon sliceA and sliceB tile coordinates.

Parameters

inDB CDB enabled database for access to browser data.

sliceA String representing bottom corner of AntiCore region.

sliceB String representing top corner of AntiCore region.

Definition at line 70 of file AntiCoreV5.cpp.

Here is the call graph for this function:



8.3.3 Member Function Documentation

8.3.3.1 `bmNameToTypeMap` & `openpr::AntiCoreV5::genMacroPlacement (int busWidth, fstream & ucfFile, string busMacroPrefix) [virtual]`

Generate placement of bus macros, and return map representing relationship of busmacro name to type.

Parameters

busWidth The width of bus entering the region.

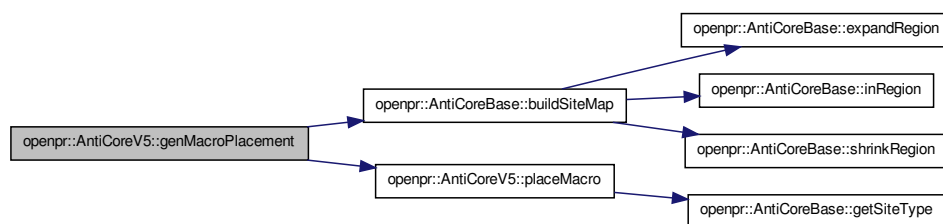
ucfFile `fstream` for output of constraints

busMacroPrefix String containing the prefix of the busMacro name.

Implements [openpr::AntiCoreBase](#).

Definition at line 81 of file `AntiCoreV5.cpp`.

Here is the call graph for this function:



8.3.3.2 `string openpr::AntiCoreV5::placeMacro (torc::architecture::xilinx::TileIndex ti, bool minOrMax) [private, virtual]`

Given a specific tile, find a SLICE to place the macro in.

Parameters

ti `torc::architecture::xilinx::TileIndex` of tile to look in.

minOrMax Indicates whether macro should be placed at most extreme slice or least extreme slice in tile.

Implements [openpr::AntiCoreBase](#).

Definition at line 20 of file AntiCoreV5.cpp.

Here is the call graph for this function:



8.3.4 Member Data Documentation

8.3.4.1 const int openpr::AntiCoreV5::tilesPerRegion [private]

Definition at line 22 of file AntiCoreV5.hpp.

The documentation for this class was generated from the following files:

- openpr/anticore/[AntiCoreV5.hpp](#)
- openpr/anticore/[AntiCoreV5.cpp](#)

8.4 openpr::bitstream::architecture Class Reference

```
#include <architecture.h>
```

Protected Attributes

- int [frame_words](#)
- int [frame_height](#)
- int [tile_frames](#) [NUM_TILE_TYPES]

Friends

- class [virtex4](#)
- class [virtex5](#)
- class [device](#)

8.4.1 Detailed Description

Definition at line 75 of file architecture.h.

8.4.2 Friends And Related Function Documentation

8.4.2.1 friend class device [friend]

Definition at line 86 of file architecture.h.

8.4.2.2 friend class virtex4 [friend]

Definition at line 84 of file architecture.h.

8.4.2.3 friend class virtex5 [friend]

Definition at line 85 of file architecture.h.

8.4.3 Member Data Documentation

8.4.3.1 int openpr::bitstream::architecture::frame_height [protected]

Definition at line 80 of file architecture.h.

8.4.3.2 int openpr::bitstream::architecture::frame_words [protected]

Definition at line 78 of file architecture.h.

8.4.3.3 int openpr::bitstream::architecture::tile_frames[NUM_TILE_TYPES] [protected]

Definition at line 82 of file architecture.h.

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

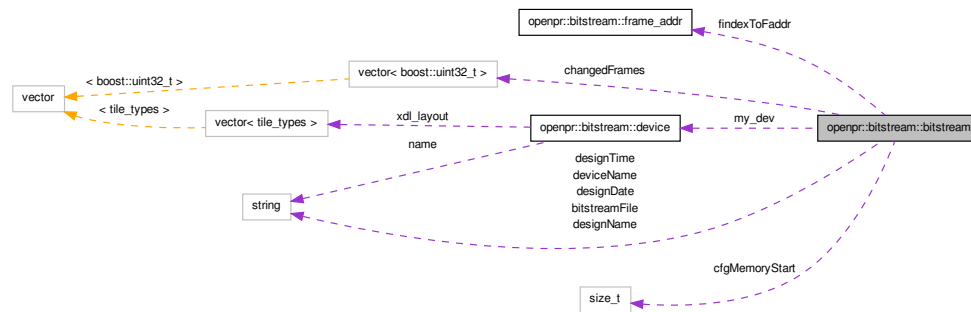
- openpr/bitstream/[architecture.h](#)

8.5 openpr::bitstream::bitstream Class Reference

```
#include <bitstream.h>
```

Inherited by [openpr::bitstream::v4_bitstream](#), and [openpr::bitstream::v5_bitstream](#).

Collaboration diagram for openpr::bitstream::bitstream:



Public Member Functions

- [bitstream](#) ([string](#) device_name, bool frame_ecc=false)
- bool [loadFile](#) ([string](#) bitstream_name)
- bool [writeFrames](#) (fstream &outStream, int startFrame, int numFrames)
- bool [writeBitstream](#) (const [string](#) &outBitstream)
- int [mapBitstream](#) ()
- int [mapBRAM](#) (int x, int y)
- [string](#) [buildXDLName](#) ([string](#) tileType, [openpr::bitstream::tile_coord](#) coord)
- void [buildPartial](#) (vector< [string](#) > regionTiles)
- [~bitstream](#) ()

Protected Member Functions

- void [buildItoaMap](#) ([openpr::bitstream::frame_addr](#) majorAddress, boost::uint32_t startIndex, boost::uint32_t tileFrames)
- void [buildGCLKItoaMap](#) ()
- bool [writeHeader](#) (fstream &outStream)
- virtual bool [readPackets](#) (fstream &inStream)=0
- virtual bool [writePackets](#) (fstream &outStream)=0
- virtual bool [writePacketsPartial](#) (fstream &outStream)=0
- virtual [openpr::bitstream::frame_addr](#) [farToStruct](#) (boost::uint32_t far)=0
- virtual boost::uint32_t [structToFar](#) ([openpr::bitstream::frame_addr](#) far)=0
- [bitstream](#) (void)

Static Protected Member Functions

- static bool [expect](#) (fstream &inStream, boost::uint8_t inExpected)
- static bool [expect](#) (fstream &inStream, boost::uint16_t inExpected)
- static bool [expect](#) (fstream &inStream, boost::uint32_t inExpected)
- static void [readXilinxString](#) (fstream &inStream, [string](#) &outString)
- static bool [readHeader](#) (fstream &inStream, [string](#) &outDesignName, [string](#) &outDeviceName, [string](#) &outDesignDate, [string](#) &outDesignTime, boost::uint32_t &outBitstreamLength)

- static bool [write](#) (fstream &outStream, boost::uint8_t outVal)
- static bool [write](#) (fstream &outStream, boost::uint16_t outVal)
- static bool [write](#) (fstream &outStream, boost::uint32_t outVal)
- static void [writeXilinxString](#) (fstream &outStream, [string](#) inString)

Protected Attributes

- [string](#) [bitstreamFile](#)
- bool [frameECC](#)
- [openpr::bitstream::device](#) * [my_dev](#)
- char ** [frame_array](#)
- boost::uint8_t * [mFrameData](#)
- int [num_frames](#)
- [openpr::bitstream::frame_addr](#) * [findIndexToFAddr](#)
- boost::unordered_map< [openpr::bitstream::tile_coord](#), [openpr::bitstream::tile_data](#) *, boost::hash< [openpr::bitstream::tile_coord](#) > > [tile_map](#)
- boost::unordered_map< std::string, [openpr::bitstream::tile_data](#) *, boost::hash< std::string > > [tileMap](#)
- [string](#) [designName](#)
- [string](#) [deviceName](#)
- [string](#) [designDate](#)
- [string](#) [designTime](#)
- boost::uint32_t [bitstreamLength](#)
- boost::uint32_t [bitstreamWordCount](#)
- size_t [cfgMemoryStart](#)
- bool [isPartial](#)
- vector< boost::uint32_t > [changedFrames](#)
- boost::dynamic_bitset [frameBitmap](#)

8.5.1 Detailed Description

The bitstream class acts as the interface/controller class for bitLib. Most useful bitstream exploration can be done via the public functions of this class. A typical use case for this class will look like this:

- Pass a device name to the [bitstream::bitstream\(string device_name\)](#) constructor
- Call the [bitstream::mapBitstream](#) function
- Call the [bitstream::loadFile](#) function with the bitstream filename as the parameter.
- Use the [bitstream::getArc](#), [bitstream::setArc](#), [bitstream::getCfg](#), [bitstream::setCfg](#) functions to view or modify routing and logic configuration bits.

Examples of how to use the bitstream class can be found in [bit_test.cc](#) and [v5_bit_test.cc](#)

Definition at line 59 of file [bitstream.h](#).

8.5.2 Constructor & Destructor Documentation

8.5.2.1 [openpr::bitstream::bitstream::bitstream \(void \)](#) [[inline](#), [protected](#)]

Null constructor used by the standalone bitstream reader

Definition at line 128 of file [bitstream.h](#).

8.5.2.2 openpr::bitstream::bitstream::bitstream (string *device_name*, bool *frame_ecc* = *false*)

Create and return a new bitstream compatible with the supplied bitstream file.

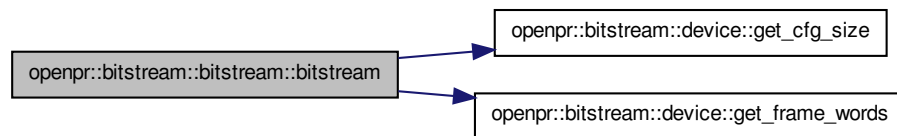
Parameters

inFilename Name of the reference bitstream. Allocate device object based on supplied name.

device_name Specific part to be used.

Definition at line 22 of file bitstream.cpp.

Here is the call graph for this function:



8.5.2.3 openpr::bitstream::bitstream::~~bitstream ()

Compare two unsigned bytes of data and find the bits that differ.

Parameters

a The first byte to compare.

b The second byte to compare. Free all dynamically allocated memory

Definition at line 192 of file bitstream.cpp.

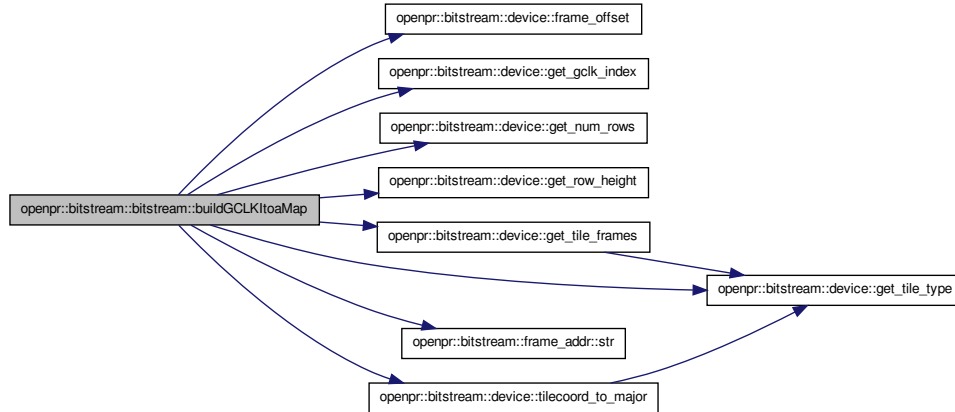
8.5.3 Member Function Documentation

8.5.3.1 void openpr::bitstream::bitstream::buildGCLKItoaMap () [protected]

Fill in the appropriate addresses for GCLK columns.

Definition at line 166 of file bitstream.cpp.

Here is the call graph for this function:



8.5.3.2 void openpr::bitstream::bitstream::buildItoaMap (openpr::bitstream::frame_addr majorAddress, boost::uint32_t startIndex, boost::uint32_t tileFrames) [protected]

Build a mapping from frame index to frame address.

Parameters

- majorAddress** Frame Address for the major column
- startIndex** Starting Index for the tile frames
- tileFrames** number of frame for the tileType

Definition at line 156 of file bitstream.cpp.

8.5.3.3 void openpr::bitstream::bitstream::buildPartial (vector< string > regionTiles)

Take a tileName string and return the [tile_data](#) struct that corresponds to it.

Parameters

- tileName** string containing the XDL name of the tile. Return struct containing Frame Address, Frame Number, Byte Offset, and number of frames the tile spans along with a pointer to the first frame in the frame_array with data for this tile
- x** X Coordinate of Tile
- y** Y Coordinate of Tile Dump all frames in the bitstream in a human readable manner. Dump the requested frames to stdout.
- startFrame** The first frame to print out.
- endFrame** The last frame to print + 1. Compare two bitstreams and find the bits that differ.
- bitstreamA** The first bitstream to compare.
- bitstreamB** The second bitstream to compare. Build vector representing all frames in static bitstream.

regionTiles vector of all tiles in the region, use CDB to generate this list.

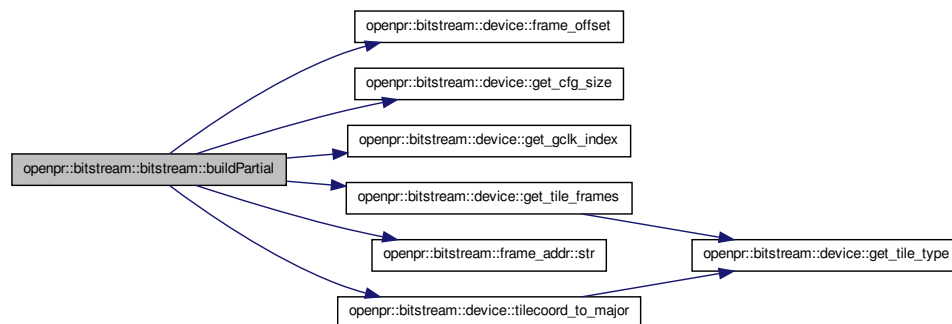
Build the frameBitmap structure to represent which frames should be written out to the bitstream when necessary.

Parameters

regionTiles A vector of all the tiles in a partial region.

Definition at line 456 of file bitstream.cpp.

Here is the call graph for this function:



8.5.3.4 string openpr::bitstream::bitstream::buildXDLName (string *tileType*, openpr::bitstream::tile_coord *coord*)

Extract XDL tile type from tileName and return string.

Parameters

tileName string containing XDL name of tile. Extract XDL coordinates from tileName and return [tile_coord](#) object.

tileName string containing XDL name of tile. Build an XDL name using the tile type and tile coordinate.

tileType String containing tile type info.

coord XDL coordinates of tile.

Definition at line 185 of file bitstream.cpp.

8.5.3.5 bool openpr::bitstream::bitstream::expect (fstream & *inStream*, boost::uint16_t *inExpected*) [static, protected]

Definition at line 225 of file bitstream.cpp.

8.5.3.6 bool openpr::bitstream::bitstream::expect (fstream & *inStream*, boost::uint32_t *inExpected*) [static, protected]

Definition at line 233 of file bitstream.cpp.

8.5.3.7 `bool openpr::bitstream::bitstream::expect (fstream & inStream, boost::uint8_t inExpected) [static, protected]`

Definition at line 217 of file `bitstream.cpp`.

8.5.3.8 `virtual openpr::bitstream::frame_addr openpr::bitstream::bitstream::farToStruct (boost::uint32_t far) [protected, pure virtual]`

Write a packet header to the bitstream.

Parameters

- outStream* Bitstream to be written.
- packetType* uint indicating type 1 or type 2 packet.
- opcode* Operation to be performed on register.
- address* Register address to be written.
- reserved* Param for reserved opcodes.
- count* Number of words to be written in packet.

Implemented in [openpr::bitstream::v4_bitstream](#), and [openpr::bitstream::v5_bitstream](#).

8.5.3.9 `bool openpr::bitstream::bitstream::loadFile (string bitstream_name)`

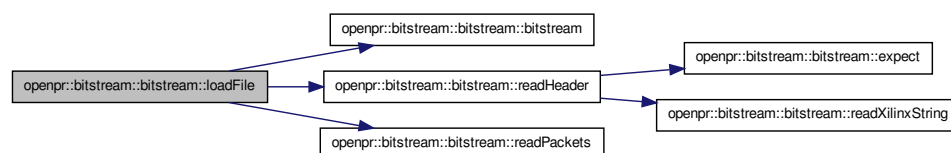
Allocate memory for `frame_array` and sequentially load bitstream.

Parameters

- bitstream_name* File name of bitstream to load

Definition at line 413 of file `bitstream.cpp`.

Here is the call graph for this function:

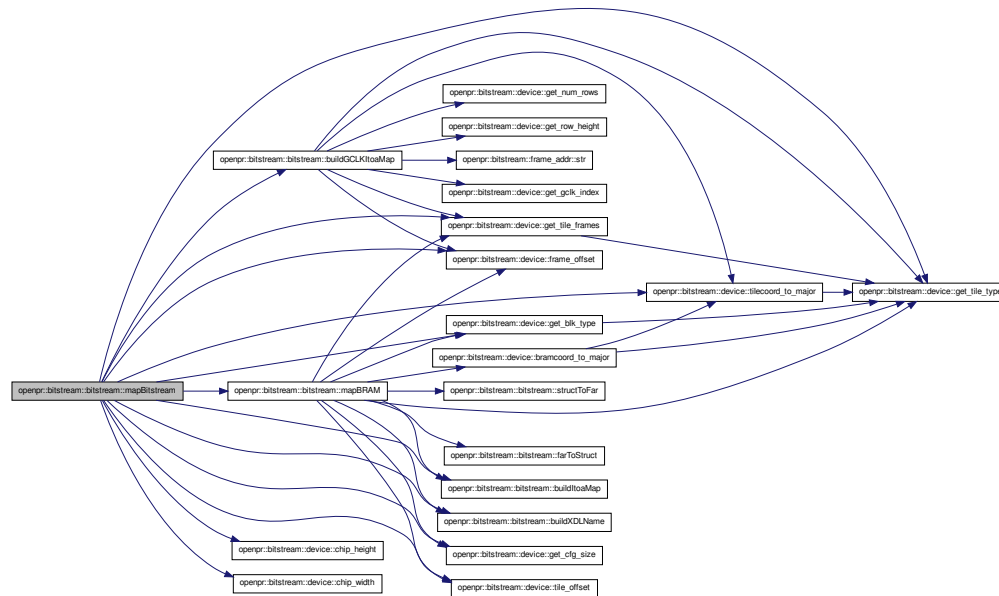


8.5.3.10 `int openpr::bitstream::bitstream::mapBitstream ()`

Build tilemap using CDB. Map tile coordinates to `frame_array` indices

Definition at line 71 of file `bitstream.cpp`.

Here is the call graph for this function:



8.5.3.11 int openpr::bitstream::bitstream::mapBRAM (int x, int y)

Map bram to tile name.

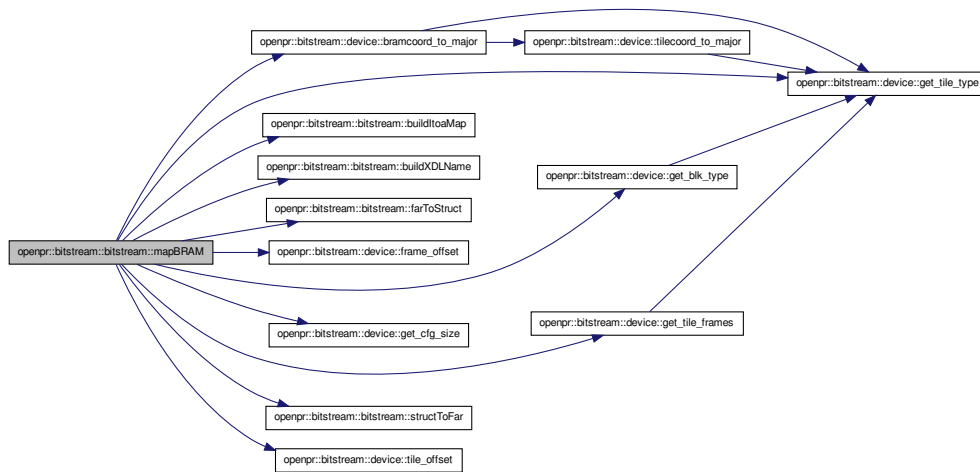
Parameters

x X Coordinate of BRAM

y Y Coordinate of BRAM.

Definition at line 132 of file bitstream.cpp.

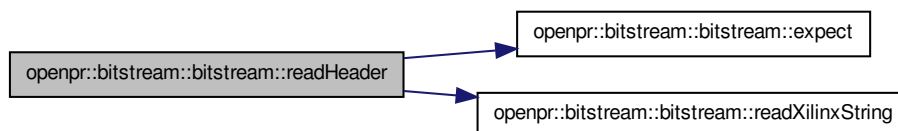
Here is the call graph for this function:



8.5.3.12 `bool openpr::bitstream::bitstream::readHeader (fstream & inStream, string & outDesignName, string & outDeviceName, string & outDesignDate, string & outDesignTime, boost::uint32_t & outBitstreamLength) [static, protected]`

Definition at line 260 of file bitstream.cpp.

Here is the call graph for this function:



8.5.3.13 `virtual bool openpr::bitstream::bitstream::readPackets (fstream & inStream) [protected, pure virtual]`

Implemented in [openpr::bitstream::v5_bitstream](#).

8.5.3.14 `void openpr::bitstream::bitstream::readXilinxString (fstream & inStream, string & outString) [static, protected]`

Definition at line 242 of file bitstream.cpp.

8.5.3.15 `virtual boost::uint32_t openpr::bitstream::bitstream::structToFar (openpr::bitstream::frame_addr far) [protected, pure virtual]`

Implemented in [openpr::bitstream::v4_bitstream](#), and [openpr::bitstream::v5_bitstream](#).

8.5.3.16 `bool openpr::bitstream::bitstream::write (fstream & outStream, boost::uint8_t outVal) [static, protected]`

Definition at line 313 of file bitstream.cpp.

8.5.3.17 `bool openpr::bitstream::bitstream::write (fstream & outStream, boost::uint16_t outVal) [static, protected]`

Definition at line 320 of file bitstream.cpp.

8.5.3.18 `bool openpr::bitstream::bitstream::write (fstream & outStream, boost::uint32_t outVal) [static, protected]`

Definition at line 328 of file bitstream.cpp.

8.5.3.19 `bool openpr::bitstream::bitstream::writeBitstream (const string & outBitstream)`

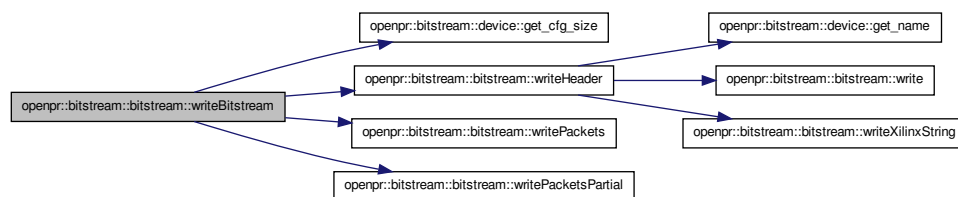
Write a bitstream to the specified file.

Parameters

outBitstream Bitstream file to be written.

Definition at line 390 of file bitstream.cpp.

Here is the call graph for this function:



8.5.3.20 `bool openpr::bitstream::bitstream::writeFrames (fstream & outStream, int startFrame, int numFrames)`

Write configuration memory back to bitstream. Write out partial bitstream.

Parameters

outStream bitstream file to be written Write out frames to to outStream file.

outStream Bitstream file to be written
startFrame first frame to be written
numFrames number of frames to be written

Definition at line 484 of file bitstream.cpp.

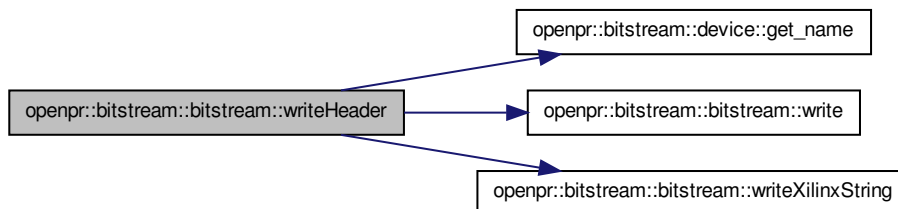
Here is the call graph for this function:



8.5.3.21 **bool openpr::bitstream::bitstream::writeHeader (fstream & outStream)** **[protected]**

Definition at line 350 of file bitstream.cpp.

Here is the call graph for this function:



8.5.3.22 **virtual bool openpr::bitstream::bitstream::writePackets (fstream & outStream)** **[protected, pure virtual]**

Implemented in [openpr::bitstream::v5_bitstream](#).

8.5.3.23 **virtual bool openpr::bitstream::bitstream::writePacketsPartial (fstream & outStream)** **[protected, pure virtual]**

Implemented in [openpr::bitstream::v5_bitstream](#).

8.5.3.24 void openpr::bitstream::bitstream::writeXilinxString (fstream & *outStream*, string *inString*) [static, protected]

Definition at line 336 of file bitstream.cpp.

8.5.4 Member Data Documentation

8.5.4.1 string openpr::bitstream::bitstream::bitstreamFile [protected]

Definition at line 62 of file bitstream.h.

8.5.4.2 boost::uint32_t openpr::bitstream::bitstream::bitstreamLength [protected]

Definition at line 76 of file bitstream.h.

8.5.4.3 boost::uint32_t openpr::bitstream::bitstream::bitstreamWordCount [protected]

Definition at line 77 of file bitstream.h.

8.5.4.4 size_t openpr::bitstream::bitstream::cfgMemoryStart [protected]

Definition at line 78 of file bitstream.h.

8.5.4.5 vector<boost::uint32_t> openpr::bitstream::bitstream::changedFrames [protected]

Definition at line 80 of file bitstream.h.

8.5.4.6 string openpr::bitstream::bitstream::designDate [protected]

Definition at line 74 of file bitstream.h.

8.5.4.7 string openpr::bitstream::bitstream::designName [protected]

Definition at line 72 of file bitstream.h.

8.5.4.8 string openpr::bitstream::bitstream::designTime [protected]

Definition at line 75 of file bitstream.h.

8.5.4.9 string openpr::bitstream::bitstream::deviceName [protected]

Definition at line 73 of file bitstream.h.

8.5.4.10 openpr::bitstream::frame_addr* openpr::bitstream::bitstream::findxToFaddr [protected]

Map between frame indices and frameAddresses for fast lookup

Definition at line 69 of file bitstream.h.

8.5.4.11 char openpr::bitstream::bitstream::frame_array [protected]**

Definition at line 65 of file bitstream.h.

8.5.4.12 boost::dynamic_bitset openpr::bitstream::bitstream::frameBitmap [protected]

Definition at line 81 of file bitstream.h.

8.5.4.13 bool openpr::bitstream::bitstream::frameECC [protected]

Definition at line 63 of file bitstream.h.

8.5.4.14 bool openpr::bitstream::bitstream::isPartial [protected]

Definition at line 79 of file bitstream.h.

8.5.4.15 boost::uint8_t* openpr::bitstream::bitstream::mFrameData [protected]

Definition at line 66 of file bitstream.h.

8.5.4.16 openpr::bitstream::device* openpr::bitstream::bitstream::my_dev [protected]

Definition at line 64 of file bitstream.h.

8.5.4.17 int openpr::bitstream::bitstream::num_frames [protected]

Definition at line 67 of file bitstream.h.

8.5.4.18 boost::unordered_map< openpr::bitstream::tile_coord, openpr::bitstream::tile_data*, boost::hash < openpr::bitstream::tile_coord > > openpr::bitstream::bitstream::tile_map [protected]

Definition at line 70 of file bitstream.h.

8.5.4.19 boost::unordered_map< std::string, openpr::bitstream::tile_data*, boost::hash < std::string > > openpr::bitstream::bitstream::tileMap [protected]

Definition at line 71 of file bitstream.h.

The documentation for this class was generated from the following files:

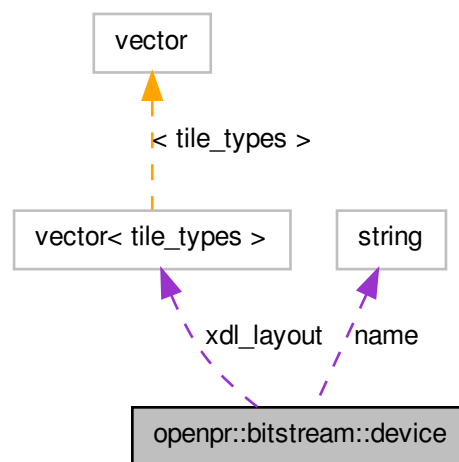
- [openpr/bitstream/bitstream.h](#)
- [openpr/bitstream/bitstream.cpp](#)

8.6 openpr::bitstream::device Class Reference

#include <device.h>

Inherited by [openpr::bitstream::virtex4](#), and [openpr::bitstream::virtex5](#).

Collaboration diagram for openpr::bitstream::device:



Public Member Functions

- `int get_chip_id ()`
- `string get_name () const`
- `int get_num_rows () const`
- `int get_row_width (int type)`
- `int chip_height ()`
- `int get_row_height ()`
- `int chip_width ()`
- `int get_cfg_size (int type)`
- `int get_cfg_size ()`
- `int get_gclk_index ()`
- `tile_types get_tile_type (int x)`
- `int get_blk_type (int x)`
- `int get_blk_type (tile_types tile_type)`
- `virtual int get_addressable_blk_types ()`
- `frame_addr tilecoord_to_major (int x, int y)`

- [frame_addr bramcoord_to_major](#) (int x, int y)
- int [frame_offset](#) ([frame_addr](#) frame_address)
- virtual int [tile_offset](#) (int x, int y)=0
- int [get_tile_frames](#) (int x)
- int [get_tile_frames](#) ([tile_types](#) tile_type)
- int [get_frame_words](#) (void)
- [~device](#) ()

Public Attributes

- const char * [routing_table](#)
- const char * [logic_table](#)

Protected Member Functions

- [device](#) (const int [num_rows](#), const int [num_cols](#), const [tile_types](#) row_layout[], const string name, const int [id](#), const int [frame_words](#), const int [frame_height](#), const int [tile_frames](#)[], const int [block_type](#)[], int [clb_slices](#), int [num_blk_types](#), const char *[routing_table](#), const char *[logic_table](#))
- void [build_xdl_layout](#) ()

Protected Attributes

- const string [name](#)
- const unsigned int [id](#)
- const int [num_rows](#)
- const int [num_cols](#)
- const int [frame_words](#)
- const int [frame_height](#)
- int [tile_width](#)
- const int * [tile_frames](#)
- const int * [block_type](#)
- const int [clb_slices](#)
- const int [num_blk_types](#)
- const [tile_types](#) * [row_layout](#)
- vector< [tile_types](#) > [xdl_layout](#)
- int [gclk_index](#)
- int * [row_width](#)

8.6.1 Detailed Description

Definition at line 29 of file device.h.

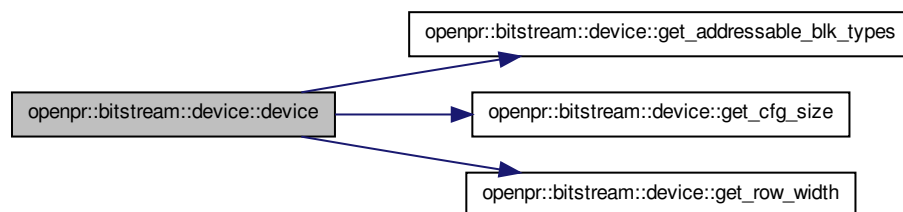
8.6.2 Constructor & Destructor Documentation

8.6.2.1 `openpr::bitstream::device::device (const int num_rows, const int num_cols, const file_types row_layout[], const string name, const int id, const int frame_words, const int frame_height, const int tile_frames[], const int block_type[], int clb_slices, int num_blk_types, const char * routing_table, const char * logic_table) [inline, protected]`

Inline Constructor to initialize const values passed from subclass constructors.

Definition at line 50 of file device.h.

Here is the call graph for this function:



8.6.2.2 `openpr::bitstream::device::~~device () [inline]`

Destructor frees any dynamically allocated memory.

Definition at line 195 of file device.h.

8.6.3 Member Function Documentation

8.6.3.1 `frame_addr openpr::bitstream::device::bramcoord_to_major (int x, int y)`

Translate BRAM XDL Style Tile coordinate to Major Frame Address.

Parameters

x X Coordinate of Tile

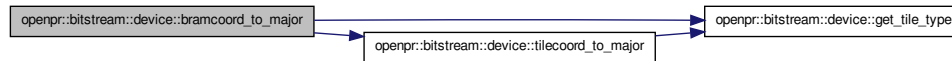
y Y Coordinate of Tile

Todo

Change this so it isn't a dirty hack

Definition at line 143 of file device.cpp.

Here is the call graph for this function:



8.6.3.2 void openpr::bitstream::device::build_xdl_layout () [protected]

Initialize xdl_layout vector to make looking up tile types of XDL coordinates easier.

Definition at line 55 of file device.cpp.

8.6.3.3 int openpr::bitstream::device::chip_height ()

Return height of chip in Tiles

Definition at line 33 of file device.cpp.

8.6.3.4 int openpr::bitstream::device::chip_width ()

Return number of tile columns

Definition at line 37 of file device.cpp.

8.6.3.5 int openpr::bitstream::device::frame_offset (frame_addr frame_address)

Return offset of frame within bitstream. Currently working for CLB Block Types and BRAM INT Type

Parameters

frame_address Frame address to be located

Definition at line 163 of file device.cpp.

8.6.3.6 int openpr::bitstream::device::get_addressable_blk_types () [virtual]

Return the number of addressable block types.

Reimplemented in [openpr::bitstream::virtex4](#), and [openpr::bitstream::virtex5](#).

Definition at line 87 of file device.cpp.

8.6.3.7 int openpr::bitstream::device::get_blk_type (int x)

Return block type given an XDL name X coordinate. Due to overlaps in XDL style tile coordinates, we cannot distinguish between IOBs and GCLKs or BRAM_INTs/BRAMs

Parameters

x X Coordinate of Tile

Definition at line 78 of file device.cpp.

Here is the call graph for this function:



8.6.3.8 int openpr::bitstream::device::get_blk_type (tile_types *tile_type*) [inline]

Return block type given a tiletype.

Parameters

tile_type Enumerated tile_type id.

Definition at line 137 of file device.h.

8.6.3.9 int openpr::bitstream::device::get_cfg_size (int *type*)

Return size of config block for block type (frames).

Parameters

type block type to get configuration size for

Definition at line 51 of file device.cpp.

8.6.3.10 int openpr::bitstream::device::get_cfg_size ()

Return size of configuration memory.

Definition at line 43 of file device.cpp.

8.6.3.11 int openpr::bitstream::device::get_chip_id ()

Return chip ID.

Definition at line 17 of file device.cpp.

8.6.3.12 int openpr::bitstream::device::get_frame_words (void) [inline]

Return the number of words per frame for the architecture.

Returns

Number of words per frame

Definition at line 191 of file device.h.

8.6.3.13 int openpr::bitstream::device::get_gclk_index () [inline]

Return the index of the GCLK tile.

Definition at line 117 of file device.h.

8.6.3.14 string openpr::bitstream::device::get_name () const [inline]

Return device name.

Definition at line 84 of file device.h.

8.6.3.15 int openpr::bitstream::device::get_num_rows () const [inline]

Return number of frame address rows in device.

Definition at line 88 of file device.h.

8.6.3.16 int openpr::bitstream::device::get_row_height () [inline]

Return height of one clock region.

Definition at line 100 of file device.h.

8.6.3.17 int openpr::bitstream::device::get_row_width (int type)

Return Width of Row in Frames.

Definition at line 21 of file device.cpp.

8.6.3.18 int openpr::bitstream::device::get_tile_frames (tile_types tile_type) [inline]

Return number of frames that this tile spans in bitstream.

Parameters

tile_type Tile type enum.

Returns

number of frames that this tile spans in bitstream.

Definition at line 186 of file device.h.

8.6.3.19 int openpr::bitstream::device::get_tile_frames (int x)

Return tile number within column when given byte offset.

Parameters

byte_offset The byte offset within the frame.

Returns

Tile index within column (-1 if in ECC word).

Definition at line 203 of file device.cpp.

Here is the call graph for this function:

**8.6.3.20 tile_types openpr::bitstream::device::get_tile_type (int x)**

Return tile type given an XDL name X coordinate. Due to overlaps in XDL style tile coordinates, cannot we distinguish between IOBs and GCLKs or BRAM_INTs/BRAMs

Parameters

x X Coordinate of Tile

Definition at line 70 of file device.cpp.

8.6.3.21 virtual int openpr::bitstream::device::tile_offset (int x, int y) [pure virtual]

Return byte offset of tile within frames. Only works for CLB Block Types

Parameters

x X coordinate of tile

y Y coordinate of tile

Returns

byte offset of tile within a frame

Implemented in [openpr::bitstream::virtex4](#), and [openpr::bitstream::virtex5](#).

8.6.3.22 frame_addr openpr::bitstream::device::tilecoord_to_major (int x, int y)

Translate XDL Style Tile Coordinate to Major Frame Address. Due to overlaps in XDL style tile coordinates, cannot we distinguish between IOBs and GCLKs or BRAM_INTs/BRAMs

Parameters

x X Coordinate of Tile

y Y Coordinate of Tile

Definition at line 92 of file device.cpp.

Here is the call graph for this function:



8.6.4 Member Data Documentation

8.6.4.1 `const int* openpr::bitstream::device::block_type` `[protected]`

Definition at line 39 of file device.h.

8.6.4.2 `const int openpr::bitstream::device::clb_slices` `[protected]`

Definition at line 40 of file device.h.

8.6.4.3 `const int openpr::bitstream::device::frame_height` `[protected]`

Definition at line 36 of file device.h.

8.6.4.4 `const int openpr::bitstream::device::frame_words` `[protected]`

Definition at line 35 of file device.h.

8.6.4.5 `int openpr::bitstream::device::gclk_index` `[protected]`

Definition at line 44 of file device.h.

8.6.4.6 `const unsigned int openpr::bitstream::device::id` `[protected]`

Definition at line 32 of file device.h.

8.6.4.7 `const char* openpr::bitstream::device::logic_table`

Definition at line 76 of file device.h.

8.6.4.8 `const string openpr::bitstream::device::name` `[protected]`

Definition at line 31 of file device.h.

8.6.4.9 const int openpr::bitstream::device::num_blk_types [protected]

Reimplemented in [openpr::bitstream::virtex4](#).

Definition at line 41 of file device.h.

8.6.4.10 const int openpr::bitstream::device::num_cols [protected]

Definition at line 34 of file device.h.

8.6.4.11 const int openpr::bitstream::device::num_rows [protected]

Definition at line 33 of file device.h.

8.6.4.12 const char* openpr::bitstream::device::routing_table

Definition at line 75 of file device.h.

8.6.4.13 const tile_types* openpr::bitstream::device::row_layout [protected]

Definition at line 42 of file device.h.

8.6.4.14 int* openpr::bitstream::device::row_width [protected]

Definition at line 45 of file device.h.

8.6.4.15 const int* openpr::bitstream::device::tile_frames [protected]

Definition at line 38 of file device.h.

8.6.4.16 int openpr::bitstream::device::tile_width [protected]

Definition at line 37 of file device.h.

8.6.4.17 vector< tile_types > openpr::bitstream::device::xdl_layout [protected]

Definition at line 43 of file device.h.

The documentation for this class was generated from the following files:

- [openpr/bitstream/device.h](#)
- [openpr/bitstream/device.cpp](#)

8.7 openpr::netlist::eq_net Struct Reference

```
#include <NetHashStruct.h>
```

Public Member Functions

- bool [operator\(\)](#) (const [Net](#) *t1, const [Net](#) *t2) const

8.7.1 Detailed Description

Definition at line 26 of file NetHashStruct.h.

8.7.2 Member Function Documentation

8.7.2.1 bool [openpr::netlist::eq_net::operator\(\)](#) (const [Net](#) * *t1*, const [Net](#) * *t2*) const
[[inline](#)]

Definition at line 27 of file NetHashStruct.h.

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

- [openpr/netlist/NetHashStruct.h](#)

8.8 openpr::netlist::eq_pip Struct Reference

```
#include <HashStructs.h>
```

Public Member Functions

- bool [operator\(\)](#) (const [Pin](#) *t1, const [Pin](#) *t2) const

8.8.1 Detailed Description

Definition at line 29 of file HashStructs.h.

8.8.2 Member Function Documentation

8.8.2.1 bool [openpr::netlist::eq_pip::operator\(\)](#) (const [Pin](#) * *t1*, const [Pin](#) * *t2*) const
[[inline](#)]

Definition at line 30 of file HashStructs.h.

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

- [openpr/netlist/HashStructs.h](#)

8.9 openpr::netlist::eq_point Struct Reference

```
#include <HashStructs.h>
```


Public Member Functions

- bool [operator\(\)](#) (const [Point](#) *t1, const [Point](#) *t2) const

8.9.1 Detailed Description

Definition at line 43 of file HashStructs.h.

8.9.2 Member Function Documentation

8.9.2.1 bool openpr::netlist::eq_point::operator() (const [Point](#) * t1, const [Point](#) * t2) const [\[inline\]](#)

Definition at line 44 of file HashStructs.h.

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

- openpr/netlist/[HashStructs.h](#)

8.10 openpr::netlist::eq_segment Struct Reference

```
#include <HashStructs.h>
```

Public Member Functions

- bool [operator\(\)](#) (const [torc::architecture::xilinx::CompactSegmentIndex](#) t1, const [torc::architecture::xilinx::CompactSegmentIndex](#) t2) const

8.10.1 Detailed Description

Definition at line 53 of file HashStructs.h.

8.10.2 Member Function Documentation

8.10.2.1 bool openpr::netlist::eq_segment::operator() (const [torc::architecture::xilinx::CompactSegmentIndex](#) t1, const [torc::architecture::xilinx::CompactSegmentIndex](#) t2) const [\[inline\]](#)

Definition at line 55 of file HashStructs.h.

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

- openpr/netlist/[HashStructs.h](#)

8.11 openpr::bitstream::frame_addr Struct Reference

```
#include <architecture.h>
```

Public Member Functions

- [frame_addr](#) ()
- [frame_addr](#) (int [type](#), int [tb](#), int [row](#), int [col](#), int [mna](#))
- [string str](#) () const

Public Attributes

- int [type](#)
- int [tb](#)
- int [row](#)
- int [col](#)
- int [mna](#)

8.11.1 Detailed Description

Definition at line 39 of file architecture.h.

8.11.2 Constructor & Destructor Documentation

8.11.2.1 `openpr::bitstream::frame_addr::frame_addr () [inline]`

Definition at line 45 of file architecture.h.

8.11.2.2 `openpr::bitstream::frame_addr::frame_addr (int type, int tb, int row, int col, int mna) [inline]`

Initialization constructor.

Parameters

- type* Block type.
- tb* Top bottom indicator.
- row* Row address.
- col* column address.
- mna* minor address.

Definition at line 61 of file architecture.h.

8.11.3 Member Function Documentation

8.11.3.1 `string openpr::bitstream::frame_addr::str () const [inline]`

Definition at line 63 of file architecture.h.

8.11.4 Member Data Documentation

8.11.4.1 int openpr::bitstream::frame_addr::col

Definition at line 43 of file architecture.h.

8.11.4.2 int openpr::bitstream::frame_addr::mna

Definition at line 44 of file architecture.h.

8.11.4.3 int openpr::bitstream::frame_addr::row

Definition at line 42 of file architecture.h.

8.11.4.4 int openpr::bitstream::frame_addr::tb

Definition at line 41 of file architecture.h.

8.11.4.5 int openpr::bitstream::frame_addr::type

Definition at line 40 of file architecture.h.

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

- openpr/bitstream/[architecture.h](#)

8.12 openpr::netlist::hash_net Struct Reference

```
#include <NetHashStruct.h>
```

Public Member Functions

- `size_t operator() (const Net *t) const`

8.12.1 Detailed Description

Definition at line 21 of file NetHashStruct.h.

8.12.2 Member Function Documentation

8.12.2.1 `size_t openpr::netlist::hash_net::operator() (const Net * t) const [inline]`

Definition at line 22 of file NetHashStruct.h.

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

- openpr/netlist/[NetHashStruct.h](#)

8.13 openpr::netlist::hash_pip Struct Reference

```
#include <HashStructs.h>
```

Public Member Functions

- `size_t operator() (const Pin *t) const`

8.13.1 Detailed Description

Definition at line 23 of file HashStructs.h.

8.13.2 Member Function Documentation

8.13.2.1 `size_t openpr::netlist::hash_pip::operator() (const Pin * t) const` [`inline`]

Definition at line 24 of file HashStructs.h.

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

- openpr/netlist/[HashStructs.h](#)

8.14 openpr::netlist::hash_point Struct Reference

```
#include <HashStructs.h>
```

Public Member Functions

- `size_t operator() (const Point *t) const`

8.14.1 Detailed Description

Definition at line 38 of file HashStructs.h.

8.14.2 Member Function Documentation

8.14.2.1 `size_t openpr::netlist::hash_point::operator() (const Point * t) const` [`inline`]

Definition at line 39 of file HashStructs.h.

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

- openpr/netlist/[HashStructs.h](#)

8.15 openpr::netlist::hash_segment Struct Reference

```
#include <HashStructs.h>
```

Public Member Functions

- `size_t operator() (const torc::architecture::xilinx::CompactSegmentIndex t) const`

8.15.1 Detailed Description

Definition at line 48 of file HashStructs.h.

8.15.2 Member Function Documentation

8.15.2.1 `size_t openpr::netlist::hash_segment::operator() (const torc::architecture::xilinx::CompactSegmentIndex t) const` `[inline]`

Definition at line 49 of file HashStructs.h.

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

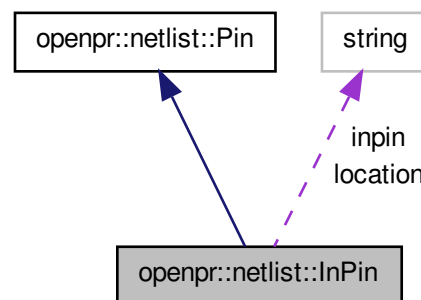
- openpr/netlist/[HashStructs.h](#)

8.16 openpr::netlist::InPin Class Reference

```
#include <InPin.h>
```

Inherits [openpr::netlist::Pin](#).

Collaboration diagram for openpr::netlist::InPin:



Public Member Functions

- `InPin (std::string inpin, std::string location)`
- `virtual ~InPin ()`
- `void printData (std::ofstream *outStream)`
- `void printData (std::ostream *outStream)`

- `bool operator!= (Pin &other)`
- `bool operator!= (Pin *other)`
- `bool operator== (const Pin &other) const`
- `bool operator== (const Pin *other) const`
- `bool operator< (Pin &other)`
- `bool operator< (Pin *other)`
- `bool operator> (Pin &other)`
- `bool operator> (Pin *other)`
- `Pin & operator= (Pin &other)`
- `Pin * operator= (Pin *other)`
- `size_t operator() (const Pin &x) const`
- `size_t operator() (const Pin *x) const`

Private Attributes

- `std::string inpin`
- `std::string location`

8.16.1 Detailed Description

Definition at line 22 of file InPin.h.

8.16.2 Constructor & Destructor Documentation

8.16.2.1 `openpr::netlist::InPin::InPin (std::string inpin, std::string location)`

Definition at line 22 of file InPin.cpp.

8.16.2.2 `openpr::netlist::InPin::~InPin () [virtual]`

Definition at line 28 of file InPin.cpp.

8.16.3 Member Function Documentation

8.16.3.1 `bool openpr::netlist::InPin::operator!= (Pin & other) [virtual]`

Implements `openpr::netlist::Pin`.

Definition at line 38 of file InPin.cpp.

8.16.3.2 `bool openpr::netlist::InPin::operator!= (Pin * other) [virtual]`

Implements `openpr::netlist::Pin`.

Definition at line 123 of file InPin.cpp.

8.16.3.3 `std::size_t openpr::netlist::InPin::operator()(const Pin & x) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 110 of file InPin.cpp.

8.16.3.4 `std::size_t openpr::netlist::InPin::operator()(const Pin * x) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 198 of file InPin.cpp.

8.16.3.5 `bool openpr::netlist::InPin::operator<(Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 138 of file InPin.cpp.

8.16.3.6 `bool openpr::netlist::InPin::operator<(Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 53 of file InPin.cpp.

8.16.3.7 `Pin & openpr::netlist::InPin::operator=(Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 70 of file InPin.cpp.

8.16.3.8 `Pin * openpr::netlist::InPin::operator=(Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 155 of file InPin.cpp.

8.16.3.9 `bool openpr::netlist::InPin::operator==(const Pin * other) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 167 of file InPin.cpp.

8.16.3.10 `bool openpr::netlist::InPin::operator==(const Pin & other) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 82 of file InPin.cpp.

8.16.3.11 `bool openpr::netlist::InPin::operator>(Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 94 of file InPin.cpp.

8.16.3.12 `bool openpr::netlist::InPin::operator> (Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 182 of file InPin.cpp.

8.16.3.13 `void openpr::netlist::InPin::printData (std::ofstream * outStream) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 32 of file InPin.cpp.

8.16.3.14 `void openpr::netlist::InPin::printData (std::ostream * outStream) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 35 of file InPin.cpp.

8.16.4 Member Data Documentation

8.16.4.1 `std::string openpr::netlist::InPin::inpin [private]`

Definition at line 41 of file InPin.h.

8.16.4.2 `std::string openpr::netlist::InPin::location [private]`

Definition at line 42 of file InPin.h.

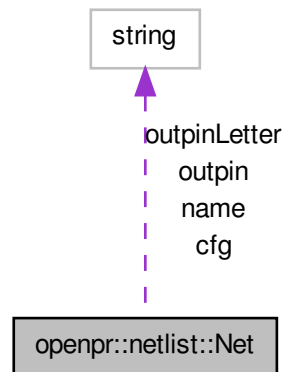
The documentation for this class was generated from the following files:

- [openpr/netlist/InPin.h](#)
- [openpr/netlist/InPin.cpp](#)

8.17 openpr::netlist::Net Class Reference

```
#include <Net.h>
```


Collaboration diagram for openpr::netlist::Net:



Public Member Functions

- [Net](#) ()
- [~Net](#) ()
- [Net](#) (std::string *inputString)
- [Net](#) (const [Net](#) *other)
- void [addCfg](#) (std::string cfg)
- void [addInPin](#) ([InPin](#) *inpin)
- void [addOutPin](#) ([Pin](#) *outpin)
- void [insertPip](#) ([Pip](#) *pip)
- void [remotePin](#) ([Pin](#) *tempPin, int type)
- void [clearInPins](#) ()
- void [printData](#) (std::ofstream *outstream)
- void [printData](#) (std::ostream *outstream)
- bool [operator!=](#) (const [Net](#) &other) const
- bool [operator==](#) (const [Net](#) &other) const
- bool [operator<](#) (const [Net](#) &other) const
- bool [operator>](#) (const [Net](#) &other) const
- [Net](#) & [operator=](#) (const [Net](#) &other)
- bool [operator!=](#) (const [Net](#) *other) const
- bool [operator==](#) (const [Net](#) *other) const
- bool [operator<](#) (const [Net](#) *other) const
- bool [operator>](#) (const [Net](#) *other) const
- [Net](#) & [operator=](#) (const [Net](#) *other)
- void [mergePips](#) (const [Net](#) &other)
- std::size_t [operator\(\)](#) (const [Net](#) *&other) const
- std::size_t [operator\(\)](#) (const [Net](#) &other) const
- std::size_t [hash_value](#) (const [Net](#) &net) const
- std::size_t [hash_value](#) (const [Net](#) *net) const

Public Attributes

- `std::string` [name](#)

Private Attributes

- `boost::unordered_set< Pin *, hash_pip, eq_pip >` [pips](#)
- `boost::unordered_set< Pin *, hash_pip, eq_pip >` [outPins](#)
- `boost::unordered_set< Pin *, hash_pip, eq_pip >` [inPins](#)
- `std::string` [outpin](#)
- `std::string` [outpinLetter](#)
- `std::string` [cfg](#)
- `bool` [configure](#)

8.17.1 Detailed Description

Definition at line 26 of file `Net.h`.

8.17.2 Constructor & Destructor Documentation

8.17.2.1 `openpr::netlist::Net::Net ()`

Definition at line 23 of file `Net.cpp`.

8.17.2.2 `openpr::netlist::Net::~~Net ()`

Definition at line 27 of file `Net.cpp`.

8.17.2.3 `openpr::netlist::Net::Net (std::string * inputString)`

Definition at line 37 of file `Net.cpp`.

8.17.2.4 `openpr::netlist::Net::Net (const Net * other)`

Definition at line 65 of file `Net.cpp`.

8.17.3 Member Function Documentation

8.17.3.1 `void openpr::netlist::Net::addCfg (std::string cfg)`

Definition at line 69 of file `Net.cpp`.

8.17.3.2 `void openpr::netlist::Net::addInPin (InPin * inpin)`

Definition at line 74 of file `Net.cpp`.

8.17.3.3 void openpr::netlist::Net::addOutPin (Pin * *outpin*)

Definition at line 77 of file Net.cpp.

8.17.3.4 void openpr::netlist::Net::clearInPins ()

Definition at line 300 of file Net.cpp.

8.17.3.5 std::size_t openpr::netlist::Net::hash_value (const Net & *net*) const

Definition at line 207 of file Net.cpp.

8.17.3.6 std::size_t openpr::netlist::Net::hash_value (const Net * *net*) const

Definition at line 212 of file Net.cpp.

8.17.3.7 void openpr::netlist::Net::insertPip (Pip * *pip*)

Definition at line 80 of file Net.cpp.

8.17.3.8 void openpr::netlist::Net::mergePips (const Net & *other*)

Definition at line 198 of file Net.cpp.

8.17.3.9 bool openpr::netlist::Net::operator!= (const Net & *other*) const

only compares net name

Definition at line 151 of file Net.cpp.

8.17.3.10 bool openpr::netlist::Net::operator!= (const Net * *other*) const

Definition at line 217 of file Net.cpp.

8.17.3.11 std::size_t openpr::netlist::Net::operator() (const Net & *other*) const

Definition at line 270 of file Net.cpp.

8.17.3.12 std::size_t openpr::netlist::Net::operator() (const Net *& *other*) const

Definition at line 265 of file Net.cpp.

8.17.3.13 bool openpr::netlist::Net::operator< (const Net * *other*) const

Definition at line 229 of file Net.cpp.

8.17.3.14 `bool openpr::netlist::Net::operator< (const Net & other) const`

Definition at line 163 of file Net.cpp.

8.17.3.15 `Net & openpr::netlist::Net::operator= (const Net * other)`

Definition at line 241 of file Net.cpp.

8.17.3.16 `Net & openpr::netlist::Net::operator= (const Net & other)`

Definition at line 175 of file Net.cpp.

8.17.3.17 `bool openpr::netlist::Net::operator== (const Net & other) const`

Definition at line 157 of file Net.cpp.

8.17.3.18 `bool openpr::netlist::Net::operator== (const Net * other) const`

Definition at line 223 of file Net.cpp.

8.17.3.19 `bool openpr::netlist::Net::operator> (const Net * other) const`

Definition at line 235 of file Net.cpp.

8.17.3.20 `bool openpr::netlist::Net::operator> (const Net & other) const`

Definition at line 169 of file Net.cpp.

8.17.3.21 `void openpr::netlist::Net::printData (std::ostream * ostream)`

Definition at line 116 of file Net.cpp.

Here is the call graph for this function:



8.17.3.22 void openpr::netlist::Net::printData (std::ostream * *outstream*)

Definition at line 83 of file Net.cpp.

Here is the call graph for this function:

**8.17.3.23 void openpr::netlist::Net::remotePin (Pin * *tempPin*, int *type*)**

Definition at line 275 of file Net.cpp.

8.17.4 Member Data Documentation**8.17.4.1 std::string openpr::netlist::Net::cfg [private]**

Definition at line 62 of file Net.h.

8.17.4.2 bool openpr::netlist::Net::configure [private]

Definition at line 63 of file Net.h.

8.17.4.3 boost::unordered_set<Pin*, hash_pip, eq_pip> openpr::netlist::Net::inPins [private]

Definition at line 59 of file Net.h.

8.17.4.4 std::string openpr::netlist::Net::name

Definition at line 55 of file Net.h.

8.17.4.5 std::string openpr::netlist::Net::outpin [private]

Definition at line 60 of file Net.h.

8.17.4.6 std::string openpr::netlist::Net::outpinLetter [private]

Definition at line 61 of file Net.h.

8.17.4.7 `boost::unordered_set<Pin*, hash_pip, eq_pip> openpr::netlist::Net::outPins` `[private]`

Definition at line 58 of file Net.h.

8.17.4.8 `boost::unordered_set<Pin*, hash_pip, eq_pip> openpr::netlist::Net::pips` `[private]`

Definition at line 57 of file Net.h.

The documentation for this class was generated from the following files:

- [openpr/netlist/Net.h](#)
- [openpr/netlist/Net.cpp](#)

8.18 `openpr::netlist::NetList` Class Reference

```
#include <NetList.h>
```

Public Member Functions

- [NetList](#) ()
- [NetList](#) (std::string inputFile, torc::architecture::DDB *mDB)
- [NetList](#) (std::string inputFile, std::string outputFile, torc::architecture::DDB *mDB)
- [~NetList](#) ()
- void [printData](#) ()
- void [printData](#) (std::ofstream *ofstream)
- [Net](#) * [findNet](#) ([Net](#) *input)
- [Net](#) * [findNet](#) (std::string input)
- [Net](#) * [findNet](#) ([Pin](#) *input)
- [Pin](#) * [findPin](#) ([Pin](#) *input)
- [Pip](#) * [findPip](#) ([Point](#) *input)
- int [insertNet](#) ([Net](#) *inputNet)
- int [insertPip](#) ([Net](#) *inputNet, [Pip](#) *inputPip)
- int [removePip](#) ([Pip](#) *inputPip)
- std::vector< [Net](#) * > * [getNetList](#) ()
- boost::unordered_map< [Pin](#) *, [Net](#) *, [hash_pip](#), [eq_pip](#) > * [getPipToNet](#) ()

Public Attributes

- std::ofstream * [outputXDL](#)

Private Member Functions

- void [topLevelParser](#) (std::ifstream *fStream)
- void [netParser](#) (std::ifstream *fin, std::string *netString)

Private Attributes

- `std::vector< Net * > netList`
- `boost::unordered_set< Net *, hash_net, eq_net > netToPip`
- `boost::unordered_map< Pin *, Net *, hash_pip, eq_pip > pipToNet`
- `boost::unordered_map< Point *, Pip *, hash_point, eq_point > pointToPip`
- `boost::unordered_map< torc::architecture::xilinx::CompactSegmentIndex, Net *, hash_segment, eq_segment > segmentToNet`
- `torc::architecture::DDB * mDB`

8.18.1 Detailed Description

Definition at line 26 of file NetList.h.

8.18.2 Constructor & Destructor Documentation

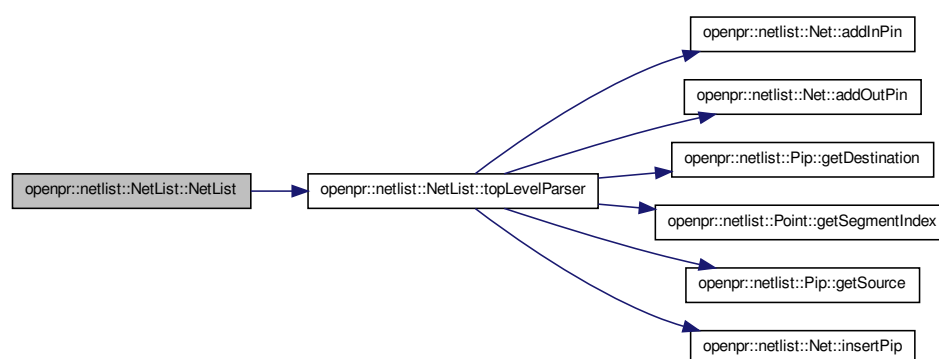
8.18.2.1 openpr::netlist::NetList::NetList ()

Definition at line 25 of file NetList.cpp.

8.18.2.2 openpr::netlist::NetList::NetList (std::string *inputFile*, torc::architecture::DDB * *mDB*)

Definition at line 37 of file NetList.cpp.

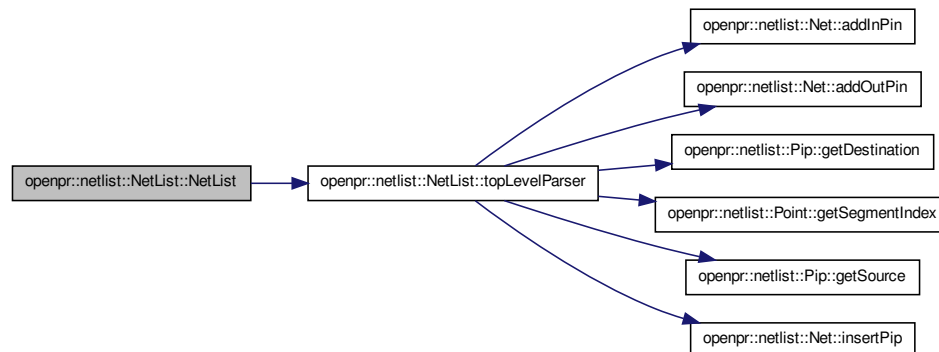
Here is the call graph for this function:



8.18.2.3 openpr::netlist::NetList::NetList (std::string *inputFile*, std::string *outputFile*, torc::architecture::DDB * *mDB*)

Definition at line 48 of file NetList.cpp.

Here is the call graph for this function:



8.18.2.4 `openpr::netlist::NetList::~~NetList ()`

Definition at line 29 of file `NetList.cpp`.

8.18.3 Member Function Documentation

8.18.3.1 `Net * openpr::netlist::NetList::findNet (Net * input)`

Definition at line 235 of file `NetList.cpp`.

8.18.3.2 `Net * openpr::netlist::NetList::findNet (std::string input)`

Definition at line 246 of file `NetList.cpp`.

Here is the call graph for this function:



8.18.3.3 `Net * openpr::netlist::NetList::findNet (Pin * input)`

Definition at line 251 of file `NetList.cpp`.

8.18.3.4 `Pin * openpr::netlist::NetList::findPin (Pin * input)`

Definition at line 260 of file NetList.cpp.

8.18.3.5 `Pip* openpr::netlist::NetList::findPip (Point * input)`**8.18.3.6** `std::vector< Net * > * openpr::netlist::NetList::getNetList ()`

Definition at line 341 of file NetList.cpp.

8.18.3.7 `boost::unordered_map< Pin *, Net *, hash_pip, eq_pip > *
openpr::netlist::NetList::getPipToNet ()`

Definition at line 344 of file NetList.cpp.

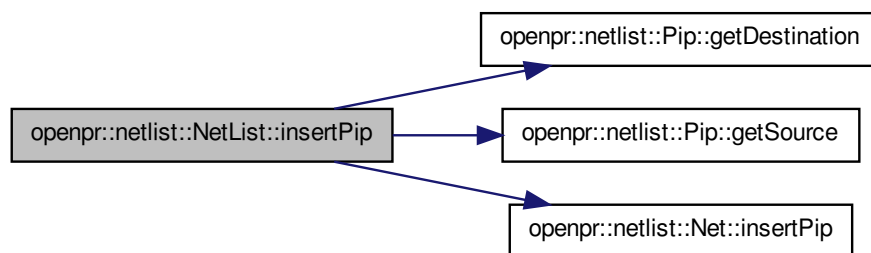
8.18.3.8 `int openpr::netlist::NetList::insertNet (Net * inputNet)`

Definition at line 270 of file NetList.cpp.

8.18.3.9 `int openpr::netlist::NetList::insertPip (Net * inputNet, Pip * inputPip)`

Definition at line 276 of file NetList.cpp.

Here is the call graph for this function:

**8.18.3.10** `void openpr::netlist::NetList::netParser (std::ifstream * fin, std::string * netString)
[private]`**8.18.3.11** `void openpr::netlist::NetList::printData (std::ofstream * ofstream)`

Definition at line 229 of file NetList.cpp.

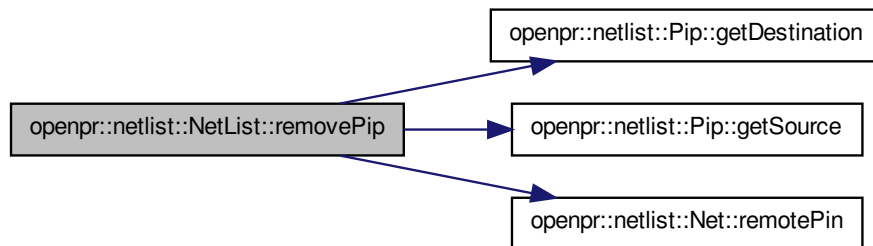
8.18.3.12 void openpr::netlist::NetList::printData ()

Definition at line 225 of file NetList.cpp.

8.18.3.13 int openpr::netlist::NetList::removePip (Pip * *inputPip*)

Definition at line 305 of file NetList.cpp.

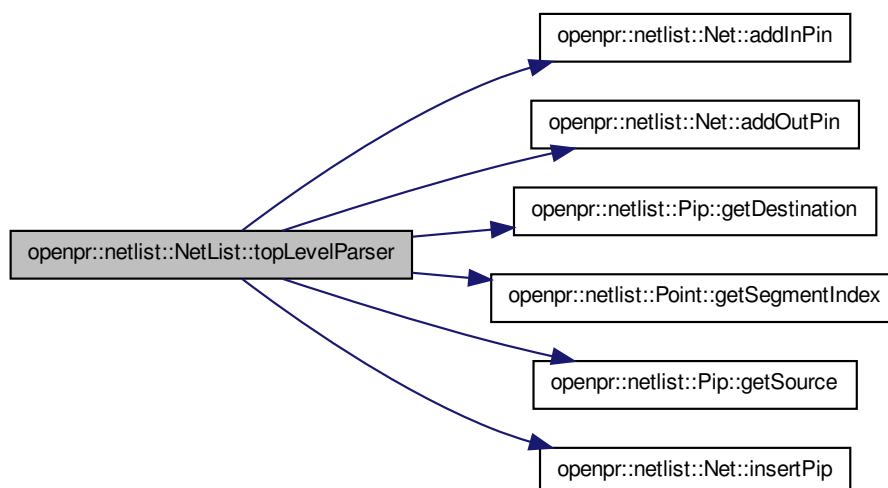
Here is the call graph for this function:



8.18.3.14 void openpr::netlist::NetList::topLevelParser (std::ifstream * *fStream*) [private]

Definition at line 59 of file NetList.cpp.

Here is the call graph for this function:



8.18.4 Member Data Documentation

8.18.4.1 `torc::architecture::DDB* openpr::netlist::NetList::mDB` `[private]`

Definition at line 55 of file NetList.h.

8.18.4.2 `std::vector<Net*> openpr::netlist::NetList::netList` `[private]`

Definition at line 50 of file NetList.h.

8.18.4.3 `boost::unordered_set<Net*, hash_net, eq_net> openpr::netlist::NetList::netToPip` `[private]`

Definition at line 51 of file NetList.h.

8.18.4.4 `std::ofstream* openpr::netlist::NetList::outputXDL`

Definition at line 35 of file NetList.h.

8.18.4.5 `boost::unordered_map<Pin*, Net*, hash_pip, eq_pip>` `openpr::netlist::NetList::pipToNet` `[private]`

Definition at line 52 of file NetList.h.

8.18.4.6 `boost::unordered_map<Point*, Pip*, hash_point, eq_point>` `openpr::netlist::NetList::pointToPip` `[private]`

Definition at line 53 of file NetList.h.

8.18.4.7 `boost::unordered_map<torc::architecture::xilinx::CompactSegmentIndex, Net*,` `hash_segment, eq_segment> openpr::netlist::NetList::segmentToNet` `[private]`

Definition at line 54 of file NetList.h.

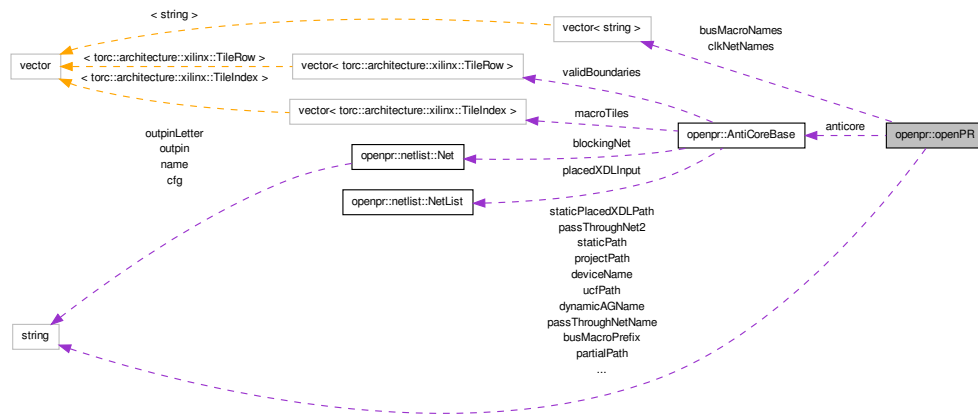
The documentation for this class was generated from the following files:

- [openpr/netlist/NetList.h](#)
- [openpr/netlist/NetList.cpp](#)

8.19 openpr::openPR Class Reference

```
#include <OpenPR.hpp>
```

Collaboration diagram for openpr::openPR:



Public Member Functions

- [openPR \(\)](#)
- [~openPR \(\)](#)
- void [buildRelativePaths \(\)](#)
- void [setupAntiCore \(\)](#)
- void [setupDynamicRegion \(\)](#)
- bool [genPlaceConstraints \(\)](#)
- bool [placeMacros \(\)](#)
- bool [siteBlocker \(\)](#)
- bool [routeBlocker \(\)](#)
- bool [mergeClockTree \(\)](#)
- std::string [buildBlockingNet \(\)](#)
- bool [genPassThroughScripts \(\)](#)
- bool [genLockConstraints \(\)](#)
- bool [genPartialBitstream \(\)](#)

Private Member Functions

- template<class Archive >
void [serialize](#) (Archive &ar, const unsigned int version)

Private Attributes

- [string designName](#)
- [string projectPath](#)
- [string dynamicAGName](#)
- [string staticPath](#)
- [string partialPath](#)
- [string ucfPath](#)

- [string staticPlacedXDLPath](#)
- [string deviceName](#)
- [string busMacroPrefix](#)
- [string passThroughNetName](#)
- [string passThroughNet2](#)
- [vector< string > busMacroNames](#)
- [vector< string > clkNetNames](#)
- [bool regionDefined](#)
- [bool isPartial](#)
- [int busWidth](#)
- [torc::architecture::xilinx::TileRow yMin](#)
- [torc::architecture::xilinx::TileRow yMax](#)
- [torc::architecture::xilinx::TileCol xMin](#)
- [torc::architecture::xilinx::TileCol xMax](#)
- [boost::uint16_t l_yMin](#)
- [boost::uint16_t l_yMax](#)
- [boost::uint16_t l_xMin](#)
- [boost::uint16_t l_xMax](#)
- [torc::architecture::DDB * db](#)
- [openpr::AntiCoreBase * anticore](#)
- [const boost::filesystem::path buildPath](#)
- [boost::filesystem::path placedXdlPath](#)
- [boost::filesystem::path blockedXdlPath](#)
- [boost::filesystem::path mergedXdlPath](#)
- [boost::filesystem::path routedXdlPath](#)
- [boost::filesystem::path fullXdlPath](#)
- [boost::filesystem::path fullBsPath](#)
- [boost::filesystem::path partialBsPath](#)
- [boost::filesystem::path pcfPath](#)
- [boost::filesystem::path fullUcfPath](#)
- [const boost::filesystem::path busMacroPath](#)
- [const boost::filesystem::path routePTScriptPath](#)
- [const boost::filesystem::path unroutePTScriptPath](#)

Friends

- [class boost::serialization::access](#)

8.19.1 Detailed Description

Definition at line 33 of file OpenPR.hpp.

8.19.2 Constructor & Destructor Documentation

8.19.2.1 openpr::openPR::openPR ()

Default constructor.

Definition at line 20 of file OpenPR.cpp.

8.19.2.2 openpr::openPR::~~openPR ()

Definition at line 30 of file OpenPR.cpp.

8.19.3 Member Function Documentation

8.19.3.1 std::string openpr::openPR::buildBlockingNet ()

Take prefix of blocking net and create the full net name that can be blocked.

Definition at line 292 of file OpenPR.cpp.

8.19.3.2 void openpr::openPR::buildRelativePaths ()

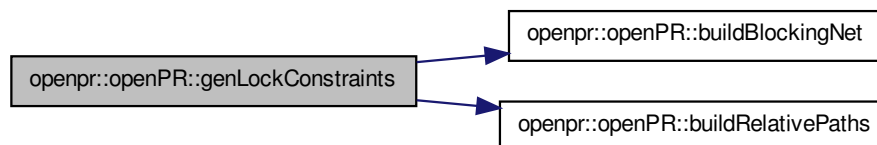
Definition at line 33 of file OpenPR.cpp.

8.19.3.3 bool openpr::openPR::genLockConstraints ()

Generate constraints that lock both busmacro nets and blocking nets

Definition at line 340 of file OpenPR.cpp.

Here is the call graph for this function:



8.19.3.4 bool openpr::openPR::genPartialBitstream ()

Generate a partial bitstream for the defined region from a full bitstream of the partial design.

Definition at line 373 of file OpenPR.cpp.

Generate passthrough scripts automatically.

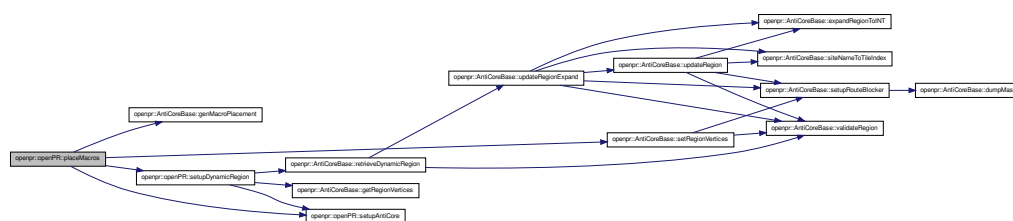
Here is the call graph for this function:



Generate all placement constraints and link appropriate bus macro files.

Generated on Mon Apr 25 2011 18:27:05 for OpenPR by Doxygen

Here is the call graph for this function:

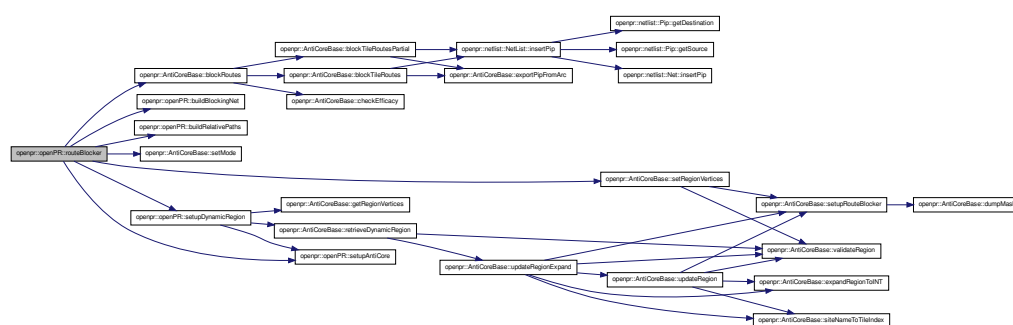


8.19.3.9 bool openpr::openPR::routeBlocker ()

Generate blocking routes in design.

Definition at line 219 of file OpenPR.cpp.

Here is the call graph for this function:



8.19.3.10 template<class Archive > void openpr::openPR::serialize (Archive & ar, const unsigned int version) [inline, private]

Constant defining relative path to ucFile const boost::filesystem::path ucFilePath; Use boost.serialize to define an XML archive for the OpenPR class.

Parameters

ar Archive template class.

version Defines which version of the archive this is, currently unused.

Definition at line 96 of file OpenPR.hpp.

8.19.3.11 void openpr::openPR::setupAntiCore ()

Setup CAntiCore object depending on the device type.

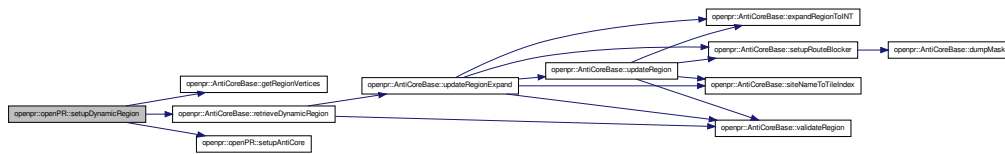
Definition at line 54 of file OpenPR.cpp.

8.19.3.12 void openpr::openPR::setupDynamicRegion ()

Setup dimensions of dynamic region based on AREA_GROUPS defined in UCF file.

Definition at line 82 of file OpenPR.cpp.

Here is the call graph for this function:

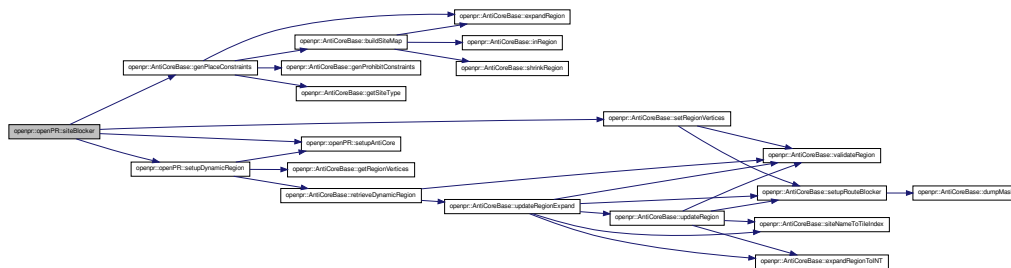


8.19.3.13 bool openpr::openPR::siteBlocker ()

Generate constraints that block placement within sites.

Definition at line 192 of file OpenPR.cpp.

Here is the call graph for this function:



8.19.4 Friends And Related Function Documentation

8.19.4.1 friend class boost::serialization::access [friend]

Definition at line 34 of file OpenPR.hpp.

8.19.5 Member Data Documentation

8.19.5.1 openpr::AntiCoreBase* openpr::openPR::anticore [private]

Definition at line 61 of file OpenPR.hpp.

8.19.5.2 boost::filesystem::path openpr::openPR::blockedXdlPath [private]

Constant defining the relative path of the routeBlocked xdl file.

Definition at line 67 of file OpenPR.hpp.

8.19.5.3 const boost::filesystem::path openpr::openPR::buildPath [private]

Constant defining relative location of build path

Definition at line 63 of file OpenPR.hpp.

8.19.5.4 vector<string> openpr::openPR::busMacroNames [private]

Definition at line 47 of file OpenPR.hpp.

8.19.5.5 const boost::filesystem::path openpr::openPR::busMacroPath [private]

Constant defining the relative path of the busmacros

Definition at line 83 of file OpenPR.hpp.

8.19.5.6 string openpr::openPR::busMacroPrefix [private]

Definition at line 44 of file OpenPR.hpp.

8.19.5.7 int openpr::openPR::busWidth [private]

Definition at line 51 of file OpenPR.hpp.

8.19.5.8 vector<string> openpr::openPR::clkNetNames [private]

Definition at line 48 of file OpenPR.hpp.

8.19.5.9 torc::architecture::DDB* openpr::openPR::db [private]

Definition at line 60 of file OpenPR.hpp.

8.19.5.10 string openpr::openPR::designName [private]

Definition at line 36 of file OpenPR.hpp.

8.19.5.11 string openpr::openPR::deviceName [private]

Definition at line 43 of file OpenPR.hpp.

8.19.5.12 string openpr::openPR::dynamicAGName [private]

Definition at line 38 of file OpenPR.hpp.

8.19.5.13 boost::filesystem::path openpr::openPR::fullBsPath [private]

Constant defining the relative path of the full bitstream file.

Definition at line 75 of file OpenPR.hpp.

8.19.5.14 boost::filesystem::path openpr::openPR::fullUcfPath [private]

Constant defining the relative path of the UCF file

Definition at line 81 of file OpenPR.hpp.

8.19.5.15 boost::filesystem::path openpr::openPR::fullXdlPath [private]

Constant defining the relative path of the routeBlocked xdl file.

Definition at line 73 of file OpenPR.hpp.

8.19.5.16 bool openpr::openPR::isPartial [private]

Definition at line 50 of file OpenPR.hpp.

8.19.5.17 boost::uint16_t openpr::openPR::l_xMax [private]

Definition at line 59 of file OpenPR.hpp.

8.19.5.18 boost::uint16_t openpr::openPR::l_xMin [private]

Definition at line 58 of file OpenPR.hpp.

8.19.5.19 boost::uint16_t openpr::openPR::l_yMax [private]

Definition at line 57 of file OpenPR.hpp.

8.19.5.20 boost::uint16_t openpr::openPR::l_yMin [private]

Definition at line 56 of file OpenPR.hpp.

8.19.5.21 boost::filesystem::path openpr::openPR::mergedXdlPath [private]

Constant defining the relative path of the routeBlocked xdl file.

Definition at line 69 of file OpenPR.hpp.

8.19.5.22 boost::filesystem::path openpr::openPR::partialBsPath [private]

Constant defining the relative path of the partial bitstream file.

Definition at line 77 of file OpenPR.hpp.

8.19.5.23 string openpr::openPR::partialPath [private]

Definition at line 40 of file OpenPR.hpp.

8.19.5.24 string openpr::openPR::passThroughNet2 [private]

Definition at line 46 of file OpenPR.hpp.

8.19.5.25 string openpr::openPR::passThroughNetName [private]

Definition at line 45 of file OpenPR.hpp.

8.19.5.26 boost::filesystem::path openpr::openPR::pcfPath [private]

Constant defining the relative path of the Physical constraints file

Definition at line 79 of file OpenPR.hpp.

8.19.5.27 boost::filesystem::path openpr::openPR::placedXdlPath [private]

Constant defining the relative path of a placed xdl file.

Definition at line 65 of file OpenPR.hpp.

8.19.5.28 string openpr::openPR::projectPath [private]

Definition at line 37 of file OpenPR.hpp.

8.19.5.29 bool openpr::openPR::regionDefined [private]

Definition at line 49 of file OpenPR.hpp.

8.19.5.30 boost::filesystem::path openpr::openPR::routedXdlPath [private]

Constant defining the relative path of the routeBlocked xdl file.

Definition at line 71 of file OpenPR.hpp.

8.19.5.31 const boost::filesystem::path openpr::openPR::routePTScriptPath [private]

Constant defining the relative path of the route passthrough script

Definition at line 85 of file OpenPR.hpp.

8.19.5.32 string openpr::openPR::staticPath [private]

Definition at line 39 of file OpenPR.hpp.

8.19.5.33 string openpr::openPR::staticPlacedXDLPath [private]

Definition at line 42 of file OpenPR.hpp.

8.19.5.34 string openpr::openPR::ucfPath [private]

Definition at line 41 of file OpenPR.hpp.

8.19.5.35 const boost::filesystem::path openpr::openPR::unroutePTScriptPath [private]

Constant defining the relative path of the unroute passthrough script

Definition at line 87 of file OpenPR.hpp.

8.19.5.36 torc::architecture::xilinx::TileCol openpr::openPR::xMax [private]

Definition at line 55 of file OpenPR.hpp.

8.19.5.37 torc::architecture::xilinx::TileCol openpr::openPR::xMin [private]

Definition at line 54 of file OpenPR.hpp.

8.19.5.38 torc::architecture::xilinx::TileRow openpr::openPR::yMax [private]

Definition at line 53 of file OpenPR.hpp.

8.19.5.39 torc::architecture::xilinx::TileRow openpr::openPR::yMin [private]

Definition at line 52 of file OpenPR.hpp.

The documentation for this class was generated from the following files:

- [openpr/anticore/OpenPR.hpp](#)
- [openpr/anticore/OpenPR.cpp](#)

8.20 openpr::OpenPRTree Class Reference

```
#include <OpenPRTree.hpp>
```

Public Member Functions

- [OpenPRTree](#) (const char *argv0)

Static Public Member Functions

- static const boost::filesystem::path & [relativePath](#) (void)
- static const boost::filesystem::path & [workingPath](#) (void)

- static const boost::filesystem::path & [executablePath](#) (void)
- static const boost::filesystem::path & [edaPath](#) (void)
- static const boost::filesystem::path & [databasePath](#) (void)
- static const boost::filesystem::path & [logPath](#) (void)

Static Protected Attributes

- static boost::filesystem::path [sRelativePath](#)
- static boost::filesystem::path [sWorkingPath](#)
- static boost::filesystem::path [sExecutablePath](#)
- static boost::filesystem::path [sEdaPath](#)
- static boost::filesystem::path [sDatabasePath](#)
- static boost::filesystem::path [sLogPath](#)

8.20.1 Detailed Description

Definition at line 19 of file OpenPRTree.hpp.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 openpr::OpenPRTree::OpenPRTree (const char * argv0)

Definition at line 26 of file OpenPRTree.cpp.

8.20.3 Member Function Documentation

8.20.3.1 static const boost::filesystem::path& openpr::OpenPRTree::databasePath (void) [inline, static]

Definition at line 44 of file OpenPRTree.hpp.

8.20.3.2 static const boost::filesystem::path& openpr::OpenPRTree::edaPath (void) [inline, static]

Definition at line 41 of file OpenPRTree.hpp.

8.20.3.3 static const boost::filesystem::path& openpr::OpenPRTree::executablePath (void) [inline, static]

Definition at line 38 of file OpenPRTree.hpp.

8.20.3.4 static const boost::filesystem::path& openpr::OpenPRTree::logPath (void) [inline, static]

Definition at line 47 of file OpenPRTree.hpp.

8.20.3.5 `static const boost::filesystem::path& openpr::OpenPRTree::relativePath (void)`
`[inline, static]`

Definition at line 32 of file OpenPRTree.hpp.

8.20.3.6 `static const boost::filesystem::path& openpr::OpenPRTree::workingPath (void)`
`[inline, static]`

Definition at line 35 of file OpenPRTree.hpp.

8.20.4 Member Data Documentation

8.20.4.1 `boost::filesystem::path openpr::OpenPRTree::sDatabasePath` `[static, protected]`

Definition at line 26 of file OpenPRTree.hpp.

8.20.4.2 `boost::filesystem::path openpr::OpenPRTree::sEdaPath` `[static, protected]`

Definition at line 25 of file OpenPRTree.hpp.

8.20.4.3 `boost::filesystem::path openpr::OpenPRTree::sExecutablePath` `[static, protected]`

Definition at line 24 of file OpenPRTree.hpp.

8.20.4.4 `boost::filesystem::path openpr::OpenPRTree::sLogPath` `[static, protected]`

Definition at line 27 of file OpenPRTree.hpp.

8.20.4.5 `boost::filesystem::path openpr::OpenPRTree::sRelativePath` `[static, protected]`

Definition at line 22 of file OpenPRTree.hpp.

8.20.4.6 `boost::filesystem::path openpr::OpenPRTree::sWorkingPath` `[static, protected]`

Definition at line 23 of file OpenPRTree.hpp.

The documentation for this class was generated from the following files:

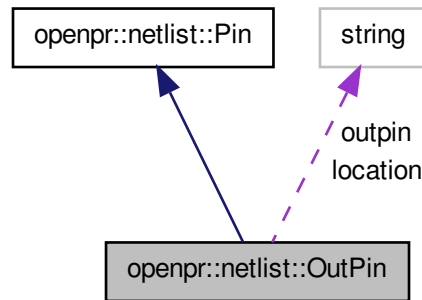
- openpr/anticore/[OpenPRTree.hpp](#)
- openpr/anticore/[OpenPRTree.cpp](#)

8.21 openpr::netlist::OutPin Class Reference

```
#include <OutPin.h>
```


Inherits [openpr::netlist::Pin](#).

Collaboration diagram for openpr::netlist::OutPin:



Public Member Functions

- [OutPin](#) ()
- [OutPin](#) (std::string [outpin](#), std::string [location](#))
- virtual [~OutPin](#) ()
- void [printData](#) (std::ofstream *outStream)
- void [printData](#) (std::ostream *outStream)
- bool [operator!=](#) (Pin &other)
- bool [operator!=](#) (Pin *other)
- bool [operator==](#) (const Pin &other) const
- bool [operator==](#) (const Pin *other) const
- bool [operator<](#) (Pin &other)
- bool [operator<](#) (Pin *other)
- bool [operator>](#) (Pin &other)
- bool [operator>](#) (Pin *other)
- Pin & [operator=](#) (Pin &other)
- Pin * [operator=](#) (Pin *other)
- size_t [operator\(\)](#) (const Pin &x) const
- size_t [operator\(\)](#) (const Pin *x) const

Private Attributes

- std::string [outpin](#)
- std::string [location](#)

8.21.1 Detailed Description

Definition at line 23 of file OutPin.h.

8.21.2 Constructor & Destructor Documentation

8.21.2.1 `openpr::netlist::OutPin::OutPin ()`

8.21.2.2 `openpr::netlist::OutPin::OutPin (std::string outpin, std::string location)`

Definition at line 23 of file OutPin.cpp.

8.21.2.3 `openpr::netlist::OutPin::~~OutPin () [virtual]`

Definition at line 29 of file OutPin.cpp.

8.21.3 Member Function Documentation

8.21.3.1 `bool openpr::netlist::OutPin::operator!= (Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 39 of file OutPin.cpp.

8.21.3.2 `bool openpr::netlist::OutPin::operator!= (Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 124 of file OutPin.cpp.

8.21.3.3 `std::size_t openpr::netlist::OutPin::operator() (const Pin & x) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 111 of file OutPin.cpp.

8.21.3.4 `std::size_t openpr::netlist::OutPin::operator() (const Pin * x) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 199 of file OutPin.cpp.

8.21.3.5 `bool openpr::netlist::OutPin::operator< (Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 139 of file OutPin.cpp.

8.21.3.6 `bool openpr::netlist::OutPin::operator< (Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 54 of file OutPin.cpp.

8.21.3.7 Pin & openpr::netlist::OutPin::operator= (Pin & *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 71 of file OutPin.cpp.

8.21.3.8 Pin * openpr::netlist::OutPin::operator= (Pin * *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 156 of file OutPin.cpp.

8.21.3.9 bool openpr::netlist::OutPin::operator== (const Pin * *other*) const [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 168 of file OutPin.cpp.

8.21.3.10 bool openpr::netlist::OutPin::operator== (const Pin & *other*) const [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 83 of file OutPin.cpp.

8.21.3.11 bool openpr::netlist::OutPin::operator> (Pin & *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 95 of file OutPin.cpp.

8.21.3.12 bool openpr::netlist::OutPin::operator> (Pin * *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 183 of file OutPin.cpp.

8.21.3.13 void openpr::netlist::OutPin::printData (std::ofstream * *outStream*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 33 of file OutPin.cpp.

8.21.3.14 void openpr::netlist::OutPin::printData (std::ostream * *outStream*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 36 of file OutPin.cpp.

8.21.4 Member Data Documentation

8.21.4.1 `std::string openpr::netlist::OutPin::location` [private]

Definition at line 44 of file OutPin.h.

8.21.4.2 `std::string openpr::netlist::OutPin::outpin` [private]

Definition at line 43 of file OutPin.h.

The documentation for this class was generated from the following files:

- [openpr/netlist/OutPin.h](#)
- [openpr/netlist/OutPin.cpp](#)

8.22 `openpr::netlist::Pin` Class Reference

```
#include <Pin.h>
```

Inherited by [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

Public Member Functions

- [Pin](#) ()
- virtual [~Pin](#) ()
- virtual void [printData](#) (std::ofstream *outStream)=0
- virtual void [printData](#) (std::ostream *outStream)=0
- virtual bool [operator!=](#) ([Pin](#) &other)=0
- virtual bool [operator!=](#) ([Pin](#) *other)=0
- virtual bool [operator==](#) (const [Pin](#) &other) const =0
- virtual bool [operator==](#) (const [Pin](#) *other) const =0
- virtual bool [operator<](#) ([Pin](#) &other)=0
- virtual bool [operator<](#) ([Pin](#) *other)=0
- virtual bool [operator>](#) ([Pin](#) &other)=0
- virtual bool [operator>](#) ([Pin](#) *other)=0
- virtual [Pin](#) & [operator=](#) ([Pin](#) &other)=0
- virtual [Pin](#) * [operator=](#) ([Pin](#) *other)=0
- virtual size_t [operator\(\)](#) (const [Pin](#) &x) const =0
- virtual size_t [operator\(\)](#) (const [Pin](#) *x) const =0

Friends

- std::size_t [hash_value](#) ([Pin](#) *p)
- std::size_t [hash_value](#) ([Pin](#) &p)

8.22.1 Detailed Description

Definition at line 35 of file Pin.h.

8.22.2 Constructor & Destructor Documentation

8.22.2.1 openpr::netlist::Pin::Pin ()

Definition at line 23 of file Pin.cpp.

8.22.2.2 openpr::netlist::Pin::~~Pin () [virtual]

Definition at line 27 of file Pin.cpp.

8.22.3 Member Function Documentation

8.22.3.1 virtual bool openpr::netlist::Pin::operator!=(Pin & other) [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.2 virtual bool openpr::netlist::Pin::operator!=(Pin * other) [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.3 virtual size_t openpr::netlist::Pin::operator() (const Pin & x) const [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.4 virtual size_t openpr::netlist::Pin::operator() (const Pin * x) const [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.5 virtual bool openpr::netlist::Pin::operator< (Pin * other) [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.6 virtual bool openpr::netlist::Pin::operator< (Pin & other) [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.7 virtual Pin& openpr::netlist::Pin::operator= (Pin & other) [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.8 virtual Pin* openpr::netlist::Pin::operator= (Pin * other) [pure virtual]

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.9 `virtual bool openpr::netlist::Pin::operator==(const Pin * other) const [pure virtual]`

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.10 `virtual bool openpr::netlist::Pin::operator==(const Pin & other) const [pure virtual]`

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.11 `virtual bool openpr::netlist::Pin::operator> (Pin & other) [pure virtual]`

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.12 `virtual bool openpr::netlist::Pin::operator> (Pin * other) [pure virtual]`

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.13 `virtual void openpr::netlist::Pin::printData (std::ofstream * outStream) [pure virtual]`

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.3.14 `virtual void openpr::netlist::Pin::printData (std::ostream * outStream) [pure virtual]`

Implemented in [openpr::netlist::InPin](#), [openpr::netlist::OutPin](#), and [openpr::netlist::Pip](#).

8.22.4 Friends And Related Function Documentation

8.22.4.1 `std::size_t hash_value (Pin * p) [friend]`

Definition at line 34 of file [Pin.cpp](#).

8.22.4.2 `std::size_t hash_value (Pin & p) [friend]`

Definition at line 30 of file [Pin.cpp](#).

The documentation for this class was generated from the following files:

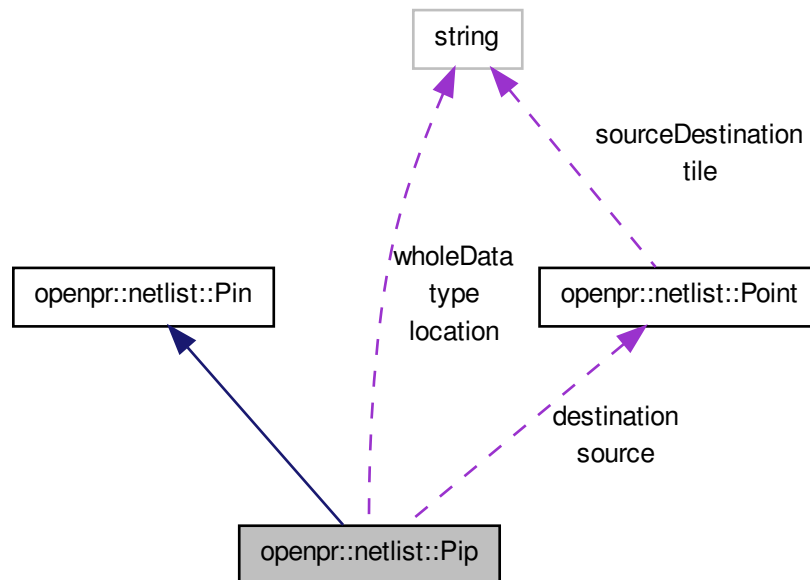
- [openpr/netlist/Pin.h](#)
- [openpr/netlist/Pin.cpp](#)

8.23 openpr::netlist::Pip Class Reference

```
#include <Pip.h>
```

Inherits [openpr::netlist::Pin](#).

Collaboration diagram for openpr::netlist::Pip:



Public Member Functions

- [Pip](#) ()
- [Pip](#) (std::string tok, torc::architecture::DDB *mdb)
- [Pip](#) (std::string [type](#), int x, int y, std::string [source](#), std::string [destination](#), torc::architecture::DDB *mdb)
- [Pip](#) (std::string [location](#), std::string [source](#), std::string [destination](#), torc::architecture::DDB *mdb)
- virtual [~Pip](#) ()
- void [printData](#) (std::ofstream *outStream)
- void [printData](#) (std::ostream *outStream)
- bool [operator!=](#) (Pin &other)
- bool [operator!=](#) (Pin *other)
- bool [operator==](#) (const Pin &other) const
- bool [operator==](#) (const Pin *other) const
- bool [operator<](#) (Pin &other)
- bool [operator<](#) (Pin *other)
- bool [operator>](#) (Pin &other)
- bool [operator>](#) (Pin *other)
- Pin & [operator=](#) (Pin &other)
- Pin * [operator=](#) (Pin *other)
- size_t [operator\(\)](#) (const Pin &x) const
- size_t [operator\(\)](#) (const Pin *x) const
- size_t [operator\(\)](#) (const Pip &x) const

- `size_t operator() (const Pip *x) const`
- `Point * getSource () const`
- `Point * getDestination () const`
- `std::string getTileStr ()`
- `std::string getSourceStr ()`
- `std::string getSinkStr ()`

Private Member Functions

- `void parseLocation (std::string location)`
- `void generateFullStream ()`
- `std::string generateLocation () const`

Private Attributes

- `std::string wholeData`
- `std::string location`
- `std::string type`
- `Point * source`
- `Point * destination`
- `int xLoc`
- `int yLoc`
- `torc::architecture::DDB * mDB`

8.23.1 Detailed Description

Definition at line 23 of file Pip.h.

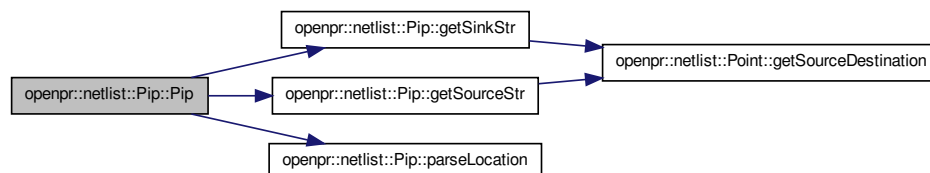
8.23.2 Constructor & Destructor Documentation

8.23.2.1 `openpr::netlist::Pip::Pip ()`

8.23.2.2 `openpr::netlist::Pip::Pip (std::string tok, torc::architecture::DDB * mdb)`

Definition at line 26 of file Pip.cpp.

Here is the call graph for this function:



8.23.2.3 openpr::netlist::Pip::Pip (std::string *type*, int *x*, int *y*, std::string *source*, std::string *destination*, torc::architecture::DDB * *mDB*)

This constructor takes in a parameterized list of inputs to construct at [Pip](#)

Parameters

- type*** This should be CLBLM, INT ...
- x*** This is the X tile location
- y*** This is the Y tile location
- source*** This is the source of pip
- destination*** This is the destination(sink) of the pip

Definition at line 62 of file Pip.cpp.

Here is the call graph for this function:



8.23.2.4 openpr::netlist::Pip::Pip (std::string *location*, std::string *source*, std::string *destination*, torc::architecture::DDB * *mDB*)

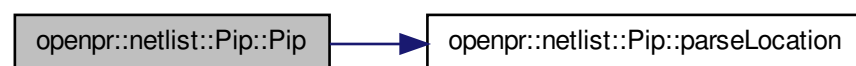
This constructor merges the two above constructors

Parameters

- location*** this should be INT_X27Y70
- source*** this is the source of the pip
- destination*** this is the destination(sink) of the pip

Definition at line 78 of file Pip.cpp.

Here is the call graph for this function:



8.23.2.5 openpr::netlist::Pip::~~Pip () [virtual]

Definition at line 102 of file Pip.cpp.

8.23.3 Member Function Documentation

8.23.3.1 void openpr::netlist::Pip::generateFullStream () [private]

8.23.3.2 std::string openpr::netlist::Pip::generateLocation () const [private]

Definition at line 409 of file Pip.cpp.

8.23.3.3 Point * openpr::netlist::Pip::getDestination () const

Definition at line 418 of file Pip.cpp.

8.23.3.4 std::string openpr::netlist::Pip::getSinkStr ()

Definition at line 430 of file Pip.cpp.

Here is the call graph for this function:



8.23.3.5 Point * openpr::netlist::Pip::getSource () const

Definition at line 421 of file Pip.cpp.

8.23.3.6 std::string openpr::netlist::Pip::getSourceStr ()

Definition at line 427 of file Pip.cpp.

Here is the call graph for this function:



8.23.3.7 `std::string openpr::netlist::Pip::getTileStr ()`

Definition at line 424 of file Pip.cpp.

8.23.3.8 `bool openpr::netlist::Pip::operator!=(Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 124 of file Pip.cpp.

8.23.3.9 `bool openpr::netlist::Pip::operator!=(Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 237 of file Pip.cpp.

8.23.3.10 `std::size_t openpr::netlist::Pip::operator() (const Pip * x) const`

Definition at line 400 of file Pip.cpp.

8.23.3.11 `std::size_t openpr::netlist::Pip::operator() (const Pip & x) const`

Definition at line 390 of file Pip.cpp.

8.23.3.12 `std::size_t openpr::netlist::Pip::operator() (const Pin & x) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 357 of file Pip.cpp.

8.23.3.13 `std::size_t openpr::netlist::Pip::operator() (const Pin * x) const [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 373 of file Pip.cpp.

8.23.3.14 `bool openpr::netlist::Pip::operator< (Pin * other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 277 of file Pip.cpp.

8.23.3.15 `bool openpr::netlist::Pip::operator< (Pin & other) [virtual]`

Implements [openpr::netlist::Pin](#).

Definition at line 160 of file Pip.cpp.

8.23.3.16 Pin & openpr::netlist::Pip::operator= (Pin & *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 220 of file Pip.cpp.

8.23.3.17 Pin * openpr::netlist::Pip::operator= (Pin * *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 339 of file Pip.cpp.

8.23.3.18 bool openpr::netlist::Pip::operator== (const Pin & *other*) const [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 142 of file Pip.cpp.

8.23.3.19 bool openpr::netlist::Pip::operator== (const Pin * *other*) const [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 255 of file Pip.cpp.

8.23.3.20 bool openpr::netlist::Pip::operator> (Pin & *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 190 of file Pip.cpp.

8.23.3.21 bool openpr::netlist::Pip::operator> (Pin * *other*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 307 of file Pip.cpp.

8.23.3.22 void openpr::netlist::Pip::parseLocation (std::string *location*) [private]

This parses the location into individual parts(Type, x, and y)

Parameters

location something like INT_X27Y70

Definition at line 90 of file Pip.cpp.

8.23.3.23 void openpr::netlist::Pip::printData (std::ostream * *outStream*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 115 of file Pip.cpp.

Here is the call graph for this function:



8.23.3.24 void openpr::netlist::Pip::printData (std::ofstream * *outStream*) [virtual]

Implements [openpr::netlist::Pin](#).

Definition at line 106 of file Pip.cpp.

Here is the call graph for this function:



8.23.4 Member Data Documentation

8.23.4.1 Point* openpr::netlist::Pip::destination [private]

Definition at line 59 of file Pip.h.

8.23.4.2 std::string openpr::netlist::Pip::location [private]

Definition at line 56 of file Pip.h.

8.23.4.3 torc::architecture::DDB* openpr::netlist::Pip::mDB [private]

Definition at line 65 of file Pip.h.

8.23.4.4 Point* openpr::netlist::Pip::source [private]

Definition at line 58 of file Pip.h.

8.23.4.5 std::string openpr::netlist::Pip::type [private]

Definition at line 57 of file Pip.h.

8.23.4.6 `std::string openpr::netlist::Pip::wholeData` `[private]`

Definition at line 55 of file Pip.h.

8.23.4.7 `int openpr::netlist::Pip::xLoc` `[private]`

Definition at line 60 of file Pip.h.

8.23.4.8 `int openpr::netlist::Pip::yLoc` `[private]`

Definition at line 61 of file Pip.h.

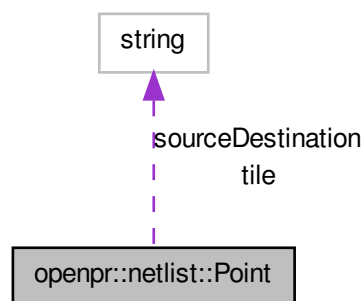
The documentation for this class was generated from the following files:

- [openpr/netlist/Pip.h](#)
- [openpr/netlist/Pip.cpp](#)

8.24 `openpr::netlist::Point` Class Reference

```
#include <Point.h>
```

Collaboration diagram for `openpr::netlist::Point`:



Public Member Functions

- [Point](#) ()
- [Point](#) (std::string [tile](#), std::string [sourceDestination](#), bool [source](#), torc::architecture::DDB *mDB)
- bool [operator>](#) (const [Point](#) &other) const
- bool [operator>](#) (const [Point](#) *other) const
- bool [operator<](#) (const [Point](#) &other) const
- bool [operator<](#) (const [Point](#) *other) const
- bool [operator!=](#) (const [Point](#) &other) const

- bool `operator!=` (const [Point](#) *other) const
- bool `operator==` (const [Point](#) &other) const
- bool `operator==` (const [Point](#) *other) const
- size_t `operator()` (const [Point](#) &x) const
- size_t `operator()` (const [Point](#) *x) const
- bool `getSource` () const
- std::string `getSourceDestination` () const
- virtual `~Point` ()
- torc::architecture::xilinx::CompactSegmentIndex `getSegmentIndex` () const

Private Member Functions

- void `setIndices` (torc::architecture::xilinx::TileIndex tileIndex, torc::architecture::xilinx::CompactSegmentIndex segmentIndex)

Private Attributes

- bool `source`
- std::string `tile`
- std::string `sourceDestination`
- torc::architecture::xilinx::CompactSegmentIndex `segmentIndex`

8.24.1 Detailed Description

Definition at line 23 of file Point.h.

8.24.2 Constructor & Destructor Documentation

8.24.2.1 openpr::netlist::Point::Point ()

8.24.2.2 openpr::netlist::Point::Point (std::string *tile*, std::string *sourceDestination*, bool *source*, torc::architecture::DDB * *mDB*)

This is the normal constructor

Parameters

tile This is the tile name for a source destination pair.

sourceDestination This is either the source or destination location for a point

source if it is a source it is true if it is a destination it is false

Definition at line 32 of file Point.cpp.

8.24.2.3 openpr::netlist::Point::~~Point () [virtual]

Definition at line 23 of file Point.cpp.

8.24.3 Member Function Documentation

8.24.3.1 `torc::architecture::xilinx::CompactSegmentIndex openpr::netlist::Point::getSegmentIndex () const`

Definition at line 210 of file Point.cpp.

8.24.3.2 `bool openpr::netlist::Point::getSource () const`

Returns

returns a const of the bool source true=source false=destination

Definition at line 197 of file Point.cpp.

8.24.3.3 `std::string openpr::netlist::Point::getSourceDestination () const`

Returns

returns a const of the source destination field. This is the point within the tile

Definition at line 203 of file Point.cpp.

8.24.3.4 `bool openpr::netlist::Point::operator!= (const Point & other) const`

Parameters

other this is the other point to be compared against this Boolean operator overload Only compares tile and the sourceDestination. (bool)source is not evaluated

Definition at line 171 of file Point.cpp.

8.24.3.5 `bool openpr::netlist::Point::operator!= (const Point * other) const`

Parameters

other this is the other point to be compared against this Boolean operator overload Only compares tile and the sourceDestination. (bool)source is not evaluated

Definition at line 185 of file Point.cpp.

8.24.3.6 `size_t openpr::netlist::Point::operator() (const Point & x) const`

Parameters

x point to be investigated This function evaluates the hash of a point. If is effectively a static function

Definition at line 75 of file Point.cpp.

8.24.3.7 `size_t openpr::netlist::Point::operator() (const Point * x) const`**Parameters**

x point to be investigated This function evaluates the hash of a point. If is effectively a static function

Definition at line 87 of file Point.cpp.

8.24.3.8 `bool openpr::netlist::Point::operator< (const Point & other) const`**Parameters**

other this is the other point to be compared against this Boolean operator overload Only compares tile and the sourceDestination. (bool)source is not evaluated

Definition at line 135 of file Point.cpp.

8.24.3.9 `bool openpr::netlist::Point::operator< (const Point * other) const`**Parameters**

other this is the other point to be compared against this Boolean operator overload Only compares tile and the sourceDestination. (bool)source is not evaluated

Definition at line 153 of file Point.cpp.

8.24.3.10 `bool openpr::netlist::Point::operator== (const Point * other) const`**Parameters**

other other point to be compared to this This is the equality operator it compares all three paramaters of tile, sourcedestination and (bool)source

Definition at line 62 of file Point.cpp.

8.24.3.11 `bool openpr::netlist::Point::operator== (const Point & other) const`**Parameters**

other other point to be compared to this This is the equality operator it compares all three of tile, sourcedestination and (bool)source

Definition at line 49 of file Point.cpp.

8.24.3.12 `bool openpr::netlist::Point::operator> (const Point & other) const`**Parameters**

other this is the other point to be compared against this Boolean operator overload Only compares tile and the sourceDestination. (bool)source is not evaluated

Definition at line 100 of file Point.cpp.

8.24.3.13 `bool openpr::netlist::Point::operator> (const Point * other) const`

Parameters

other this is the other point to be compared against this Boolean operator overload Only compares tile and the sourceDestination. (bool)source is not evaluated

Definition at line 118 of file Point.cpp.

8.24.3.14 `void openpr::netlist::Point::setIndices (torc::architecture::xilinx::TileIndex tileIndex, torc::architecture::xilinx::CompactSegmentIndex segmentIndex) [private]`

Definition at line 206 of file Point.cpp.

8.24.4 Member Data Documentation

8.24.4.1 `torc::architecture::xilinx::CompactSegmentIndex openpr::netlist::Point::segmentIndex [private]`

Definition at line 48 of file Point.h.

8.24.4.2 `bool openpr::netlist::Point::source [private]`

Definition at line 45 of file Point.h.

8.24.4.3 `std::string openpr::netlist::Point::sourceDestination [private]`

Definition at line 47 of file Point.h.

8.24.4.4 `std::string openpr::netlist::Point::tile [private]`

Definition at line 46 of file Point.h.

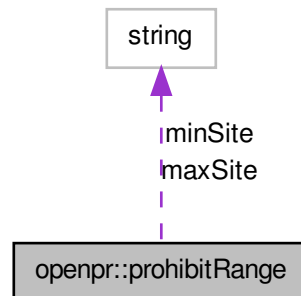
The documentation for this class was generated from the following files:

- [openpr/netlist/Point.h](#)
- [openpr/netlist/Point.cpp](#)

8.25 `openpr::prohibitRange` Struct Reference

```
#include <ProhibitRange.hpp>
```

Collaboration diagram for openpr::prohibitRange:



Public Member Functions

- [prohibitRange](#) ()
- [prohibitRange](#) (string minSite, string maxSite)
- [prohibitRange](#) (const [prohibitRange](#) &rhs)
- bool [operator<](#) (const [prohibitRange](#) &rhs)
- bool [operator>](#) (const [prohibitRange](#) &rhs)

Public Attributes

- [string minSite](#)
- [string maxSite](#)

8.25.1 Detailed Description

The [prohibitRange](#) struct represents a range of Sites that can be added to a PROHIBIT constraint by the tools.

Definition at line 23 of file ProhibitRange.hpp.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 openpr::prohibitRange::prohibitRange () [inline]

Definition at line 28 of file ProhibitRange.hpp.

8.25.2.2 openpr::prohibitRange::prohibitRange (string minSite, string maxSite) [inline]

Definition at line 31 of file ProhibitRange.hpp.

8.25.2.3 `openpr::prohibitRange::prohibitRange (const prohibitRange & rhs) [inline]`

Copy constructor.

Parameters

rhs object to copy from.

Definition at line 38 of file ProhibitRange.hpp.

8.25.3 Member Function Documentation

8.25.3.1 `bool openpr::prohibitRange::operator< (const prohibitRange & rhs) [inline]`

Compare two prohibitRanges and check whether the RHS or LHS value should represent the bottom corner of the range.

Parameters

rhs Right hand side value to be compared

Definition at line 47 of file ProhibitRange.hpp.

8.25.3.2 `bool openpr::prohibitRange::operator> (const prohibitRange & rhs) [inline]`

Compare two prohibitRanges and check whether the RHS or LHS value should represent the top corner of the range.

Parameters

rhs [prohibitRange](#) object to compare too

Definition at line 83 of file ProhibitRange.hpp.

8.25.4 Member Data Documentation

8.25.4.1 `string openpr::prohibitRange::maxSite`

Last site in range for this SITE type

Definition at line 27 of file ProhibitRange.hpp.

8.25.4.2 `string openpr::prohibitRange::minSite`

First site in range for this SITE type

Definition at line 25 of file ProhibitRange.hpp.

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

- [openpr/anticore/ProhibitRange.hpp](#)

8.26 openpr::bitstream::tile_coord Struct Reference

```
#include <tile.h>
```

Public Member Functions

- [tile_coord](#) ()
- [tile_coord](#) (int *x*, int *y*)
- void [set](#) (int *_x*, int *_y*)
- bool [operator==](#) ([tile_coord](#) const &*other*) const

Public Attributes

- int *x*
- int *y*

Friends

- std::size_t [hash_value](#) ([tile_coord](#) const &*to_hash*)

8.26.1 Detailed Description

Definition at line 16 of file `tile.h`.

8.26.2 Constructor & Destructor Documentation

8.26.2.1 openpr::bitstream::tile_coord::tile_coord () [inline]

Definition at line 20 of file `tile.h`.

8.26.2.2 openpr::bitstream::tile_coord::tile_coord (int *x*, int *y*) [inline]

Definition at line 25 of file `tile.h`.

8.26.3 Member Function Documentation

8.26.3.1 bool openpr::bitstream::tile_coord::operator== ([tile_coord](#) const & *other*) const [inline]

Definition at line 31 of file `tile.h`.

8.26.3.2 void openpr::bitstream::tile_coord::set (int *_x*, int *_y*) [inline]

Definition at line 26 of file `tile.h`.

8.26.4 Friends And Related Function Documentation

8.26.4.1 `std::size_t hash_value (tile_coord const & to_hash) [friend]`

Definition at line 36 of file tile.h.

8.26.5 Member Data Documentation

8.26.5.1 `int openpr::bitstream::tile_coord::x`

Definition at line 18 of file tile.h.

8.26.5.2 `int openpr::bitstream::tile_coord::y`

Definition at line 19 of file tile.h.

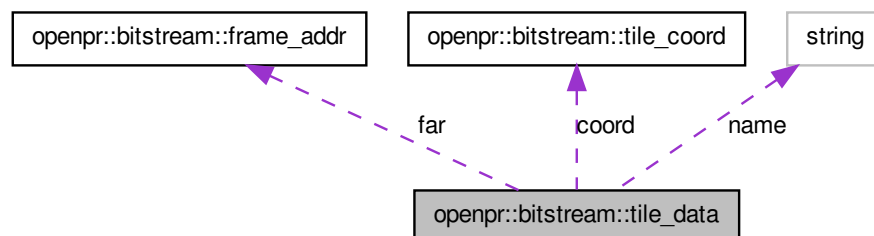
The documentation for this struct was generated from the following file:

- [openpr/bitstream/tile.h](#)

8.27 `openpr::bitstream::tile_data` Struct Reference

```
#include <tile.h>
```

Collaboration diagram for `openpr::bitstream::tile_data`:



Public Member Functions

- `tile_data` (const [string](#) _name, const int x, const int y, const [frame_addr](#) _far, const int _num_frames, const int _frame_num, const int _byte_off, char *const _first_frame)
- void [print](#) ()

Public Attributes

- [string](#) name

- [tile_coord](#) coord
- [frame_addr](#) far
- int [frame_num](#)
- int [byte_off](#)
- int [num_frames](#)
- char * [first_frame](#)

8.27.1 Detailed Description

Definition at line 45 of file tile.h.

8.27.2 Constructor & Destructor Documentation

8.27.2.1 `openpr::bitstream::tile_data::tile_data (const string _name, const int x, const int y, const frame_addr _far, const int _num_frames, const int _frame_num, const int _byte_off, char *const _first_frame) [inline]`

Definition at line 54 of file tile.h.

Here is the call graph for this function:



8.27.3 Member Function Documentation

8.27.3.1 `void openpr::bitstream::tile_data::print () [inline]`

Definition at line 66 of file tile.h.

Here is the call graph for this function:



8.27.4 Member Data Documentation

8.27.4.1 `int openpr::bitstream::tile_data::byte_off`

Definition at line 51 of file tile.h.

8.27.4.2 `tile_coord openpr::bitstream::tile_data::coord`

Definition at line 48 of file tile.h.

8.27.4.3 `frame_addr openpr::bitstream::tile_data::far`

Definition at line 49 of file tile.h.

8.27.4.4 `char* openpr::bitstream::tile_data::first_frame`

Definition at line 53 of file tile.h.

8.27.4.5 `int openpr::bitstream::tile_data::frame_num`

Definition at line 50 of file tile.h.

8.27.4.6 `string openpr::bitstream::tile_data::name`

Definition at line 47 of file tile.h.

8.27.4.7 `int openpr::bitstream::tile_data::num_frames`

Definition at line 52 of file tile.h.

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

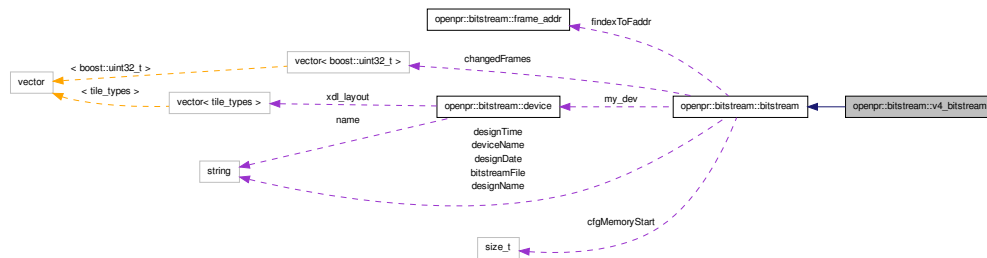
- `openpr/bitstream/tile.h`

8.28 `openpr::bitstream::v4_bitstream` Class Reference

```
#include <v4_bitstream.h>
```

Inherits `openpr::bitstream::bitstream`.

Collaboration diagram for openpr::bitstream::v4_bitstream:



Public Member Functions

- [v4_bitstream](#) (std::string device_name, bool frame_ecc=false)
- [v4_bitstream](#) (void)
- virtual bool [readPackets](#) (std::fstream &inStream)
- virtual bool [writePacketHeader](#) (std::fstream &outStream, boost::uint32_t packetType, boost::uint32_t opcode, boost::uint32_t address, boost::uint32_t reserved, boost::uint32_t count)
- virtual bool [writePacketsPartial](#) (std::fstream &outStream)
- virtual bool [writePackets](#) (std::fstream &outStream)
- virtual bool [writePartialFrames](#) (std::fstream &outStream)

Protected Types

- enum [EPacketType](#) { [eType1](#) = 1, [eType2](#), [eTypeCount](#) = 8 }
- enum [EOpcode](#) {
[eOpNOP](#) = 0, [eOpRead](#), [eOpWrite](#), [eOpReserved](#),
[eOpcodeCount](#) }
- enum [ERegister](#) {
[eRegCRC](#) = 0, [eRegFAR](#), [eRegFDRI](#), [eRegFDRO](#),
[eRegCMD](#), [eRegCTL](#), [eRegMASK](#), [eRegSTAT](#),
[eRegLOUT](#), [eRegCOR](#), [eRegMFWR](#), [eRegCBC](#),
[eRegIDCODE](#), [eRegAXSS](#), [eRegCount](#) }
- enum [ECommand](#) {
[eCmdNULL](#) = 0, [eCmdWCFG](#), [eCmdMFWR](#), [eCmdLFRM](#),
[eCmdRCFG](#), [eCmdSTART](#), [eCmdRCAP](#), [eCmdRCRC](#),
[eCmdAGHIGH](#), [eCmdSWITCH](#), [eCmdGRESTORE](#), [eCmdSHUTDOWN](#),
[eCmdGCAPTURE](#), [eCmdDESYNC](#), [eCmdCount](#) }
- enum [EShifts](#) {
[eShiftPacketType](#) = 29, [eShiftPacketOpcode](#) = 27, [eShiftType1Address](#) = 13, [eShiftType1Reserved](#) = 11,
[eShiftType1Count](#) = 0, [eShiftType2Count](#) = 0 }

- enum `EMasks` {
`eMaskPacketType` = 0x00000007, `eMaskPacketOpcode` = 0x00000003, `eMaskType1Address` = 0x00003fff, `eMaskType1Reserved` = 0x00000003,
`eMaskType1Count` = 0x000007ff, `eMaskType2Count` = 0x07ffffff }
- enum `EWords` { `eDummyWord` = 0xFFFFFFFF, `eSyncWord` = 0xAA995566 }
- enum `EShiftFAR` {
`eShiftBlockType` = 19, `eShiftTB` = 22, `eShiftRow` = 14, `eShiftMajor` = 6,
`eShiftMNA` = 0 }

Protected Member Functions

- virtual bool `writeFrameData` (std::fstream &inStream, int inCount)
- void `unmangleTilePair` (boost::uint16_t *ptr)
- void `reverseFrameBits` (boost::uint8_t *ptr)
- `openpr::bitstream::frame_addr` `farToStruct` (boost::uint32_t far)
- boost::uint32_t `structToFar` (`openpr::bitstream::frame_addr` far)

Protected Attributes

- boost::uint32_t `mRegister` [eRegCount]

Static Protected Attributes

- static const int `top` = 0
- static const char * `sTypeName` [eTypeCount]
- static const char * `sOpcodeName` [eOpcodeCount]
- static const char * `sRegisterName` [eRegCount]
- static const char * `sCommandName` [eCmdCount]

8.28.1 Detailed Description

Definition at line 21 of file `v4_bitstream.h`.

8.28.2 Member Enumeration Documentation

8.28.2.1 enum `openpr::bitstream::v4_bitstream::ECommand` [protected]

See also

CMD register commands: UG071, v1.10, April 8, 2008, Table 7-6

Enumerator:

eCmdNULL
eCmdWCFG
eCmdMFWR
eCmdLFRM

eCmdRCFG
eCmdSTART
eCmdRCAP
eCmdRCRC
eCmdAGHIGH
eCmdSWITCH
eCmdGRESTORE
eCmdSHUTDOWN
eCmdGCAPTURE
eCmdDESYNC
eCmdCount

Definition at line 36 of file v4_bitstream.h.

8.28.2.2 enum openpr::bitstream::v4_bitstream::EMasks [protected]

See also

type 1 packet format: UG071, v1.10, April 8, 2008, Table 7-2
type 2 packet format: UG071, v1.10, April 8, 2008, Table 7-4

Enumerator:

eMaskPacketType
eMaskPacketOpcode
eMaskType1Address
eMaskType1Reserved
eMaskType1Count
eMaskType2Count

Definition at line 55 of file v4_bitstream.h.

8.28.2.3 enum openpr::bitstream::v4_bitstream::EOpcode [protected]

See also

opcode format: UG071, v1.10, April 8, 2008, Table 7-3

Enumerator:

eOpNOP
eOpRead
eOpWrite
eOpReserved
eOpcodeCount

Definition at line 29 of file v4_bitstream.h.

8.28.2.4 enum openpr::bitstream::v4_bitstream::EPacketType [protected]

See also

packet type: UG071, v1.10, April 8, 2008, Tables 7-2 and 7-4

Enumerator:

eType1

eType2

eTypeCount

Definition at line 26 of file v4_bitstream.h.

8.28.2.5 enum openpr::bitstream::v4_bitstream::ERegister [protected]

See also

configuration registers: UG071, v1.10, April 8, 2008, Table 7-5

Enumerator:

eRegCRC

eRegFAR

eRegFDRI

eRegFDRO

eRegCMD

eRegCTL

eRegMASK

eRegSTAT

eRegLOUT

eRegCOR

eRegMFWR

eRegCBC

eRegIDCODE

eRegAXSS

eRegCount

Definition at line 32 of file v4_bitstream.h.

8.28.2.6 enum openpr::bitstream::v4_bitstream::EShiftFAR [protected]

See also

Frame Address Register Description: UG071, v1.10, April 8, 2008, Table 7-8

Enumerator:

eShiftBlockType

eShiftTB

eShiftRow
eShiftMajor
eShiftMNA

Definition at line 73 of file v4_bitstream.h.

8.28.2.7 enum openpr::bitstream::v4_bitstream::EShifts [protected]

See also

type 1 packet format: UG071, v1.10, April 8, 2008, Table 7-2
type 2 packet format: UG071, v1.10, April 8, 2008, Table 7-4

Enumerator:

eShiftPacketType
eShiftPacketOpcode
eShiftType1Address
eShiftType1Reserved
eShiftType1Count
eShiftType2Count

Definition at line 41 of file v4_bitstream.h.

8.28.2.8 enum openpr::bitstream::v4_bitstream::EWords [protected]

Enumerator:

eDummyWord
eSyncWord

Definition at line 67 of file v4_bitstream.h.

8.28.3 Constructor & Destructor Documentation

8.28.3.1 openpr::bitstream::v4_bitstream::v4_bitstream (std::string *device_name*, bool *frame_ecc* = *false*)

8.28.3.2 openpr::bitstream::v4_bitstream::v4_bitstream (void)

Definition at line 37 of file v4_bitstream.cpp.

8.28.4 Member Function Documentation

8.28.4.1 frame_addr openpr::bitstream::v4_bitstream::farToStruct (boost::uint32_t *far*)
[protected, virtual]

Prepare frame for writeback.

Parameters

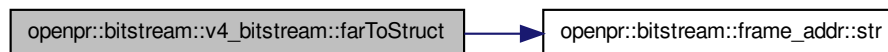
fIndex index of frame to prepare Convert the hex Frame Address Register to a [frame_addr](#) struct.

far Frame Address Register value to be converted.

Implements [openpr::bitstream::bitstream](#).

Definition at line 290 of file v4_bitstream.cpp.

Here is the call graph for this function:



8.28.4.2 `virtual bool openpr::bitstream::v4_bitstream::readPackets (std::fstream & inStream)`
[**virtual**]

8.28.4.3 `void openpr::bitstream::v4_bitstream::reverseFrameBits (boost::uint8_t * ptr)`
[**protected**]

remangle the bytes for a tile pair

Parameters

ptr A pointer to the frame reverse the order of the bits in a frame

ptr A pointer to the frame

Definition at line 386 of file v4_bitstream.cpp.

8.28.4.4 `boost::uint32_t openpr::bitstream::v4_bitstream::structToFar (`
`openpr::bitstream::frame_addr far)` [**protected**, **virtual**]

Convert the [frame_addr](#) struct to the 32bit hex frame address register format.

Parameters

far [frame_addr](#) struct to be converted.

Implements [openpr::bitstream::bitstream](#).

Definition at line 304 of file v4_bitstream.cpp.

8.28.4.5 `void openpr::bitstream::v4_bitstream::unmangleTilePair (boost::uint16_t * ptr)`
[**protected**]

unmangle the bytes for a tile pair

Parameters

ptr A pointer to the frame

Definition at line 367 of file v4_bitstream.cpp.

8.28.4.6 `virtual bool openpr::bitstream::v4_bitstream::writeFrameData (std::fstream & inStream, int inCount) [protected, virtual]`

8.28.4.7 `virtual bool openpr::bitstream::v4_bitstream::writePacketHeader (std::fstream & outStream, boost::uint32_t packetType, boost::uint32_t opcode, boost::uint32_t address, boost::uint32_t reserved, boost::uint32_t count) [virtual]`

Write a packet header to the bitstream.

Parameters

outStream Bitstream to be written.

packetType uint indicating type 1 or type 2 packet.

opcode Operation to be performed on register.

address Register address to be written.

reserved Param for reserved opcodes.

count Number of words to be written in packet.

8.28.4.8 `virtual bool openpr::bitstream::v4_bitstream::writePackets (std::fstream & outStream) [virtual]`

Write packets in proper sequence for a full bitstream.

Parameters

outStream Bitstream file to be written

Todo

test whether this actually works

8.28.4.9 `virtual bool openpr::bitstream::v4_bitstream::writePacketsPartial (std::fstream & outStream) [virtual]`

Write packets in proper sequence for a partial bitstream.

Parameters

outStream Bitstream file to be written

Todo

get actual partial bitstream sequence and write this function

8.28.4.10 `virtual bool openpr::bitstream::v4_bitstream::writePartialFrames (std::fstream & outStream) [virtual]`

Write out partial bitstream.

Parameters

outStream bitstream file to be written

8.28.5 Member Data Documentation

8.28.5.1 `boost::uint32_t openpr::bitstream::v4_bitstream::mRegister[eRegCount] [protected]`

Definition at line 81 of file v4_bitstream.h.

8.28.5.2 `const char * openpr::bitstream::v4_bitstream::sCommandName [static, protected]`

Initial value:

```
{
    "NULL", "WCFG", "MFWR", "LFRM", "RCFG", "START", "RCAP", "RCRC", "AGHIGH"
, "SWITCH", "GRESTORE", "SHUTDOWN",
  "GCAPTURE", "DESYNC"
}
```

Definition at line 85 of file v4_bitstream.h.

8.28.5.3 `const char * openpr::bitstream::v4_bitstream::sOpcodeName [static, protected]`

Initial value:

```
{
    "NOP", "READ", "WRITE", "RESERVED"
}
```

Definition at line 83 of file v4_bitstream.h.

8.28.5.4 `const char * openpr::bitstream::v4_bitstream::sRegisterName [static, protected]`

Initial value:

```
{
    "CRC", "FAR", "FDRI", "FDRO", "CMD", "CTL", "MASK", "STAT", "LOUT", "COR"
, "MFWR", "CBC", "IDCODE", "AXSS"
}
```

Definition at line 84 of file v4_bitstream.h.

8.28.5.5 const char * openpr::bitstream::v4_bitstream::sTypeName [static, protected]

Initial value:

```
{
    "[UNKNOWN TYPE 0]", "TYPE1", "TYPE2", "[UNKNOWN TYPE 3]", "[UNKNOWN TYPE
4]", "[UNKNOWN TYPE 5]",
    "[UNKNOWN TYPE 6]", "[UNKNOWN TYPE 7]"
}
```

Definition at line 82 of file v4_bitstream.h.

8.28.5.6 const int openpr::bitstream::v4_bitstream::top = 0 [static, protected]

Definition at line 80 of file v4_bitstream.h.

The documentation for this class was generated from the following files:

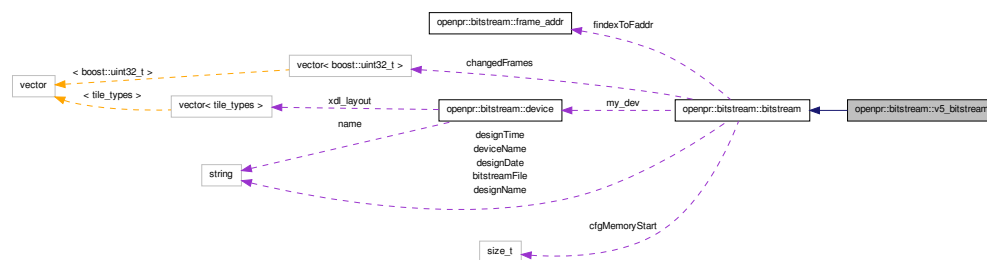
- [openpr/bitstream/v4_bitstream.h](#)
- [openpr/bitstream/v4_bitstream.cpp](#)

8.29 openpr::bitstream::v5_bitstream Class Reference

#include <v5_bitstream.h>

Inherits [openpr::bitstream::bitstream](#).

Collaboration diagram for openpr::bitstream::v5_bitstream:



Public Member Functions

- [v5_bitstream](#) ([string](#) device_name)
- [v5_bitstream](#) (void)
- virtual bool [readPackets](#) (fstream &inStream)
- virtual bool [writePacketHeader](#) (fstream &outStream, boost::uint32_t packetType, boost::uint32_t opcode, boost::uint32_t address, boost::uint32_t reserved, boost::uint32_t count)
- virtual bool [writePackets](#) (fstream &outStream)
- virtual bool [writePacketsPartial](#) (fstream &outStream)
- virtual bool [writePartialFrames](#) (fstream &outStream)

Protected Types

- enum [EPacketType](#) { [eType1](#) = 1, [eType2](#), [eTypeCount](#) = 8 }
- enum [EOpcode](#) {
[eOpNOP](#) = 0, [eOpRead](#), [eOpWrite](#), [eOpReserved](#),
[eOpcodeCount](#) }
- enum [ERegister](#) {
[eRegCRC](#) = 0, [eRegFAR](#), [eRegFDRI](#), [eRegFDRO](#),
[eRegCMD](#), [eRegCTL0](#), [eRegMASK](#), [eRegSTAT](#),
[eRegLOUT](#), [eRegCOR0](#), [eRegMFWR](#), [eRegCBC](#),
[eRegIDCODE](#), [eRegAXSS](#), [eRegCOR1](#), [eRegCSOB](#),
[eRegWBSTAR](#), [eRegTIMER](#), [eRegBOOTSTS](#) = 22, [eRegCTL1](#) = 24,
[eRegCount](#) }
- enum [ECommand](#) {
[eCmdNULL](#) = 0, [eCmdWCFG](#), [eCmdMFW](#), [eCmdDGHIGH](#),
[eCmdLFRM](#) = [eCmdDGHIGH](#), [eCmdRCFG](#), [eCmdSTART](#), [eCmdRCAP](#),
[eCmdRCRC](#), [eCmdAGHIGH](#), [eCmdSWITCH](#), [eCmdGRESTORE](#),
[eCmdSHUTDOWN](#), [eCmdGCAPTURE](#), [eCmdDESYNCH](#), [eCmdIPROG](#) = 15,
[eCmdLTIMER](#) = 17, [eCmdCount](#) }
- enum [EShifts](#) {
[eShiftPacketType](#) = 29, [eShiftPacketOpcode](#) = 27, [eShiftType1Address](#) = 13, [eShiftType1Reserved](#)
= 11,
[eShiftType1Count](#) = 0, [eShiftType2Count](#) = 0 }
- enum [EMasks](#) {
[eMaskPacketType](#) = 0x00000007, [eMaskPacketOpcode](#) = 0x00000003, [eMaskType1Address](#) =
0x000003fff, [eMaskType1Reserved](#) = 0x00000003,
[eMaskType1Count](#) = 0x000007ff, [eMaskType2Count](#) = 0x07ffffff }
- enum [EWords](#) { [eDummyWord](#) = 0xFFFFFFFF, [eBusWidthWord](#) = 0x000000BB, [eBusWidth](#) =
0x11220044, [eSyncWord](#) = 0xAA995566 }
- enum [EShiftFAR](#) {
[eShiftBlockType](#) = 21, [eShiftTB](#) = 20, [eShiftRow](#) = 15, [eShiftMajor](#) = 7,
[eShiftMNA](#) = 0 }

Protected Member Functions

- virtual bool [writeFrameData](#) (fstream &inStream, int inCount)
- [openpr::bitstream::frame_addr](#) [farToStruct](#) (boost::uint32_t far)
- boost::uint32_t [structToFar](#) ([openpr::bitstream::frame_addr](#) far)

Protected Attributes

- boost::uint32_t [mRegister](#) [[eRegCount](#)]

Static Protected Attributes

- static const char * [sTypeName](#) [eTypeCount]
- static const char * [sOpcodeName](#) [eOpcodeCount]
- static const char * [sRegisterName](#) [eRegCount]
- static const char * [sCommandName](#) [eCmdCount]

8.29.1 Detailed Description

Definition at line 21 of file v5_bitstream.h.

8.29.2 Member Enumeration Documentation

8.29.2.1 enum openpr::bitstream::v5_bitstream::ECommand [protected]

Enumerator:

eCmdNULL
eCmdWCFG
eCmdMFW
eCmdDGHIGH
eCmdLFRM
eCmdRCFG
eCmdSTART
eCmdRCAP
eCmdRCRC
eCmdAGHIGH
eCmdSWITCH
eCmdGRESTORE
eCmdSHUTDOWN
eCmdGCAPTURE
eCmdDESYNCH
eCmdIPROG
eCmdLTIMER
eCmdCount

Definition at line 37 of file v5_bitstream.h.

8.29.2.2 enum openpr::bitstream::v5_bitstream::EMasks [protected]

Enumerator:

eMaskPacketType
eMaskPacketOpcode
eMaskType1Address
eMaskType1Reserved

eMaskType1Count

eMaskType2Count

Definition at line 55 of file v5_bitstream.h.

8.29.2.3 enum openpr::bitstream::v5_bitstream::EOpcode [protected]

Enumerator:

eOpNOP

eOpRead

eOpWrite

eOpReserved

eOpcodeCount

Definition at line 29 of file v5_bitstream.h.

8.29.2.4 enum openpr::bitstream::v5_bitstream::EPacketType [protected]

Enumerator:

eType1

eType2

eTypeCount

Definition at line 26 of file v5_bitstream.h.

8.29.2.5 enum openpr::bitstream::v5_bitstream::ERegister [protected]

Enumerator:

eRegCRC

eRegFAR

eRegFDRI

eRegFDRO

eRegCMD

eRegCTL0

eRegMASK

eRegSTAT

eRegLOUT

eRegCOR0

eRegMFWR

eRegCBC

eRegIDCODE

eRegAXSS

eRegCOR1

eRegCSOB
eRegWBSTAR
eRegTIMER
eRegBOOTSTS
eRegCTL1
eRegCount

Definition at line 32 of file v5_bitstream.h.

8.29.2.6 enum openpr::bitstream::v5_bitstream::EShiftFAR [protected]

Enumerator: See also

eShiftBlockType Frame Address Register bitfields obtained from UG191 v3.6 Figure 6-10.
eShiftTB
eShiftRow
eShiftMajor
eShiftMNA

Definition at line 76 of file v5_bitstream.h.

8.29.2.7 enum openpr::bitstream::v5_bitstream::EShifts [protected]

Enumerator:

eShiftPacketType
eShiftPacketOpcode
eShiftType1Address
eShiftType1Reserved
eShiftType1Count
eShiftType2Count

Definition at line 41 of file v5_bitstream.h.

8.29.2.8 enum openpr::bitstream::v5_bitstream::EWords [protected]

Enumerator:

eDummyWord
eBusWidthWord
eBusWidth
eSyncWord

Definition at line 69 of file v5_bitstream.h.

8.29.3 Constructor & Destructor Documentation

8.29.3.1 `openpr::bitstream::v5_bitstream::v5_bitstream (string device_name)`

Definition at line 44 of file `v5_bitstream.cpp`.

8.29.3.2 `openpr::bitstream::v5_bitstream::v5_bitstream (void)`

Definition at line 39 of file `v5_bitstream.cpp`.

8.29.4 Member Function Documentation

8.29.4.1 `frame_addr openpr::bitstream::v5_bitstream::farToStruct (boost::uint32_t far)` [protected, virtual]

Convert the hex Frame Address Register to a `frame_addr` struct.

Parameters

far Frame Address Register value to be converted.

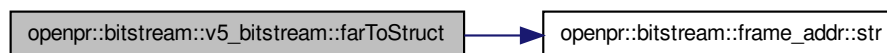
See also

Frame Address Register bitfields obtained from UG191 v3.6 Figure 6-10.

Implements `openpr::bitstream::bitstream`.

Definition at line 486 of file `v5_bitstream.cpp`.

Here is the call graph for this function:

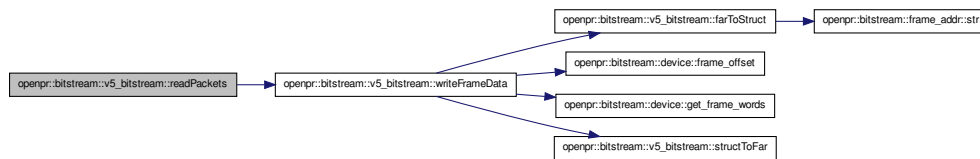


8.29.4.2 `bool openpr::bitstream::v5_bitstream::readPackets (fstream & inStream)` [virtual]

Implements `openpr::bitstream::bitstream`.

Definition at line 49 of file `v5_bitstream.cpp`.

Here is the call graph for this function:



8.29.4.3 boost::uint32_t openpr::bitstream::v5_bitstream::structToFar (openpr::bitstream::frame_addr far) [protected, virtual]

Convert the [frame_addr](#) struct to the 32bit hex frame address register format.

Parameters

far [frame_addr](#) struct to be converted.

See also

Frame Address Register bitfields obtained from UG191 v3.6 Figure 6-10.

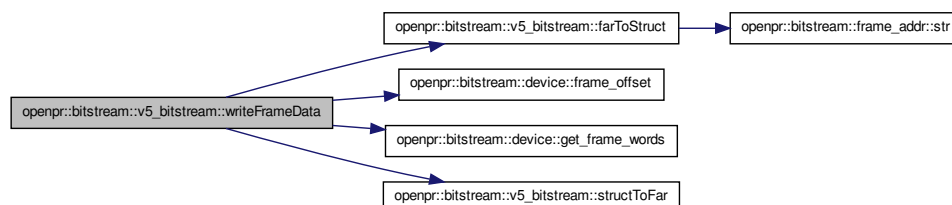
Implements [openpr::bitstream::bitstream](#).

Definition at line 501 of file v5_bitstream.cpp.

8.29.4.4 bool openpr::bitstream::v5_bitstream::writeFrameData (fstream & inStream, int inCount) [protected, virtual]

Definition at line 515 of file v5_bitstream.cpp.

Here is the call graph for this function:



8.29.4.5 bool openpr::bitstream::v5_bitstream::writePacketHeader (fstream & outStream, boost::uint32_t packetType, boost::uint32_t opcode, boost::uint32_t address, boost::uint32_t reserved, boost::uint32_t count) [virtual]

Write a packet header to the bitstream.

Parameters

outStream Bitstream to be written.

packetType uint indicating type 1 or type 2 packet.

opcode Operation to be performed on register.

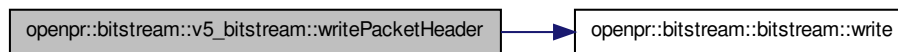
address Register address to be written.

reserved Param for reserved opcodes.

count Number of words to be written in packet.

Definition at line 188 of file v5_bitstream.cpp.

Here is the call graph for this function:



8.29.4.6 `bool openpr::bitstream::v5_bitstream::writePackets (fstream & outStream)` [virtual]

Write packets in proper sequence for a full bitstream.

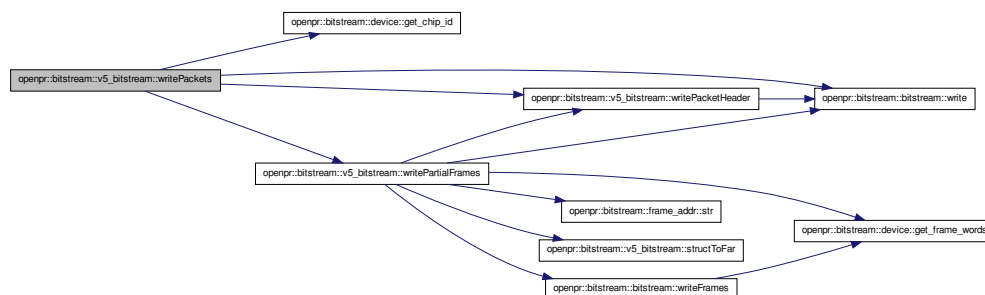
Parameters

outStream Bitstream file to be written

Implements [openpr::bitstream::bitstream](#).

Definition at line 216 of file v5_bitstream.cpp.

Here is the call graph for this function:



8.29.4.7 bool openpr::bitstream::v5_bitstream::writePacketsPartial (fstream & outStream) [virtual]

Write packets in proper sequence for a partial bitstream.

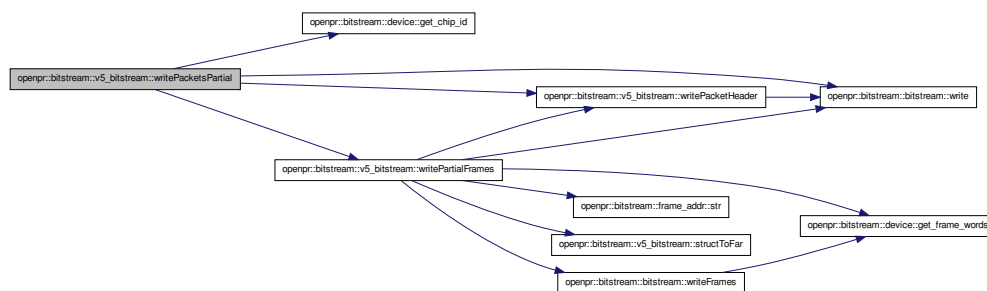
Parameters

outStream Bitstream file to be written

Implements [openpr::bitstream::bitstream](#).

Definition at line 334 of file v5_bitstream.cpp.

Here is the call graph for this function:



8.29.4.8 bool openpr::bitstream::v5_bitstream::writePartialFrames (fstream & outStream) [virtual]

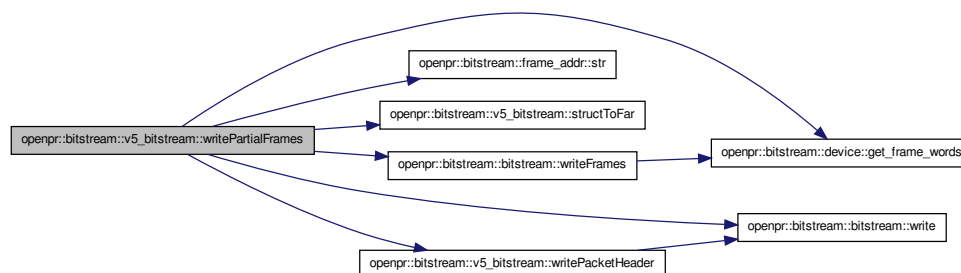
Write out partial bitstream.

Parameters

outStream bitstream file to be written

Definition at line 435 of file v5_bitstream.cpp.

Here is the call graph for this function:



8.29.5 Member Data Documentation

8.29.5.1 `boost::uint32_t openpr::bitstream::v5_bitstream::mRegister[eRegCount]` [protected]

Definition at line 85 of file v5_bitstream.h.

8.29.5.2 `const char * openpr::bitstream::v5_bitstream::sCommandName` [static, protected]

Initial value:

```
{
    "NULL", "WCFG", "MFW", "DGHIGH/LFRM", "RCFG", "START", "RCAP", "RCRC", "A
    GHIGH", "SWITCH", "GRESTORE", "SHUTDOWN",
    "GCAPTURE", "DESYNCH", "[UNKNOWN CMD 14]", "IPROG", "[UNKNOWN CMD 16]", "
    LTIMER"
}
```

Definition at line 89 of file v5_bitstream.h.

8.29.5.3 `const char * openpr::bitstream::v5_bitstream::sOpcodeName` [static, protected]

Initial value:

```
{
    "NOP", "READ", "WRITE", "RESERVED"
}
```

Definition at line 87 of file v5_bitstream.h.

8.29.5.4 `const char * openpr::bitstream::v5_bitstream::sRegisterName` [static, protected]

Initial value:

```
{
    "CRC", "FAR", "FDRI", "FDRO", "CMD", "CTL0", "MASK", "STAT", "LOUT", "COR
    0", "MFWR", "CBC", "IDCODE", "AXSS",
    "COR1", "CSOB", "WBSTAR", "TIMER", "[UNKNOWN REG 18]", "[UNKNOWN REG 19]"
    , "[UNKNOWN REG 20]", "[UNKNOWN REG 21]",
    "BOOTSTS", "[UNKNOWN REG 23]", "CTL1"
}
```

Definition at line 88 of file v5_bitstream.h.

8.29.5.5 `const char * openpr::bitstream::v5_bitstream::sTypeName` [static, protected]

Initial value:

```
{
    "[UNKNOWN TYPE 0]", "TYPE1", "TYPE2", "[UNKNOWN TYPE 3]", "[UNKNOWN TYPE
    4]", "[UNKNOWN TYPE 5]",
    "[UNKNOWN TYPE 6]", "[UNKNOWN TYPE 7]"
}
```

Definition at line 86 of file v5_bitstream.h.

The documentation for this class was generated from the following files:

- [openpr/bitstream/v5_bitstream.h](#)
- [openpr/bitstream/v5_bitstream.cpp](#)

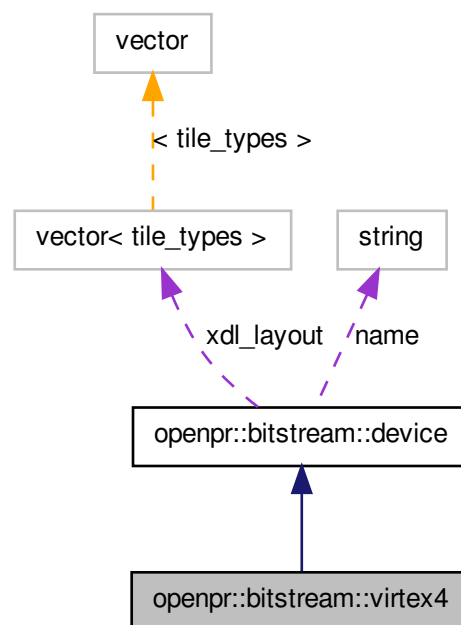
8.30 openpr::bitstream::virtex4 Class Reference

```
#include <virtex4.h>
```

Inherits [openpr::bitstream::device](#).

Inherited by [openpr::bitstream::xc4vfx60](#), [openpr::bitstream::xc4vlx15](#), and [openpr::bitstream::xc4vlx60](#).

Collaboration diagram for openpr::bitstream::virtex4:



Protected Types

- enum { [virtex4_frame_words](#) = 41, [virtex4_frame_height](#), [virtex4_clb_slices](#) = 4, [num_blk_types](#) = 3 }

Protected Member Functions

- `virtex4` (const int `num_rows`, const int `num_cols`, const `tile_types row_layout`[], const `string name`, const int `id`)
- int `get_addressable_blk_types` ()
- int `tile_offset` (int x, int y)
- `~virtex4` ()

Static Protected Attributes

- static const int `virtex4_tile_frames` []
- static const int `virtex4_block_type` []
- static const char * `virtex4_routing_table` = "Virtex4Bits"
- static const char * `virtex4_logic_table` = "Virtex4LogicBits"

8.30.1 Detailed Description

Definition at line 20 of file `virtex4.h`.

8.30.2 Member Enumeration Documentation

8.30.2.1 anonymous enum [protected]

Enumerator:

virtex4_frame_words
virtex4_frame_height
virtex4_clb_slices
num_blk_types

Definition at line 23 of file `virtex4.h`.

8.30.3 Constructor & Destructor Documentation

8.30.3.1 `openpr::bitstream::virtex4::virtex4` (const int *num_rows*, const int *num_cols*, const *tile_types row_layout*[], const string *name*, const int *id*) [inline, protected]

Virtex4 Class Constructor. Send V4 specific parameters to device class to be initialized

Definition at line 38 of file `virtex4.h`.

8.30.3.2 `openpr::bitstream::virtex4::~~virtex4` () [protected]

Return tile number within column when given byte offset.

Parameters

byte_offset The byte offset within the frame.

Returns

Tile index within column (-1 if in ECC word). Return the type of interconnect tile. Virtex4 Class Destructor. Does nothing so far

Todo

Remove if not used

Definition at line 54 of file virtex4.cpp.

8.30.4 Member Function Documentation**8.30.4.1 int openpr::bitstream::virtex4::get_addressable_blk_types () [inline, protected, virtual]**

Return number of addressable block types for Virtex5.

Reimplemented from [openpr::bitstream::device](#).

Definition at line 53 of file virtex4.h.

8.30.4.2 int openpr::bitstream::virtex4::tile_offset (int x, int y) [protected, virtual]

Return tile_offset within a frame (in bytes).

Implements [openpr::bitstream::device](#).

Definition at line 46 of file virtex4.cpp.

8.30.5 Member Data Documentation**8.30.5.1 const int openpr::bitstream::virtex4::virtex4_block_type [static, protected]**

Initial value:

```
{
    0,
    0,
    0,
    0,
    2,
    1,
    0,
    0
}
```

Definition at line 29 of file virtex4.h.

8.30.5.2 const char * openpr::bitstream::virtex4::virtex4_logic_table = "Virtex4LogicBits" [static, protected]

Definition at line 33 of file virtex4.h.

8.30.5.3 `const char * openpr::bitstream::virtex4::virtex4_routing_table = "Virtex4Bits"`
`[static, protected]`

Definition at line 31 of file virtex4.h.

8.30.5.4 `const int openpr::bitstream::virtex4::virtex4_tile_frames` `[static, protected]`

Initial value:

```
{
    30,
    3,
    22,
    21,
    64,
    20,
    20,
    2
}
```

Definition at line 27 of file virtex4.h.

The documentation for this class was generated from the following files:

- [openpr/bitstream/virtex4.h](#)
- [openpr/bitstream/virtex4.cpp](#)

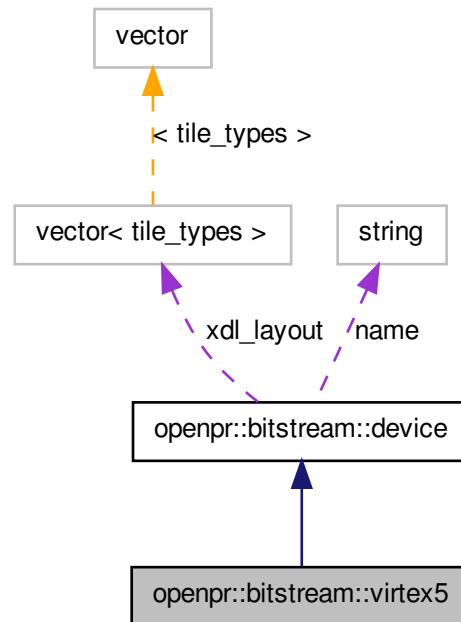
8.31 openpr::bitstream::virtex5 Class Reference

```
#include <virtex5.h>
```

Inherits [openpr::bitstream::device](#).

Inherited by [openpr::bitstream::xc5vlx110t](#), [openpr::bitstream::xc5vlx50](#), [openpr::bitstream::xc5vlx50t](#), and [openpr::bitstream::xc5vsx95t](#).

Collaboration diagram for openpr::bitstream::virtex5:



Protected Types

- enum { `virtex5_frame_words` = 41, `virtex5_frame_height`, `virtex5_clb_slices` = 2, `virtex5_num_blk_types` = 2 }

Protected Member Functions

- `virtex5` (const int `num_rows`, const int `num_cols`, const `tile_types` row_layout[], const string `name`, const int `id`)
- int `get_addressable_blk_types` ()
- virtual int `tile_offset` (int x, int y)
- `~virtex5` ()

Static Protected Attributes

- static const int `virtex5_tile_frames` []
- static const int `virtex5_block_type` []
- static const char * `virtex5_routing_table` = "Virtex5Bits"
- static const char * `virtex5_logic_table` = "Virtex5LogicBits"

8.31.1 Detailed Description

Definition at line 21 of file virtex5.h.

8.31.2 Member Enumeration Documentation

8.31.2.1 anonymous enum [protected]

Enumerator:

virtex5_frame_words
virtex5_frame_height
virtex5_clb_slices
virtex5_num_blk_types

Definition at line 24 of file virtex5.h.

8.31.3 Constructor & Destructor Documentation

8.31.3.1 openpr::bitstream::virtex5::virtex5 (const int num_rows, const int num_cols, const tile_types row_layout[], const string name, const int id) [inline, protected]

Virtex4 Class Constructor. Send V4 specific parameters to device class to be initialized

Definition at line 39 of file virtex5.h.

8.31.3.2 openpr::bitstream::virtex5::~~virtex5 () [protected]

Return tile number within column when given byte offset.

Parameters

byte_offset The byte offset within the frame.

Returns

Tile index within column (-1 if in ECC word). Return the type of interconnect tile. Virtex4 Class Destructor. Does nothing so far

Todo

Remove if not used

Definition at line 55 of file virtex5.cpp.

8.31.4 Member Function Documentation

8.31.4.1 int openpr::bitstream::virtex5::get_addressable_blk_types () [inline, protected, virtual]

Return number of addressable block types for Virtex5.

Reimplemented from [openpr::bitstream::device](#).

Definition at line 53 of file virtex5.h.

8.31.4.2 int openpr::bitstream::virtex5::tile_offset (int x, int y) [protected, virtual]

Return byte offset of tile within frames. Only works for CLB Block Types

Parameters

x X coordinate of tile
y Y coordinate of tile

Returns

byte offset of tile within a frame

Implements [openpr::bitstream::device](#).

Definition at line 46 of file virtex5.cpp.

8.31.5 Member Data Documentation

8.31.5.1 const int openpr::bitstream::virtex5::virtex5_block_type [static, protected]

Initial value:

```
{  
    0,  
    0,  
    0,  
    0,  
    1,  
    0,  
    0,  
    0  
}
```

Definition at line 30 of file virtex5.h.

8.31.5.2 const char * openpr::bitstream::virtex5::virtex5_logic_table = "Virtex5LogicBits" [static, protected]

Definition at line 34 of file virtex5.h.

8.31.5.3 const char * openpr::bitstream::virtex5::virtex5_routing_table = "Virtex5Bits" [static, protected]

Definition at line 32 of file virtex5.h.

8.31.5.4 const int openpr::bitstream::virtex5::virtex5_tile_frames [static, protected]

Initial value:

```
{  
    54,  
    4,  
    36,  
    28,  
}
```

```

    128,
    30,
    32,
    2
}

```

Definition at line 28 of file virtex5.h.

The documentation for this class was generated from the following files:

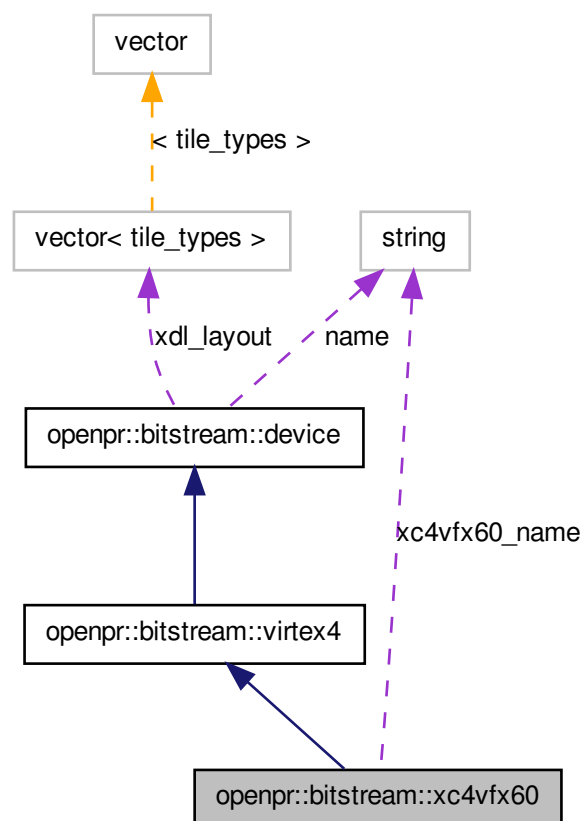
- [openpr/bitstream/virtex5.h](#)
- [openpr/bitstream/virtex5.cpp](#)

8.32 openpr::bitstream::xc4vfx60 Class Reference

```
#include <v4_devices.h>
```

Inherits [openpr::bitstream::virtex4](#).

Collaboration diagram for openpr::bitstream::xc4vfx60:



Public Member Functions

- [xc4vfx60\(\)](#)

Protected Types

- enum { [xc4vfx60_num_rows](#) = 8, [xc4vfx60_num_cols](#) = 76 }

Static Protected Attributes

- static const [tile_types xc4vfx60_row_layout](#) []
- static const [string xc4vfx60_name](#) = "XC4VFX60"
- static const int [xc4vfx60_id](#) = 0x01EB4093U

8.32.1 Detailed Description

Definition at line 46 of file v4_devices.h.

8.32.2 Member Enumeration Documentation

8.32.2.1 anonymous enum [protected]

Enumerator:

[xc4vfx60_num_rows](#)
[xc4vfx60_num_cols](#)

Definition at line 48 of file v4_devices.h.

8.32.3 Constructor & Destructor Documentation

8.32.3.1 openpr::bitstream::xc4vfx60::xc4vfx60() [inline]

Definition at line 53 of file v4_devices.h.

Here is the call graph for this function:



8.32.4 Member Data Documentation

8.32.4.1 `const int openpr::bitstream::xc4vfx60::xc4vfx60_id = 0x01EB4093U` `[static, protected]`

Definition at line 51 of file v4_devices.h.

8.32.4.2 `const string openpr::bitstream::xc4vfx60::xc4vfx60_name = "XC4VFX60"` `[static, protected]`

XC4VLX60 Part

Definition at line 50 of file v4_devices.h.

8.32.4.3 `const tile_types openpr::bitstream::xc4vfx60::xc4vfx60_row_layout` `[static, protected]`

Definition at line 49 of file v4_devices.h.

The documentation for this class was generated from the following files:

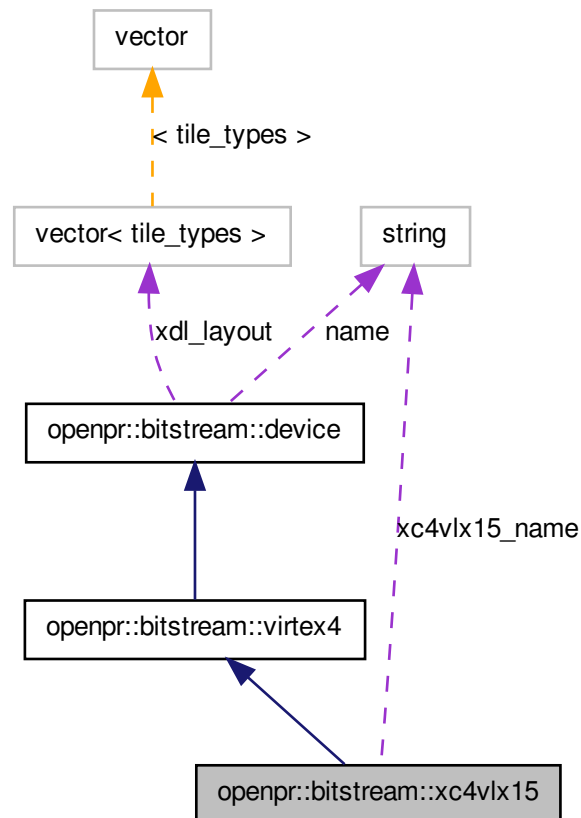
- [openpr/bitstream/v4_devices.h](#)
- [openpr/bitstream/v4_devices.cpp](#)

8.33 openpr::bitstream::xc4vlx15 Class Reference

```
#include <v4_devices.h>
```

Inherits [openpr::bitstream::virtex4](#).

Collaboration diagram for openpr::bitstream::xc4vlx15:



Public Member Functions

- [xc4vlx15\(\)](#)

Protected Types

- enum { [xc4vlx15_num_rows](#) = 4, [xc4vlx15_num_cols](#) = 35 }

Static Protected Attributes

- static const [tile_types](#) [xc4vlx15_row_layout](#) []
- static const [string](#) [xc4vlx15_name](#) = "XC4VLX15"
- static const int [xc4vlx15_id](#) = 0x01658093U

8.33.1 Detailed Description

Definition at line 20 of file v4_devices.h.

8.33.2 Member Enumeration Documentation

8.33.2.1 anonymous enum [protected]

Enumerator:

xc4vlx15_num_rows

xc4vlx15_num_cols

Definition at line 22 of file v4_devices.h.

8.33.3 Constructor & Destructor Documentation

8.33.3.1 openpr::bitstream::xc4vlx15::xc4vlx15 () [inline]

Definition at line 27 of file v4_devices.h.

Here is the call graph for this function:



8.33.4 Member Data Documentation

8.33.4.1 const int openpr::bitstream::xc4vlx15::xc4vlx15_id = 0x01658093U [static, protected]

Definition at line 25 of file v4_devices.h.

8.33.4.2 const string openpr::bitstream::xc4vlx15::xc4vlx15_name = "XC4VLX15" [static, protected]

Definition at line 24 of file v4_devices.h.

8.33.4.3 const file_types openpr::bitstream::xc4vlx15::xc4vlx15_row_layout [static, protected]

Definition at line 23 of file v4_devices.h.

The documentation for this class was generated from the following files:

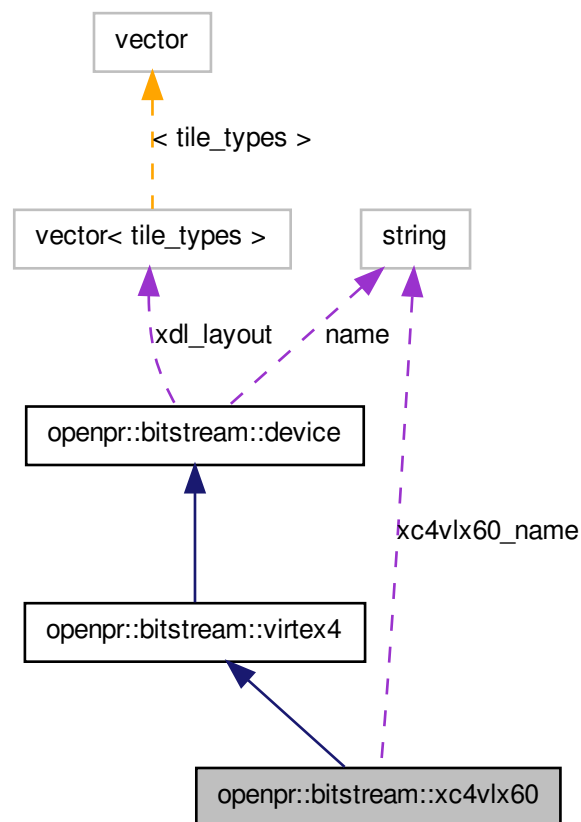
- [openpr/bitstream/v4_devices.h](#)
- [openpr/bitstream/v4_devices.cpp](#)

8.34 openpr::bitstream::xc4vlx60 Class Reference

#include <v4_devices.h>

Inherits [openpr::bitstream::virtex4](#).

Collaboration diagram for openpr::bitstream::xc4vlx60:



Public Member Functions

- [xc4vlx60\(\)](#)

Protected Types

- enum { [xc4vlx60_num_rows](#) = 8, [xc4vlx60_num_cols](#) = 67 }

Static Protected Attributes

- static const [tile_types xc4vlx60_row_layout](#) []
- static const [string xc4vlx60_name](#) = "XC4VLX60"
- static const int [xc4vlx60_id](#) = 0x016B4093U

8.34.1 Detailed Description

Definition at line 33 of file v4_devices.h.

8.34.2 Member Enumeration Documentation

8.34.2.1 anonymous enum [protected]

Enumerator:

xc4vlx60_num_rows
xc4vlx60_num_cols

Definition at line 35 of file v4_devices.h.

8.34.3 Constructor & Destructor Documentation

8.34.3.1 openpr::bitstream::xc4vlx60::xc4vlx60 () [inline]

Definition at line 40 of file v4_devices.h.

Here is the call graph for this function:



8.34.4 Member Data Documentation

8.34.4.1 const int openpr::bitstream::xc4vlx60::xc4vlx60_id = 0x016B4093U [static, protected]

Definition at line 38 of file v4_devices.h.

8.34.4.2 const string openpr::bitstream::xc4vlx60::xc4vlx60_name = "XC4VLX60" [static, protected]

XC4VLX60 Part

Definition at line 37 of file v4_devices.h.

8.34.4.3 const tile_types openpr::bitstream::xc4vlx60::xc4vlx60_row_layout [static, protected]

Definition at line 36 of file v4_devices.h.

The documentation for this class was generated from the following files:

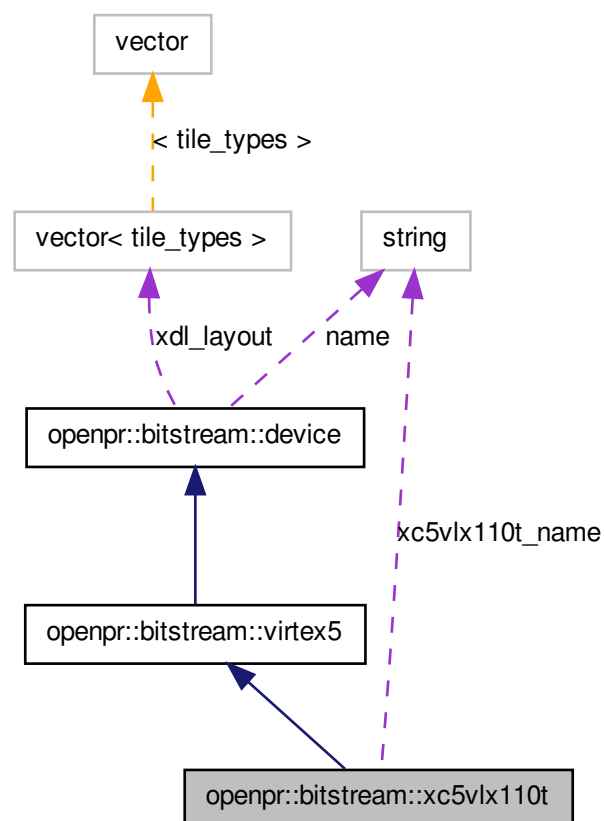
- [openpr/bitstream/v4_devices.h](#)
- [openpr/bitstream/v4_devices.cpp](#)

8.35 openpr::bitstream::xc5vlx110t Class Reference

```
#include <v5_devices.h>
```

Inherits [openpr::bitstream::virtex5](#).

Collaboration diagram for openpr::bitstream::xc5vlx110t:



Public Member Functions

- [xc5vlx110t\(\)](#)

Protected Types

- enum { [xc5vlx110t_num_rows](#) = 8, [xc5vlx110t_num_cols](#) = 70 }

Static Protected Attributes

- static const [tile_types](#) [xc5vlx110t_row_layout](#) []
- static const [string](#) [xc5vlx110t_name](#) = "XC5VLX110T"
- static const int [xc5vlx110t_id](#) = 0x2AD6093

8.35.1 Detailed Description

Definition at line 61 of file v5_devices.h.

8.35.2 Member Enumeration Documentation

8.35.2.1 anonymous enum [protected]

Enumerator:

xc5vlx110t_num_rows
xc5vlx110t_num_cols

Definition at line 63 of file v5_devices.h.

8.35.3 Constructor & Destructor Documentation

8.35.3.1 openpr::bitstream::xc5vlx110t::xc5vlx110t() [inline]

Definition at line 68 of file v5_devices.h.

Here is the call graph for this function:



8.35.4 Member Data Documentation

8.35.4.1 `const int openpr::bitstream::xc5vlx110t::xc5vlx110t_id = 0x2AD6093` `[static, protected]`

Definition at line 66 of file v5_devices.h.

8.35.4.2 `const string openpr::bitstream::xc5vlx110t::xc5vlx110t_name = "XC5VLX110T"` `[static, protected]`

Definition at line 65 of file v5_devices.h.

8.35.4.3 `const tile_types openpr::bitstream::xc5vlx110t::xc5vlx110t_row_layout` `[static, protected]`

Definition at line 64 of file v5_devices.h.

The documentation for this class was generated from the following files:

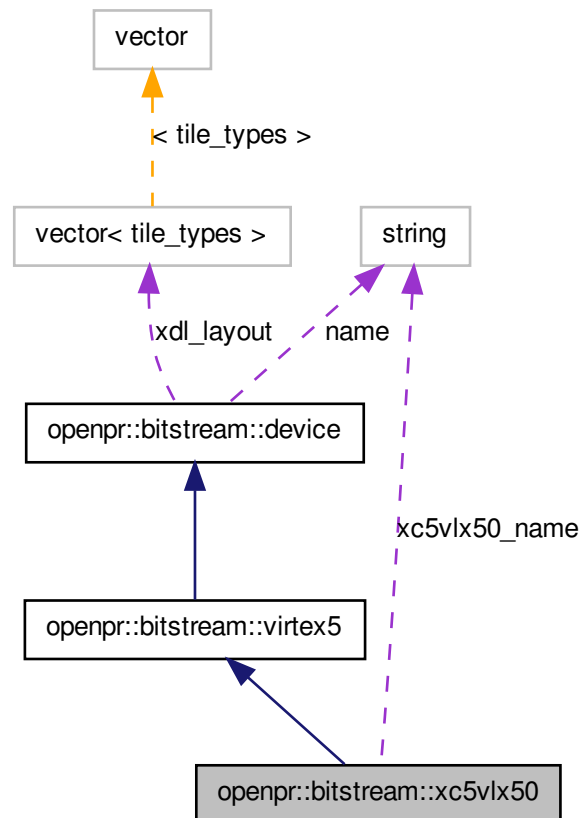
- [openpr/bitstream/v5_devices.h](#)
- [openpr/bitstream/v5_devices.cpp](#)

8.36 openpr::bitstream::xc5vlx50 Class Reference

```
#include <v5_devices.h>
```

Inherits [openpr::bitstream::virtex5](#).

Collaboration diagram for openpr::bitstream::xc5vlx50:



Public Member Functions

- [xc5vlx50\(\)](#)

Protected Types

- enum { [xc5vlx50_num_rows](#) = 6, [xc5vlx50_num_cols](#) = 39 }

Static Protected Attributes

- static const [tile_types](#) [xc5vlx50_row_layout](#) []
- static const [string](#) [xc5vlx50_name](#) = "XC5VLX50"
- static const int [xc5vlx50_id](#) = 0x2896093

8.36.1 Detailed Description

Definition at line 22 of file v5_devices.h.

8.36.2 Member Enumeration Documentation

8.36.2.1 anonymous enum [protected]

Enumerator:

xc5vlx50_num_rows

xc5vlx50_num_cols

Definition at line 24 of file v5_devices.h.

8.36.3 Constructor & Destructor Documentation

8.36.3.1 openpr::bitstream::xc5vlx50::xc5vlx50 () [inline]

Definition at line 29 of file v5_devices.h.

Here is the call graph for this function:



8.36.4 Member Data Documentation

8.36.4.1 const int openpr::bitstream::xc5vlx50::xc5vlx50_id = 0x2896093 [static, protected]

Definition at line 27 of file v5_devices.h.

8.36.4.2 const string openpr::bitstream::xc5vlx50::xc5vlx50_name = "XC5VLX50" [static, protected]

Definition at line 26 of file v5_devices.h.

8.36.4.3 const tile_types openpr::bitstream::xc5vlx50::xc5vlx50_row_layout [static, protected]

Definition at line 25 of file v5_devices.h.

The documentation for this class was generated from the following files:

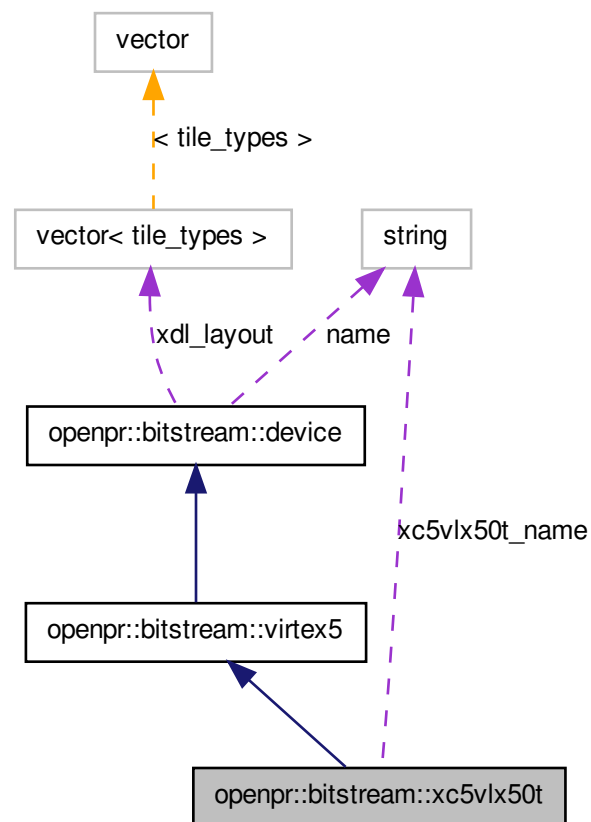
- [openpr/bitstream/v5_devices.h](#)
- [openpr/bitstream/v5_devices.cpp](#)

8.37 openpr::bitstream::xc5vlx50t Class Reference

#include <v5_devices.h>

Inherits [openpr::bitstream::virtex5](#).

Collaboration diagram for openpr::bitstream::xc5vlx50t:



Public Member Functions

- [xc5vlx50t\(\)](#)

Protected Types

- enum { [xc5vlx50t_num_rows](#) = 6, [xc5vlx50t_num_cols](#) = 42 }

Static Protected Attributes

- static const [tile_types xc5vlx50t_row_layout](#) []
- static const [string xc5vlx50t_name](#) = "XC5VLX50T"
- static const int [xc5vlx50t_id](#) = 0x02A96093

8.37.1 Detailed Description

Definition at line 35 of file v5_devices.h.

8.37.2 Member Enumeration Documentation

8.37.2.1 anonymous enum [protected]

Enumerator:

xc5vlx50t_num_rows
xc5vlx50t_num_cols

Definition at line 37 of file v5_devices.h.

8.37.3 Constructor & Destructor Documentation

8.37.3.1 openpr::bitstream::xc5vlx50t::xc5vlx50t () [inline]

Definition at line 42 of file v5_devices.h.

Here is the call graph for this function:



8.37.4 Member Data Documentation

8.37.4.1 const int openpr::bitstream::xc5vlx50t::xc5vlx50t_id = 0x02A96093 [static, protected]

Definition at line 40 of file v5_devices.h.

8.37.4.2 const string openpr::bitstream::xc5vlx50t::xc5vlx50t_name = "XC5VLX50T" [static, protected]

Definition at line 39 of file v5_devices.h.

8.37.4.3 `const tile_types openpr::bitstream::xc5vlx50t::xc5vlx50t_row_layout` `[static, protected]`

Definition at line 38 of file `v5_devices.h`.

The documentation for this class was generated from the following files:

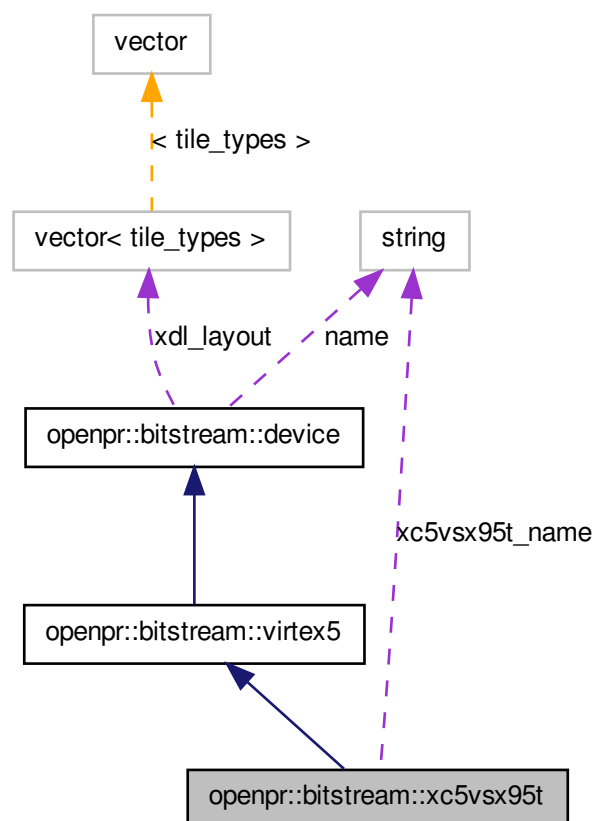
- [openpr/bitstream/v5_devices.h](#)
- [openpr/bitstream/v5_devices.cpp](#)

8.38 `openpr::bitstream::xc5vsx95t` Class Reference

```
#include <v5_devices.h>
```

Inherits [openpr::bitstream::virtex5](#).

Collaboration diagram for `openpr::bitstream::xc5vsx95t`:



Public Member Functions

- [xc5vsx95t](#) ()

Protected Types

- enum { [xc5vsx95t_num_rows](#) = 8, [xc5vsx95t_num_cols](#) = 77 }

Static Protected Attributes

- static const [tile_types](#) [xc5vsx95t_row_layout](#) []
- static const [string](#) [xc5vsx95t_name](#) = "xc5vsx95t"
- static const int [xc5vsx95t_id](#) = 0x02ece093

8.38.1 Detailed Description

Definition at line 48 of file v5_devices.h.

8.38.2 Member Enumeration Documentation

8.38.2.1 anonymous enum [protected]

Enumerator:

xc5vsx95t_num_rows
xc5vsx95t_num_cols

Definition at line 50 of file v5_devices.h.

8.38.3 Constructor & Destructor Documentation

8.38.3.1 openpr::bitstream::xc5vsx95t::xc5vsx95t () [inline]

Definition at line 55 of file v5_devices.h.

Here is the call graph for this function:



8.38.4 Member Data Documentation

8.38.4.1 `const int openpr::bitstream::xc5vsx95t::xc5vsx95t_id = 0x02ece093` `[static, protected]`

Definition at line 53 of file v5_devices.h.

8.38.4.2 `const string openpr::bitstream::xc5vsx95t::xc5vsx95t_name = "xc5vsx95t"` `[static, protected]`

Definition at line 52 of file v5_devices.h.

8.38.4.3 `const tile_types openpr::bitstream::xc5vsx95t::xc5vsx95t_row_layout` `[static, protected]`

Definition at line 51 of file v5_devices.h.

The documentation for this class was generated from the following files:

- [openpr/bitstream/v5_devices.h](#)
- [openpr/bitstream/v5_devices.cpp](#)

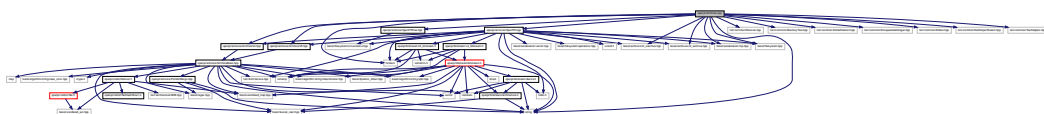
Chapter 9

File Documentation

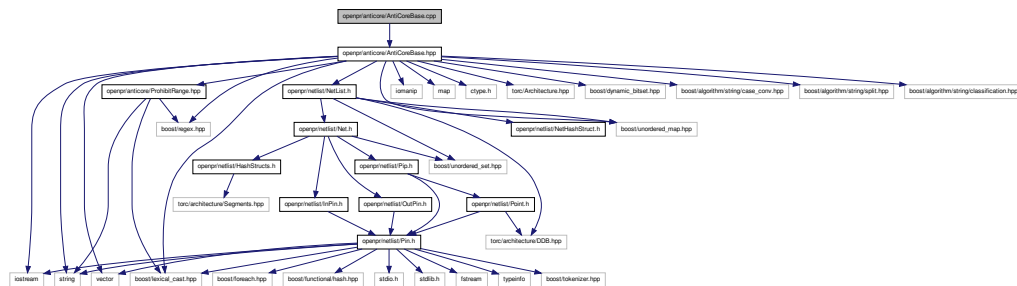
9.1 openpr/AntiCore.cpp File Reference

```
#include "openpr/anticore/AntiCoreV5.hpp"
#include "openpr/anticore/AntiCoreV4.hpp"
#include "openpr/anticore/OpenPRTree.hpp"
#include "openpr/anticore/OpenPR.hpp"
#include "torc/common/Devices.hpp"
#include "torc/common/DirectoryTree.hpp"
#include "torc/common/DottedVersion.hpp"
#include "torc/common/EncapsulatedInteger.hpp"
#include "torc/common/Endian.hpp"
#include "torc/common/NullOutputStream.hpp"
#include "torc/common/TestHelpers.hpp"
#include <string>
#include <fstream>
#include <boost/archive/xml_oarchive.hpp>
#include <boost/archive/xml_iarchive.hpp>
#include <boost/serialization/nvp.hpp>
#include <boost/filesystem.hpp>
```

Include dependency graph for AntiCore.cpp:



Include dependency graph for AntiCoreBase.cpp:



Namespaces

- namespace `openpr`

9.3 openpr/anticore/AntiCoreBase.hpp File Reference

```
#include <iostream>
#include <iomanip>
#include <vector>
#include <map>
#include <string>
#include <ctype.h>
#include "torc/Architecture.hpp"
#include "openpr/anticore/ProhibitRange.hpp"
#include "openpr/netlist/NetList.h"
#include <boost/dynamic_bitset.hpp>
#include <boost/unordered_map.hpp>
#include <boost/regex.hpp>
#include <boost/lexical_cast.hpp>
#include <boost/algorithm/string/case_conv.hpp>
#include <boost/algorithm/string/split.hpp>
#include <boost/algorithm/string/classification.hpp>
```



```
#include "openpr/anticore/AntiCoreV4.hpp"
```

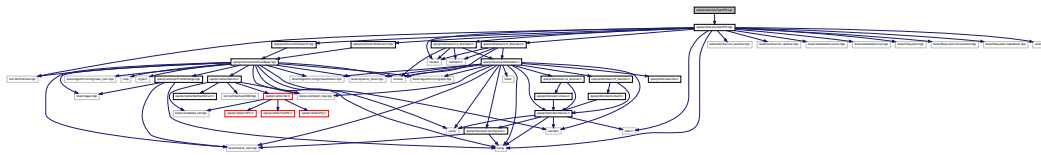
[illegible]

- namespace `openpr`

```
#include "openpr/anticore/AntiCoreBase.hpp"
```

[illegible]

Include dependency graph for OpenPR.cpp:



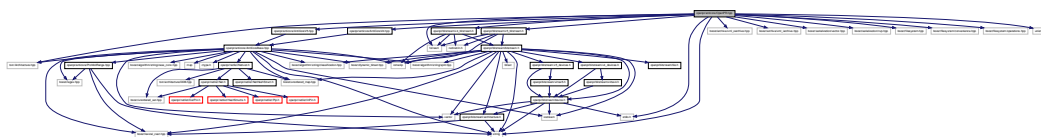
Namespaces

- namespace `openpr`

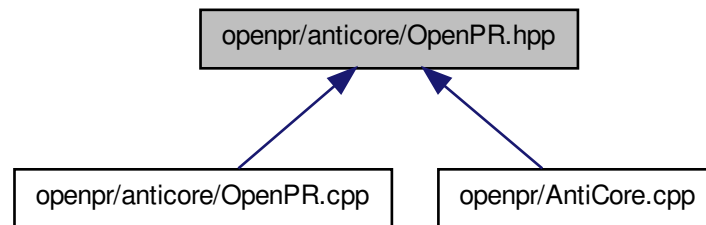
9.9 openpr/anticore/OpenPR.hpp File Reference

```
#include "openpr/anticore/AntiCoreV4.hpp"
#include "openpr/anticore/AntiCoreV5.hpp"
#include "torc/Architecture.hpp"
#include "openpr/bitstream/v5_bitstream.h"
#include "openpr/bitstream/v4_bitstream.h"
#include <fstream>
#include <string>
#include <boost/archive/xml_oarchive.hpp>
#include <boost/archive/xml_iarchive.hpp>
#include <boost/serialization/vector.hpp>
#include <boost/serialization/nvp.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/convenience.hpp>
#include <boost/filesystem/operations.hpp>
#include <stdio.h>
#include <unistd.h>
```

Include dependency graph for OpenPR.hpp:



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



Classes

- class [openpr::openPR](#)

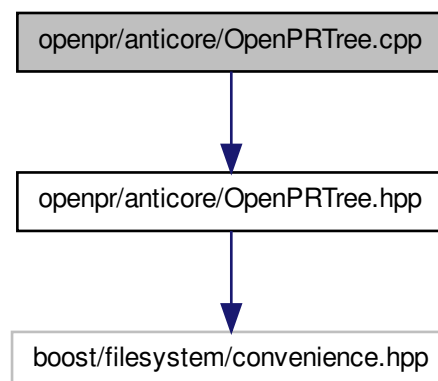
Namespaces

- namespace [openpr](#)

9.10 openpr/anticore/OpenPRTree.cpp File Reference

```
#include "openpr/anticore/OpenPRTree.hpp"
```

Include dependency graph for OpenPRTree.cpp:



Namespaces

- namespace `openpr`

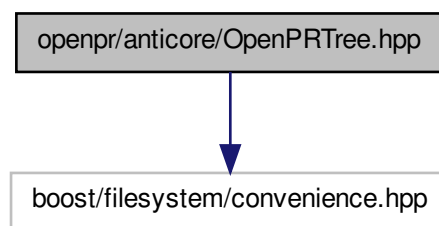
Functions

- const std::string `openpr::cEdaNameConst` ("eda")
- const std::string `openpr::cArchitectureNameConst` ("architecture")
- const std::string `openpr::cXilinxNameConst` ("xilinx")

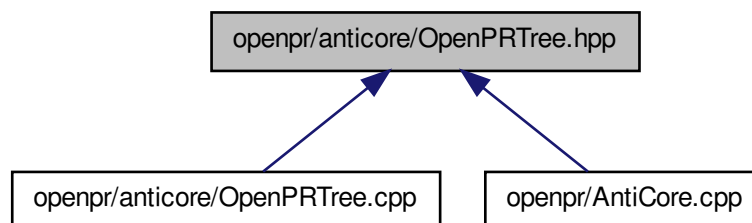
9.11 openpr/anticore/OpenPRTree.hpp File Reference

```
#include <boost/filesystem/convenience.hpp>
```

Include dependency graph for OpenPRTree.hpp:



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



Classes

- class [openpr::OpenPRTree](#)

Namespaces

- namespace [openpr](#)

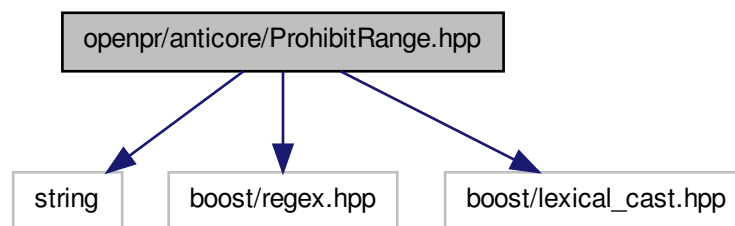
9.12 openpr/anticore/ProhibitRange.hpp File Reference

```
#include <string>
```

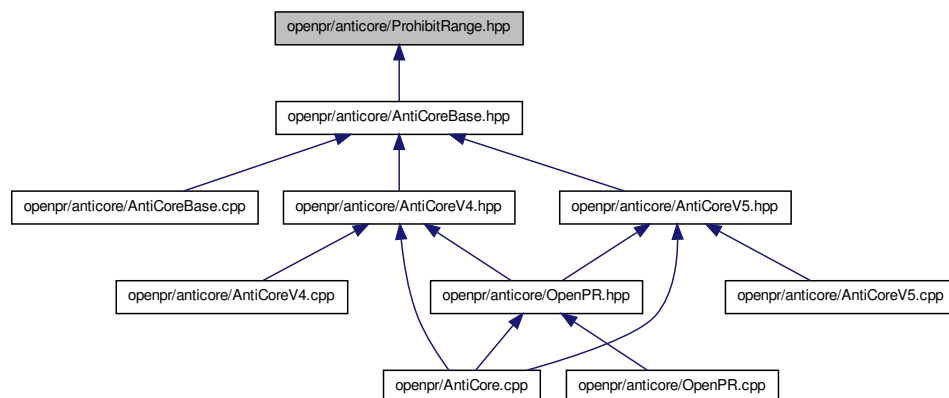
```
#include <boost/regex.hpp>
```

```
#include <boost/lexical_cast.hpp>
```

Include dependency graph for ProhibitRange.hpp:



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



Classes

- struct [openpr::prohibitRange](#)

Namespaces

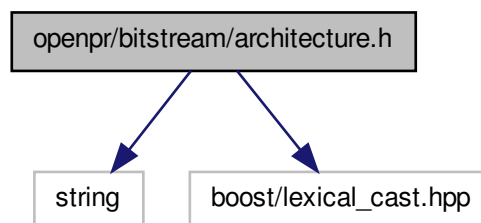
- namespace [openpr](#)

9.13 openpr/bitstream/architecture.h File Reference

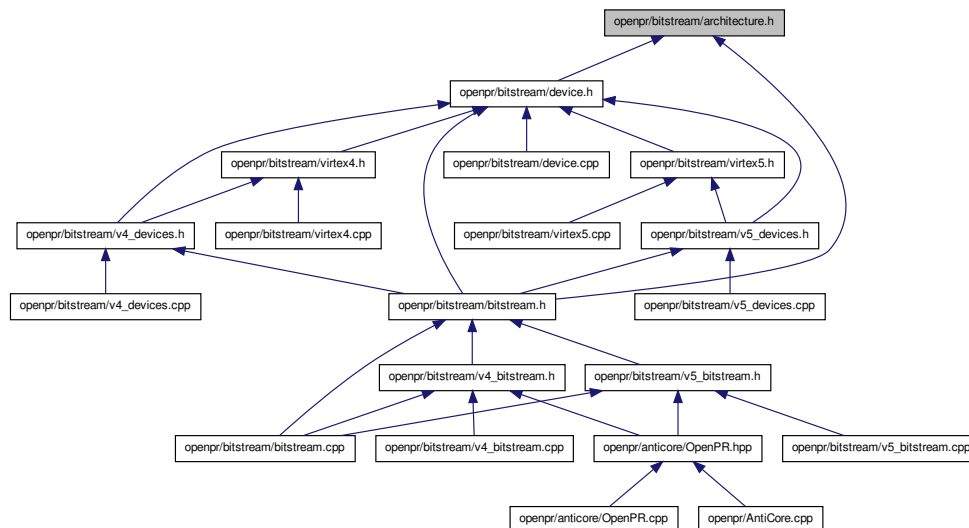
```
#include <string>
```

```
#include <boost/lexical_cast.hpp>
```

Include dependency graph for architecture.h:



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



Classes

- struct [openpr::bitstream::frame_addr](#)
- class [openpr::bitstream::architecture](#)

Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

Enumerations

- enum [openpr::bitstream::tile_types](#) {

openpr::bitstream::IOB,	openpr::bitstream::GCLK,	openpr::bitstream::CLB,
openpr::bitstream::DSP48,		
openpr::bitstream::BRAM,	openpr::bitstream::BRAM_INT,	openpr::bitstream::TRANSCV,
openpr::bitstream::PAD,		
openpr::bitstream::NUM_TILE_TYPES,		openpr::bitstream::MULTIPLE,
openpr::bitstream::INVALID }		

9.14 openpr/bitstream/bitstream.cpp File Reference

```
#include "openpr/bitstream/bitstream.h"
#include "openpr/bitstream/v4_bitstream.h"
```

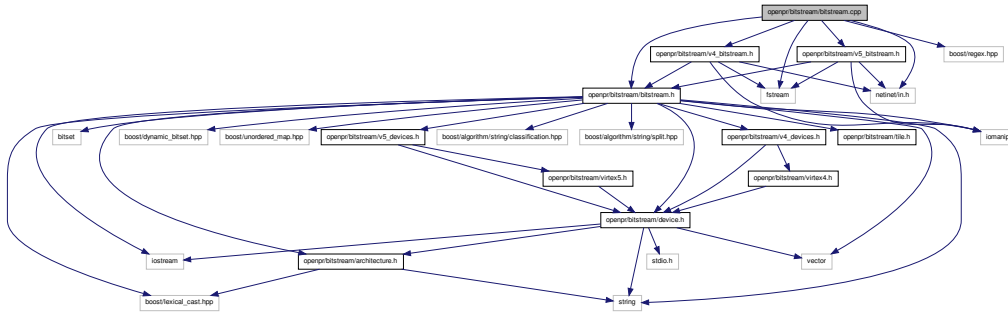
```
#include "openpr/bitstream/v5_bitstream.h"
```

```
#include <fstream>
```

```
#include <netinet/in.h>
```

```
#include <boost/regex.hpp>
```

Include dependency graph for bitstream.cpp:



Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.15 openpr/bitstream/bitstream.h File Reference

```
#include <string>
```

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <bitset>
```

```
#include <vector>
```

```
#include <boost/dynamic_bitset.hpp>
```

```
#include <boost/unordered_map.hpp>
```

```
#include <boost/lexical_cast.hpp>
```

```
#include <boost/algorithm/string/classification.hpp>
```

```
#include <boost/algorithm/string/split.hpp>
```

```
#include "openpr/bitstream/architecture.h"
```

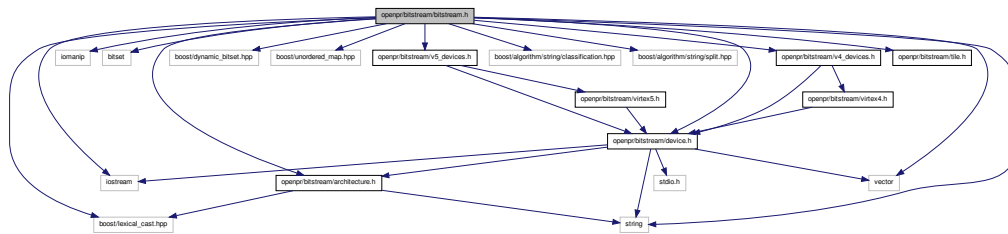
```
#include "openpr/bitstream/device.h"
```

```
#include "openpr/bitstream/tile.h"
```

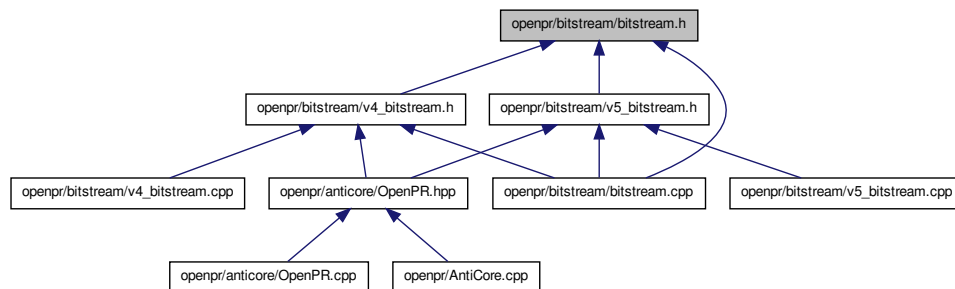
```
#include "openpr/bitstream/v4_devices.h"
```

```
#include "openpr/bitstream/v5_devices.h"
```


Include dependency graph for bitstream.h:



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



Classes

- class `openpr::bitstream::bitstream`

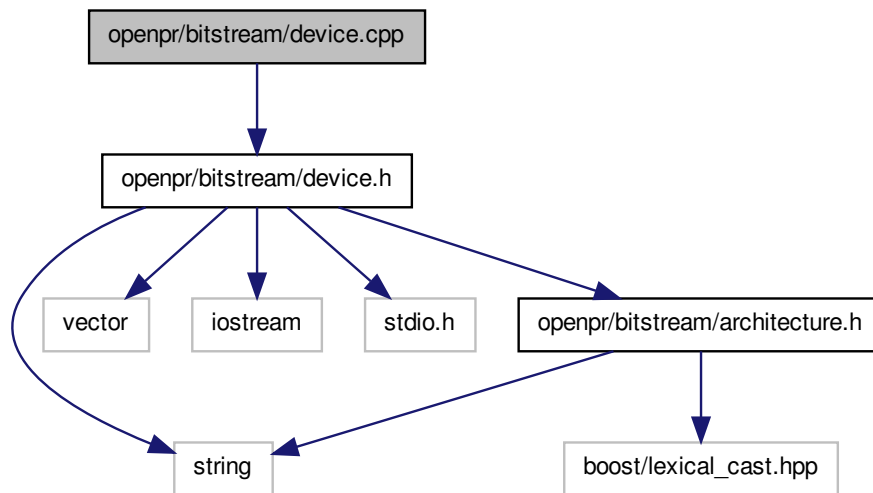
Namespaces

- namespace `openpr`
- namespace `openpr::bitstream`

9.16 openpr/bitstream/device.cpp File Reference

```
#include "openpr/bitstream/device.h"
```

Include dependency graph for device.cpp:



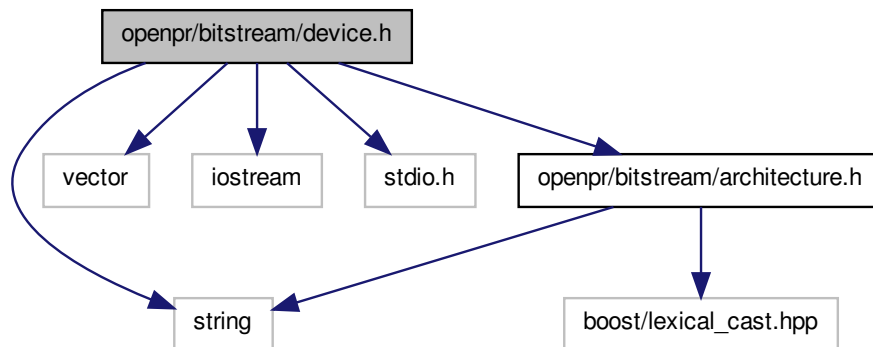
Namespaces

- namespace `openpr`
- namespace `openpr::bitstream`

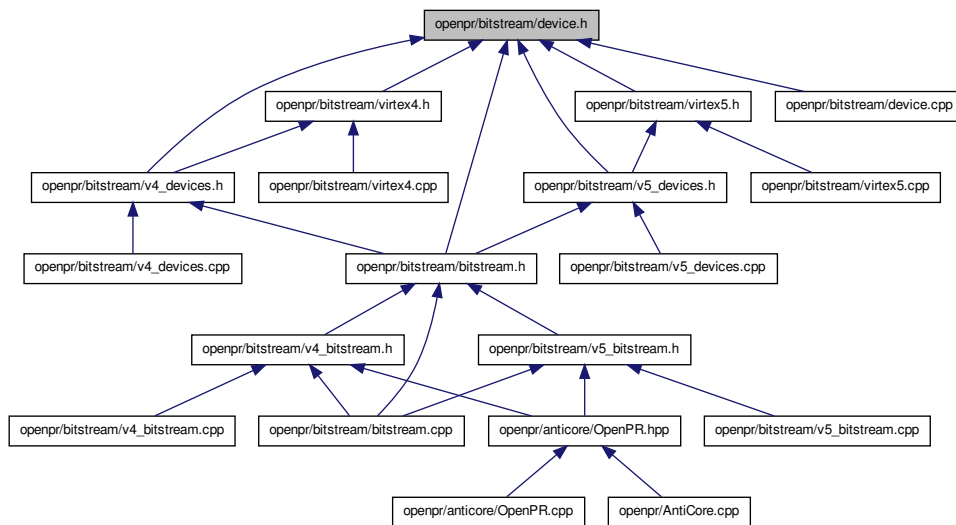
9.17 openpr/bitstream/device.h File Reference

```
#include <string>
#include <vector>
#include <iostream>
#include <stdio.h>
#include "openpr/bitstream/architecture.h"
```

Include dependency graph for device.h:



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



Classes

- class `openpr::bitstream::device`

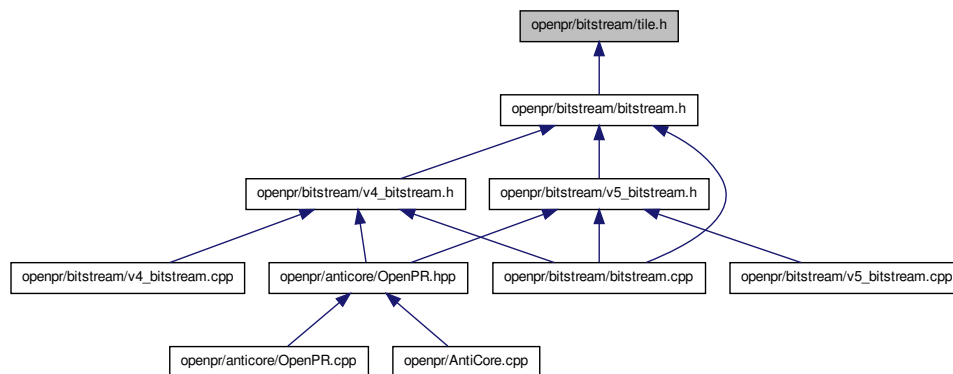
Namespaces

- namespace `openpr`

- namespace [openpr::bitstream](#)

9.18 openpr/bitstream/tile.h File Reference

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



Classes

- struct [openpr::bitstream::tile_coord](#)
- struct [openpr::bitstream::tile_data](#)

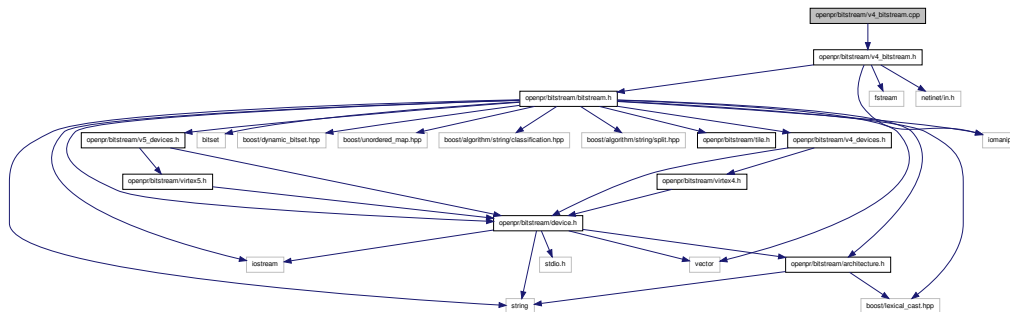
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.19 openpr/bitstream/v4_bitstream.cpp File Reference

```
#include "openpr/bitstream/v4_bitstream.h"
```

Include dependency graph for v4_bitstream.cpp:



Namespaces

- namespace `openpr`
- namespace `openpr::bitstream`

9.20 openpr/bitstream/v4_bitstream.h File Reference

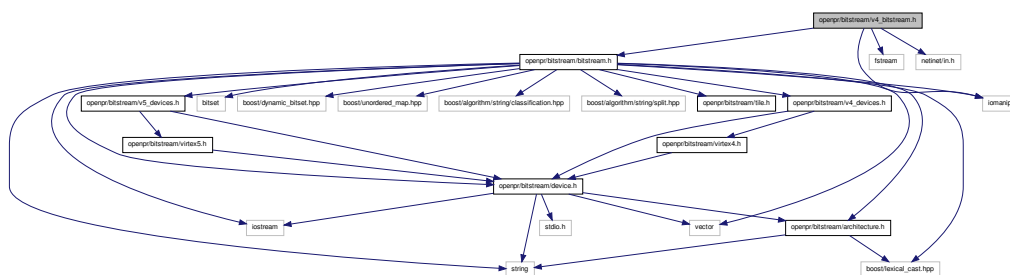
```
#include "openpr/bitstream/bitstream.h"
```

```
#include <fstream>
```

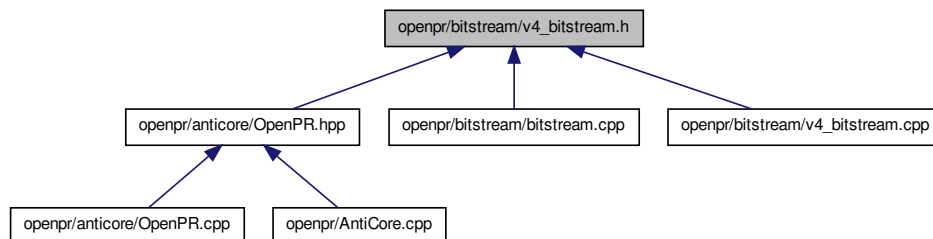
```
#include <iomanip>
```

```
#include <netinet/in.h>
```

Include dependency graph for v4_bitstream.h:



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



Classes

- class [openpr::bitstream::v4_bitstream](#)

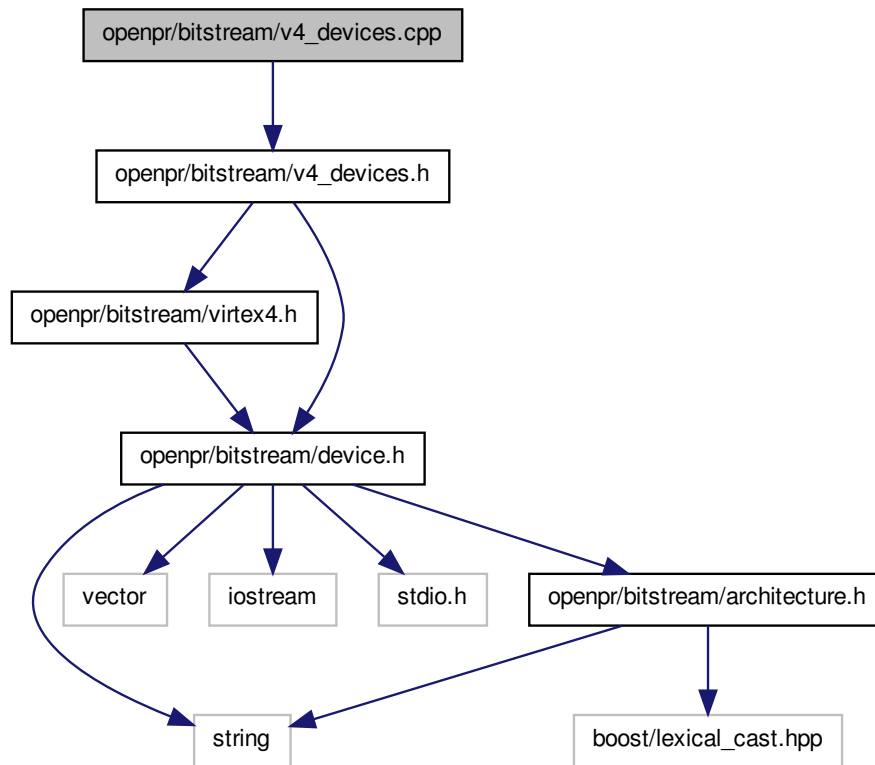
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.21 openpr/bitstream/v4_devices.cpp File Reference

```
#include "openpr/bitstream/v4_devices.h"
```

Include dependency graph for v4_devices.cpp:



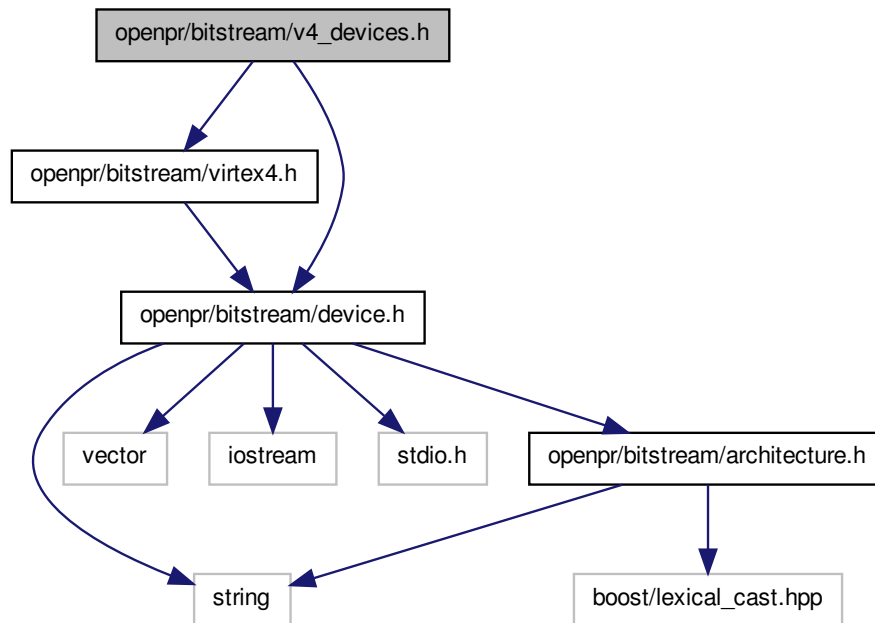
Namespaces

- namespace `openpr`
- namespace `openpr::bitstream`

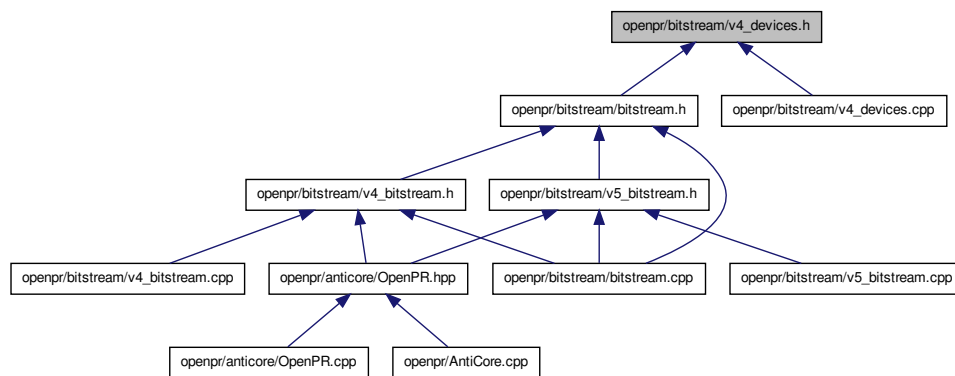
9.22 openpr/bitstream/v4_devices.h File Reference

```
#include "openpr/bitstream/virtex4.h"
#include "openpr/bitstream/device.h"
```

Include dependency graph for `v4_devices.h`:



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



Classes

- class `openpr::bitstream::xc4vlx15`
- class `openpr::bitstream::xc4vlx60`

- class [openpr::bitstream::xc4vfx60](#)

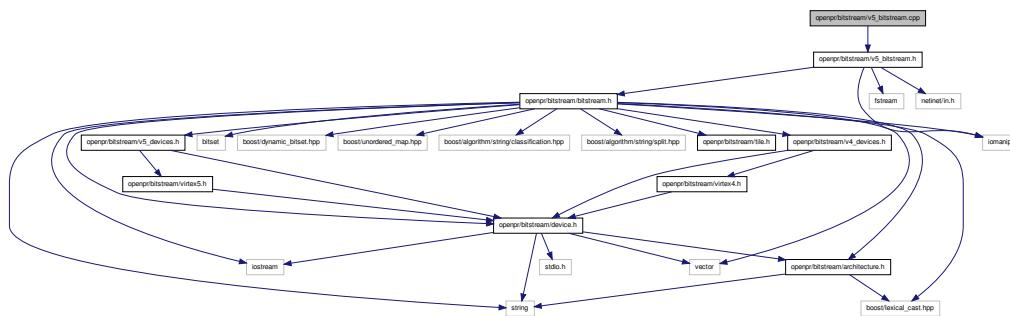
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.23 openpr/bitstream/v5_bitstream.cpp File Reference

```
#include "openpr/bitstream/v5_bitstream.h"
```

Include dependency graph for v5_bitstream.cpp:



Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.24 openpr/bitstream/v5_bitstream.h File Reference

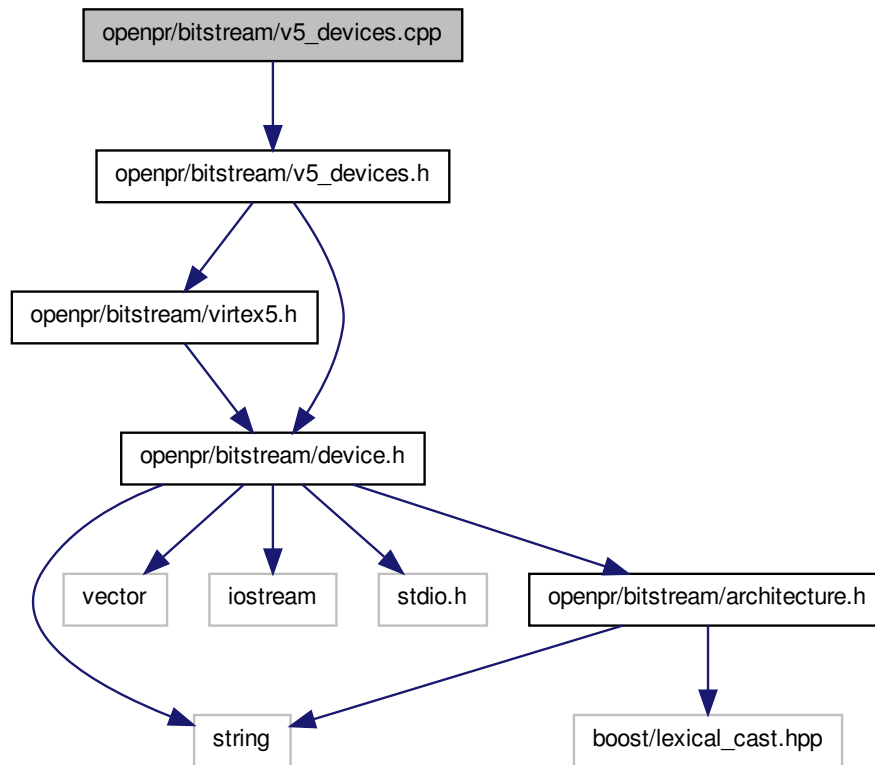
```
#include "openpr/bitstream/bitstream.h"
```

```
#include <fstream>
```

```
#include <iomanip>
```

```
#include <netinet/in.h>
```


Include dependency graph for v5_devices.cpp:



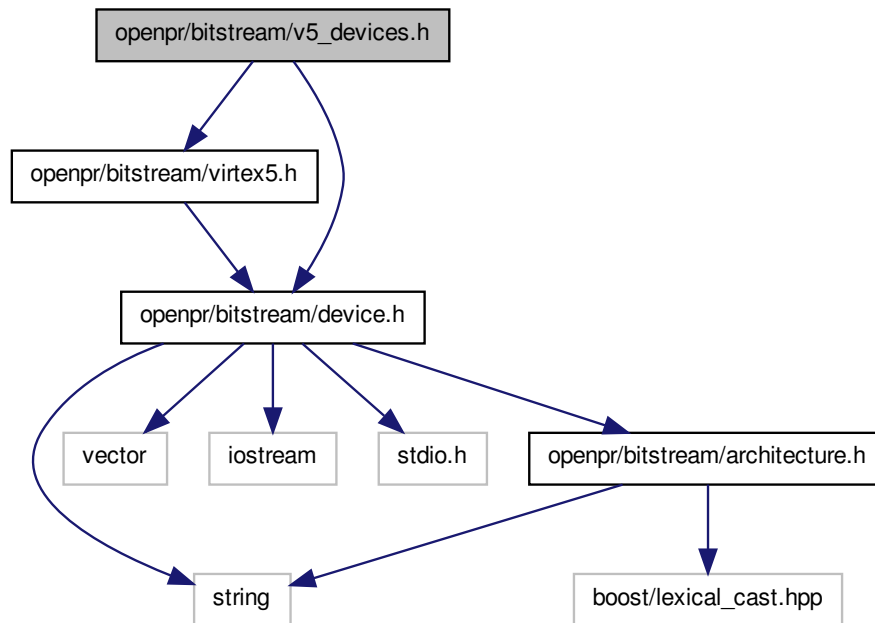
Namespaces

- namespace `openpr`
- namespace `openpr::bitstream`

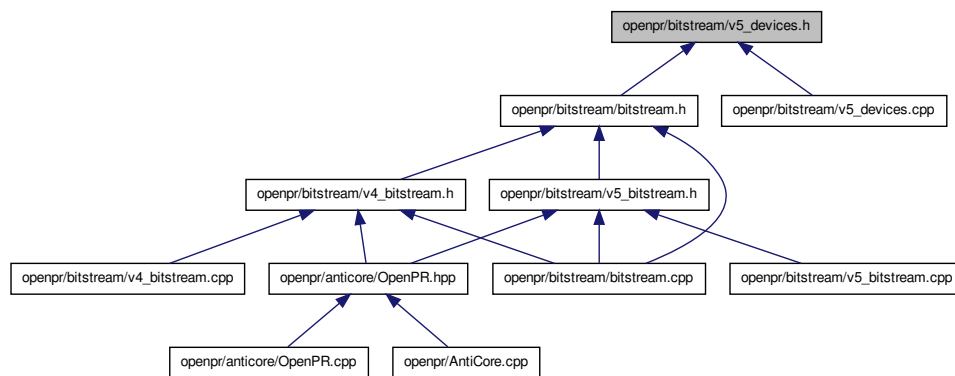
9.26 openpr/bitstream/v5_devices.h File Reference

```
#include "openpr/bitstream/virtex5.h"
#include "openpr/bitstream/device.h"
```

Include dependency graph for v5_devices.h:



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



Classes

- class `openpr::bitstream::xc5vlx50`
- class `openpr::bitstream::xc5vlx50t`

- class [openpr::bitstream::xc5vsx95t](#)
- class [openpr::bitstream::xc5vlx110t](#)

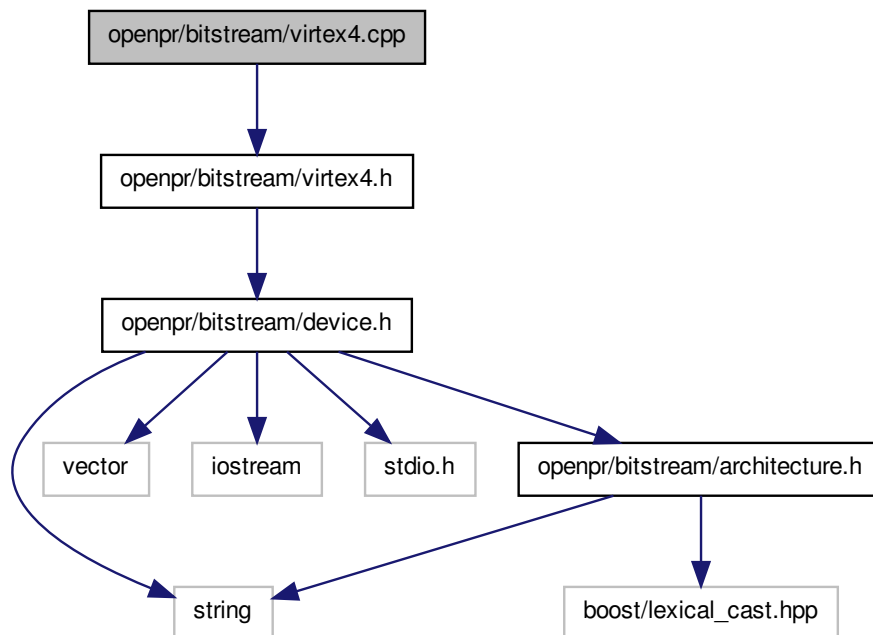
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.27 openpr/bitstream/virtex4.cpp File Reference

```
#include "openpr/bitstream/virtex4.h"
```

Include dependency graph for virtex4.cpp:



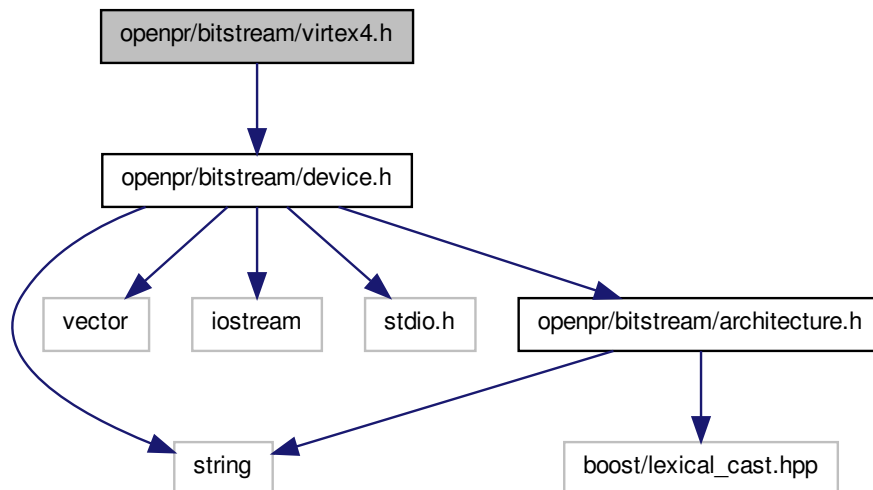
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

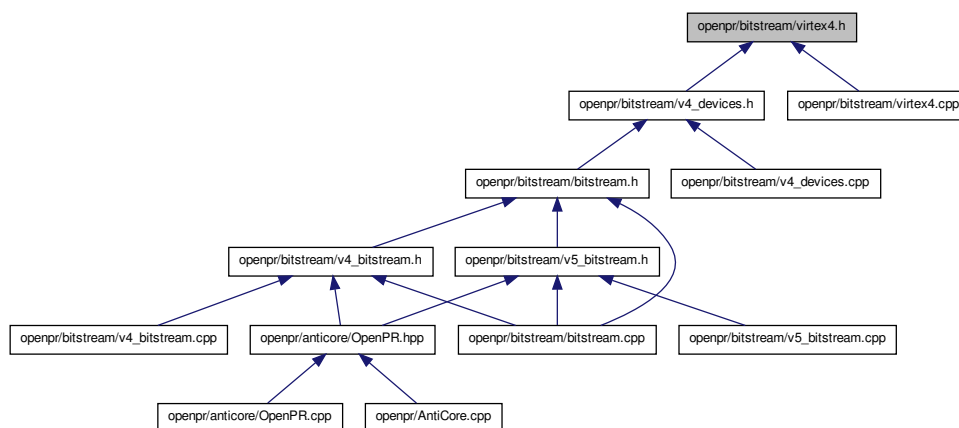
9.28 openpr/bitstream/virtex4.h File Reference

```
#include "openpr/bitstream/device.h"
```

Include dependency graph for virtex4.h:



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



Classes

- class [openpr::bitstream::virtex4](#)

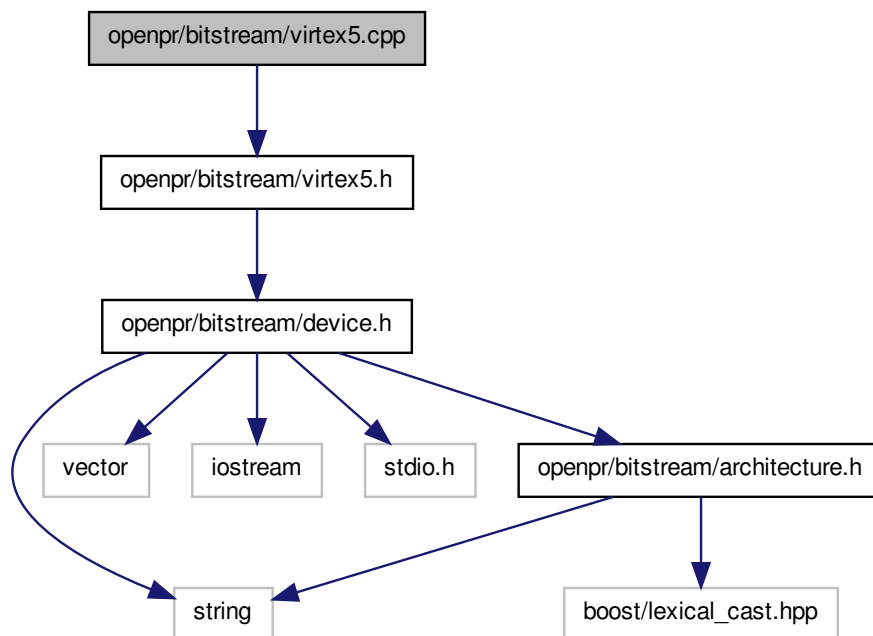
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

9.29 openpr/bitstream/virtex5.cpp File Reference

```
#include "openpr/bitstream/virtex5.h"
```

Include dependency graph for virtex5.cpp:



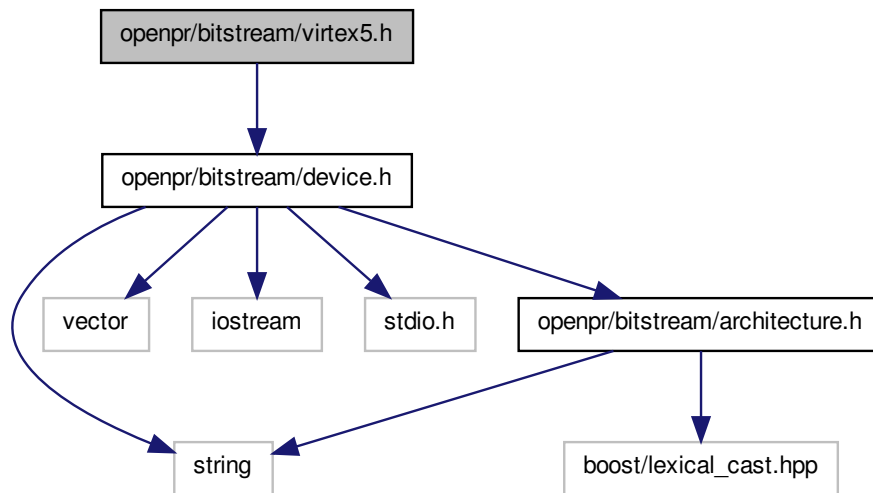
Namespaces

- namespace [openpr](#)
- namespace [openpr::bitstream](#)

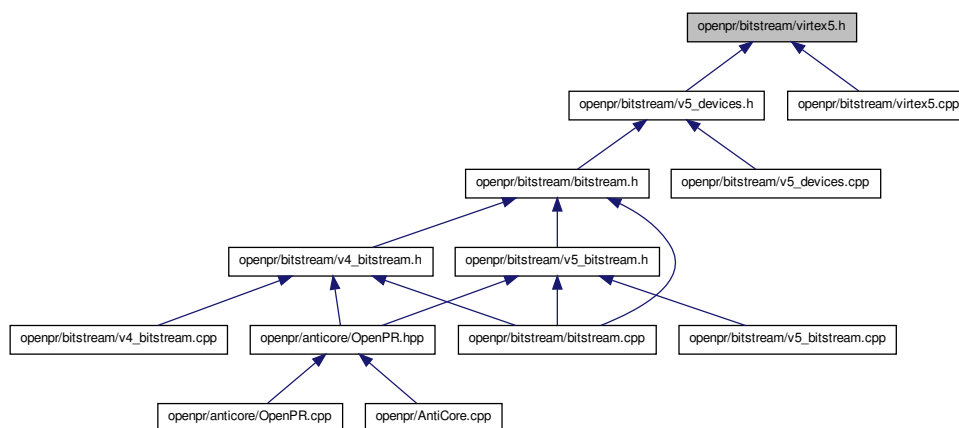
9.30 openpr/bitstream/virtex5.h File Reference

```
#include "openpr/bitstream/device.h"
```

Include dependency graph for virtex5.h:



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



Classes

- class [openpr::bitstream::virtex5](#)

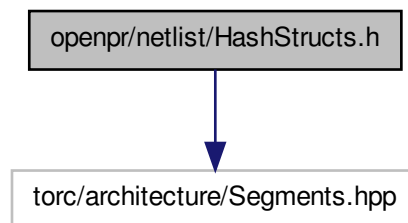
Namespaces

- namespace `openpr`
- namespace `openpr::bitstream`

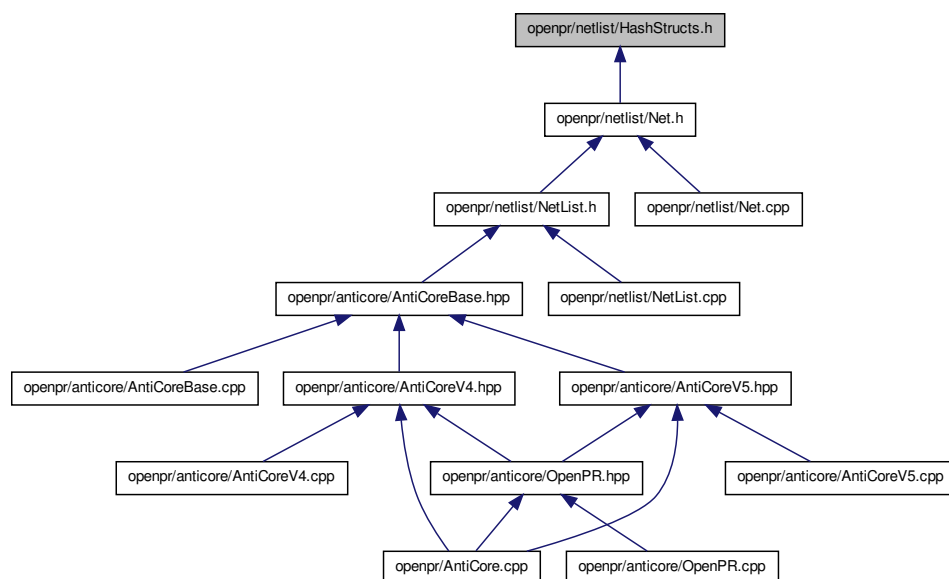
9.31 openpr/netlist/HashStructs.h File Reference

```
#include "torc/architecture/Segments.hpp"
```

Include dependency graph for HashStructs.h:



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



Classes

- struct [openpr::netlist::hash_pip](#)
- struct [openpr::netlist::eq_pip](#)
- struct [openpr::netlist::hash_point](#)
- struct [openpr::netlist::eq_point](#)
- struct [openpr::netlist::hash_segment](#)
- struct [openpr::netlist::eq_segment](#)

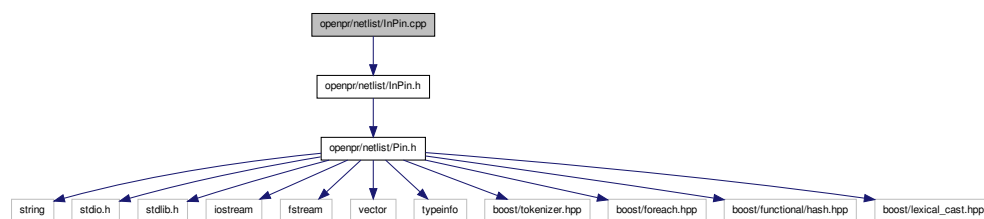
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.32 openpr/netlist/InPin.cpp File Reference

```
#include "openpr/netlist/InPin.h"
```

Include dependency graph for InPin.cpp:



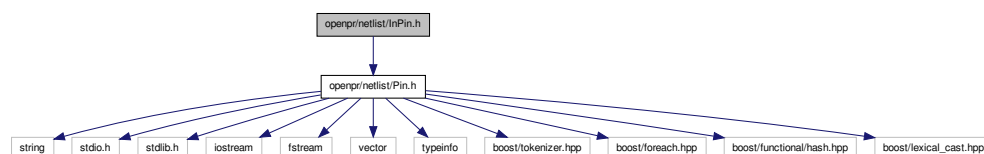
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

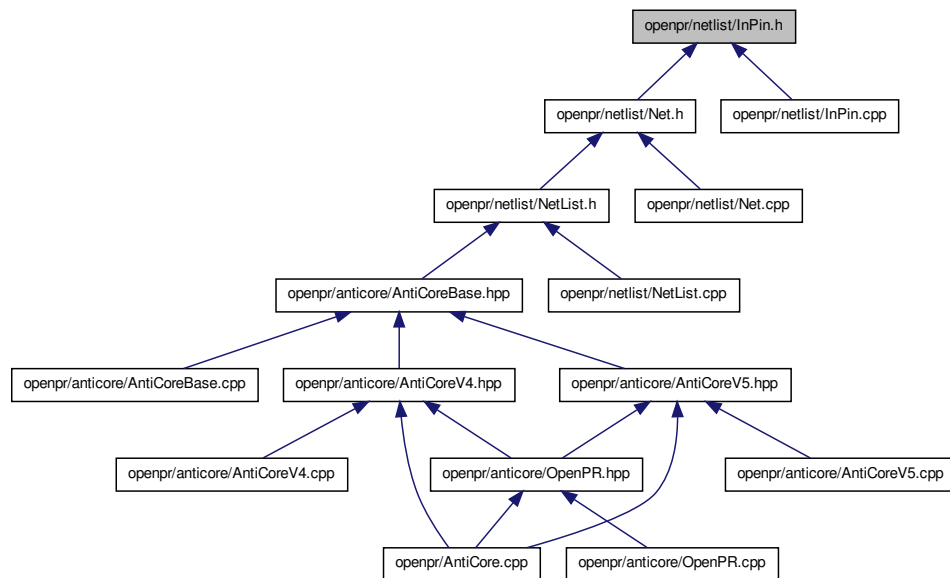
9.33 openpr/netlist/InPin.h File Reference

```
#include "openpr/netlist/Pin.h"
```

Include dependency graph for InPin.h:



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



Classes

- class [openpr::netlist::InPin](#)

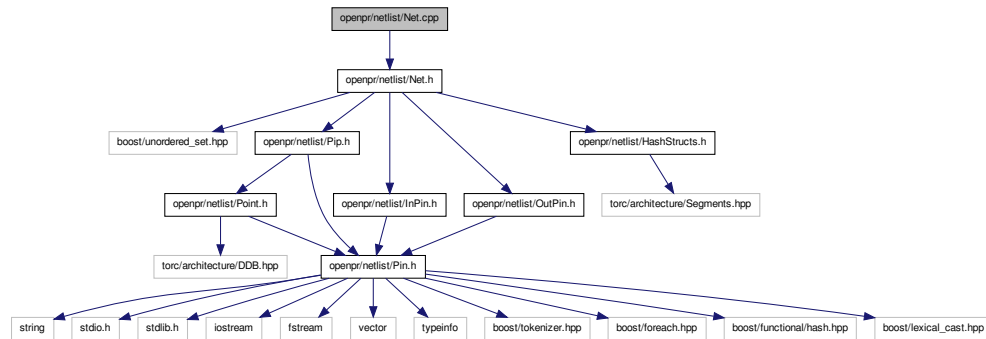
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.34 openpr/netlist/Net.cpp File Reference

```
#include "openpr/netlist/Net.h"
```

Include dependency graph for Net.cpp:



Namespaces

- namespace `openpr`
- namespace `openpr::netlist`

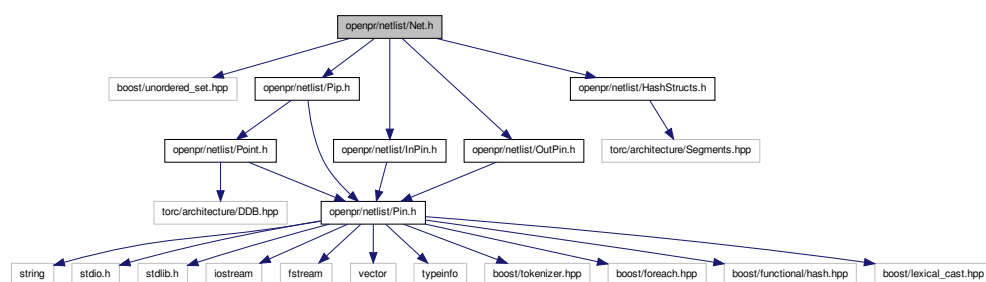
9.35 openpr/netlist/Net.h File Reference

```

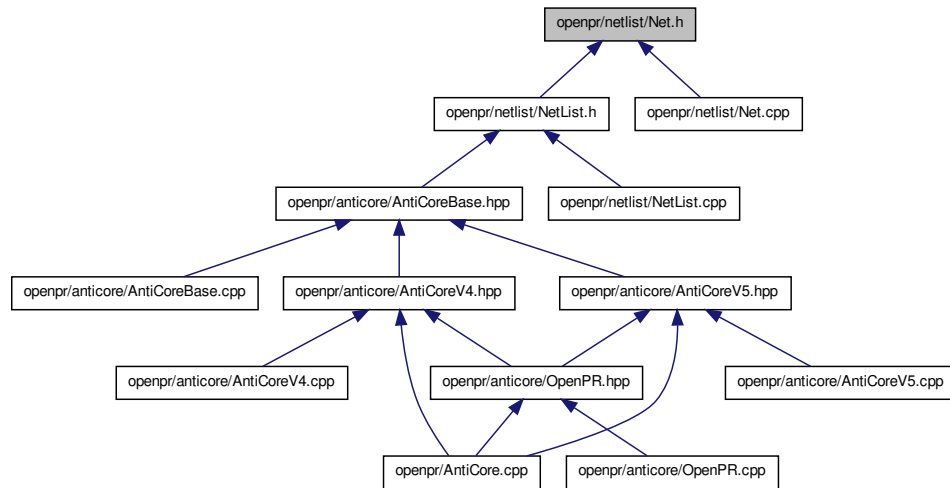
#include <boost/unordered_set.hpp>
#include "openpr/netlist/Pip.h"
#include "openpr/netlist/InPin.h"
#include "openpr/netlist/OutPin.h"
#include "openpr/netlist/HashStructs.h"

```

Include dependency graph for Net.h:



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



Classes

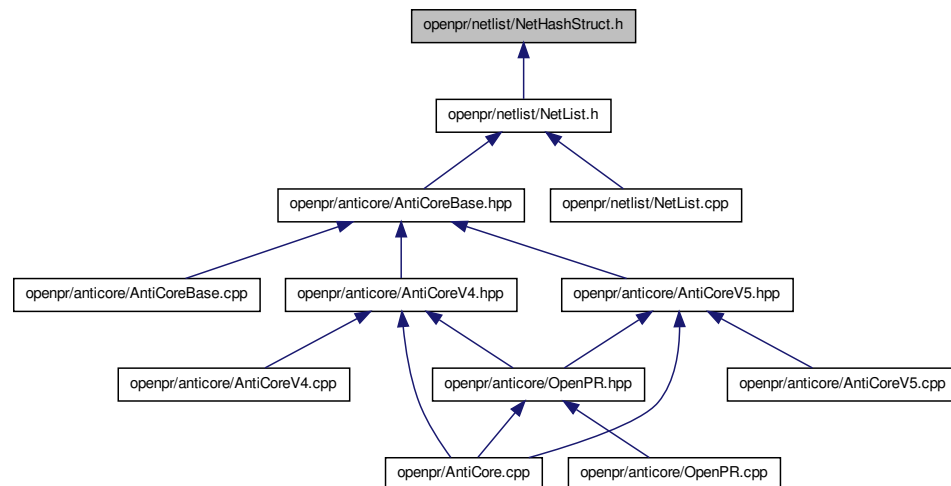
- class [openpr::netlist::Net](#)

Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.36 openpr/netlist/NetHashStruct.h File Reference

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



Classes

- struct [openpr::netlist::hash_net](#)
- struct [openpr::netlist::eq_net](#)

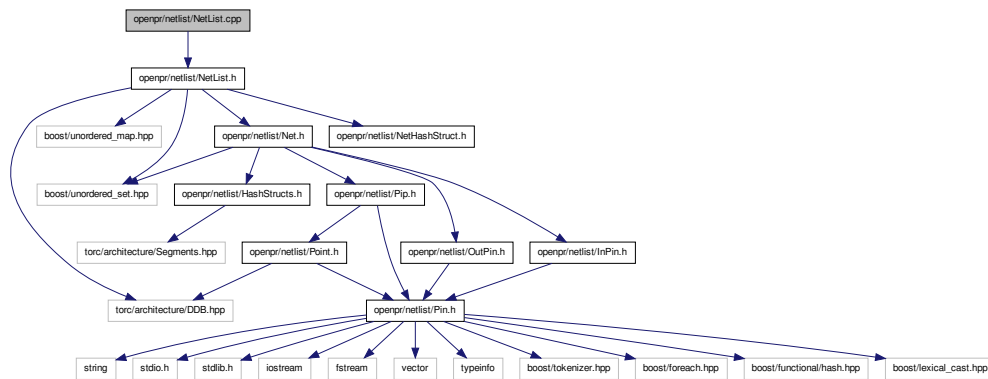
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.37 openpr/netlist/NetList.cpp File Reference

```
#include "openpr/netlist/NetList.h"
```

Include dependency graph for NetList.cpp:



Namespaces

- namespace `openpr`
- namespace `openpr::netlist`

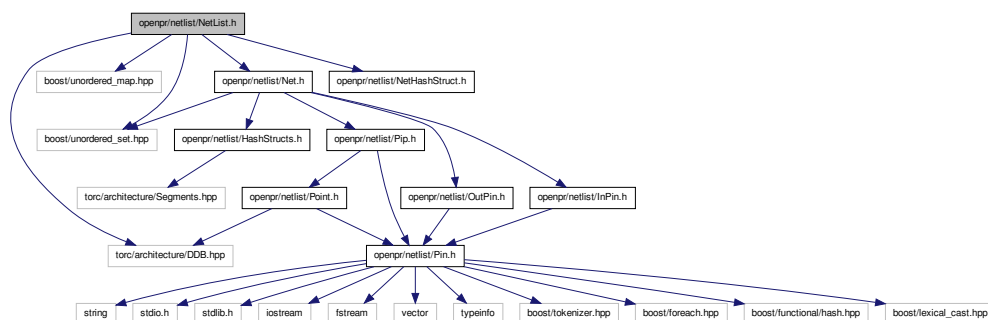
9.38 openpr/netlist/NetList.h File Reference

```

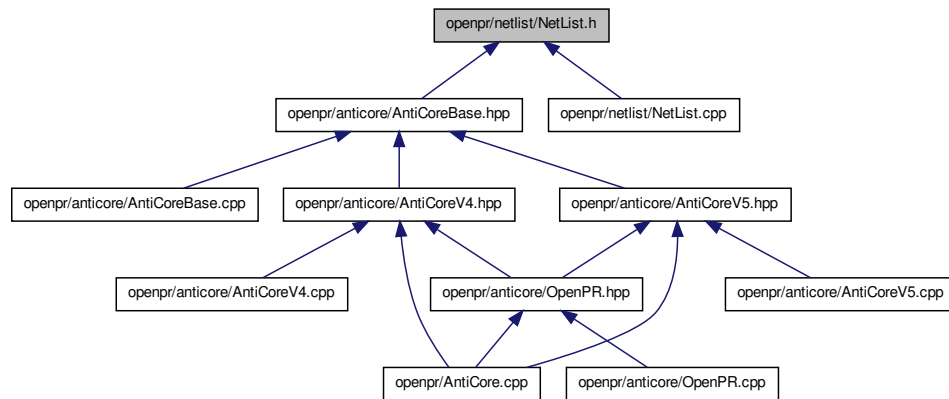
#include <boost/unordered_set.hpp>
#include <boost/unordered_map.hpp>
#include "torc/architecture/DDB.hpp"
#include "openpr/netlist/Net.h"
#include "openpr/netlist/NetHashStruct.h"

```

Include dependency graph for NetList.h:



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



Classes

- class [openpr::netlist::NetList](#)

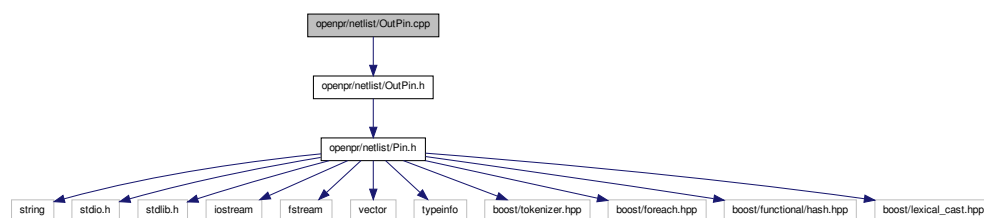
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.39 openpr/netlist/OutPin.cpp File Reference

```
#include "openpr/netlist/OutPin.h"
```

Include dependency graph for OutPin.cpp:



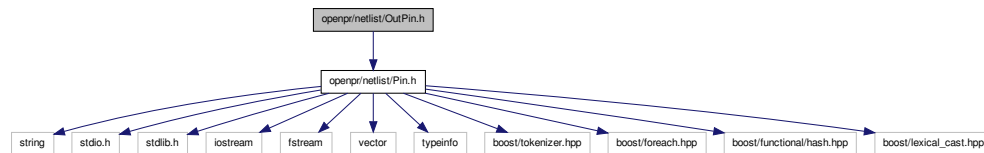
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

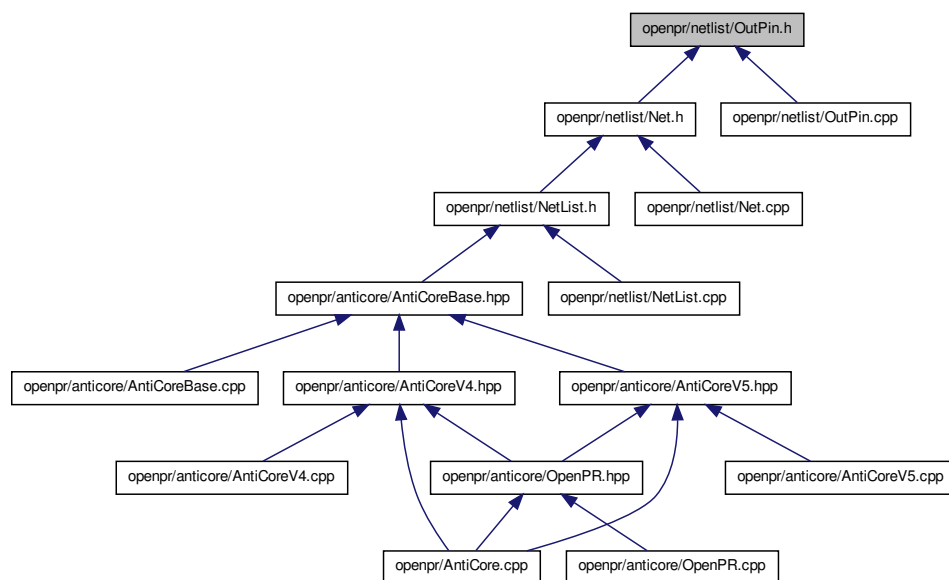
9.40 openpr/netlist/OutPin.h File Reference

```
#include "openpr/netlist/Pin.h"
```

Include dependency graph for OutPin.h:



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



Classes

- class [openpr::netlist::OutPin](#)

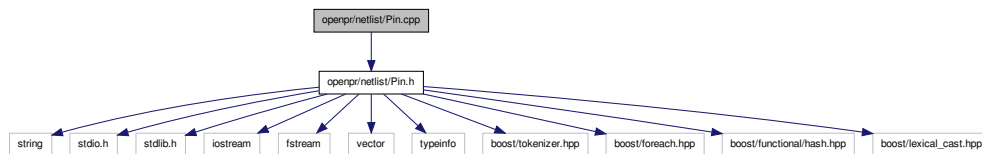
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.41 openpr/netlist/Pin.cpp File Reference

```
#include "openpr/netlist/Pin.h"
```

Include dependency graph for Pin.cpp:



Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

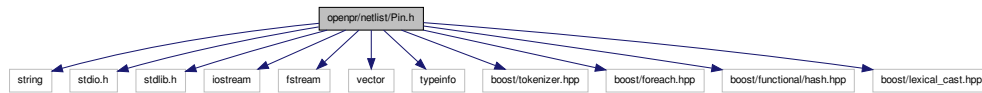
Functions

- `std::size_t openpr::netlist::hash_value` (Pin &pin)
- `std::size_t openpr::netlist::hash_value` (Pin *pin)

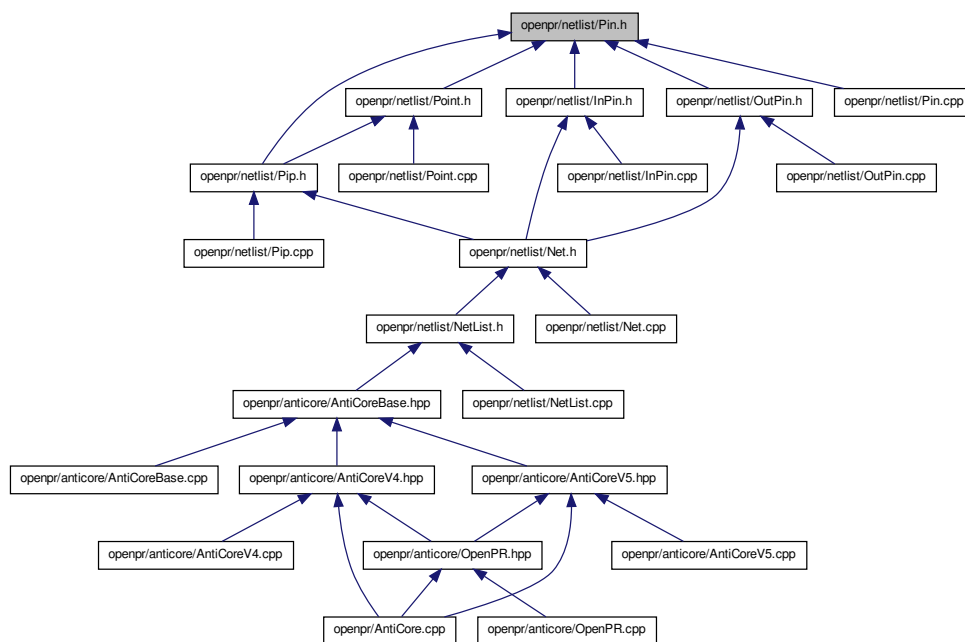
9.42 openpr/netlist/Pin.h File Reference

```
#include <string>
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <fstream>
#include <vector>
#include <typeinfo>
#include <boost/tokenizer.hpp>
#include <boost/foreach.hpp>
#include <boost/functional/hash.hpp>
#include <boost/lexical_cast.hpp>
```

Include dependency graph for Pin.h:



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



Classes

- class `openpr::netlist::Pin`

Namespaces

- namespace `openpr`
- namespace `openpr::netlist`

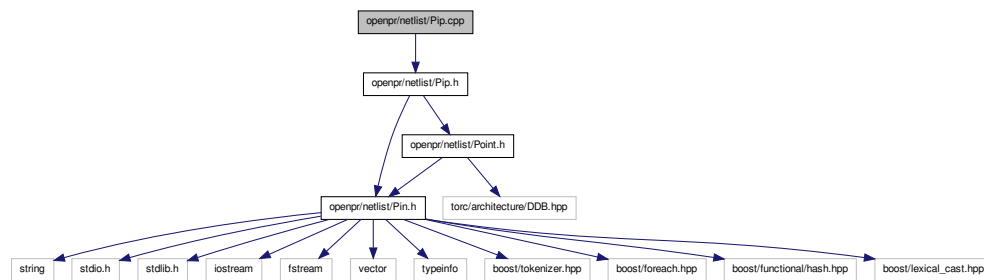
Typedefs

- typedef `boost::tokenizer< boost::char_separator< char > > openpr::netlist::tokenizer`

9.43 openpr/netlist/Pip.cpp File Reference

```
#include "openpr/netlist/Pip.h"
```

Include dependency graph for Pip.cpp:



Namespaces

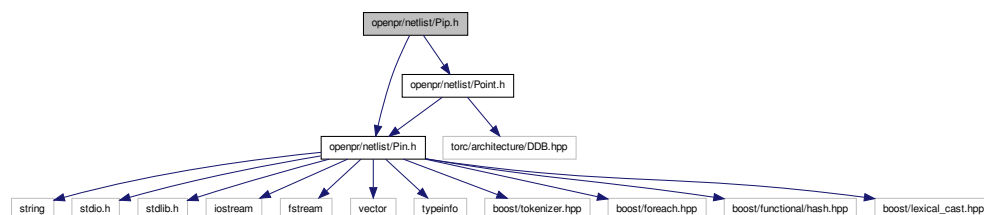
- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.44 openpr/netlist/Pip.h File Reference

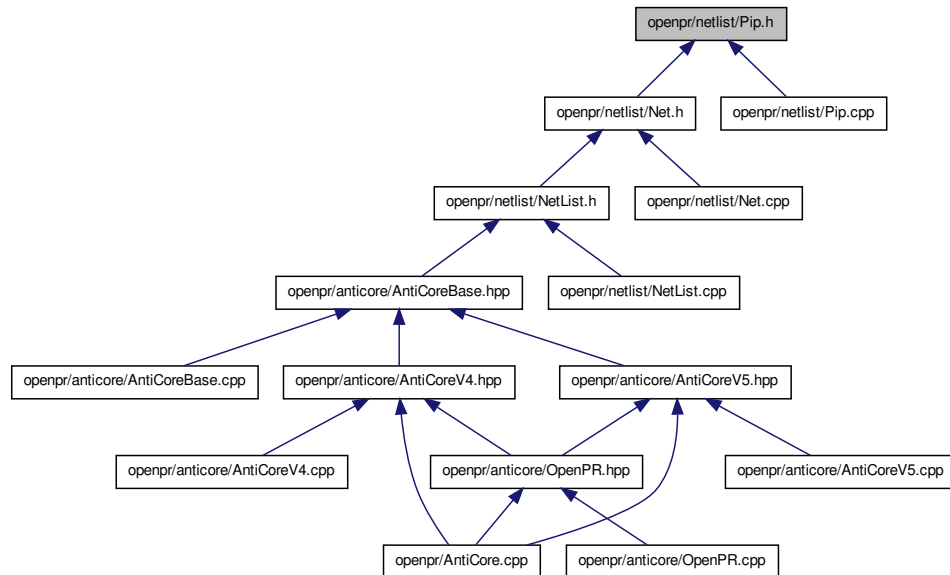
```
#include "openpr/netlist/Pin.h"
```

```
#include "openpr/netlist/Point.h"
```

Include dependency graph for Pip.h:



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



Classes

- class [openpr::netlist::Pip](#)

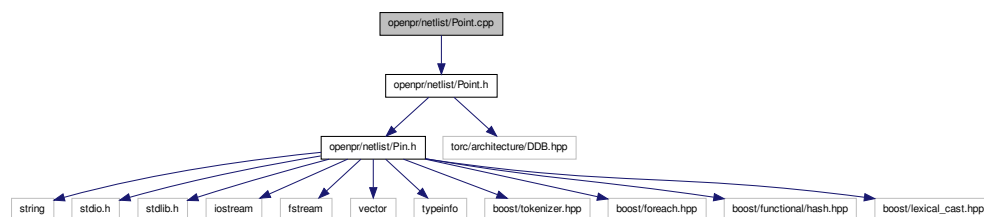
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

9.45 openpr/netlist/Point.cpp File Reference

```
#include "openpr/netlist/Point.h"
```

Include dependency graph for Point.cpp:



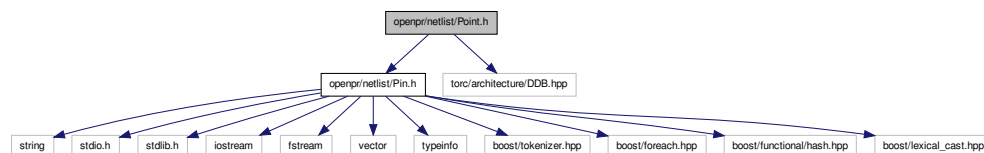
Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

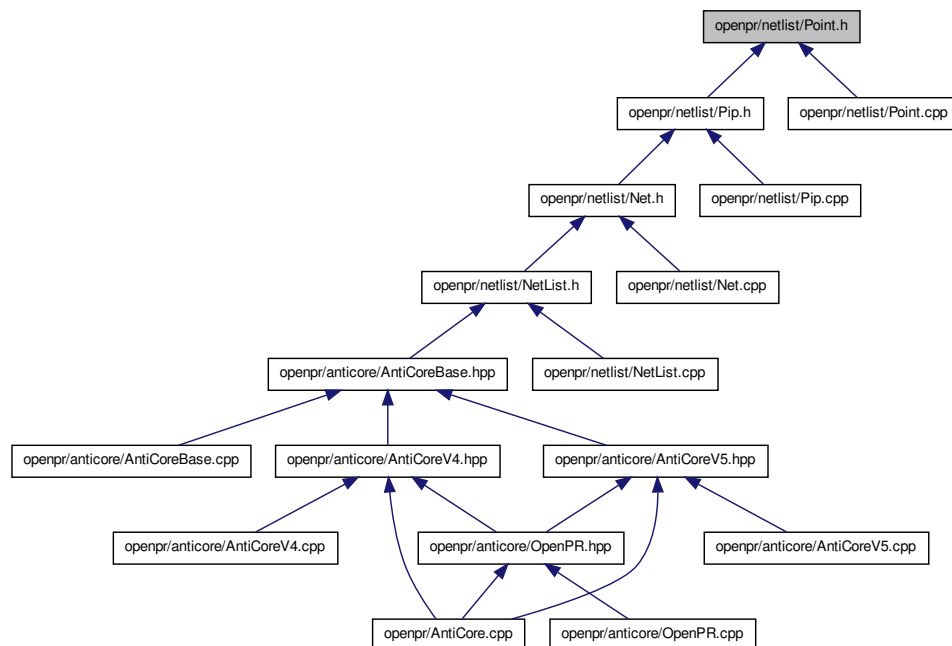
9.46 openpr/netlist/Point.h File Reference

```
#include "openpr/netlist/Pin.h"
#include "torc/architecture/DDB.hpp"
```

Include dependency graph for Point.h:



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



Classes

- class [openpr::netlist::Point](#)

Namespaces

- namespace [openpr](#)
- namespace [openpr::netlist](#)

Index

- ~AntiCoreBase
 - openpr::AntiCoreBase, [20](#)
- ~InPin
 - openpr::netlist::InPin, [68](#)
- ~Net
 - openpr::netlist::Net, [72](#)
- ~NetList
 - openpr::netlist::NetList, [78](#)
- ~OutPin
 - openpr::netlist::OutPin, [96](#)
- ~Pin
 - openpr::netlist::Pin, [99](#)
- ~Pip
 - openpr::netlist::Pip, [103](#)
- ~Point
 - openpr::netlist::Point, [109](#)
- ~bitstream
 - openpr::bitstream::bitstream, [43](#)
- ~device
 - openpr::bitstream::device, [55](#)
- ~openPR
 - openpr::openPR, [83](#)
- ~virtex4
 - openpr::bitstream::virtex4, [138](#)
- ~virtex5
 - openpr::bitstream::virtex5, [142](#)
- addCfg
 - openpr::netlist::Net, [72](#)
- addInPin
 - openpr::netlist::Net, [72](#)
- addOutPin
 - openpr::netlist::Net, [72](#)
- allocateMask
 - openpr::AntiCoreBase, [20](#)
- anticore
 - openpr::openPR, [88](#)
- AntiCore.cpp
 - main, [162](#)
- AntiCoreBase
 - openpr::AntiCoreBase, [19](#)
- AntiCoreV4
 - openpr::AntiCoreV4, [34](#)
- AntiCoreV5
 - openpr::AntiCoreV5, [37](#)
- bitstream
 - openpr::bitstream::bitstream, [42](#)
- bitstreamFile
 - openpr::bitstream::bitstream, [51](#)
- bitstreamLength
 - openpr::bitstream::bitstream, [51](#)
- bitstreamWordCount
 - openpr::bitstream::bitstream, [51](#)
- block_type
 - openpr::bitstream::device, [60](#)
- blockedXdlPath
 - openpr::openPR, [88](#)
- blockingNet
 - openpr::AntiCoreBase, [30](#)
- blockRoutes
 - openpr::AntiCoreBase, [20](#), [21](#)
- blockSites
 - openpr::AntiCoreBase, [21](#)
- blockTileRoutes
 - openpr::AntiCoreBase, [22](#)
- blockTileRoutesPartial
 - openpr::AntiCoreBase, [22](#)
- bmNameToTypeMap
 - openpr, [14](#)
- boost::serialization::access
 - openpr::openPR, [88](#)
- BRAM
 - openpr::bitstream, [15](#)
- BRAM_INT
 - openpr::bitstream, [15](#)
- bramcoord_to_major
 - openpr::bitstream::device, [55](#)
- build_xdl_layout
 - openpr::bitstream::device, [56](#)
- buildBlockingNet
 - openpr::openPR, [84](#)
- buildGCLKItoaMap
 - openpr::bitstream::bitstream, [43](#)
- buildItoaMap
 - openpr::bitstream::bitstream, [44](#)
- buildPartial
 - openpr::bitstream::bitstream, [44](#)
- buildPath
 - openpr::openPR, [89](#)
- buildRelativePaths

- openpr::openPR, 84
- buildSiteMap
 - openpr::AntiCoreBase, 23
- buildValidBoundaries
 - openpr::AntiCoreBase, 23
- buildXDLName
 - openpr::bitstream::bitstream, 45
- busMacroMap
 - openpr::AntiCoreBase, 30
- busMacroNames
 - openpr::openPR, 89
- busMacroPath
 - openpr::openPR, 89
- busMacroPrefix
 - openpr::openPR, 89
- busWidth
 - openpr::openPR, 89
- byte_off
 - openpr::bitstream::tile_data, 118
- cArchitectureNameConst
 - openpr, 14
- cEdaNameConst
 - openpr, 14
- cfg
 - openpr::netlist::Net, 75
- cfgMemoryStart
 - openpr::bitstream::bitstream, 51
- changedFrames
 - openpr::bitstream::bitstream, 51
- checkEfficacy
 - openpr::AntiCoreBase, 23
- chip_height
 - openpr::bitstream::device, 56
- chip_width
 - openpr::bitstream::device, 56
- CLB
 - openpr::bitstream, 15
- clb_slices
 - openpr::bitstream::device, 60
- clearInPins
 - openpr::netlist::Net, 73
- clkNetNames
 - openpr::openPR, 89
- col
 - openpr::bitstream::frame_addr, 65
- configure
 - openpr::netlist::Net, 75
- coord
 - openpr::bitstream::tile_data, 118
- currentMode
 - openpr::AntiCoreBase, 31
- cXilinxNameConst
 - openpr, 14
- databasePath
 - openpr::OpenPRTree, 93
- db
 - openpr::openPR, 89
- designDate
 - openpr::bitstream::bitstream, 51
- designName
 - openpr::bitstream::bitstream, 51
 - openpr::openPR, 89
- designTime
 - openpr::bitstream::bitstream, 51
- destination
 - openpr::netlist::Pip, 107
- device
 - openpr::bitstream::architecture, 40
 - openpr::bitstream::device, 55
- deviceName
 - openpr::bitstream::bitstream, 51
 - openpr::openPR, 89
- DSP48
 - openpr::bitstream, 15
- dumpMask
 - openpr::AntiCoreBase, 24
- dynamicAGName
 - openpr::openPR, 89
- eBusWidth
 - openpr::bitstream::v5_bitstream, 131
- eBusWidthWord
 - openpr::bitstream::v5_bitstream, 131
- eCmdAGHIGH
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdCount
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdDESYNC
 - openpr::bitstream::v4_bitstream, 121
- eCmdDESYNCH
 - openpr::bitstream::v5_bitstream, 129
- eCmdDGHIGH
 - openpr::bitstream::v5_bitstream, 129
- eCmdGCAPTURE
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdGRESTORE
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdIPLPG
 - openpr::bitstream::v5_bitstream, 129
- eCmdLFRM
 - openpr::bitstream::v4_bitstream, 120
 - openpr::bitstream::v5_bitstream, 129
- eCmdLTIMER

- openpr::bitstream::v5_bitstream, 129
- eCmdMFW
 - openpr::bitstream::v5_bitstream, 129
- eCmdMFWR
 - openpr::bitstream::v4_bitstream, 120
- eCmdNULL
 - openpr::bitstream::v4_bitstream, 120
 - openpr::bitstream::v5_bitstream, 129
- eCmdRCAP
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdRCFG
 - openpr::bitstream::v4_bitstream, 120
 - openpr::bitstream::v5_bitstream, 129
- eCmdRCRC
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdSHUTDOWN
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdSTART
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdSWITCH
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eCmdWCFG
 - openpr::bitstream::v4_bitstream, 120
 - openpr::bitstream::v5_bitstream, 129
- ECommand
 - openpr::bitstream::v4_bitstream, 120
 - openpr::bitstream::v5_bitstream, 129
- edaPath
 - openpr::OpenPRTree, 93
- eDummyWord
 - openpr::bitstream::v4_bitstream, 123
 - openpr::bitstream::v5_bitstream, 131
- eMaskPacketOpcode
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eMaskPacketType
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- EMasks
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eMaskType1Address
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eMaskType1Count
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 129
- eMaskType1Reserved
 - openpr::bitstream::v4_bitstream, 121
- openpr::bitstream::v5_bitstream, 129
- eMaskType2Count
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- eMode
 - openpr, 14
- endTile
 - openpr::AntiCoreBase, 31
- EOpcode
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- EOpcodeCount
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- eOpNOP
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- eOpRead
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- eOpReserved
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- eOpWrite
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- EPacketType
 - openpr::bitstream::v4_bitstream, 121
 - openpr::bitstream::v5_bitstream, 130
- ePartial
 - openpr, 14
- eRegAXSS
 - openpr::bitstream::v4_bitstream, 122
 - openpr::bitstream::v5_bitstream, 130
- eRegBOOTSTS
 - openpr::bitstream::v5_bitstream, 131
- eRegCBC
 - openpr::bitstream::v4_bitstream, 122
 - openpr::bitstream::v5_bitstream, 130
- eRegCMD
 - openpr::bitstream::v4_bitstream, 122
 - openpr::bitstream::v5_bitstream, 130
- eRegCOR
 - openpr::bitstream::v4_bitstream, 122
- eRegCOR0
 - openpr::bitstream::v5_bitstream, 130
- eRegCOR1
 - openpr::bitstream::v5_bitstream, 130
- eRegCount
 - openpr::bitstream::v4_bitstream, 122
 - openpr::bitstream::v5_bitstream, 131
- eRegCRC
 - openpr::bitstream::v4_bitstream, 122
 - openpr::bitstream::v5_bitstream, 130

- eRegCSOB
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegCTL
 - openpr::bitstream::v4_bitstream, [122](#)
- eRegCTL0
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegCTL1
 - openpr::bitstream::v5_bitstream, [131](#)
- eRegFAR
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegFDRI
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegFDRO
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegIDCODE
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- ERegister
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegLOUT
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegMASK
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegMFWR
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegSTAT
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eRegTIMER
 - openpr::bitstream::v5_bitstream, [131](#)
- eRegWBSTAR
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftBlockType
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- EShiftFAR
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftMajor
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftMNA
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftPacketOpcode
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftPacketType
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftRow
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- EShifts
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftTB
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftType1Address
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftType1Count
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftType1Reserved
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eShiftType2Count
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eStatic
 - openpr, [14](#)
- eSyncWord
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- eType1
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eType2
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- eTypeCount
 - openpr::bitstream::v4_bitstream, [122](#)
 - openpr::bitstream::v5_bitstream, [130](#)
- EWords
 - openpr::bitstream::v4_bitstream, [123](#)
 - openpr::bitstream::v5_bitstream, [131](#)
- executablePath
 - openpr::OpenPRTree, [93](#)
- expandRegion
 - openpr::AntiCoreBase, [24](#)
- expandRegionToINT
 - openpr::AntiCoreBase, [24](#)
- expect
 - openpr::bitstream::bitstream, [45](#)
- exportPipFromArc
 - openpr::AntiCoreBase, [24](#)
- far
 - openpr::bitstream::tile_data, [118](#)

- farToStruct
 - openpr::bitstream::bitstream, 46
 - openpr::bitstream::v4_bitstream, 123
 - openpr::bitstream::v5_bitstream, 132
- findIndexToFAddr
 - openpr::bitstream::bitstream, 51
- findNet
 - openpr::netlist::NetList, 78
- findPin
 - openpr::netlist::NetList, 78
- findPip
 - openpr::netlist::NetList, 79
- first_frame
 - openpr::bitstream::tile_data, 118
- frame_addr
 - openpr::bitstream::frame_addr, 64
- frame_array
 - openpr::bitstream::bitstream, 52
- frame_height
 - openpr::bitstream::architecture, 40
 - openpr::bitstream::device, 60
- frame_num
 - openpr::bitstream::tile_data, 118
- frame_offset
 - openpr::bitstream::device, 56
- frame_words
 - openpr::bitstream::architecture, 40
 - openpr::bitstream::device, 60
- frameBitmap
 - openpr::bitstream::bitstream, 52
- frameECC
 - openpr::bitstream::bitstream, 52
- fullBsPath
 - openpr::openPR, 89
- fullUcfPath
 - openpr::openPR, 90
- fullXdlPath
 - openpr::openPR, 90
- GCLK
 - openpr::bitstream, 15
- gclk_index
 - openpr::bitstream::device, 60
- generateFullStream
 - openpr::netlist::Pip, 104
- generateLocation
 - openpr::netlist::Pip, 104
- genLockConstraints
 - openpr::openPR, 84
- genMacroPlacement
 - openpr::AntiCoreBase, 24
 - openpr::AntiCoreV4, 35
 - openpr::AntiCoreV5, 38
- genPartialBitstream
 - openpr::openPR, 84
- genPassThroughScripts
 - openpr::openPR, 85
- genPlaceConstraints
 - openpr::AntiCoreBase, 24
 - openpr::openPR, 85
- genProhibitConstraints
 - openpr::AntiCoreBase, 25
- get_addressable_blk_types
 - openpr::bitstream::device, 56
 - openpr::bitstream::virtex4, 139
 - openpr::bitstream::virtex5, 142
- get_blk_type
 - openpr::bitstream::device, 56, 57
- get_cfg_size
 - openpr::bitstream::device, 57
- get_chip_id
 - openpr::bitstream::device, 57
- get_frame_words
 - openpr::bitstream::device, 57
- get_gclk_index
 - openpr::bitstream::device, 57
- get_name
 - openpr::bitstream::device, 58
- get_num_rows
 - openpr::bitstream::device, 58
- get_row_height
 - openpr::bitstream::device, 58
- get_row_width
 - openpr::bitstream::device, 58
- get_tile_frames
 - openpr::bitstream::device, 58
- get_tile_type
 - openpr::bitstream::device, 59
- getDestination
 - openpr::netlist::Pip, 104
- getNetList
 - openpr::netlist::NetList, 79
- getPipToNet
 - openpr::netlist::NetList, 79
- getRegionTiles
 - openpr::AntiCoreBase, 25
- getRegionVertices
 - openpr::AntiCoreBase, 25
- getSegmentIndex
 - openpr::netlist::Point, 110
- getSinkStr
 - openpr::netlist::Pip, 104
- getSiteType
 - openpr::AntiCoreBase, 26
- getSource
 - openpr::netlist::Pip, 104
 - openpr::netlist::Point, 110
- getSourceDestination

- openpr::netlist::Point, 110
- getSourceStr
 - openpr::netlist::Pip, 104
- getTileStr
 - openpr::netlist::Pip, 104
- hash_value
 - openpr::bitstream::tile_coord, 116
 - openpr::netlist, 16
 - openpr::netlist::Net, 73
 - openpr::netlist::Pin, 100
- id
 - openpr::bitstream::device, 60
- importXDL
 - openpr::AntiCoreBase, 26
- InPin
 - openpr::netlist::InPin, 68
- inpin
 - openpr::netlist::InPin, 70
- inPins
 - openpr::netlist::Net, 75
- inRegion
 - openpr::AntiCoreBase, 26
- insertNet
 - openpr::netlist::NetList, 79
- insertPip
 - openpr::netlist::Net, 73
 - openpr::netlist::NetList, 79
- INVALID
 - openpr::bitstream, 15
- IOB
 - openpr::bitstream, 15
- isPartial
 - openpr::bitstream::bitstream, 52
 - openpr::openPR, 90
- l_xMax
 - openpr::openPR, 90
- l_xMin
 - openpr::openPR, 90
- l_yMax
 - openpr::openPR, 90
- l_yMin
 - openpr::openPR, 90
- loadFile
 - openpr::bitstream::bitstream, 46
- location
 - openpr::netlist::InPin, 70
 - openpr::netlist::OutPin, 98
 - openpr::netlist::Pip, 107
- logic_table
 - openpr::bitstream::device, 60
- logPath
 - openpr::OpenPRTree, 93
- macroTiles
 - openpr::AntiCoreBase, 31
- macroWidth
 - openpr::AntiCoreBase, 31
- main
 - AntiCore.cpp, 162
- mapBitstream
 - openpr::bitstream::bitstream, 46
- mapBRAM
 - openpr::bitstream::bitstream, 47
- mask
 - openpr::AntiCoreBase, 31
- maxSite
 - openpr::prohibitRange, 114
- mDB
 - openpr::AntiCoreBase, 31
 - openpr::netlist::NetList, 81
 - openpr::netlist::Pip, 107
- mergeClockTree
 - openpr::AntiCoreBase, 26
 - openpr::openPR, 86
- mergedXdlPath
 - openpr::openPR, 90
- mergePips
 - openpr::netlist::Net, 73
- mFrameData
 - openpr::bitstream::bitstream, 52
- minSite
 - openpr::prohibitRange, 114
- mna
 - openpr::bitstream::frame_addr, 65
- mRegister
 - openpr::bitstream::v4_bitstream, 126
 - openpr::bitstream::v5_bitstream, 136
- mSegments
 - openpr::AntiCoreBase, 31
- mTiles
 - openpr::AntiCoreBase, 31
- MULTIPLE
 - openpr::bitstream, 15
- my_dev
 - openpr::bitstream::bitstream, 52
- name
 - openpr::bitstream::device, 60
 - openpr::bitstream::tile_data, 118
 - openpr::netlist::Net, 75
- Net
 - openpr::netlist::Net, 72
- NetList
 - openpr::netlist::NetList, 77
- netList

- openpr::netlist::NetList, 81
- netParser
 - openpr::netlist::NetList, 79
- netToPip
 - openpr::netlist::NetList, 81
- num_blk_types
 - openpr::bitstream::virtex4, 138
- NUM_TILE_TYPES
 - openpr::bitstream, 15
- num_blk_types
 - openpr::bitstream::device, 60
- num_cols
 - openpr::bitstream::device, 61
- num_frames
 - openpr::bitstream::bitstream, 52
 - openpr::bitstream::tile_data, 118
- num_rows
 - openpr::bitstream::device, 61
- openPR
 - openpr::openPR, 83
- openpr, 13
 - bmNameToTypeMap, 14
 - cArchitectureNameConst, 14
 - cEdaNameConst, 14
 - cXilinxNameConst, 14
 - eMode, 14
 - ePartial, 14
 - eStatic, 14
 - string, 14
- openpr/AntiCore.cpp, 161
- openpr/anticore/AntiCoreBase.cpp, 162
- openpr/anticore/AntiCoreBase.hpp, 163
- openpr/anticore/AntiCoreV4.cpp, 165
- openpr/anticore/AntiCoreV4.hpp, 165
- openpr/anticore/AntiCoreV5.cpp, 166
- openpr/anticore/AntiCoreV5.hpp, 167
- openpr/anticore/OpenPR.cpp, 167
- openpr/anticore/OpenPR.hpp, 168
- openpr/anticore/OpenPRTree.cpp, 169
- openpr/anticore/OpenPRTree.hpp, 170
- openpr/anticore/ProhibitRange.hpp, 171
- openpr/bitstream/architecture.h, 172
- openpr/bitstream/bitstream.cpp, 173
- openpr/bitstream/bitstream.h, 174
- openpr/bitstream/device.cpp, 175
- openpr/bitstream/device.h, 176
- openpr/bitstream/tile.h, 178
- openpr/bitstream/v4_bitstream.cpp, 178
- openpr/bitstream/v4_bitstream.h, 179
- openpr/bitstream/v4_devices.cpp, 180
- openpr/bitstream/v4_devices.h, 181
- openpr/bitstream/v5_bitstream.cpp, 183
- openpr/bitstream/v5_bitstream.h, 183
- openpr/bitstream/v5_devices.cpp, 184
- openpr/bitstream/v5_devices.h, 185
- openpr/bitstream/virtex4.cpp, 187
- openpr/bitstream/virtex4.h, 187
- openpr/bitstream/virtex5.cpp, 189
- openpr/bitstream/virtex5.h, 189
- openpr/netlist/HashStructs.h, 191
- openpr/netlist/InPin.cpp, 192
- openpr/netlist/InPin.h, 192
- openpr/netlist/Net.cpp, 193
- openpr/netlist/Net.h, 194
- openpr/netlist/NetHashStruct.h, 196
- openpr/netlist/NetList.cpp, 196
- openpr/netlist/NetList.h, 197
- openpr/netlist/OutPin.cpp, 198
- openpr/netlist/OutPin.h, 199
- openpr/netlist/Pin.cpp, 200
- openpr/netlist/Pin.h, 200
- openpr/netlist/Pip.cpp, 202
- openpr/netlist/Pip.h, 202
- openpr/netlist/Point.cpp, 203
- openpr/netlist/Point.h, 204
- openpr::AntiCoreBase, 17
 - ~AntiCoreBase, 20
 - allocateMask, 20
 - AntiCoreBase, 19
 - blockingNet, 30
 - blockRoutes, 20, 21
 - blockSites, 21
 - blockTileRoutes, 22
 - blockTileRoutesPartial, 22
 - buildSiteMap, 23
 - buildValidBoundaries, 23
 - busMacroMap, 30
 - checkEfficacy, 23
 - currentMode, 31
 - dumpMask, 24
 - endTile, 31
 - expandRegion, 24
 - expandRegionToINT, 24
 - exportPipFromArc, 24
 - genMacroPlacement, 24
 - genPlaceConstraints, 24
 - genProhibitConstraints, 25
 - getRegionTiles, 25
 - getRegionVertices, 25
 - getSiteType, 26
 - importXDL, 26
 - inRegion, 26
 - macroTiles, 31
 - macroWidth, 31
 - mask, 31
 - mDB, 31
 - mergeClockTree, 26

- mSegments, 31
- mTiles, 31
- placedXDLInput, 32
- placeMacro, 27
- prohibitedSites, 32
- retrieveDynamicRegion, 27
- setMode, 28
- setRegionVertices, 28
- setupRouteBlocker, 28
- shrinkRegion, 29
- sinks_buf, 32
- siteMap, 32
- siteNameToTileIndex, 29
- sources_buf, 32
- startTile, 32
- tileToSiteMap, 19
- updateRegion, 29
- updateRegionExpand, 30
- validateRegion, 30
- validBoundaries, 32
- wires_buf, 32
- xMax, 32
- xMin, 33
- yMax, 33
- yMin, 33
- openpr::AntiCoreV4, 33
 - AntiCoreV4, 34
 - genMacroPlacement, 35
 - placeMacro, 35
 - tilesPerRegion, 36
- openpr::AntiCoreV5, 36
 - AntiCoreV5, 37
 - genMacroPlacement, 38
 - placeMacro, 38
 - tilesPerRegion, 39
- openpr::bitstream, 14
 - BRAM, 15
 - BRAM_INT, 15
 - CLB, 15
 - DSP48, 15
 - GCLK, 15
 - INVALID, 15
 - IOB, 15
 - MULTIPLE, 15
 - NUM_TILE_TYPES, 15
 - PAD, 15
 - tile_types, 15
 - TRANSCV, 15
- openpr::bitstream::architecture, 39
 - device, 40
 - frame_height, 40
 - frame_words, 40
 - tile_frames, 40
 - virtex4, 40
 - virtex5, 40
- openpr::bitstream::bitstream, 40
 - ~bitstream, 43
 - bitstream, 42
 - bitstreamFile, 51
 - bitstreamLength, 51
 - bitstreamWordCount, 51
 - buildGCLKItoaMap, 43
 - buildItoaMap, 44
 - buildPartial, 44
 - buildXDLName, 45
 - cfgMemoryStart, 51
 - changedFrames, 51
 - designDate, 51
 - designName, 51
 - designTime, 51
 - deviceName, 51
 - expect, 45
 - farToStruct, 46
 - findexToFaddr, 51
 - frame_array, 52
 - frameBitmap, 52
 - frameECC, 52
 - isPartial, 52
 - loadFile, 46
 - mapBitstream, 46
 - mapBRAM, 47
 - mFrameData, 52
 - my_dev, 52
 - num_frames, 52
 - readHeader, 48
 - readPackets, 48
 - readXilinxString, 48
 - structToFar, 48
 - tile_map, 52
 - tileMap, 52
 - write, 49
 - writeBitstream, 49
 - writeFrames, 49
 - writeHeader, 50
 - writePackets, 50
 - writePacketsPartial, 50
 - writeXilinxString, 50
- openpr::bitstream::device, 53
 - ~device, 55
 - block_type, 60
 - bramcoord_to_major, 55
 - build_xdl_layout, 56
 - chip_height, 56
 - chip_width, 56
 - clb_slices, 60
 - device, 55
 - frame_height, 60
 - frame_offset, 56

- frame_words, 60
- gclk_index, 60
- get_addressable_blk_types, 56
- get_blk_type, 56, 57
- get_cfg_size, 57
- get_chip_id, 57
- get_frame_words, 57
- get_gclk_index, 57
- get_name, 58
- get_num_rows, 58
- get_row_height, 58
- get_row_width, 58
- get_tile_frames, 58
- get_tile_type, 59
- id, 60
- logic_table, 60
- name, 60
- num_blk_types, 60
- num_cols, 61
- num_rows, 61
- routing_table, 61
- row_layout, 61
- row_width, 61
- tile_frames, 61
- tile_offset, 59
- tile_width, 61
- tilecoord_to_major, 59
- xdl_layout, 61
- openpr::bitstream::frame_addr, 63
 - col, 65
 - frame_addr, 64
 - mna, 65
 - row, 65
 - str, 64
 - tb, 65
 - type, 65
- openpr::bitstream::tile_coord, 115
 - hash_value, 116
 - operator==, 115
 - set, 115
 - tile_coord, 115
 - x, 116
 - y, 116
- openpr::bitstream::tile_data, 116
 - byte_off, 118
 - coord, 118
 - far, 118
 - first_frame, 118
 - frame_num, 118
 - name, 118
 - num_frames, 118
 - print, 117
 - tile_data, 117
- openpr::bitstream::v4_bitstream
 - eCmdAGHIGH, 121
 - eCmdCount, 121
 - eCmdDESYNC, 121
 - eCmdGCAPTURE, 121
 - eCmdGRESTORE, 121
 - eCmdLFRM, 120
 - eCmdMFWR, 120
 - eCmdNULL, 120
 - eCmdRCAP, 121
 - eCmdRCFG, 120
 - eCmdRCRC, 121
 - eCmdSHUTDOWN, 121
 - eCmdSTART, 121
 - eCmdSWITCH, 121
 - eCmdWCFG, 120
 - eDummyWord, 123
 - eMaskPacketOpcode, 121
 - eMaskPacketType, 121
 - eMaskType1Address, 121
 - eMaskType1Count, 121
 - eMaskType1Reserved, 121
 - eMaskType2Count, 121
 - eOpcodeCount, 121
 - eOpNOP, 121
 - eOpRead, 121
 - eOpReserved, 121
 - eOpWrite, 121
 - eRegAXSS, 122
 - eRegCBC, 122
 - eRegCMD, 122
 - eRegCOR, 122
 - eRegCount, 122
 - eRegCRC, 122
 - eRegCTL, 122
 - eRegFAR, 122
 - eRegFDRI, 122
 - eRegFDRO, 122
 - eRegIDCODE, 122
 - eRegLOUT, 122
 - eRegMASK, 122
 - eRegMFWR, 122
 - eRegSTAT, 122
 - eShiftBlockType, 122
 - eShiftMajor, 123
 - eShiftMNA, 123
 - eShiftPacketOpcode, 123
 - eShiftPacketType, 123
 - eShiftRow, 122
 - eShiftTB, 122
 - eShiftType1Address, 123
 - eShiftType1Count, 123
 - eShiftType1Reserved, 123
 - eShiftType2Count, 123
 - eSyncWord, 123

- eType1, [122](#)
- eType2, [122](#)
- eTypeCount, [122](#)
- openpr::bitstream::v4_bitstream, [118](#)
 - ECommand, [120](#)
 - EMasks, [121](#)
 - EOpcode, [121](#)
 - EPacketType, [121](#)
 - ERegister, [122](#)
 - EShiftFAR, [122](#)
 - EShifts, [123](#)
 - EWords, [123](#)
 - farToStruct, [123](#)
 - mRegister, [126](#)
 - readPackets, [124](#)
 - reverseFrameBits, [124](#)
 - sCommandName, [126](#)
 - sOpcodeName, [126](#)
 - sRegisterName, [126](#)
 - structToFar, [124](#)
 - sTypeName, [126](#)
 - top, [127](#)
 - unmangleTilePair, [124](#)
 - v4_bitstream, [123](#)
 - writeFrameData, [125](#)
 - writePacketHeader, [125](#)
 - writePackets, [125](#)
 - writePacketsPartial, [125](#)
 - writePartialFrames, [125](#)
- openpr::bitstream::v5_bitstream
 - eBusWidth, [131](#)
 - eBusWidthWord, [131](#)
 - eCmdAGHIGH, [129](#)
 - eCmdCount, [129](#)
 - eCmdDESYNCH, [129](#)
 - eCmdDGHIGH, [129](#)
 - eCmdGCAPTURE, [129](#)
 - eCmdGRESTORE, [129](#)
 - eCmdIPROG, [129](#)
 - eCmdLFRM, [129](#)
 - eCmdLTIMER, [129](#)
 - eCmdMFW, [129](#)
 - eCmdNULL, [129](#)
 - eCmdRCAP, [129](#)
 - eCmdRCFG, [129](#)
 - eCmdRCRC, [129](#)
 - eCmdSHUTDOWN, [129](#)
 - eCmdSTART, [129](#)
 - eCmdSWITCH, [129](#)
 - eCmdWCFG, [129](#)
 - eDummyWord, [131](#)
 - eMaskPacketOpcode, [129](#)
 - eMaskPacketType, [129](#)
 - eMaskType1Address, [129](#)
 - eMaskType1Count, [129](#)
 - eMaskType1Reserved, [129](#)
 - eMaskType2Count, [130](#)
 - eOpcodeCount, [130](#)
 - eOpNOP, [130](#)
 - eOpRead, [130](#)
 - eOpReserved, [130](#)
 - eOpWrite, [130](#)
 - eRegAXSS, [130](#)
 - eRegBOOTSTS, [131](#)
 - eRegCBC, [130](#)
 - eRegCMD, [130](#)
 - eRegCOR0, [130](#)
 - eRegCOR1, [130](#)
 - eRegCount, [131](#)
 - eRegCRC, [130](#)
 - eRegCSOB, [130](#)
 - eRegCTL0, [130](#)
 - eRegCTL1, [131](#)
 - eRegFAR, [130](#)
 - eRegFDRI, [130](#)
 - eRegFDRO, [130](#)
 - eRegIDCODE, [130](#)
 - eRegLOUT, [130](#)
 - eRegMASK, [130](#)
 - eRegMFW, [130](#)
 - eRegSTAT, [130](#)
 - eRegTIMER, [131](#)
 - eRegWBSTAR, [131](#)
 - eShiftBlockType, [131](#)
 - eShiftMajor, [131](#)
 - eShiftMNA, [131](#)
 - eShiftPacketOpcode, [131](#)
 - eShiftPacketType, [131](#)
 - eShiftRow, [131](#)
 - eShiftTB, [131](#)
 - eShiftType1Address, [131](#)
 - eShiftType1Count, [131](#)
 - eShiftType1Reserved, [131](#)
 - eShiftType2Count, [131](#)
 - eSyncWord, [131](#)
 - eType1, [130](#)
 - eType2, [130](#)
 - eTypeCount, [130](#)
- openpr::bitstream::v5_bitstream, [127](#)
 - ECommand, [129](#)
 - EMasks, [129](#)
 - EOpcode, [130](#)
 - EPacketType, [130](#)
 - ERegister, [130](#)
 - EShiftFAR, [131](#)
 - EShifts, [131](#)
 - EWords, [131](#)
 - farToStruct, [132](#)

- mRegister, 136
- readPackets, 132
- sCommandName, 136
- sOpcodeName, 136
- sRegisterName, 136
- structToFar, 133
- sTypeName, 136
- v5_bitstream, 132
- writeFrameData, 133
- writePacketHeader, 133
- writePackets, 134
- writePacketsPartial, 134
- writePartialFrames, 135
- openpr::bitstream::virtex4, 137
 - ~virtex4, 138
 - get_addressable_blk_types, 139
 - num_blk_types, 138
 - tile_offset, 139
 - virtex4, 138
 - virtex4_clb_slices, 138
 - virtex4_frame_height, 138
 - virtex4_frame_words, 138
 - virtex4_block_type, 139
 - virtex4_logic_table, 139
 - virtex4_routing_table, 139
 - virtex4_tile_frames, 140
- openpr::bitstream::virtex5, 140
 - ~virtex5, 142
 - get_addressable_blk_types, 142
 - tile_offset, 142
 - virtex5, 142
 - virtex5_clb_slices, 142
 - virtex5_frame_height, 142
 - virtex5_frame_words, 142
 - virtex5_num_blk_types, 142
 - virtex5_block_type, 143
 - virtex5_logic_table, 143
 - virtex5_routing_table, 143
 - virtex5_tile_frames, 143
- openpr::bitstream::xc4vfx60, 144
 - xc4vfx60, 145
 - xc4vfx60_num_cols, 145
 - xc4vfx60_num_rows, 145
 - xc4vfx60_id, 146
 - xc4vfx60_name, 146
 - xc4vfx60_row_layout, 146
- openpr::bitstream::xc4vlx15, 146
 - xc4vlx15, 148
 - xc4vlx15_num_cols, 148
 - xc4vlx15_num_rows, 148
 - xc4vlx15_id, 148
 - xc4vlx15_name, 148
 - xc4vlx15_row_layout, 148
- openpr::bitstream::xc4vlx60, 149
 - xc4vlx60, 150
 - xc4vlx60_num_cols, 150
 - xc4vlx60_num_rows, 150
 - xc4vlx60_id, 150
 - xc4vlx60_name, 150
 - xc4vlx60_row_layout, 150
- openpr::bitstream::xc5vlx110t, 151
 - xc5vlx110t, 152
 - xc5vlx110t_num_cols, 152
 - xc5vlx110t_num_rows, 152
 - xc5vlx110t_id, 153
 - xc5vlx110t_name, 153
 - xc5vlx110t_row_layout, 153
- openpr::bitstream::xc5vlx50, 153
 - xc5vlx50, 155
 - xc5vlx50_num_cols, 155
 - xc5vlx50_num_rows, 155
 - xc5vlx50_id, 155
 - xc5vlx50_name, 155
 - xc5vlx50_row_layout, 155
- openpr::bitstream::xc5vlx50t, 156
 - xc5vlx50t, 157
 - xc5vlx50t_num_cols, 157
 - xc5vlx50t_num_rows, 157
 - xc5vlx50t_id, 157
 - xc5vlx50t_name, 157
 - xc5vlx50t_row_layout, 157
- openpr::bitstream::xc5vsx95t, 158
 - xc5vsx95t, 159
 - xc5vsx95t_num_cols, 159
 - xc5vsx95t_num_rows, 159
 - xc5vsx95t_id, 160
 - xc5vsx95t_name, 160
 - xc5vsx95t_row_layout, 160
- openpr::netlist, 16
 - hash_value, 16
 - tokenizer, 16
- openpr::netlist::eq_net, 61
 - operator(), 62
- openpr::netlist::eq_pip, 62
 - operator(), 62
- openpr::netlist::eq_point, 62
 - operator(), 63
- openpr::netlist::eq_segment, 63
 - operator(), 63
- openpr::netlist::hash_net, 65
 - operator(), 65
- openpr::netlist::hash_pip, 66
 - operator(), 66
- openpr::netlist::hash_point, 66
 - operator(), 66
- openpr::netlist::hash_segment, 66
 - operator(), 67
- openpr::netlist::InPin, 67

- ~InPin, 68
- InPin, 68
- inpin, 70
- location, 70
- operator<, 69
- operator>, 69
- operator(), 68, 69
- operator=, 69
- operator==, 69
- printData, 70
- openpr::netlist::Net, 70
 - ~Net, 72
 - addCfg, 72
 - addInPin, 72
 - addOutPin, 72
 - cfg, 75
 - clearInPins, 73
 - configure, 75
 - hash_value, 73
 - inPins, 75
 - insertPip, 73
 - mergePips, 73
 - name, 75
 - Net, 72
 - operator<, 73
 - operator>, 74
 - operator(), 73
 - operator=, 74
 - operator==, 74
 - outpin, 75
 - outpinLetter, 75
 - outPins, 75
 - pips, 76
 - printData, 74
 - remotePin, 75
- openpr::netlist::NetList, 76
 - ~NetList, 78
 - findNet, 78
 - findPin, 78
 - findPip, 79
 - getNetList, 79
 - getPipToNet, 79
 - insertNet, 79
 - insertPip, 79
 - mDB, 81
 - NetList, 77
 - netList, 81
 - netParser, 79
 - netToPip, 81
 - outputXDL, 81
 - pipToNet, 81
 - pointToPip, 81
 - printData, 79
 - removePip, 80
 - segmentToNet, 81
 - topLevelParser, 80
- openpr::netlist::OutPin, 94
 - ~OutPin, 96
 - location, 98
 - operator<, 96
 - operator>, 97
 - operator(), 96
 - operator=, 96, 97
 - operator==, 97
 - OutPin, 96
 - outpin, 98
 - printData, 97
- openpr::netlist::Pin, 98
 - ~Pin, 99
 - hash_value, 100
 - operator<, 99
 - operator>, 100
 - operator(), 99
 - operator=, 99
 - operator==, 99, 100
 - Pin, 99
 - printData, 100
- openpr::netlist::Pip, 100
 - ~Pip, 103
 - destination, 107
 - generateFullStream, 104
 - generateLocation, 104
 - getDestination, 104
 - getSinkStr, 104
 - getSource, 104
 - getSourceStr, 104
 - getTileStr, 104
 - location, 107
 - mDB, 107
 - operator<, 105
 - operator>, 106
 - operator(), 105
 - operator=, 105, 106
 - operator==, 106
 - parseLocation, 106
 - Pip, 102, 103
 - printData, 106, 107
 - source, 107
 - type, 107
 - wholeData, 107
 - xLoc, 108
 - yLoc, 108
- openpr::netlist::Point, 108
 - ~Point, 109
 - getSegmentIndex, 110
 - getSource, 110
 - getSourceDestination, 110
 - operator<, 111

- operator>, 111
- operator(), 110
- operator==, 111
- Point, 109
- segmentIndex, 112
- setIndices, 112
- source, 112
- sourceDestination, 112
- tile, 112
- openpr::openPR, 81
 - ~openPR, 83
 - anticore, 88
 - blockedXdlPath, 88
 - boost::serialization::access, 88
 - buildBlockingNet, 84
 - buildPath, 89
 - buildRelativePaths, 84
 - busMacroNames, 89
 - busMacroPath, 89
 - busMacroPrefix, 89
 - busWidth, 89
 - clkNetNames, 89
 - db, 89
 - designName, 89
 - deviceName, 89
 - dynamicAGName, 89
 - fullBsPath, 89
 - fullUcfPath, 90
 - fullXdlPath, 90
 - genLockConstraints, 84
 - genPartialBitstream, 84
 - genPassThroughScripts, 85
 - genPlaceConstraints, 85
 - isPartial, 90
 - l_xMax, 90
 - l_xMin, 90
 - l_yMax, 90
 - l_yMin, 90
 - mergeClockTree, 86
 - mergedXdlPath, 90
 - openPR, 83
 - partialBsPath, 90
 - partialPath, 90
 - passThroughNet2, 91
 - passThroughNetName, 91
 - pcfPath, 91
 - placedXdlPath, 91
 - placeMacros, 86
 - projectPath, 91
 - regionDefined, 91
 - routeBlocker, 87
 - routedXdlPath, 91
 - routePTScriptPath, 91
 - serialize, 87
 - setupAntiCore, 87
 - setupDynamicRegion, 87
 - siteBlocker, 88
 - staticPath, 91
 - staticPlacedXdlPath, 91
 - ucfPath, 92
 - unroutePTScriptPath, 92
 - xMax, 92
 - xMin, 92
 - yMax, 92
 - yMin, 92
- openpr::OpenPRTree, 92
 - databasePath, 93
 - edaPath, 93
 - executablePath, 93
 - logPath, 93
 - OpenPRTree, 93
 - relativePath, 93
 - sDatabasePath, 94
 - sEdaPath, 94
 - sExecutablePath, 94
 - sLogPath, 94
 - sRelativePath, 94
 - sWorkingPath, 94
 - workingPath, 94
- openpr::prohibitRange, 112
 - maxSite, 114
 - minSite, 114
 - operator<, 114
 - operator>, 114
 - prohibitRange, 113
- OpenPRTree
 - openpr::OpenPRTree, 93
- operator<
 - openpr::netlist::InPin, 69
 - openpr::netlist::Net, 73
 - openpr::netlist::OutPin, 96
 - openpr::netlist::Pin, 99
 - openpr::netlist::Pip, 105
 - openpr::netlist::Point, 111
 - openpr::prohibitRange, 114
- operator>
 - openpr::netlist::InPin, 69
 - openpr::netlist::Net, 74
 - openpr::netlist::OutPin, 97
 - openpr::netlist::Pin, 100
 - openpr::netlist::Pip, 106
 - openpr::netlist::Point, 111
 - openpr::prohibitRange, 114
- operator()
 - openpr::netlist::eq_net, 62
 - openpr::netlist::eq_pip, 62
 - openpr::netlist::eq_point, 63
 - openpr::netlist::eq_segment, 63

- openpr::netlist::hash_net, 65
- openpr::netlist::hash_pip, 66
- openpr::netlist::hash_point, 66
- openpr::netlist::hash_segment, 67
- openpr::netlist::InPin, 68, 69
- openpr::netlist::Net, 73
- openpr::netlist::OutPin, 96
- openpr::netlist::Pin, 99
- openpr::netlist::Pip, 105
- openpr::netlist::Point, 110
- operator=
 - openpr::netlist::InPin, 69
 - openpr::netlist::Net, 74
 - openpr::netlist::OutPin, 96, 97
 - openpr::netlist::Pin, 99
 - openpr::netlist::Pip, 105, 106
- operator==
 - openpr::bitstream::tile_coord, 115
 - openpr::netlist::InPin, 69
 - openpr::netlist::Net, 74
 - openpr::netlist::OutPin, 97
 - openpr::netlist::Pin, 99, 100
 - openpr::netlist::Pip, 106
 - openpr::netlist::Point, 111
- OutPin
 - openpr::netlist::OutPin, 96
- outpin
 - openpr::netlist::Net, 75
 - openpr::netlist::OutPin, 98
- outpinLetter
 - openpr::netlist::Net, 75
- outPins
 - openpr::netlist::Net, 75
- outputXDL
 - openpr::netlist::NetList, 81
- PAD
 - openpr::bitstream, 15
- parseLocation
 - openpr::netlist::Pip, 106
- partialBsPath
 - openpr::openPR, 90
- partialPath
 - openpr::openPR, 90
- passThroughNet2
 - openpr::openPR, 91
- passThroughNetName
 - openpr::openPR, 91
- pcfPath
 - openpr::openPR, 91
- Pin
 - openpr::netlist::Pin, 99
- Pip
 - openpr::netlist::Pip, 102, 103
- pips
 - openpr::netlist::Net, 76
- pipToNet
 - openpr::netlist::NetList, 81
- placedXDLInput
 - openpr::AntiCoreBase, 32
- placedXdlPath
 - openpr::openPR, 91
- placeMacro
 - openpr::AntiCoreBase, 27
 - openpr::AntiCoreV4, 35
 - openpr::AntiCoreV5, 38
- placeMacros
 - openpr::openPR, 86
- Point
 - openpr::netlist::Point, 109
- pointToPip
 - openpr::netlist::NetList, 81
- print
 - openpr::bitstream::tile_data, 117
- printData
 - openpr::netlist::InPin, 70
 - openpr::netlist::Net, 74
 - openpr::netlist::NetList, 79
 - openpr::netlist::OutPin, 97
 - openpr::netlist::Pin, 100
 - openpr::netlist::Pip, 106, 107
- prohibitedSites
 - openpr::AntiCoreBase, 32
- prohibitRange
 - openpr::prohibitRange, 113
- projectPath
 - openpr::openPR, 91
- readHeader
 - openpr::bitstream::bitstream, 48
- readPackets
 - openpr::bitstream::bitstream, 48
 - openpr::bitstream::v4_bitstream, 124
 - openpr::bitstream::v5_bitstream, 132
- readXilinxString
 - openpr::bitstream::bitstream, 48
- regionDefined
 - openpr::openPR, 91
- relativePath
 - openpr::OpenPRTree, 93
- remotePin
 - openpr::netlist::Net, 75
- removePip
 - openpr::netlist::NetList, 80
- retrieveDynamicRegion
 - openpr::AntiCoreBase, 27
- reverseFrameBits
 - openpr::bitstream::v4_bitstream, 124

- routeBlocker
 - openpr::openPR, 87
- routedXdlPath
 - openpr::openPR, 91
- routePTScriptPath
 - openpr::openPR, 91
- routing_table
 - openpr::bitstream::device, 61
- row
 - openpr::bitstream::frame_addr, 65
- row_layout
 - openpr::bitstream::device, 61
- row_width
 - openpr::bitstream::device, 61
- sCommandName
 - openpr::bitstream::v4_bitstream, 126
 - openpr::bitstream::v5_bitstream, 136
- sDatabasePath
 - openpr::OpenPRTree, 94
- sEdaPath
 - openpr::OpenPRTree, 94
- segmentIndex
 - openpr::netlist::Point, 112
- segmentToNet
 - openpr::netlist::NetList, 81
- serialize
 - openpr::openPR, 87
- set
 - openpr::bitstream::tile_coord, 115
- setIndices
 - openpr::netlist::Point, 112
- setMode
 - openpr::AntiCoreBase, 28
- setRegionVertices
 - openpr::AntiCoreBase, 28
- setupAntiCore
 - openpr::openPR, 87
- setupDynamicRegion
 - openpr::openPR, 87
- setupRouteBlocker
 - openpr::AntiCoreBase, 28
- sExecutablePath
 - openpr::OpenPRTree, 94
- shrinkRegion
 - openpr::AntiCoreBase, 29
- sinks_buf
 - openpr::AntiCoreBase, 32
- siteBlocker
 - openpr::openPR, 88
- siteMap
 - openpr::AntiCoreBase, 32
- siteNameToTileIndex
 - openpr::AntiCoreBase, 29
- sLogPath
 - openpr::OpenPRTree, 94
- sOpcodeName
 - openpr::bitstream::v4_bitstream, 126
 - openpr::bitstream::v5_bitstream, 136
- source
 - openpr::netlist::Pip, 107
 - openpr::netlist::Point, 112
- sourceDestination
 - openpr::netlist::Point, 112
- sources_buf
 - openpr::AntiCoreBase, 32
- sRegisterName
 - openpr::bitstream::v4_bitstream, 126
 - openpr::bitstream::v5_bitstream, 136
- sRelativePath
 - openpr::OpenPRTree, 94
- startTile
 - openpr::AntiCoreBase, 32
- staticPath
 - openpr::openPR, 91
- staticPlacedXDLPath
 - openpr::openPR, 91
- str
 - openpr::bitstream::frame_addr, 64
- string
 - openpr, 14
- structToFar
 - openpr::bitstream::bitstream, 48
 - openpr::bitstream::v4_bitstream, 124
 - openpr::bitstream::v5_bitstream, 133
- sTypeName
 - openpr::bitstream::v4_bitstream, 126
 - openpr::bitstream::v5_bitstream, 136
- sWorkingPath
 - openpr::OpenPRTree, 94
- tb
 - openpr::bitstream::frame_addr, 65
- tile
 - openpr::netlist::Point, 112
- tile_coord
 - openpr::bitstream::tile_coord, 115
- tile_data
 - openpr::bitstream::tile_data, 117
- tile_frames
 - openpr::bitstream::architecture, 40
 - openpr::bitstream::device, 61
- tile_map
 - openpr::bitstream::bitstream, 52
- tile_offset
 - openpr::bitstream::device, 59
 - openpr::bitstream::virtex4, 139
 - openpr::bitstream::virtex5, 142

- tile_types
 - openpr::bitstream, 15
- tile_width
 - openpr::bitstream::device, 61
- tilecoord_to_major
 - openpr::bitstream::device, 59
- tileMap
 - openpr::bitstream::bitstream, 52
- tilesPerRegion
 - openpr::AntiCoreV4, 36
 - openpr::AntiCoreV5, 39
- tileToSiteMap
 - openpr::AntiCoreBase, 19
- tokenizer
 - openpr::netlist, 16
- top
 - openpr::bitstream::v4_bitstream, 127
- topLevelParser
 - openpr::netlist::NetList, 80
- TRANSCV
 - openpr::bitstream, 15
- type
 - openpr::bitstream::frame_addr, 65
 - openpr::netlist::Pip, 107
- ucfPath
 - openpr::openPR, 92
- unmangleTilePair
 - openpr::bitstream::v4_bitstream, 124
- unroutePTScriptPath
 - openpr::openPR, 92
- updateRegion
 - openpr::AntiCoreBase, 29
- updateRegionExpand
 - openpr::AntiCoreBase, 30
- v4_bitstream
 - openpr::bitstream::v4_bitstream, 123
- v5_bitstream
 - openpr::bitstream::v5_bitstream, 132
- validateRegion
 - openpr::AntiCoreBase, 30
- validBoundaries
 - openpr::AntiCoreBase, 32
- virtex4
 - openpr::bitstream::architecture, 40
 - openpr::bitstream::virtex4, 138
- virtex4_clb_slices
 - openpr::bitstream::virtex4, 138
- virtex4_frame_height
 - openpr::bitstream::virtex4, 138
- virtex4_frame_words
 - openpr::bitstream::virtex4, 138
- virtex4_block_type
 - openpr::bitstream::virtex4, 139
- virtex4_logic_table
 - openpr::bitstream::virtex4, 139
- virtex4_routing_table
 - openpr::bitstream::virtex4, 139
- virtex4_tile_frames
 - openpr::bitstream::virtex4, 140
- virtex5
 - openpr::bitstream::architecture, 40
 - openpr::bitstream::virtex5, 142
- virtex5_clb_slices
 - openpr::bitstream::virtex5, 142
- virtex5_frame_height
 - openpr::bitstream::virtex5, 142
- virtex5_frame_words
 - openpr::bitstream::virtex5, 142
- virtex5_num_blk_types
 - openpr::bitstream::virtex5, 142
- virtex5_block_type
 - openpr::bitstream::virtex5, 143
- virtex5_logic_table
 - openpr::bitstream::virtex5, 143
- virtex5_routing_table
 - openpr::bitstream::virtex5, 143
- virtex5_tile_frames
 - openpr::bitstream::virtex5, 143
- wholeData
 - openpr::netlist::Pip, 107
- wires_buf
 - openpr::AntiCoreBase, 32
- workingPath
 - openpr::OpenPRTree, 94
- write
 - openpr::bitstream::bitstream, 49
- writeBitstream
 - openpr::bitstream::bitstream, 49
- writeFrameData
 - openpr::bitstream::v4_bitstream, 125
 - openpr::bitstream::v5_bitstream, 133
- writeFrames
 - openpr::bitstream::bitstream, 49
- writeHeader
 - openpr::bitstream::bitstream, 50
- writePacketHeader
 - openpr::bitstream::v4_bitstream, 125
 - openpr::bitstream::v5_bitstream, 133
- writePackets
 - openpr::bitstream::bitstream, 50
 - openpr::bitstream::v4_bitstream, 125
 - openpr::bitstream::v5_bitstream, 134
- writePacketsPartial
 - openpr::bitstream::bitstream, 50
 - openpr::bitstream::v4_bitstream, 125

- openpr::bitstream::v5_bitstream, 134
- writePartialFrames
 - openpr::bitstream::v4_bitstream, 125
 - openpr::bitstream::v5_bitstream, 135
- writeXilinxString
 - openpr::bitstream::bitstream, 50
- x
 - openpr::bitstream::tile_coord, 116
- xc4vfx60
 - openpr::bitstream::xc4vfx60, 145
- xc4vfx60_num_cols
 - openpr::bitstream::xc4vfx60, 145
- xc4vfx60_num_rows
 - openpr::bitstream::xc4vfx60, 145
- xc4vfx60_id
 - openpr::bitstream::xc4vfx60, 146
- xc4vfx60_name
 - openpr::bitstream::xc4vfx60, 146
- xc4vfx60_row_layout
 - openpr::bitstream::xc4vfx60, 146
- xc4vlx15
 - openpr::bitstream::xc4vlx15, 148
- xc4vlx15_num_cols
 - openpr::bitstream::xc4vlx15, 148
- xc4vlx15_num_rows
 - openpr::bitstream::xc4vlx15, 148
- xc4vlx15_id
 - openpr::bitstream::xc4vlx15, 148
- xc4vlx15_name
 - openpr::bitstream::xc4vlx15, 148
- xc4vlx15_row_layout
 - openpr::bitstream::xc4vlx15, 148
- xc4vlx60
 - openpr::bitstream::xc4vlx60, 150
- xc4vlx60_num_cols
 - openpr::bitstream::xc4vlx60, 150
- xc4vlx60_num_rows
 - openpr::bitstream::xc4vlx60, 150
- xc4vlx60_id
 - openpr::bitstream::xc4vlx60, 150
- xc4vlx60_name
 - openpr::bitstream::xc4vlx60, 150
- xc4vlx60_row_layout
 - openpr::bitstream::xc4vlx60, 150
- xc5vlx110t
 - openpr::bitstream::xc5vlx110t, 152
- xc5vlx110t_num_cols
 - openpr::bitstream::xc5vlx110t, 152
- xc5vlx110t_num_rows
 - openpr::bitstream::xc5vlx110t, 152
- xc5vlx110t_id
 - openpr::bitstream::xc5vlx110t, 153
- xc5vlx110t_name
 - openpr::bitstream::xc5vlx110t, 153
- xc5vlx50
 - openpr::bitstream::xc5vlx50, 155
- xc5vlx50_num_cols
 - openpr::bitstream::xc5vlx50, 155
- xc5vlx50_num_rows
 - openpr::bitstream::xc5vlx50, 155
- xc5vlx50_id
 - openpr::bitstream::xc5vlx50, 155
- xc5vlx50_name
 - openpr::bitstream::xc5vlx50, 155
- xc5vlx50_row_layout
 - openpr::bitstream::xc5vlx50, 155
- xc5vlx50t
 - openpr::bitstream::xc5vlx50t, 157
- xc5vlx50t_num_cols
 - openpr::bitstream::xc5vlx50t, 157
- xc5vlx50t_num_rows
 - openpr::bitstream::xc5vlx50t, 157
- xc5vlx50t_id
 - openpr::bitstream::xc5vlx50t, 157
- xc5vlx50t_name
 - openpr::bitstream::xc5vlx50t, 157
- xc5vsx95t
 - openpr::bitstream::xc5vsx95t, 159
- xc5vsx95t_num_cols
 - openpr::bitstream::xc5vsx95t, 159
- xc5vsx95t_num_rows
 - openpr::bitstream::xc5vsx95t, 159
- xc5vsx95t_id
 - openpr::bitstream::xc5vsx95t, 160
- xc5vsx95t_name
 - openpr::bitstream::xc5vsx95t, 160
- xc5vsx95t_row_layout
 - openpr::bitstream::xc5vsx95t, 160
- xdl_layout
 - openpr::bitstream::device, 61
- xLoc
 - openpr::netlist::Pip, 108
- xMax
 - openpr::AntiCoreBase, 32
 - openpr::openPR, 92
- xMin
 - openpr::AntiCoreBase, 33
 - openpr::openPR, 92
- y
 - openpr::bitstream::tile_coord, 116
- yLoc
 - openpr::netlist::Pip, 108

yMax

openpr::AntiCoreBase, [33](#)

openpr::openPR, [92](#)

yMin

openpr::AntiCoreBase, [33](#)

openpr::openPR, [92](#)