

# Spartan-3E Libraries Guide for HDL Designs

UG617 (v 11.3) September 16, 2009

## Xilinx Trademarks and Copyright Information



Xilinx is disclosing this user guide, manual, release note, and/or specification (the "Documentation") to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU "AS-IS" WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© Copyright 2002-2009 Xilinx Inc. All Rights Reserved. XILINX, the Xilinx logo, the Brand Window and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

# Table of Contents

---

Xilinx Trademarks and Copyright Information.....	2
Chapter 1 About this Guide.....	17
About Design Elements .....	17
Design Entry Methods.....	17
Chapter 2 Functional Categories.....	19
Chapter 3 About Design Elements.....	25
BSCAN_SPARTAN3 .....	26
Introduction.....	26
Port Descriptions.....	26
Design Entry Method .....	27
For More Information.....	27
BUFCF .....	28
Introduction.....	28
Design Entry Method .....	28
For More Information.....	28
BUFG.....	29
Introduction.....	29
Port Descriptions.....	29
Design Entry Method .....	29
For More Information.....	30
BUFGCE .....	31
Introduction.....	31
Logic Table .....	31
Design Entry Method .....	31
For More Information.....	32
BUFGCE_1 .....	33
Introduction.....	33
Logic Table .....	33
Design Entry Method .....	33
For More Information.....	34
BUFGMUX.....	35
Introduction.....	35
Logic Table .....	35
Port Descriptions.....	35
Design Entry Method .....	35
Available Attributes .....	36
For More Information.....	36
BUFGMUX_1.....	37
Introduction.....	37
Logic Table .....	37
Design Entry Method .....	37
For More Information.....	38
CAPTURE_SPARTAN3 .....	39
Introduction.....	39
Port Descriptions.....	39
Design Entry Method .....	39
Available Attributes .....	39
For More Information.....	40
DCM_SP .....	41
Introduction.....	41
Port Descriptions.....	41
Design Entry Method .....	42
Available Attributes .....	42
For More Information.....	45
FDCE .....	46

Introduction.....	46
Logic Table .....	46
Design Entry Method .....	46
Available Attributes .....	46
For More Information.....	47
<b>FDCE_1.....</b>	<b>48</b>
Introduction.....	48
Logic Table .....	48
Design Entry Method .....	48
Available Attributes .....	48
For More Information.....	49
<b>FDCPE.....</b>	<b>50</b>
Introduction.....	50
Logic Table .....	50
Port Descriptions.....	50
Design Entry Method .....	51
Available Attributes .....	51
For More Information.....	51
<b>FDCPE_1.....</b>	<b>52</b>
Introduction.....	52
Logic Table .....	52
Port Descriptions.....	52
Design Entry Method .....	53
Available Attributes .....	53
For More Information.....	53
<b>FDRSE .....</b>	<b>54</b>
Introduction.....	54
Logic Table .....	54
Design Entry Method .....	54
Available Attributes .....	54
For More Information.....	55
<b>FDRSE_1 .....</b>	<b>56</b>
Introduction.....	56
Logic Table .....	56
Design Entry Method .....	56
Available Attributes .....	56
For More Information.....	57
<b>IBUF .....</b>	<b>58</b>
Introduction.....	58
Port Descriptions.....	58
Design Entry Method .....	58
Available Attributes .....	58
For More Information.....	59
<b>IBUFDS.....</b>	<b>60</b>
Introduction.....	60
Logic Table .....	60
Port Descriptions.....	60
Design Entry Method .....	60
Available Attributes .....	61
For More Information.....	62
<b>IBUFG.....</b>	<b>63</b>
Introduction.....	63
Port Descriptions.....	63
Design Entry Method .....	63
Available Attributes .....	63
For More Information.....	64
<b>IBUFGDS .....</b>	<b>65</b>
Introduction.....	65
Logic Table .....	65

Port Descriptions.....	65
Design Entry Method .....	65
Available Attributes .....	66
For More Information.....	66
IDDR2.....	67
Introduction.....	67
Logic Table .....	67
Design Entry Method .....	67
Available Attributes .....	68
For More Information.....	69
IOBUF.....	70
Introduction.....	70
Logic Table .....	70
Port Descriptions.....	70
Design Entry Method .....	70
Available Attributes .....	71
For More Information.....	72
IOBUFDS .....	73
Introduction.....	73
Logic Table .....	73
Port Descriptions.....	73
Design Entry Method .....	73
Available Attributes .....	74
For More Information.....	75
KEEPER .....	76
Introduction.....	76
Port Descriptions.....	76
Design Entry Method .....	76
For More Information.....	77
LDCPE.....	78
Introduction.....	78
Logic Table .....	78
Port Descriptions.....	78
Design Entry Method .....	79
Available Attributes .....	79
For More Information.....	79
LUT1 .....	80
Introduction.....	80
Logic Table .....	80
Design Entry Method .....	80
Available Attributes .....	81
For More Information.....	81
LUT1_D .....	82
Introduction.....	82
Logic Table .....	82
Design Entry Method .....	82
Available Attributes .....	82
For More Information.....	83
LUT1_L.....	84
Introduction.....	84
Logic Table .....	84
Design Entry Method .....	84
Available Attributes .....	84
For More Information.....	85
LUT2 .....	86
Introduction.....	86
Logic Table .....	86
Design Entry Method .....	86
Available Attributes .....	87

---

For More Information.....	87
LUT2_D .....	88
Introduction.....	88
Logic Table .....	88
Design Entry Method .....	88
Available Attributes .....	88
For More Information.....	89
LUT2_L.....	90
Introduction.....	90
Logic Table .....	90
Design Entry Method .....	90
Available Attributes .....	91
For More Information.....	91
LUT3 .....	92
Introduction.....	92
Logic Table .....	92
Design Entry Method .....	93
Available Attributes .....	93
For More Information.....	93
LUT3_D .....	94
Introduction.....	94
Logic Table .....	94
Design Entry Method .....	94
Available Attributes .....	95
For More Information.....	95
LUT3_L.....	96
Introduction.....	96
Logic Table .....	96
Design Entry Method .....	97
Available Attributes .....	97
For More Information.....	97
LUT4 .....	98
Introduction.....	98
Logic Table .....	99
Design Entry Method .....	99
Available Attributes .....	99
For More Information.....	100
LUT4_D .....	101
Introduction.....	101
Logic Table .....	102
Design Entry Method .....	102
Available Attributes .....	102
For More Information.....	103
LUT4_L.....	104
Introduction.....	104
Logic Table .....	105
Design Entry Method .....	105
Available Attributes .....	105
For More Information.....	106
MULT_AND.....	107
Introduction.....	107
Logic Table .....	107
Design Entry Method .....	107
For More Information.....	108
MULT18X18 .....	109
Introduction.....	109
Logic Table .....	109
Design Entry Method .....	109
For More Information.....	110

---

MULT18X18S.....	111
Introduction.....	111
Logic Table .....	111
Design Entry Method .....	111
For More Information.....	112
MULT18X18SIO.....	113
Introduction.....	113
Design Entry Method .....	113
Available Attributes .....	113
For More Information.....	114
MUXCY .....	115
Introduction.....	115
Logic Table .....	115
Design Entry Method .....	115
For More Information.....	116
MUXCY_D .....	117
Introduction.....	117
Logic Table .....	117
Design Entry Method .....	117
For More Information.....	118
MUXCY_L.....	119
Introduction.....	119
Logic Table .....	119
Design Entry Method .....	119
For More Information.....	120
MUXF5 .....	121
Introduction.....	121
Logic Table .....	121
Design Entry Method .....	121
For More Information.....	122
MUXF5_D .....	123
Introduction.....	123
Logic Table .....	123
Design Entry Method .....	123
For More Information.....	124
MUXF5_L.....	125
Introduction.....	125
Logic Table .....	125
Design Entry Method .....	125
For More Information.....	126
MUXF6 .....	127
Introduction.....	127
Logic Table .....	127
Design Entry Method .....	127
For More Information.....	128
MUXF6_D .....	129
Introduction.....	129
Logic Table .....	129
Design Entry Method .....	129
For More Information.....	130
MUXF6_L.....	131
Introduction.....	131
Logic Table .....	131
Design Entry Method .....	131
For More Information.....	132
MUXF7 .....	133
Introduction.....	133
Logic Table .....	133
Port Descriptions.....	133

---

Design Entry Method .....	133
For More Information .....	134
MUXF7_D .....	135
Introduction.....	135
Logic Table .....	135
Port Descriptions.....	135
Design Entry Method .....	135
For More Information.....	136
MUXF7_L.....	137
Introduction.....	137
Logic Table .....	137
Port Descriptions.....	137
Design Entry Method .....	137
For More Information.....	138
MUXF8 .....	139
Introduction.....	139
Logic Table .....	139
Port Descriptions.....	139
Design Entry Method .....	139
For More Information.....	140
MUXF8_D .....	141
Introduction.....	141
Logic Table .....	141
Port Descriptions.....	141
Design Entry Method .....	141
For More Information.....	142
MUXF8_L.....	143
Introduction.....	143
Logic Table .....	143
Port Descriptions.....	143
Design Entry Method .....	143
For More Information.....	144
OBUF.....	145
Introduction.....	145
Port Descriptions.....	145
Design Entry Method .....	145
Available Attributes .....	145
For More Information.....	146
OBUFDS .....	147
Introduction.....	147
Logic Table .....	147
Port Descriptions.....	147
Design Entry Method .....	147
Available Attributes .....	147
For More Information.....	148
OBUFT.....	149
Introduction.....	149
Logic Table .....	149
Port Descriptions.....	149
Design Entry Method .....	149
Available Attributes .....	150
For More Information.....	150
OBUFTDS .....	151
Introduction.....	151
Logic Table .....	151
Port Descriptions.....	151
Design Entry Method .....	151
Available Attributes .....	151
For More Information.....	152

---

ODDR2 .....	153
Introduction.....	153
Logic Table .....	153
Design Entry Method .....	153
Available Attributes .....	154
For More Information.....	155
PULLDOWN .....	156
Introduction.....	156
Port Descriptions.....	156
Design Entry Method .....	156
For More Information.....	157
PULLUP.....	158
Introduction.....	158
Port Descriptions.....	158
Design Entry Method .....	158
For More Information.....	159
RAM16X1D .....	160
Introduction.....	160
Logic Table .....	160
Design Entry Method .....	161
Available Attributes .....	161
For More Information.....	162
RAM16X1D_1.....	163
Introduction.....	163
Logic Table .....	163
Port Descriptions.....	164
Design Entry Method .....	164
Available Attributes .....	164
For More Information.....	165
RAM16X1S.....	166
Introduction.....	166
Logic Table .....	166
Design Entry Method .....	166
Available Attributes .....	166
For More Information.....	167
RAM16X1S_1.....	168
Introduction.....	168
Logic Table .....	168
Design Entry Method .....	168
Available Attributes .....	169
For More Information.....	169
RAM16X2S.....	170
Introduction.....	170
Logic Table .....	170
Design Entry Method .....	171
Available Attributes .....	171
For More Information.....	172
RAM16X4S.....	173
Introduction.....	173
Logic Table .....	173
Design Entry Method .....	173
Available Attributes .....	174
For More Information.....	175
RAM16X8S.....	176
Introduction.....	176
Logic Table .....	176
Design Entry Method .....	176
Available Attributes .....	176
For More Information.....	177

---

RAM32X1S.....	178
Introduction.....	178
Logic Table .....	178
Design Entry Method .....	178
Available Attributes .....	178
For More Information.....	179
RAM32X1S_1.....	180
Introduction.....	180
Logic Table .....	180
Design Entry Method .....	180
Available Attributes .....	180
For More Information.....	181
RAM32X2S.....	182
Introduction.....	182
Logic Table .....	182
Design Entry Method .....	182
Available Attributes .....	183
For More Information.....	183
RAM32X4S.....	184
Introduction.....	184
Logic Table .....	184
Design Entry Method .....	184
Available Attributes .....	185
For More Information.....	186
RAM32X8S.....	187
Introduction.....	187
Logic Table .....	187
Design Entry Method .....	187
Available Attributes .....	188
For More Information.....	189
RAM64X1S.....	190
Introduction.....	190
Logic Table .....	190
Design Entry Method .....	190
Available Attributes .....	190
For More Information.....	191
RAM64X1S_1.....	192
Introduction.....	192
Logic Table .....	192
Design Entry Method .....	192
Available Attributes .....	192
For More Information.....	193
RAM64X2S.....	194
Introduction.....	194
Logic Table .....	194
Design Entry Method .....	194
Available Attributes .....	195
For More Information.....	196
RAMB16_S1 .....	197
Introduction.....	197
Logic Table .....	197
Initialization .....	198
Design Entry Method .....	199
Available Attributes .....	199
For More Information.....	202
RAMB16_S1_S1 .....	203
Introduction.....	203
Logic Table .....	204
Port Descriptions.....	205

---

Design Entry Method .....	209
Available Attributes .....	209
For More Information .....	213
RAMB16_S1_S18.....	214
Introduction.....	214
Logic Table .....	215
Port Descriptions.....	216
Design Entry Method .....	220
Available Attributes .....	220
For More Information.....	224
RAMB16_S1_S2 .....	225
Introduction.....	225
Logic Table .....	225
Port Descriptions.....	227
Design Entry Method .....	231
Available Attributes .....	231
For More Information.....	235
RAMB16_S1_S36.....	236
Introduction.....	236
Logic Table .....	237
Port Descriptions.....	238
Design Entry Method .....	242
Available Attributes .....	242
For More Information.....	246
RAMB16_S1_S4 .....	247
Introduction.....	247
Logic Table .....	248
Port Descriptions.....	249
Design Entry Method .....	253
Available Attributes .....	253
For More Information.....	257
RAMB16_S1_S9 .....	258
Introduction.....	258
Logic Table .....	259
Port Descriptions.....	260
Design Entry Method .....	264
Available Attributes .....	264
For More Information.....	268
RAMB16_S18.....	269
Introduction.....	269
Logic Table .....	269
Initialization .....	270
Design Entry Method .....	271
Available Attributes .....	271
For More Information.....	274
RAMB16_S18_S18.....	275
Introduction.....	275
Logic Table .....	275
Port Descriptions.....	277
Design Entry Method .....	281
Available Attributes .....	281
For More Information.....	286
RAMB16_S18_S36.....	287
Introduction.....	287
Logic Table .....	287
Port Descriptions.....	289
Design Entry Method .....	293
Available Attributes .....	293
For More Information.....	298

---

RAMB16_S2 .....	299
Introduction.....	299
Logic Table .....	299
Initialization .....	300
Design Entry Method .....	301
Available Attributes .....	301
For More Information.....	304
RAMB16_S2_S18.....	305
Introduction.....	305
Logic Table .....	305
Port Descriptions.....	307
Design Entry Method .....	311
Available Attributes .....	311
For More Information.....	312
RAMB16_S2_S2 .....	313
Introduction.....	313
Logic Table .....	313
Port Descriptions.....	315
Design Entry Method .....	319
Available Attributes .....	319
For More Information.....	323
RAMB16_S2_S36.....	324
Introduction.....	324
Logic Table .....	324
Port Descriptions.....	326
Design Entry Method .....	330
Available Attributes .....	330
For More Information.....	334
RAMB16_S2_S4 .....	335
Introduction.....	335
Logic Table .....	335
Port Descriptions.....	337
Design Entry Method .....	341
Available Attributes .....	341
For More Information.....	345
RAMB16_S2_S9 .....	346
Introduction.....	346
Logic Table .....	346
Port Descriptions.....	348
Design Entry Method .....	352
Available Attributes .....	352
For More Information.....	357
RAMB16_S36.....	358
Introduction.....	358
Logic Table .....	358
Initialization .....	359
Design Entry Method .....	360
Available Attributes .....	360
For More Information.....	363
RAMB16_S36_S36.....	364
Introduction.....	364
Logic Table .....	365
Port Descriptions.....	366
Design Entry Method .....	370
Available Attributes .....	370
For More Information.....	375
RAMB16_S4 .....	376
Introduction.....	376
Logic Table .....	376

Initialization .....	377
Design Entry Method .....	378
Available Attributes .....	378
For More Information .....	381
RAMB16_S4_S18.....	382
Introduction.....	382
Logic Table .....	383
Port Descriptions.....	384
Design Entry Method .....	388
Available Attributes .....	388
For More Information.....	392
RAMB16_S4_S36.....	393
Introduction.....	393
Logic Table .....	393
Port Descriptions.....	395
Design Entry Method .....	399
Available Attributes .....	399
For More Information.....	403
RAMB16_S4_S4 .....	404
Introduction.....	404
Logic Table .....	404
Port Descriptions.....	406
Design Entry Method .....	410
Available Attributes .....	410
For More Information.....	414
RAMB16_S4_S9 .....	415
Introduction.....	415
Logic Table .....	415
Port Descriptions.....	417
Design Entry Method .....	421
Available Attributes .....	421
For More Information.....	425
RAMB16_S9 .....	426
Introduction.....	426
Logic Table .....	426
Initialization .....	427
Design Entry Method .....	428
Available Attributes .....	428
For More Information.....	431
RAMB16_S9_S18.....	432
Introduction.....	432
Logic Table .....	432
Port Descriptions.....	434
Design Entry Method .....	438
Available Attributes .....	438
For More Information.....	443
RAMB16_S9_S36.....	444
Introduction.....	444
Logic Table .....	444
Port Descriptions.....	446
Design Entry Method .....	450
Available Attributes .....	450
For More Information.....	455
RAMB16_S9_S9 .....	456
Introduction.....	456
Logic Table .....	456
Port Descriptions.....	458
Design Entry Method .....	462
Available Attributes .....	462

---

For More Information.....	467
ROM128X1 .....	468
Introduction.....	468
Logic Table .....	468
Design Entry Method .....	469
Available Attributes .....	469
For More Information.....	469
ROM16X1.....	470
Introduction.....	470
Logic Table .....	470
Design Entry Method .....	471
Available Attributes .....	471
For More Information.....	471
ROM256X1 .....	472
Introduction.....	472
Logic Table .....	472
Design Entry Method .....	473
Available Attributes .....	473
For More Information.....	474
ROM32X1.....	475
Introduction.....	475
Logic Table .....	475
Design Entry Method .....	476
Available Attributes .....	476
For More Information.....	476
ROM64X1.....	477
Introduction.....	477
Logic Table .....	477
Design Entry Method .....	478
Available Attributes .....	478
For More Information.....	478
SRL16 .....	479
Introduction.....	479
Logic Table .....	479
Design Entry Method .....	479
Available Attributes .....	480
For More Information.....	480
SRL16_1 .....	481
Introduction.....	481
Logic Table .....	481
Design Entry Method .....	481
Available Attributes .....	482
For More Information.....	482
SRL16E .....	483
Introduction.....	483
Logic Table .....	483
Port Descriptions.....	484
Design Entry Method .....	484
Available Attributes .....	484
For More Information.....	485
SRL16E_1 .....	486
Introduction.....	486
Logic Table .....	486
Design Entry Method .....	486
Available Attributes .....	487
For More Information.....	487
SRLC16 .....	488
Introduction.....	488
Logic Table .....	488

---

Design Entry Method .....	488
Available Attributes .....	489
For More Information .....	489
SRLC16_1.....	490
Introduction.....	490
Logic Table .....	490
Design Entry Method .....	490
Available Attributes .....	491
For More Information.....	491
SRLC16E.....	492
Introduction.....	492
Logic Table .....	492
Design Entry Method .....	492
Available Attributes .....	493
For More Information.....	493
SRLC16E_1.....	494
Introduction.....	494
Logic Table .....	494
Design Entry Method .....	495
Available Attributes .....	495
For More Information.....	496
STARTUP_SPARTAN3E.....	497
Introduction.....	497
Design Entry Method .....	497
For More Information.....	498
XORCY .....	499
Introduction.....	499
Logic Table .....	499
Design Entry Method .....	499
For More Information.....	500
XORCY_D .....	501
Introduction.....	501
Logic Table .....	501
Design Entry Method .....	501
For More Information.....	502
XORCY_L .....	503
Introduction.....	503
Logic Table .....	503
Design Entry Method .....	503
For More Information.....	504



# About this Guide

---

This HDL guide is part of the ISE documentation collection. A separate version of this guide is available if you prefer to work with schematics.

This guide contains the following:

- Introduction.
- A list of design elements supported in this architecture, organized by functional categories.
- Individual descriptions of each available primitive.

## About Design Elements

This version of the Libraries Guide describes the primitives that comprise the Xilinx Unified Libraries for this architecture, and includes examples of instantiation code for each element.

Primitives are Xilinx components that are native to the FPGA you are targeting. If you instantiate a primitive in your design, after the translation process you will end up with the exact same component in the back end. For example, if you instantiate the Virtex-5 element known as ISERDES\_NODELAY as a user primitive, after you run translate (ngdbuild) you will end up with an ISERDES\_NODELAY in the back end as well. If you were using ISERDES in a Virtex-5 device, then this will automatically retarget to an ISERDES\_NODELAY for Virtex-5 in the back end. Hence, this concept of a “primitive” differs from other uses of that term in this technology.

Xilinx maintains software libraries with hundreds of functional design elements (unimacros and primitives) for different device architectures. New functional elements are assembled with each release of development system software. In addition to a comprehensive Unified Library containing all design elements, beginning in 2003, Xilinx developed a separate library for each architecture. This guide is one in a series of architecture-specific libraries.

## Design Entry Methods

For each design element in this guide, Xilinx evaluates the four options and recommends what we believe is the best solution for you. The four options are:

- **Instantiation** - This component can be instantiated directly into the design. This method is useful if you want to control the exact placement of the individual blocks.
- **Inference** - This component can be inferred by most supported synthesis tools. You should use this method if you want to have complete flexibility and portability of the code to multiple architectures. Inference also gives the tools the ability to optimize for performance, area, or power, as specified by the user to the synthesis tool.
- **Coregen & Wizards** - This component can be used through Coregen or Wizards. You should use this method if you want to build large blocks of any FPGA primitive that cannot be inferred. When using this flow, you will have to re-generate your cores for each architecture that you are targeting.
- **Macro Support** - This component has a UniMacro that can be used. These components are in the UniMacro library in the Xilinx tool, and are used to instantiate primitives that are complex to instantiate by just using the primitives. The synthesis tools will automatically expand the unimacros to their underlying primitives.



# Functional Categories

---

This section categorizes, by function, the circuit design elements described in detail later in this guide. The elements (*primitives* and *macros*) are listed in alphanumeric order under each functional category.

Arithmetic Functions	RAM/ROM
Clock Components	Registers/Latches
Config/BSCAN Components	Shift Register LUT
I/O Components	Slice/CLB Primitives

## Arithmetic Functions

Design Element	Description
MULT18X18	Primitive: 18 x 18 Signed Multiplier
MULT18X18S	Primitive: 18 x 18 Signed Multiplier -- Registered Version
MULT18X18SIO	Primitive: 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset

## Clock Components

Design Element	Description
BUFG	Convenience Primitive: Global Clock Buffer
BUFGCE	Convenience Primitive: Global Clock Buffer with Clock Enable
BUFGCE_1	Convenience Primitive: Global Clock Buffer with Clock Enable and Output State 1
BUFGMUX	Convenience Primitive: Global Clock MUX Buffer
BUFGMUX_1	Convenience Primitive: Global Clock MUX Buffer with Output State 1
DCM_SP	Primitive: Digital Clock Manager
IBUFG	Primitive: Dedicated Input Clock Buffer
IBUFGDS	Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay

## Config/BSCAN Components

Design Element	Description
<a href="#">BSCAN_SPARTAN3</a>	Primitive: Spartan®-3 and Spartan-3E JTAG Boundary Scan Logic Access Circuit
<a href="#">CAPTURE_SPARTAN3</a>	Primitive: Spartan®-3 Register State Capture for Bitstream Readback
<a href="#">STARTUP_SPARTAN3E</a>	Primitive: Spartan®-3E User Interface to the GSR, GTS, Configuration Startup Sequence and Multi-Boot Trigger Circuitry

## I/O Components

Design Element	Description
<a href="#">IBUF</a>	Primitive: Input Buffer
<a href="#">IBUFDSDS</a>	Primitive: Differential Signaling Input Buffer
<a href="#">IBUFG</a>	Primitive: Dedicated Input Clock Buffer
<a href="#">IBUFGDS</a>	Primitive: Differential Signaling Dedicated Input Clock Buffer and Optional Delay
<a href="#">IOBUF</a>	Primitive: Bi-Directional Buffer
<a href="#">IOBUFDSDS</a>	Primitive: 3-State Differential Signaling I/O Buffer with Active Low Output Enable
<a href="#">KEEPER</a>	Primitive: KEEPER Symbol
<a href="#">OBUF</a>	Primitive: Output Buffer
<a href="#">OBUFDSDS</a>	Primitive: Differential Signaling Output Buffer
<a href="#">OBUFTDS</a>	Primitive: 3-State Output Buffer with Active Low Output Enable
<a href="#">OBUFTDS</a>	Primitive: 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
<a href="#">PULLDOWN</a>	Primitive: Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs
<a href="#">PULLUP</a>	Primitive: Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs

## RAM/ROM

Design Element	Description
<a href="#">RAM16X1D</a>	Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM
<a href="#">RAM16X1D_1</a>	Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock
<a href="#">RAM16X1S</a>	Primitive: 16-Deep by 1-Wide Static Synchronous RAM
<a href="#">RAM16X1S_1</a>	Primitive: 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
<a href="#">RAM16X2S</a>	Primitive: 16-Deep by 2-Wide Static Synchronous RAM
<a href="#">RAM16X4S</a>	Primitive: 16-Deep by 4-Wide Static Synchronous RAM
<a href="#">RAM16X8S</a>	Primitive: 16-Deep by 8-Wide Static Synchronous RAM
<a href="#">RAM32X1S</a>	Primitive: 32-Deep by 1-Wide Static Synchronous RAM
<a href="#">RAM32X1S_1</a>	Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
<a href="#">RAM32X2S</a>	Primitive: 32-Deep by 2-Wide Static Synchronous RAM
<a href="#">RAM32X4S</a>	Primitive: 32-Deep by 4-Wide Static Synchronous RAM
<a href="#">RAM32X8S</a>	Primitive: 32-Deep by 8-Wide Static Synchronous RAM

Design Element	Description
RAM64X1S	Primitive: 64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM64X2S	Primitive: 64-Deep by 2-Wide Static Synchronous RAM
RAMB16_S1	Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 1-bit Port
RAMB16_S1_S1	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit Ports
RAMB16_S1_S18	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 18-bit Ports
RAMB16_S1_S2	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 2-bit Ports
RAMB16_S1_S36	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 36-bit Ports
RAMB16_S1_S4	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 4-bit Ports
RAMB16_S1_S9	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 9-bit Ports
RAMB16_S18	Primitive: 16K-bit Data + 2K-bit Parity Memory, Single-Port Synchronous Block RAM with 18-bit Port
RAMB16_S18_S18	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit Ports
RAMB16_S18_S36	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit and 36-bit Ports
RAMB16_S2	Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 2-bit Port
RAMB16_S2_S18	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 18-bit Ports
RAMB16_S2_S2	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit Ports
RAMB16_S2_S36	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 36-bit Ports
RAMB16_S2_S4	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 4-bit Ports
RAMB16_S2_S9	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 9-bit Ports
RAMB16_S36	Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 36-bit Port
RAMB16_S36_S36	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with Two 36-bit Ports
RAMB16_S4	Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 4-bit Port
RAMB16_S4_S18	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 18-bit Ports
RAMB16_S4_S36	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 36-bit Ports
RAMB16_S4_S4	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit Ports

Design Element	Description
RAMB16_S4_S9	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 9-bit Ports
RAMB16_S9	Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 9-bit Port
RAMB16_S9_S18	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 18-bit Ports
RAMB16_S9_S36	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 36-bit Ports
RAMB16_S9_S9	Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit Ports
ROM128X1	Primitive: 128-Deep by 1-Wide ROM
ROM16X1	Primitive: 16-Deep by 1-Wide ROM
ROM256X1	Primitive: 256-Deep by 1-Wide ROM
ROM32X1	Primitive: 32-Deep by 1-Wide ROM
ROM64X1	Primitive: 64-Deep by 1-Wide ROM

## Registers/Latches

Design Element	Description
FDCE	Primitive: D Flip-Flop with Clock Enable and Asynchronous Clear
FDCE_1	Primitive: D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Clear
FDCPE	Primitive: D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
FDCPE_1	Primitive: D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Preset and Clear
FDRSE	Primitive: D Flip-Flop with Synchronous Reset and Set and Clock Enable
FDRSE_1	Primitive: D Flip-Flop with Negative-Clock Edge, Synchronous Reset and Set, and Clock Enable
IDDR2	Primitive: Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset
LDCPE	Primitive: Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable
ODDR2	Primitive: Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset

## Shift Register LUT

Design Element	Description
SRL16	Primitive: 16-Bit Shift Register Look-Up Table (LUT)
SRL16_1	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock
SRL16E	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable
SRL16E_1	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable
SRLC16	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry
SRLC16_1	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock
SRLC16E	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable
SRLC16E_1	Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable

## Slice/CLB Primitives

Design Element	Description
BUFCF	Primitive: Fast Connect Buffer
LUT1	Primitive: 1-Bit Look-Up Table with General Output
LUT1_D	Primitive: 1-Bit Look-Up Table with Dual Output
LUT1_L	Primitive: 1-Bit Look-Up Table with Local Output
LUT2	Primitive: 2-Bit Look-Up Table with General Output
LUT2_D	Primitive: 2-Bit Look-Up Table with Dual Output
LUT2_L	Primitive: 2-Bit Look-Up Table with Local Output
LUT3	Primitive: 3-Bit Look-Up Table with General Output
LUT3_D	Primitive: 3-Bit Look-Up Table with Dual Output
LUT3_L	Primitive: 3-Bit Look-Up Table with Local Output
LUT4	Primitive: 4-Bit Look-Up-Table with General Output
LUT4_D	Primitive: 4-Bit Look-Up Table with Dual Output
LUT4_L	Primitive: 4-Bit Look-Up Table with Local Output
MULT_AND	Primitive: Fast Multiplier AND
MUXCY	Primitive: 2-to-1 Multiplexer for Carry Logic with General Output
MUXCY_D	Primitive: 2-to-1 Multiplexer for Carry Logic with Dual Output
MUXCY_L	Primitive: 2-to-1 Multiplexer for Carry Logic with Local Output
MUXF5	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF5_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF5_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
MUXF6	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF6_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF6_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
MUXF7	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF7_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF7_L	Primitive: 2-to-1 look-up table Multiplexer with Local Output
MUXF8	Primitive: 2-to-1 Look-Up Table Multiplexer with General Output
MUXF8_D	Primitive: 2-to-1 Look-Up Table Multiplexer with Dual Output
MUXF8_L	Primitive: 2-to-1 Look-Up Table Multiplexer with Local Output
XORCY	Primitive: XOR for Carry Logic with General Output
XORCY_D	Primitive: XOR for Carry Logic with Dual Output
XORCY_L	Primitive: XOR for Carry Logic with Local Output

# About Design Elements

---

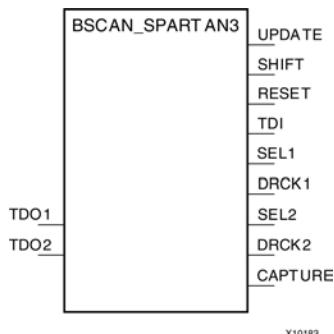
This section describes the design elements that can be used with this architecture. The design elements are organized alphabetically.

The following information is provided for each design element, where applicable:

- Name of element
- Brief description
- Schematic symbol (if any)
- Logic table (if any)
- Port descriptions
- Design Entry Method
- Available attributes (if any)
- Example instantiation code
- For more information

## BSCAN\_SPARTAN3

Primitive: Spartan®-3 and Spartan-3E JTAG Boundary Scan Logic Access Circuit



## Introduction

This design element allows access to and from internal logic by the JTAG Boundary Scan logic controller. This allows for communication between the internal running design and the dedicated JTAG pins of the FPGA.

**Note** For specific information on boundary scan for an architecture, see the Programmable Logic Data Sheet for this element.

## Port Descriptions

Port	Direction	Width	Function
TDI	Output	1	A mirror of the TDI input pin to the FPGA.
DRCK1, DRK2	Output	1	A mirror of the TCK input pin to the FPGA when the JTAG USER instruction is loaded and the JTAG TAP controller is in the SHIFT-DR state. DRK1 applies to the USER1 logic while DRK2 applies to USER2.
RESET	Output	1	Active upon the loading of the USER instruction. It asserts High when the JTAG TAP controller is in the TEST-LOGIC-RESET state.
SEL1, SEL2	Output	1	Indicates when the USER1 or USER2 instruction has been loaded into the JTAG Instruction Register. SEL1 or SEL2 becomes active in the UPDATE-IR state, and stays active until a new instruction is loaded.
SHIFT	Output	1	Active upon the loading of the USER instruction. It asserts High when the JTAG TAP controller is in the SHIFT-DR state.
CAPTURE	Output	1	Active upon the loading of the USER instruction. Asserts High when the JTAG TAP controller is in the CAPTURE-DR state.
UPDATE	Output	1	Active upon the loading of the USER instruction. It asserts High when the JTAG TAP controller is in the UPDATE-DR state.
TDO1, TDO2	Input	1	Active upon the loading of the USER1 or USER2 instruction. External JTAG TDO pin reflects data input to the component's TDO1 (USER1) or TDO2 (USER2) pin.

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BSCAN_SPARTAN3: Boundary Scan primitive for connecting internal logic to
--                  JTAG interface. Spartan-3
-- Xilinx HDL Libraries Guide, version 11.2

BSCAN_SPARTAN3_inst : BSCAN_SPARTAN3
port map (
    CAPTURE => CAPTURE, -- CAPTURE output from TAP controller
    DRCK1 => DRCK1, -- Data register output for USER1 functions
    DRCK2 => DRCK2, -- Data register output for USER2 functions
    RESET => RESET, -- Reset output from TAP controller
    SEL1 => SEL1, -- USER1 active output
    SEL2 => SEL2, -- USER2 active output
    SHIFT => SHIFT, -- SHIFT output from TAP controller
    TDI => TDI, -- TDI output from TAP controller
    UPDATE => UPDATE, -- UPDATE output from TAP controller
    TDO1 => TDO1, -- Data input for USER1 function
    TDO2 => TDO2 -- Data input for USER2 function
);
-- End of BSCAN_SPARTAN3_inst instantiation
```

### Verilog Instantiation Template

```
// BSCAN_SPARTAN3: Boundary Scan primitive for connecting internal logic to
//                  JTAG interface.
//                  Spartan-3/3E
// Xilinx HDL Libraries Guide, version 11.2

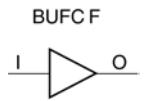
BSCAN_SPARTAN3 BSCAN_SPARTAN3_inst (
    .CAPTURE(CAPTURE), // CAPTURE output from TAP controller
    .DRCK1(DRCK1), // Data register output for USER1 functions
    .DRCK2(DRCK2), // Data register output for USER2 functions
    .RESET(RESET), // Reset output from TAP controller
    .SEL1(SEL1), // USER1 active output
    .SEL2(SEL2), // USER2 active output
    .SHIFT(SHIFT), // SHIFT output from TAP controller
    .TDI(TDI), // TDI output from TAP controller
    .UPDATE(UPDATE), // UPDATE output from TAP controller
    .TDO1(TDO1), // Data input for USER1 function
    .TDO2(TDO2) // Data input for USER2 function
);
// End of BSCAN_SPARTAN3_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## BUFCF

**Primitive:** Fast Connect Buffer



## Introduction

This design element is a single fast connect buffer used to connect the outputs of the LUTs and some dedicated logic directly to the input of another LUT. Using this buffer implies CLB packing. No more than four LUTs may be connected together as a group.

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFCF: Fast connect buffer used to connect the outputs of the LUTs
--         and some dedicated logic directly to the input of another LUT.
--         For use with all FPGAs.
-- Xilinx HDL Libraries Guide, version 11.2

BUFCF_inst: BUFCF (
port map (
    O => O, -- Connect to the output of a LUT
    I => I -- Connect to the input of a LUT
);
-- End of BUFCF_inst instantiation
```

### Verilog Instantiation Template

```
// BUFCF: Fast connect buffer used to connect the outputs of the LUTs
//         and some dedicated logic directly to the input of another LUT.
//         For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

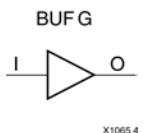
BUFCF BUFCF_inst (
    .O(O), // Connect to the output of a LUT
    .I(I) // Connect to the input of a LUT
);
// End of BUFCF_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

# BUFG

**Primitive:** Global Clock Buffer



## Introduction

This design element is a high-fanout buffer that connects signals to the global routing resources for low skew distribution of the signal. BUFGs are typically used on clock nets.

## Port Descriptions

Port	Type	Width	Function
I	Input	1	Clock buffer output
O	Output	1	Clock buffer input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFG: Global Clock Buffer
--         Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFG_inst : BUFG
generic map (
)
port map (
    O => O, -- 1-bit Clock buffer output
    I => I -- 1-bit Clock buffer input
);
-- End of BUFG_inst instantiation
```

## Verilog Instantiation Template

```
// BUFG: Global Clock Buffer (source by an internal signal)
//      All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

BUFG BUFG_inst (
    .O(O),          // Clock buffer output
    .I(I)           // Clock buffer input
);

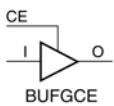
// End of BUFG_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## BUFGCE

**Primitive:** Global Clock Buffer with Clock Enable



X9384

### Introduction

This design element is a global clock buffer with a single gated input. Its O output is "0" when clock enable (CE) is Low (inactive). When clock enable (CE) is High, the I input is transferred to the O output.

### Logic Table

Inputs		Outputs
I	CE	O
X	0	0
I	1	I

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCE: Global Clock Buffer with Clock Enable (active high)
--          Virtex4/5/6, Spartan-3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

BUFGCE_inst : BUFGCE
port map (
    O => O,      -- Clock buffer ouptput
    CE => CE,    -- Clock enable input
    I => I       -- Clock buffer input
);
-- End of BUFGCE_inst instantiation
```

## Verilog Instantiation Template

```
// BUFGCE: Global Clock Buffer with Clock Enable (active high)
//           Virtex-4/5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

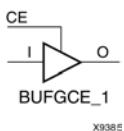
BUFGCE BUFGCE_inst (
    .O(O),      // Clock buffer output
    .CE(CE),    // Clock enable input
    .I(I)       // Clock buffer input
);
// End of BUFGCE_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## BUFGCE\_1

Primitive: Global Clock Buffer with Clock Enable and Output State 1



### Introduction

This design element is a multiplexed global clock buffer with a single gated input. Its O output is High (1) when clock enable (CE) is Low (inactive). When clock enable (CE) is High, the I input is transferred to the O output.

### Logic Table

Inputs		Outputs
I	CE	O
X	0	1
I	1	I

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
--          Virtex-4/5/6, Spartan-3/3E/3A/6
--          Xilinx HDL Libraries Guide, version 11.2

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,      -- Clock buffer output
    CE => CE,    -- Clock enable input
    I => I       -- Clock buffer input
);
-- End of BUFGCE_1_inst instantiation
```

## Verilog Instantiation Template

```
// BUFGCE_1: Global Clock Buffer with Clock Enable (active low)
//           Virtex-4/5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

BUFGCE_1 BUFGCE_1_inst (
    .O(O),      // Clock buffer output
    .CE(CE),    // Clock enable input
    .I(I)       // Clock buffer input
);
// End of BUFGCE_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## BUFGMUX

**Primitive:** Global Clock MUX Buffer

BUFGMUX



X9251

### Introduction

BUFGMUX is a multiplexed global clock buffer that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (O). When the select input (S) is High, the signal on I1 is selected for output.

BUFGMUX and BUFGMUX\_1 are distinguished by the state the output assumes when that output switches between clocks in response to a change in its select input. BUFGMUX assumes output state 0 and BUFGMUX\_1 assumes output state 1.

**Note** BUFGMUX guarantees that when S is toggled, the state of the output remains in the inactive state until the next active clock edge (either I0 or I1) occurs.

### Logic Table

Inputs			Outputs
I0	I1	S	O
I0	X	0	I0
X	I1	1	I1
X	X	↑	0
X	X	↓	0

### Port Descriptions

Port	Type	Width	Function
I0	Input	1	Clock0 input
I1	Input	1	Clock1 input
O	Output	1	Clock MUX output
S	Input	1	Clock select input

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
CLK_SEL_TYPE	String	"SYNC", "ASYNC"	"SYNC"	Specifies synchronous or asynchronous clock.
DISABLE_VALUE	String	"HIGH", "LOW"	"LOW"	Specifies the state the output assumes when switching between inputs.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGMUX: Global Clock MUX Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFGMUX_inst : BUFGMUX
generic map (
    CLK_SEL_TYPE => "SYNC",
)
port map (
    O => O,    -- 1-bit Clock MUX output
    I0 => I0,   -- 1-bit Clock0 input
    I1 => I1,   -- 1-bit Clock1 input
    S => S     -- 1-bit Clock select input
);
-- End of BUFGMUX_inst instantiation
```

## Verilog Instantiation Template

```
// BUFGMUX: Global Clock Buffer 2-to-1 MUX
//          Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

BUFGMUX BUFGMUX_inst (
    .O(O),      // Clock MUX output
    .I0(I0),    // Clock0 input
    .I1(I1),    // Clock1 input
    .S(S)       // Clock select input
);
// End of BUFGMUX_inst instantiation
```

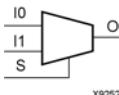
## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## BUFGMUX\_1

**Primitive:** Global Clock MUX Buffer with Output State 1

BUFGMUX\_1



X9252

### Introduction

This design element is a multiplexed global clock buffer that can select between two input clocks: I0 and I1. When the select input (S) is Low, the signal on I0 is selected for output (0). When the select input (S) is High, the signal on I1 is selected for output.

This design element is distinguished from BUFGMUX by the state the output assumes when that output switches between clocks in response to a change in its select input. BUFGMUX assumes output state 0 and BUFGMUX\_1 assumes output state 1.

### Logic Table

Inputs			Outputs
I0	I1	S	O
I0	X	0	I0
X	I1	1	I1
X	X	↑	1
X	X	↓	1

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
--          Spartan-3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

BUFGMUX_1_inst : BUFGMUX_1
port map (
    O => O,      -- Clock MUX output
    I0 => I0,     -- Clock0 input
    I1 => I1,     -- Clock1 input
    S => S       -- Clock select input
);
-- End of BUFGMUX_1_inst instantiation
```

## Verilog Instantiation Template

```
// BUFGMUX_1: Global Clock Buffer 2-to-1 MUX (inverted select)
//          Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

BUFGMUX_1 BUFGMUX_1_inst (
    .O(O),      // Clock_MUX output
    .I0(I0),    // Clock0 input
    .I1(I1),    // Clock1 input
    .S(S)       // Clock select input
);

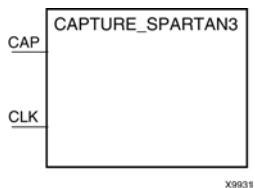
// End of BUFGMUX_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## CAPTURE\_SPARTAN3

**Primitive:** Spartan®-3 Register State Capture for Bitstream Readback



### Introduction

This element provides user control and synchronization over when and how the capture register (flip-flop and latch) information task is requested. The readback function is provided through dedicated configuration port instructions. However, without this element, the readback data is synchronized to the configuration clock. Only register (flip-flop and latch) states can be captured. Although LUT RAM, SRL, and block RAM states are readback, they cannot be captured.

An asserted high CAP signal indicates that the registers in the device are to be captured at the next Low-to-High clock transition. By default, data is captured after every trigger when transition on CLK while CAP is asserted. To limit the readback operation to a single data capture, add the ONESHOT=TRUE attribute to this element.

### Port Descriptions

Port	Direction	Width	Function
CAP	Input	1	Readback capture trigger
CLK	Input	1	Readback capture clock

### Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

Connect all inputs and outputs to the design in order to ensure proper operation.

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
ONESHOT	Boolean	TRUE, FALSE	TRUE	Specifies the procedure for performing single readback per CAP trigger.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- CAPTURE_SPARTAN3: Register State Capture for Bitstream Readback
--          Spartan-3
-- Xilinx HDL Libraries Guide, version 11.2

CAPTURE_SPARTAN3_inst : CAPTURE_SPARTAN3
port map (
    CAP => CAP,      -- Capture input
    CLK => CLK       -- Clock input
);
-- End of CAPTURE_SPARTAN3_inst instantiation
```

## Verilog Instantiation Template

```
// CAPTURE_SPARTAN3: Register State Capture for Bitstream Readback
//          Spartan-3/3E
// Xilinx HDL Libraries Guide, version 11.2

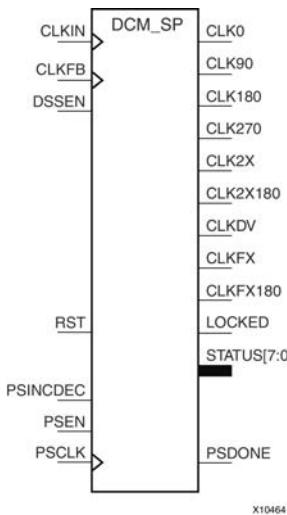
CAPTURE_SPARTAN3 CAPTURE_SPARTAN3_inst (
    .CAP(CAP),      // Capture input
    .CLK(CLK)       // Clock input
);
// End of CAPTURE_SPARTAN3_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## DCM\_SP

**Primitive:** Digital Clock Manager



## Introduction

This design element is a digital clock manager that provides multiple functions. It can implement a clock delay locked loop (DLL), a digital frequency synthesizer (DFS), and a digital phase shifter (DPS). DCM\_SPs are useful for eliminating the clock delay coming on and off the chip, shifting the clock phase to improve data capture, deriving different frequency clocks, as well as other useful clocking functions.

## Port Descriptions

Port	Type	Width	Function
CLKDV	Output	1	Divided version of CLK0. Divide value is programmable.
CLKFB	Input	1	Feedback clock input to DCM. The feedback input is required unless the DFS is used stand-alone. The source of CLKFB must be CLK0 or CLK2X output from the DCM.
CLKFX	Output	1	Digital Frequency Synthesizer output (DFS).
CLKFX180	Output	1	180 degree shifted version of the CLKFX clock.
CLKIN	Input	1	Clock input for the DCM.
CLK0	Output	1	Same frequency as CLKIN, 0 degree phase shift.
CLK2X	Output	1	Two times CLKIN frequency clock, aligned with CLK0.
CLK2X180	Output	1	180 degree shifted version of the CLK2X clock.
CLK90	Output	1	Same frequency as CLKIN, 90 degree phase shift.
CLK180	Output	1	Same frequency as CLKIN, 180 degree phase shift.
CLK270	Output	1	Same frequency as CLKIN, 180 degree phase shift.
LOCKED	Output	1	Signal indicating when the DCM has LOCKed.
PSCLK	Input	1	Phase shift clock input. The PSCLK input pin provides the source clock for the DCM phase shift.

Port	Type	Width	Function
PSDONE	Output	1	Output signal that indicates variable phase shift is done.
PSEN	Input	1	Variable Phase Shift enable signal, synchronous with PSCLK.
PSINCDEC	Input	1	The phase shift increment/decrement (PSINCDEC) input signal must be synchronous with PSCLK. The PSINCDEC signal is used to increment or decrement the phase shift factor when PSEN is activated. The PSINCDEC is asserted High for increment and Low for decrement.
RST	Input	1	The reset input pin (RST) resets the DCM circuitry. The RST signal is an active High asynchronous reset.
STATUS[7:0]	Output	8	<p>The status output bus provides DCM status.</p> <ul style="list-style-type: none"> <li>• STATUS[0] - Variable Phase Shift Overflow.</li> <li>• STATUS[1] - CLKIN Stopped.</li> <li>• STATUS[2] - CLKFX or CLKFX180 Stopped.</li> <li>• STATUS[7] - CLKFB stopped.</li> <li>• STATUS[6:3] - Reserved.</li> </ul>

## Design Entry Method

Instantiation	Yes
Inference	No
CORE Generator™ and wizards	Recommended
Macro support	No

## Available Attributes

Attribute	Type	Allowed_Values	Default	Description
CLKDV_DIVIDE	1 significant digit Float	2.0, 1.5, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0	2.0	Specifies the extent to which the CLKDLL, CLKDLLE, CLKDLLHF, or DCM_SP clock divider (CLKDV output) is to be frequency divided.
CLK_FEEDBACK	String	"1X", "2X", "NONE"	"1X"	Defines the DCM feedback mode. <ul style="list-style-type: none"> <li>• 1X: CLK0 as feedback.</li> <li>• 2X: CLK2X as feedback.</li> </ul>
CLKFX_DIVIDE	Integer	1 to 32	1	Specifies the frequency divider value for the CLKFX output.
CLKFX_MULTIPLY	Integer	2 to 32	4	Specifies the frequency multiplier value for the CLKFX output.
CLKIN_DIVIDE_BY_2	Boolean	FALSE, TRUE	FALSE	Enables CLKIN divide by two features.
CLKIN_PERIOD	String	0 bit String	"10.0"	Specifies the input period to the DCM_SP CLKIN input in ns.

Attribute	Type	Allowed_Values	Default	Description
CLKOUT_PHASE_SHIFT	String	“NONE”, “FIXED”, “VARIABLE”	“NONE”	This attribute specifies the phase shift mode. <ul style="list-style-type: none"> <li>• NONE: No phase shift capability. Any set value has no effect.</li> <li>• FIXED: DCM outputs are a fixed phase shift from CLKIN. Value is specified by PHASE_SHIFT attribute.</li> <li>• VARIABLE: Allows the DCM outputs to be shifted in a positive and negative range relative to CLKIN. Starting value is specified by PHASE_SHIFT.</li> </ul>
DESKEW_ADJUST	String	“SYSTEM_SYNCHRONOUS”, “SOURCE_SYNCHRONOUS”	“SYSTEM_SYNCHRONOUS”	Sets configuration bits affecting the clock delay alignment between the DCM_SP output clocks and an FPGA clock input pin.
DLL_FREQUENCY_MODE	String	“LOW”, “HIGH”	“LOW”	AUTO mode allows DLL to do automatic frequency search to decide whether DLL will operate in Low or High mode. This is a legacy attribute where the High and Low value has no affect, it is always in auto mode.
DSS_MODE	String	“NONE”, “SPREAD_2”, “SPREAD_4”, “SPREAD_6”, “SPREAD_8”	“NONE”	Specifies a frequency spread for output clocks. <ul style="list-style-type: none"> <li>• NONE - The default, specifies no spread factors. The digital spread spectrum function is disabled.</li> <li>• SPREAD_2 - Creates a new clock period that is +/- 50 ps of the current clock period</li> <li>• SPREAD_4 - Creates a new clock period that is +/- 100 ps of the current clock period.</li> <li>• SPREAD_6 - Creates a new clock period that is +/- 150 ps of the current clock period.</li> <li>• SPREAD_8 - Creates a new clock period that is +/- 200 ps of the current clock period.</li> </ul> <p>The spreading is cumulative as the SPREAD_# is increased. For example, SPREAD_2 creates two additional clock frequencies at +/-50 ps relative to the input clock frequency; SPREAD_4 does the same as SPREAD_2, plus it creates two additional clock frequencies at +/-100 ps.</p>
DUTY_CYCLE_CORRECTION	Boolean	TRUE, FALSE	TRUE	Corrects the duty cycle of the CLK0, CLK90, CLK180, and CLK270 outputs.
PHASE_SHIFT	Integer	-255 to 255	0	Defines the amount of fixed phase shift from -255 to 255.
STARTUP_WAIT	Boolean	FALSE, TRUE	FALSE	Delays configuration DONE until DCM LOCK.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- DCM_SP: Digital Clock Manager
--   Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

DCM_SP_inst : DCM_SP
generic map (
    CLKDV_DIVIDE => 2.0,                                -- Specifies the extent to which the CLKDLL, CLKDLLE, CLKDLLHF, or
    CLKFX_MULTIPLY => 4,                                 -- DCM_SP clock divider (CLKDV output) is to be frequency divided.
    CLKIN_DIVIDE_BY_2 => FALSE,                          -- Specifies the frequency divider value for the CLKFX output.
    CLKIN_PERIOD => "10.0",                             -- Specifies the frequency multiplier value for the CLKFX output.
    CLKOUT_PHASE_SHIFT => "NONE",                         -- Enables CLKIN divide by two features.
    CLK_FEEDBACK => "1X",                               -- Specifies the input period to the DCM_SP CLKIN input in ns.
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS",             -- This attribute specifies the phase shift mode. NONE = No phase
    DLL_FREQUENCY_MODE => "LOW",                        -- shift capability. Any set value has no effect. FIXED = DCM
    DSS_MODE => "NONE",                                -- outputs are a fixed phase shift from CLKIN. Value is specified
    DUTY_CYCLE_CORRECTION => TRUE,                      -- by PHASE_SHIFT attribute. VARIABLE = Allows the DCM outputs to
    PHASE_SHIFT => 0,                                    -- be shifted in a positive and negative range relative to CLKIN.
    STARTUP_WAIT => FALSE,                             -- Starting value is specified by PHASE_SHIFT.
)
port map (
    CLK0 => CLK0,                                     -- 1-bit Same frequency as CLKIN, 0 degree phase shift.
    CLK180 => CLK180,                                 -- 1-bit Same frequency as CLKIN, 180 degree phase shift.
    CLK270 => CLK270,                                 -- 1-bit Same frequency as CLKIN, 180 degree phase shift.
    CLK2X => CLK2X,                                  -- 1-bit Two times CLKIN frequency clock, aligned with CLK0.
    CLK2X180 => CLK2X180,                            -- 1-bit 180 degree shifted version of the CLK2X clock.
    CLK90 => CLK90,                                   -- 1-bit Same frequency as CLKIN, 90 degree phase shift.
    CLKDV => CLKDV,                                 -- 1-bit Divided version of CLK0. Divide value is programmable.
    CLKFX => CLKFX,                                 -- 1-bit Digital Frequency Synthesizer output (DFS).
    CLKFX180 => CLKFX180,                           -- 1-bit 180 degree shifted version of the CLKFX clock.
    LOCKED => LOCKED,                              -- 1-bit Signal indicating when the DCM has LOCKed.
    PSDONE => PSDONE,                             -- 1-bit Output signal that indicates variable phase shift is done.
    STATUS => STATUS,                            -- 8-bit DCM Status Bits
    CLKFB => CLKFB,                               -- 1-bit Feedback clock input to DCM. The feedback input is required unless the DFS
                                                -- is used stand-alone. The source of CLKFB must be CLK0 or CLK2X output from the
                                                -- DCM.

    CLKIN => CLKIN,                                -- 1-bit Clock input for the DCM.
    DSSEN => DSSEN,                                -- 1-bit Phase shift clock input. The PSCLK input pin provides the source clock for
    PSCLK => PSCLK,                                -- the DCM phase shift.

    PSEN => PSEN,                                 -- 1-bit Variable Phase Shift enable signal, synchronous with PSCLK.
    PSINCDEC => PSINCDEC,                         -- 1-bit The phase shift increment/decrement (PSINCDEC) input signal must be
                                                -- synchronous with PSCLK. The PSINCDEC signal is used to increment or decrement
                                                -- the phase shift factor when PSEN is activated. The PSINCDEC is asserted HIGH for
                                                -- increment and LOW for decrement.

    RST => RST,                                    -- 1-bit The reset input pin (RST) resets the DCM circuitry. The RST signal is an
                                                -- active HIGH asynchronous reset.
);
-- End of DCM_SP_inst instantiation

```

## Verilog Instantiation Template

```

// DCM_SP: Digital Clock Manager Circuit
//          Spartan-3E/3A, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

DCM_SP #(
    .CLKDV_DIVIDE(2.0), // Divide by: 1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,6.5
               //    7.0,7.5,8.0,9.0,10.0,11.0,12.0,13.0,14.0,15.0 or 16.0
    .CLKFX_DIVIDE(1), // Can be any integer from 1 to 32
    .CLKFX_MULTIPLY(4), // Can be any integer from 2 to 32
    .CLKIN_DIVIDE_BY_2("FALSE"), // TRUE/FALSE to enable CLKIN divide by two feature
    .CLKIN_PERIOD(0.0), // Specify period of input clock
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift of NONE, FIXED or VARIABLE
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE, 1X or 2X
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
               // an integer from 0 to 15
    .DLL_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 255
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_SP_inst (
    .CLK0(CLK0), // 0 degree DCM CLK output
    .CLK180(CLK180), // 180 degree DCM CLK output
    .CLK270(CLK270), // 270 degree DCM CLK output
    .CLK2X(CLK2X), // 2X DCM CLK output
    .CLK2X180(CLK2X180), // 2X, 180 degree DCM CLK out
    .CLK90(CLK90), // 90 degree DCM CLK output
    .CLKDV(CLKDV), // Divided DCM CLK out (CLKDV_DIVIDE)
    .CLKFX(CLKFX), // DCM CLK synthesis out (M/D)
    .CLKFX180(CLKFX180), // 180 degree CLK synthesis out
    .LOCKED(LOCKED), // DCM LOCK status output
    .PSDONE(PSDONE), // Dynamic phase adjust done output
    .STATUS(STATUS), // 8-bit DCM status bits output
    .CLKFB(CLKFB), // DCM clock feedback
    .CLKIN(CLKIN), // Clock input (from IBUFG, BUFG or DCM)
    .PSCLK(PSCLK), // Dynamic phase adjust clock input
    .PSEN(PSEN), // Dynamic phase adjust enable input
    .PSINCDEC(PSINCDEC), // Dynamic phase adjust increment/decrement
    .RST(RST) // DCM asynchronous reset input
);
// End of DCM_SP_inst instantiation

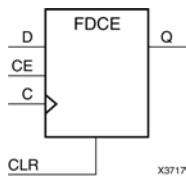
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## FDCE

Primitive: D Flip-Flop with Clock Enable and Asynchronous Clear



### Introduction

This design element is a single D-type flip-flop with clock enable and asynchronous clear. When clock enable (CE) is High and asynchronous clear (CLR) is Low, the data on the data input (D) of this design element is transferred to the corresponding data output (Q) during the Low-to-High clock (C) transition. When CLR is High, it overrides all other inputs and resets the data output (Q) Low. When CE is Low, clock transitions are ignored.

This flip-flop is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate STARTUP\_architecture symbol.

### Logic Table

Inputs				Outputs
CLR	CE	D	C	Q
1	X	X	X	0
0	0	X	X	No Change
0	1	D	↑	D

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0	0	Sets the initial value of Q output after configuration.  For Spartan®-6 devices, the INIT value should always match the polarity of the set or reset. In the case of FDCE, the INIT should be 0. If set to 1, an asynchronous circuit must be created to exhibit this behavior, which Xilinx does not recommend.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--        Clock Enable (posedge clk). All families.
-- Xilinx HDL Libraries Guide, version 11.2

FDCE_inst : FDCE
generic map (
    INIT => '0') -- Initial value of register ('0' or '1')
port map (
    Q => Q,          -- Data output
    C => C,          -- Clock input
    CE => CE,         -- Clock enable input
    CLR => CLR,       -- Asynchronous clear input
    D => D           -- Data input
);
-- End of FDCE_inst instantiation
```

## Verilog Instantiation Template

```
// FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
//        Clock Enable (posedge clk).
// All families.
// Xilinx HDL Libraries Guide, version 11.2

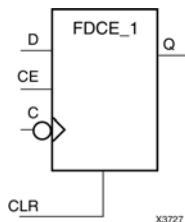
FDCE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCE_inst (
    .Q(Q),          // Data output
    .C(C),          // Clock input
    .CE(CE),         // Clock enable input
    .CLR(CLR),       // Asynchronous clear input
    .D(D)           // Data input
);
// End of FDCE_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## FDCE\_1

Primitive: D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Clear



### Introduction

This design element is a single D-type flip-flop with data (D), clock enable (CE), asynchronous clear (CLR) inputs, and data output (Q). The asynchronous CLR input, when High, overrides all other inputs and sets the Q output Low. The data on the (D) input is loaded into the flip-flop when CLR is Low and CE is High on the High-to-Low clock (C) transition. When CE is Low, the clock transitions are ignored.

This flip-flop is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate STARTUP\_architecture symbol.

### Logic Table

Inputs					Outputs
CLR	CE	D	C	Q	
1	X	X	X	0	
0	0	X	X	No Change	
0	1	D	↓	D	

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0,1	0	Sets the initial value of Q output after configuration.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCE_1: Single Data Rate D Flip-Flop with Asynchronous Clear and
--          Clock Enable (negedge clock). All families.
-- Xilinx HDL Libraries Guide, version 11.2

FDCE_1_inst : FDCE_1
generic map (
    INIT => '0' -- Initial value of register ('0' or '1')
port map (
    Q => Q,      -- Data output
    C => C,      -- Clock input
    CE => CE,    -- Clock enable input
    CLR => CLR,  -- Asynchronous clear input
    D => D       -- Data input
);
-- End of FDCE_1_inst instantiation
```

## Verilog Instantiation Template

```
// FDCE_1: Single Data Rate D Flip-Flop with Asynchronous Clear and
//          Clock Enable (negedge clock).
//          All families.
// Xilinx HDL Libraries Guide, version 11.2

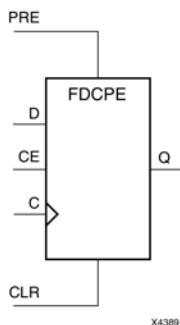
FDCE_1 #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCE_1_inst (
    .Q(Q),      // Data output
    .C(C),      // Clock input
    .CE(CE),    // Clock enable input
    .CLR(CLR),  // Asynchronous clear input
    .D(D)       // Data input
);
// End of FDCE_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## FDCPE

Primitive: D Flip-Flop with Clock Enable and Asynchronous Preset and Clear



## Introduction

This design element is a single D-type flip-flop with data (D), clock enable (CE), asynchronous preset (PRE), and asynchronous clear (CLR) inputs. The asynchronous active high PRE sets the Q output High; that active high CLR resets the output Low and has precedence over the PRE input. Data on the D input is loaded into the flip-flop when PRE and CLR are Low and CE is High on the Low-to-High clock (C) transition. When CE is Low, the clock transitions are ignored and the previous value is retained. The FDCPE is generally implemented as a slice or IOB register within the device.

For FPGA devices, upon power-up, the initial value of this component is specified by the INIT attribute. If a subsequent GSR (Global Set/Reset) is asserted, the flop is asynchronously set to the INIT value.

**Note** While this device supports the use of asynchronous set and reset, it is not generally recommended to be used for in most cases. Use of asynchronous signals pose timing issues within the design that are difficult to detect and control and also have an adverse affect on logic optimization causing a larger design that can consume more power than if a synchronous set or reset is used.

## Logic Table

Inputs						Outputs
CLR	PRE	CE	D	C		Q
1	X	X	X	X		0
0	1	X	X	X		1
0	0	0	X	X		No Change
0	0	1	D	↑		D

## Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Data output
C	Input	1	Clock input
CE	Input	1	Clock enable input
CLR	Input	1	Asynchronous clear input
D	Input	1	Data input
PRE	Input	1	Asynchronous set input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0,1	0	Sets the initial value of Q output after configuration and on GSR.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
-- Clock Enable (posedge clk).
-- Virtex-4/5, Spartan-3/3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

FDCPE_inst : FDCPE
generic map (
    INIT => '0' -- Initial value of register ('0' or '1')
port map (
    Q => Q,          -- Data output
    C => C,          -- Clock input
    CE => CE,         -- Clock enable input
    CLR => CLR,        -- Asynchronous clear input
    D => D,          -- Data input
    PRE => PRE        -- Asynchronous set input
);
-- End of FDCPE_inst instantiation
```

### Verilog Instantiation Template

```
// FDCPE: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
// Clock Enable (posedge clk).
// Virtex-4/5, Spartan-3/3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

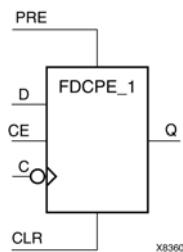
FDCPE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCPE_inst (
    .Q(Q),          // Data output
    .C(C),          // Clock input
    .CE(CE),         // Clock enable input
    .CLR(CLR),        // Asynchronous clear input
    .D(D),          // Data input
    .PRE(PRE)        // Asynchronous set input
);
// End of FDCPE_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## FDCPE\_1

Primitive: D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Preset and Clear



### Introduction

FDCPE\_1 is a single D-type flip-flop with data (D), clock enable (CE), asynchronous preset (PRE), and asynchronous clear (CLR) inputs and data output (Q). The asynchronous PRE, when High, sets the (Q) output High; CLR, when High, resets the output Low. Data on the (D) input is loaded into the flip-flop when PRE and CLR are Low and CE is High on the High-to-Low clock (C) transition. When CE is Low, the clock transitions are ignored.

This flip-flop is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate STARTUP\_architecture symbol.

### Logic Table

Inputs					Outputs
CLR	PRE	CE	D	C	Q
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	No Change
0	0	1	D	↓	D

### Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Data output
C	Input	1	Clock input
CE	Input	1	Clock enable input
CLR	Input	1	Asynchronous clear input
D	Input	1	Data input
PRE	Input	1	Asynchronous set input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0,1	0	Sets the initial value of Q output after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDCPE_1: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
--           Clock Enable (negedge clock).
--           Virtex-4/5, Spartan-3/3E/3A/3A DSP
--           Xilinx HDL Libraries Guide, version 11.2

FDCPE_1_inst : FDCPE_1
generic map (
    INIT => '0' -- Initial value of register ('0' or '1')
port map (
    Q => Q,          -- Data output
    C => C,          -- Clock input
    CE => CE,         -- Clock enable input
    CLR => CLR,        -- Asynchronous clear input
    D => D,          -- Data input
    PRE => PRE        -- Asynchronous set input
);
-- End of FDCPE_1_inst instantiation
```

### Verilog Instantiation Template

```
// FDCPE_1: Single Data Rate D Flip-Flop with Asynchronous Clear, Set and
//           Clock Enable (negedge clock).
//           Virtex-4/5, Spartan-3/3E/3A/3A DSP
//           Xilinx HDL Libraries Guide, version 11.2

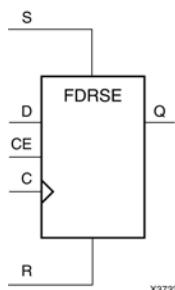
FDCPE_1 #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDCPE_1_inst (
    .Q(Q),          // Data output
    .C(C),          // Clock input
    .CE(CE),         // Clock enable input
    .CLR(CLR),        // Asynchronous clear input
    .D(D),          // Data input
    .PRE(PRE)        // Asynchronous set input
);
// End of FDCPE_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## FDRSE

Primitive: D Flip-Flop with Synchronous Reset and Set and Clock Enable



X3732

### Introduction

FDRSE is a single D-type flip-flop with synchronous reset (R), synchronous set (S), clock enable (CE) inputs. The reset (R) input, when High, overrides all other inputs and resets the Q output Low during the Low-to-High clock transition. (Reset has precedence over Set.) When the set (S) input is High and R is Low, the flip-flop is set, output High, during the Low-to-High clock (C) transition. Data on the D input is loaded into the flip-flop when R and S are Low and CE is High during the Low-to-High clock transition.

Upon power-up, the initial value of this component is specified by the INIT attribute. If a subsequent GSR (Global Set/Reset) is asserted, the flop is asynchronously set to the INIT value.

### Logic Table

Inputs					Outputs
R	S	CE	D	C	Q
1	X	X	X	↑	0
0	1	X	X	↑	1
0	0	0	X	X	No Change
0	0	1	1	↑	1
0	0	1	0	↑	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0, 1	0	Sets the initial value of Q output after configuration and on GSR.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
--          Clock Enable (posedge clk).
--          Virtex-4/5, Spartan-3/3E/3A/DSP
--          Xilinx HDL Libraries Guide, version 11.2

FDRSE_inst : FDRSE
generic map (
    INIT => '0') -- Initial value of register ('0' or '1')
port map (
    Q => Q,          -- Data output
    C => C,          -- Clock input
    CE => CE,        -- Clock enable input
    D => D,          -- Data input
    R => R,          -- Synchronous reset input
    S => S           -- Synchronous set input
);
-- End of FDRSE_inst instantiation
```

## Verilog Instantiation Template

```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//          Clock Enable (posedge clk).
//          Virtex-4/5, Spartan-3/3E/3A/DSP
//          Xilinx HDL Libraries Guide, version 11.2

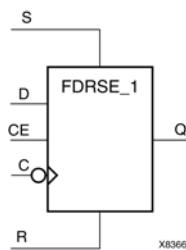
FDRSE #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDRSE_inst (
    .Q(Q),          // Data output
    .C(C),          // Clock input
    .CE(CE),        // Clock enable input
    .D(D),          // Data input
    .R(R),          // Synchronous reset input
    .S(S)           // Synchronous set input
);
// End of FDRSE_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## FDRSE\_1

Primitive: D Flip-Flop with Negative-Clock Edge, Synchronous Reset and Set, and Clock Enable



### Introduction

FDRSE\_1 is a single D-type flip-flop with synchronous reset (R), synchronous set (S), and clock enable (CE) inputs and data output (Q). The reset (R) input, when High, overrides all other inputs and resets the (Q) output Low during the High-to-Low clock transition. (Reset has precedence over Set.) When the set (S) input is High and R is Low, the flip-flop is set, output High, during the High-to-Low clock (C) transition. Data on the (D) input is loaded into the flip-flop when (R) and (S) are Low and (CE) is High during the High-to-Low clock transition.

This flip-flop is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate STARTUP\_architecture symbol.

### Logic Table

Inputs					Outputs
R	S	CE	D	C	Q
1	X	X	X	↓	0
0	1	X	X	↓	1
0	0	0	X	X	No Change
0	0	1	D	↓	D

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary	0, 1	0	Sets the initial value of Q output after configuration and on GSR.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- FDRSE_1: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
--           Clock Enable (negedge clock).
--           Virtex-4/5, Spartan-3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

FDRSE_1_inst : FDRSE_1
generic map (
    INIT => '0') -- Initial value of register ('0' or '1')
port map (
    Q => Q,          -- Data output
    C => C,          -- Clock input
    CE => CE,        -- Clock enable input
    D => D,          -- Data input
    R => R,          -- Synchronous reset input
    S => S,          -- Synchronous set input
);
-- End of FDRSE_1_inst instantiation
```

## Verilog Instantiation Template

```
// FDRSE_1: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//           Clock Enable (negedge clock).
//           Virtex-4/5, Spartan-3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

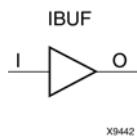
FDRSE_1 #(
    .INIT(1'b0) // Initial value of register (1'b0 or 1'b1)
) FDRSE_1_inst (
    .Q(Q),          // Data output
    .C(C),          // Clock input
    .CE(CE),        // Clock enable input
    .D(D),          // Data input
    .R(R),          // Synchronous reset input
    .S(S)           // Synchronous set input
);
// End of FDRSE_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## IBUF

**Primitive:** Input Buffer



## Introduction

This design element is automatically inserted (inferred) by the synthesis tool to any signal directly connected to a top-level input or in-out port of the design. You should generally let the synthesis tool infer this buffer. However, it can be instantiated into the design if required. In order to do so, connect the input port (I) directly to the associated top-level input or in-out port, and connect the output port (O) to the logic sourced by that port. Modify any necessary generic maps (VHDL) or named parameter value assignment (Verilog) in order to change the default behavior of the component.

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Buffer output
I	Input	1	Buffer input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

In general, this element is inferred by the synthesis tool for any specified top-level input port to the design. It is generally not necessary to specify them in the source code. However, if desired, they be manually instantiated by either copying the instantiation code from the appropriate Libraries Guide HDL template and pasting it into the top-level entity/module of your code. It is recommended to always put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level input port of the design and the O port to the logic in which this input is to source. Specify the desired generic/defparam values in order to configure the proper behavior of the buffer.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
IBUF_DELAY_VALUE	String	"0" through "12"	"0"	Specifies the amount of additional delay to add to the non-registered path out of the IOB.
IFD_DELAY_VALUE	String	"AUTO", "0" through "6"	"AUTO"	Specifies the amount of additional delay to add to the registered path within the IOB.
IOSTANDARD	String	See Data Sheet.	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUF: Single-ended Input Buffer
--      All devices
-- Xilinx HDL Libraries Guide, version 11.2

IBUF_inst : IBUF
generic map (
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
    -- "0"--"12" (Spartan-3E)
    -- "0"--"16" (Spartan-3A)
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
    -- "AUTO", "0"--"6" (Spartan-3E)
    -- "AUTO", "0"--"8" (Spartan-3A)
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Buffer output
    I => I      -- Buffer input (connect directly to top-level port)
);
-- End of IBUF_inst instantiation
```

## Verilog Instantiation Template

```
// IBUF: Single-ended Input Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 11.2

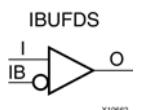
IBUF #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for
    // the buffer: "0"--"12" (Spartan-3E)
    //           "0"--"16" (Spartan-3A)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input
    // register: "AUTO", "0"--"6" (Spartan-3E)
    //           "AUTO", "0"--"8" (Spartan-3A)
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
)IBUF_inst (
    .O(O),      // Buffer output
    .I(I)        // Buffer input (connect directly to top-level port)
);
// End of IBUF_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## IBUFDS

**Primitive:** Differential Signaling Input Buffer



## Introduction

This design element is an input buffer that supports low-voltage, differential signaling. In IBUFDS, a design level interface signal is represented as two distinct ports (I and IB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET\_P and MYNET\_N). Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components.

## Logic Table

Inputs		Outputs
I	IB	O
0	0	No Change
0	1	0
1	0	1
1	1	No Change

## Port Descriptions

Port	Type	Width	Function
I	Input	1	Diff_p Buffer Input
IB	Input	1	Diff_n Buffer Input
O	Output	1	Buffer Output

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

Put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level "master" input port of the design, the IB port to the top-level "slave" input port, and the O port to the logic in which this input is to source. Specify the desired generic/defparam values in order to configure the proper behavior of the buffer.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Enables the built-in differential termination resistor.
IBUF_DELAY_VALUE	String	"0" through "12"	"0"	Specifies the amount of additional delay to add to the non-registered path out of the IOB.
IFD_DELAY_VALUE	String	"AUTO", "0" thru "6"	"AUTO"	Specifies the amount of additional delay to add to the registered path within the IOB.
IOSTANDARD	String	See Data Sheet.	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFDS: Differential Input Buffer
--          Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

IBUFDS_inst : IBUFDS
generic map (
    CAPACITANCE => "DONT_CARE", -- "LOW", "NORMAL", "DONT_CARE" (Virtex-4 only)
    DIFF_TERM => FALSE, -- Differential Termination (Virtex-4/5, Spartan-3E/3A)
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
                           -- "0"--"12" (Spartan-3E)
                           -- "0"--"16" (Spartan-3A)
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
                               -- "AUTO", "0"--"6" (Spartan-3E)
                               -- "AUTO", "0"--"8" (Spartan-3A)
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Clock buffer output
    I => I, -- Diff_p clock buffer input (connect directly to top-level port)
    IB => IB -- Diff_n clock buffer input (connect directly to top-level port)
);
-- End of IBUFDS_inst instantiation
```

## Verilog Instantiation Template

```
// IBUFDS: Differential Input Buffer
//          Virtex-4/5, Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 11.2

IBUFDS #(
    .CAPACITANCE("DONT_CARE"), // "LOW", "NORMAL", "DONT_CARE" (Virtex-4 only)
    .DIFF_TERM("FALSE"), // Differential Termination (Virtex-4/5, Spartan-3E/3A)
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for
                           // the buffer: "0"--"12" (Spartan-3E)
                           // "0"--"16" (Spartan-3A)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input
                           // register: "AUTO", "0"--"6" (Spartan-3E)
                           // "AUTO", "0"--"8" (Spartan-3A)
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFDS_inst (
    .O(O), // Buffer output
    .I(I), // Diff_p buffer input (connect directly to top-level port)
    .IB(IB) // Diff_n buffer input (connect directly to top-level port)
);

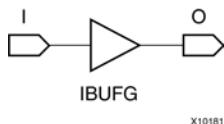
// End of IBUFDS_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## IBUFG

**Primitive:** Dedicated Input Clock Buffer



## Introduction

The IBUFG is a dedicated input to the device which should be used to connect incoming clocks to the FPGA's global clock routing resources. The IBUFG provides dedicated connections to the DCM\_SP and BUFG providing the minimum amount of clock delay and jitter to the device. The IBUFG input can only be driven by the global clock pins. The IBUFG output can drive CLKIN of a DCM\_SP, BUFG, or your choice of logic.

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Clock Buffer output
I	Input	1	Clock Buffer input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
IBUF_DELAY_VALUE	String	"0" through "12"	"0"	Specifies the amount of additional delay to add to the non-registered path out of the IOB
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFG: Global Clock Buffer (sourced by an external pin)
-- Xilinx HDL Libraries Guide, version 11.2

IBUFG_inst : IBUFG
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O, -- Clock buffer output
    I => I -- Clock buffer input (connect directly to top-level port)
);
-- End of IBUFG_inst instantiation
```

## Verilog Instantiation Template

```
// IBUFG: Global Clock Buffer (sourced by an external pin)
//          All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

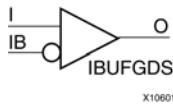
IBUFG #(
    .IOSTANDARD("DEFAULT"),
    .IBUF_DELAY_VALUE("0") // Specify the amount of added input delay for
                           //   the buffer: "0"- "12" (Spartan-3E)
                           //   "0"- "16" (Spartan-3A)
) IBUFG_inst (
    .O(O), // Clock buffer output
    .I(I)  // Clock buffer input (connect directly to top-level port)
);
// End of IBUFG_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## IBUFGDS

**Primitive:** Differential Signaling Dedicated Input Clock Buffer and Optional Delay



X10601

## Introduction

This design element is a dedicated differential signaling input buffer for connection to the clock buffer (BUFG) or . In IBUFGDS, a design-level interface signal is represented as two distinct ports (I and IB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET\_P and MYNET\_N). Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components. Also available is a programmable delay to assist in the capturing of incoming data to the device.

## Logic Table

Inputs		Outputs
I	IB	O
0	0	No Change
0	1	0
1	0	1
1	1	No Change

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Clock Buffer output
IB	Input	1	Diff_n Clock Buffer Input
I	Input	1	Diff_p Clock Buffer Input

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

Put all I/O components on the top-level of the design to help facilitate hierarchical design methods. Connect the I port directly to the top-level "master" input port of the design, the IB port to the top-level "slave" input port and the O port to , BUFG or logic in which this input is to source. Some synthesis tools infer the BUFG automatically if necessary, when connecting an IBUFG to the clock resources of the FPGA. Specify the desired generic/defparam values in order to configure the proper behavior of the buffer.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DIFF_TERM	Boolean	TRUE or FALSE	FALSE	Enables the built-in differential termination resistor.
IBUF_DELAY_VALUE	String	"0" through "12"	"0"	Specifies the amount of additional delay to add to the non-registered path out of the IOB
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IBUFGDS: Differential Global Clock Buffer (sourced by an external pin)
--           Virtex-4/5, Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

IBUFGDS_inst : IBUFGDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,    -- Clock buffer output
    I => I,    -- Diff_p clock buffer input
    IB => IB -- Diff_n clock buffer input
);
-- End of IBUFGDS_inst instantiation
```

## Verilog Instantiation Template

```
// IBUFGDS: Differential Global Clock Buffer (sourced by an external pin)
//           Virtex-4/5, Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 11.2

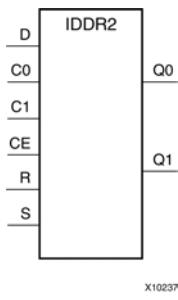
IBUFGDS #(
    .DIFF_TERM("FALSE"), // Differential Termination (Virtex-4/5, Spartan-3E/3A)
    .IOSTANDARD("DEFAULT") // Specifies the I/O standard for this buffer
    .IBUF_DELAY_VALUE("0") // Specify the amount of added input delay for
                           //   the buffer: "0"-"12" (Spartan-3E)
                           //   "0"-"16" (Spartan-3A)
) IBUFGDS_inst (
    .O(O), // Clock buffer output
    .I(I), // Diff_p clock buffer input
    .IB(IB) // Diff_n clock buffer input
);
// End of IBUFGDS_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## IDDR2

**Primitive: Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset**



## Introduction

This design element is a dedicated input register designed to receive external dual data rate (DDR) signals into Xilinx® FPGAs. The IDDR2 requires two clocks to be connected to the component, C0 and C1, so that data is captured at the positive edge of both C0 and C1 clocks. The IDDR2 features an active high clock enable port, CE, which can be used to suspend the operation of the registers, and both set and reset ports that can be configured to be synchronous or asynchronous to the respective clocks. The IDDR2 has an optional alignment feature that allows both output data ports to be aligned to a single clock.

## Logic Table

Input						Output	
S	R	CE	D	C0	C1	Q0	Q1
1	x	x	x	x	x	INIT_Q0	INIT_Q1
0	1	x	x	x	x	not INIT_Q0	not INIT_Q1
0	0	0	x	x	x	No Change	No Change
0	0	1	D	↑	x	D	No Change
0	0	1	D	x	↑	No Change	D
Set/Reset can be synchronous via SRTYPE value							

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

To change the default behavior of the IDDR2, modify attributes via the generic map (VHDL) or named parameter value assignment (Verilog) as a part of the instantiated component. The IDDR2 can be connected directly to a top-level input port in the design, where an appropriate input buffer can be inferred, or directly to an instantiated IBUF, IOBUF, IBUFDS or IOBUFDS. All inputs and outputs of this component should either be connected or properly tied off.

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DDR_ALIGNMENT	String	NONE, "C0", "C1"	"NONE"	Sets the output alignment more for the DDR register <ul style="list-style-type: none"> <li>• NONE - Makes the data available on the Q0 and Q1 outputs shortly after the corresponding C0 or C1 positive clock edge.</li> <li>• C0 – Makes the data on both Q0 and Q1 align to the positive edge of the C0 clock.</li> <li>• C1 - Makes the data on both Q0 and Q1 align to the positive edge of the C1 clock.</li> </ul>
INIT_Q0	Integer	0, 1	0	Sets initial state of the Q0 output to 0 or 1.
INIT_Q1	Integer	0, 1	0	Sets initial state of the Q1 output to 0 or 1.
SRTYPE	String	"SYNC", "ASYNC"	"SYNC"	Specifies SYNC" or "ASYNC" set/reset.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IDDR2: Input Double Data Rate Input Register with Set, Reset
--         and Clock Enable. Spartan-3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

IDDR2_inst : IDDR2
generic map(
    DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
    INIT_Q0 => '0', -- Sets initial state of the Q0 output to '0' or '1'
    INIT_Q1 => '0', -- Sets initial state of the Q1 output to '0' or '1'
    SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYNC" set/reset
port map (
    Q0 => Q0, -- 1-bit output captured with C0 clock
    Q1 => Q1, -- 1-bit output captured with C1 clock
    C0 => C0, -- 1-bit clock input
    C1 => C1, -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    D => D, -- 1-bit data input
    R => R, -- 1-bit reset input
    S => S -- 1-bit set input
);
-- End of IDDR2_inst instantiation
```

## Verilog Instantiation Template

```
// IDDR2: Input Double Data Rate Input Register with Set, Reset
//         and Clock Enable.
//         Spartan-3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

IDDR2 #(
    .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
    .INIT_Q0(1'b0), // Sets initial state of the Q0 output to 1'b0 or 1'b1
    .INIT_Q1(1'b0), // Sets initial state of the Q1 output to 1'b0 or 1'b1
    .SRTYPE("SYNC") // Specifies "SYNC" or "ASYNC" set/reset
) IDDR2_inst (
    .Q0(Q0), // 1-bit output captured with C0 clock
    .Q1(Q1), // 1-bit output captured with C1 clock
    .C0(C0), // 1-bit clock input
    .C1(C1), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D(D), // 1-bit DDR data input
    .R(R), // 1-bit reset input
```

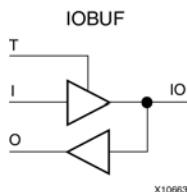
```
.S(S)      // 1-bit set input  
);  
// End of IDDR2_inst instantiation
```

## For More Information

See the [Spartan-3 Generation FPGA User Guide](#).

## IOBUF

**Primitive:** Bi-Directional Buffer



X10663

## Introduction

The design element is a bidirectional single-ended I/O Buffer used to connect internal logic to an external bidirectional pin.

## Logic Table

Inputs		Bidirectional	Outputs
T	I	IO	O
1	X	Z	X
0	1	1	1
0	0	0	0

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Buffer output
IO	Inout	1	Buffer inout
I	Input	1	Buffer input
T	Input	1	3-State enable input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DRIVE	Integer	2, 4, 6, 8, 12, 16, 24	12	Selects output drive strength (mA) for the SelectIO™ buffers that use the LVTTL, LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25, or LVCMOS33 interface I/O standard.
IBUF_DELAY_VALUE	String	"0" through "12"	"0"	Specifies the amount of additional delay to add to the non-registered path out of the IOB
IFD_DELAY_VALUE	String	"AUTO", "0" through "6"	"AUTO"	Specifies the amount of additional delay to add to the registered path within the IOB
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW", "FAST", "QUIETIO"	"SLOW"	Sets the output rise and fall time. See the Data Sheet for recommendations of the best setting for this attribute.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUF: Single-ended Bi-directional Buffer
-- All devices
-- Xilinx HDL Libraries Guide, version 11.2

IOBUF_inst : IOBUF
generic map (
    DRIVE => 12,
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
    -- "0"--"12" (Spartan-3E)
    -- "0"--"16" (Spartan-3A)
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
    -- "AUTO", "0"--"6" (Spartan-3E)
    -- "AUTO", "0"--"8" (Spartan-3A)
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output
    IO => IO,    -- Buffer inout port (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T      -- 3-state enable input, high=input, low=output
);
-- End of IOBUF_inst instantiation
```

## Verilog Instantiation Template

```
// IOBUF: Single-ended Bi-directional Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 11.2

IOBUF #(
    .DRIVE(12), // Specify the output drive strength
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for the buffer,
                           // "0"-"12" (Spartan-3E only), "0"-"16" (Spartan-3A only)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input register,
                           // "AUTO", "0"-"6" (Spartan-3E only), "0"-"8" (Spartan-3A only)
    .IOSTANDARD("DEFAULT"), // Specify the I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) IOBUF_inst (
    .O(O), // Buffer output
    .IO(IO), // Buffer inout port (connect directly to top-level port)
    .I(I), // Buffer input
    .T(T) // 3-state enable input, high=input, low=output
);

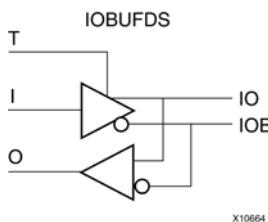
// End of IOBUF_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## IOBUFDS

**Primitive:** 3-State Differential Signaling I/O Buffer with Active Low Output Enable



## Introduction

The design element is a bidirectional buffer that supports low-voltage, differential signaling. For the IOBUFDS, a design level interface signal is represented as two distinct ports (IO and IOB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET\_P and MYNET\_N). Optionally, a programmable differential termination feature is available to help improve signal integrity and reduce external components. Also available is a programmable delay is to assist in the capturing of incoming data to the device.

## Logic Table

Inputs		Bidirectional		Outputs
I	T	IO	IOB	O
X	1	Z	Z	No Change
0	0	0	1	0
I	0	1	0	1

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Buffer output
IO	Inout	1	Diff_p inout
IOB	Inout	1	Diff_n inout
I	Input	1	Buffer input
T	Input	1	3-state enable input

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
IBUF_DELAY_VALUE	String	"0" through "12"	"0"	Specifies the amount of additional delay to add to the non-registered path out of the IOB
IFD_DELAY_VALUE	String	"AUTO", "0" through "6"	"AUTO"	Specifies the amount of additional delay to add to the registered path within the IOB
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- IOBUFDS: Differential Bi-directional Buffer
--          Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

IOBUFDS_inst : IOBUFDS
generic map (
    IBUF_DELAY_VALUE => "0", -- Specify the amount of added input delay for buffer,
    -- "0"--"12" (Spartan-3E)
    -- "0"--"16" (Spartan-3A)
    IFD_DELAY_VALUE => "AUTO", -- Specify the amount of added delay for input register,
    -- "AUTO", "0"--"6" (Spartan-3E)
    -- "AUTO", "0"--"8" (Spartan-3A)
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Buffer output
    IO => IO,     -- Diff_p inout (connect directly to top-level port)
    IOB => IOB,   -- Diff_n inout (connect directly to top-level port)
    I => I,       -- Buffer input
    T => T       -- 3-state enable input, high=input, low=output
);
-- End of IOBUFDS_inst instantiation
```

## Verilog Instantiation Template

```
// IOBUFDS: Differential Bi-directional Buffer
//          Virtex-4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 11.2

IOBUFDS #(
    .IBUF_DELAY_VALUE("0"), // Specify the amount of added input delay for the buffer,
    // "0"--"12" (Spartan-3E only), "0"--"16" (Spartan-3A only)
    .IFD_DELAY_VALUE("AUTO"), // Specify the amount of added delay for input register,
    // "AUTO", "0"--"6" (Spartan-3E only), "0"--"8" (Spartan-3A only)
    .IOSTANDARD("DEFAULT") // Specify the I/O standard
) IOBUFDS_inst (
    .O(O),      // Buffer output
    .IO(IO),    // Diff_p inout (connect directly to top-level port)
    .IOB(IOB),  // Diff_n inout (connect directly to top-level port)
    .I(I),       // Buffer input
    .T(T)        // 3-state enable input, high=input, low=output
);
// End of IOBUFDS_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

# KEEPER

Primitive: KEEPER Symbol



X10669

## Introduction

The design element is a weak keeper element that retains the value of the net connected to its bidirectional O pin. For example, if a logic 1 is being driven onto the net, KEEPER drives a weak/resistive 1 onto the net. If the net driver is then 3-stated, KEEPER continues to drive a weak/resistive 1 onto the net.

## Port Descriptions

Name	Direction	Width	Function
O	Output	1-Bit	Keeper output

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- KEEPER: I/O Buffer Weak Keeper
--          All FPGA, CoolRunner-II
-- Xilinx HDL Libraries Guide, version 11.2

KEEPER_inst : KEEPER
port map (
    O => O      -- Keeper output (connect directly to top-level port)
);
-- End of KEEPER_inst instantiation
```

## Verilog Instantiation Template

```
// KEEPER: I/O Buffer Weak Keeper
//          All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 11.2

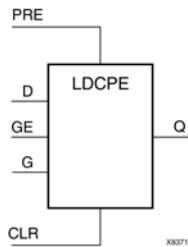
KEEPER KEEPER_inst (
    .O(O)      // Keeper output (connect directly to top-level port)
);
// End of KEEPER_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## LDCPE

Primitive: Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable



## Introduction

This design element is a transparent data latch with data (D), asynchronous clear (CLR), asynchronous preset (PRE), and gate enable (GE). When (CLR) is High, it overrides the other inputs and resets the data (Q) output Low. When (PRE) is High and (CLR) is Low, it presets the data (Q) output High. Q reflects the data (D) input while the gate (G) input and gate enable (GE) are High and (CLR) and PRE are Low. The data on the (D) input during the High-to-Low gate transition is stored in the latch. The data on the Q output remains unchanged as long as (G) or (GE) remains Low.

This latch is asynchronously cleared, outputs Low, when power is applied. For FPGA devices, power-on conditions are simulated when global set/reset (GSR) is active. GSR defaults to active-High but can be inverted by adding an inverter in front of the GSR input of the appropriate STARTUP\_architecture symbol.

## Logic Table

Inputs					Outputs
CLR	PRE	GE	G	D	Q
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	No Change
0	0	1	1	0	0
0	0	1	1	1	1
0	0	1	0	X	No Change
0	0	1	↓	D	D

## Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Data Output
CLR	Input	1	Asynchronous clear/reset input
D	Input	1	Data Input
G	Input	1	Gate Input
GE	Input	1	Gate Enable Input
PRE	Input	1	Asynchronous preset/set input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Integer	0, 1	0	Sets the initial value of Q output after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LDCPE: Transparent latch with Asynchronous Reset, Preset and
-- Gate Enable.
-- Virtex-4/5, Spartan-3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

LDCPE_inst : LDCPE
generic map (
    INIT => '0' -- Initial value of latch ('0' or '1')
port map (
    Q => Q,          -- Data output
    CLR => CLR,      -- Asynchronous clear/reset input
    D => D,          -- Data input
    G => G,          -- Gate input
    GE => GE,        -- Gate enable input
    PRE => PRE       -- Asynchronous preset/set input
);
-- End of LDCPE_inst instantiation
```

### Verilog Instantiation Template

```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
// Gate Enable.
// Virtex-4/5, Spartan-3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

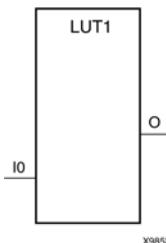
LDCPE #(
    .INIT(1'b0) // Initial value of latch (1'b0 or 1'b1)
) LDCPE_inst (
    .Q(Q),          // Data output
    .CLR(CLR),      // Asynchronous clear/reset input
    .D(D),          // Data input
    .G(G),          // Gate input
    .GE(GE),        // Gate enable input
    .PRE(PRE)       // Asynchronous preset/set input
);
// End of LDCPE_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT1

Primitive: 1-Bit Look-Up Table with General Output



## Introduction

This design element is a 1-bit look-up table (LUT) with general output (O).

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs	Outputs
I0	O
0	INIT[0]
1	INIT[1]

INIT = Binary number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 2-Bit Value	All zeros	Initializes look-up tables.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT1: 1-input Look-Up Table with general output
-- Xilinx HDL Libraries Guide, version 11.2

LUT1_inst : LUT1
generic map (
    INIT => "00")
port map (
    O => O,    -- LUT general output
    I0 => I0   -- LUT input
);
-- End of LUT1_inst instantiation
```

### Verilog Instantiation Template

```
// LUT1: 1-input Look-Up Table with general output
//      For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

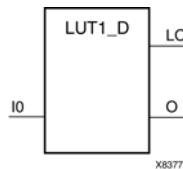
LUT1 #(
    .INIT(2'b00) // Specify LUT Contents
) LUT1_inst (
    .O(O),    // LUT general output
    .I0(I0)   // LUT input
);
// End of LUT1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT1\_D

Primitive: 1-Bit Look-Up Table with Dual Output



## Introduction

This design element is a 1-bit look-up table (LUT) with two functionally identical outputs, O and LO. It provides a look-up table version of a buffer or inverter.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs	Outputs	
I0	O	LO
0	INIT[0]	INIT[0]
1	INIT[1]	INIT[1]

INIT = Binary number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 2-Bit Value	All zeros	Initializes look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT1_D: 1-input Look-Up Table with general and local outputs
-- Xilinx HDL Libraries Guide, version 11.2

LUT1_D_inst : LUT1_D
generic map (
    INIT => "00")
port map (
    LO => LO, -- LUT local output
    O => O,   -- LUT general output
    I0 => I0  -- LUT input
);
-- End of LUT1_D_inst instantiation
```

## Verilog Instantiation Template

```
// LUT1_D: 1-input Look-Up Table with general and local outputs
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

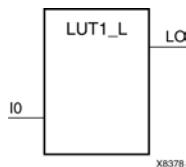
LUT1_D #(
    .INIT(2'b00) // Specify LUT Contents
) LUT1_D_inst (
    .LO(LO), // LUT local output
    .O(O),   // LUT general output
    .I0(I0)  // LUT input
);
// End of LUT1_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT1\_L

Primitive: 1-Bit Look-Up Table with Local Output



## Introduction

This design element is a 1-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs	Outputs
I0	LO
0	INIT[0]
1	INIT[1]
INIT = Binary number assigned to the INIT attribute	

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 2-Bit Value	All zeros	Initializes look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT1_L: 1-input Look-Up Table with local output
-- Xilinx HDL Libraries Guide, version 11.2

LUT1_L_inst : LUT1_L
generic map (
    INIT => "00")
port map (
    LO => LO, -- LUT local output
    I0 => I0 -- LUT input
);
-- End of LUT1_L_inst instantiation
```

## Verilog Instantiation Template

```
// LUT1_L: 1-input Look-Up Table with local output
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

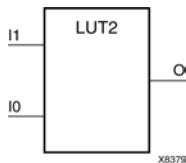
LUT1_L #(
    .INIT(2'b00) // Specify LUT Contents
) LUT1_L_inst (
    .LO(LO), // LUT local output
    .I0(I0) // LUT input
);
// End of LUT1_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT2

**Primitive:** 2-Bit Look-Up Table with General Output



### Introduction

This design element is a 2-bit look-up table (LUT) with general output (O).

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

### Logic Table

Inputs		Outputs
I1	I0	O
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 4-Bit Value	All zeros	Initializes look-up tables.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2: 2-input Look-Up Table with general output
-- Xilinx HDL Libraries Guide, version 11.2

LUT2_inst : LUT2
generic map (
    INIT => X"0")
port map (
    O => O,    -- LUT general output
    I0 => I0,   -- LUT input
    I1 => I1   -- LUT input
);
-- End of LUT2_inst instantiation
```

### Verilog Instantiation Template

```
// LUT2: 2-input Look-Up Table with general output
// For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

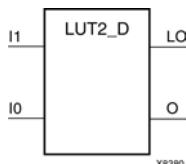
LUT2 #(
    .INIT(4'h0)  // Specify LUT Contents
) LUT2_inst (
    .O(O),    // LUT general output
    .I0(I0),   // LUT input
    .I1(I1)   // LUT input
);
// End of LUT2_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT2\_D

**Primitive:** 2-Bit Look-Up Table with Dual Output



## Introduction

This design element is a 2-bit look-up table (LUT) with two functionally identical outputs, O and LO.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The LogicTable Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs		Outputs	
I1	I0	O	LO
0	0	INIT[0]	INIT[0]
0	1	INIT[1]	INIT[1]
1	0	INIT[2]	INIT[2]
1	1	INIT[3]	INIT[3]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 4-Bit Value	All zeros	Initializes look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2_D: 2-input Look-Up Table with general and local outputs
-- Xilinx HDL Libraries Guide, version 11.2

LUT2_D_inst : LUT2_D
generic map (
    INIT => X"0")
port map (
    LO => LO, -- LUT local output
    O => O,   -- LUT general output
    I0 => I0, -- LUT input
    I1 => I1 -- LUT input
);
-- End of LUT2_D_inst instantiation
```

## Verilog Instantiation Template

```
// LUT2_D: 2-input Look-Up Table with general and local outputs
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

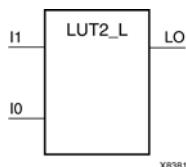
LUT2_D #(
    .INIT(4'h0) // Specify LUT Contents
) LUT2_D_inst (
    .LO(LO), // LUT local output
    .O(O),   // LUT general output
    .I0(I0), // LUT input
    .I1(I1) // LUT input
);
// End of LUT2_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT2\_L

**Primitive:** 2-Bit Look-Up Table with Local Output



## Introduction

This design element is a 2-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs		Outputs
I1	I0	LO
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 4-Bit Value	All zeros	Initializes look-up tables.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT2_L: 2-input Look-Up Table with local output
-- Xilinx HDL Libraries Guide, version 11.2

LUT2_L_inst : LUT2_L
generic map (
    INIT => X"0")
port map (
    LO => LO, -- LUT local output
    I0 => I0, -- LUT input
    I1 => I1 -- LUT input
);
-- End of LUT2_L_inst instantiation
```

### Verilog Instantiation Template

```
// LUT2_L: 2-input Look-Up Table with local output
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

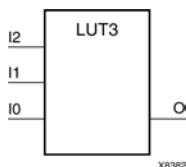
LUT2_L #(
    .INIT(4'h0) // Specify LUT Contents
) LUT2_L_inst (
    .LO(LO), // LUT local output
    .I0(I0), // LUT input
    .I1(I1) // LUT input
);
// End of LUT2_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT3

**Primitive:** 3-Bit Look-Up Table with General Output



## Introduction

This design element is a 3-bit look-up table (LUT) with general output (O). A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs			Outputs
I2	I1	I0	O
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 8-Bit Value	All zeros	Initializes look-up tables.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT3: 3-input Look-Up Table with general output
-- Xilinx HDL Libraries Guide, version 11.2

LUT3_inst : LUT3
generic map (
    INIT => X"00")
port map (
    O => O,    -- LUT general output
    I0 => I0,   -- LUT input
    I1 => I1,   -- LUT input
    I2 => I2   -- LUT input
);
-- End of LUT3_inst instantiation
```

### Verilog Instantiation Template

```
// LUT3: 3-input Look-Up Table with general output
// For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

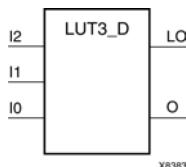
LUT3 #(
    .INIT(8'h00) // Specify LUT Contents
) LUT3_inst (
    .O(O),    // LUT general output
    .I0(I0),   // LUT input
    .I1(I1),   // LUT input
    .I2(I2)    // LUT input
);
// End of LUT3_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT3\_D

**Primitive:** 3-Bit Look-Up Table with Dual Output



### Introduction

This design element is a 3-bit look-up table (LUT) with two functionally identical outputs, O and LO.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

### Logic Table

Inputs			Outputs	
I2	I1	I0	O	LO
0	0	0	INIT[0]	INIT[0]
0	0	1	INIT[1]	INIT[1]
0	1	0	INIT[2]	INIT[2]
0	1	1	INIT[3]	INIT[3]
1	0	0	INIT[4]	INIT[4]
1	0	1	INIT[5]	INIT[5]
1	1	0	INIT[6]	INIT[6]
1	1	1	INIT[7]	INIT[7]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 8-Bit Value	All zeros	Initializes look-up tables.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT3_D: 3-input Look-Up Table with general and local outputs
-- Xilinx HDL Libraries Guide, version 11.2

LUT3_D_inst : LUT3_D
generic map (
    INIT => X"00")
port map (
    LO => LO,    -- LUT local output
    O => O,      -- LUT general output
    I0 => I0,    -- LUT input
    I1 => I1,    -- LUT input
    I2 => I2    -- LUT input
);
-- End of LUT3_D_inst instantiation
```

### Verilog Instantiation Template

```
// LUT3_D: 3-input Look-Up Table with general and local outputs
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

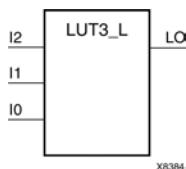
LUT3_D #(
    .INIT(8'h00) // Specify LUT Contents
) LUT3_D_inst (
    .LO(LO), // LUT local output
    .O(O),   // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2)  // LUT input
);
// End of LUT3_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT3\_L

Primitive: 3-Bit Look-Up Table with Local Output



## Introduction

This design element is a 3-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs			Outputs
I2	I1	I0	LO
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 8-Bit Value	All zeros	Initializes look-up tables.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT3_L: 3-input Look-Up Table with local output
-- Xilinx HDL Libraries Guide, version 11.2

LUT3_L_inst : LUT3_L
generic map (
    INIT => X"00")
port map (
    LO => LO,    -- LUT local output
    I0 => I0,    -- LUT input
    I1 => I1,    -- LUT input
    I2 => I2    -- LUT input
);
-- End of LUT3_L_inst instantiation
```

### Verilog Instantiation Template

```
// LUT3_L: 3-input Look-Up Table with local output
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

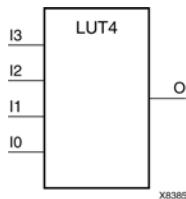
LUT3_L #(
    .INIT(8'h00) // Specify LUT Contents
) LUT3_L_inst (
    .LO(LO), // LUT local output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2) // LUT input
);
// End of LUT3_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT4

Primitive: 4-Bit Look-Up-Table with General Output



### Introduction

This design element is a 4-bit look-up table (LUT) with general output (O).

An INIT attribute with an appropriate number of hexadecimal digits for the number of inputs must be attached to the LUT to specify its function. This element provides a look-up table version of a buffer or inverter. These elements are the basic building blocks. Two LUTs are available in each CLB slice; four LUTs are available in each CLB. Multiple variants of LUTs accommodate additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs				Outputs
I3	I2	I1	I0	O
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Initializes look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4: 4-input Look-Up Table with general output
-- Xilinx HDL Libraries Guide, version 11.2

LUT4_inst : LUT4
generic map (
    INIT => X"0000")
port map (
    O => O,      -- LUT general output
    I0 => I0,    -- LUT input
    I1 => I1,    -- LUT input
    I2 => I2,    -- LUT input
    I3 => I3    -- LUT input
);
-- End of LUT4_inst instantiation
```

## Verilog Instantiation Template

```
// LUT4: 4-input Look-Up Table with general output
//      For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

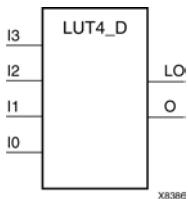
LUT4 #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_inst (
    .O(O),      // LUT general output
    .I0(I0),    // LUT input
    .I1(I1),    // LUT input
    .I2(I2),    // LUT input
    .I3(I3)    // LUT input
);
// End of LUT4_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## LUT4\_D

Primitive: 4-Bit Look-Up Table with Dual Output



### Introduction

This design element is a 4-bit look-up table (LUT) with two functionally identical outputs, O and LO.

The O output is a general interconnect. The LO output is used to connect to another output within the same CLB slice and to the fast connect buffer. A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and is more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs				Outputs	
I3	I2	I1	I0	O	LO
0	0	0	0	INIT[0]	INIT[0]
0	0	0	1	INIT[1]	INIT[1]
0	0	1	0	INIT[2]	INIT[2]
0	0	1	1	INIT[3]	INIT[3]
0	1	0	0	INIT[4]	INIT[4]
0	1	0	1	INIT[5]	INIT[5]
0	1	1	0	INIT[6]	INIT[6]
0	1	1	1	INIT[7]	INIT[7]
1	0	0	0	INIT[8]	INIT[8]
1	0	0	1	INIT[9]	INIT[9]
1	0	1	0	INIT[10]	INIT[10]
1	0	1	1	INIT[11]	INIT[11]
1	1	0	0	INIT[12]	INIT[12]
1	1	0	1	INIT[13]	INIT[13]
1	1	1	0	INIT[14]	INIT[14]
1	1	1	1	INIT[15]	INIT[15]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Initializes look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_D: 4-input Look-Up Table with general and local outputs
-- Xilinx HDL Libraries Guide, version 11.2

LUT4_D_inst : LUT4_D
generic map (
    INIT => X"0000")
port map (
    LO => LO, -- LUT local output
    O => O, -- LUT general output
    I0 => I0, -- LUT input
    I1 => I1, -- LUT input
    I2 => I2, -- LUT input
    I3 => I3 -- LUT input
);
-- End of LUT4_D_inst instantiation
```

## Verilog Instantiation Template

```
// LUT4_D: 4-input Look-Up Table with general and local outputs
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

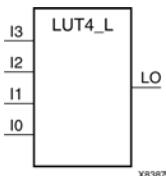
LUT4_D #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_D_inst (
    .LO(LO), // LUT local output
    .O(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3) // LUT input
);
// End of LUT4_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## LUT4\_L

Primitive: 4-Bit Look-Up Table with Local Output



## Introduction

This design element is a 4-bit look-up table (LUT) with a local output (LO) that is used to connect to another output within the same CLB slice and to the fast connect buffer. It provides a look-up table version of a buffer or inverter.

A mandatory INIT attribute, with an appropriate number of hexadecimal digits for the number of inputs, must be attached to the LUT to specify its function.

The INIT parameter for the FPGA LUT primitive is what gives the LUT its logical value. By default, this value is zero, thus driving the output to a zero regardless of the input values (acting as a ground). However, in most cases a new INIT value must be determined in order to specify the logic function for the LUT primitive. There are at least two methods by which the LUT value can be determined:

**The Logic Table Method** -A common method to determine the desired INIT value for a LUT is using a logic table. To do so, simply create a binary logic table of all possible inputs, specify the desired logic value of the output and then create the INIT string from those output values.

**The Equation Method** -Another method to determine the LUT value is to define parameters for each input to the LUT that correspond to their listed truth value and use those to build the logic equation you are after. This method is easier to understand once you have grasped the concept and more self-documenting than the above method. However, this method does require the code to first specify the appropriate parameters.

## Logic Table

Inputs				Outputs
I3	I2	I1	I0	LO
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]

INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Initializes look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- LUT4_L: 4-input Look-Up Table with local output
-- Xilinx HDL Libraries Guide, version 11.2

LUT4_L_inst : LUT4_L
generic map (
    INIT => X"0000")
port map (
    LO => LO, -- LUT local output
    I0 => I0, -- LUT input
    I1 => I1, -- LUT input
    I2 => I2, -- LUT input
    I3 => I3 -- LUT input
);
-- End of LUT4_L_inst instantiation
```

## Verilog Instantiation Template

```
// LUT4_L: 4-input Look-Up Table with local output
//          For use with all FPGAs.
// Xilinx HDL Libraries Guide, version 11.2

LUT4_L #(
    .INIT(16'h0000) // Specify LUT Contents
) LUT4_L_inst (
    .LO(LO), // LUT local output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3) // LUT input
);
// End of LUT4_L_inst instantiation
```

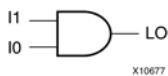
## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## MULT\_AND

**Primitive:** Fast Multiplier AND

MULT\_AND



## Introduction

The design element is an AND component located within the slice where the two inputs are shared with the 4-input LUT and the output drives into the carry logic. This added logic is especially useful for building fast and smaller multipliers. However, it can be used for other purposes as well. The I1 and I0 inputs must be connected to the I1 and I0 inputs of the associated LUT. The LO output must be connected to the DI input of the associated MUXCY, MUXCY\_D, or MUXCY\_L.

## Logic Table

Inputs		Outputs
I1	I0	LO
0	0	0
0	1	0
1	0	0
1	1	1

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT_AND: 2-input AND gate connected to Carry chain
--           For use with Virtex-4, Spartan3/3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

MULT_AND_inst : MULT_AND
port map (
    LO => LO,    -- MULT_AND output (connect to MUXCY DI)
    I0 => I0,    -- MULT_AND data[0] input
    I1 => I1    -- MULT_AND data[1] input
);
-- End of MULT_AND_inst instantiation
```

## Verilog Instantiation Template

```
// MULT_AND: 2-input AND gate connected to Carry chain
//           For use with Virtex-4, Spartan-3/3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

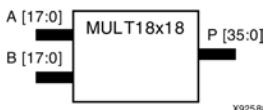
MULT_AND MULT_AND_inst (
    .LO(LO),      // MULT_AND output (connect to MUXCY DI)
    .I0(I0),      // MULT_AND data[0] input
    .I1(I1)       // MULT_AND data[1] input
);
// End of MULT_AND_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MULT18X18

**Primitive:** 18 x 18 Signed Multiplier



## Introduction

MULT18X18 is a combinational signed 18-bit by 18-bit multiplier. The value represented in the 18-bit input A is multiplied by the value represented in the 18-bit input B. Output P is the 36-bit product of A and B.

## Logic Table

Inputs	Output	
A	B	P
A	B	A x B
A, B, and P are two's complement.		

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT18X18: 18 x 18 signed asynchronous multiplier
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 11.2

MULT18X18_inst : MULT18X18
port map (
    P => P,      -- 36-bit multiplier output
    A => A,      -- 18-bit multiplier input
    B => B       -- 18-bit multiplier input
);
-- End of MULT18X18_inst instantiation
```

## Verilog Instantiation Template

```
// MULT18X18: 18 x 18 signed asynchronous multiplier
//           Spartan-3
// Xilinx HDL Libraries Guide, version 11.2

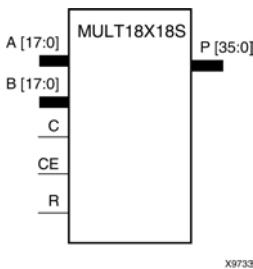
MULT18X18 MULT18X18_inst (
    .P(P),      // 36-bit multiplier output
    .A(A),      // 18-bit multiplier input
    .B(B)       // 18-bit multiplier input
);
// End of MULT18X18_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MULT18X18S

Primitive: 18 x 18 Signed Multiplier – Registered Version



### Introduction

MULT18X18S is the registered version of the 18 x 18 signed multiplier with output P and inputs A, B, C, CE, and R. The registers are initialized to 0 after the GSR pulse.

The value represented in the 18-bit input A is multiplied by the value represented in the 18-bit input B. Output P is the 36-bit product of A and B.

### Logic Table

Inputs					Output
C	CE	Am	Bn	R	P
↑	X	X	X	1	0
↑	1	Am	Bn	0	A × B
X	0	X	X	0	No Change
A, B, and P are two's complement.					

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT18X18S: 18 x 18 signed synchronous multiplier
-- Spartan-3
-- Xilinx HDL Libraries Guide, version 11.2

MULT18X18S_inst : MULT18X18S
port map (
    P => P,      -- 36-bit multiplier output
    A => A,      -- 18-bit multiplier input
    B => B,      -- 18-bit multiplier input
    C => C,      -- Clock input
    CE => CE,    -- Clock enable input
    R => R       -- Synchronous reset input
);
-- End of MULT18X18S_inst instantiation
```

## Verilog Instantiation Template

```
// MULT18X18S: 18 x 18 signed synchronous multiplier
// Spartan-3
// Xilinx HDL Libraries Guide, version 11.2

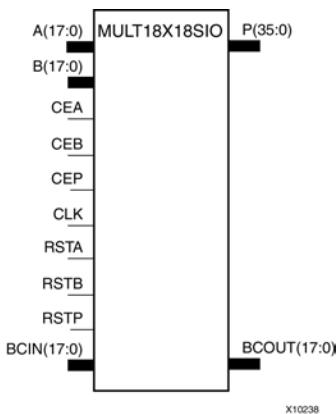
MULT18X18S MULT18X18S_inst (
    .P(P),      // 36-bit multiplier output
    .A(A),      // 18-bit multiplier input
    .B(B),      // 18-bit multiplier input
    .C(C),      // Clock input
    .CE(CE),    // Clock enable input
    .R(R)       // Synchronous reset input
);
// End of MULT18X18S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MULT18X18SIO

**Primitive: 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset**



## Introduction

This design element is a 36-bit output, 18x18-bit input dedicated signed multiplier. This component can perform asynchronous multiplication operations when the attributes AREG, BREG and PREG are all set to 0. Alternatively, synchronous multiplication operations of different latency and performance characteristics can be performed when any combination of those attributes is set to 1. When using the multiplier in synchronous operation, the MULT18X18SIO features active high clock enables for each set of register banks in the multiplier, CEA, CEB and CEP, as well as synchronous resets, RSTA, RSTB, and RSTP. Multiple MULT18X18SIOs can be cascaded to create larger multiplication functions using the BCIN and BCOUT ports in combination with the B\_INPUT attribute.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
AREG	Integer	0, 1	1	Specifies the use of the input registers on the A port. A zero disables the use of the register; a one enables the register.
BREG	Integer	0, 1	1	Specifies the use of the input registers on the B port. A zero disables the use of the register; a one enables the register.
B_INPUT	String	"DIRECT" or "CASCADE"	"DIRECT"	Specifies whether the B port is connected to the general FPGA fabric, "DIRECT" or is connected to the BCOUT port of another MULT18X18SIO.
PREG	Integer	0, 1	1	Specifies the use of the output registers of the multiplier. A zero disables the use of the register; a one enables the register.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MULT18X18SIO: 18 x 18 cascadable, signed synchronous/asynchronous multiplier
-- Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

MULT18X18SIO_inst : MULT18X18SIO
generic map (
    AREG => 1, -- Enable the input registers on the A port (1=on, 0=off)
    BREG => 1, -- Enable the input registers on the B port (1=on, 0=off)
    B_INPUT => "DIRECT", -- B cascade input "DIRECT" or "CASCADE"
    PREG => 1) -- Enable the input registers on the P port (1=on, 0=off)
port map (
    BCOUT => BCOUT, -- 18-bit cascade output
    P => P, -- 36-bit multiplier output
    A => A, -- 18-bit multiplier input
    B => B, -- 18-bit multiplier input
    BCIN => BCIN, -- 18-bit cascade input
    CEA => CEA, -- Clock enable input for the A port
    CEB => CEB, -- Clock enable input for the B port
    CEP => CEP, -- Clock enable input for the P port
    CLK => CLK, -- Clock input
    RSTA => RSTA, -- Synchronous reset input for the A port
    RSTB => RSTB, -- Synchronous reset input for the B port
    RSTP => RSTP, -- Synchronous reset input for the P port
);
-- End of MULT18X18SIO_inst instantiation
```

## Verilog Instantiation Template

```
// MULT18X18SIO: 18 x 18 cascadable, signed synchronous/asynchronous multiplier
// Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 11.2

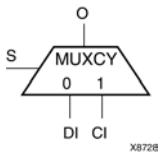
MULT18X18SIO #(
    .AREG(1), // Enable the input registers on the A port (1=on, 0=off)
    .BREG(1), // Enable the input registers on the B port (1=on, 0=off)
    .B_INPUT("DIRECT"), // B cascade input "DIRECT" or "CASCADE"
    .PREG(1) // Enable the input registers on the P port (1=on, 0=off)
) MULT18X18SIO_inst (
    .BCOUT(BCOUT), // 18-bit cascade output
    .P(P), // 36-bit multiplier output
    .A(A), // 18-bit multiplier input
    .B(B), // 18-bit multiplier input
    .BCIN(BCIN), // 18-bit cascade input
    .CEA(CEA), // Clock enable input for the A port
    .CEB(CEB), // Clock enable input for the B port
    .CEP(CEP), // Clock enable input for the P port
    .CLK(CLK), // Clock input
    .RSTA(RSTA), // Synchronous reset input for the A port
    .RSTB(RSTB), // Synchronous reset input for the B port
    .RSTP(RSTP) // Synchronous reset input for the P port
);
// End of MULT18X18SIO_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXCY

**Primitive:** 2-to-1 Multiplexer for Carry Logic with General Output



## Introduction

The direct input (DI) of a slice is connected to the (DI) input of the MUXCY. The carry in (CI) input of an LC is connected to the CI input of the MUXCY. The select input (S) of the MUXCY is driven by the output of the look-up table (LUT) and configured as a MUX function. The carry out (O) of the MUXCY reflects the state of the selected input and implements the carry out function of each LC. When Low, S selects DI; when High, S selects CI.

The variants “MUXCY\_D” and “MUXCY\_L” provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

## Logic Table

Inputs			Outputs
S	DI	CI	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXCY: Carry-Chain MUX with general output
-- Xilinx HDL Libraries Guide, version 11.2

MUXCY_inst : MUXCY
port map (
    O => O,      -- Carry output signal
    CI => CI,    -- Carry input signal
    DI => DI,    -- Data input signal
    S => S       -- MUX select, tie to '1' or LUT4 out
);
-- End of MUXCY_inst instantiation
```

## Verilog Instantiation Template

```
// MUXCY: Carry-Chain MUX with general output
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

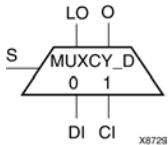
MUXCY MUXCY_inst (
    .O(O),      // Carry output signal
    .CI(CI),    // Carry input signal
    .DI(DI),    // Data input signal
    .S(S)       // MUX select, tie to '1' or LUT4 out
);
// End of MUXCY_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXCY\_D

**Primitive:** 2-to-1 Multiplexer for Carry Logic with Dual Output



### Introduction

This design element implements a 1-bit, high-speed carry propagate function. One such function can be implemented per logic cell (LC), for a total of 4-bits per configurable logic block (CLB). The direct input (DI) of an LC is connected to the DI input of the MUXCY\_D. The carry in (CI) input of an LC is connected to the CI input of the MUXCY\_D. The select input (S) of the MUX is driven by the output of the look-up table (LUT) and configured as an XOR function. The carry out (O and LO) of the MUXCY\_D reflects the state of the selected input and implements the carry out function of each LC. When Low, S selects DI; when High, S selects CI.

Outputs O and LO are functionally identical. The O output is a general interconnect. See also "MUXCY" and "MUXCY\_L".

### Logic Table

Inputs			Outputs	
S	DI	CI	O	LO
0	1	X	1	1
0	0	X	0	0
1	X	1	1	1
1	X	0	0	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXCY_D: Carry-Chain MUX with general and local outputs
-- Xilinx HDL Libraries Guide, version 11.2

MUXCY_D_inst : MUXCY_D
port map (
    LO => LO, -- Carry local output signal
    O => O,   -- Carry general output signal
    CI => CI, -- Carry input signal
    DI => DI, -- Data input signal
    S => S    -- MUX select, tie to '1' or LUT4 out
);
-- End of MUXCY_D_inst instantiation
```

## Verilog Instantiation Template

```
// MUXCY_D: Carry-Chain MUX with general and local outputs
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

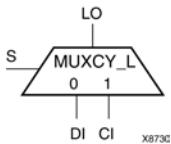
MUXCY_D MUXCY_D_inst (
    .LO(LO), // Carry local output signal
    .O(O),   // Carry general output signal
    .CI(CI), // Carry input signal
    .DI(DI), // Data input signal
    .S(S)    // MUX select, tie to '1' or LUT4 out
);
// End of MUXCY_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXCY\_L

**Primitive:** 2-to-1 Multiplexer for Carry Logic with Local Output



### Introduction

This design element implements a 1-bit high-speed carry propagate function. One such function is implemented per logic cell (LC), for a total of 4-bits per configurable logic block (CLB). The direct input (DI) of an LC is connected to the DI input of the MUXCY\_L. The carry in (CI) input of an LC is connected to the CI input of the MUXCY\_L. The select input (S) of the MUXCY\_L is driven by the output of the look-up table (LUT) and configured as an XOR function. The carry out (LO) of the MUXCY\_L reflects the state of the selected input and implements the carry out function of each (LC). When Low, (S) selects DI; when High, (S) selects (CI).

See also “MUXCY” and “MUXCY\_D.”

### Logic Table

Inputs			Outputs
S	DI	CI	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXCY_L: Carry-Chain MUX with local output
-- Xilinx HDL Libraries Guide, version 11.2

MUXCY_L_inst : MUXCY_L
port map (
    LO => LO, -- Carry local output signal
    CI => CI, -- Carry input signal
    DI => DI, -- Data input signal
    S => S    -- MUX select, tie to '1' or LUT4 out
);
-- End of MUXCY_L_inst instantiation
```

## Verilog Instantiation Template

```
// MUXCY_L: Carry-Chain MUX with local output
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

MUXCY_L MUXCY_L_inst (
    .LO(LO), // Carry local output signal
    .CI(CI), // Carry input signal
    .DI(DI), // Data input signal
    .S(S)    // MUX select, tie to '1' or LUT4 out
);

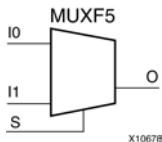
// End of MUXCY_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF5

**Primitive:** 2-to-1 Look-Up Table Multiplexer with General Output



### Introduction

This design element provides a multiplexer function in a CLB slice for creating a function-of-5 look-up table or a 4-to-1 multiplexer in combination with the associated look-up tables. The local outputs (LO) from the two look-up tables are connected to the I0 and I1 inputs of the MUXF5. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, “MUXF5\_D” and “MUXF5\_L”, provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

### Logic Table

Inputs			Outputs
S	I0	I1	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF5: Slice MUX to tie two LUT4's together with general output
-- For use with Virtex-4, Spartan3/3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

MUXF5_inst : MUXF5
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,     -- Input (tie directly to the output of LUT4)
    I1 => I1,     -- Input (tie directly to the output of LUT4)
    S => S       -- Input select to MUX
);
-- End of MUXF5_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF5: Slice MUX to tie two LUT4's together with general output
//          For use with Virtex-4, Spartan-3/3E/3A/DSP
// Xilinx HDL Libraries Guide, version 11.2

MUXF5 MUXF5_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie directly to the output of LUT4)
    .I1(I1),    // Input (tie directly to the output of LUT4)
    .S(S)       // Input select to MUX
);

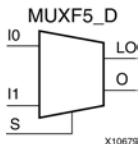
// End of MUXF5_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF5\_D

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Dual Output



### Introduction

This design element provides a multiplexer function in a CLB slice for creating a function-of-5 look-up table or a 4-to-1 multiplexer in combination with the associated look-up tables. The local outputs (LO) from the two look-up tables are connected to the I0 and I1 inputs of the MUXF5. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice. See also "MUXF5" and "MUXF5\_L".

### Logic Table

Inputs			Outputs	
S	I0	I1	O	LO
0	1	X	1	1
0	0	X	0	0
1	X	1	1	1
1	X	0	0	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
--           For use with Virtex-4, Spartan3/3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

MUXF5_D_inst : MUXF5_D
port map (
    LO => LO,    -- Output of MUX to local routing
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie directly to the output of LUT4)
    I1 => I1,    -- Input (tie directoy to the output of LUT4)
    S => S      -- Input select to MUX
);
-- End of MUXF5_D_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
//          For use with Virtex-4, Spartan-3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

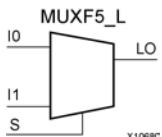
MUXF5_D MUXF5_D_inst (
    .LO(LO), // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .I0(I0), // Input (tie directly to the output of LUT4)
    .I1(I1), // Input (tie directoy to the output of LUT4)
    .S(S)    // Input select to MUX
);
// End of MUXF5_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF5\_L

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Local Output



### Introduction

This design element provides a multiplexer function in a CLB slice for creating a function-of-5 look-up table or a 4-to-1 multiplexer in combination with the associated look-up tables. The local outputs (LO) from the two look-up tables are connected to the I0 and I1 inputs of the MUXF5. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

See also “MUXF5” and “MUXF5\_D”.

### Logic Table

Inputs			Output
S	I0	I1	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF5_L: Slice MUX to tie two LUT4's together with local output
--           For use with Virtex-4, Spartan3/3E/3A/3A DSP
--           Xilinx HDL Libraries Guide, version 11.2

MUXF5_L_inst : MUXF5_L
port map (
    LO => LO,    -- Output of MUX to local routing
    I0 => I0,    -- Input (tie directly to the output of LUT4)
    I1 => I1,    -- Input (tie directly to the output of LUT4)
    S => S      -- Input select to MUX
);
-- End of MUXF5_L_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF5_L: Slice MUX to tie two LUT4's together with local output
//           For use with Virtex-4, Spartan-3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

MUXF5_L MUXF5_L_inst (
    .LO(LO), // Output of MUX to local routing
    .I0(I0), // Input (tie directly to the output of LUT4)
    .I1(I1), // Input (tie directly to the output of LUT4)
    .S(S)    // Input select to MUX
);

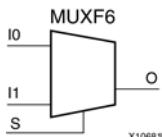
// End of MUXF5_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF6

**Primitive:** 2-to-1 Look-Up Table Multiplexer with General Output



## Introduction

This design element provides a multiplexer function in two slices for creating a function-of-6 look-up table or an 8-to-1 multiplexer in combination with the associated four look-up tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, "MUXF6\_D" and "MUXF6\_L", provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

## Logic Table

Inputs			Outputs
S	I0	I1	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF6: CLB MUX to tie two MUXF5's together with general output
--          For use with Virtex-4, Spartan3/3E/3A/3A DSP
--          Xilinx HDL Libraries Guide, version 11.2

MUXF6_inst : MUXF6
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF5 LO out)
    I1 => I1,    -- Input (tie to MUXF5 LO out)
    S => S      -- Input select to MUX
);
-- End of MUXF6_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF6: CLB MUX to tie two MUXF5's together with general output
//          For use with Virtex-4, Spartan-3/3E/3A/DSP
// Xilinx HDL Libraries Guide, version 11.2

MUXF6 MUXF6_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF5 LO out)
    .I1(I1),    // Input (tie to MUXF5 LO out)
    .S(S)       // Input select to MUX
);

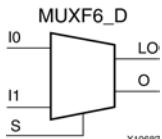
// End of MUXF6_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF6\_D

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Dual Output



### Introduction

This design element provides a multiplexer function in a two slices for creating a function-of-6 look-up table or an 8-to-1 multiplexer in combination with the associated four look-up tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice.

### Logic Table

Inputs			Outputs	
S	I0	I1	O	LO
0	1	X	1	1
0	0	X	0	0
1	X	1	1	1
1	X	0	0	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
--          For use with Virtex-4, Spartan3/3E/3A/3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

MUXF6_D_inst : MUXF6_D
port map (
    LO => LO,    -- Output of MUX to local routing
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF5 LO out)
    I1 => I1,    -- Input (tie to MUXF5 LO out)
    S => S      -- Input select to MUX
);
-- End of MUXF6_D_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF6_D: CLB MUX to tie two MUXF5's together with general and local outputs
//          For use with Virtex-4, Spartan-3E/3A/DSP
// Xilinx HDL Libraries Guide, version 11.2

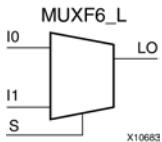
MUXF6_D MUXF6_D_inst (
    .LO(LO), // Output of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .I0(I0), // Input (tie to MUXF5 LO out)
    .I1(I1), // Input (tie to MUXF5 LO out)
    .S(S)    // Input select to MUX
);
// End of MUXF6_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF6\_L

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Local Output



## Introduction

This design element provides a multiplexer function for use in creating a function-of-6 look-up table or an 8-to-1 multiplexer in combination with the associated four look-up tables and two MUXF5s. The local outputs (LO) from the two MUXF5s in the CLB are connected to the I0 and I1 inputs of the MUXF6. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

## Logic Table

Inputs			Output
S	I0	I1	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF6_L: CLB MUX to tie two MUXF5's together with local output
--          For use with Virtex-4, Spartan3/3E/3A/3A DSP
--          Xilinx HDL Libraries Guide, version 11.2

MUXF6_L_inst : MUXF6_L
port map (
    LO => LO,    -- Output of MUX to local routing
    I0 => I0,    -- Input (tie to MUXF5 LO out)
    I1 => I1,    -- Input (tie to MUXF5 LO out)
    S => S      -- Input select to MUX
);
-- End of MUXF6_L_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF6_L: CLB MUX to tie two MUXF5's together with local output
//           For use with Virtex-4, Spartan-3E/3A/3A DSP
// Xilinx HDL Libraries Guide, version 11.2

MUXF6_L MUXF6_L_inst (
    .LO(LO), // Output of MUX to local routing
    .I0(I0), // Input (tie to MUXF5 LO out)
    .I1(I1), // Input (tie to MUXF5 LO out)
    .S(S)    // Input select to MUX
);

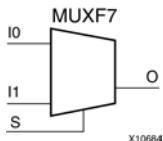
// End of MUXF6_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF7

**Primitive:** 2-to-1 Look-Up Table Multiplexer with General Output



X10684

### Introduction

This design element provides a multiplexer function for use in creating a function-of-7 look-up table or an 8-to-1 multiplexer in combination with the associated look-up tables. Local outputs (LO) of MUXF6 are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The variants, “MUXF7\_D” and “MUXF7\_L”, provide additional types of outputs that can be used by different timing models for more accurate pre-layout timing estimation.

### Logic Table

Inputs			Outputs
S	I0	I1	O
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### Port Descriptions

Port	Direction	Width	Function
O	Output	1	Output of MUX to general routing
I0	Input	1	Input (tie to MUXF6 LO out)
I1	Input	1	Input (tie to MUXF6 LO out)
S	Input	1	Input select to MUX

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF7: CLB MUX to tie two MUXF6's together with general output
--          For use with all FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

MUXF7_inst : MUXF7
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S      -- Input select to MUX
);
-- End of MUXF7_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF7: CLB MUX to tie two LUT6's or MUXF6's together with general output
//          For use with all FPGAs
// Xilinx HDL Libraries Guide, version 11.2

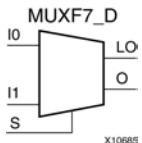
MUXF7 MUXF7_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1),    // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)       // Input select to MUX
);
// End of MUXF7_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF7\_D

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Dual Output



### Introduction

This design element provides a multiplexer function for use in creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with the associated look-up tables. Local outputs (LO) of MUXF6 are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice.

### Logic Table

Inputs			Outputs	
S	I0	I1	O	LO
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

### Port Descriptions

Port	Direction	Width	Function
O	Output	1	Output of MUX to general routing
LO	Output	1	Output of MUX to local routing
I0	Input	1	Input (tie to MUXF6 LO out)
I1	Input	1	Input (tie to MUXF6 LO out)
S	Input	1	Input select to MUX

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF7_D: CLB MUX to tie two MUXF6's together with general and local outputs
--          For use with all FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,    -- Ouptut of MUX to local routing
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF6 LO out or LUT6 06 pin)
    I1 => I1,    -- Input (tie to MUXF6 LO out or LUT6 06 pin)
    S => S      -- Input select to MUX
);
-- End of MUXF7_D_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF7_D: CLB MUX to tie two LUT6's or MUXF6's together with general and local outputs
//          For use with all FPGAs
// Xilinx HDL Libraries Guide, version 11.2

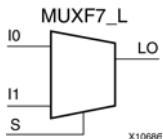
MUXF7_D MUXF7_D_inst (
    .LO(LO),    // Ouptut of MUX to local routing
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF6 LO out or LUT6 06 pin)
    .I1(I1),    // Input (tie to MUXF6 LO out or LUT6 06 pin)
    .S(S)       // Input select to MUX
);
// End of MUXF7_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF7\_L

**Primitive:** 2-to-1 look-up table Multiplexer with Local Output



### Introduction

This design element provides a multiplexer function for use in creating a function-of-7 look-up table or a 16-to-1 multiplexer in combination with the associated look-up tables. Local outputs (LO) of MUXF6 are connected to the I0 and I1 inputs of the MUXF7. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

### Logic Table

Inputs			Output
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### Port Descriptions

Port	Direction	Width	Function
LO	Output	1	Output of MUX to local routing
I0	Input	1	Input
I1	Input	1	Input
S	Input	1	Input select to MUX

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF7_L: CLB MUX to tie two MUXF6's together with local output
--           For use with all FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

MUXF7_L_inst : MUXF7_L
port map (
    LO => LO,    -- Output of MUX to local routing
    I0 => I0,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,    -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S      -- Input select to MUX
);
-- End of MUXF7_L_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF7_L: CLB MUX to tie two LUT6's or MUXF6's together with local output
//           For use with all FPGAs
// Xilinx HDL Libraries Guide, version 11.2

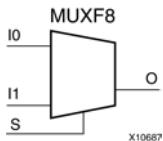
MUXF7_L#(MUXF7_L) MUXF7_L_inst (
    .LO(LO), // Output of MUX to local routing
    .I0(I0), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)    // Input select to MUX
);
// End of MUXF7_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF8

**Primitive:** 2-to-1 Look-Up Table Multiplexer with General Output



## Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 16-to-1 multiplexer in combination with the associated look-up tables, MUXF5s, MUXF6s, and MUXF7s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

## Logic Table

Inputs			Outputs
S	I0	I1	O
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Output of MUX to general routing
I0	Input	1	Input (tie to MUXF7 LO out)
I1	Input	1	Input (tie to MUXF7 LO out)
S	Input	1	Input select to MUX

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF8: CLB MUX to tie two MUXF7's together with general output
--          For use with all FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

MUXF8_inst : MUXF8
port map (
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF7 LO out)
    I1 => I1,    -- Input (tie to MUXF7 LO out)
    S => S      -- Input select to MUX
);
-- End of MUXF8_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF8: CLB MUX to tie two MUXF7's together with general output
//          For use with all FPGAs
// Xilinx HDL Libraries Guide, version 11.2

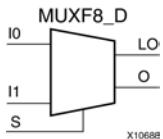
MUXF8 MUXF8_inst (
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF7 LO out)
    .I1(I1),    // Input (tie to MUXF7 LO out)
    .S(S)       // Input select to MUX
);
// End of MUXF8_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF8\_D

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Dual Output



## Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 32-to-1 multiplexer in combination with the associated four look-up tables and two MUXF8s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

Outputs O and LO are functionally identical. The O output is a general interconnect. The LO output connects to other inputs in the same CLB slice.

## Logic Table

Inputs			Outputs	
S	I0	I1	O	LO
0	I0	X	I0	I0
1	X	I1	I1	I1
X	0	0	0	0
X	1	1	1	1

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Output of MUX to general routing
LO	Output	1	Output of MUX to local routing
I0	Input	1	Input (tie to MUXF7 LO out)
I1	Input	1	Input (tie to MUXF7 LO out)
S	Input	1	Input select to MUX

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
--          For use with all FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

MUXF8_D_inst : MUXF8_D
port map (
    LO => LO,    -- Ouptut of MUX to local routing
    O => O,      -- Output of MUX to general routing
    I0 => I0,    -- Input (tie to MUXF7 LO out)
    I1 => I1,    -- Input (tie to MUXF7 LO out)
    S => S      -- Input select to MUX
);
-- End of MUXF8_D_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF8_D: CLB MUX to tie two MUXF7's together with general and local outputs
//          For use with all FPGAs
// Xilinx HDL Libraries Guide, version 11.2

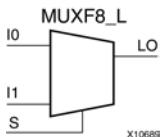
MUXF8_D MUXF8_D_inst (
    .LO(LO),    // Ouptut of MUX to local routing
    .O(O),      // Output of MUX to general routing
    .I0(I0),    // Input (tie to MUXF7 LO out)
    .I1(I1),    // Input (tie to MUXF7 LO out)
    .S(S)       // Input select to MUX
);
// End of MUXF8_D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## MUXF8\_L

**Primitive:** 2-to-1 Look-Up Table Multiplexer with Local Output



### Introduction

This design element provides a multiplexer function in eight slices for creating a function-of-8 look-up table or a 32-to-1 multiplexer in combination with the associated four look-up tables and two MUXF8s. Local outputs (LO) of MUXF7 are connected to the I0 and I1 inputs of the MUXF8. The S input is driven from any internal net. When Low, S selects I0. When High, S selects I1.

The LO output connects to other inputs in the same CLB slice.

### Logic Table

Inputs			Output
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

### Port Descriptions

Port	Direction	Width	Function
LO	Output	1	Output of MUX to local routing
I0	Input	1	Input (tie to MUXF7 LO out)
I1	Input	1	Input (tie to MUXF7 LO out)
S	Input	1	Input select to MUX

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- MUXF8_L: CLB MUX to tie two MUXF7's together with local output
--           For use with all FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

MUXF8_L_inst : MUXF8_L
port map (
    LO => LO,    -- Output of MUX to local routing
    I0 => I0,    -- Input (tie to MUXF7 LO out)
    I1 => I1,    -- Input (tie to MUXF7 LO out)
    S => S      -- Input select to MUX
);
-- End of MUXF8_L_inst instantiation
```

## Verilog Instantiation Template

```
// MUXF8_L: CLB MUX to tie two MUXF7's together with local output
//           For use with all FPGAs
// Xilinx HDL Libraries Guide, version 11.2

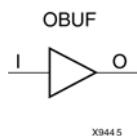
MUXF8_L#(MUXF8_L) MUXF8_L_inst (
    .LO(LO), // Output of MUX to local routing
    .I0(I0), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);
// End of MUXF8_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## OBUF

**Primitive:** Output Buffer



## Introduction

This design element is a simple output buffer used to drive output signals to the FPGA device pins that do not need to be 3-stated (constantly driven). Either an OBUF, OBUFT, OBUFDS, or OBUFTDS must be connected to every output port in the design.

This element isolates the internal circuit and provides drive current for signals leaving a chip. It exists in input/output blocks (IOB). Its output (O) is connected to an OPAD or an IOPAD. The interface standard used by this element is LVTTL. Also, this element has selectable drive and slew rates using the DRIVE and SLOW or FAST constraints. The defaults are DRIVE=12 mA and SLOW slew.

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Output of OBUF to be connected directly to top-level output port.
I	Input	1	Input of OBUF. Connect to the logic driving the output port.

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DRIVE	Integer	2, 4, 6, 8, 12, 16, 24	12	Specifies the output current drive strength of the I/O. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements.
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW" or "FAST"	"SLOW"	Specifies the slew rate of the output driver. Consult the product Data Sheet for recommendations of the best setting for this attribute.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUF: Single-ended Output Buffer
--      All devices
-- Xilinx HDL Libraries Guide, version 11.2

OBUF_inst : OBUF
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output (connect directly to top-level port)
    I => I      -- Buffer input
);
-- End of OBUF_inst instantiation
```

## Verilog Instantiation Template

```
// OBUF: Single-ended Output Buffer
//      All devices
// Xilinx HDL Libraries Guide, version 11.2

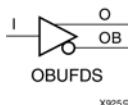
OBUF #(
    .DRIVE(12),    // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) OBUF_inst (
    .O(O),        // Buffer output (connect directly to top-level port)
    .I(I)         // Buffer input
);
// End of OBUF_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## OBUFDS

**Primitive:** Differential Signaling Output Buffer



### Introduction

This design element is a single output buffer that supports low-voltage, differential signaling (1.8 v CMOS). OBUFDS isolates the internal circuit and provides drive current for signals leaving the chip. Its output is represented as two distinct ports (O and OB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET and MYNETB).

### Logic Table

Inputs	Outputs	
I	O	OB
0	0	1
1	1	0

### Port Descriptions

Port	Direction	Width	Function
O	Output	1	Diff_p output (connect directly to top level port)
OB	Output	1	Diff_n output (connect directly to top level port)
I	Input	1	Buffer input

### Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUFDS: Differential Output Buffer
--          Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

OBUFDS_inst : OBUFDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I       -- Buffer input
);
-- End of OBUFDS_inst instantiation
```

## Verilog Instantiation Template

```
// OBUFDS: Differential Output Buffer
//          Virtex-4/5, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 11.2

OBUFDS #(
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFDS_inst (
    .O(O),      // Diff_p output (connect directly to top-level port)
    .OB(OB),    // Diff_n output (connect directly to top-level port)
    .I(I)       // Buffer input
);
// End of OBUFDS_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## OBUFT

**Primitive:** 3-State Output Buffer with Active Low Output Enable

OBUFT



X9449

### Introduction

This design element is a single, 3-state output buffer with input I, output O, and active-Low output enables (T). This element uses the LVTTL standard and has selectable drive and slew rates using the DRIVE and SLOW or FAST constraints. The defaults are DRIVE=12 mA and SLOW slew.

When T is Low, data on the inputs of the buffers is transferred to the corresponding outputs. When T is High, the output is high impedance (off or Z state). OBUFTs are generally used when a single-ended output is needed with a 3-state capability, such as the case when building bidirectional I/O.

### Logic Table

Inputs		Outputs	
T	I	O	
1	X	Z	
0	I	F	

### Port Descriptions

Port	Direction	Width	Function
O	Output	1	Buffer output (connect directly to top-level port)
I	Input	1	Buffer input
T	Input	1	3-state enable input

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
DRIVE	Integer	2, 4, 6, 8, 12, 16, 24	12	Specifies the output current drive strength of the I/O. It is suggested that you set this to the lowest setting tolerable for the design drive and timing requirements.
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.
SLEW	String	"SLOW" or "FAST"	"SLOW"	Specifies the slew rate of the output driver. See the Data Sheet for recommendations of the best setting for this attribute.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUFT: Single-ended 3-state Output Buffer
--          All devices
-- Xilinx HDL Libraries Guide, version 11.2

OBUFT_inst : OBUFT
generic map (
    DRIVE => 12,
    IOSTANDARD => "DEFAULT",
    SLEW => "SLOW")
port map (
    O => O,      -- Buffer output (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T      -- 3-state enable input
);
-- End of OBUFT_inst instantiation
```

## Verilog Instantiation Template

```
// OBUFT: Single-ended 3-state Output Buffer
//          All devices
// Xilinx HDL Libraries Guide, version 11.2

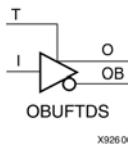
OBUFT #(
    .DRIVE(12),    // Specify the output drive strength
    .IOSTANDARD("DEFAULT"), // Specify the output I/O standard
    .SLEW("SLOW") // Specify the output slew rate
) OBUFT_inst (
    .O(O),        // Buffer output (connect directly to top-level port)
    .I(I),        // Buffer input
    .T(T)         // 3-state enable input
);
// End of OBUFT_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## OBUFTDS

Primitive: 3-State Output Buffer with Differential Signaling, Active-Low Output Enable



### Introduction

This design element is an output buffer that supports low-voltage, differential signaling. For the OBUFTDS, a design level interface signal is represented as two distinct ports (O and OB), one deemed the "master" and the other the "slave." The master and the slave are opposite phases of the same logical signal (for example, MYNET\_P and MYNET\_N).

### Logic Table

Inputs		Outputs	
I	T	O	OB
X	1	Z	Z
0	0	0	1
1	0	1	0

### Port Descriptions

Port	Direction	Width	Function
O	Output	1	Diff_p output (connect directly to top level port)
OB	Output	1	Diff_n output (connect directly to top level port)
I	Input	1	Buffer input
T	Input	1	3-state enable input

### Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
IOSTANDARD	String	See Data Sheet	"DEFAULT"	Assigns an I/O standard to the element.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- OBUFTDS: Differential 3-state Output Buffer
--          Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

OBUFTDS_inst : OBUFTDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T      -- 3-state enable input
);
-- End of OBUFTDS_inst instantiation
```

## Verilog Instantiation Template

```
// OBUFTDS: Differential 3-state Output Buffer
//          Virtex-4/5, Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 11.2

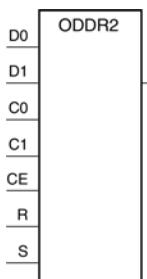
OBUFTDS #((
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFTDS_inst (
    .O(O),      // Diff_p output (connect directly to top-level port)
    .OB(OB),    // Diff_n output (connect directly to top-level port)
    .I(I),      // Buffer input
    .T(T)       // 3-state enable input
);
// End of OBUFTDS_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## ODDR2

**Primitive: Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset**



X10236

## Introduction

The design element is an output double data rate (DDR) register useful in producing double data rate signals exiting the FPGA. The ODDR2 requires two clocks (C0 and C1) to be connected to the component so that data is provided at the positive edge of both clocks. The ODDR2 features an active high clock enable port, CE, which can be used to suspend the operation of the registers and both set and reset ports that can be configured to be synchronous or asynchronous to the respective clocks. The ODDR2 has an optional alignment feature, which allows data to be captured by a single clock and clocked out by two clocks.

## Logic Table

Inputs							Outputs
S	R	CE	D0	D1	C0	C1	O
1	X	X	X	X	X	X	1
0	1	X	X	X	X	X	not INIT
0	0	0	X	X	X	X	No Change
0	0	1	D0	X	↑	X	D0
0	0	1	X	D1	X	↑	D1

Set/Reset can be synchronous via SRTYPE value

## Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
DDR_ALIGNMENT	String	"NONE", "C0", "C1"	"NONE"	Sets the input capture behavior for the DDR register. "NONE" clocks in data to the D0 input on the positive transition of the C0 clock and D1 on the positive transition of the C1 clock. "C0" allows the input clocking of both D0 and D1 align to the positive edge of the C0 clock. "C1" allows the input clocking of both D0 and D1 align to the positive edge of the C1 clock.
INIT	Integer	0, 1	0	Sets initial state of the Q0 output to 0 or 1.
SRTYPE	String	"SYNC", "ASYNC"	"SYNC"	Specifies "SYNC" or "ASYNC" set/reset.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ODDR2: Output Double Data Rate Output Register with Set, Reset
--         and Clock Enable.
--         Spartan-3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

ODDR2_inst : ODDR2
generic map(
    DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
    INIT => '0', -- Sets initial state of the Q output to '0' or '1'
    SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYNC" set/reset
port map (
    Q => Q, -- 1-bit output data
    C0 => C0, -- 1-bit clock input
    C1 => C1, -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    D0 => D0, -- 1-bit data input (associated with C0)
    D1 => D1, -- 1-bit data input (associated with C1)
    R => R, -- 1-bit reset input
    S => S -- 1-bit set input
);
-- End of ODDR2_inst instantiation
```

## Verilog Instantiation Template

```
// ODDR2: Output Double Data Rate Output Register with Set, Reset
//         and Clock Enable.
//         Spartan-3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

ODDR2 #(
    .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
    .INIT(1'b0), // Sets initial state of the Q output to 1'b0 or 1'b1
    .SRTYPE("SYNC") // Specifies "SYNC" or "ASYNC" set/reset
) ODDR2_inst (
    .Q(Q), // 1-bit DDR output data
    .C0(C0), // 1-bit clock input
    .C1(C1), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D0(D0), // 1-bit data input (associated with C0)
    .D1(D1), // 1-bit data input (associated with C1)
    .R(R), // 1-bit reset input
    .S(S) // 1-bit set input
);
```

```
// End of ODDR2_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## PULLDOWN

**Primitive:** Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs

PULLDOWN



X10690

### Introduction

This resistor element is connected to input, output, or bidirectional pads to guarantee a logic Low level for nodes that might float.

### Port Descriptions

Port	Direction	Width	Function
O	Output	1	Pulldown output (connect directly to top level port)

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- PULLDOWN: I/O Buffer Weak Pull-down
--           All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

PULLDOWN_inst : PULLDOWN
port map (
    O => O      -- Pulldown output (connect directly to top-level port)
);
-- End of PULLDOWN_inst instantiation
```

### Verilog Instantiation Template

```
// PULLDOWN: I/O Buffer Weak Pull-down
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

PULLDOWN PULLDOWN_inst (
    .O(O)      // Pulldown output (connect directly to top-level port)
);
// End of PULLDOWN_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## PULLUP

**Primitive:** Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs

PULLUP



X10691

## Introduction

This design element allows for an input, 3-state output or bi-directional port to be driven to a weak high value when not being driven by an internal or external source. This element establishes a High logic level for open-drain elements and macros when all the drivers are off.

## Port Descriptions

Port	Direction	Width	Function
O	Output	1	Pullup output (connect directly to top level port)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- PULLUP: I/O Buffer Weak Pull-up
--          All FPGA, CoolRunner-II
-- Xilinx HDL Libraries Guide, version 11.2

PULLUP_inst : PULLUP
port map (
    O => O      -- Pullup output (connect directly to top-level port)
);
-- End of PULLUP_inst instantiation
```

## Verilog Instantiation Template

```
// PULLUP: I/O Buffer Weak Pull-up
//          All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 11.2

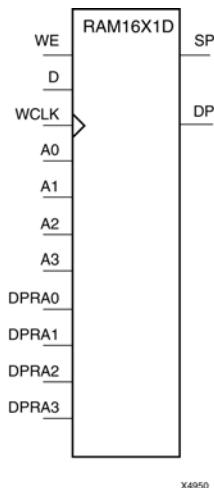
PULLUP PULLUP_inst (
    .O(O)      // Pullup output (connect directly to top-level port)
);
// End of PULLUP_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAM16X1D

Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM



X4950

### Introduction

This element is a 16-word by 1-bit static dual port random access memory with synchronous write capability. The device has two address ports: the read address (DPRA3:DPRA0) and the write address (A3:A0). These two address ports are asynchronous. The read address controls the location of the data driven out of the output pin (DPO), and the write address controls the destination of a valid write transaction. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected.

When WE is High, any positive transition on (WCLK) loads the data on the data input (D) into the word selected by the 4-bit write address. For predictable performance, write address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The SPO output reflects the data in the memory cell addressed by A3:A0. The DPO output reflects the data in the memory cell addressed by DPRA3:DPRA0.

**Note** The write process is not affected by the address on the read address port.

You can use the INIT attribute to directly specify an initial value. The value must be a hexadecimal number, for example, INIT=ABAC. If the INIT attribute is not specified, the RAM is initialized with all zeros.

### Logic Table

Mode selection is shown in the following logic table:

Inputs			Outputs	
WE (mode)	WCLK	D	SPO	DPO
0 (read)	X	X	data_a	data_d
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↑	D	D	data_d
1 (read)	↓	X	data_a	data_d
data_a = word addressed by bits A3-A0				
data_d = word addressed by bits DPRA3-DPRA0				

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros.	Initializes RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
--           All FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X1D_inst : RAM16X1D
generic map (
    INIT => X"0000")
port map (
    DPO => DPO,      -- Read-only 1-bit data output for DPRA
    SPO => SPO,      -- R/W 1-bit data output for A0-A3
    A0 => A0,        -- R/W address[0] input bit
    A1 => A1,        -- R/W address[1] input bit
    A2 => A2,        -- R/W address[2] input bit
    A3 => A3,        -- R/W address[3] input bit
    D => D,          -- Write 1-bit data input
    DPRA0 => DPRA0,  -- Read-only address[0] input bit
    DPRA1 => DPRA1,  -- Read-only address[1] input bit
    DPRA2 => DPRA2,  -- Read-only address[2] input bit
    DPRA3 => DPRA3,  -- Read-only address[3] input bit
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);
-- End of RAM16X1D_inst instantiation
```

## Verilog Instantiation Template

```
// RAM16X1D: 16 x 1 positive edge write, asynchronous read dual-port distributed RAM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

RAM16X1D #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_inst (
    .DPO(DPO),      // Read-only 1-bit data output for DPRA
    .SPO(SPO),      // R/W 1-bit data output for A0-A3
    .A0(A0),        // R/W address[0] input bit
    .A1(A1),        // R/W address[1] input bit
    .A2(A2),        // R/W address[2] input bit
    .A3(A3),        // R/W address[3] input bit
    .D(D),          // Write 1-bit data input
    .DPRA0(DPRA0),  // Read address[0] input bit
    .DPRA1(DPRA1),  // Read address[1] input bit
    .DPRA2(DPRA2),  // Read address[2] input bit
    .DPRA3(DPRA3),  // Read address[3] input bit
    .WCLK(WCLK),    // Write clock input
    .WE(WE)         // Write enable input
);

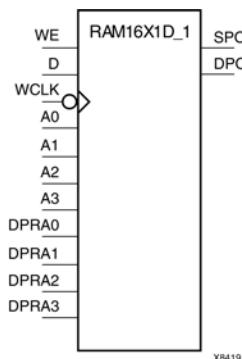
// End of RAM16X1D_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM16X1D\_1

Primitive: 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock



### Introduction

This is a 16-word by 1-bit static dual port random access memory with synchronous write capability and negative-edge clock. The device has two separate address ports: the read address (DPRA3:DPRA0) and the write address (A3:A0). These two address ports are asynchronous. The read address controls the location of the data driven out of the output pin (DPO), and the write address controls the destination of a valid write transaction.

When the write enable (WE) is set to Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 4-bit write address. For predictable performance, write address and data inputs must be stable before a High-to-Low WCLK transition. This RAM block assumes an active-High (WCLK). (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

You can initialize RAM16X1D\_1 during configuration using the INIT attribute.

The SPO output reflects the data in the memory cell addressed by A3:A0. The DPO output reflects the data in the memory cell addressed by DPRA3:DPRA0.

**Note** The write process is not affected by the address on the read address port.

### Logic Table

Mode selection is shown in the following logic table:

Inputs			Outputs	
WE (mode)	WCLK	D	SPO	DPO
0 (read)	X	X	data_a	data_d
1 (read)	0	X	data_a	data_d
1 (read)	1	X	data_a	data_d
1 (write)	↓	D	D	data_d
1 (read)	↑	X	data_a	data_d

data\_a = word addressed by bits A3:A0  
data\_d = word addressed by bits DPRA3:DPRA0

## Port Descriptions

Port	Direction	Width	Function
DPO	Output	1	Read-only 1-Bit data output
SPO	Output	1	R/W 1-Bit data output
A0	Input	1	R/W address[0] input
A1	Input	1	R/W address[1] input
A2	Input	1	R/W address[2] input
A3	Input	1	R/W address[3] input
D	Input	1	Write 1-Bit data input
DPRA0	Input	1	Read-only address[0] input
DPRA1	Input	1	Read-only address[1] input
DPRA2	Input	1	Read-only address[2] input
DPRA3	Input	1	Read-only address[3] input
WCLK	Input	1	Write clock input
WE	Input	1	Write enable input

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Initializes RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X1D_1_inst : RAM16X1D_1
generic map (
    INIT => X"0000")
port map (
    DPO => DPO,      -- Read-only 1-bit data output for DPRA
    SPO => SPO,      -- R/W 1-bit data output for A0-A3
    A0 => A0,        -- R/W address[0] input bit
    A1 => A1,        -- R/W address[1] input bit
    A2 => A2,        -- R/W address[2] input bit
    A3 => A3,        -- R/W address[3] input bit
    D => D,          -- Write 1-bit data input
    DPRA0 => DPRA0, -- Read-only address[0] input bit
    DPRA1 => DPRA1, -- Read-only address[1] input bit
    DPRA2 => DPRA2, -- Read-only address[2] input bit
    DPRA3 => DPRA3, -- Read-only address[3] input bit
    WCLK => WCLK,   -- Write clock input
    WE => WE        -- Write enable input
);
-- End of RAM16X1D_1_inst instantiation
```

## Verilog Instantiation Template

```
// RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
// All FPGA
// Xilinx HDL Libraries Guide, version 11.2

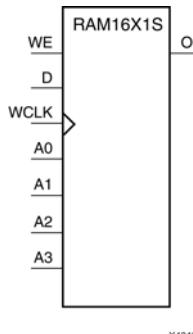
RAM16X1D_1 #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_1_inst (
    .DPO(DPO),      // Read-only 1-bit data output
    .SPO(SPO),      // R/W 1-bit data output
    .A0(A0),        // R/W address[0] input bit
    .A1(A1),        // R/W address[1] input bit
    .A2(A2),        // R/W address[2] input bit
    .A3(A3),        // R/W address[3] input bit
    .D(D),          // Write 1-bit data input
    .DPRA0(DPRA0),  // Read-only address[0] input bit
    .DPRA1(DPRA1),  // Read-only address[1] input bit
    .DPRA2(DPRA2),  // Read-only address[2] input bit
    .DPRA3(DPRA3),  // Read-only address[3] input bit
    .WCLK(WCLK),   // Write clock input
    .WE(WE)         // Write enable input
);
// End of RAM16X1D_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM16X1S

Primitive: 16-Deep by 1-Wide Static Synchronous RAM



X4942

### Introduction

This element is a 16-word by 1-bit static random access memory with synchronous write capability. When the write enable (WE) is set Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is set High, any positive transition on WCLK loads the data on the data input (D) into the word selected by the 4-bit address (A3:A0). This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM16X1S during configuration using the INIT attribute.

### Logic Table

Inputs			Outputs
WE(mode)	WCLK	D	O
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D	D
1 (read)	↓	X	Data
Data = word addressed by bits A3:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Specifies initial contents of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X1S: 16 x 1 posedge write distributed  => LUT RAM
--          All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X1S_inst : RAM16X1S
generic map (
    INIT => X"0000")
port map (
    O => O,           -- RAM output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    D => D,           -- RAM data input
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);
-- End of RAM16X1S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM16X1S: 16 x 1 posedge write distributed (LUT) RAM
//          All FPGA
// Xilinx HDL Libraries Guide, version 11.2

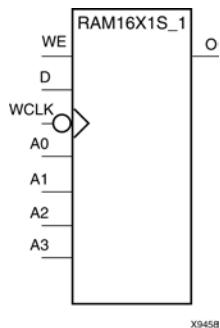
RAM16X1S #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1S_inst (
    .O(O),           // RAM output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .D(D),           // RAM data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);
// End of RAM16X1S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM16X1S\_1

Primitive: 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### Introduction

This element is a 16-word by 1-bit static random access memory with synchronous write capability and negative-edge clock. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 4-bit address (A3:A0). For predictable performance, address and data inputs must be stable before a High-to-Low WCLK transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can initialize this element during configuration using the INIT attribute.

### Logic Table

Inputs			Outputs
WE(mode)	WCLK	D	O
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = word addressed by bits A3:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Specifies initial contents of the RAM.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X1S_1: 16 x 1 negedge write distributed  => LUT RAM
--          All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X1S_1_inst : RAM16X1S_1
generic map (
    INIT => X"0000")
port map (
    O => O,          -- RAM output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    D => D,          -- RAM data input
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);
-- End of RAM16X1S_1_inst instantiation
```

### Verilog Instantiation Template

```
// RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
//          All FPGA
// Xilinx HDL Libraries Guide, version 11.2

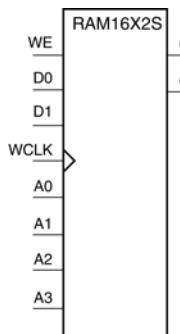
RAM16X1S_1 #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1S_1_inst (
    .O(O),          // RAM output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .D(D),          // RAM data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);
// End of RAM16X1S_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM16X2S

Primitive: 16-Deep by 2-Wide Static Synchronous RAM



X4944

### Introduction

This element is a 16-word by 2-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data input (D1:D0) into the word selected by the 4-bit address (A3:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O1:O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can use the INIT\_xx properties to specify the initial contents of a wide RAM. INIT\_00 initializes the RAM cells corresponding to the O0 output, INIT\_01 initializes the cells corresponding to the O1 output, etc. For example, a RAM16X2S instance is initialized by INIT\_00 and INIT\_01 containing 4 hex characters each. A RAM16X8S instance is initialized by eight properties INIT\_00 through INIT\_07 containing 4 hex characters each. A RAM64x2S instance is completely initialized by two properties INIT\_00 and INIT\_01 containing 16 hex characters each.

Except for Virtex-4 devices, the initial contents of this element cannot be specified directly.

### Logic Table

Inputs			Outputs
WE (mode)	WCLK	D1:D0	O1:O0
0 (read)	X	X	Data
1(read)	0	X	Data
1(read)	1	X	Data
1(write)	↑	D1:D0	D1:D0
1(read)	↓	X	Data
Data = word addressed by bits A3:A0			

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_01	Hexadecimal	Any 16-Bit Value	All zeros	Initializes RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X2S: 16 x 2 posedge write distributed => LUT RAM
--           All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X2S_inst : RAM16X2S
generic map (
    INIT_00 => X"0000", -- INIT for bit 0 of RAM
    INIT_01 => X"0000") -- INIT for bit 1 of RAM
port map (
    O0 => O0,      -- RAM data[0] output
    O1 => O1,      -- RAM data[1] output
    A0 => A0,      -- RAM address[0] input
    A1 => A1,      -- RAM address[1] input
    A2 => A2,      -- RAM address[2] input
    A3 => A3,      -- RAM address[3] input
    D0 => D0,      -- RAM data[0] input
    D1 => D1,      -- RAM data[1] input
    WCLK => WCLK, -- Write clock input
    WE => WE      -- Write enable input
);
-- End of RAM16X2S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
//          All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM16X2S #(
    .INIT_00(16'h0000), // Initial contents of bit 0 of RAM
    .INIT_01(16'h0000)  // Initial contents of bit 1 of RAM
) RAM16X2S_inst (
    .O0(O0),           // RAM data[0] output
    .O1(O1),           // RAM data[1] output
    .A0(A0),           // RAM address[0] input
    .A1(A1),           // RAM address[1] input
    .A2(A2),           // RAM address[2] input
    .A3(A3),           // RAM address[3] input
    .D0(D0),           // RAM data[0] input
    .D1(D1),           // RAM data[1] input
    .WCLK(WCLK),       // Write clock input
    .WE(WE)            // Write enable input
);

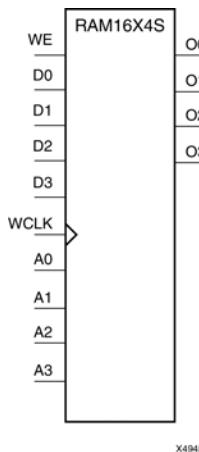
// End of RAM16X2S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM16X4S

**Primitive: 16-Deep by 4-Wide Static Synchronous RAM**



### Introduction

This element is a 16-word by 4-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data input (D3:D0) into the word selected by the 4-bit address (A3:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O3:O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

### Logic Table

Inputs			Outputs
WE (mode)	WCLK	D3:D0	O3:O0
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D3:D0	D3:D0
1 (read)	↓	X	Data

Data = word addressed by bits A3:A0.

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_03	Hexadecimal	Any 16-Bit Value	All zeros	INIT for bit 0 of RAM

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X4S: 16 x 4 posedge write distributed => LUT RAM
-- Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X4S_inst : RAM16X4S
generic map (
    INIT_00 => X"0000", -- INIT for bit 0 of RAM
    INIT_01 => X"0000", -- INIT for bit 1 of RAM
    INIT_02 => X"0000", -- INIT for bit 2 of RAM
    INIT_03 => X"0000") -- INIT for bit 3 of RAM
port map (
    O0 => O0,      -- RAM data[0] output
    O1 => O1,      -- RAM data[1] output
    O2 => O2,      -- RAM data[2] output
    O3 => O3,      -- RAM data[3] output
    A0 => A0,      -- RAM address[0] input
    A1 => A1,      -- RAM address[1] input
    A2 => A2,      -- RAM address[2] input
    A3 => A3,      -- RAM address[3] input
    D0 => D0,      -- RAM data[0] input
    D1 => D1,      -- RAM data[1] input
    D2 => D2,      -- RAM data[2] input
    D3 => D3,      -- RAM data[3] input
    WCLK => WCLK, -- Write clock input
    WE => WE      -- Write enable input
);
-- End of RAM16X4S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM16X4S: 16 x 4 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAM16X4S #(
    .INIT_00(16'h0000), // INIT for bit 0 of RAM
    .INIT_01(16'h0000), // INIT for bit 1 of RAM
    .INIT_02(16'h0000), // INIT for bit 2 of RAM
    .INIT_03(16'h0000) // INIT for bit 3 of RAM
) RAM16X4S_inst (
    .O0(O0),      // RAM data[0] output
    .O1(O1),      // RAM data[1] output
    .O2(O2),      // RAM data[2] output
    .O3(O3),      // RAM data[3] output
    .A0(A0),      // RAM address[0] input
    .A1(A1),      // RAM address[1] input
    .A2(A2),      // RAM address[2] input
    .A3(A3),      // RAM address[3] input
    .D0(D0),      // RAM data[0] input
    .D1(D1),      // RAM data[1] input
    .D2(D2),      // RAM data[2] input
    .D3(D3),      // RAM data[3] input
    .WCLK(WCLK), // Write clock input
    .WE(WE)       // Write enable input
);

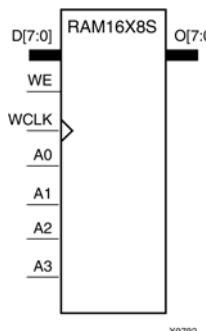
// End of RAM16X4S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM16X8S

Primitive: 16-Deep by 8-Wide Static Synchronous RAM



### Introduction

This element is a 16-word by 8-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on data inputs (D7:D0) into the word selected by the 4-bit address (A3:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O7:O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

### Logic Table

Inputs			Outputs
WE (mode)	WCLK	D7:D0	O7:O0
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D7:D0	D7:D0
1 (read)	↓	X	Data
Data = word addressed by bits A3–A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_07	Hexadecimal	Any 16-Bit Value	All zeros	Initializes RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM16X8S: 16 x 8 posedge write distributed  => LUT RAM
--          Virtex-4 and Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X8S_inst : RAM16X8S
generic map (
    INIT_00 => X"0000", -- INIT for bit 0 of RAM
    INIT_01 => X"0000", -- INIT for bit 1 of RAM
    INIT_02 => X"0000", -- INIT for bit 2 of RAM
    INIT_03 => X"0000", -- INIT for bit 3 of RAM
    INIT_04 => X"0000", -- INIT for bit 4 of RAM
    INIT_05 => X"0000", -- INIT for bit 5 of RAM
    INIT_06 => X"0000", -- INIT for bit 6 of RAM
    INIT_07 => X"0000") -- INIT for bit 7 of RAM
port map (
    O => O,           -- 8-bit RAM data output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    D => D,           -- 8-bit RAM data input
    WCLK => WCLK,    -- Write clock input
    WE => WE,         -- Write enable input
);
-- End of RAM16X8S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM16X8S: 16 x 8 posedge write distributed (LUT) RAM
//          Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

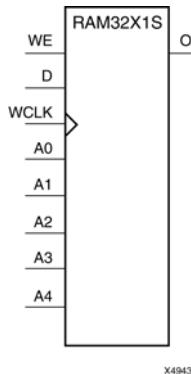
RAM16X8S #(
    .INIT_00(16'h0000), // INIT for bit 0 of RAM
    .INIT_01(16'h0000), // INIT for bit 1 of RAM
    .INIT_02(16'h0000), // INIT for bit 2 of RAM
    .INIT_03(16'h0000), // INIT for bit 3 of RAM
    .INIT_04(16'h0000), // INIT for bit 4 of RAM
    .INIT_05(16'h0000), // INIT for bit 5 of RAM
    .INIT_06(16'h0000), // INIT for bit 6 of RAM
    .INIT_07(16'h0000) // INIT for bit 7 of RAM
) RAM16X8S_inst (
    .O(O),           // 8-bit RAM data output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .D(D),           // 8-bit RAM data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);
// End of RAM16X8S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM32X1S

Primitive: 32-Deep by 1-Wide Static Synchronous RAM



### Introduction

The design element is a 32-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D) into the word selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM32X1S during configuration using the INIT attribute.

### Logic Table

Inputs			Outputs
WE (Mode)	WCLK	D	O
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies initial contents of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X1S: 32 x 1 posedge write distributed  => LUT RAM
--          All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X1S_inst : RAM32X1S
generic map (
    INIT => X"00000000")
port map (
    O => O,           -- RAM output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    A4 => A4,         -- RAM address[4] input
    D => D,           -- RAM data input
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);
-- End of RAM32X1S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
//          All FPGA
// Xilinx HDL Libraries Guide, version 11.2

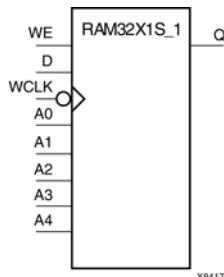
RAM32X1S #(
    .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1S_inst (
    .O(O),           // RAM output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .A4(A4),         // RAM address[4] input
    .D(D),           // RAM data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);
// End of RAM32X1S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM32X1S\_1

Primitive: 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



X8417

### Introduction

The design element is a 32-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins. You can initialize RAM32X1S\_1 during configuration using the INIT attribute.

### Logic Table

Inputs			Outputs
WE (Mode)	WCLK	D	O
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = word addressed by bits A4:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
INIT	Hexadecimal	Any 32-Bit Value	0	Initializes RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X1S_1: 32 x 1 negedge write distributed => LUT RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X1S_1_inst : RAM32X1S_1
generic map (
    INIT => X"00000000")
port map (
    O => O,          -- RAM output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    A4 => A4,         -- RAM address[4] input
    D => D,           -- RAM data input
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);
-- End of RAM32X1S_1_inst instantiation
```

## Verilog Instantiation Template

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
// All FPGA
// Xilinx HDL Libraries Guide, version 11.2

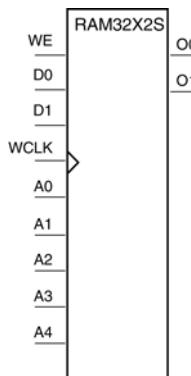
RAM32X1S_1 #(
    .INIT(32'h00000000) // Initial contents of RAM
)RAM32X1S_1_inst (
    .O(O),          // RAM output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .A4(A4),         // RAM address[4] input
    .D(D),           // RAM data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);
// End of RAM32X1S_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM32X2S

Primitive: 32-Deep by 2-Wide Static Synchronous RAM



X4947

### Introduction

The design element is a 32-word by 2-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any positive transition on (WCLK) loads the data on the data input (D1-D0) into the word selected by the 5-bit address (A4-A0). For predictable performance, address and data inputs must be stable before a Low-to-High (WCLK) transition. This RAM block assumes an active-High (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block. The signal output on the data output pins (O1-O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can use the INIT\_00 and INIT\_01 properties to specify the initial contents of RAM32X2S.

### Logic Table

Inputs			Outputs
WE (Mode)	WCLK	D	O0-O1
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D1:D0	D1:D0
1 (read)	↓	X	Data
Data = word addressed by bits A4:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Descriptions
INIT_00	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 0 of RAM.
INIT_01	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 1 of RAM.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X2S: 32 x 2 posedge write distributed  => LUT RAM
--           Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X2S_inst : RAM32X2S
generic map (
    INIT_00 => X"00000000", -- INIT for bit 0 of RAM
    INIT_01 => X"00000000") -- INIT for bit 1 of RAM
port map (
    O0 => O0,      -- RAM data[0] output
    O1 => O1,      -- RAM data[1] output
    A0 => A0,      -- RAM address[0] input
    A1 => A1,      -- RAM address[1] input
    A2 => A2,      -- RAM address[2] input
    A3 => A3,      -- RAM address[3] input
    A4 => A4,      -- RAM address[4] input
    D0 => D0,      -- RAM data[0] input
    D1 => D1,      -- RAM data[1] input
    WCLK => WCLK, -- Write clock input
    WE => WE      -- Write enable input
);
-- End of RAM32X2S_inst instantiation
```

### Verilog Instantiation Template

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

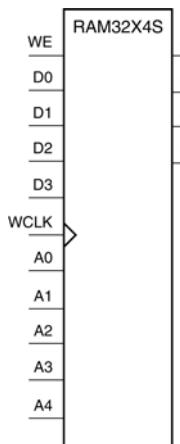
RAM32X2S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000) // INIT for bit 1 of RAM
) RAM32X2S_inst (
    .O0(O0),      // RAM data[0] output
    .O1(O1),      // RAM data[1] output
    .A0(A0),      // RAM address[0] input
    .A1(A1),      // RAM address[1] input
    .A2(A2),      // RAM address[2] input
    .A3(A3),      // RAM address[3] input
    .A4(A4),      // RAM address[4] input
    .D0(D0),      // RAM data[0] input
    .D1(D1),      // RAM data[1] input
    .WCLK(WCLK), -- Write clock input
    .WE(WE)       // Write enable input
);
// End of RAM32X2S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM32X4S

Primitive: 32-Deep by 4-Wide Static Synchronous RAM



X4948

### Introduction

This design element is a 32-word by 4-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data inputs (D3-D0) into the word selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O3-O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

### Logic Table

Inputs			Outputs
WE	WCLK	D3-D0	O3-O0
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D3:D0	D3:D0
1 (read)	↓	X	Data
Data = word addressed by bits A4:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 0 of RAM.
INIT_01	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 1 of RAM.
INIT_02	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 2 of RAM.
INIT_03	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 3 of RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X4S: 32 x 4 posedge write distributed  => LUT RAM
--          All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X4S_inst : RAM32X4S
generic map (
    INIT_00 => X"00000000", -- INIT for bit 0 of RAM
    INIT_01 => X"00000000", -- INIT for bit 1 of RAM
    INIT_02 => X"00000000", -- INIT for bit 2 of RAM
    INIT_03 => X"00000000") -- INIT for bit 3 of RAM
port map (
    O0 => O0,      -- RAM data[0] output
    O1 => O1,      -- RAM data[1] output
    O2 => O2,      -- RAM data[2] output
    O3 => O3,      -- RAM data[3] output
    A0 => A0,      -- RAM address[0] input
    A1 => A1,      -- RAM address[1] input
    A2 => A2,      -- RAM address[2] input
    A3 => A3,      -- RAM address[3] input
    A4 => A4,      -- RAM address[4] input
    D0 => D0,      -- RAM data[0] input
    D1 => D1,      -- RAM data[1] input
    D2 => D2,      -- RAM data[2] input
    D3 => D3,      -- RAM data[3] input
    WCLK => WCLK, -- Write clock input
    WE => WE      -- Write enable input
);
-- End of RAM32X4S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM32X4S: 32 x 4 posedge write distributed (LUT) RAM
//          Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM32X4S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000), // INIT for bit 1 of RAM
    .INIT_02(32'h00000000), // INIT for bit 2 of RAM
    .INIT_03(32'h00000000) // INIT for bit 3 of RAM
) RAM32X4S_inst (
    .O0(O0),      // RAM data[0] output
    .O1(O1),      // RAM data[1] output
    .O2(O2),      // RAM data[2] output
    .O3(O3),      // RAM data[3] output
    .A0(A0),      // RAM address[0] input
    .A1(A1),      // RAM address[1] input
    .A2(A2),      // RAM address[2] input
    .A3(A3),      // RAM address[3] input
    .A4(A4),      // RAM address[4] input
    .D0(D0),      // RAM data[0] input
    .D1(D1),      // RAM data[1] input
    .D2(D2),      // RAM data[2] input
    .D3(D3),      // RAM data[3] input
    .WCLK(WCLK), // Write clock input
    .WE(WE)       // Write enable input
);

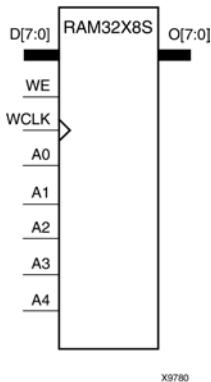
// End of RAM32X4S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM32X8S

Primitive: 32-Deep by 8-Wide Static Synchronous RAM



X9780

### Introduction

This design element is a 32-word by 8-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data inputs (D7:D0) into the word selected by the 5-bit address (A4:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O7:O0) is the data that is stored in the RAM at the location defined by the values on the address pins.

### Logic Table

Inputs			Outputs
WE (mode)	WCLK	D7:D0	O7:O0
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D7:D0	D7:D0
1 (read)	↓	X	Data
Data = word addressed by bits A4:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 0 of RAM.
INIT_01	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 1 of RAM.
INIT_02	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 2 of RAM.
INIT_03	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 3 of RAM.
INIT_04	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 4 of RAM.
INIT_05	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 5 of RAM.
INIT_06	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 6 of RAM.
INIT_07	Hexadecimal	Any 32-Bit Value	All zeros	INIT for bit 7 of RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM32X8S: 32 x 8 posedge write distributed  => LUT RAM
--           All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X8S_inst : RAM32X8S
generic map (
    INIT_00 => X"00000000", -- INIT for bit 0 of RAM
    INIT_01 => X"00000000", -- INIT for bit 1 of RAM
    INIT_02 => X"00000000", -- INIT for bit 2 of RAM
    INIT_03 => X"00000000", -- INIT for bit 3 of RAM
    INIT_04 => X"00000000", -- INIT for bit 4 of RAM
    INIT_05 => X"00000000", -- INIT for bit 5 of RAM
    INIT_06 => X"00000000", -- INIT for bit 6 of RAM
    INIT_07 => X"00000000") -- INIT for bit 7 of RAM
port map (
    O => O,          -- 8-bit RAM data output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    A4 => A4,         -- RAM address[4] input
    D => D,          -- 8-bit RAM data input
    WCLK => WCLK,    -- Write clock input
    WE => WE,         -- Write enable input
);
-- End of RAM32X8S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM32X8S: 32 x 8 posedge write distributed (LUT) RAM
//          Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM32X8S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000), // INIT for bit 1 of RAM
    .INIT_02(32'h00000000), // INIT for bit 2 of RAM
    .INIT_03(32'h00000000), // INIT for bit 3 of RAM
    .INIT_04(32'h00000000), // INIT for bit 4 of RAM
    .INIT_05(32'h00000000), // INIT for bit 5 of RAM
    .INIT_06(32'h00000000), // INIT for bit 6 of RAM
    .INIT_07(32'h00000000) // INIT for bit 7 of RAM
) RAM32X8S_inst (
    .O(O),           // 8-bit RAM data output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .A4(A4),         // RAM address[4] input
    .D(D),           // 8-bit RAM data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);

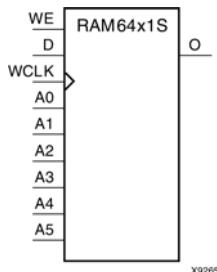
// End of RAM32X8S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM64X1S

Primitive: 64-Deep by 1-Wide Static Synchronous RAM



### Introduction

This design element is a 64-word by 1-bit static random access memory (RAM) with synchronous write capability. When the write enable is set Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is set High, any positive transition on WCLK loads the data on the data input (D) into the word selected by the 6-bit address (A5:A0). This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can initialize this element during configuration using the INIT attribute.

### Logic Table

Mode selection is shown in the following logic table

Inputs			Outputs
WE (mode)	WCLK	D	O
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D	D
1 (read)	↓	X	Data
Data = word addressed by bits A5:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Initializes ROMs, RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
--           Virtex-4/5, Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM64X1S_inst : RAM64X1S
generic map (
    INIT => X"0000000000000000")
port map (
    O => O,          -- 1-bit data output
    A0 => A0,        -- Address[0] input bit
    A1 => A1,        -- Address[1] input bit
    A2 => A2,        -- Address[2] input bit
    A3 => A3,        -- Address[3] input bit
    A4 => A4,        -- Address[4] input bit
    A5 => A5,        -- Address[5] input bit
    D => D,          -- 1-bit data input
    WCLK => WCLK,   -- Write clock input
    WE => WE,        -- Write enable input
);
-- End of RAM64X1S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM64X1S: 64 x 1 positive edge write, asynchronous read single-port distributed RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

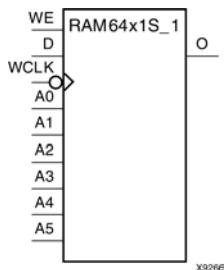
RAM64X1S #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_inst (
    .O(O),          // 1-bit data output
    .A0(A0),        // Address[0] input bit
    .A1(A1),        // Address[1] input bit
    .A2(A2),        // Address[2] input bit
    .A3(A3),        // Address[3] input bit
    .A4(A4),        // Address[4] input bit
    .A5(A5),        // Address[5] input bit
    .D(D),          // 1-bit data input
    .WCLK(WCLK),   // Write clock input
    .WE(WE)         // Write enable input
);
// End of RAM64X1S_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM64X1S\_1

Primitive: 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



### Introduction

This design element is a 64-word by 1-bit static random access memory with synchronous write capability. When the write enable is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When (WE) is High, any negative transition on (WCLK) loads the data on the data input (D) into the word selected by the 6-bit address (A5:A0). For predictable performance, address and data inputs must be stable before a High-to-Low (WCLK) transition. This RAM block assumes an active-Low (WCLK). However, (WCLK) can be active-High or active-Low. Any inverter placed on the (WCLK) input net is absorbed into the block.

The signal output on the data output pin (O) is the data that is stored in the RAM at the location defined by the values on the address pins.

You can initialize this element during configuration using the INIT attribute.

### Logic Table

Inputs			Outputs
WE (mode)	WCLK	D	O
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↓	D	D
1 (read)	↑	X	Data
Data = word addressed by bits A5:A0			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Initializes ROMs, RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
-- Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM64X1S_1_inst : RAM64X1S_1
generic map (
    INIT => X"0000000000000000")
port map (
    O => O,           -- 1-bit data output
    A0 => A0,         -- Address[0] input bit
    A1 => A1,         -- Address[1] input bit
    A2 => A2,         -- Address[2] input bit
    A3 => A3,         -- Address[3] input bit
    A4 => A4,         -- Address[4] input bit
    A5 => A5,         -- Address[5] input bit
    D => D,           -- 1-bit data input
    WCLK => WCLK,    -- Write clock input
    WE => WE,         -- Write enable input
);
-- End of RAM64X1S_1_inst instantiation
```

## Verilog Instantiation Template

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
// All FPGA
// Xilinx HDL Libraries Guide, version 11.2

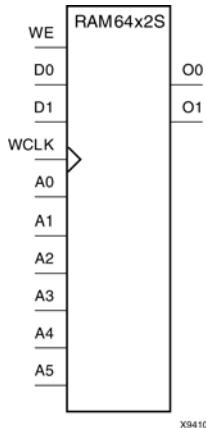
RAM64X1S_1 #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
    .O(O),           // 1-bit data output
    .A0(A0),         // Address[0] input bit
    .A1(A1),         // Address[1] input bit
    .A2(A2),         // Address[2] input bit
    .A3(A3),         // Address[3] input bit
    .A4(A4),         // Address[4] input bit
    .A5(A5),         // Address[5] input bit
    .D(D),           // 1-bit data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)          // Write enable input
);
// End of RAM64X1S_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAM64x2S

Primitive: 64-Deep by 2-Wide Static Synchronous RAM



## Introduction

This design element is a 64-word by 2-bit static random access memory with synchronous write capability. When the write enable (WE) is Low, transitions on the write clock (WCLK) are ignored and data stored in the RAM is not affected. When WE is High, any positive transition on WCLK loads the data on the data input (D1:D0) into the word selected by the 6-bit address (A5:A0). For predictable performance, address and data inputs must be stable before a Low-to-High WCLK transition. This RAM block assumes an active-High WCLK. However, WCLK can be active-High or active-Low. Any inverter placed on the WCLK input net is absorbed into the block.

The signal output on the data output pins (O1:O0) is the data that is stored in the RAM at the location defined by the values on the address pins. You can use the INIT\_00 and INIT\_01 properties to specify the initial contents of this design element.

## Logic Table

Inputs		Outputs	
WE (mode)	WCLK	D0:D1	O0:O1
0 (read)	X	X	Data
1 (read)	0	X	Data
1 (read)	1	X	Data
1 (write)	↑	D1:D0	D1:D0
1 (read)	↓	X	Data
Data = word addressed by bits A5:A0			

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00	Hexadecimal	Any 64-Bit Value	All zeros	Initializes RAMs, registers, and look-up tables.
INIT_01	Hexadecimal	Any 64-Bit Value	All zeros	Initializes RAMs, registers, and look-up tables.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAM64X2S: 64 x 2 positive edge write, asynchronous read single-port distributed RAM
--          Virtex-4/5
--          Xilinx HDL Libraries Guide, version 11.2

RAM64X2S_inst : RAM64X2S
generic map (
    INIT_00 => X"0000000000000000", -- INIT for bit 0 of RAM
    INIT_01 => X"0000000000000000") -- INIT for bit 1 of RAM
port map (
    O0 => O0,      -- Data[0] output
    O1 => O1,      -- Data[1] output bit
    A0 => A0,      -- Address[0] input bit
    A1 => A1,      -- Address[1] input bit
    A2 => A2,      -- Address[2] input bit
    A3 => A3,      -- Address[3] input bit
    A4 => A4,      -- Address[4] input bit
    A5 => A5,      -- Address[5] input bit
    D0 => D0,      -- Data[0] input
    D1 => D1,      -- Data[1] input
    WCLK => WCLK, -- Write clock input
    WE => WE       -- Write enable input
);
-- End of RAM64X2S_inst instantiation
```

## Verilog Instantiation Template

```
// RAM64X2S: 64 x 2 positive edge write, asynchronous read single-port distributed RAM
//          Virtex-II/II-Pro
//          Xilinx HDL Libraries Guide, version 10.1.2

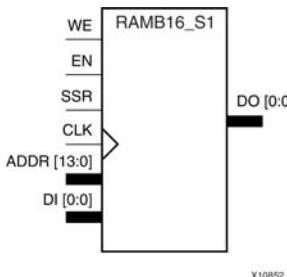
RAM64X2S #(
    .INIT_00(64'h0000000000000000), // INIT for RAM bit 0
    .INIT_01(64'h0000000000000000) // INIT for RAM bit 1
) RAM64X2S_inst (
    .O0(O0),      // Data[0] output
    .O1(O1),      // Data[1] output bit
    .A0(A0),      // Address[0] input bit
    .A1(A1),      // Address[1] input bit
    .A2(A2),      // Address[2] input bit
    .A3(A3),      // Address[3] input bit
    .A4(A4),      // Address[4] input bit
    .A5(A5),      // Address[5] input bit
    .D0(D0),      // Data[0] input
    .D1(D1),      // Data[1] input
    .WCLK(WCLK), -- Write clock input
    .WE(WE)       // Write enable input
);
// End of RAM64X2S_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S1

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 1-bit Port



## Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

Data Cells		Parity Cells				
Depth	Width	Depth	Width	Address Bus	Data Bus	Parity Bus
16384	1	-	-	(13:0)	(0:0)	-

The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

## Logic Table

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT	INIT	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM (addr)	RAM (addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change(a) RAM (addr)(b) data(c)	No Change(a) RAM (addr)(b) pdata(c)	RAM (addr)=>data	RAM (addr)=>pdata

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
GSR=Global Set Reset signal											
INIT=Value specified by the INIT attribute for data memory. Default is all zeros.											
SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.											
addr=RAM address.											
RAM(addr)=RAM contents at address ADDR.											
data=RAM input data.											
pdata=RAM parity data.											
(a) WRITE_MODE=NO_CHANGE											
(b) WRITE_MODE=READ_FIRST											
(c) WRITE_MODE=WRITE_FIRST											

## Initialization

### Initializing Memory Contents

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

Any INIT\_xx or INITP\_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

### Initializing the Output Register

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16\_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16\_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

### Selecting Write Mode

The WRITE\_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE\_MODE is set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE\_MODE to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE\_MODE to NO\_CHANGE to have the input written to memory without changing the output.

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary/ Hexadecimal	Any Hex Value	All zeros	Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM.
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the data portion of the RAM array.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SRVAL	Binary/ Hexadecimal	Any Hex Value	All zeros	Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM.
WRITE_MODE	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```
) ;

-- End of RAMB16_S1_inst instantiation
```

## Verilog Instantiation Template

```
// RAMB16_S1: Spartan-3E/3A/3AN/3AD 16kx1 Single-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1 #(
    .INIT(1'b0), // Value of output RAM registers at startup
    .SRVAL(1'b0), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 4095
    .INIT_00(256'h00000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000),
    // Address 4096 to 8191
    .INIT_10(256'h00000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000),
    .INIT_1A(256'h00000000000000000000000000000000),
    .INIT_1B(256'h00000000000000000000000000000000),
    .INIT_1C(256'h00000000000000000000000000000000),
    .INIT_1D(256'h00000000000000000000000000000000),
    .INIT_1E(256'h00000000000000000000000000000000),
    .INIT_1F(256'h00000000000000000000000000000000),
    // Address 8192 to 12287
    .INIT_20(256'h00000000000000000000000000000000),
    .INIT_21(256'h00000000000000000000000000000000),
    .INIT_22(256'h00000000000000000000000000000000),
    .INIT_23(256'h00000000000000000000000000000000),
    .INIT_24(256'h00000000000000000000000000000000),
    .INIT_25(256'h00000000000000000000000000000000),
    .INIT_26(256'h00000000000000000000000000000000),
    .INIT_27(256'h00000000000000000000000000000000),
    .INIT_28(256'h00000000000000000000000000000000),
    .INIT_29(256'h00000000000000000000000000000000),
    .INIT_2A(256'h00000000000000000000000000000000),
    .INIT_2B(256'h00000000000000000000000000000000),
    .INIT_2C(256'h00000000000000000000000000000000),
    .INIT_2D(256'h00000000000000000000000000000000),
    .INIT_2E(256'h00000000000000000000000000000000),
    .INIT_2F(256'h00000000000000000000000000000000),
    // Address 12288 to 16383
    .INIT_30(256'h00000000000000000000000000000000),
    .INIT_31(256'h00000000000000000000000000000000),
    .INIT_32(256'h00000000000000000000000000000000),
    .INIT_33(256'h00000000000000000000000000000000),
    .INIT_34(256'h00000000000000000000000000000000)
```

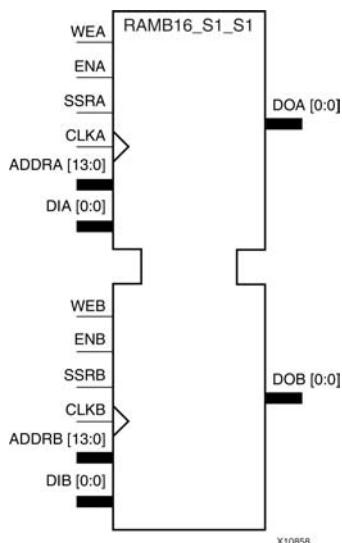
```
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S1_inst (
    .DO(DO),           // 1-bit Data Output
    .ADDR(ADDR),       // 14-bit Address Input
    .CLK(CLK),         // Clock
    .DI(DI),           // 1-bit Data Input
    .EN(EN),           // RAM Enable Input
    .SSR(SSR),         // Synchronous Set/Reset Input
    .WE(WE)            // Write Enable Input
);
// End of RAMB16_S1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S1\_S1

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S1_S1	16384 x 1	-	(13:0)	(0:0)	-	16384 x 1	-	(13:0)	(0:0)	-	
(a) Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7																																0
8	2048	<--	3																																0
16	1024	<--	1																																
32	512	<--	0																																

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																
1	2048	<----	3														1															0	
2	1024	<----	1														0																
4	512	<----	0																														

### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.

Attribute	Type	Allowed Values	Default	Description
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

Library UNISIM;



```

INIT_36 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",

port map (
    DOA => DOA,          -- Port A 1-bit Data Output
    DOB => DOB,          -- Port B 1-bit Data Output
    ADDRA => ADDRA,      -- Port A 14-bit Address Input
    ADDR_B => ADDR_B,    -- Port B 14-bit Address Input
    CLKA => CLKA,        -- Port A Clock
    CLKB => CLKB,        -- Port B Clock
    DIA => DIA,          -- Port A 1-bit Data Input
    DIB => DIB,          -- Port B 1-bit Data Input
    ENA => ENA,          -- Port A RAM Enable Input
    ENB => ENB,          -- Port B RAM Enable Input
    SSRA => SSRA,        -- Port A Synchronous Set/Reset Input
    SSRB => SSRB,        -- Port B Synchronous Set/Reset Input
    WEA => WEA,          -- Port A Write Enable Input
    WEB => WEB,          -- Port B Write Enable Input
);

-- End of RAMB16_S1_S1_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S1_S1: Spartan-3/3E/3A/3AD 16k x 1 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S1 #(
    .INIT_A(1'b0),    // Value of output RAM registers on Port A at startup
    .INIT_B(1'b0),    // Value of output RAM registers on Port B at startup
    .SRVAL_A(1'b0),   // Port A output value upon SSR assertion
    .SRVAL_B(1'b0),   // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 4095
    .INIT_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),

    // Address 4096 to 8191
    .INIT_10(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000)
);

```

```

.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 8192 to 12287
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 12288 to 16383
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S1_S1_inst (
    .DOA(DOA),          // Port A 1-bit Data Output
    .DOB(DOB),          // Port B 1-bit Data Output
    .ADDRA(ADDRA),      // Port A 14-bit Address Input
    .ADDRB(ADDRB),      // Port B 14-bit Address Input
    .CLKA(CLKA),        // Port A Clock
    .CLKB(CLKB),        // Port B Clock
    .DIA(DIA),          // Port A 1-bit Data Input
    .DIB(DIB),          // Port B 1-bit Data Input
    .ENA(ENA),          // Port A RAM Enable Input
    .ENB(ENB),          // Port B RAM Enable Input
    .SSRA(SSRA),        // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),        // Port B Synchronous Set/Reset Input
    .WEA(WEA),          // Port A Write Enable Input
    .WEB(WEB)           // Port B Write Enable Input
);
// End of RAMB16_S1_S1_inst instantiation

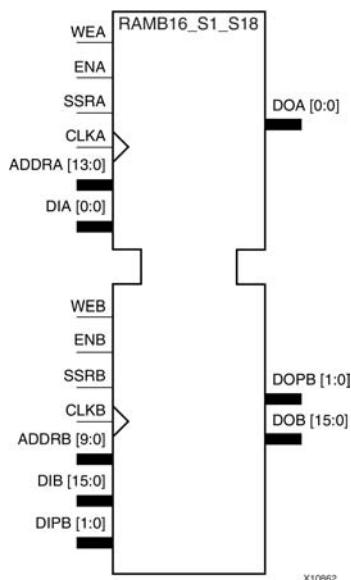
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S1\_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 18-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S1_S18	16384 x 1	-	(13:0)	(0:0)	-	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	
(a) Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7				6				5				4				3				2			1								0		
8	2048	<--	3							2									1																0	
16	1024	<--	1																0																	
32	512	<--	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																	
1	2048	<----	3						2						1																0			
2	1024	<----	1													0																		
4	512	<----	0																															

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,          -- Port A 1-bit Data Output
DOB => DOB,          -- Port B 16-bit Data Output
DOPB => DOPB,         -- Port B 2-bit Parity Output
ADRA => ADRA,         -- Port A 14-bit Address Input
ADDRB => ADDR,        -- Port B 10-bit Address Input
CLKA => CLKA,         -- Port A Clock
CLKB => CLKB,         -- Port B Clock
DIA => DIA,          -- Port A 1-bit Data Input
DIB => DIB,           -- Port B 16-bit Data Input
DIPB => DIPB,          -- Port-B 2-bit parity Input
ENA => ENA,           -- Port A RAM Enable Input
ENB => ENB,            -- PortB RAM Enable Input
SSRA => SSRA,          -- Port A Synchronous Set/Reset Input
SSRB => SSRB,          -- Port B Synchronous Set/Reset Input
WEA => WEA,            -- Port A Write Enable Input
WEB => WEB,             -- Port B Write Enable Input
);

-- End of RAMB16_S1_S18_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S1_S18: Spartan-3E/3A/3AN/3AD 16k/1k x 1/16 + 0/2 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S18 #(
    .INIT_A(1'b0),      // Value of output RAM registers on Port A at startup
    .INIT_B(18'h00000), // Value of output RAM registers on Port B at startup
    .SRVAL_A(1'b0),     // Port A output value upon SSR assertion
    .SRVAL_B(18'h00000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 4095, Port B Address 0 to 255
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 4096 to 8191, Port B Address 256 to 511
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 8192 to 12287, Port B Address 512 to 767
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000)
);

```

```

.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 12288 to 16383, Port B Address 768 to 1023
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 255
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 256 to 511
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 512 to 767
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 768 to 1023
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S1_S18_inst (
    .DOA(DOA),           // Port A 1-bit Data Output
    .DOB(DOB),           // Port B 16-bit Data Output
    .DOPB(DOPB),          // Port B 2-bit Parity Output
    .ADDRA(ADDRA),         // Port A 14-bit Address Input
    .ADDRB(ADDRB),         // Port B 10-bit Address Input
    .CLKA(CLKA),          // Port A Clock
    .CLKB(CLKB),          // Port B Clock
    .DIA(DIA),           // Port A 1-bit Data Input
    .DIB(DIB),           // Port B 16-bit Data Input
    .DIPB(DIPB),          // Port-B 2-bit parity Input
    .ENA(ENA),            // Port A RAM Enable Input
    .ENB(ENB),            // Port B RAM Enable Input
    .SSRA(SSRA),          // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),          // Port B Synchronous Set/Reset Input
    .WEA(WEA),             // Port A Write Enable Input
    .WEB(WEB)             // Port B Write Enable Input
);
// End of RAMB16_S1_S18_inst instantiation

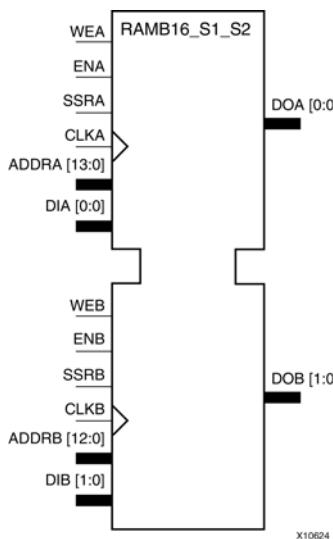
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S1\_S2

**Primitive:** 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 2-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs								Outputs			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
								DOA	DOPA	Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
								Data RAM	Parity RAM	

RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S1_S2	16384 x 1	-	(13:0)	(0:0)	-	8192 x 2	-	(12:0)	(1:0)	-	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables show address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7				6			5				4				3				2				1							0		
8	2048	<--	3							2									1															0	
16	1024	<--	1																0																
32	512	<--	0																																

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																															
1	2048	<----	3						2							1														0		
2	1024	<----	1														0															
4	512	<----	0																													

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan-3A, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

WEB => WEB      -- Port B Write Enable Input
);

-- End of RAMB16_S1_S2_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S1_S2: Spartan-3E/3A/3AN/3AD 16k/8k x 1/2 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S2 #(
    .INIT_A(1'b0),      // Value of output RAM registers on Port A at startup
    .INIT_B(2'b00),     // Value of output RAM registers on Port B at startup
    .SRVAL_A(1'b0),     // Port A output value upon SSR assertion
    .SRVAL_B(2'b00),    // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 4095, Port B Address 0 to 2047
    .INIT_00(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 4096 to 8191, Port B Address 2048 to 4095
    .INIT_10(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 8192 to 12287, Port B Address 4095 to 6143
    .INIT_20(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_22(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_23(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_24(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_25(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_26(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_27(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_28(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_29(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2A(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2B(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2C(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2D(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2E(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2F(256'h000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 12288 to 16383, Port B Address 6144 to 8091

```

```

.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S1_S2_inst (
    .DOA(DOA),           // Port A 1-bit Data Output
    .DOB(DOB),           // Port B 2-bit Data Output
    .ADDRA(ADDRA),       // Port A 14-bit Address Input
    .ADDRB(ADDRB),       // Port B 13-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 1-bit Data Input
    .DIB(DIB),           // Port B 2-bit Data Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);
// End of RAMB16_S1_S2_inst instantiation

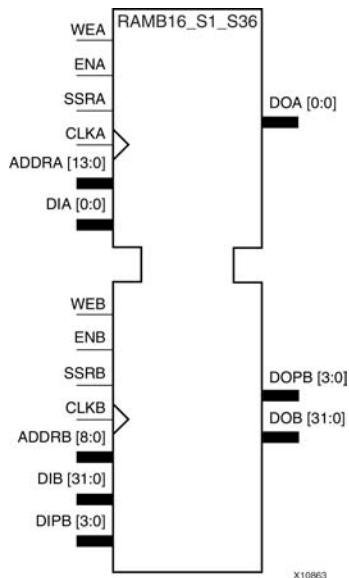
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S1\_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 36-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S1_S36	16384 x 1	-	(13:0)	(0:0)	-	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	
(a) Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the Port Address Mapping for Data table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in Port Address Mapping for Parity table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables show address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7																																0
8	2048	<--	3																																0
16	1024	<--	1																																
32	512	<--	0																																

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																	
1	2048	<----	3														1															0		
2	1024	<----	1																															
4	512	<----	0																															

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 1-bit Data Output
DOB => DOB,      -- Port B 32-bit Data Output
DOPB => DOPB,    -- Port B 4-bit Parity Output
ADRA => ADRA,    -- Port A 14-bit Address Input
ADDRB => ADDR,   -- Port B 9-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 1-bit Data Input
DIB => DIB,      -- Port B 32-bit Data Input
DIPB => DIPB,    -- Port-B 4-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S1_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S1_S36: Spartan-3E/3A/3AN/3AD 16k/512 x 1/32 + 0/4 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S36 #(
    .INIT_A(1'b0),          // Value of output RAM registers on Port A at startup
    .INIT_B(36'h0000000000), // Value of output RAM registers on Port B at startup
    .SRVAL_A(1'b0),         // Port A output value upon SSR assertion
    .SRVAL_B(36'h0000000000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 4095, Port B Address 0 to 127
    .INIT_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

    // Port A Address 4096 to 8191, Port B Address 128 to 255
    .INIT_10(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

    // Port A Address 8192 to 12287, Port B Address 256 to 383
    .INIT_20(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
);

```

```

.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 12288 to 16383, Port B Address 384 to 512
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 127
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 128 to 255
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 256 to 383
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 384 to 512
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S1_S36_inst (
    .DOA(DOA),           // Port A 1-bit Data Output
    .DOB(DOB),           // Port B 32-bit Data Output
    .DOPB(DOPB),          // Port B 4-bit Parity Output
    .ADDRA(ADDRA),         // Port A 14-bit Address Input
    .ADDRB(ADDRB),         // Port B 9-bit Address Input
    .CLKA(CLKA),          // Port A Clock
    .CLKB(CLKB),          // Port B Clock
    .DIA(DIA),           // Port A 1-bit Data Input
    .DIB(DIB),           // Port B 32-bit Data Input
    .DIPB(DIPB),          // Port-B 4-bit parity Input
    .ENA(ENA),            // Port A RAM Enable Input
    .ENB(ENB),            // Port B RAM Enable Input
    .SSRA(SSRA),          // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),          // Port B Synchronous Set/Reset Input
    .WEA(WEA),             // Port A Write Enable Input
    .WEB(WEB)             // Port B Write Enable Input
);

// End of RAMB16_S1_S36_inst instantiation

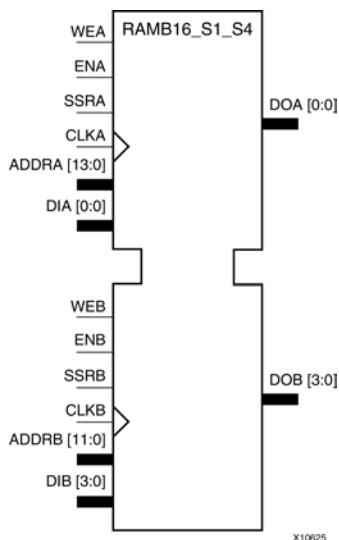
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S1\_S4

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 4-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S1_S4	16384 x 1	-	(13:0)	(0:0)	-	4096 x 4	-	(11:0)	(3:0)	-	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7			6			5			4				3				2			1				1			0						
8	2048	<--	3							2								1									0									
16	1024	<--	1																0																	
32	512	<--	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																		
1	2048	<----	3						2							1										0									
2	1024	<----	1														0																		
4	512	<----	0																																

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design</p>

Attribute	Type	Allowed Values	Default	Description
				go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S1_S4: Spartan-3/3E 16k/4k x 1/4 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S4_inst : RAMB16_S1_S4
generic map (
```



```

INIT_3D => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
port map (
  DOA => DOA,      -- Port A 1-bit Data Output
  DOB => DOB,      -- Port B 4-bit Data Output
  ADDRA => ADDRA,  -- Port A 14-bit Address Input
  ADDR_B => ADDR_B, -- Port B 12-bit Address Input
  CLKA => CLKA,    -- Port A Clock
  CLK_B => CLK_B,   -- Port B Clock
  DIA => DIA,      -- Port A 1-bit Data Input
  DIB => DIB,      -- Port B 4-bit Data Input
  ENA => ENA,      -- Port A RAM Enable Input
  ENB => ENB,      -- Port B RAM Enable Input
  SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
  WEA => WEA,      -- Port A Write Enable Input
  WEB => WEB,      -- Port B Write Enable Input
);
-- End of RAMB16_S1_S4_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S1_S4: Spartan-3/3E/3A/3AN/3AD 16k/4k x 1/4 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S4 #(
  .INIT_A(1'b0), // Value of output RAM registers on Port A at startup
  .INIT_B(4'h0), // Value of output RAM registers on Port B at startup
  .SRVAL_A(1'b0), // Port A output value upon SSR assertion
  .SRVAL_B(4'h0), // Port B output value upon SSR assertion
  .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

  // The following INIT_xx declarations specify the initial contents of the RAM
  // Port A Address 0 to 4095, Port B Address 0 to 1023
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
  // Port A Address 4096 to 8191, Port B Address 1024 to 2047
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
  // Port A Address 8192 to 12287, Port B Address 2048 to 3071

```

```

.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 12288 to 16383, Port B Address 3072 to 4095
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S1_S4_inst (
.DOA(DOA),          // Port A 1-bit Data Output
.DOB(DOB),          // Port B 4-bit Data Output
.ADDRA(ADDRA),      // Port A 14-bit Address Input
.ADDRB(ADDRB),      // Port B 12-bit Address Input
.CLKA(CLKA),        // Port A Clock
.CLKB(CLKB),        // Port B Clock
.DIA(DIA),          // Port A 1-bit Data Input
.DIB(DIB),          // Port B 4-bit Data Input
.ENA(ENA),          // Port A RAM Enable Input
.ENB(ENB),          // Port B RAM Enable Input
.SSRA(SSRA),        // Port A Synchronous Set/Reset Input
.SSRB(SSRB),        // Port B Synchronous Set/Reset Input
.WEA(WEA),          // Port A Write Enable Input
.WEB(WEB)           // Port B Write Enable Input
);
// End of RAMB16_S1_S4_inst instantiation

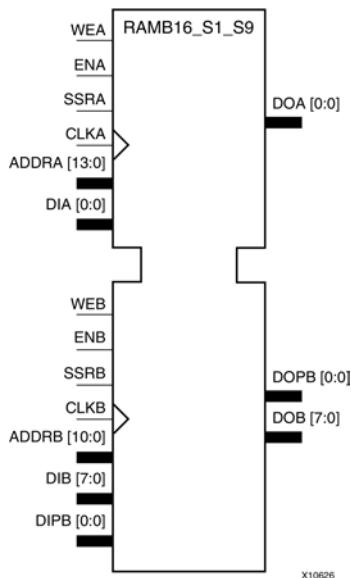
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S1\_S9

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 1-bit and 9-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S1_S9	16384 x 1	-	(13:0)	(0:0)	-	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	
(a) Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

## *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)\*(Widthport)) -1

End=(ADDRport)\*(Widthport)

The following tables shows address mapping for each port width.

## *Port Address Mapping for Data*

Port Address Mapping for Parity

Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 1-bit Data Output
DOB => DOB,      -- Port B 8-bit Data Output
DOPB => DOPB,    -- Port B 1-bit Parity Output
ADRA => ADRA,    -- Port A 14-bit Address Input
ADDRB => ADDR,   -- Port B 11-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 1-bit Data Input
DIB => DIB,      -- Port B 8-bit Data Input
DIPB => DIPB,    -- Port-B 1-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S1_S9_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S1_S9: Spartan-3E/3A/3AN/3AD 16k/2k x 1/8 + 0/1 Parity bit Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S1_S9 #(
    .INIT_A(1'b0),      // Value of output RAM registers on Port A at startup
    .INIT_B(9'h000),    // Value of output RAM registers on Port B at startup
    .SRVAL_A(1'b0),    // Port A output value upon SSR assertion
    .SRVAL_B(9'h000),  // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 4095, Port B Address 0 to 511
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 4096 to 8191, Port B Address 512 to 1023
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 8192 to 12287, Port B Address 1024 to 1535
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000)
);

```

```

.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 12288 to 16383, Port B Address 1535 to 2047
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 511
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 512 to 1023
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 1024 to 1535
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 1535 to 2047
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S1_S9_inst (
    .DOA(DOA),           // Port A 1-bit Data Output
    .DOB(DOB),           // Port B 8-bit Data Output
    .DOPB(DOPB),         // Port B 1-bit Parity Output
    .ADDRA(ADDRA),       // Port A 14-bit Address Input
    .ADDRB(ADDRB),       // Port B 11-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 1-bit Data Input
    .DIB(DIB),           // Port B 8-bit Data Input
    .DIPB(DIPB),         // Port-B 1-bit parity Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),            // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);

// End of RAMB16_S1_S9_inst instantiation

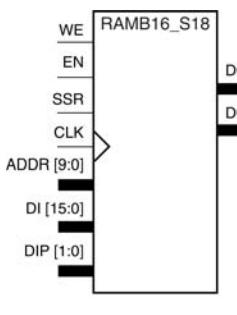
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S18

**Primitive: 16K-bit Data + 2K-bit Parity Memory, Single-Port Synchronous Block RAM with 18-bit Port**



### Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

Data Cells		Parity Cells				
Depth	Width	Depth	Width	Address Bus	Data Bus	Parity Bus
1024	16	1024	2	(9:0)	(15:0)	(1:0)

The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

### Logic Table

Inputs									Outputs		
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
									Data RAM		Parity RAM
1	X	X	X	X	X	X	X	INIT	INIT	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM (addr)	RAM (addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change(a) RAM	No Change(a) RAM	RAM (addr)=>data	RAM (addr)=>pdata

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
								(addr)(b) data(c)	(addr)(b) pdata(c)		

GSR=Global Set Reset signal  
INIT=Value specified by the INIT attribute for data memory. Default is all zeros.  
SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
(a) WRITE\_MODE=NO\_CHANGE  
(b) WRITE\_MODE=READ\_FIRST  
(c) WRITE\_MODE=WRITE\_FIRST

## Initialization

### *Initializing Memory Contents*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

Any INIT\_xx or INITP\_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register*

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16\_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16\_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

### *Selecting Write Mode*

The WRITE\_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE\_MODE is set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE\_MODE to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE\_MODE to NO\_CHANGE to have the input written to memory without changing the output.

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary/ Hexadecimal	Any Hex Value	All zeros	Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM.
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the data portion of the RAM array.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SRVAL	Binary/ Hexadecimal	Any Hex Value	All zeros	Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM.
WRITE_MODE	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

INITP_04 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INITP_06 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
port map (
    DO => DO,          -- 16-bit Data Output
    DOP => DOP,         -- 2-bit parity Output
    ADDR => ADDR,       -- 10-bit Address Input
    CLK => CLK,         -- Clock
    DI => DI,          -- 16-bit Data Input
    DIP => DIP,         -- 2-bit parity Input
    EN => EN,           -- RAM Enable Input
    SSR => SSR,         -- Synchronous Set/Reset Input
    WE => WE           -- Write Enable Input
);
-- End of RAMB16_S18_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S18: Spartan-3/3E 1k x 16 + 2 Parity bits Single-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S18 #(
    .INIT(18'h00000),   // Value of output RAM registers at startup
    .SRVAL(18'h00000),  // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 255
    .INIT_00(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_01(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_02(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_03(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_04(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_05(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_06(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_07(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_08(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_09(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 256 to 511
    .INIT_10(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_11(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_12(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_13(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_14(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_15(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_16(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_17(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_18(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_19(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 512 to 767
    .INIT_20(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_21(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_22(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_23(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_24(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_25(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_26(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000)
);

```

```

.INIT_27(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_28(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_29(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 768 to 1023
.INIT_30(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_31(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_32(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_33(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_34(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_35(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_36(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_37(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_38(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_39(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// The next set of INITP_xx are for the parity bits
// Address 0 to 255
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 511
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 512 to 767
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 768 to 1023
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S18_inst (
    .DO(DO),           // 16-bit Data Output
    .DOP(DOP),         // 2-bit parity Output
    .ADDR(ADDR),       // 10-bit Address Input
    .CLK(CLK),         // Clock
    .DI(DI),           // 16-bit Data Input
    .DIP(DIP),         // 2-bit parity Input
    .EN(EN),           // RAM Enable Input
    .SSR(SSR),         // Synchronous Set/Reset Input
    .WE(WE)            // Write Enable Input
);
// End of RAMB16_S18_inst instantiation

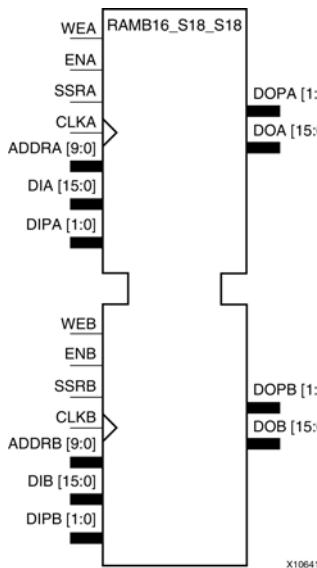
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S18\_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
								Data RAM	Parity RAM	

addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S18_S18	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

Start=((ADDR port+1)\*(Widthport)) -1

End=(ADDRport)\*(Widthport)

The following tables show address mapping for each port width.

Port Address Mapping for Data

Port Address Mapping for Parity

## Initializing Memory Contents of a Dual-Port RAMB16

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP 00 through INITP 07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A or above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 16-bit Data Output
DOB => DOB,      -- Port B 16-bit Data Output
DOPA => DOPA,    -- Port A 2-bit Parity Output
DOPB => DOPB,    -- Port B 2-bit Parity Output
ADDRA => ADDRA,  -- Port A 10-bit Address Input
ADDRB => ADDRB,  -- Port B 10-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 16-bit Data Input
DIB => DIB,      -- Port B 16-bit Data Input
DIPA => DIPA,    -- Port A 2-bit parity Input
DIPB => DIPB,    -- Port-B 2-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S18_S18_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S18_S18: Spartan-3/3E 1k x 16 + 2 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S18_S18 #(
    .INIT_A(18'h00000), // Value of output RAM registers on Port A at startup
    .INIT_B(18'h00000), // Value of output RAM registers on Port B at startup
    .SRVAL_A(18'h00000), // Port A output value upon SSR assertion
    .SRVAL_B(18'h00000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 255
    .INIT_00(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_01(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_02(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_03(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_04(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_05(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_06(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_07(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_08(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_09(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_0F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 256 to 511
    .INIT_10(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_11(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_12(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_13(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_14(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_15(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_16(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_17(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_18(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_19(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    .INIT_1F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
    // Address 512 to 767
)

```

```

.INIT_20(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_21(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_22(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_23(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_24(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_25(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_26(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_27(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_28(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_29(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_2F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
// Address 768 to 1023
.INIT_30(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_31(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_32(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_33(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_34(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_35(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_36(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_37(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_38(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_39(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3A(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3B(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3C(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3D(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3E(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),
.INIT_3F(256'h0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000_0000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 255
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 511
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 512 to 767
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 768 to 1023
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S18_S18_inst (
    .DOA(DOA),           // Port A 16-bit Data Output
    .DOB(DOB),           // Port B 16-bit Data Output
    .DOPA(DOPA),         // Port A 2-bit Parity Output
    .DOPB(DOPB),         // Port B 2-bit Parity Output
    .ADDRA(ADDRA),       // Port A 10-bit Address Input
    .ADDRB(ADDRB),       // Port B 10-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 16-bit Data Input
    .DIB(DIB),           // Port B 16-bit Data Input
    .DIPA(DIPA),         // Port A 2-bit parity Input
    .DIPB(DIPB),         // Port-B 2-bit parity Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);

// End of RAMB16_S18_S18_inst instantiation

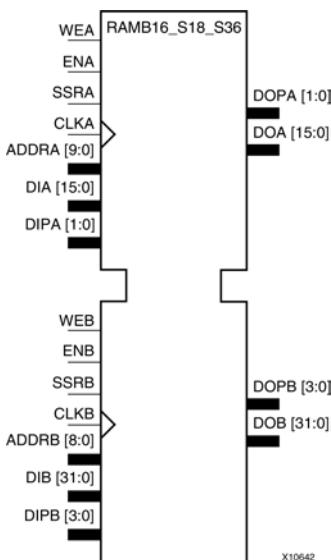
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S18\_S36

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 18-bit and 36-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>pdata	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>pdata	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
								Data RAM	Parity RAM	

RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S18_S36	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																	
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
4	4096	<--	7				6				5				4			3			2				1							0		
8	2048	<--	3								2							1							0									
16	1024	<--	1															0																
32	512	<--	0																															

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																															
1	2048	<----	3							2							1								0							
2	1024	<----	1														0															
4	512	<----	0																													

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,          -- Port A 16-bit Data Output
DOB => DOB,          -- Port B 32-bit Data Output
DOPA => DOPA,         -- Port A 2-bit Parity Output
DOPB => DOPB,         -- Port B 4-bit Parity Output
ADDRA => ADDRA,       -- Port A 10-bit Address Input
ADDRB => ADDRB,       -- Port B 9-bit Address Input
CLKA => CLKA,         -- Port A Clock
CLKB => CLKB,         -- Port B Clock
DIA => DIA,           -- Port A 16-bit Data Input
DIB => DIB,           -- Port B 32-bit Data Input
DIPA => DIPA,          -- Port A 2-bit parity Input
DIPB => DIPB,          -- Port-B 4-bit parity Input
ENA => ENA,            -- Port A RAM Enable Input
ENB => ENB,            -- PortB RAM Enable Input
SSRA => SSRA,          -- Port A Synchronous Set/Reset Input
SSRB => SSRB,          -- Port B Synchronous Set/Reset Input
WEA => WEA,             -- Port A Write Enable Input
WEB => WEB,             -- Port B Write Enable Input
);

-- End of RAMB16_S18_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S18_S36: Spartan-3/3E 1k/512 x 16/32 + 2/4 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S18_S36 #(
    .INIT_A(18'h00000),      // Value of output RAM registers on Port A at startup
    .INIT_B(36'h0000000000),  // Value of output RAM registers on Port B at startup
    .SRVAL_A(18'h00000),     // Port A output value upon SSR assertion
    .SRVAL_B(36'h0000000000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 255, Port B Address 0 to 127
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 256 to 511, Port B Address 128 to 255
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000)
    // Port A Address 512 to 767, Port B Address 256 to 383

```

```

.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 255, Port B Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S18_S36_inst (
    .DOA(DOA),           // Port A 16-bit Data Output
    .DOB(DOB),           // Port B 32-bit Data Output
    .DOPA(DOPA),         // Port A 2-bit Parity Output
    .DOPB(DOPB),         // Port B 4-bit Parity Output
    .ADDRA(ADDRA),       // Port A 10-bit Address Input
    .ADDRB(ADDRB),       // Port B 9-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 16-bit Data Input
    .DIB(DIB),           // Port B 32-bit Data Input
    .DIPA(DIPA),         // Port A 2-bit parity Input
    .DIPB(DIPB),         // Port-B 4-bit parity Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),            // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);

// End of RAMB16_S18_S36_inst instantiation

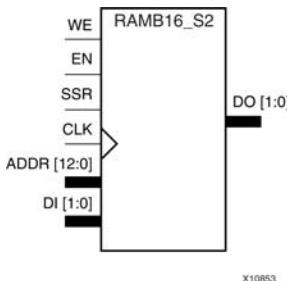
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S2

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 2-bit Port



### Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

Data Cells		Parity Cells				
Depth	Width	Depth	Width	Address Bus	Data Bus	Parity Bus
8192	2	-	-	(12:0)	(1:0)	-

The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDR\_A) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDR\_A) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

### Logic Table

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT	INIT	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM (addr)	RAM (addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change(a) RAM (addr)(b) data(c)	No Change(a) RAM (addr)(b) pdata(c)	RAM (addr)=>data	RAM (addr)=>pdata

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
GSR=Global Set Reset signal											
INIT=Value specified by the INIT attribute for data memory. Default is all zeros.											
SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.											
addr=RAM address.											
RAM(addr)=RAM contents at address ADDR.											
data=RAM input data.											
pdata=RAM parity data.											
(a) WRITE_MODE=NO_CHANGE											
(b) WRITE_MODE=READ_FIRST											
(c) WRITE_MODE=WRITE_FIRST											

## Initialization

### *Initializing Memory Contents*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

Any INIT\_xx or INITP\_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register*

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16\_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16\_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

### *Selecting Write Mode*

The WRITE\_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE\_MODE is set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE\_MODE to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE\_MODE to NO\_CHANGE to have the input written to memory without changing the output.

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary/ Hexadecimal	Any Hex Value	All zeros	Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM.
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the data portion of the RAM array.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SRVAL	Binary/ Hexadecimal	Any Hex Value	All zeros	Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM.
WRITE_MODE	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```
) ;

-- End of RAMB16_S2_inst instantiation
```

## Verilog Instantiation Template

```
// RAMB16_S2: Spartan-3E/3A/3AN/3AD 8k x 2 Single-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S2 #(
    .INIT(2'b00), // Value of output RAM registers at startup
    .SRVAL(2'b00), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 2047
    .INIT_00(256'h00000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000),
    // Address 2048 to 4095
    .INIT_10(256'h00000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000),
    .INIT_1A(256'h00000000000000000000000000000000),
    .INIT_1B(256'h00000000000000000000000000000000),
    .INIT_1C(256'h00000000000000000000000000000000),
    .INIT_1D(256'h00000000000000000000000000000000),
    .INIT_1E(256'h00000000000000000000000000000000),
    .INIT_1F(256'h00000000000000000000000000000000),
    // Address 4096 to 6143
    .INIT_20(256'h00000000000000000000000000000000),
    .INIT_21(256'h00000000000000000000000000000000),
    .INIT_22(256'h00000000000000000000000000000000),
    .INIT_23(256'h00000000000000000000000000000000),
    .INIT_24(256'h00000000000000000000000000000000),
    .INIT_25(256'h00000000000000000000000000000000),
    .INIT_26(256'h00000000000000000000000000000000),
    .INIT_27(256'h00000000000000000000000000000000),
    .INIT_28(256'h00000000000000000000000000000000),
    .INIT_29(256'h00000000000000000000000000000000),
    .INIT_2A(256'h00000000000000000000000000000000),
    .INIT_2B(256'h00000000000000000000000000000000),
    .INIT_2C(256'h00000000000000000000000000000000),
    .INIT_2D(256'h00000000000000000000000000000000),
    .INIT_2E(256'h00000000000000000000000000000000),
    .INIT_2F(256'h00000000000000000000000000000000),
    // Address 6143 to 8191
    .INIT_30(256'h00000000000000000000000000000000),
    .INIT_31(256'h00000000000000000000000000000000),
    .INIT_32(256'h00000000000000000000000000000000),
    .INIT_33(256'h00000000000000000000000000000000),
    .INIT_34(256'h00000000000000000000000000000000)
```

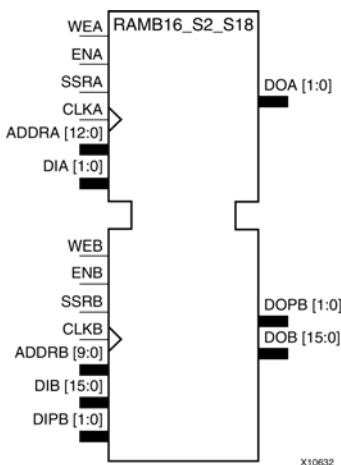
```
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S2_inst (
    .DO(DO),           // 2-bit Data Output
    .ADDR(ADDR),       // 13-bit Address Input
    .CLK(CLK),         // Clock
    .DI(DI),           // 2-bit Data Input
    .EN(EN),           // RAM Enable Input
    .SSR(SSR),         // Synchronous Set/Reset Input
    .WE(WE)            // Write Enable Input
);
// End of RAMB16_S2_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S2\_S18

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 18-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDR A	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
pdata=RAM parity data. 1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S2_S18	8192 x 2	-	(12:0)	(1:0)	-	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<-	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1	0				
4	4096	<-	7			6			5				4				3				2				1				0							
8	2048	<-	3							2								1								0										
16	1024	<-	1																0																	
32	512	<-	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																			
1	2048	<----	3						2								1								0											
2	1024	<----	1															0																		
4	512	<----	0																																	

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectIO™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

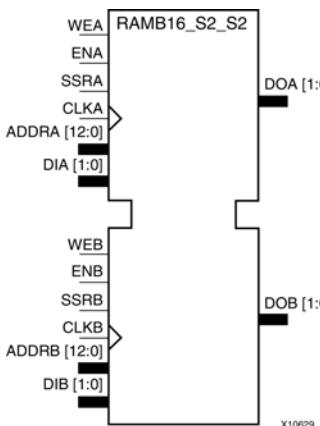
Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S2\_S2

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs								Outputs			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
								Data RAM	Parity RAM		
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S2_S2	8192 x 2	-	(12:0)	(1:0)	-	8192 x 2	-	(12:0)	(1:0)	-	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																				
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0				
4	4096	<--	7			6				5				4				3				2			1									0			
8	2048	<--	3								2								1																0		
16	1024	<--	1																	0																	
32	512	<--	0																																		

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																				
1	2048	<----	3						2						1																	0					
2	1024	<----	1													0																					
4	512	<----	0																																		

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", or "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the</p>

Attribute	Type	Allowed Values	Default	Description
				value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.
SRVAL_A	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S2: Spartan-3/3E 8k x 2 Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16_S2_S2_inst : RAMB16_S2_S2
generic map (
    INIT_A => X"0", -- Value of output RAM registers on Port A at startup
    INIT_B => X"0", -- Value of output RAM registers on Port B at startup
    SRVAL_A => X"0", -- Port A output value upon SSR assertion
    SRVAL_B => X"0", -- Port B output value upon SSR assertion

```



```

DOA => DOA,      -- Port A 2-bit Data Output
DOB => DOB,      -- Port B 2-bit Data Output
ADDRA => ADDR_A, -- Port A 13-bit Address Input
ADDRB => ADDR_B, -- Port B 13-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 2-bit Data Input
DIB => DIB,      -- Port B 2-bit Data Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- Port B RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S2_S2_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S2_S2: Spartan-3E/3A/3AD 8k x 2 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S2_S2 #(
    .INIT_A(2'b00), // Value of output RAM registers on Port A at startup
    .INIT_B(2'b00), // Value of output RAM registers on Port B at startup
    .SRVAL_A(2'b00), // Port A output value upon SSR assertion
    .SRVAL_B(2'b00), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
    // Address 0 to 2047
    .INIT_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    // Address 2048 to 4095
    .INIT_10(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    // Address 4096 to 6143
    .INIT_20(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000)
);

```

```

.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 6143 to 8191
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S2_S2_inst (
    .DOA(DOA),           // Port A 2-bit Data Output
    .DOB(DOB),           // Port B 2-bit Data Output
    .ADDRA(ADDRA),       // Port A 13-bit Address Input
    .ADDRB(ADDRB),       // Port B 13-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 2-bit Data Input
    .DIB(DIB),           // Port B 2-bit Data Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),          // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),          // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);
// End of RAMB16_S2_S2_inst instantiation

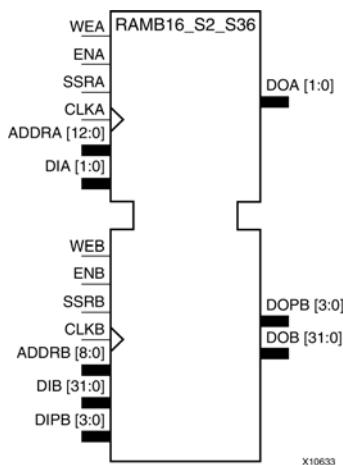
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S2\_S36

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 36-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDR A	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
pdata=RAM parity data. 1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S2_S36	8192 x 2	-	(12:0)	(1:0)	-	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																				
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0				
4	4096	<--	7			6			5			4				3				2				1										0			
8	2048	<--	3							2								1																	0		
16	1024	<--	1																0																		
32	512	<--	0																																		

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																				
1	2048	<----	3						2						1																	0					
2	1024	<----	1													0																					
4	512	<----	0																																		

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 2-bit Data Output
DOB => DOB,      -- Port B 32-bit Data Output
DOPB => DOPB,    -- Port B 4-bit Parity Output
ADDRA => ADDRA,  -- Port A 13-bit Address Input
ADDRB => ADDRB,  -- Port B 9-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 2-bit Data Input
DIB => DIB,      -- Port B 32-bit Data Input
DIPB => DIPB,    -- Port-B 4-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S2_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S2_S36: Spartan-3E/3A/3AN/3AD 8k/512 x 2/32 + 0/4 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S2_S36 #(
    .INIT_A(2'b00),           // Value of output RAM registers on Port A at startup
    .INIT_B(36'h0000000000),  // Value of output RAM registers on Port B at startup
    .SRVAL_A(2'b00),          // Port A output value upon SSR assertion
    .SRVAL_B(36'h0000000000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 2047, Port B Address 0 to 127
    .INIT_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

    // Port A Address 2048 to 4095, Port B Address 128 to 255
    .INIT_10(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

    // Port A Address 4096 to 6143, Port B Address 256 to 383
    .INIT_20(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
);

```

```

.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 6144 to 8191, Port B Address 384 to 511
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 127
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 128 to 255
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 256 to 383
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 384 to 511
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S2_S36_inst (
    .DOA(DOA),           // Port A 2-bit Data Output
    .DOB(DOB),           // Port B 32-bit Data Output
    .DOPB(DOPB),         // Port B 4-bit Parity Output
    .ADDRA(ADDRA),       // Port A 13-bit Address Input
    .ADDRB(ADDRB),       // Port B 9-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 2-bit Data Input
    .DIB(DIB),           // Port B 32-bit Data Input
    .DIPB(DIPB),         // Port-B 4-bit parity Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);

// End of RAMB16_S2_S36_inst instantiation

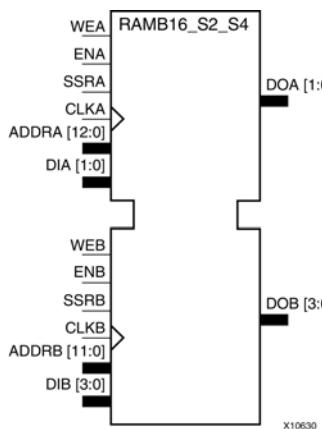
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S2\_S4

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 4-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells (a)	Parity Cells (a)	Address Bus	Data Bus	Parity Bus	Data Cells (a)	Parity Cells (a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S2_S4	8192 x 2	-	(12:0)	(1:0)	-	4096 x 4	-	(11:0)	(3:0)	-	
(a) Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																					
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0					
4	4096	<--	7			6			5			4					3				2			1														
8	2048	<--	3							2									1																		0	
16	1024	<--	1																	0																		
32	512	<--	0																																			

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																					
1	2048	<----	3						2							1																			0			
2	1024	<----	1														0																					
4	512	<----	0																																			

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "NONE", "WARNING", or "GENERATE_X_ONLY"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

WEB => WEB      -- Port B Write Enable Input
);

-- End of RAMB16_S2_S4_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S2_S4: Spartan-3E/3A/3AN/3AD 8k/4k x 2/4 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S2_S4 #(
    .INIT_A(2'b00), // Value of output RAM registers on Port A at startup
    .INIT_B(4'h0), // Value of output RAM registers on Port B at startup
    .SRVAL_A(2'b00), // Port A output value upon SSR assertion
    .SRVAL_B(4'h0), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 2047, Port B Address 0 to 1023
    .INIT_00(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 2048 to 4095, Port B Address 1024 to 2047
    .INIT_10(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 4096 to 6143, Port B Address 2048 to 3071
    .INIT_20(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_22(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_23(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_24(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_25(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_26(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_27(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_28(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_29(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2A(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2B(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2C(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2D(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2E(256'h000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2F(256'h000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 6144 to 8191, Port B Address 3072 to 4095

```

```

.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S2_S4_inst (
    .DOA(DOA),           // Port A 2-bit Data Output
    .DOB(DOB),           // Port B 4-bit Data Output
    .ADDRA(ADDRA),       // Port A 13-bit Address Input
    .ADDRB(ADDRB),       // Port B 12-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 2-bit Data Input
    .DIB(DIB),           // Port B 4-bit Data Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);
// End of RAMB16_S2_S4_inst instantiation

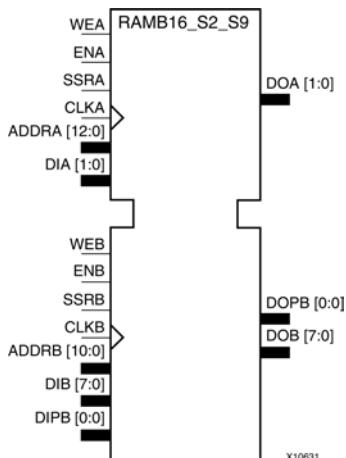
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S2\_S9

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 2-bit and 9-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDR A	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
pdata=RAM parity data. 1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S2_S9	8192 x 2	-	(12:0)	(1:0)	-	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																	
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
4	4096	<--	7				6				5				4				3				2			1							0	
8	2048	<--	3									2								1													0	
16	1024	<--	1																	0														
32	512	<--	0																															

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																													
1	2048	<----	3							2							1												0	
2	1024	<----	1															0												
4	512	<----	0																											

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan®-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexidecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexidecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "NONE", "WARNING", or "GENERATE_X_ONLY"	"ALL"	Specifies the behavior during simulation in the event of a data collision (data being read or written to the same address from both ports of the Ram simultaneously. "ALL" issues a warning to simulator console and generate an X or all unknown data due to the collision. This is the recommended setting. "WARNING" generates a warning only and "GENERATE_X_ONLY" generates an X for unknown data but won't output the occurrence to the simulation console. "NONE" completely ignores the error. It is suggested to only change this attribute if you can ensure the data generated during a collision is discarded.

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexidecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16_S2_S9: Spartan-3E 8k/2k x 2/8 + 0/1 Parity bit Dual-Port RAM
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16_S2_S9_inst : RAMB16_S2_S9
generic map (
    INIT_A => X"0", -- Value of output RAM registers on Port A at startup
    INIT_B => X"000", -- Value of output RAM registers on Port B at startup
    SRVAL_A => X"0", -- Port A output value upon SSR assertion
    SRVAL_B => X"000", -- Port B output value upon SSR assertion
    WRITE_MODE_A => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
    WRITE_MODE_B => "WRITE_FIRST", -- WRITE_FIRST, READ_FIRST or NO_CHANGE
    SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
```



# Verilog Instantiation Template

// RAMB16\_S2\_S9: Spartan-3/3E/3A/3AN/3AD 8k/2k x 2/8 + 0/1 Parity bit Dual-Port RAM  
// Xilinx HDL Libraries Guide, version 11.2



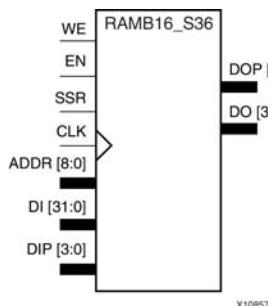
```
// End of RAMB16_S2_S9_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S36

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 36-bit Port



### Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

Data Cells		Parity Cells				
Depth	Width	Depth	Width	Address Bus	Data Bus	Parity Bus
512	32	512	4	(8:0)	(31:0)	(3:0)

The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDR) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDR) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

### Logic Table

Inputs									Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT	INIT	No Change	No Change	
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change	
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Change	No Change	
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata	
0	1	0	0	↑	addr	X	X	RAM (addr)	RAM (addr)	No Change	No Change	
0	1	0	1	↑	addr	data	pdata	No Change(a) RAM	No Change(a) RAM	RAM (addr)=>data	RAM (addr)=>pdata	

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
								(addr)(b) data(c)	(addr)(b) pdata(c)		

GSR=Global Set Reset signal  
INIT=Value specified by the INIT attribute for data memory. Default is all zeros.  
SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
(a) WRITE\_MODE=NO\_CHANGE  
(b) WRITE\_MODE=READ\_FIRST  
(c) WRITE\_MODE=WRITE\_FIRST

## Initialization

### *Initializing Memory Contents*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

Any INIT\_xx or INITP\_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register*

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16\_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16\_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

### *Selecting Write Mode*

The WRITE\_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE\_MODE is set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE\_MODE to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE\_MODE to NO\_CHANGE to have the input written to memory without changing the output.

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary/ Hexadecimal	Any Hex Value	All zeros	Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM.
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the data portion of the RAM array.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SRVAL	Binary/ Hexadecimal	Any Hex Value	All zeros	Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM.
WRITE_MODE	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

INITP_04 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INITP_06 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000"
port map (
    DO => DO,          -- 32-bit Data Output
    DOP => DOP,         -- 4-bit parity Output
    ADDR => ADDR,       -- 9-bit Address Input
    CLK => CLK,          -- Clock
    DI => DI,          -- 32-bit Data Input
    DIP => DIP,         -- 4-bit parity Input
    EN => EN,          -- RAM Enable Input
    SSR => SSR,        -- Synchronous Set/Reset Input
    WE => WE           -- Write Enable Input
);
-- End of RAMB16_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S36: Spartan-3E 512 x 32 + 4 Parity bits Single-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S36 #(
    .INIT(36'h00000000), // Value of output RAM registers at startup
    .SRVAL(36'h00000000), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 127
    .INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_04(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_05(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_06(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_07(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_08(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_09(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 128 to 255
    .INIT_10(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_11(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_12(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_13(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_14(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_15(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_16(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_17(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_18(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_19(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 256 to 383
    .INIT_20(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_21(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_22(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_23(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_24(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_25(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_26(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000)
);

```

```

.INIT_27(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_28(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_29(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 384 to 511
.INIT_30(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_31(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_32(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_33(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_34(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_35(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_36(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_37(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_38(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_39(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// The next set of INITP_xx are for the parity bits
// Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S36_inst (
    .DO(DO),           // 32-bit Data Output
    .DOP(DOP),         // 4-bit parity Output
    .ADDR(ADDR),       // 9-bit Address Input
    .CLK(CLK),         // Clock
    .DI(DI),           // 32-bit Data Input
    .DIP(DIP),         // 4-bit parity Input
    .EN(EN),           // RAM Enable Input
    .SSR(SSR),         // Synchronous Set/Reset Input
    .WE(WE)            // Write Enable Input
);
// End of RAMB16_S36_inst instantiation

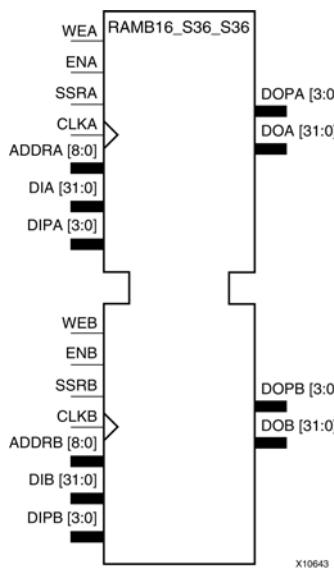
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S36\_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with Two 36-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S36_S36	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables show address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																		
1	16384	<-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<-	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<-	7																																0
8	2048	<-	3																																0
16	1024	<-	1																																0
32	512	<-	0																																0

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																		
1	2048	<---	3														1															0			
2	1024	<---	1														0																		
4	512	<---	0																																

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 32-bit Data Output
DOB => DOB,      -- Port B 32-bit Data Output
DOPA => DOPA,    -- Port A 4-bit Parity Output
DOPB => DOPB,    -- Port B 4-bit Parity Output
ADDRA => ADDRA,  -- Port A 9-bit Address Input
ADDRB => ADDRB,  -- Port B 9-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 32-bit Data Input
DIB => DIB,      -- Port B 32-bit Data Input
DIPA => DIPA,    -- Port A 4-bit parity Input
DIPB => DIPB,    -- Port-B 4-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S36_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S36_S36: Spartan-3/3E 512 x 32 + 4 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S36_S36 #(
    .INIT_A(36'h0000000000), // Value of output RAM registers on Port A at startup
    .INIT_B(36'h0000000000), // Value of output RAM registers on Port B at startup
    .SRVAL_A(36'h0000000000), // Port A output value upon SSR assertion
    .SRVAL_B(36'h0000000000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 127
    .INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_04(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_05(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_06(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_07(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_08(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_09(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_0F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 128 to 255
    .INIT_10(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_11(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_12(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_13(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_14(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_15(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_16(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_17(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_18(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_19(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    .INIT_1F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
    // Address 256 to 383
)

```

```

.INIT_20(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_21(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_22(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_23(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_24(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_25(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_26(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_27(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_28(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_29(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_2F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
// Address 384 to 511
.INIT_30(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_31(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_32(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_33(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_34(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_35(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_36(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_37(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_38(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_39(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
.INIT_3F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),

// The next set of INITP_xx are for the parity bits
// Address 0 to 127
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 383
.INITP_04(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 384 to 511
.INITP_06(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S36_S36_inst (
    .DOA(DOA),           // Port A 32-bit Data Output
    .DOB(DOB),           // Port B 32-bit Data Output
    .DOPA(DOPA),         // Port A 4-bit Parity Output
    .DOPB(DOPB),         // Port B 4-bit Parity Output
    .ADDRA(ADDRA),       // Port A 9-bit Address Input
    .ADDRB(ADDRB),       // Port B 9-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 32-bit Data Input
    .DIB(DIB),           // Port B 32-bit Data Input
    .DIPA(DIPA),         // Port A 4-bit parity Input
    .DIPB(DIPB),         // Port-B 4-bit parity Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);

// End of RAMB16_S36_S36_inst instantiation

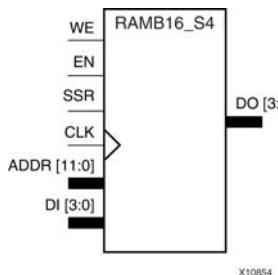
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S4

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 4-bit Port



## Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

Data Cells		Parity Cells				
Depth	Width	Depth	Width	Address Bus	Data Bus	Parity Bus
4096	4	-	-	(11:0)	(3:0)	-

The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDR\_A) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDR\_A) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

## Logic Table

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT	INIT	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM (addr)	RAM (addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change(a) RAM (addr)(b) data(c)	No Change(a) RAM (addr)(b) pdata(c)	RAM (addr)=>data	RAM (addr)=>pdata

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
GSR=Global Set Reset signal											
INIT=Value specified by the INIT attribute for data memory. Default is all zeros.											
SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.											
addr=RAM address.											
RAM(addr)=RAM contents at address ADDR.											
data=RAM input data.											
pdata=RAM parity data.											
(a) WRITE_MODE=NO_CHANGE											
(b) WRITE_MODE=READ_FIRST											
(c) WRITE_MODE=WRITE_FIRST											

## Initialization

### *Initializing Memory Contents*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

Any INIT\_xx or INITP\_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register*

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16\_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16\_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

### *Selecting Write Mode*

The WRITE\_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE\_MODE is set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE\_MODE to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE\_MODE to NO\_CHANGE to have the input written to memory without changing the output.

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary/ Hexadecimal	Any Hex Value	All zeros	Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM.
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the data portion of the RAM array.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SRVAL	Binary/ Hexadecimal	Any Hex Value	All zeros	Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM.
WRITE_MODE	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```
) ;

-- End of RAMB16_S4_inst instantiation
```

## Verilog Instantiation Template

```
// RAMB16_S4: Spartan-3E/3A/3AN/3AD 4k x 4 Single-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S4 #(
    .INIT(4'h0), // Value of output RAM registers at startup
    .SRVAL(4'h0), // Output value upon SSR assertion
    .WRITE_MODE("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 1023
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 1024 to 2047
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 2048 to 3071
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 3072 to 4095
    .INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000)
```

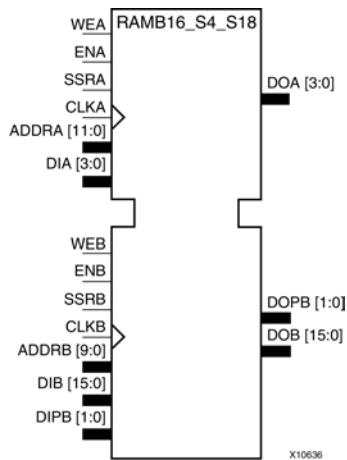
```
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S4_inst (
    .DO(DO),           // 4-bit Data Output
    .ADDR(ADDR),       // 12-bit Address Input
    .CLK(CLK),         // Clock
    .DI(DI),           // 4-bit Data Input
    .EN(EN),           // RAM Enable Input
    .SSR(SSR),         // Synchronous Set/Reset Input
    .WE(WE)            // Write Enable Input
);
// End of RAMB16_S4_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S4\_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 18-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.  
SRVAL\_A=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S4_S18	4096 x 4	-	(11:0)	(3:0)	-	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7			6				5			4				3				2			1										0		
8	2048	<--	3								2							1																	0	
16	1024	<--	1																0																	
32	512	<--	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																
1	2048	<----	3						2							1															0		
2	1024	<----	1														0																
4	512	<----	0																														

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

The following tables show address mapping for each port width.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 4-bit Data Output
DOB => DOB,      -- Port B 16-bit Data Output
DOPB => DOPB,    -- Port B 2-bit Parity Output
ADDRA => ADDRA,  -- Port A 12-bit Address Input
ADDRB => ADDRB,  -- Port B 10-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 4-bit Data Input
DIB => DIB,      -- Port B 16-bit Data Input
DIPB => DIPB,    -- Port-B 2-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S4_S18_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S4_S18: Spartan-3E/3A/3AN/3AD 4k/1k x 4/16 + 0/2 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S4_S18 #(
    .INIT_A(4'h0),          // Value of output RAM registers on Port A at startup
    .INIT_B(18'h00000),     // Value of output RAM registers on Port B at startup
    .SRVAL_A(4'h0),         // Port A output value upon SSR assertion
    .SRVAL_B(18'h00000),   // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 1023, Port B Address 0 to 255
    .INIT_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

    // Port A Address 1024 to 2047, Port B Address 256 to 511
    .INIT_10(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

    // Port A Address 2048 to 3071, Port B Address 512 to 767
    .INIT_20(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
);

```

```

.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 3072 to 4095, Port B Address 768 to 1023
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 1023, Port B Address 0 to 255
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1024 to 2047, Port B Address 256 to 511
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 2048 to 3071, Port B Address 512 to 767
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 3072 to 4095, Port B Address 768 to 1023
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S4_S18_inst (
    .DOA(DOA),           // Port A 4-bit Data Output
    .DOB(DOB),           // Port B 16-bit Data Output
    .DOPB(DOPB),         // Port B 2-bit Parity Output
    .ADDRA(ADDRA),       // Port A 12-bit Address Input
    .ADDRB(ADDRB),       // Port B 10-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 4-bit Data Input
    .DIB(DIB),           // Port B 16-bit Data Input
    .DIPB(DIPB),         // Port-B 2-bit parity Input
    .ENA(ENA),            // Port A RAM Enable Input
    .ENB(ENB),            // Port B RAM Enable Input
    .SSRA(SSRA),          // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),          // Port B Synchronous Set/Reset Input
    .WEA(WEA),             // Port A Write Enable Input
    .WEB(WEB)             // Port B Write Enable Input
);
// End of RAMB16_S4_S18_inst instantiation

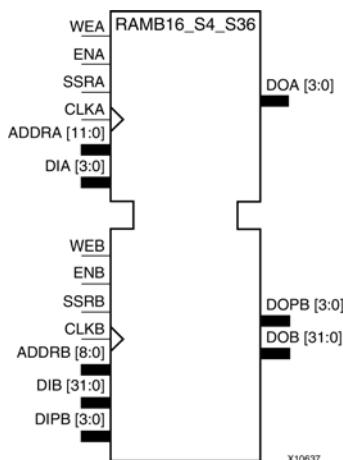
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S4\_S36

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 36-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDR A	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
pdata=RAM parity data. 1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S4_S36	4096 x 4	-	(11:0)	(3:0)	-	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7			6				5				4				3				2				1							0		
8	2048	<--	3								2								1															0	
16	1024	<--	1																	0															
32	512	<--	0																																

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																															
1	2048	<----	3						2							1														0		
2	1024	<----	1														0															
4	512	<----	0																													

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 4-bit Data Output
DOB => DOB,      -- Port B 32-bit Data Output
DOPB => DOPB,    -- Port B 4-bit Parity Output
ADRA => ADRA,    -- Port A 12-bit Address Input
ADDRB => ADDR,   -- Port B 9-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 4-bit Data Input
DIB => DIB,      -- Port B 32-bit Data Input
DIPB => DIPB,    -- Port-B 4-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S4_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S4_S36: Spartan-3E/3A/3AN/3AD 4k/512 x 4/32 + 0/4 Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S4_S36 #(
    .INIT_A(4'h0),           // Value of output RAM registers on Port A at startup
    .INIT_B(36'h0000000000000000000000000000000000000000000000000000000000000000),
    .SRVAL_A(4'h0),          // Port A output value upon SSR assertion
    .SRVAL_B(36'h0000000000000000000000000000000000000000000000000000000000000000),
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 1023, Port B Address 0 to 127
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),

    // Port A Address 1024 to 2047, Port B Address 128 to 255
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),

    // Port A Address 2048 to 3071, Port B Address 256 to 383
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000)
);

```

```

.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 3072 to 4095, Port B Address 384 to 511
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 1023, Port B Address 0 to 127
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1024 to 2047, Port B Address 128 to 255
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 2048 to 3071, Port B Address 256 to 383
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 3072 to 4095, Port B Address 384 to 511
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S4_S36_inst (
    .DOA(DOA),           // Port A 4-bit Data Output
    .DOB(DOB),           // Port B 32-bit Data Output
    .DOPB(DOPB),         // Port B 4-bit Parity Output
    .ADDRA(ADDRA),       // Port A 12-bit Address Input
    .ADDRB(ADDRB),       // Port B 9-bit Address Input
    .CLKA(CLKA),         // Port A Clock
    .CLKB(CLKB),         // Port B Clock
    .DIA(DIA),           // Port A 4-bit Data Input
    .DIB(DIB),           // Port B 32-bit Data Input
    .DIPB(DIPB),         // Port-B 4-bit parity Input
    .ENA(ENA),           // Port A RAM Enable Input
    .ENB(ENB),           // Port B RAM Enable Input
    .SSRA(SSRA),         // Port A Synchronous Set/Reset Input
    .SSRB(SSRB),         // Port B Synchronous Set/Reset Input
    .WEA(WEA),           // Port A Write Enable Input
    .WEB(WEB)            // Port B Write Enable Input
);
// End of RAMB16_S4_S36_inst instantiation

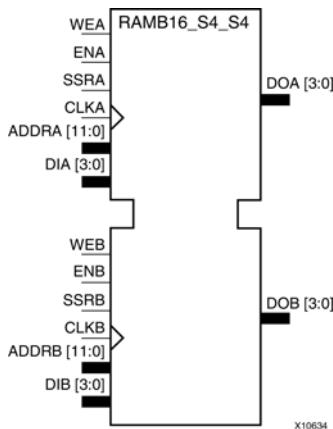
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S4\_S4

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs								Outputs			
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
								Data RAM	Parity RAM		
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

pdata=RAM parity data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
1WRITE_MODE_A=NO_CHANGE.										
2WRITE_MODE_A=READ_FIRST.										
3WRITE_MODE_A=WRITE_FIRST.								Data RAM    Parity RAM		

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S4_S4	4096 x 4	-	(11:0)	(3:0)	-	4096 x 4	-	(11:0)	(3:0)	-	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7			6				5				4				3				2			1									0		
8	2048	<--	3							2								1																	0	
16	1024	<--	1															0																		
32	512	<--	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																		
1	2048	<----	3						2						1																	0			
2	1024	<----	1													0																			
4	512	<----	0																																

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

<b>WEA</b>	<b>WEB</b>	<b>CLKA</b>	<b>CLKB</b>	<b>DIA</b>	<b>DIB</b>	<b>DIPA</b>	<b>DIPB</b>	<b>DOA</b>	<b>DOB</b>	<b>DOPA</b>	<b>DOPB</b>	<b>Data RAM</b>	<b>Parity RAM</b>
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

WEB => WEB      -- Port B Write Enable Input
);
-- End of RAMB16_S4_S4_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S4_S4: Spartan-3E/3A/3AN/3AD 4k x 4 Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S4_S4 #(
    .INIT_A(4'h0), // Value of output RAM registers on Port A at startup
    .INIT_B(4'h0), // Value of output RAM registers on Port B at startup
    .SRVAL_A(4'h0), // Port A output value upon SSR assertion
    .SRVAL_B(4'h0), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Address 0 to 1023
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 1024 to 2047
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 2048 to 3071
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Address 3072 to 4095
)

```

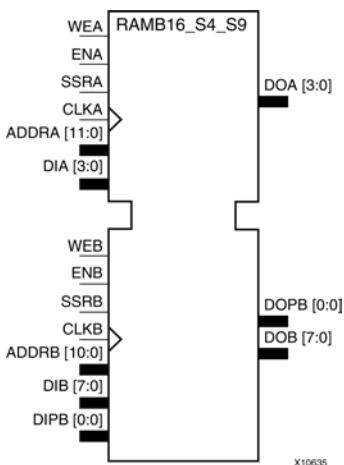
```
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S4_S4_inst (
.DOA(DOA),           // Port A 4-bit Data Output
.DOB(DOB),           // Port B 4-bit Data Output
.ADDRA(ADDRA),       // Port A 12-bit Address Input
.ADDRB(ADDRB),       // Port B 12-bit Address Input
.CLKA(CLKA),         // Port A Clock
.CLKB(CLKB),         // Port B Clock
.DIA(DIA),           // Port A 4-bit Data Input
.DIB(DIB),           // Port B 4-bit Data Input
.ENA(ENA),           // Port A RAM Enable Input
.ENB(ENB),           // Port B RAM Enable Input
.SSRA(SSRA),         // Port A Synchronous Set/Reset Input
.SSRB(SSRB),         // Port B Synchronous Set/Reset Input
.WEA(WEA),           // Port A Write Enable Input
.WEB(WEB)            // Port B Write Enable Input
);
// End of RAMB16_S4_S4_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S4\_S9

**Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 4-bit and 9-bit Ports**



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDR A	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

addr=RAM address.

RAM(addr)=RAM contents at address ADDR.

data=RAM input data.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
pdata=RAM parity data. 1WRITE_MODE_A=NO_CHANGE. 2WRITE_MODE_A=READ_FIRST. 3WRITE_MODE_A=WRITE_FIRST.								Data RAM	Parity RAM	

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S4_S9	4096 x 4	-	(11:0)	(3:0)	-	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7			6			5			4				3				2			1											0		
8	2048	<--	3							2									1																0	
16	1024	<--	1																0																	
32	512	<--	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																
1	2048	<----	3						2						1																0		
2	1024	<----	1													0																	
4	512	<----	0																														

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 4-bit Data Output
DOB => DOB,      -- Port B 8-bit Data Output
DOPB => DOPB,    -- Port B 1-bit Parity Output
ADRA => ADRA,    -- Port A 12-bit Address Input
ADDRB => ADDR,   -- Port B 11-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 4-bit Data Input
DIB => DIB,      -- Port B 8-bit Data Input
DIPB => DIPB,    -- Port-B 1-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S4_S9_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S4_S9: Spartan-3E/3A/3AN/3AD 4k/2k x 4/8 + 0/1 Parity bit Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S4_S9 #(
    .INIT_A(4'h0),      // Value of output RAM registers on Port A at startup
    .INIT_B(9'h000),    // Value of output RAM registers on Port B at startup
    .SRVAL_A(4'h0),    // Port A output value upon SSR assertion
    .SRVAL_B(9'h000),  // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 1023, Port B Address 0 to 511
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 1024 to 2047, Port B Address 512 to 1023
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 2048 to 3071, Port B Address 1024 to 1535
    .INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

```

.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 3072 to 4095, Port B Address 1536 to 2047
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// The next set of INITP_xx are for the parity bits
// Port B Address 0 to 511
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 512 to 1023
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 1024 to 1535
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port B Address 1536 to 2047
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
) RAMB16_S4_S9_inst (
.DOA(DOA), // Port A 4-bit Data Output
.DOB(DOB), // Port B 8-bit Data Output
.DOPB(DOPB), // Port B 1-bit Parity Output
.ADDRA(ADDRA), // Port A 12-bit Address Input
.ADDRB(ADDRB), // Port B 11-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 4-bit Data Input
.DIB(DIB), // Port B 8-bit Data Input
.DIPB(DIPB), // Port-B 1-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);
// End of RAMB16_S4_S9_inst instantiation

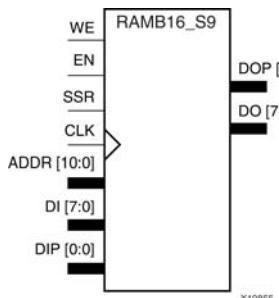
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S9

Primitive: 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with 9-bit Port



### Introduction

This design element is a dedicated random access memory block with synchronous write capability. The block RAM port has 16384 bits of data memory. The cell configuration for this element is listed in the following table.

Data Cells		Parity Cells				
Depth	Width	Depth	Width	Address Bus	Data Bus	Parity Bus
2048	8	2048	1	(10:0)	(7:0)	(0:0)

The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The above description assumes an active High EN, WE, SSR, and CLK. However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

### Logic Table

Inputs									Outputs		
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
									Data RAM		Parity RAM
1	X	X	X	X	X	X	X	INIT	INIT	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL	SRVAL	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL	SRVAL	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM (addr)	RAM (addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change(a) RAM	No Change(a) RAM	RAM (addr)=>data	RAM (addr)=>pdata

Inputs								Outputs			
GSR	EN	SSR	WE	CLK	ADDR	DI	DIP	DO	DOP	RAM Contents	
										Data RAM	Parity RAM
								(addr)(b) data(c)	(addr)(b) pdata(c)		

GSR=Global Set Reset signal  
INIT=Value specified by the INIT attribute for data memory. Default is all zeros.  
SRVAL=Value after assertion of SSR as specified by the SRVAL attribute.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
(a) WRITE\_MODE=NO\_CHANGE  
(b) WRITE\_MODE=READ\_FIRST  
(c) WRITE\_MODE=WRITE\_FIRST

## Initialization

### *Initializing Memory Contents*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16 is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

Any INIT\_xx or INITP\_xx attribute that is not specified is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register*

In Spartan®-3A and above devices, each bit in the output register can be initialized at power on to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Two types of properties control initialization of the output register for a single-port RAMB16: INIT and SRVAL. The INIT attribute specifies the output register value at power on. You can use the SRVAL attribute to define the state resulting from assertion of the SSR (set/reset) input.

The INIT and SRVAL attributes specify the initialization value as a hexadecimal String containing one bit for each bit in the output port. For example, for a RAMB16\_S1 with port width equal to 1, the output register contains 1 bit. Therefore, the INIT or SRVAL value can only be specified as a 1 or 0. For RAMB16\_S4 with port width equal to 4, the output register contains 4 bits. In this case, you can specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT or SRVAL value.

### *Selecting Write Mode*

The WRITE\_MODE attribute controls RAMB16 memory and output contents. By default, the WRITE\_MODE is set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the WRITE\_MODE to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the WRITE\_MODE to NO\_CHANGE to have the input written to memory without changing the output.

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Binary/ Hexadecimal	Any Hex Value	All zeros	Identifies the initial value of the DO output port after completing configuration. The bit width is dependent on the width of the A or B port of the RAM.
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the data portion of the RAM array.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any Hex Value	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SRVAL	Binary/ Hexadecimal	Any Hex Value	All zeros	Allows the individual selection of whether the DO output port sets (go to a one) or reset (go to a zero) upon the assertion of the SSR pin. The bit width is dependent on the width of the A or B port of the RAM.
WRITE_MODE	String	"WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DO port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



## Verilog Instantiation Template

```
// RAMB16_S9: Spartan-3/3E/3A/3AN/3AD 2k x 8 + 1 Parity bit Single-Port RAM
// Xilinx HDL Libraries Guide, version 11.2
```

```

.INIT_27(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_28(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_29(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2A(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2B(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2C(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2D(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2E(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2F(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
// Address 1536 to 2047
.INIT_30(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_31(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_32(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_33(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_34(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_35(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_36(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_37(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_38(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_39(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3A(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3B(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3C(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3D(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3E(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3F(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
) RAMB16_S9_inst (
    .DO(DO),          // 8-bit Data Output
    .DOP(DOP),        // 1-bit parity Output
    .ADDR(ADDR),      // 11-bit Address Input
    .CLK(CLK),        // Clock
    .DI(DI),          // 8-bit Data Input
    .DIP(DIP),        // 1-bit parity Input
    .EN(EN),          // RAM Enable Input
    .SSR(SSR),        // Synchronous Set/Reset Input
    .WE(WE)           // Write Enable Input
);
// End of RAMB16_S9_inst instantiation

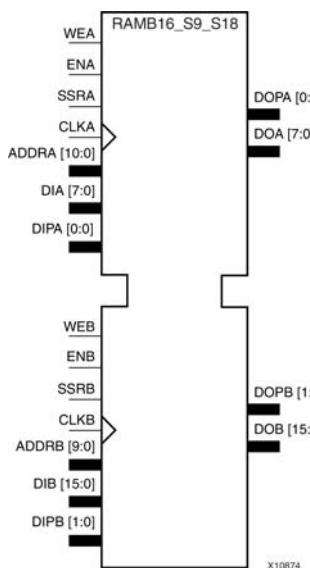
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## RAMB16\_S9\_S18

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 18-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
								Data RAM	Parity RAM	

addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Component	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S9_S18	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	1024 x 16	1024 x 2	(9:0)	(15:0)	(1:0)	
(a) Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables show address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																			
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0			
4	4096	<--	7			6				5				4				3			2			1										0		
8	2048	<--	3								2								1																0	
16	1024	<--	1																0																	
32	512	<--	0																																	

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																		
1	2048	<----	3							2						1															0				
2	1024	<----	1														0																		
4	512	<----	0																																

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity Ram
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 8-bit Data Output
DOB => DOB,      -- Port B 16-bit Data Output
DOPA => DOPA,    -- Port A 1-bit Parity Output
DOPB => DOPB,    -- Port B 2-bit Parity Output
ADDRA => ADDRA,  -- Port A 11-bit Address Input
ADDRB => ADDRB,  -- Port B 10-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 8-bit Data Input
DIB => DIB,      -- Port B 16-bit Data Input
DIPA => DIPA,    -- Port A 1-bit parity Input
DIPB => DIPB,    -- Port-B 2-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S9_S18_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S9_S18: Spartan-3E 2k/1k x 8/16 + 1/2 Parity bits Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S9_S18 #(
    .INIT_A(9'h000),      // Value of output RAM registers on Port A at startup
    .INIT_B(18'h00000),   // Value of output RAM registers on Port B at startup
    .SRVAL_A(9'h000),     // Port A output value upon SSR assertion
    .SRVAL_B(18'h00000),  // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 511, Port B Address 0 to 255
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 512 to 1023, Port B Address 256 to 511
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000)
    // Port A Address 1024 to 1535, Port B Address 512 to 767
);

```

```

.INIT_20(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1536 to 2047, Port B Address 768 to 1024
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 511, Port B Address 0 to 255
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 1023, Port B Address 256 to 511
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1024 to 1535, Port B Address 512 to 767
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1536 to 2047, Port B Address 768 to 1024
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)

) RAMB16_S9_S18_inst (
    .DOA(DOA), // Port A 8-bit Data Output
    .DOB(DOB), // Port B 16-bit Data Output
    .DOPA(DOPA), // Port A 1-bit Parity Output
    .DOPB(DOPB), // Port B 2-bit Parity Output
    .ADDRA(ADDRA), // Port A 11-bit Address Input
    .ADDRB(ADDRB), // Port B 10-bit Address Input
    .CLKA(CLKA), // Port A Clock
    .CLKB(CLKB), // Port B Clock
    .DIA(DIA), // Port A 8-bit Data Input
    .DIB(DIB), // Port B 16-bit Data Input
    .DIPA(DIPA), // Port A 1-bit parity Input
    .DIPB(DIPB), // Port-B 2-bit parity Input
    .ENA(ENA), // Port A RAM Enable Input
    .ENB(ENB), // Port B RAM Enable Input
    .SSRA(SSRA), // Port A Synchronous Set/Reset Input
    .SSRB(SSRB), // Port B Synchronous Set/Reset Input
    .WEA(WEA), // Port A Write Enable Input
    .WEB(WEB) // Port B Write Enable Input
);
;

// End of RAMB16_S9_S18_inst instantiation

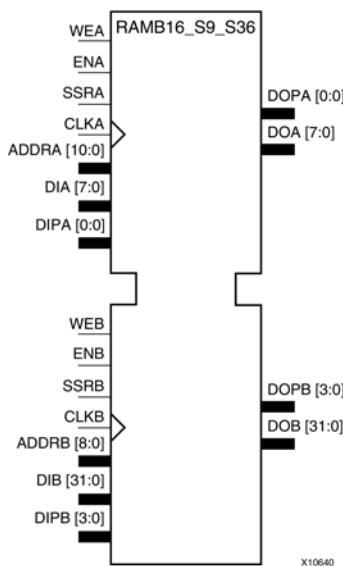
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S9\_S36

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit and 36-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

*Truth Table A*

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
								Data RAM	Parity RAM	

addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S9_S36	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	512 x 32	512 x 4	(8:0)	(31:0)	(3:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the “Port Address Mapping for Data” table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in “Port Address Mapping for Parity” table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables shows address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																		
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0		
4	4096	<--	7			6				5				4				3				2			1									0	
8	2048	<--	3								2								1																0
16	1024	<--	1																	0															
32	512	<--	0																																

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																															
1	2048	<----	3							2						1															0	
2	1024	<----	1														0															
4	512	<----	0																													

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

#### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

#### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

#### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

#### *WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

#### *WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,      -- Port A 8-bit Data Output
DOB => DOB,      -- Port B 32-bit Data Output
DOPA => DOPA,    -- Port A 1-bit Parity Output
DOPB => DOPB,    -- Port B 4-bit Parity Output
ADDRA => ADDRA,  -- Port A 11-bit Address Input
ADDRB => ADDRB,  -- Port B 9-bit Address Input
CLKA => CLKA,    -- Port A Clock
CLKB => CLKB,    -- Port B Clock
DIA => DIA,      -- Port A 8-bit Data Input
DIB => DIB,      -- Port B 32-bit Data Input
DIPA => DIPA,    -- Port A 1-bit parity Input
DIPB => DIPB,    -- Port-B 4-bit parity Input
ENA => ENA,      -- Port A RAM Enable Input
ENB => ENB,      -- PortB RAM Enable Input
SSRA => SSRA,    -- Port A Synchronous Set/Reset Input
SSRB => SSRB,    -- Port B Synchronous Set/Reset Input
WEA => WEA,      -- Port A Write Enable Input
WEB => WEB,      -- Port B Write Enable Input
);

-- End of RAMB16_S9_S36_inst instantiation

```

## Verilog Instantiation Template

```

// RAMB16_S9_S36: Spartan-3E 2k/512 x 8/32 + 1/4 Parity bits Parity bits Dual-Port RAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB16_S9_S36 #(
    .INIT_A(9'h000),          // Value of output RAM registers on Port A at startup
    .INIT_B(36'h0000000000),  // Value of output RAM registers on Port B at startup
    .SRVAL_A(9'h000),         // Port A output value upon SSR assertion
    .SRVAL_B(36'h0000000000), // Port B output value upon SSR assertion
    .WRITE_MODE_A("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .WRITE_MODE_B("WRITE_FIRST"), // WRITE_FIRST, READ_FIRST or NO_CHANGE
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"

    // The following INIT_xx declarations specify the initial contents of the RAM
    // Port A Address 0 to 511, Port B Address 0 to 127
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // Port A Address 512 to 1023, Port B Address 128 to 255
    .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000)
    // Port A Address 1024 to 1535, Port B Address 255 to 383
);

```

```

.INIT_20(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1536 to 2047, Port B Address 384 to 511
.INIT_30(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_35(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// The next set of INITP_xx are for the parity bits
// Port A Address 0 to 511, Port B Address 0 to 127
.INITP_00(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 1023, Port B Address 128 to 255
.INITP_02(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1024 to 1535, Port B Address 256 to 383
.INITP_04(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Port A Address 1536 to 2047, Port B Address 384 to 511
.INITP_06(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000)
) RAMB16_S9_S36_inst (
    .DOA(DOA), // Port A 8-bit Data Output
    .DOB(DOB), // Port B 32-bit Data Output
    .DOPA(DOPA), // Port A 1-bit Parity Output
    .DOPB(DOPB), // Port B 4-bit Parity Output
    .ADDRA(ADDRA), // Port A 11-bit Address Input
    .ADDRB(ADDRB), // Port B 9-bit Address Input
    .CLKA(CLKA), // Port A Clock
    .CLKB(CLKB), // Port B Clock
    .DIA(DIA), // Port A 8-bit Data Input
    .DIB(DIB), // Port B 32-bit Data Input
    .DIPA(DIPA), // Port A 1-bit parity Input
    .DIPB(DIPB), // Port-B 4-bit parity Input
    .ENA(ENA), // Port A RAM Enable Input
    .ENB(ENB), // Port B RAM Enable Input
    .SSRA(SSRA), // Port A Synchronous Set/Reset Input
    .SSRB(SSRB), // Port B Synchronous Set/Reset Input
    .WEA(WEA), // Port A Write Enable Input
    .WEB(WEB) // Port B Write Enable Input
);
// End of RAMB16_S9_S36_inst instantiation

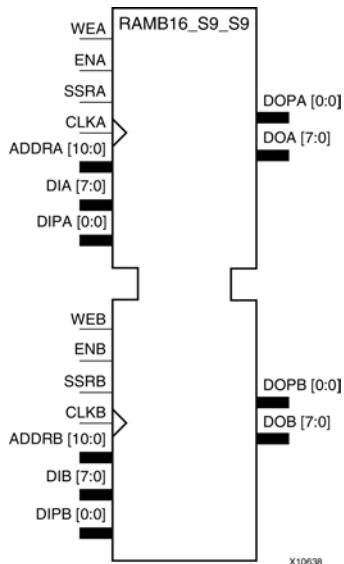
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## RAMB16\_S9\_S9

Primitive: 16K-bit Data and 2K-bit Parity Dual-Port Synchronous Block RAM with 9-bit Ports



## Introduction

This design element is a dual-ported dedicated random access memory block with synchronous write capability. Each block RAM port has 16384 bits of data memory. Ports configured as 9, 18, or 36 bits wide have an additional 2048 bits of parity memory. Each port is independent of the other while accessing the same set of 16384 data memory cells. Each port is independently configured to a specific data width. The possible port and cell configurations for this element are listed in the "Port Descriptions" section.

## Logic Table

Truth Table A

Inputs									Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents	
									Data RAM	Parity RAM	
1	X	X	X	X	X	X	X	INIT_A	INIT_A	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_A	SRVAL_A	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_A	SRVAL_A	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr)2, data3	No Change1, RAM(addr)2, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.

INIT\_A=Value specified by the INIT\_A attribute for output register. Default is all zeros.

SRVAL\_A=register value.

Inputs								Outputs		
GSR	ENA	SSRA	WEA	CLKA	ADDRA	DIA	DIPA	DOA	DOPA	RAM Contents
								Data RAM	Parity RAM	

addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_A=NO\_CHANGE.  
2WRITE\_MODE\_A=READ\_FIRST.  
3WRITE\_MODE\_A=WRITE\_FIRST.

Truth Table B

Inputs								Outputs			
GSR	ENB	SSRB	WEB	CLKB	ADDRB	DIB	DIPB	DOB	DOPB	RAM Contents	
										Data RAM	Parity RAM
1	X	X	X	X	X	X	X	INIT_B	INIT_B	No Change	No Change
0	0	X	X	X	X	X	X	No Change	No Change	No Change	No Change
0	1	1	0	↑	X	X	X	SRVAL_B	SRVAL_B	No Change	No Change
0	1	1	1	↑	addr	data	pdata	SRVAL_B	SRVAL_B	RAM(addr) =>data	RAM(addr) =>pdata
0	1	0	0	↑	addr	X	X	RAM(addr)	RAM(addr)	No Change	No Change
0	1	0	1	↑	addr	data	pdata	No Change1, RAM(addr) =>data, data3	No Change1, RAM(addr) =>data, pdata3	RAM(addr) =>data	RAM(addr) =>pdata

GSR=Global Set Reset.  
INIT\_B=Value specified by the INIT\_B attribute for output registers. Default is all zeros.  
SRVAL\_B=register value.  
addr=RAM address.  
RAM(addr)=RAM contents at address ADDR.  
data=RAM input data.  
pdata=RAM parity data.  
1WRITE\_MODE\_B=NO\_CHANGE.  
2WRITE\_MODE\_B=READ\_FIRST.  
3WRITE\_MODE\_B=WRITE\_FIRST.

## Port Descriptions

Port A						Port B					
Design Element	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	Data Cells(a)	Parity Cells(a)	Address Bus	Data Bus	Parity Bus	
RAMB16_S9_S9	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	2048 x 8	2048 x 1	(10:0)	(7:0)	(0:0)	
(a)Depth x Width											

Each port is fully synchronous with independent clock pins. All Port A input pins have setup time referenced to the CLKA pin and its data output bus DOA has a clock-to-out time referenced to the CLKA. All Port B input pins have setup time referenced to the CLKB pin and its data output bus DOB has a clock-to-out time referenced to the CLKB. The enable ENA pin controls read, write, and reset for Port A. When ENA is Low, no data is written and the outputs (DOA and DOPA) retain the last state. When ENA is High and reset (SSRA) is High, DOA and DOPA are set to SRVAL\_A during the Low-to-High clock (CLKA) transition; if write enable (WEA) is High, the memory contents reflect the data at DIA and DIPA. When ENA is High and WEA is Low, the data stored in the RAM address (ADDRA) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_A=WRITE\_FIRST, when ENA and WEA are High, the data on the data inputs (DIA and DIPA) is loaded into the word selected by the write address (ADDRA) during the Low-to-High clock transition and the data outputs (DOA and DOPA) reflect the selected (addressed) word.

The enable ENB pin controls read, write, and reset for Port B. When ENB is Low, no data is written and the outputs (DOB and DOPB) retain the last state. When ENB is High and reset (SSRB) is High, DOB and DOPB are set to SRVAL\_B during the Low-to-High clock (CLKB) transition; if write enable (WEB) is High, the memory contents reflect the data at DIB and DIPB. When ENB is High and WEB is Low, the data stored in the RAM address (ADDRB) is read during the Low-to-High clock transition. By default, WRITE\_MODE\_B=WRITE\_FIRST, when ENB and WEB are High, the data on the data inputs (DIB and PB) are loaded into the word selected by the write address (ADDRB) during the Low-to-High clock transition and the data outputs (DOB and DOPB) reflect the selected (addressed) word. The above descriptions assume active High control pins (ENA, WEA, SSRA, CLKA, ENB, WEB, SSRB, and CLKB). However, the active level can be changed by placing an inverter on the port. Any inverter placed on a RAMB16 port is absorbed into the block and does not use a CLB resource.

#### *Address Mapping*

Each port accesses the same set of 18432 memory cells using an addressing scheme that is dependent on the width of the port. For all port widths, 16384 memory cells are available for data as shown in the "Port Address Mapping for Data" table below. For 9-, 18-, and 36-bit wide ports, 2408 parity memory cells are also available as shown in "Port Address Mapping for Parity" table below. The physical RAM location that is addressed for a particular width is determined from the following formula.

$$\text{Start} = ((\text{ADDR port} + 1) * (\text{Widthport})) - 1$$

$$\text{End} = (\text{ADDRport}) * (\text{Widthport})$$

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

The following tables show address mapping for each port width.

#### *Port Address Mapping for Data*

Data Width	Port Data Addresses																																				
1	16384	<--	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
2	8192	<--	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0				
4	4096	<--	7				6			5			4				3				2			1										0			
8	2048	<--	3							2								1																	0		
16	1024	<--	1																0																		
32	512	<--	0																																		

#### *Port Address Mapping for Parity*

Parity Width	Port Parity Addresses																																				
1	2048	<----	3						2						1																	0					
2	1024	<----	1													0																					
4	512	<----	0																																		

#### *Initializing Memory Contents of a Dual-Port RAMB16*

You can use the INIT\_xx attributes to specify an initialization value for the memory contents of a RAMB16 during device configuration. The initialization of each RAMB16\_Sm\_Sn is set by 64 initialization attributes (INIT\_00 through INIT\_3F) of 64 hex values for a total of 16384 bits.

You can use the INITP\_xx attributes to specify an initial value for the parity memory during device configuration or assertion. The initialization of the parity memory for ports configured for 9, 18, or 36 bits is set by 8 initialization attributes (INITP\_00 through INITP\_07) of 64 hex values for a total of 2048 bits.

If any INIT\_xx or INITP\_xx attribute is not specified, it is configured as zeros. Partial Strings are padded with zeros to the left.

### *Initializing the Output Register of a Dual-Port RAMB16*

In Spartan®-3A and above devices, each bit in an output register can be initialized at power on (when GSR is high) to either a 0 or 1. In addition, the initial state specified for power on can be different than the state that results from assertion of a set/reset. Four properties control initialization of the output register for a dual-port RAMB16: INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B. The INIT\_A attribute specifies the output register value at power on for Port A and the INIT\_B attribute specifies the value for Port B. You can use the SRVAL\_A attribute to define the state resulting from assertion of the SSR (set/reset) input on Port A. You can use the SRVAL\_B attribute to define the state resulting from assertion of the SSR input on Port B.

The INIT\_A, INIT\_B, SRVAL\_A, and SRVAL\_B attributes specify the initialization value as a hexadecimal String. The value is dependent upon the port width. For example, for a RAMB16\_S1\_S4 with Port A width equal to 1 and Port B width equal to 4, the Port A output register contains 1 bit and the Port B output register contains 4 bits. Therefore, the INIT\_A or SRVAL\_A value can only be specified as a 1 or 0. For Port B, the output register contains 4 bits. In this case, you can use INIT\_B or SRVAL\_B to specify a hexadecimal value from 0 through F to initialize the 4 bits of the output register.

For those ports that include parity bits, the parity portion of the output register is specified in the high order bit position of the INIT\_A, INIT\_B, SRVAL\_A, or SRVAL\_B value.

The INIT and SRVAL attributes default to zero if they are not set by you.

### *Write Mode Selection*

The WRITE\_MODE\_A attribute controls the memory and output contents of Port A for a dual-port RAMB16. The WRITE\_MODE\_B attribute does the same for Port B. By default, both WRITE\_MODE\_A and WRITE\_MODE\_B are set to WRITE\_FIRST. This means that input is read, written to memory, and then passed to output. You can set the write mode for Port A and Port B to READ\_FIRST to read the memory contents, pass the memory contents to the outputs, and then write the input to memory. Or, you can set the write mode to NO\_CHANGE to have the input written to memory without changing the output. The “Port A and Port B Conflict Resolution” section describes how read/write conflicts are resolved when both Port A and Port B are attempting to read/write to the same memory cells.

### *Port A and Port B Conflict Resolution*

Spartan-3A block SelectRAM™ is True Dual-Port RAM that allows both ports to simultaneously access the same memory cell. When one port writes to a given memory cell, the other port must not address that memory cell (for a write or a read) within the clock-to-clock setup window.

The following tables summarize the collision detection behavior of the dual-port RAMB16 based on the WRITE\_MODE\_A and WRITE\_MODE\_B settings.

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=NO\_CHANGE*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	No Change	X	No Change	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	No Change	No Change	No Change	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	X	X

*WRITE\_MODE\_A= WRITE\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	DIA	X	DIPA	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	X	X	X	X	X

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=READ\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIB	DIPB

*WRITE\_MODE\_A=NO\_CHANGE and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	No Change	X	No Change	X	X	X

*WRITE\_MODE\_A=READ\_FIRST and WRITE\_MODE\_B=WRITE\_FIRST*

WEA	WEB	CLKA	CLKB	DIA	DIB	DIPA	DIPB	DOA	DOB	DOPA	DOPB	Data RAM	Parity RAM
0	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	No Change	No Change
1	0	↑	↑	DIA	DIB	DIPA	DIPB	RAM	RAM	RAM	RAM	DIA	DIPA
0	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIB	DIPB
1	1	↑	↑	DIA	DIB	DIPA	DIPB	X	DIB	X	DIPB	DIA	DIPA

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	Yes
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT_00 to INIT_3F	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the data portion of the RAM array.
INIT_A	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INIT_B	Binary/ Hexadecimal	Any	All zeros	Identifies the initial value of the DOA/DOB output port after completing configuration. For Type, the bit width is dependent on the width of the A or B port of the RAM.
INITP_00 to INITP_07	Binary/ Hexadecimal	Any	All zeros	Specifies the initial contents of the parity portion of the RAM array.
SIM_COLLISION_CHECK	String	"ALL", "WARNING_ONLY", "GENERATE_X_ONLY", "NONE"	"ALL"	<p>Allows modification of the simulation behavior if a memory collision occurs. The output is affected as follows:</p> <ul style="list-style-type: none"> <li>• "ALL" - Warning produced and affected outputs/memory location go unknown (X).</li> <li>• "WARNING_ONLY" - Warning produced and affected outputs/memory retain last value.</li> <li>• "GENERATE_X_ONLY" - No warning. However, affected outputs/memory go unknown (X).</li> <li>• "NONE" - No warning and affected outputs/memory retain last value.</li> </ul> <p><b>Note</b> Setting this to a value other than "ALL" can allow problems in the design go unnoticed during simulation. Care should be taken when changing the value of this attribute. Please see the <i>Synthesis and Simulation Design Guide</i> for more information.</p>

Attribute	Type	Allowed Values	Default	Description
SRVAL_A	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTA pin. For Type, the bit width is dependent on the width of the A port of the RAM.
SRVAL_B	Binary/ Hexadecimal	Any	All zeros	Allows the individual selection of whether the DOA/DOB output port sets (go to a one) or reset (go to a zero) upon the assertion of the RSTB pin. For Type, the bit width is dependent on the width of the B port of the RAM.
WRITE_MODE_A	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.
WRITE_MODE_B	String	"WRITE_FIRST", "READ_FIRST", "NO_CHANGE"	"WRITE_FIRST"	Specifies the behavior of the DOA/DOB port upon a write command to the respected port. If set to "WRITE_FIRST", the same port that is written to displays the contents of the written data to the outputs upon completion of the operation. "READ_FIRST" displays the prior contents of the RAM to the output port prior to writing the new data. "NO_CHANGE" keeps the previous value on the output port and won't update the output port upon a write command. This is the suggested mode if not using the read data from a particular port of the RAM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.



```

DOA => DOA,          -- Port A 8-bit Data Output
DOB => DOB,          -- Port B 8-bit Data Output
DOPA => DOPA,         -- Port A 1-bit Parity Output
DOPB => DOPB,         -- Port B 1-bit Parity Output
ADDRA => ADDRA,       -- Port A 11-bit Address Input
ADDRB => ADDRB,       -- Port B 11-bit Address Input
CLKA => CLKA,         -- Port A Clock
CLKB => CLKB,         -- Port B Clock
DIA => DIA,           -- Port A 8-bit Data Input
DIB => DIB,           -- Port B 8-bit Data Input
DIPA => DIPA,          -- Port A 1-bit parity Input
DIPB => DIPB,          -- Port-B 1-bit parity Input
ENA => ENA,            -- Port A RAM Enable Input
ENB => ENB,            -- PortB RAM Enable Input
SSRA => SSRA,          -- Port A Synchronous Set/Reset Input
SSRB => SSRB,          -- Port B Synchronous Set/Reset Input
WEA => WEA,             -- Port A Write Enable Input
WEB => WEB,             -- Port B Write Enable Input
);

-- End of RAMB16_S9_S9_inst instantiation

```

## Verilog Instantiation Template

```
// RAMB16_S9_S9: Spartan-3/3E/3A/3AN/3AD 2k x 8 + 1 Parity bit Dual-Port RAM  
// Xilinx HDL Libraries Guide, version 11.2
```

```

.INIT_20(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_21(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_22(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_23(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_24(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_25(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_26(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_27(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_28(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_29(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2A(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2B(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2C(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2D(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2E(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_2F(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
// Address 1536 to 2047
.INIT_30(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_31(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_32(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_33(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_34(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_35(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_36(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_37(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_38(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_39(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3A(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3B(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3C(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3D(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3E(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
.INIT_3F(256'h00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00_00),
// The next set of INITP_xx are for the parity bits
// Address 0 to 511
.INITP_00(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 512 to 1023
.INITP_02(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 1024 to 1535
.INITP_04(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
// Address 1536 to 2047
.INITP_06(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000),
) RAMB16_S9_S9_inst (
.DOA(DOA), // Port A 8-bit Data Output
.DOB(DOB), // Port B 8-bit Data Output
.DOPA(DOPA), // Port A 1-bit Parity Output
.DOPB(DOPB), // Port B 1-bit Parity Output
.ADDRA(ADDRA), // Port A 11-bit Address Input
.ADDRB(ADDRB), // Port B 11-bit Address Input
.CLKA(CLKA), // Port A Clock
.CLKB(CLKB), // Port B Clock
.DIA(DIA), // Port A 8-bit Data Input
.DIB(DIB), // Port B 8-bit Data Input
.DIPA(DIPA), // Port A 1-bit parity Input
.DIPB(DIPB), // Port-B 1-bit parity Input
.ENA(ENA), // Port A RAM Enable Input
.ENB(ENB), // Port B RAM Enable Input
.SSRA(SSRA), // Port A Synchronous Set/Reset Input
.SSRB(SSRB), // Port B Synchronous Set/Reset Input
.WEA(WEA), // Port A Write Enable Input
.WEB(WEB) // Port B Write Enable Input
);
// End of RAMB16_S9_S9_inst instantiation

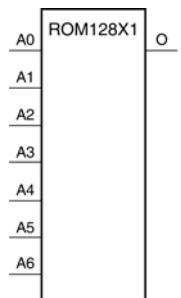
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## ROM128X1

Primitive: 128-Deep by 1-Wide ROM



X9731

## Introduction

This design element is a 128-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 7-bit address (A6:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 32 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. An error occurs if the INIT=value is not specified.

## Logic Table

Input				Output
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 128-Bit Value	All zeros	Specifies the contents of the ROM.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM128X1: 128 x 1 Asynchronous Distributed  => LUT ROM
--           Virtex-4/5/6, Spartan-3/3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

ROM128X1_inst : ROM128X1
generic map (
    INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
)
port map (
    O => O,      -- ROM output
    A0 => A0,    -- ROM address[0]
    A1 => A1,    -- ROM address[1]
    A2 => A2,    -- ROM address[2]
    A3 => A3,    -- ROM address[3]
    A4 => A4,    -- ROM address[4]
    A5 => A5,    -- ROM address[5]
    A6 => A6    -- ROM address[6]
);
-- End of ROM128X1_inst instantiation
```

### Verilog Instantiation Template

```
// ROM128X1: 128 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-4/5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

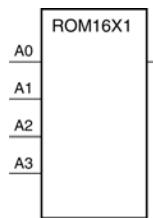
ROM128X1 #(
    .INIT(128'h0000000000000000000000000000000000000000000000000000000000000000)
) ROM128X1_inst (
    .O(O),      // ROM output
    .A0(A0),    // ROM address[0]
    .A1(A1),    // ROM address[1]
    .A2(A2),    // ROM address[2]
    .A3(A3),    // ROM address[3]
    .A4(A4),    // ROM address[4]
    .A5(A5),    // ROM address[5]
    .A6(A6)    // ROM address[6]
);
// End of ROM128X1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## ROM16X1

Primitive: 16-Deep by 1-Wide ROM



X4137

## Introduction

This design element is a 16-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 4-bit address (A3:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of four hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. For example, the INIT=10A7 parameter produces the data stream: 0001 0000 1010 0111 An error occurs if the INIT=value is not specified.

## Logic Table

Input				Output
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Specifies the contents of the ROM.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM16X1: 16 x 1 Asynchronous Distributed => LUT ROM
-- Xilinx HDL Libraries Guide, version 11.2

ROM16X1_inst : ROM16X1
generic map (
    INIT => X"0000")
port map (
    O => O,      -- ROM output
    A0 => A0,    -- ROM address[0]
    A1 => A1,    -- ROM address[1]
    A2 => A2,    -- ROM address[2]
    A3 => A3    -- ROM address[3]
);
-- End of ROM16X1_inst instantiation
```

### Verilog Instantiation Template

```
// ROM16X1: 16 x 1 Asynchronous Distributed (LUT) ROM
//          All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

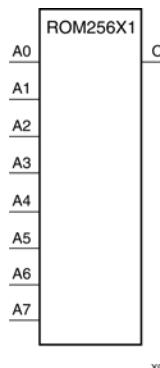
ROM16X1 #(
    .INIT(16'h0000) // Contents of ROM
) ROM16X1_inst (
    .O(O),      // ROM output
    .A0(A0),    // ROM address[0]
    .A1(A1),    // ROM address[1]
    .A2(A2),    // ROM address[2]
    .A3(A3)    // ROM address[3]
);
// End of ROM16X1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## ROM256X1

Primitive: 256-Deep by 1-Wide ROM



## Introduction

This design element is a 256-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 8-bit address (A7:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 64 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H.

An error occurs if the INIT=value is not specified.

## Logic Table

Input				Output
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

# Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 256-Bit Value	All zeros	Specifies the contents of the ROM.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

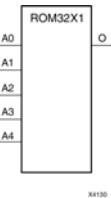
## Verilog Instantiation Template

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## ROM32X1

**Primitive:** 32-Deep by 1-Wide ROM



### Introduction

This design element is a 32-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 5-bit address (A4:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of eight hexadecimal digits that are written into the ROM from the most-significant digit A=1FH to the least-significant digit A=00H.

For example, the INIT=10A78F39 parameter produces the data stream: 0001 0000 1010 0111 1000 1111 0011 1001. An error occurs if the INIT=value is not specified.

### Logic Table

Input				Output
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 32-Bit Value	All zeros	Specifies the contents of the ROM.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM32X1: 32 x 1 Asynchronous Distributed => LUT ROM
-- Xilinx HDL Libraries Guide, version 11.2

ROM32X1_inst : ROM32X1
generic map (
    INIT => X"00000000")
port map (
    O => O,      -- ROM output
    A0 => A0,    -- ROM address[0]
    A1 => A1,    -- ROM address[1]
    A2 => A2,    -- ROM address[2]
    A3 => A3,    -- ROM address[3]
    A4 => A4    -- ROM address[4]
);
-- End of ROM32X1_inst instantiation
```

### Verilog Instantiation Template

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//          A11 FPGAs
// Xilinx HDL Libraries Guide, version 11.2

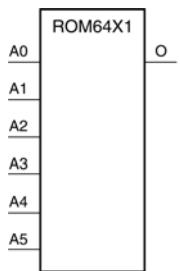
ROM32X1 #(
    .INIT(32'h00000000) // Contents of ROM
) ROM32X1_inst (
    .O(O),      // ROM output
    .A0(A0),    // ROM address[0]
    .A1(A1),    // ROM address[1]
    .A2(A2),    // ROM address[2]
    .A3(A3),    // ROM address[3]
    .A4(A4)     // ROM address[4]
);
// End of ROM32X1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## ROM64X1

Primitive: 64-Deep by 1-Wide ROM



X973G

### Introduction

This design element is a 64-word by 1-bit read-only memory. The data output (O) reflects the word selected by the 6-bit address (A5:A0). The ROM is initialized to a known value during configuration with the INIT=value parameter. The value consists of 16 hexadecimal digits that are written into the ROM from the most-significant digit A=FH to the least-significant digit A=0H. An error occurs if the INIT=value is not specified.

### Logic Table

Input				Output
I0	I1	I2	I3	O
0	0	0	0	INIT(0)
0	0	0	1	INIT(1)
0	0	1	0	INIT(2)
0	0	1	1	INIT(3)
0	1	0	0	INIT(4)
0	1	0	1	INIT(5)
0	1	1	0	INIT(6)
0	1	1	1	INIT(7)
1	0	0	0	INIT(8)
1	0	0	1	INIT(9)
1	0	1	0	INIT(10)
1	0	1	1	INIT(11)
1	1	0	0	INIT(12)
1	1	0	1	INIT(13)
1	1	1	0	INIT(14)
1	1	1	1	INIT(15)

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 64-Bit Value	All zeros	Specifies the contents of the ROM.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- ROM64X1: 64 x 1 Asynchronous Distributed => LUT ROM
--          Virtex-4/5/6, Spartan-3/3E/3A/6
--          Xilinx HDL Libraries Guide, version 11.2

ROM64X1_inst : ROM64X1
generic map (
    INIT => X"0000000000000000")
port map (
    O => O,      -- ROM output
    A0 => A0,    -- ROM address[0]
    A1 => A1,    -- ROM address[1]
    A2 => A2,    -- ROM address[2]
    A3 => A3,    -- ROM address[3]
    A4 => A4,    -- ROM address[4]
    A5 => A5    -- ROM address[5]
);
-- End of ROM64X1_inst instantiation
```

### Verilog Instantiation Template

```
// ROM64X1: 64 x 1 Asynchronous Distributed (LUT) ROM
//          Virtex-4/5/6, Spartan-3/3E/3A/6
//          Xilinx HDL Libraries Guide, version 11.2

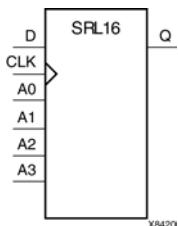
ROM64X1 #(
    .INIT(64'h0000000000000000) // Contents of ROM
) ROM64X1_inst (
    .O(O),      // ROM output
    .A0(A0),    // ROM address[0]
    .A1(A1),    // ROM address[1]
    .A2(A2),    // ROM address[2]
    .A3(A3),    // ROM address[3]
    .A4(A4),    // ROM address[4]
    .A5(A5)    // ROM address[5]
);
// End of ROM64X1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRL16

Primitive: 16-Bit Shift Register Look-Up Table (LUT)



## Introduction

This design element is a shift register look-up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions data shifts to the next highest bit position while new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

## Logic Table

Inputs			Output
Am	CLK	D	Q
Am	X	X	Q(Am)
Am	↑	D	Q(Am - 1)
m= 0, 1, 2, 3			

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of Q output after configuration.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16: 16-bit shift register LUT operating on posedge of clock
--          All FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

SRL16_inst : SRL16
generic map (
    INIT => X"0000")
port map (
    Q => Q,           -- SRL data output
    A0 => A0,         -- Select[0] input
    A1 => A1,         -- Select[1] input
    A2 => A2,         -- Select[2] input
    A3 => A3,         -- Select[3] input
    CLK => CLK,       -- Clock input
    D => D           -- SRL data input
);
-- End of SRL16_inst instantiation
```

## Verilog Instantiation Template

```
// SRL16: 16-bit shift register LUT operating on posedge of clock
//          All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

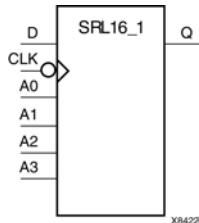
SRL16 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16_inst (
    .Q(Q),           // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CLK(CLK),       // Clock input
    .D(D)           // SRL data input
);
// End of SRL16_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRL16\_1

**Primitive:** 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock



## Introduction

This design element is a shift register look-up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition. During subsequent High-to-Low clock transitions data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

## Logic Table

Inputs			Output
Am	CLK	D	Q
Am	X	X	Q(Am)
Am	↓	D	Q(Am - 1)
m= 0, 1, 2, 3			

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of Q output after configuration

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16_1: 16-bit shift register LUT operating on negedge of clock
--           All FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

SRL16_1_inst : SRL16_1
generic map (
    INIT => X"0000")
port map (
    Q => Q,          -- SRL data output
    A0 => A0,         -- Select[0] input
    A1 => A1,         -- Select[1] input
    A2 => A2,         -- Select[2] input
    A3 => A3,         -- Select[3] input
    CLK => CLK,        -- Clock input
    D => D           -- SRL data input
);
-- End of SRL16_1_inst instantiation
```

## Verilog Instantiation Template

```
// SRL16_1: 16-bit shift register LUT operating on negedge of clock
//           All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

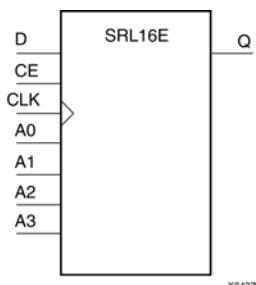
SRL16_1 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16_1_inst (
    .Q(Q),          // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CLK(CLK),        // Clock input
    .D(D)           // SRL data input
);
// End of SRL16_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRL16E

**Primitive:** 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



### Introduction

This design element is a shift register look-up table (LUT). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions, when CE is High, data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached. When CE is Low, the register ignores clock transitions.

### Logic Table

Inputs				Output
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↑	D	Q(Am - 1)
m= 0, 1, 2, 3				

## Port Descriptions

Port	Direction	Width	Function
Q	Output	1	Shift register data output
D	Input	1	Shift register data input
CLK	Input	1	Clock
CE	Input	1	Active high clock enable
A	Input	4	Dynamic depth selection of the SRL <ul style="list-style-type: none"> <li>• A=0000 ==&gt; 1-bit shift length</li> <li>• A=1111 ==&gt; 16-bit shift length</li> </ul>

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
--          All FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

SRL16E_inst : SRL16E
generic map (
    INIT => X"0000")
port map (
    Q => Q,           -- SRL data output
    A0 => A0,         -- Select[0] input
    A1 => A1,         -- Select[1] input
    A2 => A2,         -- Select[2] input
    A3 => A3,         -- Select[3] input
    CE => CE,         -- Clock enable input
    CLK => CLK,        -- Clock input
    D => D            -- SRL data input
);
-- End of SRL16E_inst instantiation

```

## Verilog Instantiation Template

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//          All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

SRL16E #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
    .Q(Q),           // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CE(CE),          // Clock enable input
    .CLK(CLK),        // Clock input
    .D(D)            // SRL data input
);

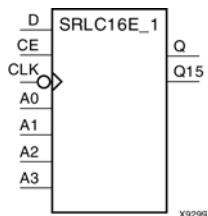
// End of SRL16E_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRL16E\_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable



### Introduction

This design element is a shift register look-up table (LUT) with clock enable (CE). The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition. During subsequent High-to-Low clock transitions, when CE is High, data is shifted to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached. When CE is Low, the register ignores clock transitions.

### Logic Table

Inputs				Output
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↓	D	Q(Am - 1)
m= 0, 1, 2, 3				

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
--           All FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

SRL16E_1_inst : SRL16E_1
generic map (
    INIT => X"0000")
port map (
    Q => Q,          -- SRL data output
    A0 => A0,         -- Select[0] input
    A1 => A1,         -- Select[1] input
    A2 => A2,         -- Select[2] input
    A3 => A3,         -- Select[3] input
    CE => CE,         -- Clock enable input
    CLK => CLK,        -- Clock input
    D => D,          -- SRL data input
);
-- End of SRL16E_1_inst instantiation
```

### Verilog Instantiation Template

```
// SRL16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
//           All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

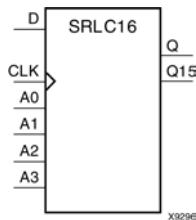
SRL16E_1 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_1_inst (
    .Q(Q),          // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);
// End of SRL16E_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRLC16

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry



### Introduction

This design element is a shift register look-up table (LUT) with Carry. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. During subsequent Low-to-High clock transitions data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

**Note** The Q15 output is available for you in cascading to multiple shift register LUTs to create larger shift registers.

### Logic Table

Inputs			Output
Am	CLK	D	Q
Am	X	X	Q(Am)
Am	↑	D	Q(Am - 1)
m= 0, 1, 2, 3			

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
--          Spartan-3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

SRLC16_inst : SRLC16
generic map (
    INIT => X"0000")
port map (
    Q => Q,          -- SRL data output
    Q15 => Q15,      -- Carry output (connect to next SRL)
    A0 => A0,          -- Select[0] input
    A1 => A1,          -- Select[1] input
    A2 => A2,          -- Select[2] input
    A3 => A3,          -- Select[3] input
    CLK => CLK,        -- Clock input
    D => D,          -- SRL data input
);
-- End of SRLC16_inst instantiation
```

### Verilog Instantiation Template

```
// SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
//          Virtex-4, Spartan-3E/3A
// Xilinx HDL Libraries Guide, version 11.2

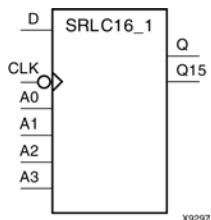
SRLC16 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16_inst (
    .Q(Q),          // SRL data output
    .Q15(Q15),      // Carry output (connect to next SRL)
    .A0(A0),          // Select[0] input
    .A1(A1),          // Select[1] input
    .A2(A2),          // Select[2] input
    .A3(A3),          // Select[3] input
    .CLK(CLK),        // Clock input
    .D(D)           // SRL data input
);
// End of SRLC16_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRLC16\_1

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock



### Introduction

This design element is a shift register look-up table (LUT) with carry and a negative-edge clock. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

**Note** The Q15 output is available for your use in cascading multiple shift register LUTs to create larger shift registers.

### Logic Table

Inputs			Output	
Am	CLK	D	Q	Q15
Am	X	X	Q(Am)	No Change
Am	↓	D	Q(Am - 1)	Q14
m= 0, 1, 2, 3				

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
--           Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

SRLC16_1_inst : SRLC16_1
generic map (
    INIT => X"0000")
port map (
    Q => Q,          -- SRL data output
    Q15 => Q15,      -- Carry output (connect to next SRL)
    A0 => A0,          -- Select[0] input
    A1 => A1,          -- Select[1] input
    A2 => A2,          -- Select[2] input
    A3 => A3,          -- Select[3] input
    CLK => CLK,        -- Clock input
    D => D,          -- SRL data input
);
-- End of SRLC16_1_inst instantiation
```

### Verilog Instantiation Template

```
// SRLC16_1: 16-bit cascadable shift register LUT operating on negedge of clock
//           Virtex-4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 11.2

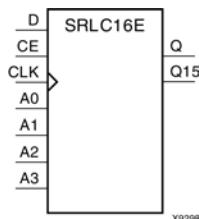
SRLC16_1 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16_1_inst (
    .Q(Q),          // SRL data output
    .Q15(Q15),      // Carry output (connect to next SRL)
    .A0(A0),          // Select[0] input
    .A1(A1),          // Select[1] input
    .A2(A2),          // Select[2] input
    .A3(A3),          // Select[3] input
    .CLK(CLK),        // Clock input
    .D(D)          // SRL data input
);
// End of SRLC16_1_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRLC16E

Primitive: 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable



### Introduction

This design element is a shift register look-up table (LUT) with carry and clock enable. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

The data (D) is loaded into the first bit of the shift register during the Low-to-High clock (CLK) transition. When CE is High, during subsequent Low-to-High clock transitions, data shifts to the next highest bit position as new data is loaded. The data appears on the Q output when the shift register length determined by the address inputs is reached.

**Note** The Q15 output is available for you in cascading to multiple shift register LUTs to create larger shift registers.

### Logic Table

Inputs				Output	
Am	CLK	CE	D	Q	Q15
Am	X	0	X	Q(Am)	Q(15)
Am	X	1	X	Q(Am)	Q(15)
Am	↑	1	D	Q(Am - 1)	Q15
m = 0, 1, 2, 3					

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRLC16E: 16-bit cascable shift register LUT with clock enable operating on posedge of clock
--          Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

SRLC16E_inst : SRLC16E
generic map (
    INIT => X"0000")
port map (
    Q => Q,          -- SRL data output
    Q15 => Q15,      -- Carry output (connect to next SRL)
    A0 => A0,          -- Select[0] input
    A1 => A1,          -- Select[1] input
    A2 => A2,          -- Select[2] input
    A3 => A3,          -- Select[3] input
    CE => CE,          -- Clock enable input
    CLK => CLK,          -- Clock input
    D => D,          -- SRL data input
);
-- End of SRLC16E_inst instantiation
```

### Verilog Instantiation Template

```
// SRLC16E: 16-bit cascadable shift register LUT with clock enable operating on posedge of clock
//          Virtex-4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 11.2

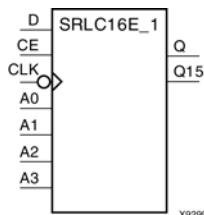
SRLC16E #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16E_inst (
    .Q(Q),          // SRL data output
    .Q15(Q15),      // Carry output (connect to next SRL)
    .A0(A0),          // Select[0] input
    .A1(A1),          // Select[1] input
    .A2(A2),          // Select[2] input
    .A3(A3),          // Select[3] input
    .CE(CE),          // Clock enable input
    .CLK(CLK),          // Clock input
    .D(D)          // SRL data input
);
// End of SRLC16E_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## SRLC16E\_1

**Primitive:** 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable



### Introduction

This design element is a shift register look-up table (LUT) with carry, clock enable, and negative-edge clock. The inputs A3, A2, A1, and A0 select the output length of the shift register.

The shift register can be of a fixed, static length or it can be dynamically adjusted.

- **To create a fixed-length shift register** -Drive the A3 through A0 inputs with static values. The length of the shift register can vary from 1 bit to 16 bits, as determined by the following formula: Length =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ . If A3, A2, A1, and A0 are all zeros (0000), the shift register is one bit long. If they are all ones (1111), it is 16 bits long.
- **To change the length of the shift register dynamically** -Change the values driving the A3 through A0 inputs. For example, if A2, A1, and A0 are all ones (111) and A3 toggles between a one (1) and a zero (0), the length of the shift register changes from 16 bits to 8 bits. Internally, the length of the shift register is always 16 bits and the input lines A3 through A0 select which of the 16 bits reach the output.

The shift register LUT contents are initialized by assigning a four-digit hexadecimal number to an INIT attribute. The first, or the left-most, hexadecimal digit is the most significant bit. If an INIT value is not specified, it defaults to a value of four zeros (0000) so that the shift register LUT is cleared during configuration.

When CE is High, the data (D) is loaded into the first bit of the shift register during the High-to-Low clock (CLK) transition. During subsequent High-to-Low clock transitions data shifts to the next highest bit position as new data is loaded when CE is High. The data appears on the Q output when the shift register length determined by the address inputs is reached.

**Note** The Q15 output is available for your use in cascading multiple shift register LUTs to create larger shift registers.

### Logic Table

Inputs				Output	
Am	CE	CLK	D	Q	Q15
Am	0	X	X	Q(Am)	No Change
Am	1	X	X	Q(Am)	No Change
Am	1	↓	D	Q(Am -1 )	Q14
m= 0, 1, 2, 3					

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## Available Attributes

Attribute	Type	Allowed Values	Default	Description
INIT	Hexadecimal	Any 16-Bit Value	All zeros	Sets the initial value of content and output of shift register after configuration.

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
-- Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

SRLC16E_1_inst : SRLC16E_1
generic map (
    INIT => X"0000")
port map (
    Q => Q,          -- SRL data output
    Q15 => Q15,      -- Carry output (connect to next SRL)
    A0 => A0,        -- Select[0] input
    A1 => A1,        -- Select[1] input
    A2 => A2,        -- Select[2] input
    A3 => A3,        -- Select[3] input
    CE => CE,        -- Clock enable input
    CLK => CLK,      -- Clock input
    D => D,          -- SRL data input
);
-- End of SRLC16E_1_inst instantiation
```

### Verilog Instantiation Template

```
// SRLC16E_1: 16-bit shift register LUT with clock enable operating on negedge of clock
// Virtex-4, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 11.2

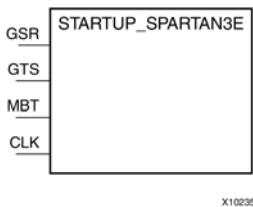
SRLC16E_1 #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRLC16E_1_inst (
    .Q(Q),          // SRL data output
    .Q15(Q15),      // Carry output (connect to next SRL)
    .A0(A0),        // Select[0] input
    .A1(A1),        // Select[1] input
    .A2(A2),        // Select[2] input
    .A3(A3),        // Select[3] input
    .CE(CE),        // Clock enable input
    .CLK(CLK),      // Clock input
    .D(D)           // SRL data input
);
// End of SRLC16E_1_inst instantiation
```

## For More Information

- See the *Spartan-3 Generation FPGA User Guide*.
- See the *Spartan-3E FPGA Family Complete Data Sheet*.

## STARTUP\_SPARTAN3E

**Primitive: Spartan®-3E User Interface to the GSR, GTS, Configuration Startup Sequence and Multi-Boot Trigger Circuitry**



X10236

### Introduction

This design element allows the connection of ports, or your circuitry, to control certain dedicated circuitry and routes within the FPGA. Signals connected to the GSR port of this component can control the global set/reset (referred to as GSR) of the device. The GSR net connects to all registers in the device and places the registers into their initial value state. Connecting a signal to the GTS port connects to the dedicated route controlling the three-state outputs of every pin in the device. Connecting a clock signal to the CLK input allows the startup sequence after configuration to be synchronized to a user-defined clock. The MBT (Multi-Boot Trigger) pin allows the triggering of a new configuration when the device is properly set up for this feature.

### Design Entry Method

Instantiation	Recommended
Inference	No
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- STARTUP_SPARTAN3E: Startup primitive for GSR, GTS, startup sequence
-- control and Multi-Boot Configuration. Spartan-3E
-- xilinx HDL Libraries Guide, version 11.2

STARTUP_SPARTAN3E_inst : STARTUP_SPARTAN3E
port map (
    CLK => CLK,          -- Clock input for start-up sequence
    GSR => GSR_PORT,    -- Global Set/Reset input (GSR cannot be used for the port name)
    GTS => GTS_PORT,    -- Global 3-state input (GTS cannot be used for the port name)
    MBT => MBT -- Multi-Boot Trigger input
);
-- End of STARTUP_SPARTAN3E_inst instantiation
```

## Verilog Instantiation Template

```
// STARTUP_SPARTAN3E: Startup primitive for GSR, GTS, startup sequence control
//           and Multi-Boot Configuration Trigger. Spartan-3E
// Xilinx HDL Libraries Guide, version 11.2

STARTUP_SPARTAN3E STARTUP_SPARTAN3E_inst (
    .CLK(CLK),          // Clock input for start-up sequence
    .GSR(GSR_PORT),    // Global Set/Reset input (GSR can not be used as a port name)
    .GTS(GTS_PORT),    // Global 3-state input (GTS can not be used as a port name)
    .MBT(MBT)          // Multi-Boot Trigger input
);

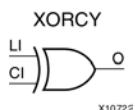
// End of STARTUP_SPARTAN3E_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## XORCY

**Primitive:** XOR for Carry Logic with General Output



## Introduction

This design element is a special XOR with general O output that generates faster and smaller arithmetic functions. The XORCY primitive is a dedicated XOR function within the carry-chain logic of the slice. It allows for fast and efficient creation of arithmetic (add/subtract) or wide logic functions (large AND/OR gate).

## Logic Table

Input		Output
LI	CI	O
0	0	0
0	1	1
1	0	1
1	1	0

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- XORCY: Carry-Chain XOR-gate with general output
-- Xilinx HDL Libraries Guide, version 11.2

XORCY_inst : XORCY
port map (
    O => O,      -- XOR output signal
    CI => CI,    -- Carry input signal
    LI => LI     -- LUT4 input signal
);
-- End of XORCY_inst instantiation
```

## Verilog Instantiation Template

```
// XORCY: Carry-Chain XOR-gate with general output
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

XORCY XORCY_inst (
    .O(O),      // XOR output signal
    .CI(CI),    // Carry input signal
    .LI(LI)     // LUT4 input signal
);
// End of XORCY_inst instantiation
```

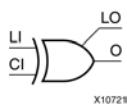
## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## XORCY\_D

**Primitive:** XOR for Carry Logic with Dual Output

XORCY\_D



### Introduction

This design element is a special XOR that generates faster and smaller arithmetic functions.

### Logic Table

Input		Output
LI	CI	O and LO
0	0	0
0	1	1
1	0	1
1	1	0

### Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

### VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- XORCY_D: Carry-Chain XOR-gate with local and general outputs
-- Xilinx HDL Libraries Guide, version 11.2

XORCY_D_inst : XORCY_D
port map (
    LO => LO, -- XOR local output signal
    O => O, -- XOR general output signal
    CI => CI, -- Carry input signal
    LI => LI -- LUT4 input signal
);
-- End of XORCY_D_inst instantiation
```

## Verilog Instantiation Template

```
// XORCY_D: Carry-Chain XOR-gate with local and general outputs
//           For use with All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

XORCY_D XORCY_D_inst (
    .LO(LO), // XOR local output signal
    .O(O), // XOR general output signal
    .CI(CI), // Carry input signal
    .LI(LI) // LUT4 input signal
);

// End of XORCY_D_inst instantiation
```

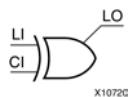
## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).

## XORCY\_L

**Primitive:** XOR for Carry Logic with Local Output

XORCY\_L



## Introduction

This design element is a special XOR with local LO output that generates faster and smaller arithmetic functions.

## Logic Table

Input		Output
LI	CI	LO
0	0	0
0	1	1
1	0	1
1	1	0

## Design Entry Method

Instantiation	Yes
Inference	Recommended
CORE Generator™ and wizards	No
Macro support	No

## VHDL Instantiation Template

Unless they already exist, copy the following two statements and paste them before the entity declaration.

```
Library UNISIM;
use UNISIM.vcomponents.all;

-- XORCY_L: Carry-Chain XOR-gate with local  => direct-connect ouput
-- Xilinx HDL Libraries Guide, version 11.2

XORCY_L_inst : XORCY_L
port map (
    LO => LO, -- XOR local output signal
    CI => CI, -- Carry input signal
    LI => LI -- LUT4 input signal
);
-- End of XORCY_L_inst instantiation
```

## Verilog Instantiation Template

```
// XORCY_L: Carry-Chain XOR-gate with local (direct-connect) output
//          For use with All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

XORCY_L XORCY_L_inst (
    .LO(LO), // XOR local output signal
    .CI(CI), // Carry input signal
    .LI(LI)  // LUT4 input signal
);
// End of XORCY_L_inst instantiation
```

## For More Information

- See the [Spartan-3 Generation FPGA User Guide](#).
- See the [Spartan-3E FPGA Family Complete Data Sheet](#).