

Spartan-3A および Spartan-3A DSP ライブラリ ガイド (HDL 用)

UG613 (v11.3) 2009 年 9 月 16 日

本資料は英語版 (v11.3) を翻訳したものです。英語の更新バージョンがリリースされている場合には、最新の英語版を必ずご参照ください。

ザイリンクス商標および著作権情報



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.

目次

ザイリックス商標および著作権情報	2
1: このマニュアルについて	13
デザイン エLEMENT	13
デザインの入力方法	14
2: ターゲット デバイス変更後のデザイン エLEMENT	15
3: ファンクション別分類	19
4: デザイン エLEMENT	25
BSCAN_SPARTAN3A	26
概要	26
ポートの説明	26
デザインの入力方法	27
詳細情報	28
BUFCF	29
概要	29
デザインの入力方法	29
詳細情報	30
BUFG	31
概要	31
ポートの説明	31
デザインの入力方法	31
詳細情報	32
BUFGCE	33
概要	33
論理表	33
デザインの入力方法	33
詳細情報	34
BUFGMUX	35
概要	35
論理表	35
ポートの説明	35
デザインの入力方法	35
使用可能な属性	36
詳細情報	36
CAPTURE_SPARTAN3A	37
概要	37
ポートの説明	37
デザインの入力方法	37
使用可能な属性	37
詳細情報	38
DCM_SP	39
概要	39
ポートの説明	39
デザインの入力方法	40
使用可能な属性	40
詳細情報	43
DNA_PORT	44
概要	44
ポートの説明	44
デザインの入力方法	44
使用可能な属性	45
詳細情報	45
DSP48A	46
概要	46
ポートの説明	46
デザインの入力方法	49

使用可能な属性.....	49
詳細情報	51
FDCPE.....	52
概要.....	52
論理表.....	52
ポートの説明	52
デザインの入力方法.....	53
使用可能な属性.....	53
詳細情報	54
FDCPE_1	55
概要.....	55
論理表.....	55
ポートの説明	55
デザインの入力方法.....	56
使用可能な属性.....	56
詳細情報	57
FDRSE	58
概要.....	58
論理表.....	58
デザインの入力方法.....	58
使用可能な属性.....	58
詳細情報	59
FDRSE_1	60
概要.....	60
論理表.....	60
デザインの入力方法.....	60
使用可能な属性.....	60
詳細情報	61
IBUF.....	62
概要.....	62
ポートの説明	62
デザインの入力方法.....	62
使用可能な属性.....	62
詳細情報	63
IBUF_DLY_ADJ.....	64
概要.....	64
ポートの説明	64
デザインの入力方法.....	64
使用可能な属性.....	64
詳細情報	65
IBUFDS.....	66
概要.....	66
論理表.....	66
ポートの説明	66
デザインの入力方法.....	66
使用可能な属性.....	67
詳細情報	68
IBUFDS_DLY_ADJ.....	69
概要.....	69
ポートの説明	69
デザインの入力方法.....	69
使用可能な属性.....	69
詳細情報	70
IBUFG	71
概要.....	71
ポートの説明	71
デザインの入力方法.....	71
使用可能な属性.....	71
詳細情報	72

IBUFGDS	73
概要.....	73
論理表.....	73
ポートの説明	73
デザインの入力方法.....	73
使用可能な属性.....	74
詳細情報	74
ICAP_SPARTAN3A.....	75
概要.....	75
ポートの説明	75
デザインの入力方法.....	75
詳細情報	76
IDDR2.....	77
概要.....	77
論理表.....	77
デザインの入力方法.....	77
使用可能な属性.....	78
詳細情報	79
IOBUF	80
概要.....	80
論理表.....	80
ポートの説明	80
デザインの入力方法.....	80
使用可能な属性.....	81
詳細情報	82
IOBUFDS	83
概要.....	83
論理表.....	83
ポートの説明	83
デザインの入力方法.....	83
使用可能な属性.....	84
詳細情報	85
KEEPER.....	86
概要.....	86
ポートの説明	86
デザインの入力方法.....	86
詳細情報	87
LDCPE.....	88
概要.....	88
論理表.....	88
ポートの説明	88
デザインの入力方法.....	89
使用可能な属性.....	89
詳細情報	90
LUT1	91
概要.....	91
論理表.....	91
デザインの入力方法.....	91
使用可能な属性.....	91
詳細情報	92
LUT1_D.....	93
概要.....	93
論理表.....	93
デザインの入力方法.....	93
使用可能な属性.....	93
詳細情報	94
LUT1_L.....	95
概要.....	95
論理表.....	95

デザインの入力方法.....	95
使用可能な属性.....	95
詳細情報	96
LUT2	97
概要.....	97
論理表.....	97
デザインの入力方法.....	97
使用可能な属性.....	98
詳細情報	98
LUT2_D.....	99
概要.....	99
論理表.....	99
デザインの入力方法.....	99
使用可能な属性.....	100
詳細情報	100
LUT2_L.....	101
概要.....	101
論理表.....	101
デザインの入力方法.....	101
使用可能な属性.....	102
詳細情報	102
LUT3	103
概要.....	103
論理表.....	103
デザインの入力方法.....	104
使用可能な属性.....	104
詳細情報	104
LUT3_D.....	105
概要.....	105
論理表.....	105
デザインの入力方法.....	106
使用可能な属性.....	106
詳細情報	106
LUT3_L.....	107
概要.....	107
論理表.....	107
デザインの入力方法.....	108
使用可能な属性.....	108
詳細情報	108
LUT4	109
概要.....	109
論理表.....	110
デザインの入力方法.....	110
使用可能な属性.....	110
詳細情報	111
LUT4_D.....	112
概要.....	112
論理表.....	113
デザインの入力方法.....	113
使用可能な属性.....	113
詳細情報	114
LUT4_L.....	115
概要.....	115
論理表.....	116
デザインの入力方法.....	116
使用可能な属性.....	116
詳細情報	117
MULT_AND	118
概要.....	118

論理表.....	118
デザインの入力方法.....	118
詳細情報	119
MULT18X18SIO	120
概要.....	120
デザインの入力方法.....	120
使用可能な属性.....	120
詳細情報	121
MUXCY.....	122
概要.....	122
論理表.....	122
デザインの入力方法.....	122
詳細情報	123
MUXCY_D	124
概要.....	124
論理表.....	124
デザインの入力方法.....	124
詳細情報	125
MUXCY_L.....	126
概要.....	126
論理表.....	126
デザインの入力方法.....	126
詳細情報	127
MUXF5	128
概要.....	128
論理表.....	128
デザインの入力方法.....	128
詳細情報	129
MUXF5_D	130
概要.....	130
論理表.....	130
デザインの入力方法.....	130
詳細情報	131
MUXF5_L	132
概要.....	132
論理表.....	132
デザインの入力方法.....	132
詳細情報	133
MUXF6	134
概要.....	134
論理表.....	134
デザインの入力方法.....	134
詳細情報	135
MUXF6_D	136
概要.....	136
論理表.....	136
デザインの入力方法.....	136
詳細情報	137
MUXF6_L	138
概要.....	138
論理表.....	138
デザインの入力方法.....	138
詳細情報	139
MUXF7	140
概要.....	140
論理表.....	140
ポートの説明	140
デザインの入力方法.....	140
詳細情報	141

MUXF7_D	142
概要	142
論理表	142
ポートの説明	142
デザインの入力方法	142
詳細情報	143
MUXF7_L	144
概要	144
論理表	144
ポートの説明	144
デザインの入力方法	144
詳細情報	145
MUXF8	146
概要	146
論理表	146
ポートの説明	146
デザインの入力方法	146
詳細情報	147
MUXF8_D	148
概要	148
論理表	148
ポートの説明	148
デザインの入力方法	148
詳細情報	149
MUXF8_L	150
概要	150
論理表	150
ポートの説明	150
デザインの入力方法	150
詳細情報	151
OBUF	152
概要	152
ポートの説明	152
デザインの入力方法	152
使用可能な属性	152
詳細情報	153
OBUFDS	154
概要	154
論理表	154
ポートの説明	154
デザインの入力方法	154
使用可能な属性	154
詳細情報	155
OBUFT	156
概要	156
論理表	156
ポートの説明	156
デザインの入力方法	156
使用可能な属性	157
詳細情報	157
OBUFTDS	158
概要	158
論理表	158
ポートの説明	158
デザインの入力方法	158
使用可能な属性	158
詳細情報	159
ODDR2	160
概要	160

論理表.....	160
デザインの入力方法.....	160
使用可能な属性.....	161
詳細情報	162
PULLDOWN	163
概要.....	163
ポートの説明	163
デザインの入力方法.....	163
詳細情報	164
PULLUP.....	165
概要.....	165
ポートの説明	165
デザインの入力方法.....	165
詳細情報	166
RAM16X1D	167
概要.....	167
論理表.....	167
デザインの入力方法.....	168
使用可能な属性.....	168
詳細情報	169
RAM16X1S	170
概要.....	170
論理表.....	170
デザインの入力方法.....	170
使用可能な属性.....	170
詳細情報	171
RAM32X1S	172
概要.....	172
論理表.....	172
デザインの入力方法.....	172
使用可能な属性.....	173
詳細情報	173
RAM64X1S	174
概要.....	174
論理表.....	174
デザインの入力方法.....	174
使用可能な属性.....	174
詳細情報	175
RAMB16BWE	176
概要.....	176
ポートの説明	176
デザインの入力方法.....	177
使用可能な属性.....	178
詳細情報	182
RAMB16BWE_S18	183
概要.....	183
ポートの説明	183
デザインの入力方法.....	184
使用可能な属性.....	185
詳細情報	189
RAMB16BWE_S18_S18	190
概要.....	190
ポートの説明	191
デザインの入力方法.....	191
使用可能な属性.....	192
詳細情報	196
RAMB16BWE_S18_S9	197
概要.....	197
ポートの説明	197

デザインの入力方法.....	198
使用可能な属性.....	199
詳細情報	203
RAMB16BWE_S36	204
概要.....	204
ポートの説明	204
デザインの入力方法.....	205
使用可能な属性.....	206
詳細情報	210
RAMB16BWE_S36_S18	211
概要.....	211
ポートの説明	212
デザインの入力方法.....	212
使用可能な属性.....	213
詳細情報	217
RAMB16BWE_S36_S36	218
概要.....	218
ポートの説明	219
デザインの入力方法.....	219
詳細情報	223
RAMB16BWE_S36_S9	224
概要.....	224
ポートの説明	224
デザインの入力方法.....	225
使用可能な属性.....	226
詳細情報	230
RAMB16BWER	231
概要.....	231
ポートの説明	231
デザインの入力方法.....	232
使用可能な属性.....	233
詳細情報	234
ROM128X1	235
概要.....	235
論理表.....	235
デザインの入力方法.....	236
使用可能な属性.....	236
詳細情報	237
ROM16X1	238
概要.....	238
論理表.....	238
デザインの入力方法.....	239
使用可能な属性.....	239
詳細情報	239
ROM256X1	240
概要.....	240
論理表.....	240
デザインの入力方法.....	241
使用可能な属性.....	241
詳細情報	242
ROM32X1	243
概要.....	243
論理表.....	243
デザインの入力方法.....	244
使用可能な属性.....	244
詳細情報	244
ROM64X1	245
概要.....	245
論理表.....	245

デザインの入力方法.....	246
使用可能な属性.....	246
詳細情報	247
SPL_ACCESS	248
概要.....	248
ポートの説明	248
デザインの入力方法.....	248
使用可能な属性.....	248
詳細情報	250
SRL16.....	251
概要.....	251
論理表.....	251
デザインの入力方法.....	251
使用可能な属性.....	252
詳細情報	252
SRL16_1	253
概要.....	253
論理表.....	253
デザインの入力方法.....	253
使用可能な属性.....	254
詳細情報	254
SRL16E	255
概要.....	255
論理表.....	255
ポートの説明	256
デザインの入力方法.....	256
使用可能な属性.....	256
詳細情報	257
SRL16E_1.....	258
概要.....	258
論理表.....	258
デザインの入力方法.....	258
使用可能な属性.....	259
詳細情報	259
SRLC16.....	260
概要.....	260
論理表.....	260
デザインの入力方法.....	261
使用可能な属性.....	261
詳細情報	262
SRLC16_1	263
概要.....	263
論理表.....	263
デザインの入力方法.....	263
使用可能な属性.....	264
詳細情報	264
SRLC16E	265
概要.....	265
論理表.....	265
デザインの入力方法.....	266
使用可能な属性.....	266
詳細情報	267
SRLC16E_1	268
概要.....	268
論理表.....	268
デザインの入力方法.....	269
使用可能な属性.....	269
詳細情報	270
STARTUP_SPARTAN3A	271

概要.....	271
ポートの説明	271
デザインの入力方法.....	271
詳細情報	272
XORCY	273
概要.....	273
論理表.....	273
デザインの入力方法.....	273
詳細情報	274
XORCY_D	275
概要.....	275
論理表.....	275
デザインの入力方法.....	275
詳細情報	276
XORCY_L	277
概要.....	277
論理表.....	277
デザインの入力方法.....	277
詳細情報	278

第 1 章

このマニュアルについて

HDL 用ライブラリ ガイドは、ISE のオンライン マニュアルの 1 つです。回路図を使用して設計する場合は、回路図用ライブラリ ガイドを参照してください。

このマニュアルには、次の内容が含まれます。

- ・ 概要
- ・ ターゲット デバイス変更後のエレメントのリスト
- ・ このアーキテクチャでサポートされるプリミティブとマクロのファンクション別リスト
- ・ 各プリミティブの詳細説明

デザイン エレメント

このバージョンのライブラリ ガイドでは、このアーキテクチャのザイリンクス ユニファイド ライブラリを構成するプリミティブとインスタンス化コード例が含まれます。

プリミティブは、ターゲットにしている FPGA デバイス用のザイリンクス コンポーネントです。プリミティブをインスタンス化して変換プロセスを実行すると、変換後のファイルには同一のコンポーネントが含まれます。たとえば、ISERDES_NODELAY という Virtex-5 エレメントをユーザー プリミティブとしてインスタンス化し、変換 (NGDBuild) を実行すると、ISERDES_NODELAY がそのまま残ります。一方 Virtex-5 デバイスで ISERDES を使用していると、自動的に Virtex-5 用の ISERDES_NODELAY に変換されます。このため、「プリミティブ」の概念は、同じ分野でもユーザーによって異なります。

ザイリンクスでは、さまざまなデバイス アーキテクチャに対応した多数のデザイン エレメント (ユニマクロおよびプリミティブ) を含むソフトウェア ライブラリを提供しています。開発システム ソフトウェアのリリースごとに、新しいデザイン エレメントが組み込まれます。すべてのデザイン エレメントを含むユニファイド ライブラリに加えて、2003 年よりアーキテクチャごとに個別のライブラリが作成されました。このマニュアルは、そのようなアーキテクチャ固有のライブラリの 1 つです。

デザインの入力方法

ザイリンクスでは、デザインごとに次の 4 つの入力方法を考慮し、最適なソリューションを提供しています。

- ・ **インスタンス化**：デザインにコンポーネントが直接インスタンス化されます。これは、各ブロックをユーザーが正確に配置する場合に有効な方法です。
- ・ **推論**：コンポーネントはサポートされる合成ツールで推論されます。コードは柔軟性およびポータビリティに優れているので、複数のアーキテクチャに使用できます。推論を実行すると、パフォーマンス、エリア、電力などをユーザーが合成ツールで指定したとおりに最適化できます。
- ・ **CORE Generator およびウィザード**：コンポーネントは CORE Generator またはウィザードから使用できます。推論ができない FPGA を使用して大型なブロックを構築する場合には、この方法を使用してください。このフローを使用する場合は、ターゲットにするアーキテクチャごとにコアを再生成する必要があります。
- ・ **マクロのサポート**：使用可能な UniMacro があります。これらのコンポーネントはザイリンクス ツールの UniMacro ライブラリに含まれ、プリミティブだけでは複雑すぎてインスタンス化しにくいようなプリミティブをインスタンス化する際に使用します。合成ツールでは、この UniMacro が自動的に下位プリミティブに展開されます。

第 2 章

ターゲット デバイス変更後のデザイン エレメント

デザイン エレメントの中には、アーキテクチャを変更すると、最新の回路設計技術を最大限に利用できるように、ISE ソフトウェアにより自動的に変更されるものがあります。

次の表に、そのエレメントと変更後のアドバンス エレメントを示します。

元のデザイン エレメント	変更後のエレメント
BUFGCE_1	BUFGCE + INV
BUFGDLL	DCM_SP + BUFG
BUFGMUX_1	BUFGMUX + INV
BUFGP	BUFG
CAPTURE_SPARTAN3	CAPTURE_SPARTAN3a
CLKDLL	DCM_SP
CLKDLLE	DCM_SP
CLKDLLHF	DCM_SP
FD	FDCPE
FD_1	FDCPE + INV
FDC	FDCPE
FDC_1	FDCPE + INV
FDCE	FDCPE
FDCE_1	FDCPE + INV
FDCP	FDCPE
FDCP_1	FDCPE + INV
FDE	FDCPE
FDE_1	FDCPE + INV
FDPE	FDCPE
FDPE_1	FDCPE + INV
FDR	FDRSE
FDR_1	FDRSE + INV
FDRE	FDRSE
FDRE_1	FDRSE + INV

元のデザイン エLEMENT	変更後のELEMENT
FDRS	FDRSE
FDRS_1	FDRSE + INV
FDS	FDRSE
FDS_1	FDRSE + INV
FDSE	FDRSE
FDSE_1	FDRSE + INV
LD	LDCPE
LD_1	LDCPE + INV
LDC	LDCPE
LDC_1	LDCPE + INV
LDCE	LDCPE
LDCE_1	LDCPE + INV
LDE	LDCPE
LDE_1	LDCPE + INV
LDP	LDCPE
LDP_1	LDCPE + INV
LDPE	LDCPE
LDPE_1	LDCPE + INV
RAM128X1S_1	RAM128x1S + クロック上に INV
RAM16X1D_1	RAM16X1D + クロック上に INV
RAM16X1S_1	RAM16X1S + クロック上に INV
RAM16X2S	RAM16x1 X 2
RAM16X4S	RAM16x1 X 4
RAM16X8S	RAM16x1 X 8
RAM32X1D_1	RAM32X1D + クロック上に INV
RAM32X1S_1	RAM32X1S + クロック上に INV
RAM32X2S	RAM32x1S X 2
RAM32X4S	RAM32x1S X 4
RAM32X8S	RAM32x1S X 8
RAM64X1S_1	RAM64X1S + クロック上に INV
RAM64X2S	RAM64X1S X 2
RAMB16_S1_S1	RAMB16BWE
RAMB16_S1_S18	RAMB16BWE
RAMB16_S1_S2	RAMB16BWE
RAMB16_S1_S36	RAMB16BWE
RAMB16_S1_S4	RAMB16BWE
RAMB16_S1_S9	RAMB16BWE
RAMB16_S1	RAMB16BWE

元のデザイン エLEMENT	変更後のELEMENT
RAMB16_S18_S18	RAMB16BWE
RAMB16_S18_S36	RAMB16BWE
RAMB16_S18	RAMB16BWE
RAMB16_S2_S18	RAMB16BWE
RAMB16_S2_S2	RAMB16BWE
RAMB16_S2_S36	RAMB16BWE
RAMB16_S2_S4	RAMB16BWE
RAMB16_S2_S9	RAMB16BWE
RAMB16_S2	RAMB16BWE
RAMB16_S36_S36	RAMB16BWE
RAMB16_S36	RAMB16BWE
RAMB16_S4_S18	RAMB16BWE
RAMB16_S4_S36	RAMB16BWE
RAMB16_S4_S4	RAMB16BWE
RAMB16_S4_S9	RAMB16BWE
RAMB16_S4	RAMB16BWE
RAMB16_S9_S18	RAMB16BWE
RAMB16_S9_S36	RAMB16BWE
RAMB16_S9_S9	RAMB16BWE
RAMB16_S9	RAMB16BWE
RAMB4_S1_S1	RAMB16BWE
RAMB4_S1_S16	RAMB16BWE
RAMB4_S1_S2	RAMB16BWE
RAMB4_S1_S4	RAMB16BWE
RAMB4_S1_S8	RAMB16BWE
RAMB4_S1	RAMB16BWE
RAMB4_S16_S16	RAMB16BWE
RAMB4_S16	RAMB16BWE
RAMB4_S2_S16	RAMB16BWE
RAMB4_S2_S2	RAMB16BWE
RAMB4_S2_S4	RAMB16BWE
RAMB4_S2_S8	RAMB16BWE
RAMB4_S2	RAMB16BWE
RAMB4_S4_S16	RAMB16BWE
RAMB4_S4_S4	RAMB16BWE
RAMB4_S4_S8	RAMB16BWE
RAMB4_S4	RAMB16BWE
RAMB4_S8_S16	RAMB16BWE

元のデザイン エLEMENT	変更後のELEMENT
RAMB4_S8_S8	RAMB16BWE
STARTUP_SPARTAN3	STARTUP_SPARTAN3A

ファンクション別分類

このセクションでは、デバイスに含まれるデザイン エLEMENTをファンクション別に分類して示します。ELEMENT (プリミティブおよびマクロのインプリメンテーション) は、各カテゴリでアルファベット順にリストしています。

四則演算ファンクション	RAM/ROM
クロック コンポーネント	レジスタおよびラッチ
コンフィギュレーション/BSCAN コンポーネント	シフトレジスタ
I/O コンポーネント	スライス/CLB プリミティブ

四則演算ファンクション

デザイン エLEMENT	説明
DSP48A	プリミティブ : Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block
MULT18X18SIO	プリミティブ : 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset

クロック コンポーネント

デザイン エLEMENT	説明
BUFG	コンビニエンス プリミティブ : Global Clock Buffer
BUFGCE	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable
BUFGMUX	コンビニエンス プリミティブ : Global Clock MUX Buffer
DCM_SP	プリミティブ : Digital Clock Manager
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay

コンフィギュレーション/BSCAN コンポーネント

デザイン エLEMENT	説明
BSCAN_SPARTAN3A	プリミティブ : Spartan®-3A JTAG Boundary Scan Logic Access Circuit
CAPTURE_SPARTAN3A	プリミティブ : Spartan®-3A Register State Capture for Bitstream Readback
DNA_PORT	プリミティブ : Device DNA Data Access Port
ICAP_SPARTAN3A	プリミティブ : Internal Configuration Access Port
SPL_ACCESS	プリミティブ : Internal Logic Access to the Serial Peripheral Interface (SPI) PROM Data
STARTUP_SPARTAN3A	プリミティブ : Spartan®-3A Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface

I/O コンポーネント

デザイン エLEMENT	説明
IBUF	プリミティブ : Input Buffer
IBUF_DLY_ADJ	プリミティブ : Dynamically Adjustable Input Delay Buffer
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFDS_DLY_ADJ	プリミティブ : Dynamically Adjustable Differential Input Delay Buffer
IBUFG	プリミティブ : Dedicated Input Clock Buffer
IBUFGDS	プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay
IOBUF	プリミティブ : Bi-Directional Buffer
IOBUFDS	プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable
KEEPER	プリミティブ : KEEPER Symbol
OBUF	プリミティブ : Output Buffer
OBUFDS	プリミティブ : Differential Signaling Output Buffer
OBUFFT	プリミティブ : 3-State Output Buffer with Active Low Output Enable
OBUFFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
PULLDOWN	プリミティブ : Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs
PULLUP	プリミティブ : Resistor to VCC for Input PADS, Open-Drain, and 3-State Outputs

RAM/ROM

デザイン エレメント	説明
RAM16X1D	プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM16X1S	プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM
RAM32X1S	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM
RAM64X1S	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM
RAMB16BWE	プリミティブ : 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM
RAMB16BWE_S18	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 18-bit Port
RAMB16BWE_S18_S18	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit Ports
RAMB16BWE_S18_S9	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit and 9-bit Ports
RAMB16BWE_S36	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 36-bit Port
RAMB16BWE_S36_S18	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 18-bit Ports
RAMB16BWE_S36_S36	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit Ports
RAMB16BWE_S36_S9	プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 9-bit Ports
RAMB16BWER	プリミティブ : 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
ROM128X1	プリミティブ : 128-Deep by 1-Wide ROM
ROM16X1	プリミティブ : 16-Deep by 1-Wide ROM
ROM256X1	プリミティブ : 256-Deep by 1-Wide ROM
ROM32X1	プリミティブ : 32-Deep by 1-Wide ROM
ROM64X1	プリミティブ : 64-Deep by 1-Wide ROM

レジスタおよびラッチ

デザイン エLEMENT	説明
FDCPE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
FDCPE_1	プリミティブ : D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Preset and Clear
FDRSE	プリミティブ : D Flip-Flop with Synchronous Reset and Set and Clock Enable
FDRSE_1	プリミティブ : D Flip-Flop with Negative-Clock Edge, Synchronous Reset and Set, and Clock Enable
IDDR2	プリミティブ : Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset
LDCPE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable
ODDR2	プリミティブ : Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset

シフト レジスタ

デザイン エLEMENT	説明
SRL16	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT)
SRL16_1	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock
SRL16E	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable
SRL16E_1	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable
SRLC16	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry
SRLC16_1	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock
SRLC16E	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable
SRLC16E_1	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable

スライス/CLB プリミティブ

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

第 4 章

デザイン エLEMENT

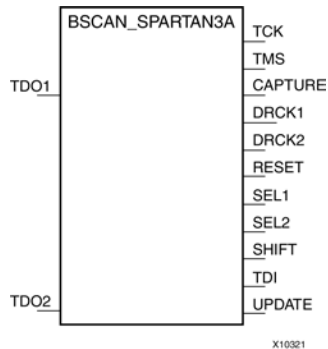
このセクションでは、このアーキテクチャで利用できるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

各ライブラリ エLEMENTについて、次の情報を示します。

- ・ 名称
- ・ 説明
- ・ 回路図シンボル (該当するELEMENTでのみ)
- ・ 論理表 (該当するELEMENTでのみ)
- ・ ポートの説明
- ・ デザインの入力方法
- ・ 使用可能な属性 (該当するELEMENTでのみ)
- ・ インスタンスエーション コードの例
- ・ その他のリソース

BSCAN_SPARTAN3A

プリミティブ : Spartan®-3A JTAG Boundary Scan Logic Access Circuit



概要

このデザイン エレメントを使用すると、JTAG バウンダリ スキャン ロジック コントローラを介して内部ロジックへアクセスできるので、内部実行デザインと FPGA の専用 JTAG ピン間の通信を可能にします。

メモ : 各アーキテクチャのバウンダリ スキャンの詳細については、データシートを参照してください。

ポートの説明

ポート名	方向	幅	機能
TDI	出力	1	FPGA の TDI 入力ピンと同じ値を出力します。
DRCK1、DRK2	出力	1	JTAG USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラが SHIFT-DR ステートになると TCK ピンと同じ値を出力します。DRK1 は USER1 ロジックに適用され、DRK2 は USER2 に適用されます。
RESET	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラが TEST-LOGIC-RESET ステートになると High にアサートされます。
SEL1、SEL2	出力	1	JTAG 命令レジスタに USER1 または USER2 命令が読み込まれたことを示します。UPDATE-IR ステートになるとアクティブになり、新しい命令が読み込まれるまでアクティブのままになります。
SHIFT	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラが SHIFT-DR ステートになると High にアサートされます。
CAPTURE	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラが CAPTURE-DR ステートになると High にアサートされます。
UPDATE	出力	1	USER 命令が読み込まれるとアクティブになり、JTAG TAP コントローラが UPDATE-DR ステートになると High にアサートされます。
TCK	出力	1	
TMS	出力	1	
TDO1、TDO2	入力	1	USER1 または USER2 命令が読み込まれるとアクティブになり、外部 JTAG TDO ピンにコンポーネントの TDO1 (USER1) または TDO2 (USER2) ピンへのデータ入力の値が反映されます。

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- BSCAN_SPARTAN3A: Boundary Scan primitive for connecting internal logic to
--                   JTAG interface.
--                   Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

BSCAN_SPARTAN3A_inst : BSCAN_SPARTAN3A
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
    TCK => TCK,          -- TCK output from TAP controller
    TDI => TDI,          -- TDI output from TAP controller
    TMS => TMS,          -- TMS 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_SPARTAN3A_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// BSCAN_SPARTAN3A: Boundary Scan primitive for connecting internal logic to
//                   JTAG interface.
//                   Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

BSCAN_SPARTAN3A BSCAN_SPARTAN3A_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
    .TCK(TCK),         // TCK output from TAP controller
    .TDI(TDI),         // TDI output from TAP controller
    .TMS(TMS),         // TMS 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_SPARTAN3A_inst instantiation
```

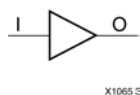
詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

BUFCF

プリミティブ : Fast Connect Buffer

BUFCF



概要

このデザイン エLEMENTは、一部の専用ロジックと LUT の出力を別の LUT の入力に直接接続するために使用する、単一の高速結合バッファです。このバッファを使用すると、CLB パックも行われます。LUT は、4 つまで 1 つのグループとして接続できます。

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

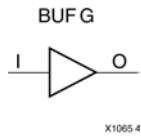
// End of BUFCF_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

BUFG

プリミティブ : Global Clock Buffer



概要

このデザイン エLEMENTは、ファンアウトが大きいバッファで、スキューを抑えて信号を分散するために、グローバル配線リソースへの信号に接続します。BUFG は通常クロック ネットで使用されます。

ポートの説明

ポート名	タイプ	幅	機能
I	入力	1	クロック バッファ出力
O	出力	1	クロック バッファ入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

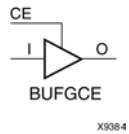
// End of BUFG_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

BUFGCE

プリミティブ : Global Clock Buffer with Clock Enable



概要

このデザイン エLEMENTは、クロック イネーブル付きグローバル クロック バッファです。O 出力は、クロック イネーブル (CE) が Low (非アクティブ) のときに 0 になります。CE が High になると、I 入力の値が O に出力されます。

論理表

入力		出力
I	CE	O
X	0	0
I	1	I

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

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

BUFGCE_inst : BUFGCE
port map (
    O => O,    -- Clock buffer output
    CE => CE,  -- Clock enable input
    I => I     -- Clock buffer input
);

-- End of BUFGCE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// 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
);

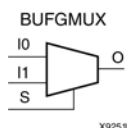
// End of BUFGCE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

BUFGMUX

プリミティブ : Global Clock MUX Buffer



概要

BUFGMUX はマルチプレクサの機能を持つグローバル クロック バッファで、2 つの入力クロック (I0 および I1) のいずれかを選択できます。セレクト入力 (S) が Low の場合、I0 の信号が出力 (O) に選択されます。S が High の場合は、I1 の信号が O に選択されます。

BUFGMUX および BUFGMUX_1 では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX_1 は出力ステートが 1 に保持されます。

メモ : BUFGMUX では、S がトグルされると、次のアクティブ クロック エッジ (I0 または I1) まで、出力のステートが非アクティブのまま保持されます。

論理表

入力			出力
I0	I1	S	O
I0	X	0	I0
X	I1	1	I1
X	X	↑	0
X	X	↓	0

ポートの説明

ポート名	タイプ	幅	機能
I0	入力	1	クロック 0 入力
I1	入力	1	クロック 1 入力
O	出力	1	クロック MUX 出力
S	入力	1	クロック セレクト入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
CLK_SEL_TYPE	文字列	SYNC、ASYNC	SYNC	同期クロックまたは非同期クロックを指定します。
DISABLE_VALUE	文字列	HIGH、LOW	LOW	入力を切り替えるときに出力が想定するステートを指定します。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

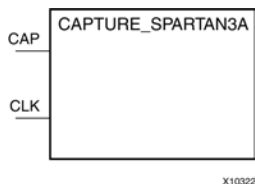
// End of BUFGMUX_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

CAPTURE_SPARTAN3A

プリミティブ : Spartan®-3A Register State Capture for Bitstream Readback



概要

このデザイン エLEMENTは、レジスタ (フリップフロップとラッチ) 情報のキャプチャ方法およびそのタイミングを制御します。リードバック機能は、専用のコンフィギュレーション ポート命令により提供されます。このELEMENTを使用しない場合は、データはコンフィギュレーション クロックに同期してリードバックされます。レジスタ (フリップフロップとラッチ) の値のみをキャプチャできます。LUT RAM、SRL、ブロック RAM の値もリードバックされますが、キャプチャできません。

CAP 信号を High にアサートすると、次にクロックが Low から High に切り替わる時にデバイス内のレジスタがキャプチャされます。デフォルトでは、トリガ (CAP をアサートしているときの CLK の遷移) のたびにデータがキャプチャされます。リードバック処理を 1 回のデータ キャプチャだけに制限するには、このELEMENTに ONESHOT=TRUE 属性を追加します。

ポートの説明

ポート名	方向	幅	機能
CAP	入力	1	リードバック キャプチャトリガ
CLK	入力	1	リードバック キャプチャ クロック

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

正しく動作するように、入力および出力をすべてデザインに接続します。

使用可能な属性

属性	タイプ	値	デフォルト	説明
ONESHOT	ブール代数	TRUE、FALSE	TRUE	CAP トリガごとに 1 回のリードバックを実行します。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- CAPTURE_SPARTAN3A: Register State Capture for Bitstream Readback
--                      Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

CAPTURE_SPARTAN3A_inst : CAPTURE_SPARTAN3A
generic map (
    ONESHOT => TRUE) -- TRUE or FALSE
port map (
    CAP => CAP,      -- Capture input
    CLK => CLK       -- Clock input
);

-- End of CAPTURE_SPARTAN3A_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// CAPTURE_SPARTAN3A: Register State Capture for Bitstream Readback
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

CAPTURE_SPARTAN3A #(
    .ONESHOT("TRUE") // "TRUE" or "FALSE"
) CAPTURE_SPARTAN3A_inst (
    .CAP(CAP),      // Capture input
    .CLK(CLK)      // Clock input
);

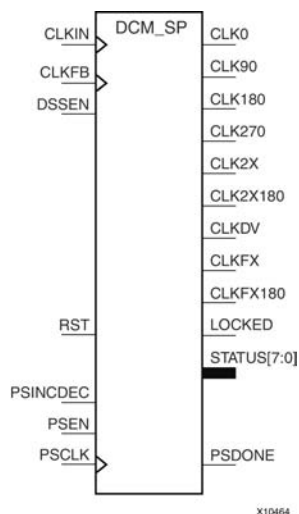
// End of CAPTURE_SPARTAN3A_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

DCM_SP

プリミティブ : Digital Clock Manager



概要

このデザイン エLEMENTは、さまざまな機能を備えたデジタル クロック マネージャで、クロック遅延ロック ループ (DLL)、デジタル周波数合成 (DFS)、デジタル位相シフト (DPS) といった機能をインプリメントできます。DCM_SP は、オンチップおよびオフチップからのクロック遅延をなくしたり、データ キャプチャを向上するためにクロック位相をシフトしたり、異なる周波数のクロックを生成させる場合などに便利です。

ポートの説明

ポート名	タイプ	幅	機能
CLKDV	出力	1	CLK0 を分周したクロック。分周値はプログラム可能です。
CLKFB	入力	1	DCM へのフィードバック クロック入力。フィードバック入力は DFS を単独で使用する場合以外で必要です。CLKFB のソースは DCM の CLK0 または CLK2X にする必要があります。
CLKFX	出力	1	デジタル周波数合成 (DFS) 出力
CLKFX180	出力	1	CLKFX クロックが 180 度シフトされたクロック
CLKIN	入力	1	DCM のクロック入力
CLK0	出力	1	CLKIN と同じ周波数で、0 度位相シフトしたクロック
CLK2X	出力	1	CLKIN 周波数クロックを 2 倍したクロック、CLK0 と同相
CLK2X180	出力	1	CLK2X クロックが 180 度位相シフトされたクロック
CLK90	出力	1	CLKIN と同じ周波数で、90 度位相シフトしたクロック
CLK180	出力	1	CLKIN と同じ周波数で、180 度位相シフトしたクロック
CLK270	出力	1	CLKIN と同じ周波数で、180 度位相シフトしたクロック
LOCKED	出力	1	DCM がロックしたことを示す信号
PSCLK	入力	1	位相シフトクロック入力。DCM 位相シフトのソース クロックを供給します。

ポート名	タイプ	幅	機能
PSDONE	出力	1	可変位相シフトが完了したことを示す出力信号
PSEN	入力	1	可変位相シフト イネーブル信号、PSCLK と同相
PSINCDEC	入力	1	位相シフト インクリメント/デクリメント (PSINCDEC) 入力信号は、PSCLK と同相である必要があります。この信号は、位相シフト係数をインクリメント/デクリメントするために使用します。信号が High の場合はインクリメント、Low の場合はデクリメントされます。
RST	入力	1	DCM 回路をリセットします。RST 信号は、アクティブ High の非同期リセットです。
STATUS[7:0]	出力	8	ステータス出力バスにより DCM のステータスが示されます。 <ul style="list-style-type: none"> STATUS[0] : 可変位相シフト オーバーフロー STATUS[1] : CLKIN の停止 STATUS[2] : CLKFX または CLKFX180 の停止 STATUS[7] : CLKFB の停止 STATUS[6:3] : 予約

デザインの入力方法

インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
CLKDV_DIVIDE	1 上位ビット 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	CLKDLL、CLKDLLE、CLKDLLHF、 DCM_SP のクロック分周器 (CLKDV 出力) の分周比を指定
CLK_FEEDBACK	文字列	1X、2X、NONE	1X	DCM フィードバック モードを定義します。 <ul style="list-style-type: none"> 1X : CLK0 をフィードバックにします。 2X : CLK2X をフィードバックにします。
CLKFX_DIVIDE	整数	1 ~ 32	1	CLKFX 出力の分周比を指定
CLKFX_MULTIPLY	整数	2 ~ 32	4	CLKFX 出力の通倍比を指定
CLKIN_DIVIDE_BY_2	ブール代数	FALSE、TRUE	FALSE	CLKIN を 2 で分周
CLKIN_PERIOD	文字列	0 ビット文字列	10.0	CLKIN 入力への入力周期を指定 (ns)

属性	タイプ	値	デフォルト	説明
CLKOUT_PHASE_SHIFT	文字列	NONE、FIXED、VARIABLE	NONE	位相シフト モードを指定 <ul style="list-style-type: none"> ・ NONE : 位相シフト機能なし。設定されている値は反映されません。 ・ FIXED : DCM の出力は CLKIN から決まった位相だけシフトされたものになります。値は PHASE_SHIFT 属性で指定されます。 ・ VARIABLE : DCM 出力が CLKIN に相対して正および負の範囲にシフトできます。開始値は PHASE_SHIFT 属性で指定されます。
DESKEW_ADJUST	文字列	SYSTEM_SYNCHRONOUS、SOURCE_SYNCHRONOUS	SYSTEM_SYNCHRONOUS	DCM_SP クロック出力と FPGA のクロック入力ピン間のクロック遅延の配分に影響するコンフィギュレーションビットを設定
DLL_FREQUENCY_MODE	文字列	LOW、HIGH	LOW	AUTO モードでは DLL により自動的に周波数が検索され、DLL が LOW モードまたは HIGH モードで動作するかが決定されます。AUTO モードはレガシ属性です。HIGH および LOW が影響しないようなとき、常に AUTO モードが使用されます。
DSS_MODE	文字列	NONE、SPREAD_2、SPREAD_4、SPREAD_6、SPREAD_8	NONE	出力クロックの周波数拡散を指定します。 <ul style="list-style-type: none"> ・ NONE : デフォルト。拡散係数が指定されていません。デジタル スペクトラム拡散はディスエーブルにされています。 ・ SPREAD_2 : 現時点のクロック周期 $\pm 50\text{ps}$ のクロック周期を作成します。 ・ SPREAD_4 : 現時点のクロック周期 $\pm 100\text{ps}$ のクロック周期を作成します。 ・ SPREAD_6 : 現時点のクロック周期 $\pm 150\text{ps}$ のクロック周期を作成します。 ・ SPREAD_8 : 現時点のクロック周期 $\pm 200\text{ps}$ のクロック周期を作成します。 SPREAD_# の数値が増えると、拡散は累積的に追加されます。たとえば、SPREAD_2 と指定すると、入力クロック周波数 $\pm 50\text{ps}$ の 2 つのクロック周波数が追加されます。SPREAD_4 と指定すると、さらに $\pm 100\text{ps}$ の 2 つのクロック周波数が追加されます。
DUTY_CYCLE_CORRECTION	ブール代数	TRUE、FALSE	TRUE	CLK0、CLK90、CLK180、CLK270 の各出力のデューティ サイクルを修正
PHASE_SHIFT	整数	-255 ~ 255	0	固定位相シフトを -255 ~ 255 の値に定義
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	DCM がロックされるまでコンフィギュレーション DONE を遅らせます。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```

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
                                -- DCM_SP clock divider (CLKDV output) is to be frequency divided.
    CLKFX_DIVIDE => 1,            -- Specifies the frequency divider value for the CLKFX output.
    CLKFX_MULTIPLY => 4,          -- Specifies the frequency multiplier value for the CLKFX output.
    CLKIN_DIVIDE_BY_2 => FALSE,   -- Enables CLKIN divide by two features.
    CLKIN_PERIOD => "10.0",       -- Specifies the input period to the DCM_SP CLKIN input in ns.
    CLKOUT_PHASE_SHIFT => "NONE", -- This attribute specifies the phase shift mode. NONE = No phase
                                -- shift capability. Any set value has no effect. FIXED = DCM
                                -- outputs are a fixed phase shift from CLKIN. Value is specified
                                -- by PHASE_SHIFT attribute. VARIABLE = Allows the DCM outputs to
                                -- be shifted in a positive and negative range relative to CLKIN.
                                -- Starting value is specified by PHASE_SHIFT.
    CLK_FEEDBACK => "1X",         -- Defines the DCM feedback mode. 1X: CLK0 as feedback 2X: CLK2X
                                -- as feedback.
    DESKEW_ADJUST => "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 => "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 => "NONE",          -- Corrects the duty cycle of the CLK0, CLK90, CLK180, and CLK270
    DUTY_CYCLE_CORRECTION => TRUE, -- outputs.
    PHASE_SHIFT => 0,            -- Defines the amount of fixed phase shift from -255 to 255
    STARTUP_WAIT => FALSE        -- Delays configuration DONE until DCM LOCK.
)
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
                                -- 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 記述 (インスタンス化)

```
// 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
);

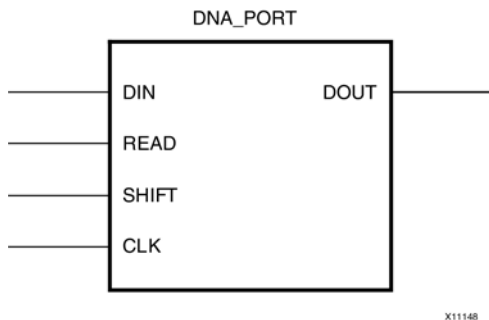
// End of DCM_SP_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

DNA_PORT

プリミティブ : Device DNA Data Access Port



概要

DNA_PORT を使用すると専用のシフトレジスタにアクセスできます。このシフトレジスタにはデバイスの Device DNA データビット (重複しない ID) が読み込まれます。このコンポーネントを使用すると、DNA データビットがシフトアウトできるだけでなく、補足データビットを含めたり、DNA データをロールオーバー (シフトアウト後の DNA データの反復) することもできます。このコンポーネントは主に FPGA ビットストリームの不正なコピー防止用に、ほかの回路と合わせて使用されます。

ポートの説明

ポート名	タイプ	幅	機能
CLK	入力	1	シフトレジスタへの入力クロック
DIN	入力	1	シフトレジスタへのユーザー データ入力
DOUT	出力	1	シリアル シフト出力レジスタ
READ	入力	1	シフトレジスタの同期ロード
SHIFT	入力	1	アクティブ High のシフトイネーブル入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

正しく動作するように、入力および出力をすべてデザインに接続します。

Device DNA データにアクセスするにはまず、アクティブ High の READ 信号を 1 クロック サイクル間 High にしてシフトレジスタを読み込む必要があります。シフトレジスタが読み込まれると、アクティブ High の SHIFT 入力がイネーブルになり、DOUT 出力ポートからデータが取り込まれ、データが同期シフトアウトします。追加のデータがある場合は、適切なロジックを DIN ポートに接続すると、57 ビットのシフトレジスタの終わりに追加できます。DNA データをロールオーバーする場合は、DOUT ポートを直接 DIN ポートに接続し、57 ビットのシフト操作の後で同じデータがシフトアウトされるようにします。追加データが必要ない場合は、DIN ポートを論理値 0 に固定できます。属性 SIM_DNA_VALUE はオプションで、DNA データシーケンスをシミュレーションするように設定できます。デフォルトでは、シミュレーション モデルの Device DNA データビットはすべて 0 です。

使用可能な属性

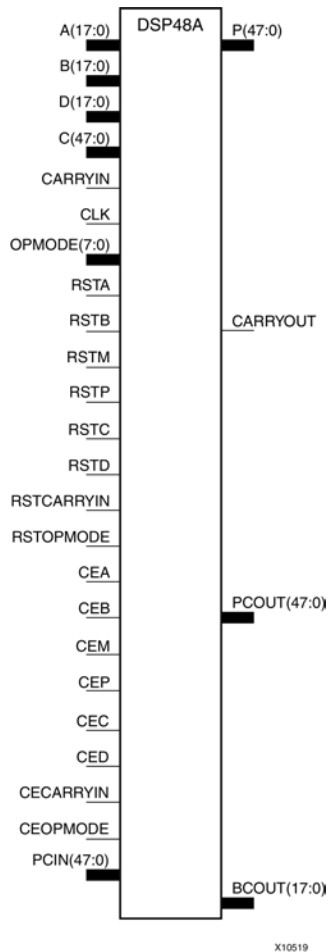
属性	タイプ	値	デフォルト	説明
SIM_DNA_VALUE	16 進数	57'h00000000 0000000 ~ 57'h1fffffffff	57'h00000000 0000000	シミュレーションに使用する DNA 値を指定 (実際の値は使用デバイス特有)

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

DSP48A

プリミティブ : Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block



概要

DSP48A は、柔軟性が高い多用途のハード IP ブロックで、多くの DSP アルゴリズムで見られる小型で高速な演算処理を作成できます。このブロックでは、コンフィギュレーション可能な 18 ビットの加減算器に、18 X 18 符号付き乗算器、48 ビットの加減/アキュムレータが順に付けられています。このブロックには、コンフィギュレーション可能なパイプラインレジスタが数個含まれているため、追加されるレイテンシをトレードオフにする高速クロックを実現できます。OpMode ピンでは、ブロック操作を 1 クロック サイクルから次サイクルに変更でき、デザインに含まれる複数の演算ファンクションに 1 つのブロックを使用できます。さらに、複数の MSPI ブロックをカスケード接続して大型の乗算および加算ファンクションを作成できます。

ポートの説明

ポート名	方向	幅	機能
データ ポート			
A	入力	18	OPMODE[1:0] の値に従い乗算器または後置加減算器に送られる 18 ビット データ 入力
B	入力	18	OPMODE[3:0] の値に従い乗算器、前置加減算器、または後置加減算器に送られる 18 ビット データ入力

ポート名	方向	幅	機能
C	入力	48	後置加減算器への 48 ビット入力
D	入力	18	前置加減算器への 18 ビット入力
CARRYIN	入力	1	後置加減算器への外部キャリー入力。別の DSP48A ブロックの CARRYOUT ピンにのみ接続します。
P	出力	48	プライマリ データ出力
CARRYOUT	出力	1	後置加減算器の外部キャリー出力信号。別の DSP48A ブロックの CARRYIN ピンにのみ接続します。
制御入力			
CLK	入力	1	DSP48A クロック
OPMODE	入力	8	DSP48A の演算処理を選択する制御入力
OPMODE[1:0]			後置加減算器への X 入力のソースを指定します。 <ul style="list-style-type: none"> 0 : すべてゼロを配置します (後置加減算器をディスエーブルにします)。 1 : POUT 出力信号を使用します。 2 : 連結された D、B、A 入力信号を使用します。 3 : 乗算器の積項を使用します。
OPMODE[3:2]			後置加減算器への Y 入力のソースを指定します。 <ul style="list-style-type: none"> 0 : すべてゼロを配置します (後置加減算器をディスエーブルにし、乗算器の積項を POUT に伝搬します)。 1 : PCIN を使用します。 2 : POUT ポート (アキュムレータ) を使用します。 3 : C ポートを使用します。
OPMODE[4]			前置加減算器を使用するように指定します。 <ul style="list-style-type: none"> 0 : 乗算器の前で B および D ポートに前置加減算器の加算または減算値を使用します。 1 : 前置加算器をバイパスして、ポート B のデータを直接乗算器に送ります。
OPMODE[5]			キャリー入力の値を後置加算器に送ります。これは CARRYINSEL = OPMODE5 のときにのみ適用されます。
OPMODE[6]			前置加減算器が加算器なのか減算器なのかを指定します。 <ul style="list-style-type: none"> 0 : 加算を実行します。 1 : 減算を実行します。
OPMODE[7]			後置加減算器が加算器なのか減算器なのかを指定します。 <ul style="list-style-type: none"> 0 : 加算を実行します。 1 : 減算を実行します。
リセット/クロック イネーブル入力			
RSTA	入力	1	A ポートレジスタ (AREG=1 または 2) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。

ポート名	方向	幅	機能
RSTB	入力	1	B ポートレジスタ (BREG!=NONE) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTD	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTM	入力	1	乗算レジスタ (MREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTP	入力	1	P 出力レジスタ (PREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTCARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTOPMODE	入力	1	OPMODE レジスタ (OPMODEREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
CEA	入力	1	A ポートレジスタ (A0REG=1 または A1REG=1) のアクティブ High のクロック イネーブルで、使用しない場合と使用しない場合と A0REG=1 または A1REG=1 の場合は論理値を 1 に、A0REG=0 または A1REG=0 の場合は論理値を 0 にします。
CEB	入力	1	B ポートレジスタ (B0REG=1 または B1REG=1) のアクティブ High のクロック イネーブルで、使用しない場合や B0REG=1 または B1REG=1 の場合は、論理値を 1 にします。B0REG=0 または B1REG=0 の場合は論理値を 0 にします。
CEC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と CREG=1 の場合は論理値を 1 に、CREG=0 の場合は論理値を 0 にします。
CED	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と DREG=1 の場合は論理値を 1 に、DREG=0 の場合は論理値を 0 にします。
CEM	入力	1	乗算レジスタ (MREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と MREG=1 の場合は論理値を 1 に、MREG=0 の場合は論理値を 0 にします。
CEP	入力	1	出力ポートレジスタ (PREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と PREG=1 の場合は論理値を 1 に、PREG=0 の場合は論理値を 0 にします。
CECARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と CARRYINREG=1 の場合は論理値を 1 に、CARRYINREG=0 の場合は論理値を 0 にします。
CEOPMODE	入力	1	OPMODE 入力レジスタ (OPMODEREG=1) のクロック イネーブルで、使用しない場合と OPMODEREG=1 の場合は論理値を 1 に、OPMODEREG=0 の場合は論理値を 0 にします。
カスケード ポート			
PCIN	入力	48	ポート P のカスケード入力、カスケード接続されている上位の DSP48A の PCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
PCOUT	出力	48	ポート P のカスケード出力、カスケード接続されている下位の DSP48A の PCIN に接続します。使用しない場合は未接続にします。
BCOUT	出力	18	ポート B のカスケード出力、カスケード接続されている下位の DSP48A の B ポートに接続します。使用しない場合は未接続にします。

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
A0REG	整数	0、1	0	1 段目の A 入力にレジスタを付けるかを指定します。
A1REG	整数	0、1	1	2 段目の A 入力にレジスタを付けるかを指定します。
B0REG	整数	0、1	0	1 段目の B 入力にレジスタを付けるかを指定します。
B1REG	整数	0、1	1	2 段目の B 入力にレジスタを付けるかを指定します。
CARRYINREG	整数	0、1	1	CARRYIN 入力にレジスタを付けるかを指定します。 CARRYINSEL 属性が CARRYIN に設定されており CARRYIN が使用されているときのみ使用します。
CARRYINSEL	文字列	CARRYIN、 OPMODE5	CARRYIN	後置加減算器のキャリー入力信号を別の DSP48A の CARRYOUT ピンに接続されている CARRYIN ピンから送 るか、OPMODE[5] 入力を使用して FPGA から直接制御 するか指定します。
CREG	整数	0、1	1	C 入力にレジスタを付けるかを指定します。
DREG	整数	0、1	1	D 入力にレジスタを付けるかを指定します。
MREG	整数	0、1	1	乗算器の段にレジスタを付けるかを指定します。イネー ブル = 1/ディスエーブル = 0
OPMODEREG	整数	0、1	1	OPMODE 入力にレジスタを付けるかを指定します。
PREG	整数	0、1	1	C 入力にレジスタを付けるかを指定します。
RSTTYPE	文字列	ASYNC、SYNC	SYNC	リセットすべてを同期リセットにするか非同期リセットにす るか指定します。タイミングの向上と回路の安定性の点 から、非同期リセットが必要でない限り常に SYNC に設 定してください。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- DSP48A: DSP Function Block
--      Spartan-3A DSP
-- Xilinx HDL Libraries Guide, version 11.2

DSP48A_inst : DSP48A
generic map (
  A0REG => 1,      -- Enable=1/disable=0 first stage A input pipeline register
  A1REG => 1,      -- Enable=1/disable=0 second stage A input pipeline register
  B0REG => 1,      -- Enable=1/disable=0 first stage B input pipeline register
  B1REG => 1,      -- Enable=1/disable=0 second stage B input pipeline register
  CARRYINREG => 1, -- Enable=1/disable=0 CARRYIN input pipeline register
  CARRYINSEL => "CARRYIN", -- Specify carry-in source, "CARRYIN" or "OPMODE5"
  CREG => 1,      -- Enable=1/disable=0 C input pipeline register
  DREG => 1,      -- Enable=1/disable=0 D pre-adder input pipeline register
```

```

MREG => 1,          -- Enable=1/disable=0 M pipeline register
OPMODEREG => 1,      -- Enable=1/disable=0 OPMODE input pipeline register
PREG => 1,          -- Enable=1/disable=0 P output pipeline register
RSTTYPE => "SYNC") -- Specify reset type, "SYNC" or "ASYNC"
port map (
  BCOUT => BCOUT,    -- 18-bit B port cascade output
  CARRYOUT => CARRYOUT, -- 1-bit carry output
  P => P,            -- 48-bit output
  PCOUT => PCOUT,    -- 48-bit cascade output
  A => A,            -- 18-bit A data input
  B => B,            -- 18-bit B data input (can be connected to fabric or BCOUT of adjacent DSP48A)
  C => C,            -- 48-bit C data input
  CARRYIN => CARRYIN, -- 1-bit carry input signal
  CEA => CEA,        -- 1-bit active high clock enable input for A input registers
  CEB => CEB,        -- 1-bit active high clock enable input for B input registers
  CEC => CEC,        -- 1-bit active high clock enable input for C input registers
  CECARRYIN => CECARRYIN, -- 1-bit active high clock enable input for CARRYIN registers
  CED => CED,        -- 1-bit active high clock enable input for D input registers
  CEM => CEM,        -- 1-bit active high clock enable input for multiplier registers
  CEOPMODE => CEOPMODE, -- 1-bit active high clock enable input for OPMODE registers
  CEP => CEP,        -- 1-bit active high clock enable input for P output registers
  CLK => CLK,        -- Clock input
  D => D,            -- 18-bit B pre-adder data input
  OPMODE => OPMODE,  -- 8-bit operation mode input
  PCIN => PCIN,      -- 48-bit P cascade input
  RSTA => RSTA,      -- 1-bit reset input for A input pipeline registers
  RSTB => RSTB,      -- 1-bit reset input for B input pipeline registers
  RSTC => RSTC,      -- 1-bit reset input for C input pipeline registers
  RSTCARRYIN => RSTCARRYIN, -- 1-bit reset input for CARRYIN input pipeline registers
  RSTD => RSTD,      -- 1-bit reset input for D input pipeline registers
  RSTM => RSTM,      -- 1-bit reset input for M pipeline registers
  RSTOPMODE => RSTOPMODE, -- 1-bit reset input for OPMODE input pipeline registers
  RSTP => RSTP      -- 1-bit reset input for P pipeline registers
);

-- End of DSP48A_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// DSP48A: DSP Function Block
//      Spartan-3A DSP
// Xilinx HDL Libraries Guide, version 11.2

DSP48A #(
    .AOREG(0), // Enable=1/disable=0 first stage A input pipeline register
    .AIREG(1), // Enable=1/disable=0 second stage A input pipeline register
    .BOREG(0), // Enable=1/disable=0 first stage B input pipeline register
    .BIREG(1), // Enable=1/disable=0 second stage B input pipeline register
    .CARRYINREG(1), // Enable=1/disable=0 CARRYIN input pipeline register
    .CARRYINSEL("CARRYIN"), // Specify carry-in source, "CARRYIN" or "OPMODE5"
    .CREG(1), // Enable=1/disable=0 C input pipeline register
    .DREG(1), // Enable=1/disable=0 D pre-adder input pipeline register
    .MREG(1), // Enable=1/disable=0 M pipeline register
    .OPMODEREG(1), // Enable=1/disable=0 OPMODE input pipeline register
    .PREG(1), // Enable=1/disable=0 P output pipeline register
    .RSTTYPE("SYNC") // Specify reset type, "SYNC" or "ASYN"
) DSP48A_inst (
    .BCOUT(BCOUT), // 18-bit B port cascade output
    .CARRYOUT(CARRYOUT), // 1-bit carry output
    .P(P), // 48-bit output
    .PCOUT(PCOUT), // 48-bit cascade output
    .A(A), // 18-bit A data input
    .B(B), // 18-bit B data input (can be connected to fabric or BCOUT of adjacent DSP48A)
    .C(C), // 48-bit C data input
    .CARRYIN(CARRYIN), // 1-bit carry input signal
    .CEA(CEA), // 1-bit active high clock enable input for A input registers
    .CEB(CEB), // 1-bit active high clock enable input for B input registers
    .CEC(CEC), // 1-bit active high clock enable input for C input registers
    .CECARRYIN(CECARRYIN), // 1-bit active high clock enable input for CARRYIN registers
    .CED(CED), // 1-bit active high clock enable input for D input registers
    .CEM(CEM), // 1-bit active high clock enable input for multiplier registers
    .CEOPMODE(CEOPMODE), // 1-bit active high clock enable input for OPMODE registers
    .CEP(CEP), // 1-bit active high clock enable input for P output registers
    .CLK(CLK), // Clock input
    .D(D), // 18-bit B pre-adder data input
    .OPMODE(OPMODE), // 8-bit operation mode input
    .PCIN(PCIN), // 48-bit P cascade input
    .RSTA(RSTA), // 1-bit reset input for A input pipeline registers
    .RSTB(RSTB), // 1-bit reset input for B input pipeline registers
    .RSTC(RSTC), // 1-bit reset input for C input pipeline registers
    .RSTCARRYIN(RSTCARRYIN), // 1-bit reset input for CARRYIN input pipeline registers
    .RSTD(RSTD), // 1-bit reset input for D input pipeline registers
    .RSTM(RSTM), // 1-bit reset input for M pipeline registers
    .RSTOPMODE(RSTOPMODE), // 1-bit reset input for OPMODE input pipeline registers
    .RSTP(RSTP) // 1-bit reset input for P output pipeline registers
);

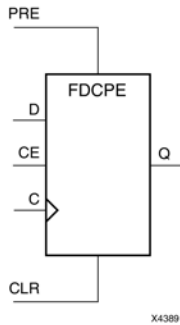
// End of DSP48A_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

FDCPE

プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset and Clear



概要

このデザイン エLEMENTは、データ (D)、クロック イネーブル (CE)、非同期プリセット (PRE)、非同期クリア (CLR) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。非同期 PRE が High になると Q 出力が High にセットされ、CLR が High になると出力が Low にリセットされます (CLR 入力が PRE 入力よりも優先される)。PRE と CLR が Low で CE が High の場合、クロック (C) が Low から High に切り替わるときに D 入力の値がフリップフロップにロードされます。CE が Low の場合クロック遷移は無視され、以前の値が保持されます。FDCPE は通常、スライスまたは IOB レジスタとしてインプリメントされます。

FPGA では電源が投入されると、INIT 属性を使用して指定した初期値に設定されます。GSR (グローバル セット/リセット) をアサートすると、INIT で指定した初期値に非同期で設定されます。

メモ : 非同期のセットおよびリセットの使用がサポートされていますが、これらの使用は通常お勧めしません。非同期信号を使用するとタイミングの問題が検出および制御しにくく、またロジックの最適化に悪影響を及ぼし、同期セットまたはリセットを使用した場合に比べて消費電力が大きくなる場合があります。

論理表

入力					出力
CLR	PRE	CE	D	C	Q
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	変化なし
0	0	1	D	↑	D

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力
C	入力	1	クロック入力
CE	入力	1	クロック イネーブル入力
CLR	入力	1	非同期クリア入力
D	入力	1	データ入力
PRE	入力	1	非同期セット入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後および GSR 入力時の Q 出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

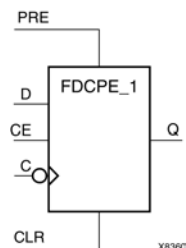
// End of FDCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

FDCPE_1

プリミティブ : D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Preset and Clear



概要

FDCPE_1 は、データ (D)、クロック イネーブル (CE)、非同期プリセット (PRE)、非同期クリア (CLR) の各入力とデータ出力 (Q) がある単一の D フリップフロップです。非同期 PRE が High になると、Q 出力が High にセットされます。CLR が High になると、出力が Low にリセットされます。PRE と CLR が Low で CE が High の場合、クロック (C) が High から Low に切り替わるたびに D 入力の値がフリップフロップにロードされます。CE が Low の場合、クロック遷移は無視されます。

電力を供給すると、このフリップフロップは非同期にクリアされ、出力が Low になります。FPGA では、グローバル セット / リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP_architecture シンボルの GSR 入力の前にインバータを追加するとアクティブ Low にできます。

論理表

入力					出力
CLR	PRE	CE	D	C	Q
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	変化なし
0	0	1	D	↓	D

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力
C	入力	1	クロック入力
CE	入力	1	クロック イネーブル入力
CLR	入力	1	非同期クリア入力
D	入力	1	データ入力
PRE	入力	1	非同期セット入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	2 進数	0, 1	0	コンフィギュレーション後の Q 出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

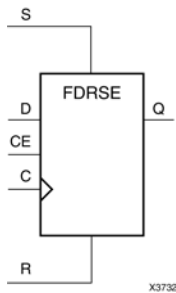
// End of FDCPE_1_inst instantiation
```


詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

FDRSE

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



概要

FDRSE は、同期リセット (R)、同期セット (S)、クロック イネーブル (CE) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期リセット入力 (R) が High になると、ほかの入力は無視され、クロック (C) が Low から High に切り替わる時に出力 (Q) が Low にリセットされます (リセットがセットよりも優先される)。セット入力 (S) が High、R が Low の場合、クロック (C) が Low から High に切り替わる時にフリップフロップがセットされ、出力が High になります。R と S が Low、CE が High の場合、クロックが Low から High に切り替わる時に D 入力の値がフリップフロップにロードされます。

電源が投入されると、INIT 属性を使用して指定した初期値に設定されます。GSR (グローバル セット/リセット) をアサートすると、INIT で指定した初期値に非同期で設定されます。

論理表

入力					出力
R	S	CE	D	C	Q
1	X	X	X	↑	0
0	1	X	X	↑	1
0	0	0	X	X	変化なし
0	0	1	1	↑	1
0	0	1	0	↑	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後および GSR 入力時の Q 出力の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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/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 記述 (インスタンスレーション)

```
// FDRSE: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//       Clock Enable (posedge clk).
//       Virtex-4/5, Spartan-3/3E/3A/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
);

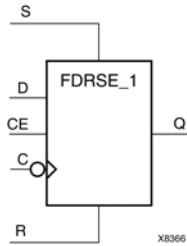
// End of FDRSE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

FDRSE_1

プリミティブ : D Flip-Flop with Negative-Clock Edge, Synchronous Reset and Set, and Clock Enable



概要

FDRSE_1 は、同期リセット (R)、同期セット (S)、クロック イネーブル (CE) の各入力とデータ出力 (Q) がある単一の D タイプ フリップフロップです。同期リセット入力 (R) が High になると、ほかの入力は無視され、クロック (C) が High から Low に切り替わるたびに、出力 (Q) が Low にリセットされます (リセットがセットよりも優先される)。S が High、R が Low の場合、クロック (C) が High から Low に切り替わるたびにフリップフロップがセットされ、出力が High になります。R と S が Low で CE が High の場合、クロックが High から Low に切り替わるたびに D 入力の値がフリップフロップにロードされます。

電力を供給すると、このフリップフロップは非同期にクリアされ、出力が Low になります。FPGA では、グローバル セット / リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP_architecture シンボルの GSR 入力の前にインバータを追加するとアクティブ Low にできます。

論理表

入力					出力
R	S	CE	D	C	Q
1	X	X	X	↓	0
0	1	X	X	↓	1
0	0	0	X	X	変化なし
0	0	1	D	↓	D

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後および GSR 入力時の Q 出力の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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-3/3E/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 記述 (インスタンスレーション)

```
// FDRSE_1: Single Data Rate D Flip-Flop with Synchronous Clear, Set and
//          Clock Enable (negedge clock).
//          Virtex-4/5, Spartan-3/3E/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
);

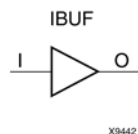
// End of FDRSE_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

IBUF

プリミティブ : Input Buffer



概要

このデザイン エレメントは、最上位の入力ポートまたは入出力ポートに接続されている信号に自動的に挿入されます。このバッファは通常、合成ツールで推論しますが、必要に応じてインスタンスエートすることも可能です。インスタンスエートするには、入力ポート (I) を関連する最上位の入力ポートまたは入出力ポートに接続し、出力ポート (O) をそのポートをソースとする FPGA ロジックに接続します。必要なジェネリック マップ (VHDL) またはパラメータ値代入 (Verilog) に変更を加えて、コンポーネントのデフォルトのビヘイビアを変更します。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
I	入力	1	バッファの入力

デザインの入力方法

インスタンスエーション	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

このエレメントは通常、デザインの最上位入力ポートに対して推論されます。通常はソースコードで指定する必要はありませんが、必要に応じてインスタンスエートできます。このコンポーネントをインスタンスエートするには、該当するライブラリ ガイドに含まれるインスタンスエーション コードを最上位エンティティ/モジュールに挿入します。デザイン階層を保つために、すべての I/O コンポーネントを必ずデザインの最上位に配置してください。I ポートをデザインの最上位入力ポートに、O ポートをこの入力供給されるロジックに直接接続します。generic/default 値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

属性	タイプ	値	デフォルト	説明
IBUF_DELAY_VALUE	文字列	0 ~ 16	0	IOB 外のレジスタを介さないパスに追加する遅延を指定
IFD_DELAY_VALUE	文字列	AUTO、0 ~ 8	AUTO	IOB 内のレジスタを介するパスに追加する遅延を指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
```

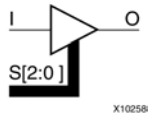
詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

IBUF_DLY_ADJ

プリミティブ : Dynamically Adjustable Input Delay Buffer

IBUF_DLY_AD J



概要

このデザイン エLEMENTは調整可能な遅延ELEMENTを含む入力バッファで、FPGA への入力信号の遅延を動的に変更できるようにします。この機能は、FPGA への高速入力データをプロセス、電圧、温度の変化に対して調整する場合に特に便利です。このコンポーネントには 3 ビットのセレクト バスがあり、入力信号に 8 個の遅延値を追加できます。また、遅延オフセットを設定することも可能で、遅延調整値を連続する 16 個の遅延値の上位 8 個にするか下位 8 個にするかを指定できます。

遅延量およびこのコンポーネントの詳細な使用法は、「詳細情報」を参照してください。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの遅延された出力
I	入力	1	差動入力データ (正)
IB	入力	1	差動入力データ (負)
S	入力	3	ダイナミック遅延調整のセレクト ライン

デザインの入力方法

インスタンスエーション	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DELAY_OFFSET	文字列	OFF、ON	OFF	IBUFDS_DLY_ADJ を OFF に設定すると、下位の遅延値が使用されます。この設定は、必要な追加遅延が小さい場合に使用します。ON に設定すると、上位の (大きい) 遅延値が使用されます。この設定は、必要な追加遅延が大きい場合に使用します。
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- IBUF_DLY_ADJ: Single-ended Input Buffer
--           Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

IBUF_DLY_ADJ_inst : IBUF_DLY_ADJ
generic map (
    DELAY_OFFSET => "OFF", -- Enable initial Delay Offset, "OFF" or "ON"
    IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
    O => O, -- Buffer output
    I => I, -- Buffer input (connect directly to top-level port)
    S => S -- 3-bit buffer delay select input
);

-- End of IBUF_DLY_ADJ_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IBUF_DLY_ADJ: Dynamically Adjustable Delay, Single-ended Input Buffer
//           Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

IBUF_DLY_ADJ #(
    .DELAY_OFFSET("OFF"), // Enable Initial Delay Offset, "OFF" or "ON"
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
)IBUF_DLY_ADJ_inst (
    .O(O), // Buffer output
    .I(I), // Buffer input (connect directly to top-level port)
    .S(S) // 3-bit buffer delay select input
);

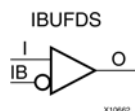
// End of IBUF_DLY_ADJ_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

IBUFDS

プリミティブ : Differential Signaling Input Buffer



概要

このデザイン エレメントは、低電圧差動信号を使用する入力バッファです。IBUFDS では、デザイン レベルのインターフェイス信号は、一方がマスタで、もう一方がスレーブとなる 2 つの異なるポート (I、IB) で表されます。マスタとスレーブは MYNET_P と MYNET_N のように、同じ論理信号の反対の状態を示します。オプションで、プログラム可能な差動終端機能を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を減らすことができます。

論理表

入力		出力
I	IB	O
0	0	変化なし
0	1	0
1	0	1
1	1	変化なし

ポートの説明

ポート名	タイプ	幅	機能
I	入力	1	Diff_p バッファの入力
IB	入力	1	Diff_p バッファの入力
O	出力	1	バッファの出力

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デザイン階層を保つために、すべての I/O コンポーネントを必ずデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスタとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O ポートをこの入力が供給されるロジックに接続します。generic/defparam 値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DIFF_TERM	ブール 代数	TRUE、FALSE	FALSE	ビルトインの差動終端抵抗をイネーブル
IBUF_DELAY_ VALUE	文字列	0 ~ 16	0	IOB 外のレジスタを介さないパスに追加する遅延を指定
IFD_DELAY_ VALUE	文字列	AUTO、 0 ~ 8	AUTO	IOB 内のレジスタを介するパスに追加する遅延を指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- IBUFDS: Differential Input Buffer
--      Spartan-3/3E/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 記述 (インスタンスレーション)

```
// IBUFDS: Differential Input Buffer
//      Virtex-4/5, Spartan-3/3E/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)
);

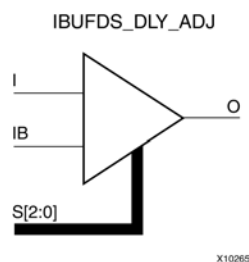
// End of IBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

IBUFDS_DLY_ADJ

プリミティブ : Dynamically Adjustable Differential Input Delay Buffer



概要

このデザイン エLEMENTは調整可能な遅延ELEMENTを含む差動入力バッファで、FPGA への入力信号の遅延を動的に変更できるようにします。この機能は、FPGA への高速入力データをプロセス、電圧、温度の変化に対して調整する場合に特に便利です。このコンポーネントには 3 ビットのセレクト バスがあり、入力信号に 8 個の遅延値を追加できます。また、遅延オフセットを設定することも可能で、遅延調整値を連続する 16 個の遅延値の上位 8 個にするか下位 8 個にするかを指定できます。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの遅延された出力
I	入力	1	差動入力データ (正)
IB	入力	1	差動入力データ (負)
S	入力	3	ダイナミック遅延調整のセレクト ライン

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DELAY_OFFSET	文字列	OFF、ON	OFF	OFF に設定すると、下位の遅延値が使用されます。この設定は、必要な追加遅延が小さい場合に使用します。ON に設定すると、上位の (大きい) 遅延値が使用されます。この設定は、必要な追加遅延が大きい場合に使用します。
DIFF_TERM	ブール代数	TRUE、FALSE	FALSE	内部差動終端をイネーブルにするかディスエーブルにするかを指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- IBUFDS_DLY_ADJ: Differential Input Buffer
--           Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

IBUFDS_DLY_ADJ_inst : IBUFDS_DLY_ADJ
generic map (
    DIFF_TERM => FALSE, -- Differential Termination
    DELAY_OFFSET => "OFF", -- Enable initial Delay Offset, "OFF" or "ON"
    IOSTANDARD => "DEFAULT") -- Specify the input I/O standard
port map (
    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)
    S => S, -- 3-bit buffer delay select input
);

-- End of IBUFDS_DLY_ADJ_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IBUFDS_DLY_ADJ: Dynamically Adjustable Delay, Differential Input Buffer
//           Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

IBUFDS_DLY_ADJ #(
    .DELAY_OFFSET("OFF"), // Enable Initial Delay Offset, "OFF" or "ON"
    .DIFF_TERM("FALSE"), // Differential Termination
    .IOSTANDARD("DEFAULT") // Specify the input I/O standard
) IBUFDS_DLY_ADJ_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)
    .S(S) // 3-bit buffer delay select input
);

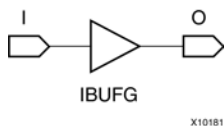
// End of IBUFDS_DLY_ADJ_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

IBUFG

プリミティブ : Dedicated Input Clock Buffer



概要

IBUFG は、FPGA への入力クロックをグローバル クロック配線リソースに接続するために使用する専用入力です。DCM_SP および BUFG への専用接続となり、デバイスのクロック遅延とジッタが最小限に抑えられます。IBUFG の入力は、グローバル クロック ピンでのみ駆動できます。IBUFG の出力は、DCM_SP、BUFG、または指定したロジックの CLKIN を駆動できます。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	クロック バッファ出力
I	入力	1	クロック バッファ入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
IBUF_DELAY_VALUE	文字列	0 ~ 16	0	IOB からのレジスタを介さないパスに追加する遅延を指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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)
);

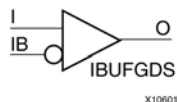
// End of IBUFG_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

IBUFGDS

プリミティブ : Differential Signaling Dedicated Input Clock Buffer and Optional Delay



概要

このデザイン エLEMENT は、クロック バッファ (BUFG) または DCM に接続するための専用の差動信号入力バッファです。IBUFGDS では、デザイン レベルのインターフェイス信号は、一方が「マスタ」で、もう一方が「スレーブ」となる 2 つの異なるポート (I、IB) で表されます。マスタとスレーブは MYNET_P と MYNET_N のように、同じ論理信号の反対の状態を示します。オプションで、プログラム可能な差動終端機能を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を減らすことができます。デバイスへの入力データの取り込みには、プログラマブル遅延を使用することもできます。

論理表

入力		出力
I	IB	O
0	0	変化なし
0	1	0
1	0	1
1	1	変化なし

ポートの説明

ポート名	方向	幅	機能
O	出力	1	クロック バッファ出力
IB	入力	1	Diff_n クロック バッファの入力
I	入力	1	Diff_p クロック バッファの入力

デザインの入力方法

インスタンスエーション	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デザイン階層を保つために、すべての I/O コンポーネントを必ずデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスタとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O ポートをこの入力をソースとする DCM、BUFG、またはロジックに接続してください。一部の合成ツールでは、IBUFG を FPGA のクロックリソースに接続すると、必要に応じて BUFG が自動的に推論されます。generic/defparam 値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DIFF_TERM	ブール 代数	TRUE、FALSE	FALSE	ビルトインの差動終端抵抗をイネーブル
IBUF_DELAY_ VALUE	文字列	0 ~ 16	0	IOB からのレジスタを介さないパスに追加する遅延を指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- IBUFGDS: Differential Global Clock Buffer (sourced by an external pin)
--          Virtex-4/5, Spartan-3/3E/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 記述 (インスタンス化)

```
// IBUFGDS: Differential Global Clock Buffer (sourced by an external pin)
//          Virtex-4/5, Spartan-3/3E/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
);

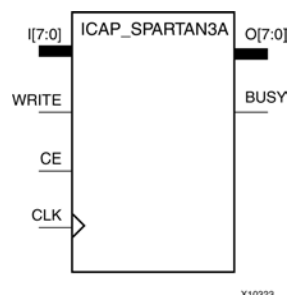
// End of IBUFGDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

ICAP_SPARTAN3A

プリミティブ : Internal Configuration Access Port



概要

このデザイン エLEMENTからは、FPGA ファブリックから FPGA のコンフィギュレーション機能にアクセスできます。このコンポーネントの主な用途は、FPGA デバイスでのマルチブート制御です。このコンポーネントを使用すると、FPGA アレイのコンフィギュレーション ロジックにコマンドおよびデータを書き込んだり、コンフィギュレーション ロジックからデータを読み出したりすることができます。このファンクションを正しく使用しないと FPGA の機能および信頼性に悪影響を与えるため、機能を完全に理解した上で使用してください。

ポートの説明

ポート名	方向	幅	機能
O	出力	8	コンフィギュレーション データ出力バス
Busy	出力	8	Busy 出力
I	入力	8	コンフィギュレーション データ入力バス
WRITE	入力	8	アクティブ Low のライト入力
CE	入力	8	アクティブ Low のイネーブル入力
CLK	入力	8	クロック入力

デザインの入力方法

インスタンシエーション	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- ICAP_SPARTAN3A: Internal Configuration Access Port
--           Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

ICAP_SPARTAN3A_inst : ICAP_SPARTAN3A
port map (
    BUSY => BUSY,    -- Busy output
    O => O,          -- 8-bit data output
    CE => CE,        -- Clock enable input
    CLK => CLK,      -- Clock input
    I => I,          -- 8-bit data input
    WRITE => WRITE   -- Write input
);

-- End of ICAP_SPARTAN3A_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ICAP_SPARTAN3A: Internal Configuration Access Port
//           Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

ICAP_SPARTAN3A ICAP_SPARTAN3A_inst (
    .BUSY(BUSY),    // Busy output
    .O(O),          // 8-bit data output
    .CE(CE),        // Clock enable input
    .CLK(CLK),      // Clock input
    .I(I),          // 8-bit data input
    .WRITE(WRITE)   // Write input
);

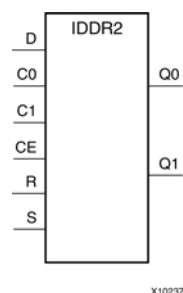
// End of ICAP_SPARTAN3A_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

IDDR2

プリミティブ : Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset



概要

このデザイン エLEMENTは、ザイリンクス FPGA で外部デュアル データレート (DDR) 信号を受信するための専用入力レジスタです。C0 と C1 の 2 つのクロックを使用してコンポーネントに接続されるので、C0 および C1 の両方の立ち上がりエッジでデータが取り込まれます。IDDR2 は、レジスタの動作を停止するために使用できるアクティブ High のクロック イネーブル (CE) ポート、対応するクロックに同期または非同期になるよう設定できるセット/リセット ポートを備えています。また、オプションの調整機能を使用すると、コンポーネントへの両方の出力データ ポートを 1 つのクロックに揃えることができます。

論理表

入力						出力	
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	変化なし	変化なし
0	0	1	D	↑	X	D	変化なし
0	0	1	D	X	↑	変化なし	D

セット/リセットは SRTYPE 値で同期に設定可能

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

デフォルトの動作を変更するには、ジェネリック マップ (VHDL) またはパラメータ値代入 (Verilog) を使用して、インスタンス化されたコンポーネントの一部として属性を変更します。IDDR2 は、入力バッファが推論されるデザインの最上位入力ポートに接続するか、インスタンス化された IBUF、IOBUF、IBUFDS、または IOBUFDS のいずれかに直接接続できます。このコンポーネントのすべての入力と出力は、接続しておく必要があります。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DDR_ALIGNMENT	文字列	NONE、C0、C1	NONE	DDR レジスタの出力アライメントを設定 <ul style="list-style-type: none"> ・ NONE : 対応する C0 または C1 の立ち上がりエッジのすぐ後で、Q0 および Q1 にデータが出力されます。 ・ C0 : Q0 と Q1 両方のデータが C0 クロックの立ち上がりエッジに同期します。 ・ C1 : Q0 と Q1 両方のデータが C1 クロックの立ち上がりエッジに同期します。
INIT_Q0	整数	0、1	0	Q0 出力の初期値を 0 または 1 に設定
INIT_Q1	整数	0、1	0	Q1 出力の初期値を 0 または 1 に設定
SRTYPE	文字列	SYNC、ASYNC	SYNC	セット/リセットを SYNC または ASYNC に設定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```

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 記述 (インスタンス化)

```
// 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 "ASYN" 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
);

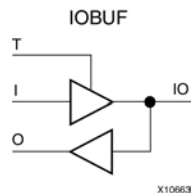
// End of IDDR2_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

IOBUF

プリミティブ : Bi-Directional Buffer



概要

このデザイン エレメントは双方向でシングルエンドの I/O バッファで、内部ロジックを外部双方向ピンに接続する場合に使用します。

論理表

入力		双方向	出力
T	I	I/O	O
1	X	Z	X
0	1	1	1
0	0	0	0

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
I/O	入出力	1	バッファの入出力
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTL、LVCMOS12、LVCMOS15、LVCMOS18、LVCMOS25 または LVCMOS33 を使用する SelectIO™ バッファの出力の駆動電流 (mA) を選択
IBUF_DELAY_VALUE	文字列	0 ~ 16	0	IOB からのレジスタを介さないパスに追加する遅延を指定
IFD_DELAY_VALUE	文字列	AUTO、0 ~ 8	AUTO	IOB 内のレジスタを介するパスに追加する遅延を指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て
SLEW	文字列	SLOW、FAST、QUIETIO	SLOW	出力の立ち上がり時間と立ち下がり時間を設定 この属性の最適な設定方法は、データシートを参照してください。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```

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 記述 (インスタンス化)

```

// 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
);

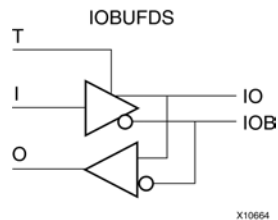
// End of IOBUF_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

IOBUFDS

プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable



概要

このデザイン エLEMENTは、低電圧差動信号を使用する双方向バッファです。IOBUFDS では、デザイン レベルのインターフェイス信号は、一方が「マスタ」で、もう一方が「スレーブ」となる 2 つの異なるポート (IO、IOB) で表されます。マスタとスレーブは MYNET_P と MYNET_N のように、同じ論理信号の反対の状態を示します。オプションで、プログラム可能な差動終端機能を使用すると、シグナル インテグリティが向上し、外部コンポーネントの数を減らすことができます。デバイスへの入力データの取り込みには、プログラマブル遅延を使用することもできます。

論理表

入力		双方向		出力
I	T	I/O	IOB	O
X	1	Z	Z	変化なし
0	0	0	1	0
1	0	1	0	1

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
I/O	入出力	1	Diff_p 入出力
IOB	入出力	1	Diff_n 入出力
I	入力	1	バッファの入力
T	入力	1	トリステート イネーブル入力

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
IBUF_DELAY_VALUE	文字列	0 ~ 16	0	IOB からのレジスタを介さないパスに追加する遅延を指定
IFD_DELAY_VALUE	文字列	AUTO、0 ~ 8	AUTO	IOB 内のレジスタを介するパスに追加する遅延を指定
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

KEEPER

プリミティブ : KEEPER Symbol



概要

このデザイン エLEMENTは、双方向出力ピンに接続されるネットの値を保持するウィークキーパ エLEMENTです。たとえば、ネットに対して論理値 1 を駆動すると、KEEPER はそのネットにウィーク/抵抗値 1 を駆動します。その後、ネットドライバがトライステートになっても、KEEPER はウィーク/抵抗値 1 を駆動し続けます。

ポートの説明

属性	方向	幅	機能
O	出力	1 ビット	キーパ出力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

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

KEEPER KEEPER_inst (
    .O(0)      // Keeper output (connect directly to top-level port)
);

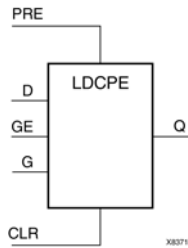
// End of KEEPER_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LDCPE

プリミティブ : Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable



概要

このデザイン エLEMENTは、データ (D)、非同期クリア (CLR)、非同期プリセット (PRE)、ゲート イネーブル (GE) がある透過データ ラッチです。CLR が High になると、ほかの入力は無視され、データ出力 (Q) が Low にリセットされます。PRE が High、CLR が Low の場合、データ出力 (Q) は High にプリセットされます。ゲート入力 (G) と GE が High で CLR と PRE が Low の場合、Q にはデータ入力 (D) が使用されます。D 入力の値は、G が High から Low に切り替わるときにラッチ内に格納されます。Q 出力の値は、G または GE が Low の間は変化しません。

電力を供給すると、ラッチは非同期にクリアされ、出力が Low になります。FPGA では、グローバル セット/リセット (GSR) をアクティブにすると、電源投入時の状態をシミュレーションできます。GSR のデフォルトはアクティブ High ですが、STARTUP_architecture シンボルの GSR 入力の前にインバータを追加するとアクティブ Low にできます。

論理表

入力					出力
CLR	PRE	GE	G	D	Q
1	X	X	X	X	0
0	1	X	X	X	1
0	0	0	X	X	変化なし
0	0	1	1	0	0
0	0	1	1	1	1
0	0	1	0	X	変化なし
0	0	1	↓	D	D

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	データ出力
CLR	入力	1	非同期クリア/リセット入力
D	入力	1	データ入力
G	入力	1	ゲート入力
GE	入力	1	ゲート イネーブル入力
PRE	入力	1	非同期プリセット/セット入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	整数	0, 1	0	コンフィギュレーション後の Q 出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- LDCPE: Transparent latch with Asynchronous Reset, Preset and
--       Gate Enable.
--       Virtex-4/5, Spartan-3/3E/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 記述 (インスタンス化)

```
// LDCPE: Transparent latch with Asynchronous Reset, Preset and
//       Gate Enable.
//       Virtex-4/5, Spartan-3/3E/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
);

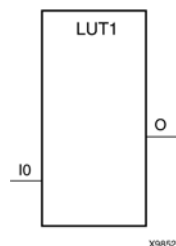
// End of LDCPE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

LUT1

プリミティブ : 1-Bit Look-Up Table with General Output



概要

このデザイン エレメントは一般出力 (O) を持つ 1 ビットのルックアップ テーブル (LUT) です。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このエレメントは、バッファまたはインバータの機能を果たします。これらのエレメントは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力	出力
I0	O
0	INIT[0]
1	INIT[1]
INIT = INIT 属性に割り当てられた 2 進数値	

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	2 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

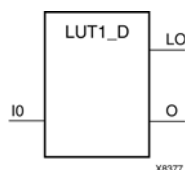
// End of LUT1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT1_D

プリミティブ : 1-Bit Look-Up Table with Dual Output



概要

このデザイン エLEMENTは 1 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。このELEMENTはバッファまたはインバータの機能を果たします。

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力	出力	
I0	O	LO
0	INIT[0]	INIT[0]
1	INIT[1]	INIT[1]
INIT = INIT 属性に割り当てられた 2 進数値		

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	2 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

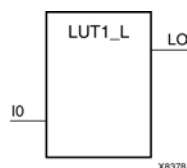
// End of LUT1_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT1_L

プリミティブ : 1-Bit Look-Up Table with Local Output



概要

このデザイン エLEMENT は、1 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバータの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力	出力
I0	LO
0	INIT[0]
1	INIT[1]
INIT = INIT 属性に割り当てられた 2 進数値	

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	2 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO  -- LUT input
);

-- End of LUT1_L_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// 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
    .IO(IO)  // LUT input
);

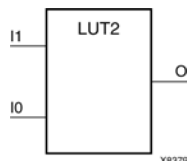
// End of LUT1_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT2

プリミティブ : 2-Bit Look-Up Table with General Output



概要

このデザイン エLEMENTは一般出力 (O) を持つ 2 ビットのルックアップ テーブル (LUT) です。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このELEMENTは、バッファまたはインバータの機能を果たします。これらのELEMENTは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力		出力
I1	I0	O
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値		

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	4 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスエーション)

```
// 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
);

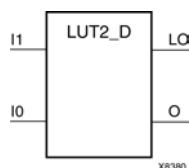
// End of LUT2_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT2_D

プリミティブ : 2-Bit Look-Up Table with Dual Output



概要

このデザイン エLEMENTは 2 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力		出力	
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	4 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスエーション)

```
// 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
);

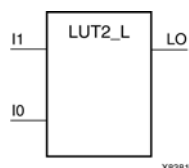
// End of LUT2_D_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

LUT2_L

プリミティブ : 2-Bit Look-Up Table with Local Output



概要

このデザイン エLEMENT は、2 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバータの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力		出力
I1	I0	LO
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値		

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	4 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスエーション)

```
// 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
);

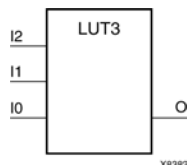
// End of LUT2_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

LUT3

プリミティブ : 3-Bit Look-Up Table with General Output



概要

このデザイン エLEMENTは一般出力 (O) を持つ 3 ビットのルックアップ テーブル (LUT) です。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このELEMENTは、バッファまたはインバータの機能を果たします。これらのELEMENTは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力			出力
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	8 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

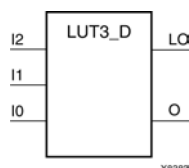
// End of LUT3_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT3_D

プリミティブ : 3-Bit Look-Up Table with Dual Output



概要

このデザイン エLEMENTは 3 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力			出力	
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値				

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	8 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

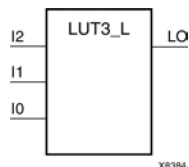
// End of LUT3_D_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

LUT3_L

プリミティブ : 3-Bit Look-Up Table with Local Output



概要

このデザイン エLEMENT は、3 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバータの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力			出力
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	8 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

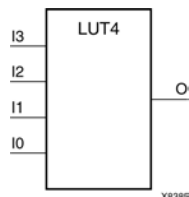
// End of LUT3_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT4

プリミティブ : 4-Bit Look-Up-Table with General Output



概要

このデザイン エLEMENTは一般出力 (O) を持つ 4 ビットのルックアップ テーブル (LUT) です。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。このELEMENTは、バッファまたはインバータの機能を果たします。これらのELEMENTは基本ブロックで、各 CLB スライスに 2 つ、各 CLB に 4 つずつあります。LUT には複数のバリエーションがあり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力				出力
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値				

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

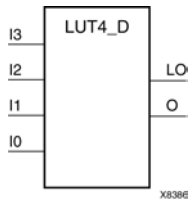
// End of LUT4_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT4_D

プリミティブ : 4-Bit Look-Up Table with Dual Output



概要

このデザイン エLEMENTは 4 ビットのルックアップ テーブル (LUT) で、同じ機能を持つ O および LO という 2 つの出力があります。

出力 O は一般的なインターコネクトです。LO 出力は同じ CLB スライス内の別の出力、または高速バッファに接続します。LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力				出力	
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

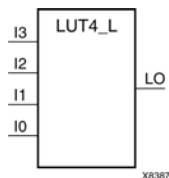
// End of LUT4_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

LUT4_L

プリミティブ : 4-Bit Look-Up Table with Local Output



概要

このデザイン エLEMENT は、4 ビットのルックアップ テーブル (LUT) で、同じ CLB スライス内にある別の出力および高速バッファへの接続に使用するローカル出力 (LO) があります。このELEMENTはバッファまたはインバータの機能を果たします。

LUT のファンクションを設定するため、INIT 属性を使用して、各入力値に対する出力値を 16 進数で指定する必要があります。

FPGA LUT プリミティブでは、INIT パラメータで論理値が設定されます。デフォルトは 0 で、入力値にかかわらず出力を 0 に駆動します (グラウンドとして機能)。ただし多くの場合、LUT プリミティブのロジック ファンクションを指定するために、新しい INIT の値を決定する必要があります。LUT の値を指定する方法には、次の 2 つがあります。

論理表を使用する方法 : LUT の INIT 値を決定する一般的な方法。バイナリの論理表にすべての入力をリストして出力のロジック値を指定し、これらの出力値から、初期値を作成します。

論理式を使用する方法 : リストされた真理値表の値に対応する LUT の各入力にパラメータを定義し、パラメータを元にロジックの論理式を生成します。概念を理解してしまえばこの方法の方が簡単で、前出の方法のようにパラメータの指定にコードを使用する必要がありません。

論理表

入力				出力
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 = INIT 属性で指定された 16 進数値を 2 進数で表した値

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	ルックアップ テーブルの初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

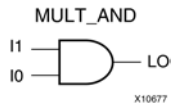
// End of LUT4_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MULT_AND

プリミティブ : Fast Multiplier AND



概要

このデザイン エLEMENTはスライス内にある AND コンポーネントです。このスライスでは 2 つの入力が 4 入力 LUT と共有され、出力がキャリー ロジックに駆動しています。この追加のロジックはその他の目的にも使用できますが、高速で小型の乗算器の作成に特に有用です。I1 および I0 入力は、対応する LUT の I1 および I0 入力に接続する必要があります。LO 出力は、対応する MUXCY、MUXCY_D、または MUXCY_L の DI 入力に接続する必要があります。

論理表

入力		出力
I1	I0	LO
0	0	0
0	1	0
1	0	0
1	1	1

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

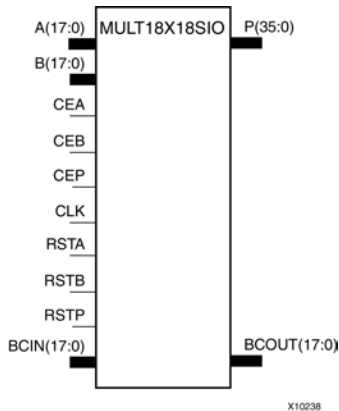
// End of MULT_AND_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MULT18X18SIO

プリミティブ : 18 x 18 Cascadable Signed Multiplier with Optional Input and Output Registers, Clock Enable, and Synchronous Reset



概要

このデザイン エLEMENTは、36 ビット出力、18 X 18 ビット入力の専用の符号付き乗算器です。AREG、BREG、PREG 属性をすべて 0 に設定すると、非同期の乗算が実行されます。逆に属性をすべて 1 に設定すると、異なるレイテンシとパフォーマンス特性で同期の乗算が実行されます。同期乗算器を使用する場合、乗算器のレジスタ バンクの各セットに対してアクティブ High のクロック イネーブル (CEA、CEB、CEP) と同期リセット (RSTA、RSTB、RSTP) が使用されます。BCIN ポートと BCOUT ポートに B_INPUT 属性を使用して MULT18X18SIO をカスケード接続すると、より大型の乗算ファンクションを作成できます。

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
AREG	整数	0、1	1	A ポートで入力レジスタを使用するかどうかを指定します。0 の場合はレジスタが使用されず、1 の場合はレジスタが使用されます。
BREG	整数	0、1	1	B ポートで入力レジスタを使用するかどうかを指定します。0 の場合はレジスタが使用されず、1 の場合はレジスタが使用されます。
B_INPUT	文字列	DIRECT、CASCADE	DIRECT	B ポートが FPGA に接続されている (DIRECT) か、別の MULT18X18SIO の BCOUT ポートに接続されているかを指定します。
PREG	整数	0、1	1	乗算器の出力レジスタを使用するかどうかを指定します。0 の場合はレジスタが使用されず、1 の場合はレジスタが使用されます。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

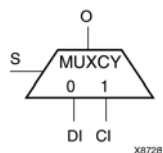
// End of MULT18X18SIO_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXCY

プリミティブ : 2-to-1 Multiplexer for Carry Logic with General Output



概要

スライスの直接入力 (DI) は、MUXCY の DI 入力に接続します。LC のキャリー入力 (CI) は、MUXCY の CI 入力に接続します。セレクト入力 (S) は、ルックアップ テーブル (LUT) の出力で駆動し、MUX ファンクションとしてコンフィギュレーションします。キャリー出力 (O) には選択された入力の値が出力され、各 LC のキャリー出力ファンクションをインプリメントします。S が Low の場合は DI が選択され、High の場合は CI が選択されます。

このほか、ローカル出力を持つ MUXCY_D および MUXCY_L があり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

論理表

入力			出力
S	DI	CI	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

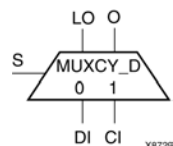
// End of MUXCY_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXCY_D

プリミティブ : 2-to-1 Multiplexer for Carry Logic with Dual Output



概要

このデザイン エLEMENTは、1 ビットの高速キャリー伝搬ファンクションをインプリメントするために使用します。このようなファンクションは、1 つのロジック セル (LC) に 1 つずつインプリメントできるので、1 つの CLB に合計 4 ビットをインプリメントできます。LC の直接入力 (DI) は MUXCY_D の DI 入力に接続し、LC のキャリー入力 (CI) は MUXCY_D の CI 入力に接続します。セレクト入力 (S) は、ルックアップ テーブル (LUT) の出力で駆動し、XOR ファンクションとしてコンフィギュレーションします。キャリー出力 (O と LO) には選択された入力の値が出力され、各 LC のキャリー出力ファンクションをインプリメントします。S が Low の場合は DI が選択され、High の場合は CI が選択されます。

出力 O と LO は、機能的に同じです。出力 O は一般的なインターコネクトです。「MUXCY」および「MUXCY_L」も参照してください。

論理表

入力			出力	
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

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

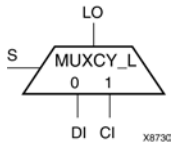
// End of MUXCY_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXCY_L

プリミティブ : 2-to-1 Multiplexer for Carry Logic with Local Output



概要

このデザイン エLEMENTは、1 ビットの高速キャリー伝搬ファンクションをインプリメントするために使用します。このようなファンクションは、1 つのロジック セル (LC) に 1 つずつインプリメントできるので、1 つの CLB に合計 4 ビットをインプリメントできます。LC の直接入力 (DI) は MUXCY_L の DI 入力に接続し、LC のキャリー入力 (CI) は MUXCY_L の CI 入力に接続します。セレクト入力 (S) は、ルックアップ テーブル (LUT) の出力で駆動し、XOR ファンクションとしてコンフィギュレーションします。キャリー出力 (LO) には選択された入力の値が出力され、各 LC のキャリー出力ファンクションをインプリメントします。S が Low の場合は DI が選択され、High の場合は CI が選択されます。

「MUXCY」および「MUXCY_D」も参照してください。

論理表

入力			出力
S	DI	CI	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

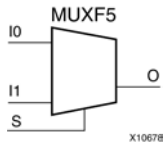
// End of MUXCY_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXF5

プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、ルックアップ テーブルと組み合わせて、5 ファンクションのルックアップ テーブルまたは 4:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、CLB スライス を 1 つ使用してインプリメントします。I0 および I1 入力には、2 つのルックアップ テーブルのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

このほか、ローカル出力を持つ MUXF5_D および MUXF5_L があり、異なるタイミング モデルでレイアウト前のタイミング 予測をより正確に行う必要がある場合に使用できます。

論理表

入力			出力
S	I0	I1	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// MUXF5: Slice MUX to tie two LUT4's together with general output
//       For use with Virtex-4, Spartan-3/3E/3A/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
);

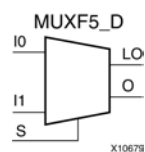
// End of MUXF5_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXF5_D

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、ルックアップ テーブルと組み合わせて、5 ファンクションのルックアップ テーブルまたは 4:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、CLB スライス を 1 つ使用してインプリメントします。I0 および I1 入力には、2 つのルックアップ テーブルのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は一般的なインターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。「MUXF5」および「MUXF5_L」も参照してください。

論理表

入力			出力	
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

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 directly to the output of LUT4)
    S => S     -- Input select to MUX
);

-- End of MUXF5_D_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// MUXF5_D: Slice MUX to tie two LUT4's together with general and local outputs
//           For use with Virtex-4, Spartan-3/3E/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 directly to the output of LUT4)
    .S(S)    // Input select to MUX
);

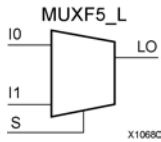
// End of MUXF5_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXF5_L

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エLEMENTは、ルックアップ テーブルと組み合わせて、5 ファンクションのルックアップ テーブルまたは 4:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、CLB スライス を 1 つ使用してインプリメントします。I0 および I1 入力には、2 つのルックアップ テーブルのローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

「MUXF5」および「MUXF5_D」も参照してください。

論理表

入力			出力
S	I0	I1	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO,  -- 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 記述 (インスタンスレーション)

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

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

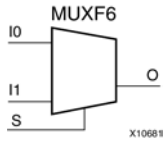
// End of MUXF5_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXF6

プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、対応する 4 つのルックアップ テーブルと 2 つの MUXF5 を組み合わせて、6 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、スライス 2 つにインプリメントします。I0 および I1 入力には、同じ CLB 内にある 2 つの MUXF5 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

このほか、ローカル出力を持つ MUXF6_D および MUXF6_L があり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

論理表

入力			出力
S	I0	I1	O
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO,   -- 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 記述 (インスタンスレーション)

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

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

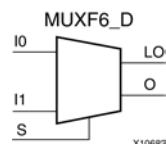
// End of MUXF6_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXF6_D

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、対応する 4 つのルックアップ テーブルと 2 つの MUXF5 を組み合わせて、6 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、スライス 2 つにインプリメントします。I0 および I1 入力には、同じ CLB 内にある 2 つの MUXF5 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は一般的なインターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

入力			出力	
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

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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,  -- Ouput of MUX to local routing
    O => O,    -- Output of MUX to general routing
    IO => IO,  -- 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 記述 (インスタンスレーション)

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

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

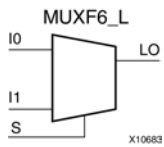
// End of MUXF6_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXF6_L

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エLEMENTは、対応する 4 つのルックアップ テーブルと 2 つの MUXF5 を組み合わせて、6 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサを作成するためのマルチプレクサ ファンクションをインプリメントします。I0 および I1 入力には、同じ CLB 内にある 2 つの MUXF5 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

入力			出力
S	I0	I1	LO
0	1	X	1
0	0	X	0
1	X	1	1
1	X	0	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンシエーション)

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

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

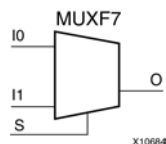
// End of MUXF6_L_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

MUXF7

プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブルと組み合わせて、7 ファンクションのルックアップ テーブルまたは 8:1 マルチプレクサを作成するためのマルチプレクサ ファンクションをインプリメントします。I0 および I1 入力には、MUXF6 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

このほか、ローカル出力を持つ MUXF7_D および MUXF7_L があり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

論理表

入力			出力
S	I0	I1	O
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

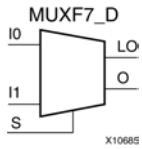
// End of MUXF7_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXF7_D

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブルと組み合わせて、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサを作成するためのマルチプレクサ ファンクションをインプリメントします。I0 および I1 入力には、MUXF6 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は一般的なインターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

入力			出力	
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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF6 LO 出力に接続)
I1	入力	1	入力 (MUXF6 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO,  -- 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_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// 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
    .IO(IO), // 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
);

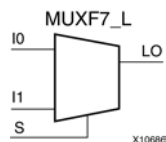
// End of MUXF7_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXF7_L

プリミティブ : 2-to-1 look-up table Multiplexer with Local Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブルと組み合わせて、7 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサを作成するためのマルチプレクサ ファンクションをインプリメントします。I0 および I1 入力には、MUXF6 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

入力			出力
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力
I1	入力	1	入力
S	入力	1	MUX への入力セレクト

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO,  -- 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 記述 (インスタンス化)

```
// 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_inst (
    .LO(LO), // Output of MUX to local routing
    .IO(IO), // 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
);

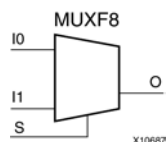
// End of MUXF7_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXF8

プリミティブ : 2-to-1 Look-Up Table Multiplexer with General Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブルと MUXF5、MUXF6、および MUXF7 を組み合わせて、8 ファンクションのルックアップ テーブルまたは 16:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、CLB 2 個 (スライス 8 個) にインプリメントします。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

論理表

入力			出力
S	I0	I1	O
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO,     -- 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 記述 (インスタンスレーション)

```
// 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
    .IO(IO),     // Input (tie to MUXF7 LO out)
    .I1(I1),     // Input (tie to MUXF7 LO out)
    .S(S)        // Input select to MUX
);

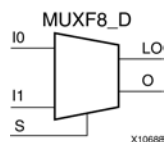
// End of MUXF8_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

MUXF8_D

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Dual Output



概要

このデザイン エLEMENTは、対応するルックアップ テーブル 4 つと MUXF8 を 2 つ組み合わせて、8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、CLB 2 個 (スライス 8 個) にインプリメントします。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

出力 O と LO は、機能的に同じです。出力 O は一般的なインターコネクトです。LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

入力			出力	
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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	汎用配線への MUX の出力
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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,  -- Ouput of MUX to local routing
    O => O,    -- Output of MUX to general routing
    IO => IO,  -- 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 記述 (インスタンスレーション)

```
// 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), // Ouput of MUX to local routing
    .O(O),  // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)   // Input select to MUX
);

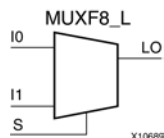
// End of MUXF8_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

MUXF8_L

プリミティブ : 2-to-1 Look-Up Table Multiplexer with Local Output



概要

このデザイン エレメントは、対応するルックアップ テーブル 4 つと MUXF8 を 2 つ組み合わせて、8 ファンクションのルックアップ テーブルまたは 32:1 マルチプレクサを作成するためのマルチプレクサ ファンクションを、CLB 2 個 (スライス 8 個) にインプリメントします。I0 および I1 入力には、MUXF7 のローカル出力 (LO) を接続します。セレクト入力 (S) は、どの内部ネットでも駆動できます。S が Low の場合は I0 が選択され、High の場合は I1 が選択されます。

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

入力			出力
S	I0	I1	LO
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

ポートの説明

ポート名	方向	幅	機能
LO	出力	1	ローカル配線への MUX の出力
I0	入力	1	入力 (MUXF7 LO 出力に接続)
I1	入力	1	入力 (MUXF7 LO 出力に接続)
S	入力	1	MUX への入力セレクト

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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
    IO => IO,  -- 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 記述 (インスタンスレーション)

```
// 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_inst (
    .LO(LO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

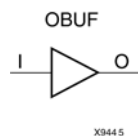
// End of MUXF8_L_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

OBUF

プリミティブ : Output Buffer



概要

このデザイン エLEMENTは単純な出力バッファで、出力信号を、トリステートでない FPGA デバイス ピンに駆動するために使用します。デザインのすべての出力ポートに OBUF、OBUFT、OBUFDS、OBUFTDS のいずれかを接続する必要があります。

このELEMENTは内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。I/O ブロック (IOB) 内にあります。出力 (O) は、OPAD または IOPAD に接続されます。このELEMENTでは、LVTTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE=12mA、スルー レートは SLOW に設定されています。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	最上位出力ポートに直接接続される OBUF の出力
I	入力	1	OBUF の入力。出力ポートを駆動するロジックに接続

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。許容範囲で最も低い値を使用してください。
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て
SLEW	文字列	SLOW、FAST	SLOW	出力ドライバのスルー レートを指定。この属性の最適な設定方法は、データシートを参照してください。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

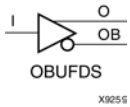
// End of OBUF_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

OBUFDS

プリミティブ : Differential Signaling Output Buffer



概要

このデザイン エLEMENTは、低電圧の差動信号 (1.8V CMOS) をサポートする単一の出力バッファです。内部回路を外部から分離し、チップから出力する信号の駆動電流を供給します。出力には 2 つの異なるポート (O および OB) があり、これらのポートをそれぞれ「マスタ」、「スレーブ」と呼びます。マスタとスレーブは MYNET と MYNETB のように、同じ論理信号の反対の状態を示します。

論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

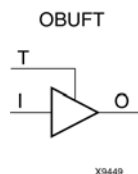
// End of OBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

OBUFT

プリミティブ : 3-State Output Buffer with Active Low Output Enable



概要

このデザイン エLEMENTは、入力 (I)、出力 (O)、アクティブ Low 出力イネーブル (T) を持つ単一のトライステート出力バッファです。このELEMENTでは、LVTTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE=12mA、スルー レートは SLOW に設定されています。

T が Low の場合、バッファに入力された値が対応する出力に送られます。T が High の場合は、出力がハイ インピーダンス (オフまたは Z ステート) になります。OBUFT は、双方向 I/O を作成するなど、トライステート機能にシングルエンド出力を使用する必要がある場合に使用します。

論理表

入力		出力
T	I	O
1	X	Z
0	I	f

ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファ出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。許容範囲で最も低い値を使用してください。
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て
SLEW	文字列	SLOW、FAST	SLOW	出力ドライバのスループートを指定。この属性の最適な設定方法は、データシートを参照してください。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

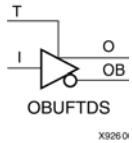
// End of OBUFT_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

OBUFTDS

プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable



概要

このデザイン エLEMENTは、低電圧差動信号を使用する出力バッファです。OBUFTDS では、デザイン レベルのインターフェイス信号は、一方が「マスタ」で、もう一方が「スレーブ」となる 2 つの異なるポート (O、OB) で表されます。マスタとスレーブは MYNET_P と MYNET_N のように、同じ論理信号の反対の状態を示します。

論理表

入力		出力	
I	T	O	OB
X	1	Z	Z
0	0	0	1
1	0	1	0

ポートの説明

ポート名	方向	幅	機能
O	出力	1	Diff_p 出力 (最上位ポートに直接接続)
OB	出力	1	Diff_n 出力 (最上位ポートに直接接続)
I	入力	1	バッファの入力
T	入力	1	トライステート イネーブル入力

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- OBUFTDS: Differential 3-state Output Buffer
--           Spartan-3/3E/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 記述 (インスタンスレーション)

```
// OBUFTDS: Differential 3-state Output Buffer
//           Virtex-4/5, Spartan-3/3E/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
);

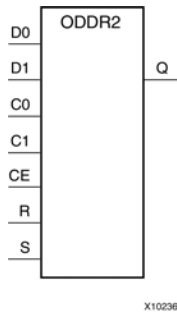
// End of OBUFTDS_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

ODDR2

プリミティブ : Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset



概要

このデザイン エLEMENTは、出力ダブル データ レート (DDR) レジスタで、FPGA から出力されるダブル データ レート 信号を生成するために使用します。ODDR2 は、C0 と C1 の 2 つのクロックを使用してコンポーネントに接続されるので、C0 および C1 の両方の立ち上がりエッジでデータが出力されます。ODDR2 は、レジスタの動作を停止するために使用できるアクティブ High のクロック イネーブル (CE) ポート、対応するクロックに同期または非同期になるよう設定できるセット/リセット ポートを備えています。ODDR2 には、1 クロックで取り込まれたデータを 2 クロックで出力するオプションの調整機能があります。

論理表

入力							出力
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	変化なし
0	0	1	D0	X	↑	X	D0
0	0	1	X	D1	X	↑	D1

セット/リセットは SRTYPE 値で同期に設定可能

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DDR_ALIGNMENT	文字列	NONE、C0、C1	NONE	DDR レジスタの入力キャプチャビヘイビアを設定します。NONE に設定すると、C0 クロックが Low から High に切り替わる時は D0 入力に、C1 クロックが Low から High に切り替わる時は D1 にデータを入力します。C0 では、D0 と D1 両方への入力が C0 クロックの立ち上がりエッジに同期します。C1 では、D0 と D1 両方への入力が C1 クロックの立ち上がりエッジに同期します。
INIT	整数	0、1	0	Q0 出力の初期値を 0 または 1 に設定
SRTYPE	文字列	SYNC、ASYNC	SYNC	セット/リセットを SYNC または ASYNC に設定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
```

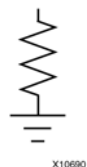
詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

PULLDOWN

プリミティブ : Resistor to GND for Input Pads, Open-Drain, and 3-State Outputs

PULLDOWN



概要

この抵抗エレメントは、入力、出力、双方向のパッドに接続し、フロートする可能性のあるノードのロジックレベルを Low にします。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルダウン出力 (最上位ポートに直接接続)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

PULLUP

プリミティブ : Resistor to VCC for Input PADs, Open-Drain, and 3-State Outputs



概要

このデザイン エLEMENTは、1 つの入力、トライステート出力、または双方向ポートが内部または外部ソースで駆動されないときに、値、weak High で駆動できます。このELEMENTは、すべてのドライバが使用されていないときにオープンドレイン ELEMENTおよびマクロのロジック レベルを 1 (High) にします。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	プルアップ出力 (最上位ポートに直接接続)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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)
);

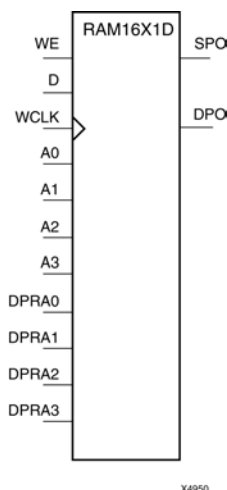
// End of PULLUP_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

RAM16X1D

プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM



概要

このエレメントは 16 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA3 ~ DPRA0) と書き込みアドレス (A3 ~ A0) の 2 種類のアドレス ポートがあります。この 2 種類のアドレス ポートは非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。

WE が High の場合、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 4 ビットの書き込みアドレスで選択されたワードにロードされます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバータを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバータは、RAM ブロック内に組み込まれます。

SPO 出力には、A3 ~ A0 で指定されたメモリ セルの値が出力されます。DPO 出力には、DPRA3 ~ DPRA0 で指定されたメモリ セルの値が出力されます。

メモ : 書き込み処理は、読み出しアドレス ポートのアドレスには影響されません。

INIT 属性を使用すると、RAM を直接初期化できます。値は、INIT=ABAC のように、16 進数で指定してください。INIT 属性を指定しない場合は、RAM は 0 に初期化されます。

論理表

モード選択を次の論理表に示します。

入力			出力	
WE (モード)	WCLK	D	SPO	DPO
0 (読み出し)	X	X	data_a	data_d
1 (読み出し)	0	X	data_a	data_d
1 (読み出し)	1	X	data_a	data_d
1 (書き込み)	↑	D	D	data_d
1 (読み出し)	↓	X	data_a	data_d
data_a = A3 ~ A0 で指定されたワード				
data_d = DPRA3 ~ DPRA0 で指定されたワード				

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	RAM、レジスタ、LUT の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

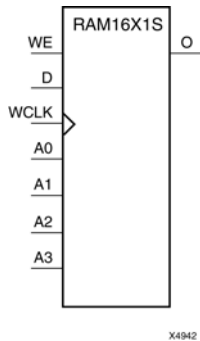
// End of RAM16X1D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

RAM16X1S

プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM



概要

このエレメントは 16 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 4 ビットのアドレス (A3 ~ A0) で選択されたワードにロードされます。WCLK はデフォルトではアクティブ High ですが、インバータを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバータは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM16X1S を初期化できます。

論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ
データ = A3 ~ A0 で指定されたワード			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```

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 記述 (インスタンスレーション)

```

// 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

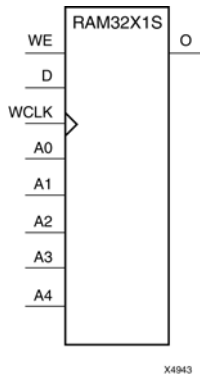
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

RAM32X1S

プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 5 ビットのアドレス (A4 ~ A0) で選択されたワードにロードされます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバータを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバータは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1S を初期化できます。

論理表

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↓	D	D
1 (読み出し)	↑	X	データ

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスエーション)

```
// 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
);

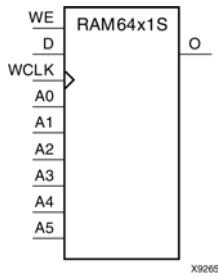
// End of RAM32X1S_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

RAM64X1S

プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 64 ワード X 1 ビットの SRAM で、同期書き込み機能を備えています。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 6 ビットのアドレス (A5 ~ A0) で選択されたワードにロードされます。WCLK はデフォルトではアクティブ High ですが、インバータを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバータは、RAM ブロック内に組み込まれます。

出力ピン (O) に出力される値は、アドレス ピンで指定された RAM 内の位置に格納されている値です。

INIT 属性を使用すると、コンフィギュレーション中にこのELEMENTを初期化できます。

論理表

モード選択を次の論理表に示します。

入力			出力
WE (モード)	WCLK	D	O
0 (読み出し)	X	X	データ
1 (読み出し)	0	X	データ
1 (読み出し)	1	X	データ
1 (書き込み)	↑	D	D
1 (読み出し)	↓	X	データ
データ = A5 ~ A0 で指定されたワード			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ROM、RAM、レジスタ、LUT の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

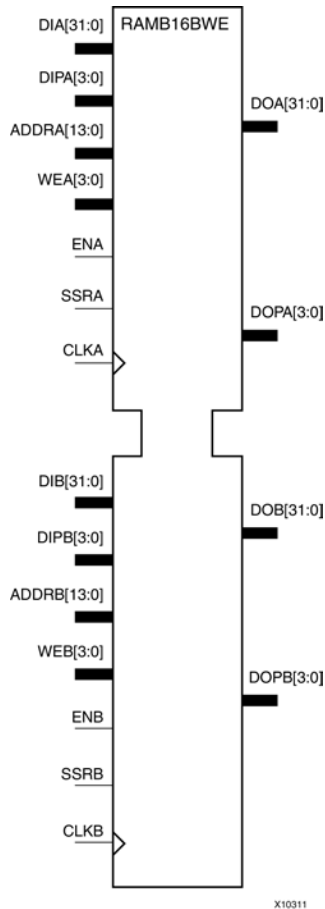
// End of RAM64X1S_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

RAMB16BWE

プリミティブ : 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM



概要

このデザイン エLEMENTは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA, DOB	出力	32	ポート A/B のデータ出力バス
DOPA, DOPB	出力	4	ポート A/B のパリティ出力バス
DIA, DIB	入力	32	ポート A/B のデータ入力バス
DIPA, DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA, ADDRb	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。

ポート名	方向	幅	機能
WEA, WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA, ENB	入力	1	ポート A/B のイネーブル
SSRA, SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA, CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイトライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要なければ、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 ・ READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 ・ NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定


```

INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
port map (
  DOA => DOA,      -- 32-bit A port Data Output
  DOB => DOB,      -- 32-bit B port Data Output
  DOPA => DOPA,    -- 4-bit A port Parity Output
  DOPB => DOPB,    -- 4-bit B port Parity Output
  ADDRA => ADDRA,  -- 14-bit A port Address Input
  ADDRb => ADDRb,  -- 14-bit B port Address Input
  CLKA => CLKA,    -- Port A Clock
  CLKB => CLKB,    -- Port B Clock
  DIA => DIA,      -- 32-bit A port Data Input
  DIB => DIB,      -- 32-bit B port Data Input
  DIPA => DIPA,    -- 4-bit A port parity Input
  DIPB => DIPB,    -- 4-bit B port parity Input
  ENA => ENA,      -- 1-bit A port Enable Input
  ENB => ENB,      -- 1-bit B port Enable Input
  SSRA => SSRA,    -- 1-bit A port Synchronous Set/Reset Input
  SSRB => SSRB,    -- 1-bit B port Synchronous Set/Reset Input
  WEA => WEA,      -- 4-bit A port Write Enable Input
  WEB => WEB       -- 4-bit B port Write Enable Input
);

-- End of RAMB16BWE_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// RAMB16BWE: 16k+2k Parity Paramatizable, byte-wide enable BlockRAM
//          Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE #(
  .DATA_WIDTH_A(0), // Valid values are 1, 2, 4, 9, 18, or 36
  .DATA_WIDTH_B(0), // Valid values are 1, 2, 4, 9, 18, or 36
  .INIT_A(36'h000000000), // Initial values on A output port
  .INIT_B(36'h000000000), // Initial values on B output port
  .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                // "GENERATE_X_ONLY" or "NONE"
  .SRVAL_A(36'h000000000), // Set/Reset value for A port output
  .SRVAL_B(36'h000000000), // Set/Reset value for B port output
  .WRITE_MODE_A("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
  .WRITE_MODE_B("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"

  // The following INIT_xx declarations specify the initial contents of the RAM
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

Spartan-3A および Spartan-3A DSP ライブラリ ガイド (HDL 用)
UG613 (v11.3) 2009 年 9 月 16 日 <http://japan.xilinx.com>

```
.DOPB(DOPB),    // 4-bit B port parity data output
.ADDRA(ADDRA),  // 14-bit A port address input
.ADDRB(ADDRB),  // 14-bit B port address input
.CLKA(CLKA),    // 1-bit A port clock input
.CLKB(CLKB),    // 1-bit B port clock input
.DIA(DIA),      // 32-bit A port data input
.DIB(DIB),      // 32-bit B port data input
.DIPA(DIPA),    // 4-bit A port parity data input
.DIPB(DIPB),    // 4-bit B port parity data input
.ENA(ENA),      // 1-bit A port enable input
.ENB(ENB),      // 1-bit B port enable input
.SSRA(SSRA),    // 1-bit A port set/reset input
.SSRB(SSRB),    // 1-bit B port set/reset input
.WEA(WEA),      // 4-bit A port write enable input
.WEB(WEB),      // 4-bit B port write enable input
);

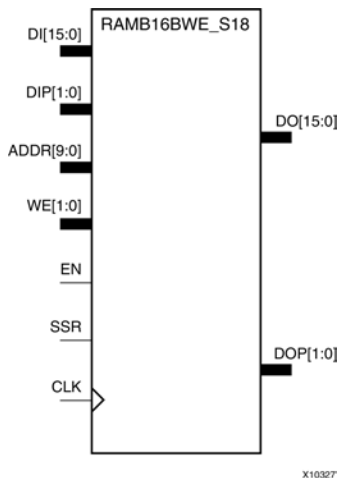
// End of RAMB16BWE_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

RAMB16BWE_S18

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 18-bit Port



概要

このデザイン エレメントは、1 ビット X 16K ワード～ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA、DOB	出力	32	ポート A/B のデータ出力バス
DOPA、DOPB	出力	4	ポート A/B のパリティ出力バス
DIA、DIB	入力	32	ポート A/B のデータ入力バス
DIPA、DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA、ADDRB	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA、WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要なければ、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 ・ READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 ・ NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

```
-- RAMB16BWE_S18: 1k x 16 + 2 Parity bits Single-Port byte-wide write RAM
--
-- Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S18_inst : RAMB16BWE_S18
generic map (
  INIT => X"00000", -- Value of output RAM registers at startup
  SRVAL => X"00000", -- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- Address 256 to 511
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- Address 512 to 767
  INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- Address 768 to 1023
  INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```

INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DO => DO,      -- 16-bit Data Output
  DOP => DOP,     -- 2-bit parity Output
  ADDR => ADDR,   -- 10-bit Address Input
  CLK => CLK,     -- 1-bit Clock
  DI => DI,       -- 16-bit Data Input
  DIP => DIP,     -- 2-bit parity Input
  EN => EN,       -- 1-bit RAM Enable Input
  SSR => SSR,     -- 1-bit Synchronous Set/Reset Input
  WE => WE        -- 2-bit Write Enable Input
);

-- End of RAMB16BWE_S18_inst instantiation

```

Verilog 記述 (インスタネーション)

```

// RAMB16BWE_S18: 1k x 16 + 2 Parity bits Single-Port byte-wide write RAM
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_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),

```

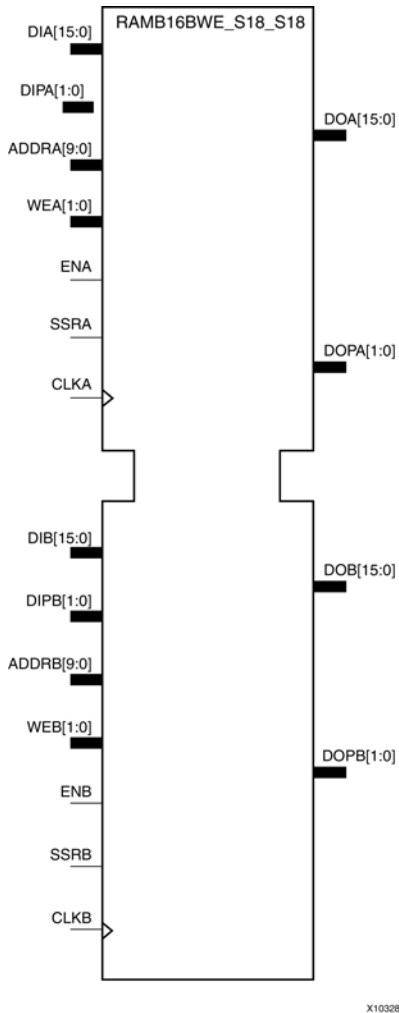
[illegible]

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

RAMB16BWE_S18_S18

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit Ports



概要

このデザイン エLEMENTは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA、DOB	出力	32	ポート A/B のデータ出力バス
DOPA、DOPB	出力	4	ポート A/B のパリティ出力バス
DIA、DIB	入力	32	ポート A/B のデータ入力バス
DIPA、DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA、ADDRB	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA、WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままでかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要なければ、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。

属性	タイプ	値	デフォルト	説明
				<ul style="list-style-type: none"> ・ READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 ・ NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16BWE_S18_S18: 1k x 16 + 2 Parity bits Dual-Port byte-wide write RAM
--                      Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S18_S18_inst : RAMB16BWE_S18_S18
generic map (
  INIT_A => X"00000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"00000", -- Value of output RAM registers on Port B at startup
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  SRVAL_A => X"00000", -- Port A output value upon SSR assertion
  SRVAL_B => X"00000", -- 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
  -- The following INIT_xx declarations specify the initial contents of the RAM
  -- Address 0 to 255
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- Address 256 to 511
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",

```

```
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 255
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 511
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 512 to 767
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 768 to 1023
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  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 1-bit Clock
  CLKB => CLKB,    -- Port B 1-bit 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 1-bit RAM Enable Input
  ENB => ENB,      -- Port B 1-bit RAM Enable Input
  SSRA => SSRA,    -- Port A 1-bit Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B 1-bit Synchronous Set/Reset Input
  WEA => WEA,      -- Port A 2-bit Write Enable Input
  WEB => WEB,      -- Port B 2-bit Write Enable Input
);

-- End of RAMB16BWE_S18_S18_inst instantiation
```

Verilog 記述 (インスタンス化)

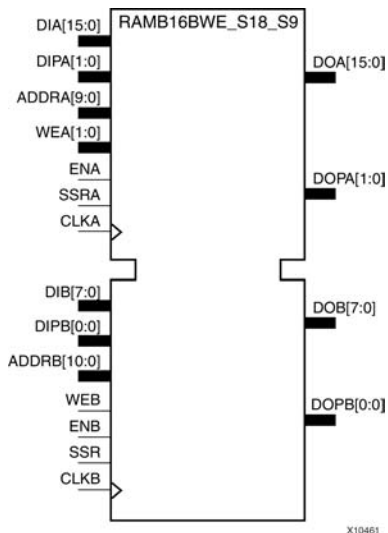
```
// RAMB16BWE_S18_S18: 1k x 16 + 2 Parity bits Dual-Port byte-wide write RAM
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_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
    .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                // "GENERATE_X_ONLY" or "NONE"
    .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),
```


RAMB16BWE_S18_S9

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 18-bit and 9-bit Ports



概要

このデザイン エLEMENTは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA, DOB	出力	32	ポート A/B のデータ出力バス
DOPA, DOPB	出力	4	ポート A/B のパリティ出力バス
DIA, DIB	入力	32	ポート A/B のデータ入力バス
DIPA, DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA, ADDRb	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA, WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA, ENB	入力	1	ポート A/B のイネーブル
SSRA, SSRb	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA, CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要なければ、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

```
-- RAMB16BWE_S18_S9: 1k/2k x 16/8 + 2/1 Parity bits Dual-Port byte-wide write RAM
--
-- Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S18_S9_inst : RAMB16BWE_S18_S9
generic map (
    INIT_A => X"00000", -- Value of output RAM registers on Port A at startup
    INIT_B => X"000", -- Value of output RAM registers on Port B at startup
    SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
    SRVAL_A => X"00000", -- 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
    -- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- Port A Address 256 to 511, Port B Address 128 to 255
    INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- Port A Address 512 to 767, Port B Address 256 to 383
    INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- Port A Address 768 to 1023, Port B Address 384 to 511
    INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
```



```

INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 16-bit Data Output
  DOB => DOB,      -- Port B 8-bit Data Output
  DOPA => DOPA,    -- Port A 2-bit Parity Output
  DOPB => DOPB,    -- Port B 1-bit Parity Output
  ADDRA => ADDRA,  -- Port A 10-bit Address Input
  ADDRb => ADDRb,  -- Port B 11-bit Address Input
  CLKA => CLKA,    -- Port A 1-bit Clock
  CLKB => CLKB,    -- Port B 1-bit Clock
  DIA => DIA,      -- Port A 16-bit Data Input
  DIB => DIB,      -- Port B 8-bit Data Input
  DIPa => DIPa,    -- Port A 2-bit parity Input
  DIPB => DIPB,    -- Port B 1-bit parity Input
  ENA => ENA,      -- Port A 1-bit RAM Enable Input
  ENB => ENB,      -- Port B 1-bit RAM Enable Input
  SSRA => SSRA,    -- Port A 1-bit Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B 1-bit Synchronous Set/Reset Input
  WEA => WEA,      -- Port A 2-bit Write Enable Input
  WEB => WEB       -- Port B 1-bit Write Enable Input
);

-- End of RAMB16BWE_S18_S9_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// RAMB16BWE_S18_S9: 1k/2k x 16/8 + 2/1 Parity bits Dual-Port byte-wide write RAM
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S18_S9 #(
  .INIT_A(18'h00000), // Value of output RAM registers on Port A at startup
  .INIT_B(9'h000),   // Value of output RAM registers on Port B at startup
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
  .SRVAL_A(18'h00000), // 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

  // 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),

```

[illegible]

```
.DOB(DOB),          // Port B 8-bit Data Output
.DOPA(DOPA),        // Port A 2-bit Parity Output
.DOPB(DOPB),        // Port B 1-bit Parity Output
.ADDRA(ADDRA),       // Port A 10-bit Address Input
.ADDRB(ADDRB),       // Port B 11-bit Address Input
.CLKA(CLKA),         // Port A 1-bit Clock
.CLKB(CLKB),         // Port B 1-bit Clock
.DIA(DIA),           // Port A 16-bit Data Input
.DIB(DIB),           // Port B 8-bit Data Input
.DIPA(DIPA),         // Port A 2-bit parity Input
.DIPB(DIPB),         // Port B 1-bit parity Input
.ENA(ENA),           // Port A 1-bit RAM Enable Input
.ENB(ENB),           // Port B 1-bit RAM Enable Input
.SSRA(SSRA),         // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),         // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),           // Port A 2-bit Write Enable Input
.WEB(WEB),           // Port B 1-bit Write Enable Input
);

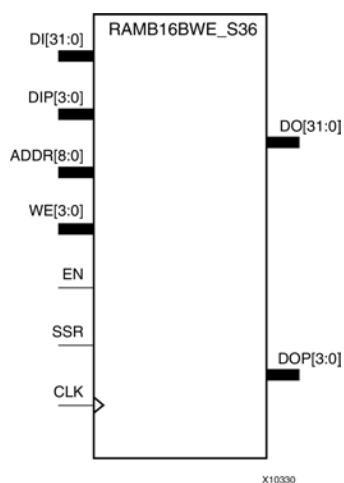
// End of RAMB16BWE_S18_S9_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

RAMB16BWE_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Single Port Block RAM with 36-Bit Port



概要

このデザイン エLEMENTは、1 ビット X 16K ワード～ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA、DOB	出力	32	ポート A/B のデータ出力バス
DOPA、DOPB	出力	4	ポート A/B のパリティ出力バス
DIA、DIB	入力	32	ポート A/B のデータ入力バス
DIPA、DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA、ADDRB	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA、WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要なければ、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 ・ READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 ・ NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

Spartan-3A および Spartan-3A DSP ライブラリ ガイド (HDL 用)
UG613 (v11.3) 2009 年 9 月 16 日 <http://japan.xilinx.com>

```

INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DO => DO,      -- 32-bit Data Output
  DOP => DOP,     -- 4-bit parity Output
  ADDR => ADDR,   -- 9-bit Address Input
  CLK => CLK,     -- 1-bit Clock
  DI => DI,       -- 32-bit Data Input
  DIP => DIP,     -- 4-bit parity Input
  EN => EN,       -- 1-bit RAM Enable Input
  SSR => SSR,     -- 1-bit Synchronous Set/Reset Input
  WE => WE        -- 4-bit Write Enable Input
);

-- End of RAMB16BWE_S36_inst instantiation

```

Verilog 記述 (インスタネーション)

```

// RAMB16BWE_S36: 512 x 32 + 4 Parity bits Single-Port byte-wide write RAM
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S36 #(
  .INIT(36'h000000000), // Value of output RAM registers at startup
  .SRVAL(36'h000000000), // 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),

```

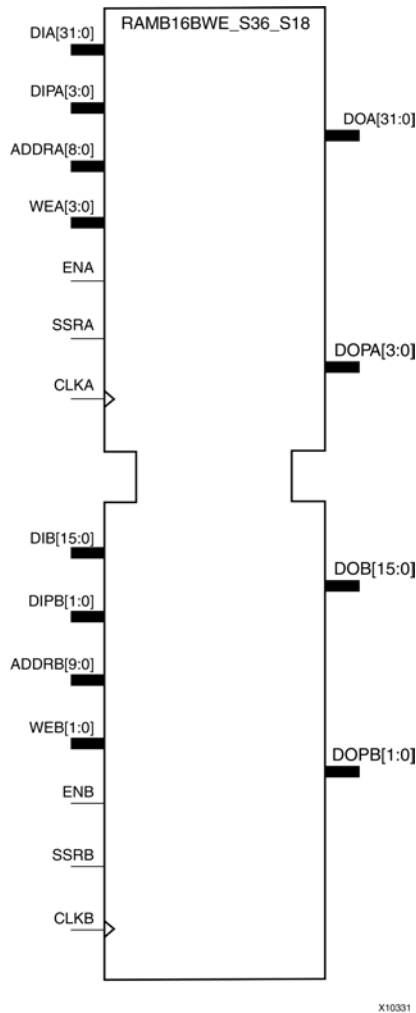

Spartan-3A および Spartan-3A DSP ライブラリ ガイド (HDL 用)
UG613 (v11.3) 2009 年 9 月 16 日 <http://japan.xilinx.com>

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

RAMB16BWE_S36_S18

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 18-bit Ports



概要

このデザイン エLEMENTは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA、DOB	出力	32	ポート A/B のデータ出力バス
DOPA、DOPB	出力	4	ポート A/B のパリティ出力バス
DIA、DIB	入力	32	ポート A/B のデータ入力バス
DIPA、DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA、ADDRB	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA、WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままでかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要なければ、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。

属性	タイプ	値	デフォルト	説明
				<ul style="list-style-type: none"> ・ READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 ・ NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB16BWE_S36_S18: 1k/2k x 16/8 + 2/1 Parity bits Dual-Port byte-wide write RAM
--                      Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S36_S18_inst : RAMB16BWE_S36_S18
generic map (
  INIT_A => X"0000000000", -- Value of output RAM registers on Port A at startup
  INIT_B => X"000000", -- Value of output RAM registers on Port B at startup
  SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
  SRVAL_A => X"0000000000", -- Port A output value upon SSR assertion
  SRVAL_B => X"000000", -- 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
  -- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  -- Port A Address 256 to 511, Port B Address 128 to 255
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",

```

```
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 32-bit Data Output
  DOB => DOB,      -- Port B 16-bit Data Output
  DOPA => DOPA,    -- Port A 4-bit Parity Output
  DOPB => DOPB,    -- Port B 2-bit Parity Output
  ADDRA => ADDRA,  -- Port A 9-bit Address Input
  ADDRb => ADDRb,  -- Port B 10-bit Address Input
  CLKA => CLKA,    -- Port A 1-bit Clock
  CLKB => CLKB,    -- Port B 1-bit Clock
  DIA => DIA,      -- Port A 32-bit Data Input
  DIB => DIB,      -- Port B 16-bit Data Input
  DIPa => DIPa,    -- Port A 4-bit parity Input
  DIPB => DIPB,    -- Port B 2-bit parity Input
  ENA => ENA,      -- Port A 1-bit RAM Enable Input
  ENB => ENB,      -- Port B 1-bit RAM Enable Input
  SSRA => SSRA,    -- Port A 1-bit Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B 1-bit Synchronous Set/Reset Input
  WEA => WEA,      -- Port A 4-bit Write Enable Input
  WEB => WEB,      -- Port B 2-bit Write Enable Input
);

-- End of RAMB16BWE_S36_S18_inst instantiation
```

Verilog 記述 (インスタンス化)

[illegible]


```
.INIT_35(256'h00000000000000000000000000000000000000000000000000000),
.INIT_36(256'h00000000000000000000000000000000000000000000000000000),
.INIT_37(256'h00000000000000000000000000000000000000000000000000000),
.INIT_38(256'h00000000000000000000000000000000000000000000000000000),
.INIT_39(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h00000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h00000000000000000000000000000000000000000000000000000),

// 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'h00000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 256 to 511, Port B Address 128 to 255
.INITP_02(256'h00000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 512 to 767, Port B Address 256 to 383
.INITP_04(256'h00000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000),
// Port A Address 768 to 1023, Port B Address 384 to 511
.INITP_06(256'h00000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000)
) RAMB16BWE_S36_S18_inst (
.DOA(DOA), // Port A 32-bit Data Output
.DOB DOB), // Port B 16-bit Data Output
.DOPA(DOPA), // Port A 4-bit Parity Output
.DOPB(DOPB), // Port B 2-bit Parity Output
.ADDRA(ADDR A), // Port A 9-bit Address Input
.ADRB(ADRB), // Port B 10-bit Address Input
.CLKA(CLKA), // Port A 1-bit Clock
.CLKB(CLKB), // Port B 1-bit Clock
.DIA(DIA), // Port A 32-bit Data Input
.DIB(DIB), // Port B 16-bit Data Input
.DIPA(DIPA), // Port A 4-bit parity Input
.DIPB(DIPB), // Port-B 2-bit parity Input
.ENA(ENA), // Port A 1-bit RAM Enable Input
.ENB(ENB), // Port B 1-bit RAM Enable Input
.SSRA(SSRA), // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB), // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA), // Port A 4-bit Write Enable Input
.WEB(WEB) // Port B 2-bit Write Enable Input
);

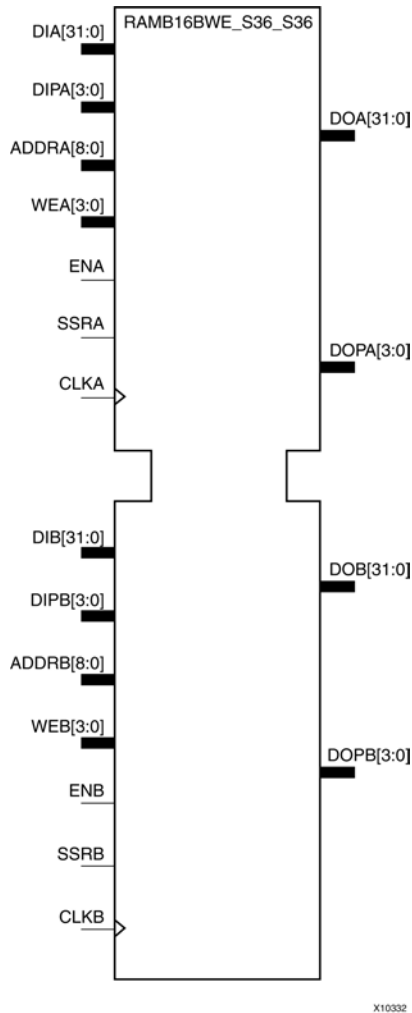
// End of RAMB16BWE S36_S18 inst instantiation
```

詳細情報

- ・ Spartan-3 ジェネレーション FPGA ユーザー ガイド
- ・ Spartan-3A FPGA ファミリ データシート

RAMB16BWE_S36_S36

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit Ports



概要

このデザイン エLEMENTは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA、DOB	出力	32	ポート A/B のデータ出力バス
DOPA、DOPB	出力	4	ポート A/B のパリティ出力バス
DIA、DIB	入力	32	ポート A/B のデータ入力バス
DIPA、DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA、ADDRB	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA、WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままでかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]


```
INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- The next set of INITP_xx are for the parity bits
-- Address 0 to 127
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  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 1-bit Clock
  CLKB => CLKB,    -- Port B 1-bit 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 1-bit RAM Enable Input
  ENB => ENB,      -- Port B 1-bit RAM Enable Input
  SSRA => SSRA,    -- Port A 1-bit Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B 1-bit Synchronous Set/Reset Input
  WEA => WEA,      -- Port A 4-bit Write Enable Input
  WEB => WEB       -- Port B 4-bit Write Enable Input
);

-- End of RAMB16BWE_S36_S36_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAMB16BWE_S36_S36: 512 x 32 + 4 Parity bits byte-wide write Dual-Port RAM
//                               Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S36_S36 #(
    .INIT_A(36'h00000000), // Value of output RAM registers on Port A at startup
    .INIT_B(36'h00000000), // Value of output RAM registers on Port B at startup
    .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
    .SRVAL_A(36'h00000000), // Port A output value upon SSR assertion
    .SRVAL_B(36'h00000000), // 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

    // 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'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 128 to 255
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 256 to 383
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
// Address 384 to 511
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB16BWE_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 1-bit Clock
.CLKB(CLKB), // Port B 1-bit 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 1-bit RAM Enable Input
.ENB(ENB), // Port B 1-bit RAM Enable Input
.SSRA(SSRA), // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB), // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA), // Port A 4-bit Write Enable Input
.WEB(WEB) // Port B 4-bit Write Enable Input
);

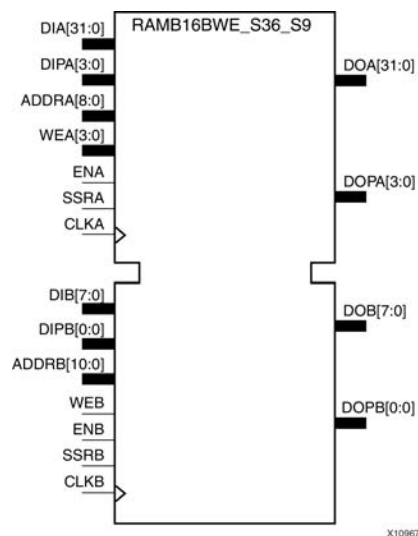
// End of RAMB16BWE_S36_S36_inst instantiation
```

詳細情報

- ・ Spartan-3 ジェネレーション FPGA ユーザー ガイド
- ・ Spartan-3A FPGA ファミリ データシート

RAMB16BWE_S36_S9

プリミティブ : 16K-bit Data and 2K-bit Parity Synchronous Dual Port Block RAM with 36-bit and 9-bit Ports



概要

このデザイン エLEMENTは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。このブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。

ポートの説明

ポート名	方向	幅	機能
DOA, DOB	出力	32	ポート A/B のデータ出力バス
DOPA, DOPB	出力	4	ポート A/B のパリティ出力バス
DIA, DIB	入力	32	ポート A/B のデータ入力バス
DIPA, DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA, ADDR[10:0]	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDR[10:0] ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA, WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA, ENB	入力	1	ポート A/B のイネーブル
SSRA, SSRB	入力	1	ポート A/B の出力レジスタの同期リセット
CLKA, CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

このエレメントは、標準の RTL 記述で RAM のビヘイビアを記述することにより、ほとんどの合成ツールで推論できます。詳細は、合成ツールのマニュアルを参照してください。または、CORE Generator™ でもこの RAMB のマクロを作成できます。直接インスタンス化して、このコンポーネントのインプリメンテーションまたは配置を細かく制御することも可能です。このコンポーネントをインスタンス化するには ISE® で HDL テンプレートを使用するか、または次のインスタンス化 テンプレートをコードに貼り付けます。必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、SSRA/SSRB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、次の表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままかまいません。使用されていない入力信号は論理値 0 に接続してください。

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイト ライト イネーブルに接続	DO[31:0]、DOP[3:0]

バイト イネーブル操作が必要であれば、代わりに RAMB16_Sm_Sn デザイン エLEMENTをインスタンス化できます。また、新しいマクロ RAMB16BWE_Sm_Sn を使用すると、バイト イネーブル操作を実行するこの RAM のインスタンス化を簡単に実行できます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE コンポーネントに変更されます。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A、 DATA_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート A および B のデータ幅を指定
INIT_A、 INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
SIM_COLLISION_ CHECK	文字列	ALL、 WARNING_ ONLY、 GENERATE_X_ ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SRVAL_A、 SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> ・ WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 ・ READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 ・ NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

```
-- RAMB16BWE_S36_S9: 2k/512 x 8/32 + 1/4 Parity bits Dual-Port byte-wide write RAM
--
-- Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S36_S9_inst : RAMB16BWE_S36_S9
generic map (
    INIT_A => X"0000000000", -- Value of output RAM registers on Port A at startup
    INIT_B => X"000", -- Value of output RAM registers on Port B at startup
    SIM_COLLISION_CHECK => "ALL", -- "NONE", "WARNING", "GENERATE_X_ONLY", "ALL"
    SRVAL_A => X"000000000", -- 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
    -- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- Port A Address 256 to 511, Port B Address 128 to 255
    INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- Port A Address 512 to 767, Port B Address 256 to 383
    INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- Port A Address 768 to 1023, Port B Address 384 to 511
    INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

```

INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_3F => X"0000000000000000000000000000000000000000000000000000000000000000",
-- 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 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 256 to 511, Port B Address 128 to 255
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 512 to 767, Port B Address 256 to 383
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
-- Port A Address 768 to 1023, Port B Address 384 to 511
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  DOA => DOA,      -- Port A 32-bit Data Output
  DOB => DOB,      -- Port B 8-bit Data Output
  DOPA => DOPA,    -- Port A 4-bit Parity Output
  DOPB => DOPB,    -- Port B 1-bit Parity Output
  ADDRA => ADDRA,  -- Port A 9-bit Address Input
  ADDRb => ADDRb,  -- Port B 11-bit Address Input
  CLKA => CLKA,    -- Port A 1-bit Clock
  CLKB => CLKB,    -- Port B 1-bit Clock
  DIA => DIA,      -- Port A 32-bit Data Input
  DIB => DIB,      -- Port B 8-bit Data Input
  DIPa => DIPa,    -- Port A 4-bit parity Input
  DIPB => DIPB,    -- Port B 1-bit parity Input
  ENA => ENA,      -- Port A 1-bit RAM Enable Input
  ENB => ENB,      -- Port B 1-bit RAM Enable Input
  SSRA => SSRA,    -- Port A 1-bit Synchronous Set/Reset Input
  SSRB => SSRB,    -- Port B 1-bit Synchronous Set/Reset Input
  WEA => WEA,      -- Port A 4-bit Write Enable Input
  WEB => WEB       -- Port B 1-bit Write Enable Input
);

-- End of RAMB16BWE_S36_S9_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// RAMB16BWE_S36_S9: 2k/512 x 8/32 + 1/4 Parity bits Dual-Port byte-wide write RAM
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

RAMB16BWE_S36_S9 #(
  .INIT_A(36'h00000000), // Value of output RAM registers on Port A at startup
  .INIT_B(9'h000),      // Value of output RAM registers on Port B at startup
  .SIM_COLLISION_CHECK("ALL"), // "NONE", "WARNING_ONLY", "GENERATE_X_ONLY", "ALL"
  .SRVAL_A(36'h00000000), // 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

  // 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),

```

Spartan-3A および Spartan-3A DSP ライブラリ ガイド (HDL 用)
UG613 (v11.3) 2009 年 9 月 16 日 <http://japan.xilinx.com>

```
.DOB(DOB),          // Port B 8-bit Data Output
.DOPA(DOPA),        // Port A 4-bit Parity Output
.DOPB(DOPB),        // Port B 1-bit Parity Output
.ADDRA(ADDRA),      // Port A 9-bit Address Input
.ADDRB(ADDRB),      // Port B 11-bit Address Input
.CLKA(CLKA),        // Port A 1-bit Clock
.CLKB(CLKB),        // Port B 1-bit Clock
.DIA(DIA),          // Port A 32-bit Data Input
.DIB(DIB),          // Port B 8-bit Data Input
.DIPA(DIPA),        // Port A 4-bit parity Input
.DIPB(DIPB),        // Port B 1-bit parity Input
.ENA(ENA),          // Port A 1-bit RAM Enable Input
.ENB(ENB),          // Port B 1-bit RAM Enable Input
.SSRA(SSRA),        // Port A 1-bit Synchronous Set/Reset Input
.SSRB(SSRB),        // Port B 1-bit Synchronous Set/Reset Input
.WEA(WEA),          // Port A 4-bit Write Enable Input
.WEB(WEB),          // Port B 1-bit Write Enable Input
);

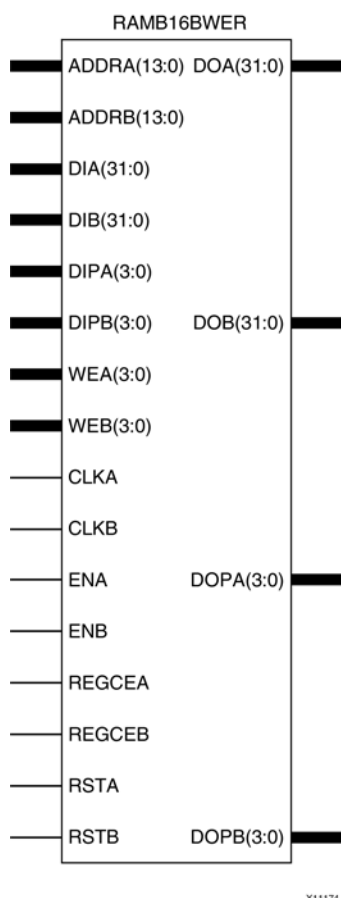
// End of RAMB16BWE_S36_S9_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

RAMB16BWER

プリミティブ : 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers



概要

複数のブロックRAMメモリが含まれていますが、これは汎用 16kb データ + 2kb パリティ RAM/ROM メモリとしてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。このコンポーネントは、1 ビット X 16K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。この RAM には、コンフィギュレーション可能な出力レジスタもあり、読み出し操作中 1 クロック サイクルのレイテンシが発生するときに RAM の clock-to-out タイムを向上させることができます。

ポートの説明

次の表に、ポート A またはポート B の DATA_WIDTH の値と、それに必要な入力接続と出力接続を示します。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続
1	DI[0]	ADDR[13:0]	WE[3:0] をシングル ユーザー WE 信号に接続
2	DI[1:0]	ADDR[13:1]	WE[3:0] をシングル ユーザー WE 信号に接続
4	DI[3:0]	ADDR[13:2]	WE[3:0] をシングル ユーザー WE 信号に接続
9	DI[7:0]、DIP[0]	ADDR[13:3]	WE[3:0] をシングル ユーザー WE 信号に接続
18	DI[15:0]、DIP[1:0]	ADDR[13:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続
36	DI[31:0]、DIP[3:0]	ADDR[13:5]	各 WE[3:0] 信号を関連したバイトライト イネーブルに接続

出力レジスタが必要ない場合は、代わりに古い RAMB16_Sm_Sn および RAMB16BWER_Sm_Sn エLEMENTをインスタンスエートできます。これらのコンポーネントのいずれかが使用されていれば、ソフトウェアで自動的に、適切にコンフィギュレーションされた RAMB16BWE エLEMENTに変更されます。

ポート名	方向	幅	機能
DOA、DOB	出力	32	ポート A/B のデータ出力バス
DOPA、DOPB	出力	4	ポート A/B のパリティ出力バス
DIA、DIB	入力	32	ポート A/B のデータ入力バス
DIPA、DIPB	入力	4	ポート A/B のパリティ入力バス
ADDRA、ADDRB	入力	14	ポート A/B のアドレス入力バス。MSB は常に ADDRA/B[13] ですが、LSB は DATA_WIDTH_A/B の設定によって決まります。
WEA、WEB	入力	4	ポート A/B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル
REGCEA、REGCEB	入力	1	出力レジスタ クロック イネーブル
RSTA、RSTB	入力	1	ポート A/B の出力レジスタのセット/リセット。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

インスタンスエーション	可
推論	推奨
CORE Generator™ およびウィザード	可
マクロのサポート	不可

必要な入力はすべて適切な信号に接続してください。CLKA/CLKB クロック信号をアクティブ クロックに、RSTA/RSTB リセット信号を論理値 0 または適切なリセット信号に接続する必要があります。ENA/ENB は、論理値 1 または適切な RAM ポート イネーブル信号のいずれかに接続します。REGCEA および REGCEB は、対応する DOA_REG または DOB_REG 属性が 1 に設定されている場合、適切な出力レジスタのクロック イネーブルまたは論理値 1 に接続する必要があります。DOA_REG が 0 に設定されている場合は、REGCEA および REGCEB を論理値 0 に設定する必要があります。

これらの信号に必要な接続は DATA_WIDTH の設定により変わるため、上記のポートの表で、必要なデータ入力、データ出力、ライト イネーブル、アドレスの接続情報を確認してください。ほかの出力信号はすべて接続しないままです。使用されていない入力信号は論理値 0 に接続してください。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A	整数	0、1、2、4、9、18、36	0	ポート A のデータ幅を指定
DATA_WIDTH_B	整数	0、1、2、4、9、18、36	0	ポート B のデータ幅を指定
DOA_REG	整数	0、1	0	RAM の出力レジスタを使用するか、バイパス接続するか指定します。
DOB_REG	整数	0、1	0	RAM の出力レジスタを使用するか、バイパス接続するか指定します。
INIT_A	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	コンフィギュレーション後のポート A の出力の初期値を指定します。
INIT_B	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
INIT_FILE	文字列	0 ビット文字列	NONE	ブロック RAM の初期値を指定するためのファイル名を指定します。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期内容を指定します。
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ メモリ アレイの初期内容を指定します。
RSTTYPE	文字列	SYNC、ASYN	SYNC	リセット、同期、または非同期を指定します。
SIM_COLLISION_CHECK	文字列	ALL、 GENERATE_X_ONLY、 WARNING_ONLY、 NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>
SRVAL_A	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	リセット信号 (RSTA) がアサートされたときのポート A の出力値を指定
SRVAL_B	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	リセット信号 (RSTB) がアサートされたときのポート B の出力値を指定します。

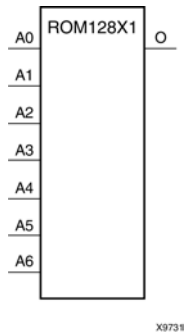
属性	タイプ	値	デフォルト	説明
WRITE_MODE_A	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	ポート A の出力ビヘイビアを指定します。 <ul style="list-style-type: none">WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。READ_FIRST に設定すると、そのメモリロケーションに直前に格納されていた値が出力ポートに出力されます。NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	ポート B の出力ビヘイビアを指定します。 <ul style="list-style-type: none">WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。READ_FIRST に設定すると、そのメモリロケーションに直前に格納されていた値が出力ポートに出力されます。NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリ データシート](#)

ROM128X1

プリミティブ : 128-Deep by 1-Wide ROM



X9731

概要

このデザイン エLEMENTは 128 ワード X 1 ビットの ROM です。データ出力 (O) には、7 ビットのアドレス (A6 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT=value で指定した値に初期化されます。初期値は 32 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT=value を指定しないと、エラーになります。

論理表

入力				出力
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)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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"00000000000000000000000000000000"
)
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 記述 (インスタンス化)

```
// 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'h00000000000000000000000000000000) // Contents of ROM
) 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]
);

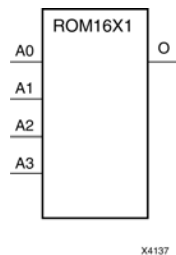
// End of ROM128X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

ROM16X1

プリミティブ : 16-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 16 ワード X 1 ビットの ROM です。データ出力 (O) には、4 ビットのアドレス (A3 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT=value で指定した値に初期化されます。初期値は 4 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。たとえば、INIT=10A7 と指定すると、「0001 0000 1010 0111」というデータストリームが生成されます。INIT=value を指定しないと、エラーになります。

論理表

入力				出力
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)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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]
);

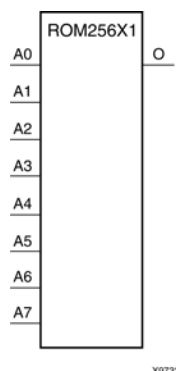
// End of ROM16X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

ROM256X1

プリミティブ : 256-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 256 ワード X 1 ビットの ROM です。データ出力 (O) には、8 ビットのアドレス (A7 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT=value で指定した値に初期化されます。初期値は 64 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。

INIT=value を指定しないと、エラーになります。

論理表

入力				出力
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)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- ROM256X1: 256 x 1 Asynchronous Distributed => LUT ROM
--           Virtex-4/5/6, Spartan-3/3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

ROM256X1_inst : ROM256X1
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]
  A7 => A7    -- ROM address[7]
);

-- End of ROM256X1_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ROM256X1: 256 x 1 Asynchronous Distributed (LUT) ROM
//           Virtex-4/5/6, Spartan-3/3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

ROM256X1 #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000) // Contents of ROM
) ROM256X1_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]
  .A7(A7)   // ROM address[7]
);

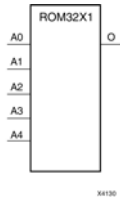
// End of ROM256X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

ROM32X1

プリミティブ : 32-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 32 ワード X 1 ビットの ROM です。データ出力 (O) には、5 ビットのアドレス (A4 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT=value で指定した値に初期化されます。初期値は 8 桁の 16 進数で、ROM には最上位ビット A=1FH から最下位ビット A=00H の順に書き込まれます。

たとえば、INIT=10A78F39 と指定すると、次のデータストリームが生成されます。0001 0000 1010 0111 1000 1111 0011 1001. INIT=value を指定しないと、エラーになります。

論理表

入力				出力
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)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// ROM32X1: 32 x 1 Asynchronous Distributed (LUT) ROM
//      All 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]
);

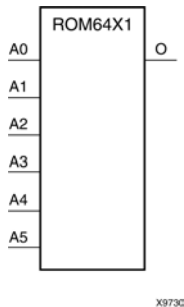
// End of ROM32X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

ROM64X1

プリミティブ : 64-Deep by 1-Wide ROM



概要

このデザイン エLEMENTは 64 ワード X 1 ビットの ROM です。データ出力 (O) には、6 ビットのアドレス (A5 ~ A0) で選択されたワードが出力されます。ROM は、コンフィギュレーションの際に INIT=value で指定した値に初期化されます。初期値は 16 桁の 16 進数で、ROM には最上位ビット A=FH から最下位ビット A=0H の順に書き込まれます。INIT=value を指定しないと、エラーになります。

論理表

入力				出力
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)

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ROM の値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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"00000000000000000000"
)
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 記述 (インスタンス化)

```
// 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'h00000000000000000000) // 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]
);

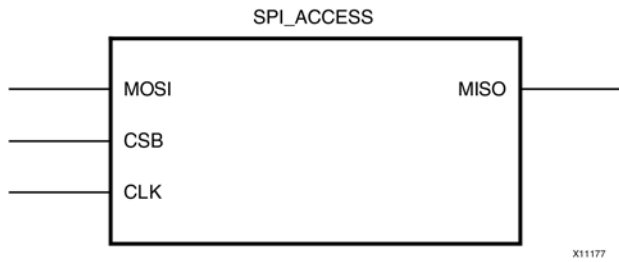
// End of ROM64X1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SPI_ACCESS

プリミティブ : Internal Logic Access to the Serial Peripheral Interface (SPI) PROM Data



概要

このデザイン エレメントを使用すると、FPGA の内部ロジックから SPI シリアル プロトコルを介して、デバイスに含まれているインシステム フラッシュ (ISF) メモリに接続できます。

ポートの説明

ポート名	方向	幅	機能
MISO	出力	1	ISF メモリのシリアル出力データ
MOSI	入力	1	ISF メモリへのシリアル入力命令/データ
CSB	入力	1	ISF メモリ イネーブル
CLK	入力	1	ISF メモリ クロック

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
SIM_DEVICE	文字列	3S50AN、3S200AN、3S400AN、3S700AN、3S1400AN	UNSPECIFIED	適切なサイズの ISF メモリが使用されるようにターゲット デバイスを指定します。この属性は必ず設定する必要があります。
SIM_USER_ID	16 進数	64 ビット値	0xFF	ISF メモリに対しプログラムされている、セキュリティレジスタ内のユーザー ID を指定します。
SIM_MEM_FILE	文字列	指定されたファイル名およびディレクトリ名	NONE	ISF メモリの初期化メモリの内容を含む HEX ファイルをオプションで指定します。
SIM_FACTORY_ID	16 進数	64 ビット値	0xFF	シミュレーションに使用する、セキュリティレジスタ内の重複しない識別子の値を指定 (実際の HW 値はデバイスに特定の値)
SIM_DELAY_TYPE	文字列	ACCURATE、SCALED	SCALED	シミュレーションの速度を上げるため、一部のタイミング遅延を減少させます。

属性	タイプ	値	デフォルト	説明
				<ul style="list-style-type: none"> ・ ACCURATE を指定した場合、タイミングと遅延はデータシートの仕様どおりです。 ・ SCALED を指定すると、シミュレーションの速度を上げるためにタイミング値を減少させますが、ビヘイビアには影響はありません。

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

[illegible]

Verilog 記述 (インスタンス化)

```
// SPI_ACCESS: Internal Logic Access to the Serial Peripheral
//           Interface (SPI) PROM Data
//           Spartan-3AN
// Xilinx HDL Libraries Guide, version 11.2

SPI_ACCESS #(
    .SIM_DELAY_TYPE("SCALED"), // "ACCURATE" spec timing delays, "SCALED" shorten delays (faster sim)
    .SIM_DEVICE("3S1400AN"),   // "3S50AN", "3S200AN", "3S400AN", "3S700AN", "3S1400AN"
    .SIM_FACTORY_ID(64'h0),     // Specifies the Pre-programmed factory ID value
    .SIM_MEM_FILE("NONE"),      // Name/location of file containing memory contents
    .SIM_USER_ID(64'h0)         // Specifies the programmed User ID value
) SPI_ACCESS_inst (
    .MISO(MISO), // Serial output data from SPI PROM
    .CLK(CLK),   // SPI PROM clock input
    .CSB(CSB),   // SPI PROM enable input
    .MOSI(MOSI)  // Serial input data to SPI PROM
);

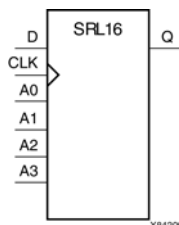
// End of SPI_ACCESS_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

SRL16

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT)



概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わると、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。

論理表

入力			出力
Am	CLK	D	Q
Am	X	X	Q(Am)
Am	↑	D	Q(Am - 1)
m = 0、1、2、3			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後の Q 出力の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

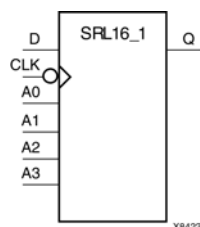
// End of SRL16_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SRL16_1

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock



概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

クロック (CLK) が High から Low に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが High から Low に切り替わると、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。

論理表

入力			出力
A _m	CLK	D	Q
A _m	X	X	Q(A _m)
A _m	↓	D	Q(A _m - 1)
m = 0、1、2、3			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後の Q 出力の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

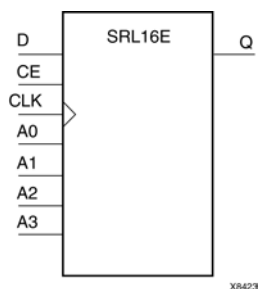
// End of SRL16_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SRL16E

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。CE が Low の場合、クロック遷移は無視されます。

論理表

入力				出力
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↑	D	Q(Am - 1)
m = 0、1、2、3				

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	4	SRL のワード数のダイナミック選択 <ul style="list-style-type: none"> ・ A=0000 ==> 1 ビット シフト長 ・ A=1111 ==> 16 ビット シフト長

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

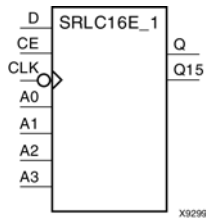
// End of SRL16E_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SRL16E_1

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Negative-Edge Clock and Clock Enable



概要

このデザイン エLEMENTは、クロック イネーブル (CE) があるシフトレジスタ ルックアップ テーブル (LUT) で、クロックの立ち下がりエッジで動作します。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が High から Low に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが High から Low に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。CE が Low の場合、クロック遷移は無視されます。

論理表

入力				出力
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↓	D	Q(Am - 1)
m = 0、1、2、3				

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

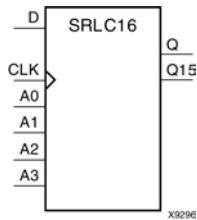
// End of SRL16E_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SRLC16

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry



概要

このデザイン エLEMENTは、キャリーがあるシフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わると、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。

メモ : Q15 の出力を使用すると、複数のシフトレジスタ LUT をカスケード接続でき、より大きなシフトレジスタを作成できます。

論理表

入力			出力
Am	CLK	D	Q
Am	X	X	Q(Am)
Am	↑	D	Q(Am - 1)
m = 0、1、2、3			

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
--          Spartan-3/3E/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 記述 (インスタンス化)

```
// SRLC16: 16-bit cascadable shift register LUT operating on posedge of clock
//          Virtex-4, Spartan-3/3E/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
);

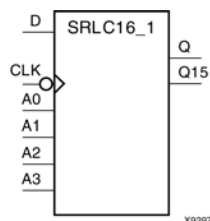
// End of SRLC16_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

SRLC16_1

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry and Negative-Edge Clock



概要

このデザイン エLEMENTは、キャリーがあるシフトレジスタ ルックアップ テーブル (LUT) で、クロックの立ち下がりエッジで動作します。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

メモ : Q15 の出力を使用すると、複数のシフトレジスタ LUT をカスケード接続でき、より大きなシフトレジスタを作成できます。

論理表

入力			出力	
A _m	CLK	D	Q	Q15
A _m	X	X	Q(A _m)	変化なし
A _m	↓	D	Q(A _m - 1)	Q14
m = 0、1、2、3				

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンスレーション)

```
// 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
);

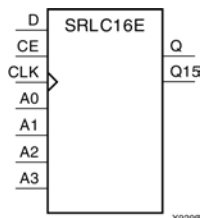
// End of SRLC16_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SRLC16E

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry and Clock Enable



概要

このデザイン エLEMENTは、キャリーとクロック イネーブルがあるシフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しいデータがロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。

メモ : Q15 の出力を使用すると、複数のシフトレジスタ LUT をカスケード接続でき、より大きなシフトレジスタを作成できます。

論理表

入力				出力	
A _m	CLK	CE	D	Q	Q15
A _m	X	0	X	Q(A _m)	Q(15)
A _m	X	1	X	Q(A _m)	Q(15)
A _m	↑	1	D	Q(A _m - 1)	Q15
m = 0、1、2、3					

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// SRLC16E: 16-bit cascable 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
);

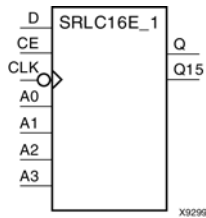
// End of SRLC16E_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

SRLC16E_1

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Carry, Negative-Edge Clock, and Clock Enable



概要

このデザイン エLEMENTは、キャリーおよびクロック イネーブルがあるシフトレジスタ ルックアップ テーブル (LUT) で、クロックの立ち下がりエッジで動作します。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ = $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$ という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が High から Low に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが High から Low に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。

メモ : Q15 の出力を使用すると、複数のシフトレジスタ LUT をカスケード接続でき、より大きなシフトレジスタを作成できます。

論理表

入力				出力	
A _m	CE	CLK	D	Q	Q15
A _m	0	X	X	Q(A _m)	変化なし
A _m	1	X	X	Q(A _m)	変化なし
A _m	1	↓	D	Q(A _m -1)	Q14
m = 0、1、2、3					

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

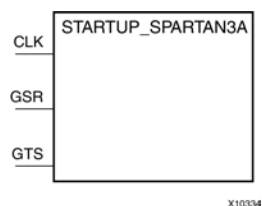
// End of SRLC16E_1_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリ データシート](#)

STARTUP_SPARTAN3A

プリミティブ : Spartan®-3A Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface



概要

このデザイン エLEMENTは、デバイス ピンおよびグローバル セット/リセット (GSR) 信号へのロジックにインターフェイスするため、または専用配線のために使用します。また、デバイスのコンフィギュレーションの終了時に、スタートアップ シーケンスに別のクロックを指定することもできます。

ポートの説明

ポート名	方向	幅	機能
GSR	入力	1	グローバル セット /リセット (GSR) 配線への入力接続
GTS	入力	1	グローバル トライステート (GTS) 配線への入力接続
CLK	入力	1	コンフィギュレーション スタートアップ シーケンス クロック (GSR) 配線への入力接続

デザインの入力方法

インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

専用の GSR 回路はソース ピンまたはロジックを GSR ピンに接続すれば使用できますが、このコンポーネントの GSR 回路の使用には、特別の注意が必要です。GSR ネットのスキューは確定できないので、セット/リセット信号の一般配線を使用して、配線遅延とスキューをデザインのタイミング解析の一部として計算できるようにするか、クロック サイクルのリリース時にスキューが回路の動作を邪魔しないような回避策をとってください。

同様に、専用のグローバル トライステートが使用される場合は、適切なソース ピンまたはロジックをこのプリミティブの GTS 入力ピンに接続します。コンフィギュレーションのスタートアップ シーケンスのクロックを指定するには、デザインからのクロックをこのデザイン エLEMENTの CLK ピンに接続します。

VHDL 記述 (インスタンスレーション)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- STARTUP_SPARTAN3A: Startup primitive for GSR, GTS or startup sequence
--                      control.
--                      Spartan-3A
-- Xilinx HDL Libraries Guide, version 11.2

STARTUP_SPARTAN3A_inst : STARTUP_SPARTAN3A
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)
);

-- End of STARTUP_SPARTAN3A_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// STARTUP_SPARTAN3A: Startup primitive for GSR, GTS or startup sequence
//                      control.
//                      Spartan-3A
// Xilinx HDL Libraries Guide, version 11.2

STARTUP_SPARTAN3A STARTUP_SPARTAN3A_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)
);

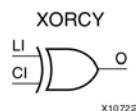
// End of STARTUP_SPARTAN3A_inst instantiation
```

詳細情報

- [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- [Spartan-3A FPGA ファミリー データシート](#)

XORCY

プリミティブ : XOR for Carry Logic with General Output



概要

このデザイン エLEMENTは、一般出力 (O) のある特殊な XOR ゲートで、高速で小型の演算ファンクションを生成するために使用します。このプリミティブは、スライスのキャリーチェーン ロジック内の専用 XOR ファンクションで、演算ファンクション (加算または除算) または多入力ロジック ファンクション (多入力 AND または OR ゲート) を高速かつ効率的に作成できます。

論理表

入力		出力
LI	CI	O
0	0	0
0	1	1
1	0	1
1	1	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

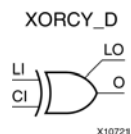
// End of XORCY_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

XORCY_D

プリミティブ : XOR for Carry Logic with Dual Output



概要

このデザイン エLEMENTは、一般出力 (O) とローカル出力 (LO) のある特殊な XOR ゲートで、高速で小型の演算ファンクションを生成するために使用します。

論理表

入力		出力
LI	CI	O および LO
0	0	0
0	1	1
1	0	1
1	1	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

```
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 記述 (インスタンス化)

```
// 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
);

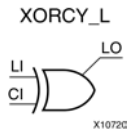
// End of XORCY_D_inst instantiation
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)

XORCY_L

プリミティブ : XOR for Carry Logic with Local Output



概要

このデザイン エLEMENTは、ローカル出力 (LO) のある特殊な XOR ゲートで、高速で小型の演算ファンクションを生成するために使用します。

論理表

入力		出力
LI	CI	LO
0	0	0
0	1	1
1	0	1
1	1	0

デザインの入力方法

インスタンス化	可
推論	推奨
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

VHDL 記述 (インスタンス化)

次の 2 つの構文が存在しない場合はコピーし、エンティティ宣言の前に貼り付けます。

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

-- XORCY_L: Carry-Chain XOR-gate with local  => direct-connect output
-- 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 記述 (インスタンス化)

```
// 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
```

詳細情報

- ・ [Spartan-3 ジェネレーション FPGA ユーザー ガイド](#)
- ・ [Spartan-3A FPGA ファミリー データシート](#)