

Virtex-4 ライブラリ ガイド (HDL 用)

UG619 (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. The PowerPC name and logo are registered trademarks of IBM Corp., and used under license. All other trademarks are the property of their respective owners.

目次

ザイリン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	15
デザイン エLEMENT	15
デザインの入力方法	16
2 : ファンクション別分類	17
3 : デザイン エLEMENT	23
BSCAN_VIRTEX4	24
概要	24
ポートの説明	24
デザインの入力方法	25
使用可能な属性	25
詳細情報	26
BUF	27
概要	27
デザインの入力方法	27
詳細情報	27
BUFCF	28
概要	28
デザインの入力方法	28
詳細情報	29
BUFG	30
概要	30
ポートの説明	30
デザインの入力方法	30
詳細情報	31
BUFGCE	32
概要	32
論理表	32
デザインの入力方法	32
詳細情報	33
BUFGCE_1	34
概要	34
論理表	34
デザインの入力方法	34
詳細情報	35
BUFGCTRL	36
概要	36
ポートの説明	36
デザインの入力方法	37
使用可能な属性	37
詳細情報	37
BUFGMUX	38
概要	38
論理表	38
ポートの説明	38
デザインの入力方法	38
使用可能な属性	39
詳細情報	39
BUFGMUX_1	40
概要	40
論理表	40
デザインの入力方法	40
詳細情報	41
BUFGMUX_VIRTEX4	42
概要	42

ポートの説明	42
デザインの入力方法	42
詳細情報	43
BUFIO	44
概要	44
ポートの説明	44
デザインの入力方法	44
詳細情報	45
BUFR	46
概要	46
ポートの説明	46
デザインの入力方法	46
使用可能な属性	46
詳細情報	47
CAPTURE_VIRTEX4	48
概要	48
ポートの説明	48
デザインの入力方法	48
使用可能な属性	48
詳細情報	49
DCIRESET	50
概要	50
ポートの説明	50
デザインの入力方法	50
使用可能な属性	50
詳細情報	51
DCM_ADV	52
概要	52
ポートの説明	52
デザインの入力方法	55
使用可能な属性	56
詳細情報	58
DCM_BASE	59
概要	59
ポートの説明	59
デザインの入力方法	61
使用可能な属性	61
詳細情報	63
DCM_PS	64
概要	64
ポートの説明	64
デザインの入力方法	66
使用可能な属性	67
詳細情報	69
DSP48	70
概要	70
ポートの説明	71
デザインの入力方法	74
使用可能な属性	74
詳細情報	76
EMAC	77
概要	77
ポートの説明	77
デザインの入力方法	80
詳細情報	80
FDCE	81
概要	81
論理表	81
デザインの入力方法	81

使用可能な属性.....	81
詳細情報	82
FDCE_1.....	83
概要.....	83
論理表.....	83
デザインの入力方法.....	83
使用可能な属性.....	83
詳細情報	84
FDCPE.....	85
概要.....	85
論理表.....	85
ポートの説明	85
デザインの入力方法.....	86
使用可能な属性.....	86
詳細情報	87
FDCPE_1.....	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
FDRSE	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
FDRSE_1	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
FIFO16	95
概要.....	95
ポートの説明	97
デザインの入力方法.....	97
使用可能な属性.....	98
詳細情報	99
FRAME_ECC_VIRTEX4.....	100
概要.....	100
ポートの説明	100
デザインの入力方法.....	100
SYNDROME の値とエラー ステータスの関係	100
詳細情報	101
GT11_CUSTOM.....	102
概要.....	102
論理表.....	102
デザインの入力方法.....	104
詳細情報	104
GT11_DUAL.....	105
概要.....	105
論理表.....	105
デザインの入力方法.....	109
詳細情報	109
GT11CLK.....	110
概要.....	110
ポートの説明	110

デザインの入力方法.....	110
詳細情報	110
GT11CLK_MGT.....	111
概要.....	111
ポートの説明	111
デザインの入力方法.....	111
詳細情報	111
IBUFDS.....	112
概要.....	112
論理表.....	112
ポートの説明	112
デザインの入力方法.....	112
使用可能な属性.....	113
詳細情報	114
IBUFG	115
概要.....	115
ポートの説明	115
デザインの入力方法.....	115
使用可能な属性.....	115
詳細情報	116
IBUFGDS	117
概要.....	117
論理表.....	117
ポートの説明	117
デザインの入力方法.....	117
使用可能な属性.....	118
詳細情報	118
ICAP_VIRTEX4.....	119
概要.....	119
ポートの説明	119
デザインの入力方法.....	119
使用可能な属性.....	119
詳細情報	120
IDDR	121
概要.....	121
ポートの説明	121
デザインの入力方法.....	122
使用可能な属性.....	122
詳細情報	123
IDELAY.....	124
概要.....	124
ポートの説明	124
デザインの入力方法.....	126
使用可能な属性.....	126
詳細情報	127
IDELAYCTRL	128
概要.....	128
ポートの説明	128
デザインの入力方法.....	128
詳細情報	129
IOBUF	130
概要.....	130
論理表.....	130
ポートの説明	130
デザインの入力方法.....	130
使用可能な属性.....	131
詳細情報	132
IOBUFDS	133
概要.....	133

論理表.....	133
ポートの説明	133
デザインの入力方法.....	133
使用可能な属性.....	134
詳細情報	134
ISERDES.....	135
概要.....	135
ポートの説明	137
デザインの入力方法.....	139
使用可能な属性.....	139
詳細情報	141
KEEPER.....	142
概要.....	142
ポートの説明	142
デザインの入力方法.....	142
詳細情報	143
LDCPE.....	144
概要.....	144
論理表.....	144
ポートの説明	144
デザインの入力方法.....	145
使用可能な属性.....	145
詳細情報	146
LUT1.....	147
概要.....	147
論理表.....	147
デザインの入力方法.....	147
使用可能な属性.....	147
詳細情報	148
LUT1_D.....	149
概要.....	149
論理表.....	149
デザインの入力方法.....	149
使用可能な属性.....	149
詳細情報	150
LUT1_L.....	151
概要.....	151
論理表.....	151
デザインの入力方法.....	151
使用可能な属性.....	151
詳細情報	152
LUT2.....	153
概要.....	153
論理表.....	153
デザインの入力方法.....	153
使用可能な属性.....	154
詳細情報	154
LUT2_D.....	155
概要.....	155
論理表.....	155
デザインの入力方法.....	155
使用可能な属性.....	156
詳細情報	156
LUT2_L.....	157
概要.....	157
論理表.....	157
デザインの入力方法.....	157
使用可能な属性.....	158
詳細情報	158

LUT3	159
概要	159
論理表	159
デザインの入力方法	160
使用可能な属性	160
詳細情報	160
LUT3_D	161
概要	161
論理表	161
デザインの入力方法	162
使用可能な属性	162
詳細情報	162
LUT3_L	163
概要	163
論理表	163
デザインの入力方法	164
使用可能な属性	164
詳細情報	164
LUT4	165
概要	165
論理表	166
デザインの入力方法	166
使用可能な属性	166
詳細情報	167
LUT4_D	168
概要	168
論理表	169
デザインの入力方法	169
使用可能な属性	169
詳細情報	170
LUT4_L	171
概要	171
論理表	172
デザインの入力方法	172
使用可能な属性	172
詳細情報	173
MULT_AND	174
概要	174
論理表	174
デザインの入力方法	174
詳細情報	175
MULT18X18	176
概要	176
論理表	176
デザインの入力方法	176
詳細情報	177
MULT18X18S	178
概要	178
論理表	178
デザインの入力方法	178
詳細情報	179
MUXCY	180
概要	180
論理表	180
デザインの入力方法	180
詳細情報	181
MUXCY_D	182
概要	182
論理表	182

デザインの入力方法.....	182
詳細情報.....	183
MUXCY_L.....	184
概要.....	184
論理表.....	184
デザインの入力方法.....	184
詳細情報.....	185
MUXF5.....	186
概要.....	186
論理表.....	186
デザインの入力方法.....	186
詳細情報.....	187
MUXF5_D.....	188
概要.....	188
論理表.....	188
デザインの入力方法.....	188
詳細情報.....	189
MUXF5_L.....	190
概要.....	190
論理表.....	190
デザインの入力方法.....	190
詳細情報.....	191
MUXF6.....	192
概要.....	192
論理表.....	192
デザインの入力方法.....	192
詳細情報.....	193
MUXF6_D.....	194
概要.....	194
論理表.....	194
デザインの入力方法.....	194
詳細情報.....	195
MUXF6_L.....	196
概要.....	196
論理表.....	196
デザインの入力方法.....	196
詳細情報.....	197
MUXF7.....	198
概要.....	198
論理表.....	198
ポートの説明.....	198
デザインの入力方法.....	198
詳細情報.....	199
MUXF7_D.....	200
概要.....	200
論理表.....	200
ポートの説明.....	200
デザインの入力方法.....	200
詳細情報.....	201
MUXF7_L.....	202
概要.....	202
論理表.....	202
ポートの説明.....	202
デザインの入力方法.....	202
詳細情報.....	203
MUXF8.....	204
概要.....	204
論理表.....	204
ポートの説明.....	204

デザインの入力方法.....	204
詳細情報	205
MUXF8_D	206
概要.....	206
論理表.....	206
ポートの説明	206
デザインの入力方法.....	206
詳細情報	207
MUXF8_L	208
概要.....	208
論理表.....	208
ポートの説明	208
デザインの入力方法.....	208
詳細情報	209
OBUF	210
概要.....	210
ポートの説明	210
デザインの入力方法.....	210
使用可能な属性.....	210
詳細情報	211
OBUFDS	212
概要.....	212
論理表.....	212
ポートの説明	212
デザインの入力方法.....	212
使用可能な属性.....	212
詳細情報	213
OBUFT	214
概要.....	214
論理表.....	214
ポートの説明	214
デザインの入力方法.....	214
使用可能な属性.....	215
詳細情報	215
OBUFTDS	216
概要.....	216
論理表.....	216
ポートの説明	216
デザインの入力方法.....	216
使用可能な属性.....	216
詳細情報	217
ODDR.....	218
概要.....	218
ポートの説明	218
デザインの入力方法.....	218
使用可能な属性.....	219
詳細情報	219
OSERDES	220
概要.....	220
ポートの説明	220
デザインの入力方法.....	221
使用可能な属性.....	223
詳細情報	224
PMCD	225
概要.....	225
ポートの説明	225
デザインの入力方法.....	226
使用可能な属性.....	226
詳細情報	227

PPC405_ADV	228
概要	228
デザインの入力方法	228
詳細情報	228
PULLDOWN	229
概要	229
ポートの説明	229
デザインの入力方法	229
詳細情報	230
PULLUP	231
概要	231
ポートの説明	231
デザインの入力方法	231
詳細情報	232
RAM16X1D	233
概要	233
論理表	233
デザインの入力方法	234
使用可能な属性	234
詳細情報	235
RAM16X1D_1	236
概要	236
論理表	236
ポートの説明	237
デザインの入力方法	237
使用可能な属性	237
詳細情報	238
RAM16X1S	239
概要	239
論理表	239
デザインの入力方法	239
使用可能な属性	239
詳細情報	240
RAM16X1S_1	241
概要	241
論理表	241
デザインの入力方法	241
使用可能な属性	242
詳細情報	242
RAM16X2S	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	245
RAM16X4S	246
概要	246
論理表	246
デザインの入力方法	246
使用可能な属性	247
詳細情報	248
RAM16X8S	249
概要	249
論理表	249
デザインの入力方法	249
使用可能な属性	249
詳細情報	250
RAM32X1S	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
RAM32X1S_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
RAM32X2S	255
概要.....	255
論理表.....	255
デザインの入力方法.....	255
使用可能な属性.....	256
詳細情報	257
RAM32X4S	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	260
RAM32X8S	261
概要.....	261
論理表.....	261
デザインの入力方法.....	261
使用可能な属性.....	262
詳細情報	263
RAM64X1S	264
概要.....	264
論理表.....	264
デザインの入力方法.....	264
使用可能な属性.....	264
詳細情報	265
RAM64X1S_1.....	266
概要.....	266
論理表.....	266
デザインの入力方法.....	266
使用可能な属性.....	266
詳細情報	267
RAM64X2S	268
概要.....	268
論理表.....	268
デザインの入力方法.....	268
使用可能な属性.....	269
詳細情報	270
RAMB16.....	271
概要.....	271
ポートの説明	272
デザインの入力方法.....	273
使用可能な属性.....	273
詳細情報	278
RAMB32_S64_ECC	279
概要.....	279
ポートの説明	279
デザインの入力方法.....	280
使用可能な属性.....	280
詳細情報	281
ROM128X1	282

概要.....	282
論理表.....	282
デザインの入力方法.....	283
使用可能な属性.....	283
詳細情報	284
ROM16X1.....	285
概要.....	285
論理表.....	285
デザインの入力方法.....	286
使用可能な属性.....	286
詳細情報	286
ROM256X1	287
概要.....	287
論理表.....	287
デザインの入力方法.....	288
使用可能な属性.....	288
詳細情報	289
ROM32X1.....	290
概要.....	290
論理表.....	290
デザインの入力方法.....	291
使用可能な属性.....	291
詳細情報	291
ROM64X1.....	292
概要.....	292
論理表.....	292
デザインの入力方法.....	293
使用可能な属性.....	293
詳細情報	294
SRL16	295
概要.....	295
論理表.....	295
デザインの入力方法.....	295
使用可能な属性.....	296
詳細情報	296
SRL16_1	297
概要.....	297
論理表.....	297
デザインの入力方法.....	297
使用可能な属性.....	298
詳細情報	298
SRL16E	299
概要.....	299
論理表.....	299
ポートの説明	300
デザインの入力方法.....	300
使用可能な属性.....	300
詳細情報	301
SRL16E_1.....	302
概要.....	302
論理表.....	302
デザインの入力方法.....	302
使用可能な属性.....	303
詳細情報	303
SRLC16.....	304
概要.....	304
論理表.....	304
デザインの入力方法.....	305
使用可能な属性.....	305

詳細情報	306
SRLC16_1	307
概要	307
論理表	307
デザインの入力方法	307
使用可能な属性	308
詳細情報	308
SRLC16E	309
概要	309
論理表	309
デザインの入力方法	310
使用可能な属性	310
詳細情報	311
SRLC16E_1	312
概要	312
論理表	312
デザインの入力方法	313
使用可能な属性	313
詳細情報	314
STARTUP_VIRTEX4	315
概要	315
ポートの説明	315
デザインの入力方法	315
詳細情報	316
USR_ACCESS_VIRTEX4	317
概要	317
ポートの説明	317
デザインの入力方法	317
詳細情報	318
XORCY	319
概要	319
論理表	319
デザインの入力方法	319
詳細情報	320
XORCY_D	321
概要	321
論理表	321
デザインの入力方法	321
詳細情報	322
XORCY_L	323
概要	323
論理表	323
デザインの入力方法	323
詳細情報	324

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
- ・ 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LEMENT

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LEMENT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LEMENT (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LEMENTが組み込まれます。すべてのデザイン エLEMENT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

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。

アドバンス	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	プロセッサ	
クロック コンポーネント	RAM/ROM	
コンフィギュレーション/BSCAN コンポーネント	レジスタおよびラッチ	
ギガビット I/O	シフト レジスタ	

アドバンス

デザイン エLEMENT	説明
EMAC	プリミティブ : Fully integrated 10/100/1000 Mb/s Ethernet Media Access Controller (Ethernet MAC)

四則演算ファンクション

デザイン エLEMENT	説明
DSP48	プリミティブ : 18x18 Signed Multiplier Followed by a Three-Input Adder with Optional Pipeline Registers
MULT18X18	プリミティブ : 18 x 18 Signed Multiplier
MULT18X18S	プリミティブ : 18 x 18 Signed Multiplier -- Registered Version

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
BUFGCE_1	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable and Output State 1
BUFGCTRL	プリミティブ : Global Clock MUX Buffer
BUFGMUX	コンビニエンス プリミティブ : Global Clock MUX Buffer
BUFGMUX_1	コンビニエンス プリミティブ : Global Clock MUX Buffer with Output State 1
BUFGMUX_VIRTEX4	プリミティブ : Global Clock MUX Buffer
BUFIO	プリミティブ : Local Clock Buffer for I/O
BUFR	プリミティブ : Regional Clock Buffer for I/O and Logic Resources
DCM_ADV	プリミティブ : Advanced Digital Clock Manager Circuit
DCM_BASE	プリミティブ : Base Digital Clock Manager Circuit
DCM_PS	プリミティブ : Digital Clock Manager with Basic and Phase Shift Features
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
PMCD	プリミティブ : Phase-Matched Clock Divider

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_VIRTEX4	プリミティブ : Virtex®-4 JTAG Boundary-Scan Logic Access Circuit
CAPTURE_VIRTEX4	プリミティブ : Virtex®-4 Boundary Scan Logic Control Circuit
FRAME_ECC_VIRTEX4	プリミティブ : Reads a Single, Virtex®-4 Configuration Frame and Computes a Hamming, Single-Error Correction, Double-Error Detection Syndrome
ICAP_VIRTEX4	プリミティブ : Virtex-4 Internal Configuration Access Port
STARTUP_VIRTEX4	プリミティブ : Virtex®-4 User Interface to Configuration Clock, Global Reset, Global 3-State Controls, and Other Configuration Signals
USR_ACCESS_VIRTEX4	プリミティブ : 32-Bit Register with a 32-Bit DATA Bus and a DATAVALID Port

ギガビット I/O

デザイン エLEMENT	説明
GT11_CUSTOM	プリミティブ : RocketIO MGTs with 622 Mb/s to 11.1 Gb/s Data Rates, 8 to 24 Transceivers per FPGA, and 2.5 GHz 5.55 GHz VCO, Less Than 1ns RMS Jitter
GT11_DUAL	プリミティブ : RocketIO MGT Tile (contains 2 GT11_CUSTOM) with 622 Mb/s to 11.1 Gb/s data rates, 8 to 24 transceivers per FPGA, and 2.5 GHz 5.55 GHz VCO, less than 1ns RMS jitter
GT11CLK	プリミティブ : A MUX That Can Select From Differential Package Input Clock, refclk From the Fabric, or rxblk to Drive the Two Vertical Reference Clock Buses for the Column of MGTs
GT11CLK_MGT	プリミティブ : Allows Differential Package Input to Drive the Two Vertical Reference Clock Buses for the Column of MGTs

I/O コンポーネント

デザイン エLEMENT	説明
BUF	プリミティブ : General Purpose Buffer
DCIRESET	プリミティブ : DCI State Machine Reset (After Configuration Has Been Completed)
IBUFDS	プリミティブ : Differential Signaling Input 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
IDELAY	プリミティブ : Input Delay Element
IDELAYCTRL	プリミティブ : IDELAY Tap Delay Value Control
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
ISERDES	プリミティブ : Dedicated I/O Buffer Input Deserializer
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
OBUFT	プリミティブ : 3-State Output Buffer with Active Low Output Enable
OBUFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
OSERDES	プリミティブ : Dedicated IOB Output Serializer
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

プロセッサ

デザイン エLEMENT	説明
PPC405_ADV	プリミティブ : Primitive for the Power PC Core

RAM/ROM

デザイン エLEMENT	説明
FIFO16	プリミティブ : Virtex-4 Block RAM Based, Built-In FIFO
RAM16X1D	プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM
RAM16X1D_1	プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock
RAM16X1S	プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM
RAM16X1S_1	プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM16X2S	プリミティブ : 16-Deep by 2-Wide Static Synchronous RAM
RAM16X4S	プリミティブ : 16-Deep by 4-Wide Static Synchronous RAM
RAM16X8S	プリミティブ : 16-Deep by 8-Wide Static Synchronous RAM
RAM32X1S	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM
RAM32X1S_1	プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM32X2S	プリミティブ : 32-Deep by 2-Wide Static Synchronous RAM
RAM32X4S	プリミティブ : 32-Deep by 4-Wide Static Synchronous RAM
RAM32X8S	プリミティブ : 32-Deep by 8-Wide Static Synchronous RAM
RAM64X1S	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM
RAM64X1S_1	プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock
RAM64X2S	プリミティブ : 64-Deep by 2-Wide Static Synchronous RAM
RAMB16	プリミティブ : 16K-bit Data and 2K-bit Parity Single-Port Synchronous Block RAM with Configurable Port Widths
RAMB32_S64_ECC	プリミティブ : 512 Deep by 64-Bit Wide Synchronous, Two-Port Block RAM with Built-In Error Correction
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
FDCE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Clear
FDCE_1	プリミティブ : D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Clear
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
IDDR	プリミティブ : Input Dual Data-Rate Register
LDCPE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Preset and Gate Enable
ODDR	プリミティブ : Dedicated Dual Data Rate (DDR) Output Register

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

デザイン エLEMENT

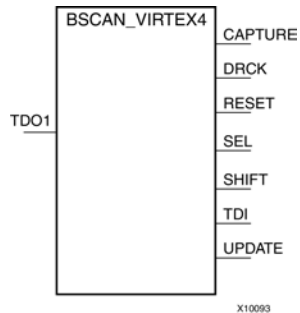
このセクションでは、このアーキテクチャで使えるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

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_VIRTEX4

プリミティブ : Virtex®-4 JTAG Boundary-Scan Logic Access Circuit



概要

このデザイン エLEMENT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。

このデザイン エLEMENTの各インスタンスでは、JTAG_CHAIN 属性の設定に従い、JTAG USER 命令 1 つ (USER1 から USER4 まで) が処理されます。USER 命令の 4 つすべてを処理するには、ELEMENTを 4 つインスタンス化して JTAG_CHAIN 属性を設定します。

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	タイプ	幅	機能
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。
DRCK	出力	1	JTAG_CHAIN によって割り当てられた 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 ステートまたは CAPTURE-DR ステートになると TCK ピンと同じ値を出力します。
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。
SEL	出力	1	JTAG 命令レジスタに USER 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。
TDI	出力	1	TDI ピンと同じ値を出力します。
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。
TDO	入力	1	USER 命令が読み込まれるとアクティブになり、外部 JTAG TDO ピンには、マクロの TDO1 ピンへのデータ入力の値が示されます。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
JTAG_CHAIN	整数	1、2、3、4	1	エレメントのインスタンスで処理可能な JTAG USER 命令数を設定します。

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;

-- BSCAN_VIRTEX4: Boundary Scan primitive for connecting internal logic to
--                JTAG interface. Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

BSCAN_VIRTEX4_inst : BSCAN_VIRTEX4
generic map (
    JTAG_CHAIN => 1) -- Value to set BSCAN site of device. Possible values: (1,2,3 or 4)
port map (
    CAPTURE => CAPTURE, -- CAPTURE output from TAP controller
    DRCK => DRCK,       -- Data register output for USER functions
    RESET => RESET,     -- Reset output from TAP controller
    SEL => SEL,         -- USER active output
    SHIFT => SHIFT,     -- SHIFT output from TAP controller
    TDI => TDI,         -- TDI output from TAP controller
    UPDATE => UPDATE,   -- UPDATE output from TAP controller
    TDO => TDO          -- Data input for USER function
);

-- End of BSCAN_VIRTEX4_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// BSCAN_VIRTEX4: Boundary Scan primitive for connecting internal logic to
//                JTAG interface.
//                Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

BSCAN_VIRTEX4 #(
    .JTAG_CHAIN(1) // Possible values: 1, 2, 3, or 4
) BSCAN_VIRTEX4_inst (
    .CAPTURE(CAPTURE), // CAPTURE output from TAP controller
    .DRCK(DRCK),       // Data register output for USER function
    .RESET(RESET),     // Reset output from TAP controller
    .SEL(SEL),         // USER active output
    .SHIFT(SHIFT),     // SHIFT output from TAP controller
    .TDI(TDI),         // TDI output from TAP controller
    .UPDATE(UPDATE),   // UPDATE output from TAP controller
    .TDO(TDO)          // Data input for USER function
);

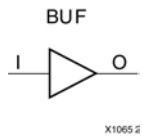
// End of BSCAN_VIRTEX4_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

BUF

プリミティブ : General Purpose Buffer



概要

このデザイン エLEMENTは、汎用の非反転バッファです。

このELEMENTは不要なので、MAP によって削除されます。

デザインの入力方法

このELEMENTは、回路図でのみ使用できます。

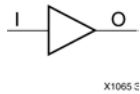
詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

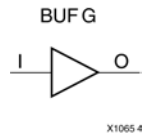
// End of BUFCF_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFG

プリミティブ : Global Clock Buffer



概要

このデザイン エレメントは、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
);

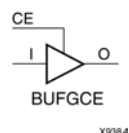
// End of BUFG_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

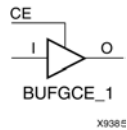
// End of BUFGCE_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGCE_1

プリミティブ : Global Clock Buffer with Clock Enable and Output State 1



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 High (1) 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	1
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_1: Global Clock Buffer with Clock Enable (active low)
--           Virtex-4/5/6, Spartan-3/3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

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

-- End of BUFGCE_1_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

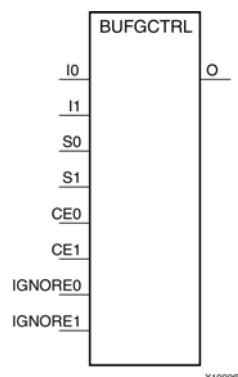
// End of BUFGCE_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGCTRL

プリミティブ : Global Clock MUX Buffer



概要

BUFGCTRL は、2 つのクロック入力を持つ同期/非同期のグリッチのない 2:1 マルチプレクサとして機能するグローバル クロック バッファです。Virtex-4 以前の FPGA に含まれるグローバル クロック バッファに比べ、制御ピンが追加されており、さまざまな機能の使用および効率的な入力の切り替えが可能です。BUFGCTRL は、クロック供給以外の用途にも使用できます。

ポートの説明

ポート名	タイプ	幅	機能
O	出力	1	クロック出力ピン
I	入力	1	クロック入力 : I0 : クロック入力ピン I1 : クロック入力ピン
CE0、CE1	入力	1 (それぞれ)	クロック イネーブル入力。CE ピンは、各クロック入力ピンのクロック イネーブル入力で、クロック入力を選択するときに使用します。入力を選択するために CE ピンを使用する場合は、セットアップ/ホールド タイムを設定する必要があります。要件を満たさない場合、クロックでグリッチが発生する可能性があります。
S0、S1	入力	1 (それぞれ)	クロック セレクト入力。S ピンは、各クロック入力ピンのクロック セレクト入力です。入力を選択するために S ピンを使用する場合は、セットアップおよびホールド タイム要件を満たす必要があります。CE ピンとは異なり、要件を満たさなくてもクロック グリッチが発生することはありませんが、出力クロックがピンに現れるのが 1 クロック サイクル後になる場合があります。
IGNORE0、IGNORE1	入力	1 (それぞれ)	クロック IGNORE 入力。IGNORE ピンは、BUFGCTRL により実行されるスイッチ アルゴリズムをバイパスする場合に使用します。

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
INIT_OUT	整数	0、1	0	コンフィギュレーション後の BUFGCTRL 出力の初期値を指定
PRESELECT_I0	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、コンフィギュレーション後に I0 入力が出力されます。
PRESELECT_I1	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、コンフィギュレーション後に I1 入力が出力されます。

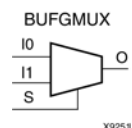
メモ : 2 つの PRESELECT 属性を同時に TRUE にすることはできません。

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

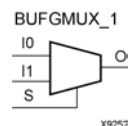
// End of BUFGMUX_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGMUX_1

プリミティブ : Global Clock MUX Buffer with Output State 1



概要

このデザイン エLEMENT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。

このデザイン エLEMENTと BUFGMUX 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。

論理表

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

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

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

-- End of BUFGMUX_1_inst instantiation
```


Verilog 記述 (インスタンス化)

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

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

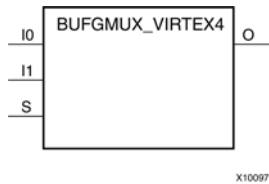
// End of BUFGMUX_1_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGMUX_VIRTEX4

プリミティブ : Global Clock MUX Buffer



概要

このデザイン エレメントは、2 つのクロック入力、1 つのクロック出力、セレクト入力を持つクロック バッファです。このプリミティブは BUFGCTRL に基づいており、一部のピンが High または Low に接続されています。

このエレメントは、S ピンをセレクト ピンとして使用します。S ピンは、グリッチを発生させずにいつでも切り替えることができます。S ピンのセットアップ/ホールド タイムによって、新しいクロックに切り替える前に、その前のクロックの余分なパルスが出力されるかどうかが決まります。S ピンがセットアップ タイム TBCCCK_S より前、I/O が High から Low に切り替わる前に変化した場合、I/O の余分なパルスは出力されません。S ピンがホールド タイムの後に変化した場合は、余分なパルスが出力されますが、グリッチが発生することはありません。いずれの場合でも、出力は遅い方のクロックの 3 クロック サイクル以内に新しいクロックに切り替わります。

S0 と S1 のセットアップ/ホールド タイム要件は、CE0 と CE1 のように立ち上がりエッジではなく、立ち下がりエッジを基準としています (INIT_OUT = 0 の場合)。

このエレメントの切り替え条件は、BUFGCTRL の S ピンと同様です。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	クロック出力
I1 : I0	入力	1	クロック入力
S0 : S1	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

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;

-- BUFGMUX_VIRTEX4: Global Clock Buffer 2-to-1 MUX
--                Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

BUFGMUX_VIRTEX4_inst : BUFGMUX_VIRTEX4
port map (
    O => O,      -- Clock MUX output
    I0 => I0,     -- Clock0 input
    I1 => I1,     -- Clock1 input
    S => S       -- Clock select input
);

-- End of BUFGMUX_VIRTEX4_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFGMUX_VIRTEX4: Global Clock Buffer 2-to-1 MUX
//                Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

BUFGMUX_VIRTEX4 BUFGMUX_VIRTEX4_inst (
    .O(O),       // Clock MUX output
    .I0(I0),     // Clock0 input
    .I1(I1),     // Clock1 input
    .S(S)        // Clock select input
);

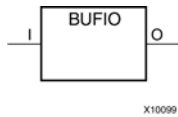
// End of BUFGMUX_VIRTEX4_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFIO

プリミティブ : Local Clock Buffer for I/O



概要

このデザイン エLEMENTはクロック バッファです。単にクロック信号を入力し、出力します。I/O 列の専用クロック ネットを駆動し、グローバル クロック リソースからは独立しているため、ソース同期データ キャプチャ (転送/受信クロック分配) に適しています。これらのELEMENTを駆動できるのは、同じクロック領域内のクロック信号を処理できる I/O のみです。これらのELEMENTでは、隣接する 2 つの I/O クロック ネット (最大 3 クロック領域まで) とリージョナル クロック バッファ (BUFR) を駆動できます。ただし、I/O クロック ネットワークの範囲は I/O 列までなので、CLB やブロック RAM などのロジック リソースは駆動できません。

ポートの説明

ポート名	タイプ	幅	機能
O	出力	1	クロック出力
I	入力	1	クロック入力

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

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;

-- BUFIO: Clock in, clock out buffer
--      Virtex-4/5/6
-- Xilinx HDL Libraries Guide, version 11.2

BUFIO_inst : BUFIO
port map (
  O => O,      -- Clock buffer output
  I => I        -- Clock buffer input
);

-- End of BUFIO_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// BUFIO: Local Clock Buffer
//      Virtex-4/5/6
// Xilinx HDL Libraries Guide, version 11.2

BUFIO BUFIO_inst (
    .O(O),      // Clock buffer output
    .I(I)       // Clock buffer input
);

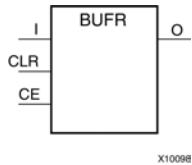
// End of BUFIO_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFR

プリミティブ : Regional Clock Buffer for I/O and Logic Resources



概要

BUFR はクロック バッファです。グローバル クロック ツリーからは独立しており、クロック領域内の専用クロック ネットにクロック信号を供給します。BUFR は I/O ロジックだけでなく、同じクロック領域および隣接するクロック領域のロジックリソース (CLB、ブロック RAM など) も駆動できます。BUFR はクロック対応ピンまたはローカル インターコネクトで駆動できます。クロック入力信号を分周したクロックを出力することもできます。分周の除算値は、1 ～ 8 の整数です。BUFR は、クロックドメインの切り替えやシリアルからパラレルへの変換が必要なソース同期アプリケーションに適しています。通常、1 つのクロック領域 (リージョナル クロック ネットワーク 2 つ) には BUFR が 2 つ含まれます。中央列には BUFR は含まれません。

ポートの説明

ポート名	タイプ	幅	機能
CE	入力	1	クロック イネーブル ポート Low でアサートされる場合、ポート O で出力クロックがディスエーブルになります。High でアサートされる場合、分周されたクロック出力を生成するのに使用されるカウンタがリセットされます。
CLR	入力	1	分周クロック出力用のカウンタリセット。High でアサートされる場合、分周されたクロック出力を生成するのに使用されるカウンタがリセットされます。
I	入力	1	クロック入力ポート。BUFR のクロック ソース ポートです。BUFIO の出力またはローカル インターコネクトで駆動できます。
O	出力	1	クロック出力ポート BUFR と同じクロック領域および 2 つの隣接するクロック領域 (最大 3 クロック領域) のクロック ネットを駆動できます。FPGA および IOB を駆動します。

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
BUFR_DIVIDE	文字列	BYPASS、1、2、3、4、5、6、7、8	BYPASS	出力クロックが入力クロックを分周したものであるかを定義します。

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;

-- BUFR: Regional Clock Buffer for I/O and Logic Resources
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFR_inst : BUFR
generic map (
    BUFR_DIVIDE => "BYPASS", -- Defines whether the output clock is a divided version of input clock.
    SIM_DEVICE => "VIRTEX4"
)
port map (
    O => O,      -- 1-bit Clock output port. This port drives the clock tracks in the clock region of the BUFR
                  -- and the two adjacent clock regions. This port drives FPGA fabric, and IOBs.

    CE => CE,     -- 1-bit Clock enable port. When asserted Low, this port disables the output clock at port O.
                  -- When asserted High, this port resets the counter used to produce the divided clock output.

    CLR => CLR,   -- 1-bit Counter reset for divided clock output. When asserted High, this port resets the
                  -- counter used to produce the divided clock output.

    I => I        -- 1-bit Clock input port. This port is the clock source port for BUFR. It can be driven by
                  -- BUFRIO output or local interconnect.
);

-- End of BUFR_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// BUFR: Regional Clock Buffer /w Enable, Clear and Division Capabilities
//      Virtex-4/5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

BUFR #(
    .BUFR_DIVIDE("BYPASS"), // "BYPASS", "1", "2", "3", "4", "5", "6", "7", "8"
    .SIM_DEVICE("VIRTEX4") // Specify target device, "VIRTEX4", "VIRTEX5", "VIRTEX6"
) BUFR_inst (
    .O(O),      // Clock buffer output
    .CE(CE),    // Clock enable input
    .CLR(CLR),  // Clock buffer reset input
    .I(I)       // Clock buffer input
);

// End of BUFR_inst instantiation

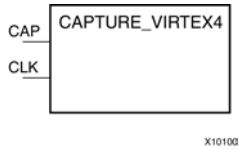
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

CAPTURE_VIRTEX4

プリミティブ : Virtex®-4 Boundary Scan Logic Control Circuit



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 に切り替わるとき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;

-- CAPTURE_VIRTEX4: Register State Capture for Bitstream Readback
--                               Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

CAPTURE_VIRTEX4_inst : CAPTURE_VIRTEX4
generic map (
    ONESHOT => TRUE) -- TRUE or FALSE
port map (
    CAP => CAP,      -- Capture input
    CLK => CLK       -- Clock input
);
-- End of CAPTURE_VIRTEX4_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// CAPTURE_VIRTEX4: Register State Capture for Bitstream Readback
//                               Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

CAPTURE_VIRTEX4 #(
    .ONESHOT("TRUE") // "TRUE" or "FALSE"
) CAPTURE_VIRTEX4_inst (
    .CAP(CAP),        // Capture input
    .CLK(CLK)         // Clock input
);

// End of CAPTURE_VIRTEX4_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

DCIRESET

プリミティブ : DCI State Machine Reset (After Configuration Has Been Completed)



概要

このデザイン エLEMENTは、コンフィギュレーション後に DCI ステート マシンをリセットするために使用します。

ポートの説明

ポート名	タイプ	幅	機能
LOCKED	出力	1	リセット後に DCI ステート マシンが安定状態になったことを示します。
RST	入力	1	DCI ステート マシンを初期ステートから開始します。

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

使用可能な属性

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;

-- DCIRESET: DCI State Machine Reset (After Configuration Has Been Completed)
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

DCIRESET_inst : DCIRESET
generic map (
)
port map (
  LOCKED => LOCKED, -- 1-bit Indicates that DCI state machine has achieved a stable state after reset.
  RST => RST        -- 1-bit Invokes the DCI state machine to start from initial state.
);

-- End of DCIRESET_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// DCIRESET: Digital Controlled Impedance (DCI) Reset Component
//          Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

DCIRESET DCIRESET_inst (
    .LOCKED(LOCKED), // 1-bit DCI LOCKED Output
    .RST(RST)        // 1-bit DCI Reset Input
);

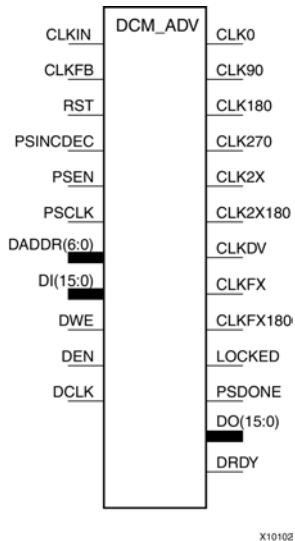
// End of DCIRESET_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

DCM_ADV

プリミティブ : Advanced Digital Clock Manager Circuit



概要

このデザイン エLEMENTは、位相制御機能および周波数合成制御機能を持つコンフィギュレーション/リコンフィギュレーション可能な DLL です。このコンポーネントは、システムで必要になる多種のクロックを発生させて制御するために、多くの FPGA アプリケーションで使用されます。ダイナミック リコンフィギュレーションが不要な場合は、DVM_BASE または DCM_PS コンポーネントを使用します。

ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLK0	出力	1	CLKIN の有効周波数と同じ周波数のクロックを出力します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。CLKFB ピンを接続した場合、CLK0 は CLKIN の位相に揃えられます。
CLK90	出力	1	CLK0 と同じ周波数で位相を 90 度シフトしたクロックを出力します。
CLK180	出力	1	CLK0 と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLK270	出力	1	CLK0 と同じ周波数で位相を 270 度シフトしたクロックを出力します。
CLK2X	出力	1	デューティ サイクルが 50-50 に自動調整された、CLK0 と位相が同じで周波数が 2 倍のクロックを出力します。DCM がロック状態になるまでは、周波数が入力クロックの 1 倍で、デューティ サイクルが 25-75 のクロックが CLK2X に出力されます。これにより、DCM がソース クロックに対して正しいエッジでロック状態になります。
CLK2X180	出力	1	CLK2X と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKDV	出力	1	CLK0 と位相が同じで、CLKIN の有効周波数を分周したクロックが出力されます。分周する係数は、CLKDV_DIVIDE 属性で指定します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。

ポート名	方向	幅	機能
CLKFX	出力	1	<p>次の式で求められる周波数のクロックを出力します。</p> $\text{CLKFX の周波数} = (M/D) \times (\text{CLKIN の有効周波数})$ <p>この式で、M は CLKFX_MULTIPLY 属性で指定し、D は CLKFX_DIVIDE 属性で指定します。M および D の値の範囲、入力および出力の周波数範囲は、このアーキテクチャのデータシートを参照してください。CLKFX 出力の立ち上がりエッジは、フィードバックパス (CLKFB) が使用されるとき、CLK0、CLK2X、および CLKDV の立ち上がりエッジに揃えられます。M と D の値に公約数がない場合、位相は D 入力クロックサイクルごとに揃えられます。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。</p>
CLKFX180	出力	1	CLKFX と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKIN	入力	1	<p>DCM にソース クロックを供給します。CLKIN の周波数はこのアーキテクチャのデータシートで指定された範囲内にする必要があります。クロック入力信号は、次のいずれかのバッファから供給します。</p> <ul style="list-style-type: none"> IBUFG : グローバル クロック入力バッファ。デバイス上で DCM と同じ側 (上または下) にある IBUFG を使用すると、クロック入力パスが調整されます。 BUFG/BUFGCTRL : 内部グローバル クロック バッファ。専用グローバル配線を使用してデバイス上のどの DCM でも駆動できます。2 つの DCM を直列に接続する場合には、DCM の CLKIN ピンを駆動できます。 IBUF : 入力バッファ。IBUF で CLKIN 入力を駆動する場合、PAD から DCM 入力へのスキューは調整されず、ジッタが増加する可能性があります。このコンフィギュレーションは、使用しないでください。
CLKFB	入力	1	<p>クロック出力の遅延を調整してクロック入力と位相を揃えるために参照するクロック信号です。DCM にフィードバックを供給するには、CLK0 出力のみを CLKFB 入力に接続しますが、内部フィードバックの場合は BUFG コンポーネントを介して、外部フィードバックの場合は OBUF - IBUFG を介します。CLK_FEEDBACK 属性を 1X に設定します。CLKFB ピンを接続すると、CLK0、CLKDV、および CLKFX が CLKIN の位相に揃えられます。CLKFB ピンを接続しない場合は、CLK_FEEDBACK を NONE に設定します。この場合、CLKFX および CLKFX180 出力は、CLKIN の位相に揃えられませんが有効になります。</p>
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力。
PSDONE	出力	1	<p>ダイナミック CLKIN セレクト入力。High (1) のときは CLKIN1 が、Low (0) のときは CLKIN2 が選択されます。2 つのクロックを選択する必要がない場合は、この入力を 1 にします。</p>

ポート名	方向	幅	機能
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。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソースクロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティサイクルが崩れたり、クロック間のスキューが調整される可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロック サイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。このクロックは、GSR がリリースされると安定します。
PSCLK	入力	1	DCM 位相シフトのソースクロックを供給します。位相シフトクロック信号は、どのクロックソース (内部または外部) でも駆動できます。 PSCLK の周波数範囲は、PSCLK_FREQ_LF/HF で定義します (このアーキテクチャのデータシートを参照)。CLKOUT_PHASE_SHIFT 属性を NONE または FIXED に設定している場合は、この入力をグラウンドに接続する必要があります。
PSINCDEC	入力	1	PSINCDEC 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードのいずれかに設定されているときに、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。位相シフト係数をインクリメント/デクリメントすると、それに応じて出力クロックの位相がシフトします。PSINCDEC 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。CLKOUT_PHASE_SHIFT 属性を NONE または FIXED に設定している場合は、この入力をグラウンドに接続する必要があります。
PSEN	入力	1	PSEN 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードに設定されているときに、この信号によって可変位相シフトを開始します。可変位相シフトを有効にするには、PSEN 信号を PSCLK の 1 サイクル分アクティブにする必要があります。位相の変更は、CLKIN の 100 周期分と PSCLK の 3 周期分を加えた時間以内に有効になり、PSDONE が High になることにより示されます。位相が変化する間、出力に突発的な変化やグリッチは発生しません。PSEN がイネーブルになってから PSDONE が High になるまでの間、DCM の出力クロックは元の位相からターゲットの位相に少しずつ移動していきます。PSDONE が High になったら、位相シフトは完了です。CLKOUT_PHASE_SHIFT 属性を NONE または FIXED に設定している場合は、この入力をグラウンドに接続する必要があります。
ダイナミック リコンフィギュレーション / DCM ステータス			
ダイナミック コンフィギュレーションの詳細は、該当デバイスの Configuration User Guide を参照してください。			

ポート名	方向	幅	機能
DO	出力	16	<p>ダイナミック リコンフィギュレーションを使用していない場合は DCM のステータス出力、使用している場合はリコンフィギュレーションのデータ出力になります。DCM ステータスが表示されている場合、次のマップが適用されます。</p> <ul style="list-style-type: none"> DO[0] : 位相シフト オーバーフロー DO[1] : CLKIN の停止 DO[2] : CLKFX の停止 DO[3] : CLKFB の停止 DO[15:4] : 割り当てなし
DRDY	出力	1	DCM のダイナミック リコンフィギュレーション機能が準備完了になったことを示します。
DI	入力	16	DI 入力バスは、ダイナミック リコンフィギュレーションのデータ入力です。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DADDR	入力	7	DADDR 入力バスは、ダイナミック リコンフィギュレーションのアドレス入力です。このバスを使用しない場合は、すべてのビットを 0 にする必要があります。
DWE	入力	1	DI データの DADDR アドレスへの書き込みを制御するライト イネーブル信号です。使用しない場合は、Low に接続する必要があります。
DEN	入力	1	ダイナミック リコンフィギュレーション機能を使用するかどうかを制御する信号です。ダイナミック リコンフィギュレーションが使用されていないときに DO 出力バスに DCM ステータス信号を反映させるには、DEN を Low に設定する必要があります。
DCLK	入力	1	DCM のダイナミック リコンフィギュレーション回路のソース クロックを供給します。DCLK には、CLKIN とは位相および周波数が非同期なクロックを使用できます。ダイナミック リコンフィギュレーション クロック信号は、どのクロック ソースでも駆動できます。DCLK の周波数範囲はこのアーキテクチャのデータシートに記載されています。ダイナミック リコンフィギュレーションを使用しない場合は、この入力をグラウンドに接続する必要があります。

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
CLK_FEEDBACK	文字列	1X、NONE	1X	クロック フィードバックを指定
CLKDV_DIVIDE	浮動 小数点	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、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、DCM のクロック分周器 (CLKDV 出力) の分周比 を指定
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 出力の乗倍比を指定
CLKIN_DIVIDE_BY_2	ブール 代数	FALSE、TRUE	FALSE	DCM の入力クロック周波数の要件に合う ように、必要に応じて入力クロック周波数 を 2 で分周
CLKIN_PERIOD	浮動 小数点	1.25 ~ 1000.00	10.0	入力クロックの周期を 1.25 ~ 1000.00 の 範囲で設定 (ns)
CLKOUT_PHASE_SHIFT	文字列	NONE、FIXED、 VARIABLE_ POSITIVE、 VARIABLE _CENTER、DIRECT	NONE	位相シフトのモードを指定
DCM_PERFORMANCE_MODE	文字列	MAX_SPEED、 MAX_RANGE	MAX_SPEED	DCM を低ジッタの高周波数クロックを生成 するよう最適化するか、位相シフト範囲が広 い低周波数のクロックを生成するよう最適 化するかを指定
DESKEW_ADJUST	文字列	SOURCE_ SYNCHRONOUS、 SYSTEM_ SYNCHRONOUS、 0 ~ 15	SYSTEM_ SYNCHRONOUS	フィードバック パスの遅延の量を制御。ソー ス同期のインターフェイスで使用する必要 があります。
DFS_FREQUENCY_MODE	文字列	LOW、HIGH	LOW	周波数合成の周波数モードを指定
DLL_FREQUENCY_MODE	文字列	LOW、HIGH	LOW	DLL の周波数モードを指定
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
FACTORY_JF	16 進数	16 ビット値	F0F0	この属性は、DCM のジッタフィルタ特性に 影響します。サイリンクスからの指示なしに このデフォルト値を変更しないでください。
PHASE_SHIFT	整数	-255 ~ 1023	0	位相シフト量を指定。この値の範囲は CLKOUT_PHASE_SHIFT の指定によって 異なります。
SIM_DEVICE	文字列	VIRTEX4、 VIRTEX5	VIRTEX5	デバイスの選択
STARTUP_WAIT	ブール 代数	FALSE、TRUE	FALSE	TRUE に設定すると、DCM がロック状態に なるまでコンフィギュレーション スタートアッ プ シーケンスの指定したサイクルで待機

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;

-- DCM_ADV: Digital Clock Manager Circuit
--      Virtex-4/5
-- Xilinx HDL Libraries Guide, version 11.2

DCM_ADV_inst : DCM_ADV
generic map (
    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 => 10.0,  -- Specify period of input clock in ns from 1.25 to 1000.00
    CLKOUT_PHASE_SHIFT => "NONE",  -- Specify phase shift mode of NONE, FIXED,
                                --      VARIABLE_POSITIVE, VARIABLE_CENTER or DIRECT
    CLK_FEEDBACK => "1X",  -- Specify clock feedback of NONE or 1X
    DCM_PERFORMANCE_MODE => "MAX_SPEED",  -- Can be MAX_SPEED or MAX_RANGE
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS",  -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                --      an integer from 0 to 15
    DFS_FREQUENCY_MODE => "LOW",  -- HIGH or LOW frequency mode for frequency synthesis
    DLL_FREQUENCY_MODE => "LOW",  -- LOW, HIGH, or HIGH_SER frequency mode for DLL
    DUTY_CYCLE_CORRECTION => TRUE,  -- Duty cycle correction, TRUE or FALSE
    FACTORY_JF => X"F0F0",  -- FACTORY JF Values Suggested to be set to X"F0F0"
    PHASE_SHIFT => 0,  -- Amount of fixed phase shift from -255 to 1023
    SIM_DEVICE => "VIRTEX4",  -- Set target device, "VIRTEX4" or "VIRTEX5"
    STARTUP_WAIT => FALSE)  -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
    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
    DO => DO,  -- 16-bit data output for Dynamic Reconfiguration Port (DRP)
    DRDY => DRDY,  -- Ready output signal from the DRP
    LOCKED => LOCKED,  -- DCM LOCK status output
    PSDONE => PSDONE,  -- Dynamic phase adjust done output
    CLKFB => CLKFB,  -- DCM clock feedback
    CLKIN => CLKIN,  -- Clock input (from IBUFG, BUFG or DCM)
    DADDR => DADDR,  -- 7-bit address for the DRP
    DCLK => DCLK,  -- Clock for the DRP
    DEN => DEN,  -- Enable input for the DRP
    DI => DI,  -- 16-bit data input for the DRP
    DWE => DWE,  -- Active high allows for writing configuration memory
    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_ADV_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// DCM_ADV: Digital Clock Manager Circuit
//          Virtex-4/5
// Xilinx HDL Libraries Guide, version 11.2

DCM_ADV #(
    .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(10.0), // Specify period of input clock in ns from 1.25 to 1000.00
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift mode of NONE, FIXED,
                                  // VARIABLE_POSITIVE, VARIABLE_CENTER or DIRECT
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE, 1X or 2X
    .DCM_PERFORMANCE_MODE("MAX_SPEED"), // Can be MAX_SPEED or MAX_RANGE
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                          // an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // LOW, HIGH, or HIGH_SER frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, "TRUE"/"FALSE"
    .FACTORY_JF(16'hf0f0), // FACTORY JF value suggested to be set to 16'hf0f0
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 1023
    .SIM_DEVICE("VIRTEX4"), // Set target device, "VIRTEX4" or "VIRTEX5"
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, "TRUE"/"FALSE"
) DCM_ADV_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
    .DO(DO),               // 16-bit data output for Dynamic Reconfiguration Port (DRP)
    .DRDY(DRDY),           // Ready output signal from the DRP
    .LOCKED(LOCKED),       // DCM LOCK status output
    .PSDONE(PSDONE),       // Dynamic phase adjust done output
    .CLKFB(CLKFB),         // DCM clock feedback
    .CLKIN(CLKIN),         // Clock input (from IBUFG, BUFG or DCM)
    .DADDR(DADDR),         // 7-bit address for the DRP
    .DCLK(DCLK),           // Clock for the DRP
    .DEN(DEN),             // Enable input for the DRP
    .DI(DI),               // 16-bit data input for the DRP
    .DWE(DWE),             // Active high allows for writing configuration memory
    .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
);

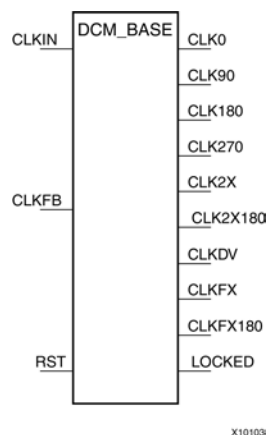
// End of DCM_ADV_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

DCM_BASE

プリミティブ : Base Digital Clock Manager Circuit



概要

このデザイン エLEMENTは、位相制御機能および周波数合成制御機能を持つコンフィギュレーション可能な DLL です。このコンポーネントは、システムで必要になる多種のクロックを発生させて制御するために、多くの FPGA アプリケーションで使用されます。ダイナミックリコンフィギュレーションが必用な場合は、DVM_ADV コンポーネントを使用し、ダイナミック位相シフトが必用な場合は、DVM_PS コンポーネントを使用します。

ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLK0	出力	1	CLKIN の有効周波数と同じ周波数のクロックを出力します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。CLKFB ピンを接続した場合、CLK0 は CLKIN の位相に揃えられます。
CLK90	出力	1	CLK0 と同じ周波数で位相を 90 度シフトしたクロックを出力します。
CLK180	出力	1	CLK0 と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLK270	出力	1	CLK0 と同じ周波数で位相を 270 度シフトしたクロックを出力します。
CLK2X	出力	1	デューティサイクルが 50-50 に自動調整された、CLK0 と位相が同じで周波数が 2 倍のクロックを出力します。DCM がロック状態になるまでは、周波数が入力クロックの 1 倍で、デューティサイクルが 25-75 のクロックが CLK2X に出力されます。これにより、DCM がソース クロックに対して正しいエッジでロック状態になります。
CLK2X180	出力	1	CLK2X と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKDV	出力	1	CLK0 と位相が同じで、CLKIN の有効周波数を分周したクロックが出力されます。分周する係数は、CLKDV_DIVIDE 属性で指定します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。

ポート名	方向	幅	機能
CLKFX	出力	1	<p>次の式で求められる周波数のクロックを出力します。</p> <p>CLKFX の周波数 = (M/D) x (CLKIN の有効周波数)</p> <p>この式で、M は CLKFX_MULTIPLY 属性で指定し、D は CLKFX_DIVIDE 属性で指定します。M および D の値の範囲、入力および出力の周波数範囲は、このアーキテクチャのデータシートを参照してください。CLKFX 出力の立ち上がりエッジは、フィードバックパス (CLKFB) が使用されるとき、CLK0、CLK2X、および CLKDV の立ち上がりエッジに揃えられます。M と D の値に公約数がない場合、位相は D 入力クロック サイクルごとに揃えられます。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。</p>
CLKFX180	出力	1	CLKFX と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKIN	入力	1	<p>DCM にソース クロックを供給します。CLKIN の周波数はこのアーキテクチャのデータシートで指定された範囲内にする必要があります。クロック入力信号は、次のいずれかのバッファから供給します。</p> <ul style="list-style-type: none"> IBUFG : グローバル クロック入力バッファ。デバイス上で DCM と同じ側 (上または下) にある IBUFG を使用すると、クロック入力パスが調整されます。 BUFG/BUFGCTRL : 内部グローバル クロック バッファ。専用グローバル配線を使用してデバイス上のどの DCM でも駆動できます。2 つの DCM を直列に接続する場合には、DCM の CLKIN ピンを駆動できます。 IBUF : 入力バッファ。IBUF で CLKIN 入力を駆動する場合、PAD から DCM 入力へのスキューは調整されず、ジッタが増加する可能性があります。このコンフィギュレーションは、使用しないください。
CLKFB	入力	1	<p>クロック出力の遅延を調整してクロック入力と位相を揃えるために参照するクロック信号です。DCM にフィードバックを供給するには、CLK0 出力のみを CLKFB 入力に接続しますが、内部フィードバックの場合は BUFG コンポーネントを介して、外部フィードバックの場合は OBUF - IBUFG を介します。CLK_FEEDBACK 属性を 1X に設定します。CLKFB ピンを接続すると、CLK0、CLKDV、および CLKFX が CLKIN の位相に揃えられます。CLKFB ピンを接続しない場合は、CLK_FEEDBACK を NONE に設定します。この場合、CLKFX および CLKFX180 出力は、CLKIN の位相に揃えられませんが有効になります。</p>
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力
RST	入力	1	<p>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。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソース クロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティ サイクルが崩れたり、クロック間のスキューが調整される可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロック サイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。</p>

ポート名	方向	幅	機能
			があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。このクロックは、GSR がリリースされると安定します。

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
CLK_FEEDBACK	文字列	1X、2X、NONE	1X	DCM へのフィードバック入力を指定 (CLK0 または CLK2X)
CLKDV_DIVIDE	浮動 小数点	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、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、DCM のクロック分周器 (CLKDV 出力) の分周比を指定
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	DCM の入力クロック周波数の要件に合うように、必要に応じて入力クロック周波数を 2 で分周
CLKIN_PERIOD	浮動 小数点	1.25 ~ 1000.00	10.0	入力クロックの周期を 1.25 ~ 1000.00 の範囲で設定 (ns)
CLKOUT_PHASE_SHIFT	文字列	NONE、FIXED、 VARIABLE_ POSITIVE、 VARIABLE_ CENTER、DIRECT	NONE	位相シフトのモードを指定
DCM_PERFORMANCE_MODE	文字列	MAX_SPEED、 MAX_RANGE	MAX_SPEED	DCM を低ジッタの高周波数クロックを生成するよう最適化するか、位相シフト範囲が広い低周波数のクロックを生成するよう最適化するかを指定
DESKEW_ADJUST	文字列	SOURCE_ SYNCHRONOUS、 SYSTEM_ SYNCHRONOUS、 0 ~ 15	SYSTEM_ SYNCHRONOUS	フィードバックパスの遅延の量を制御。ソース同期のインターフェイスで使用する必要があります。
DFS_FREQUENCY_MODE	文字列	LOW、HIGH	LOW	周波数合成の周波数モードを指定
DLL_FREQUENCY_MODE	文字列	LOW、HIGH	LOW	DLL の周波数モードを指定
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

属性	タイプ	値	デフォルト	説明
FACTORY_JF	16 進数	16 ビット値	F0F0	この属性は、DCM のジッタフィルタ特性に影響します。ザイリンクスからの指示なしにこのデフォルト値を変更しないでください。
PHASE_SHIFT	整数	-255 ~ 1023	0	位相シフト量を指定。この値の範囲は CLKOUT_PHASE_SHIFT の指定によって異なります。
STARTUP_WAIT	ブール 代数	FALSE、TRUE	FALSE	TRUE に設定すると、DCM がロック状態になるまでコンフィギュレーション スタートアップ シーケンスの指定したサイクルで待機

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;

-- DCM_BASE: Base Digital Clock Manager Circuit
--      Virtex-4/5
-- Xilinx HDL Libraries Guide, version 11.2

DCM_BASE_inst : DCM_BASE
generic map (
  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 => 10.0, -- Specify period of input clock in ns from 1.25 to 1000.00
  CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift mode of NONE or FIXED
  CLK_FEEDBACK => "1X", -- Specify clock feedback of NONE or 1X
  DCM_PERFORMANCE_MODE => "MAX_SPEED", -- Can be MAX_SPEED or MAX_RANGE
  DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                           -- an integer from 0 to 15
  DFS_FREQUENCY_MODE => "LOW", -- LOW or HIGH frequency mode for frequency synthesis
  DLL_FREQUENCY_MODE => "LOW", -- LOW, HIGH, or HIGH_SER frequency mode for DLL
  DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
  FACTORY_JF => X"F0F0", -- FACTORY JF Values Suggested to be set to X"F0F0"
  PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 1023
  STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
  CLK0 => CLK0, -- 0 degree DCM CLK ouptput
  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
  CLKFB => CLKFB, -- DCM clock feedback
  CLKIN => CLKIN, -- Clock input (from IBUFG, BUFG or DCM)
  RST => RST -- DCM asynchronous reset input
);

-- End of DCM_BASE_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// DCM_BASE: Base Digital Clock Manager Circuit
//          Virtex-4/5
// Xilinx HDL Libraries Guide, version 11.2

DCM_BASE #(
    .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(10.0), // Specify period of input clock in ns from 1.25 to 1000.00
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift mode of NONE or FIXED
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE, 1X or 2X
    .DCM_PERFORMANCE_MODE("MAX_SPEED"), // Can be MAX_SPEED or MAX_RANGE
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                          // an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // LOW or HIGH frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // LOW, HIGH, or HIGH_SER frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
    .FACTORY_JF(16'hf0f0), // FACTORY JF value suggested to be set to 16'hf0f0
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 1023
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_BASE_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
    .CLKFB(CLKFB), // DCM clock feedback
    .CLKIN(CLKIN), // Clock input (from IBUFG, BUFG or DCM)
    .RST(RST) // DCM asynchronous reset input
);

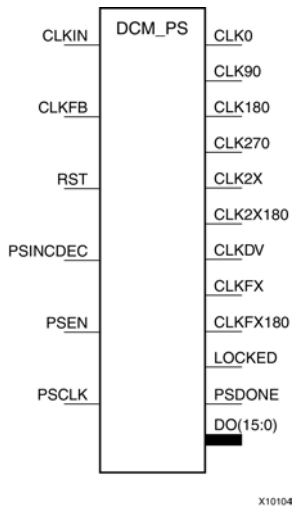
// End of DCM_BASE_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

DCM_PS

プリミティブ : Digital Clock Manager with Basic and Phase Shift Features



概要

このデザイン エレメントは、位相制御機能および周波数合成制御機能を持つコンフィギュレーション可能な DLL です。このコンポーネントは、システムで必要になる多種のクロックを発生させて制御するために、多くの FPGA アプリケーションで使用されます。ダイナミックリコンフィギュレーションが必用な場合は、DVM_ADV コンポーネントを使用し、ダイナミック位相シフトが不要な場合は、DVM_BASE コンポーネントを使用します。

ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLK0	出力	1	CLKIN の有効周波数と同じ周波数のクロックを出力します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。CLKFB ピンを接続した場合、CLK0 は CLKIN の位相に揃えられます。
CLK90	出力	1	CLK0 と同じ周波数で位相を 90 度シフトしたクロックを出力します。
CLK180	出力	1	CLK0 と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLK270	出力	1	CLK0 と同じ周波数で位相を 270 度シフトしたクロックを出力します。
CLK2X	出力	1	デューティ サイクルが 50-50 に自動調整された、CLK0 と位相が同じで周波数が 2 倍のクロックを出力します。DCM がロック状態になるまでは、周波数が入力クロックの 1 倍で、デューティ サイクルが 25-75 のクロックが CLK2X に出力されます。これにより、DCM がソース クロックに対して正しいエッジでロック状態になります。
CLK2X180	出力	1	CLK2X と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKDV	出力	1	CLK0 と位相が同じで、CLKIN の有効周波数を分周したクロックが出力されます。分周する係数は、CLKDV_DIVIDE 属性で指定します。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。

ポート名	方向	幅	機能
CLKFX	出力	1	<p>次の式で求められる周波数のクロックを出力します。</p> $\text{CLKFX の周波数} = (M/D) \times (\text{CLKIN の有効周波数})$ <p>この式で、M は CLKFX_MULTIPLY 属性で指定し、D は CLKFX_DIVIDE 属性で指定します。M および D の値の範囲、入力および出力の周波数範囲は、データシートを参照してください。CLKFX 出力の立ち上がりエッジは、フィードバックパス (CLKFB) が使用されるとき、CLK0、CLK2X、および CLKDV の立ち上がりエッジに揃えられます。M と D の値に公約数がない場合、位相は D 入力クロック サイクルごとに揃えられます。デフォルトでは、CLKIN_DIVIDE_BY_2 属性を TRUE に設定した場合以外は、CLKIN の有効周波数は CLKIN の周波数と同じになります。</p>
CLKFX180	出力	1	CLKFX と同じ周波数で位相を 180 度シフトしたクロックを出力します。
CLKIN	入力	1	<p>DCM にソース クロックを供給します。CLKIN の周波数はデータシートで指定された範囲内にする必要があります。クロック入力信号は、次のいずれかのバッファから供給します。</p> <ul style="list-style-type: none"> IBUFG : グローバル クロック入力バッファ。デバイス上で DCM と同じ側 (上または下) にある IBUFG を使用すると、クロック入力パスが調整されます。 BUFG/BUFGCTRL : 内部グローバル クロック バッファ。専用グローバル配線を使用してデバイス上のどの DCM でも駆動できます。2 つの DCM を直列に接続する場合には、DCM の CLKIN ピンを駆動できます。 IBUF : 入力バッファ。IBUF で CLKIN 入力を駆動する場合、PAD から DCM 入力へのスキューは調整されず、ジッタが増加する可能性があります。このコンフィギュレーションは、使用しないでください。
CLKFB	入力	1	<p>クロック出力の遅延を調整してクロック入力と位相を揃えるために参照するクロック信号です。DCM にフィードバックを供給するには、CLK0 出力のみを CLKFB 入力に接続しますが、内部フィードバックの場合は BUFG コンポーネントを介して、外部フィードバックの場合は OBUF - IBUFG を介します。CLK_FEEDBACK 属性を 1X に設定します。CLKFB ピンを接続すると、CLK0、CLKDV、および CLKFX が CLKIN の位相に揃えられます。CLKFB ピンを接続しない場合は、CLK_FEEDBACK を NONE に設定します。この場合、CLKFX および CLKFX180 出力は、CLKIN の位相に揃えられません。有効になります。</p>
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力
PSDONE	出力	1	<p>ダイナミック CLKIN セレクト入力。High (1) のときは CLKIN1 が、Low (0) のときは CLKIN2 が選択されます。2 つのクロックを選択する必要がない場合は、この入力を 1 にします。</p>

ポート名	方向	幅	機能
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。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソースクロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティサイクルが崩れたり、クロック間のスキューが調整される可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始します。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロックサイクル以上安定するまで RST 信号をディアサートしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。このクロックは、GSR がリリースされると安定します。
PSCLK	入力	1	DCM 位相シフトのソースクロックを供給します。位相シフトクロック信号は、どのクロックソース (内部または外部) でも駆動できます。 PSCLK の周波数範囲は、PSCLK_FREQ_LF/HF で定義します (データシートを参照)。CLKOUT_PHASE_SHIFT 属性を NONE または FIXED に設定している場合は、この入力をグラウンドに接続する必要があります。
PSINCDEC	入力	1	PSINCDEC 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードのいずれかに設定されているときに、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。位相シフト係数をインクリメント/デクリメントすると、それに応じて出力クロックの位相がシフトします。PSINCDEC 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。CLKOUT_PHASE_SHIFT 属性を NONE または FIXED に設定している場合は、この入力をグラウンドに接続する必要があります。
PSEN	入力	1	PSEN 入力は、PSCLK に同期しています。CLKOUT_PHASE_SHIFT が変数モードに設定されているときに、この信号によって可変位相シフトを開始します。可変位相シフトを有効にするには、PSEN 信号を PSCLK の 1 サイクル分アクティブにする必要があります。位相の変更は、CLKIN の 100 周期分と PSCLK の 3 周期分を加えた時間以内に有効になり、PSDONE が High になることにより示されます。位相が変化する間、出力に突発的な変化やグリッチは発生しません。PSEN がイネーブルになってから PSDONE が High になるまでの間、DCM の出力クロックは元の位相からターゲットの位相に少しずつ移動していきます。PSDONE が High になったら、位相シフトは完了です。CLKOUT_PHASE_SHIFT 属性を NONE または FIXED に設定している場合は、この入力をグラウンドに接続する必要があります。

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
CLK_FEEDBACK	文字列	1X、2X、NONE	1X	クロック フィードバックを指定
CLKDV_DIVIDE	浮動 小数点	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、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、DCM の クロック分周器 (CLKDV 出力) の分周比を 指定
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	DCM の入力クロック周波数の要件に合うよう に、必要に応じて入力クロック周波数を 2 で 分周
CLKIN_PERIOD	浮動 小数点	1.25 ~ 1000.00	10.0	入力クロックの周期を 1.25 ~ 1000.00 の範 囲で設定 (ns)
CLKOUT_PHASE_ SHIFT	文字列	NONE、FIXED、 VARIABLE_ POSITIVE、 VARIABLE_ CENTER、DIRECT	NONE	位相シフトのモードを指定
DESKEW_ADJUST	文字列	SOURCE_ SYNCHRONOUS、 SYSTEM_ SYNCHRONOUS、 0 ~ 15	SYSTEM_ SYNCHRONOUS	フィードバック パスの遅延の量を制御。ソー ス同期のインターフェイスで使用する必要が あります。
DFS_FREQUENCY_ MODE	文字列	LOW、HIGH	LOW	周波数合成の周波数モードを指定
DLL_FREQUENCY_ MODE	文字列	LOW、HIGH	LOW	DLL の周波数モードを指定
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
FACTORY_JF	16 進数	16 ビット値	F0F0	この属性は、DCM のジッタ フィルタ特性に影 響します。サイリックスからの指示なしにこの デフォルト値を変更しないでください。
PHASE_SHIFT	整数	-255 ~ 1023	0	位相シフト量を指定。この値の範囲は CLKOUT_PHASE_SHIFT の指定によって異 なります。
STARTUP_WAIT	ブール 代数	FALSE、TRUE	FALSE	TRUE に設定すると、DCM がロック状態にな るまでコンフィギュレーション スタートアップ シーケンスの指定したサイクルで待機

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_PS: Digital Clock Manager Circuit
--       Virtex-4/5
-- Xilinx HDL Libraries Guide, version 11.2

DCM_PS_inst : DCM_PS
generic map (
    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 => 10.0, -- Specify period of input clock in ns from 1.25 to 1000.00
    CLKOUT_PHASE_SHIFT => "NONE", -- Specify phase shift mode of NONE, FIXED,
                                -- VARIABLE_POSITIVE, VARIABLE_CENTER or DIRECT
    CLK_FEEDBACK => "1X", -- Specify clock feedback of NONE or 1X
    DCM_PERFORMANCE_MODE => "MAX_SPEED", -- Can be MAX_SPEED or MAX_RANGE
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                -- an integer from 0 to 15
    DFS_FREQUENCY_MODE => "LOW", -- HIGH or LOW frequency mode for frequency synthesis
    DLL_FREQUENCY_MODE => "LOW", -- LOW, HIGH, or HIGH_SER frequency mode for DLL
    DUTY_CYCLE_CORRECTION => TRUE, -- Duty cycle correction, TRUE or FALSE
    FACTORY_JF => X"F0F0", -- FACTORY JF Values Suggested to be set to X"F0F0"
    PHASE_SHIFT => 0, -- Amount of fixed phase shift from -255 to 1023
    STARTUP_WAIT => FALSE) -- Delay configuration DONE until DCM LOCK, TRUE/FALSE
port map (
    CLK0 => CLK0, -- 0 degree DCM CLK ouptput
    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
    DO => DO, -- 16-bit data output for Dynamic Reconfiguration Port (DRP)
    LOCKED => LOCKED, -- DCM LOCK status output
    PSDONE => PSDONE, -- Dynamic phase adjust done 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_PS_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// DCM_PS: Dynamic Phase Shift Digital Clock Manager Circuit
//      Virtex-4/5
// Xilinx HDL Libraries Guide, version 11.2

DCM_PS #(
    .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(10.0), // Specify period of input clock in ns from 1.25 to 1000.00
    .CLKOUT_PHASE_SHIFT("NONE"), // Specify phase shift mode of NONE, FIXED,
                                  //      VARIABLE_POSITIVE, VARIABLE_CENTER or DIRECT
    .CLK_FEEDBACK("1X"), // Specify clock feedback of NONE, 1X or 2X
    .DCM_PERFORMANCE_MODE("MAX_SPEED"), // Can be MAX_SPEED or MAX_RANGE
    .DESKEW_ADJUST("SYSTEM_SYNCHRONOUS"), // SOURCE_SYNCHRONOUS, SYSTEM_SYNCHRONOUS or
                                          //      an integer from 0 to 15
    .DFS_FREQUENCY_MODE("LOW"), // HIGH or LOW frequency mode for frequency synthesis
    .DLL_FREQUENCY_MODE("LOW"), // LOW, HIGH, or HIGH_SER frequency mode for DLL
    .DUTY_CYCLE_CORRECTION("TRUE"), // Duty cycle correction, TRUE or FALSE
    .FACTORY_JF(16'hf0f0), // FACTORY JF value suggested to be set to 16'hf0f0
    .PHASE_SHIFT(0), // Amount of fixed phase shift from -255 to 1023
    .STARTUP_WAIT("FALSE") // Delay configuration DONE until DCM LOCK, TRUE/FALSE
) DCM_PS_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
    .DO(DO), // 16-bit data output for Dynamic Reconfiguration Port (DRP)
    .LOCKED(LOCKED), // DCM LOCK status output
    .PSDONE(PSDONE), // Dynamic phase adjust done 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
);

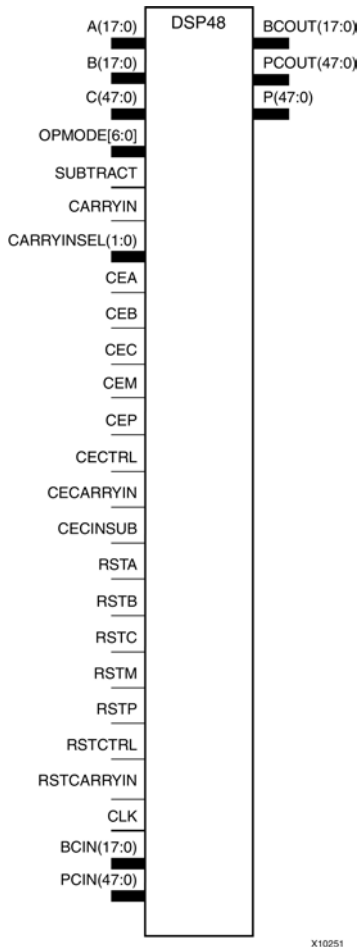
// End of DCM_PS_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

DSP48

プリミティブ : 18x18 Signed Multiplier Followed by a Three-Input Adder with Optional Pipeline Registers



概要

このデザイン エLEMENTのスライスは、48 ビットの出力を持つデジタル信号処理 (DSP) アプリケーションで使用するために開発されたコンポーネントですが、MACC ユニットだけでなくさまざまなアプリケーションで利用できる柔軟性を備えています。基本的な DSP48 スライスは、乗算器と加算器で構成されています。乗算器は、2 つの 18 ビット符号付き 2 の補数オペランドを入力とし、その結果を 36 ビット符号付き 2 の補数で出力します。結果は、48 ビットに符号拡張されます。加算器は、3 つの 48 ビット符号付き 2 の補数オペランドを入力とし、その結果を 48 ビット符号付き 2 の補数で出力します。

加算器のオペランドとしては、乗算器の出力、外部ソース、または加算器のレジスタを介した出力 (累積機能) を使用できます。48 ビット出力では、オーバーフローになるまでに 36 ビット オペランドの 4096 累積が可能です。

ポートの説明

ポート名	方向	幅	機能
CLK	I	1	DSP48 クロック
A	I	18	乗算器の A 入力。加算器の MSW 入力としても使用できます。
B	I	18	乗算器の B 入力。加算器の LSW 入力としても使用できます。
BCIN	I	18	乗算器のカスケード B 入力。加算器の LSW 入力としても使用できます。
C	I	48	加算器の C 入力
PCIN	I	48	前の DSP スライスからカスケード接続された加算器の C 入力
CARRYIN	I	1	加算器のキャリー入力
SUBTRACT	I	1	0 = add、1 = (C, PCIN) - (mult, A:B)
OPMODE	I	7	DSP48 スライスの加算器の制御入力 (OpMode の表を参照)
CARRYINSEL	I	2	キャリーのソースを選択 (CARRYINSEL の表を参照)
CEA	I	1	クロック イネーブル (0 = 保持、1 = AREG をイネーブル)
CEB	I	1	クロック イネーブル (0 = 保持、1 = BREG をイネーブル)
CEC	I	1	クロック イネーブル (0 = 保持、1 = CREG をイネーブル)
CEP	I	1	クロック イネーブル (0 = 保持、1 = PREG をイネーブル)

パイプライン レジスタを定義する合成属性

次に、パイプライン レジスタを定義する合成属性を示します。

属性	機能
AREG	0 = バイパス、1 = 1 個使用、2 = 2 個使用
BREG	0 = バイパス、1 = 1 個使用、2 = 2 個使用
CREG	0 = バイパス、1 = 1 個使用
PREG	0 = バイパス、1 = 1 個使用
MREG	0 = バイパス、1 = 1 個使用
SUBTRACTREG	0 = バイパス、1 = 1 個使用
OPMODEREG	0 = バイパス、1 = 1 個使用
CARRYINSELREG	0 = バイパス、1 = 1 個使用

2 の補数の符号付乗算器

DSP48 スライスに含まれる乗算器は、18 ビット X 18 ビットの 2 の補数乗算器で、結果を 36 ビット符号付き 2 の補数で出力します。乗算器をカスケード接続すると、より大型の乗算器を作成できます。符号付きと符号付き、符号付きと符号なし、符号なしと符号なしの乗算、論理演算、数値演算、バレル シフタ、2 の補数および絶対値での結果出力を簡単にインプリメントできます。2 つの独立したダイナミック データ入力ポートがあります。入力ポートでは、18 ビットの符号付きデータまたは 17 ビットの符号なしデータがサポートされます。

X、Y、および Z マルチプレクサ

動作モード (OpMode) 入力を使用すると、動作中にデザインの機能を変更できます。たとえば、アキュムレータを指定すると、累積プロセスが開始します。OpMode ビットは、コンフィギュレーション RAM で制御してレジスタに取り込むことができます。

次の表に、OpMode の値と、加減算器にデータを供給する 3 つのマルチプレクサの出力での機能を示します。7 ビットの OpMode 制御は、マルチプレクサのセレクトビットに分割できます。セレクトビットの組み合わせによっては、使用できないものもあります。乗算器出力を選択した場合、X および Y 両方のマルチプレクサが乗算器出力により使用されます。

OpMode 制御ビット セレクト X、Y、Z とマルチプレクサ出力

2 進 OpMode			加減算器に供給される X マルチプレクサ出力
Z	Y	X	
XXX	XX	0	0 (デフォルト)
XXX	1	1	乗算器出力
XXX	XX	10	P
XXX	XX	11	A を B と連結

OpMode 制御ビット セレクト X、Y、Z とマルチプレクサ出力

2 進 OpMode			加減算器に供給される Y マルチプレクサ出力
Z	Y	X	
XXX	0	XX	0 (デフォルト)
XXX	1	1	乗算器出力
XXX	10	XX	不正
XXX	11	XX	C

OpMode 制御ビット セレクト X、Y、Z とマルチプレクサ出力

2 進 OpMode			加減算器に供給される Y マルチプレクサ出力
Z	Y	X	
XXX	0	XX	0 (デフォルト)
XXX	1	1	乗算器出力
XXX	10	XX	不正
XXX	11	XX	C

3 入力加減算器制御ロジック

加減算器の出力は、制御入力とデータ入力により決定します。OpMode は、前のセクションで示したように、3 つの加減算器の入力に供給される X、Y、Z マルチプレクサへの入力を選択します。乗算器出力が選択されると、X および Y マルチプレクサの両方が使用されます。加減算器の入力を指定したら、加減算器自体の機能を確認する必要があります。入力マルチプレクサと同様に、OpMode ビットによりこの機能の一部を指定できます。次の表に、この機能を示します。± 記号は加算または減算を示し、減算制御のステートにより指定されます。

16 進 OpMode	2 進 OpMode	加減算器の出力	説明
[6:0]	Z Y X		
0x00	000 00 00	±CIN	0
0x02	000 00 10	±(P + CIN)	P を保持
0x03	000 00 11	±(A:B + CIN)	A:B セレクト
0x05	000 01 01	± (A × B + CIN)	乗算
0x0c	000 11 00	± (C + CIN)	C セレクト

16 進 OpMode	2 進 OpMode	加減算器の出力	説明
[6:0]	Z Y X		
0x0e	000 11 10	$\pm (C + P + CIN)$	フィードバック加算
0x0f	000 11 11	$\pm (A:B + C + CIN)$	36 ビットの加算器
0x10	001 00 00	$PCIN \pm CIN$	P カスケード セレクト
0x12	001 00 10	$PCIN \pm (P + CIN)$	P カスケード フィードバック加算
0x13	001 00 11	$PCIN \pm (A:B + CIN)$	P カスケード加算
0x15	001 01 01	$PCIN \pm (A \times B + CIN)$	P カスケード乗算/加算
0x1c	001 11 00	$PCIN \pm (C + CIN)$	P カスケード加算
0x1e	001 11 10	$PCIN \pm (C + P + CIN)$	P カスケード フィードバック加算/加算
0x1c	001 11 11	$PCIN \pm (A:B + C + CIN)$	P カスケード加算/加算
0x20	010 00 00	$P \pm CIN$	P を保持
0x22	010 00 10	$P \pm (P + CIN)$	ダブル フィードバック加算
0x23	010 00 11	$P \pm (A:B + CIN)$	フィードバック加算
0x25	010 01 01	$P \pm (A \times B + CIN)$	乗算/累積
0x2c	010 11 00	$P \pm (C + CIN)$	フィードバック加算
0x2e	010 11 10	$P \pm (C + P + CIN)$	ダブル フィードバック加算
0x2f	010 11 11	$P \pm (A:B + C + CIN)$	フィードバック加算/加算
0x30	011 00 00	$C \pm CIN$	C セレクト
0x32	011 00 10	$C \pm (P + CIN)$	フィードバック加算
0x33	011 00 11	$C \pm (A:B + CIN)$	36 ビットの加算器
0x35	011 01 01	$C \pm (A \times B + CIN)$	乗算/加算
0x3c	011 11 00	$C \pm (C + CIN)$	ダブル
0x3e	011 11 10	$C \pm (C + P + CIN)$	ダブル加算フィードバック加算
0x3f	011 11 11	$C \pm (A:B + C + CIN)$	ダブル加算
0x50	101 00 00	$Shift(PCIN) \pm CIN$	17 ビット シフト P カスケード セレクト
0x52	101 00 10	$Shift(PCIN) \pm (P + CIN)$	17 ビット シフト P カスケード フィードバック加算
0x53	101 00 11	$Shift(PCIN) \pm (A:B + CIN)$	17 ビット シフト P カスケード加算
0x55	101 01 01	$Shift(PCIN) \pm (A \times B + CIN)$	17 ビット シフト P 乗算/加算
0x5c	101 11 00	$Shift(PCIN) \pm (C + CIN)$	17 ビット シフト P カスケード加算
0x5e	101 11 10	$Shift(PCIN) \pm (C + P + CIN)$	17 ビット シフト P カスケード フィードバック加算/加算
0x5c	101 11 11	$Shift(PCIN) \pm (A:B + C + CIN)$	17 ビット シフト P カスケード加算/加算
0x60	110 00 00	$Shift(P) \pm CIN$	17 ビット シフト フィードバック
0x62	110 00 10	$Shift(P) \pm (P + CIN)$	17 ビット シフト フィードバック/フィードバック加算
0x63	110 00 11	$Shift(P) \pm (A:B + CIN)$	17 ビット シフト フィードバック加算

16 進 OpMode	2 進 OpMode	加減算器の出力	説明
[6:0]	Z Y X		
0x65	110 01 01	Shift(P) \pm (A \times B+CIN)	17 ビット シフト フィードバック乗算/加算
0x6c	110 11 00	Shift(P) \pm (C + CIN)	17 ビット シフト フィードバック加算
0x6e	110 11 10	Shift(P) \pm (C + P + CIN)	17 ビット シフト フィードバック/フィードバック加算/加算
0x6f	110 11 11	Shift(P) \pm (A:B + C + CIN)	17 ビット シフト フィードバック加算/加算

キャリー ロジックでサポートされる繰り上げ/繰り下げモード

OpMode 入力、3 入力加減算器へのデータ入力、減算制御ビットに加え、加減算器の出力はキャリー入力ロジックの結果を示します。

CarryInSel 信号、減算制御信号、および OpMode 制御信号は、コンフィギュレーション RAM で制御してレジスタに取り込むことができます (グレーのマルチプレクサ シンボルで示す)。これにより、制御信号パイプラインの遅延をデータのパイプライン遅延と一致させることができます。CarryInSel 信号、減算制御信号、OpMode 制御信号は同じリセット信号 (RSTCTRL) を使用し、減算制御信号と OpMode 制御信号は同じクロック イネーブル信号を使用します。クロック イネーブルを使用すると、必要に応じて、制御信号をディスエーブルにできます。

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、2	1	A 入力にレジスタを付けるかを指定します。
B_INPUT	文字列	DIRECT、CASCADE	DIRECT	DIRECT = 被乗数は B、CASCADE = 被乗数は BCIN
BREG	整数	0、1、2	1	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 入力のパイプライン レジスタ数
CARRYINSELREG	整数	0、1	1	CARRYINSEL のパイプライン レジスタ数
CREG	整数	0、1、2	1	C 入力にレジスタを付けるかを指定します。
LEGACY_MODE	文字列	NONE、MULT18X18、MULT18X18S	MULT18X18S	DCM の内部属性設定。デフォルト値から変更しないでください。
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 入力上のパイプライン レジスタ数
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。
SUBTRACTREG	整数	0、1	1	SUBTRACT 入力上のパイプライン レジスタ数

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;

-- DSP48: DSP Function Block
--      Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

DSP48_inst : DSP48
generic map (
  AREG => 1,  -- Number of pipeline registers on the A input, 0, 1 or 2
  BREG => 1,  -- Number of pipeline registers on the B input, 0, 1 or 2
  B_INPUT => "DIRECT", -- B input DIRECT from fabric or CASCADE from another DSP48
  CARRYINREG => 1,    -- Number of pipeline registers for the CARRYIN input, 0 or 1
  CARRYINSELREG => 1, -- Number of pipeline registers for the CARRYINSEL, 0 or 1
  CREG => 1,    -- Number of pipeline registers on the C input, 0 or 1
  LEGACY_MODE => "MULT18X18S", -- Backward compatibility, NONE, MULT18X18 or MULT18X18S
  MREG => 1,    -- Number of multiplier pipeline registers, 0 or 1
  OPMODEREG => 1, -- Number of pipeline registers on OPMODE input, 0 or 1
  PREG => 1,    -- Number of pipeline registers on the P output, 0 or 1
  SUBTRACTREG => 1) -- Number of pipeline registers on the SUBTRACT input, 0 or 1
port map (
  BCOUT => BCOUT,  -- 18-bit B cascade output
  P => P,          -- 48-bit product output
  PCOUT => PCOUT,  -- 48-bit cascade output
  A => A,          -- 18-bit A data input
  B => B,          -- 18-bit B data input
  BCIN => BCIN,    -- 18-bit B cascade input
  C => C,          -- 48-bit cascade input
  CARRYIN => CARRYIN, -- Carry input signal
  CARRYINSEL => CARRYINSEL, -- 2-bit carry input select
  CEA => CEA,      -- A data clock enable input
  CEB => CEB,      -- B data clock enable input
  CEC => CEC,      -- C data clock enable input
  CECARRYIN => CECARRYIN, -- CARRYIN clock enable input
  CECINSUB => CECINSUB, -- CINSUB clock enable input
  CECTRL => CECTRL, -- Clock Enable input for CTRL registersL
  CEM => CEM,      -- Clock Enable input for multiplier registers
  CEP => CEP,      -- Clock Enable input for P registers
  CLK => CLK,      -- Clock input
  OPMODE => OPMODE, -- 7-bit operation mode input
  PCIN => PCIN,    -- 48-bit PCIN input
  RSTA => RSTA,    -- Reset input for A pipeline registers
  RSTB => RSTB,    -- Reset input for B pipeline registers
  RSTC => RSTC,    -- Reset input for C pipeline registers
  RSTCARRYIN => RSTCARRYIN, -- Reset input for CARRYIN registers
  RSTCTRL => RSTCTRL, -- Reset input for CTRL registers
  RSTM => RSTM,    -- Reset input for multiplier registers
  RSTP => RSTP,    -- Reset input for P pipeline registers
  SUBTRACT => SUBTRACT -- SUBTRACT input
);

-- End of DSP48_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// DSP48: DSP Function Block
//      Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

DSP48 #(
    .AREG(1),           // Number of pipeline registers on the A input, 0, 1 or 2
    .BREG(1),           // Number of pipeline registers on the B input, 0, 1 or 2
    .B_INPUT("DIRECT"), // B input DIRECT from fabric or CASCADE from another DSP48
    .CARRYINREG(1),     // Number of pipeline registers for the CARRYIN input, 0 or 1
    .CARRYINSELREG(1),  // Number of pipeline registers for the CARRYINSEL, 0 or 1
    .CREG(1),           // Number of pipeline registers on the C input, 0 or 1
    .LEGACY_MODE("MULT18X18S"), // Backward compatibility, NONE, MULT18X18 or MULT18X18S
    .MREG(1),           // Number of multiplier pipeline registers, 0 or 1
    .OPMODEREG(1),      // Number of pipeline registers on OPMODE input, 0 or 1
    .PREG(1),           // Number of pipeline registers on the P output, 0 or 1
    .SUBTRACTREG(1)     // Number of pipeline registers on the SUBTRACT input, 0 or 1
) DSP48_inst (
    .BCOUT(BCOUT), // 18-bit B cascade output
    .P(P),         // 48-bit product output
    .PCOUT(PCOUT), // 48-bit cascade output
    .A(A),         // 18-bit A data input
    .B(B),         // 18-bit B data input
    .BCIN(BCIN),   // 18-bit B cascade input
    .C(C),         // 48-bit cascade input
    .CARRYIN(CARRYIN), // Carry input signal
    .CARRYINSEL(CARRYINSEL), // 2-bit carry input select
    .CEA(CEA),     // A data clock enable input
    .CEB(CEB),     // B data clock enable input
    .CEC(CEC),     // C data clock enable input
    .CECARRYIN(CECARRYIN), // CARRYIN clock enable input
    .CECINSUB(CECINSUB), // CINSUB clock enable input
    .CECTRL(CECTRL), // Clock Enable input for CTRL registers
    .CEM(CEM),     // Clock Enable input for multiplier registers
    .CEP(CEP),     // Clock Enable input for P registers
    .CLK(CLK),     // Clock input
    .OPMODE(OPMODE), // 7-bit operation mode input
    .PCIN(PCIN),   // 48-bit PCIN input
    .RSTA(RSTA),   // Reset input for A pipeline registers
    .RSTB(RSTB),   // Reset input for B pipeline registers
    .RSTC(RSTC),   // Reset input for C pipeline registers
    .RSTCARRYIN(RSTCARRYIN), // Reset input for CARRYIN registers
    .RSTCTRL(RSTCTRL), // Reset input for CTRL registers
    .RSTM(RSTM),   // Reset input for multiplier registers
    .RSTP(RSTP),   // Reset input for P pipeline registers
    .SUBTRACT(SUBTRACT) // SUBTRACT input
);

// End of DSP48_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

EMAC

プリミティブ : Fully integrated 10/100/1000 Mb/s Ethernet Media Access Controller (Ethernet MAC)

概要

このデザイン エLEMENTでは、Virtex®-4 PowerPC® プロセッサをイーサネット接続できるようにします。イーサネット MAC (EMAC) では、次の機能がサポートされています。

- 完全に統合された 10/100/1000 Mb/s イーサネット MAC
- IEEE 802.3-2002 規格に準拠
- 全二重または半二重操作を設定可能
- 物理 (PHY) レイヤでオブジェクトを制御する MII (Media Independent Interface) 制御 (MDIO) インターフェイス
- 統計ベクタ出力にアクセス可能
- VLAN フレームをサポート
- 内部フレーム ギャップ調整を設定可能
- 送信および受信パスの両方を通過するインバンド フレーム チェック シーケンス (FCS) フィールドを設定可能
- 送信で自動パディング、受信で FCS フィールド ストリップを提供
- ホスト インターフェイスから設定およびモニタ可能
- ハードウェアで選択可能なデバイス コントロール レジスタ (DCR) バスまたは 1G イーサネット MAC バス ホスト インターフェイス
- イーサネット MAC 制御 PAUSE フレームを使用したフロー制御を設定可能 (対称または非対称にイネーブル可能)
- 任意の長さのジャンボ フレームをサポート
- ユニキャスト、マルチキャスト、ブロードキャスト アドレス用の受信アドレス フィルタを設定可能
- II (Media Independent Interface)、GMII (Gigabit Media Independent Interface)、および Reduced Gigabit Media Independent Interface (RGMII)
- マルチギガビット トランシーバ (MGT) で使用して完全なオンチップ 1000BASE-X インプリメンテーションを提供するための 1000BASE-X PCS (Physical Coding Sublayer) および PMA (Physical Medium Attachment) サブレイヤを含有
- MGT インターフェイスから銅製の外部 PHY レイヤへの SGMII (Serial Gigabit Media Independent Interface) をサポート

ポートの説明

入力	出力
RESET	
TIEEMAC0CONFIGVEC [79:0]	
TIEEMAC1CONFIGVEC [79:0]	
TIEEMAC0UNICASTADDR [47:0]	
TIEEMAC1UNICASTADDR [47:0]	
PHYEMAC0GTCLK	
PHYEMAC1GTCLK	
CLIENTEMAC0DCMLOCKED	EMAC0CLIENTANINTERRUPT
CLIENTEMAC1DCMLOCKED	EMAC1CLIENTANINTERRUPT

入力	出力
CLIENTEMAC0RXCLIENTCLKIN	EMAC0CLIENTRXCLIENTCLKOUT
	EMAC0CLIENTRXD [15:0]
	EMAC0CLIENTRXDVLD
	EMAC0CLIENTRXDVLDMSW
	EMAC0CLIENTRXGOODFRAME
	EMAC0CLIENTRXBADFRAME
	EMAC0CLIENTRXFRAMEDROP
	EMAC0CLIENTRXDVREG6
	EMAC0CLIENTRXSTATS [6:0]
	EMAC0CLIENTRXSTATSBYTEVLD
	EMAC0CLIENTRXSTATSVLD
CLIENTEMAC1RXCLIENTCLKIN	EMAC1CLIENTRXCLIENTCLKOUT
	EMAC1CLIENTRXD [15:0]
	EMAC1CLIENTRXDVLD
	EMAC1CLIENTRXDVLDMSW
	EMAC1CLIENTRXGOODFRAME
	EMAC1CLIENTRXBADFRAME
	EMAC1CLIENTRXFRAMEDROP
	EMAC1CLIENTRXDVREG6
	EMAC1CLIENTRXSTATS [6:0]
	EMAC1CLIENTRXSTATSBYTEVLD
	EMAC1CLIENTRXSTATSVLD
CLIENTEMAC0TXGMIIMICLKIN	EMAC0CLIENTTXGMIIMICLKOUT
CLIENTEMAC0TXCLIENTCLKIN	EMAC0CLIENTTXCLIENTCLKOUT
CLIENTEMAC0TXD [15:0]	EMAC0CLIENTTXACK
CLIENTEMAC0TXDVLD	EMAC0CLIENTTXCOLLISION
CLIENTEMAC0TXDVLDMSW	EMAC0CLIENTTXRETRANSMIT
CLIENTEMAC0TXUNDERRUN	EMAC0CLIENTTXSTATS
CLIENTEMAC0TXIFGDELAY [7:0]	EMAC0CLIENTTXSTATSBYTEVLD
CLIENTEMAC0TXFIRSTBYTE	EMAC0CLIENTTXSTATSVLD
CLIENTEMAC1TXGMIIMICLKIN	EMAC1CLIENTTXGMIIMICLKOUT
CLIENTEMAC1TXCLIENTCLKIN	EMAC1CLIENTTXCLIENTCLKOUT
CLIENTEMAC1TXD [15:0]	EMAC1CLIENTTXACK
CLIENTEMAC1TXDVLD	EMAC1CLIENTTXCOLLISION
CLIENTEMAC1TXDVLDMSW	EMAC1CLIENTTXRETRANSMIT
CLIENTEMAC1TXUNDERRUN	EMAC1CLIENTTXSTATS
CLIENTEMAC1TXIFGDELAY [7:0]	EMAC1CLIENTTXSTATSBYTEVLD
CLIENTEMAC1TXFIRSTBYTE	EMAC1CLIENTTXSTATSVLD

入力	出力
CLIENTEMAC0PAUSEREQ	
CLIENTEMAC0PAUSEVAL [15:0]	
CLIENTEMAC1PAUSEREQ	
CLIENTEMAC1PAUSEVAL [15:0]	
HOSTADDR [9:0]	HOSTMIIMRDY
HOSTCLK	HOSTRDDATA [31:0]
HOSTMIIMSEL	
HOSTOPCODE [1:0]	
HOSTREQ	
HOSTWRDATA [31:0]	
HOSTEMAC1SEL	
DCREMACCLK	DCRHOSTDONEIR
DCREMACENABLE	EMACDCRACK
DCREMACDBUS [0:31]	EMACDCRDBUS [0:31]
DCREMACABUS [8:9]	
DCREMACREAD	
DCREMACWRITE	
PHYEMAC0RXCLK	EMAC0PHYTXCLK
PHYEMAC0RXD [7:0]	EMAC0PHYTXD [7:0]
PHYEMAC0RXDV	EMAC0PHYTXEN
PHYEMAC0RXER	EMAC0PHYTXER
PHYEMAC0MIITXCLK	
PHYEMAC0COL	
PHYEMAC0CRS	
PHYEMAC1RXCLK	EMAC1PHYTXCLK
PHYEMAC1RXD [7:0]	EMAC1PHYTXD [7:0]
PHYEMAC1RXDV	EMAC1PHYTXEN
PHYEMAC1RXER	EMAC1PHYTXER
PHYEMAC1MIITXCLK	
PHYEMAC1COL	
PHYEMAC1CRS	
PHYEMAC0SIGNALDET	EMAC0PHYENCOMMAALIGN
PHYEMAC0PHYAD [4:0]	EMAC0PHYLOOPBACKMSB
PHYEMAC0RXCLKCORCNT [2:0]	EMAC0PHYMGTRXRESET
PHYEMAC0RXBUFSTATUS [1:0]	EMAC0PHYMGTTXRESET
PHYEMAC0RXCHARISCOMMA	EMAC0PHYPOWERDOWN
PHYEMAC0RXCHARISK	EMAC0PHYSYNACQSTATUS
PHYEMAC0RXCHECKINGCRC	EMAC0PHYTXCHARDISPMODE

入力	出力
PHYEMAC0RXCOMMADET	EMAC0PHYTXCHARDISPVAL
PHYEMAC0RXDISPERR	EMAC0PHYTXCHARISK
PHYEMAC0RXLOSSOFSYNC [1:0]	
PHYEMAC0RXNOTINTABLE	
PHYEMAC0RXRUNDISP	
PHYEMAC0RXBUFERR	
PHYEMAC0TXBUFERR	
PHYEMAC1SIGNALDET	EMAC1PHYENCOMMAALIGN
PHYEMAC1PHYAD [4:0]	EMAC1PHYLOOPBACKMSB
PHYEMAC1RXCLKCORCNT [2:0]	EMAC1PHYMGTRXRESET
PHYEMAC1RXBUFSTATUS [1:0]	EMAC1PHYMGTTXRESET
PHYEMAC1RXCHARISCOMMA	EMAC1PHYPOWERDOWN
PHYEMAC1RXCHARISK	EMAC1PHYSYNACQSTATUS
PHYEMAC1RXCHECKINGCRC	EMAC1PHYTXCHARDISPMODE
PHYEMAC1RXCOMMADET	EMAC1PHYTXCHARDISPVAL
PHYEMAC1RXDISPERR	EMAC1PHYTXCHARISK
PHYEMAC1RXLOSSOFSYNC [1:0]	
PHYEMAC1RXNOTINTABLE	
PHYEMAC1RXRUNDISP	
PHYEMAC1RXBUFERR	
PHYEMAC1TXBUFERR	
PHYEMAC0MCLKIN	EMAC0PHYMCLKOUT
PHYEMAC0MDIN	EMAC0PHYMDOUT
	EMAC0PHYMDTRI
PHYEMAC1MCLKIN	EMAC1PHYMCLKOUT
PHYEMAC1MDIN	EMAC1PHYMDOUT
	EMAC1PHYMDTRI

デザインの入力方法

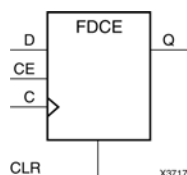
インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

FDCE

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



概要

このデザイン エLEMENTは、クロック イネーブルと非同期クリアがある単一の D タイプ フリップフロップです。クロック イネーブル (CE) が High、非同期クリア (CLR) が Low の場合、クロック (C) が Low から High に切り替わる時にデータ入力 (D) の値がデータ出力 (Q) に送られます。CLR が High になると、ほかのすべての入力は無視され、出力 (Q) の値が Low にリセットされます。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	CE	D	C	Q
1	X	X	X	0
0	0	X	X	変化なし
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	0	コンフィギュレーション後の Q 出力の初期値を指定。 Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDCE の場合、INIT 値は 0 である必要があります。1 に設定する場合は、この動作を表す非同期回路を作成する必要がありますが、ザイリンクスでは推奨されていません。

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;

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

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

-- End of FDCE_inst instantiation
```

Verilog 記述 (インスタンスエーション)

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

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

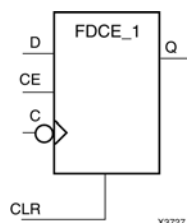
// End of FDCE_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

FDCE_1

プリミティブ : D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous 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)、非同期クリア (CLR) の各入力とデータ出力 (Q) のある単一の D タイプ フリップフロップです。非同期 CLR が High になると、ほかのすべての入力は無視され、Q 出力が Low になります。CLR が Low、CE が High の場合、クロック (C) が High から Low 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	CE	D	C	Q
1	X	X	X	0
0	0	X	X	変化なし
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;

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

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

-- End of FDCE_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

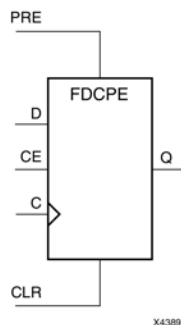
// End of FDCE_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

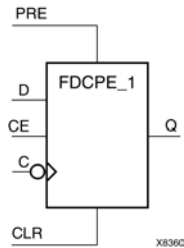
// End of FDCPE_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

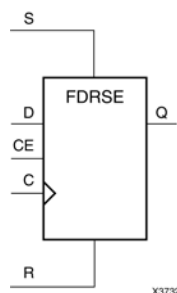
// End of FDCPE_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

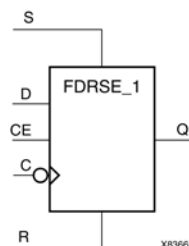
// End of FDRSE_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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 に切り替わるときに、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 に切り替わるとき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。

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
);

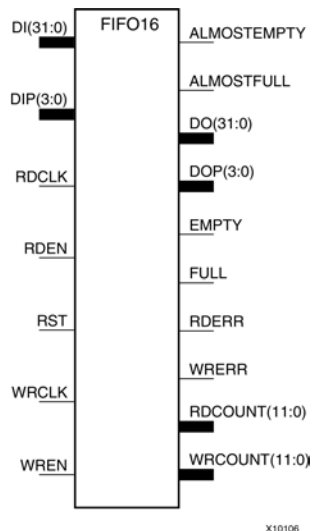
// End of FDRSE_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

FIFO16

プリミティブ : Virtex-4 Block RAM Based, Built-In FIFO



概要

多くの FPGA デザインでは、ブロック RAM を使用して FIFO をインプリメントします。Virtex®-4 アーキテクチャでは、ブロック RAM に含まれる専用ロジックにより、同期または非同期 FIFO を簡単にインプリメントできます。カウンタ、コンパレータ、またはステータスフラグの生成に追加の CLB ロジックを使用することなく、1 つのブロック RAM リソースで 1 つの FIFO を作成できます。標準モードと FWFT (First Word Fall Through) モードがサポートされています。

標準モード : 空の FIFO に最初のワードが書き込まれると、EMPTY フラグは RDCLK に同期してデアサートされます。EMPTY フラグが Low にデアサートされ、RDEN がアサートされると、RDCLK の立ち上がりエッジで DOUT に最初のワードが出力されます。

FWFT (First Word Fall Through) モード : 空の FIFO に最初のワードが書き込まれると、RDEN がアサートされなくても、その値が RDCLK の数サイクル後に DOUT に出力されます。この後読み出し操作を行うには、EMPTY が Low、RDEN が High になっている必要があります。

メモ : 独立したクロックでデュアル クロック モードを使用する場合、読み出しクロック エッジと書き込みクロック エッジ間のオフセットによっては、Empty、Almost Empty、Full、および Almost Full フラグが 1 クロック サイクル後にデアサートされることがあります。クロックが非同期のため、シミュレーション モデルではアーキテクチャのユーザー ガイドに示されているデアサートレイテンシ サイクルのみが反映されます。

次の表に、2 つのモードでの FIFO の容量を示します。

標準モード	FWFT モード
4k + 1 エントリ X 4 ビット	4k + 2 エントリ X 4 ビット
2k + 1 エントリ X 9 ビット	2k + 2 エントリ X 9 ビット
1k + 1 エントリ X 18 ビット	1k + 2 エントリ X 18 ビット
512 + 1 エントリ X 36 ビット	512 + 2 エントリ X 36 ビット

ブロック RAM は、同期または非同期操作用に、別々の読み出し/書き込みクロックを使用した非同期 FIFO メモリとしてコンフィギュレーションできます。ブロック RAM のポート A は FIFO 読み出しポートとして、ポート B は FIFO 書き込みポートとして使用します。データは、読み出しクロックの立ち上がりエッジで FIFO から読み出され、書き込みクロックの立ち上がりエッジで FIFO に書き込まれます。FIFO モードでは、読み出しポートと書き込みポートの幅を別々に選択できません。

使用可能なステータス フラグは、次のとおりです。

- ・ **フル (FULL)** : WRCLK に同期しています。FIFO にデータを書き込むスペースがなくなると、FULL がアサートされます。FIFO がフルになると、書き込みポインタは停止します。これにより、読み出しポインタと書き込みポインタにより同じエントリが指定され、オーバーフローは発生しません。FULL フラグは出力でレジスタを介し、1 書き込みクロック サイクル後にアサートされます。最後のエントリが読み出されると、3 クロック サイクル以内に WRCLK に同期してディアサートされます。
- ・ **空 (EMPTY)** : RDCLK に同期しています。
- ・ **ほぼフル (AFULL)** : WRCLK に同期しています。FIFO の空きスペースが ALMOST_FULL_OFFSET で指定した値よりも少なくなるとアサートされ、書き込みを停止するよう警告します。FIFO の空きスペースが ALMOST_FULL_OFFSET で指定した値より大きくなると、WRCLK に同期してディアサートされます。
- ・ **ほぼ空 (AEMPTY)** : RDCLK に同期しています。
- ・ **書き込みカウント (WRCOUNT)** : WRCLK に同期しています。
- ・ **書き込みエラー (WRERR)** : WRCLK に同期しています。FULL フラグがアサートされた後に書き込みを実行すると、WRERR フラグがアサートされます。書き込みイネーブルまたは FULL が Low にディアサートされると、WRERR はディアサートされます。この信号は、WRCLK に同期しています。
- ・ **読み出しカウント (RDCOUNT)** : RDCLK に同期しています。
- ・ **読み出しエラー (RDERR)** : RDCLK に同期しています。

ポートの説明

ポート名	方向	機能
DI	入力	データ入力
DIP	入力	パリティビット入力
WREN	入力	ライト イネーブル。WREN = 1 の場合、データがメモリに書き込まれます。WREN = 0 の場合、書き込みはディスエーブルになります。
WRCLK	入力	書き込みクロック
RDEN	入力	読み出しイネーブル。RDEN = 1 の場合、データが出力レジスタに読み出されます。RDEN = 0 の場合、読み出しはディスエーブルになります。
RDCLK	入力	読み出しクロック
RESET	入力	FIFO 機能、フラグ、ポインタの非同期リセット
DO	出力	データ出力 (RDCLK に同期)
DOP	出力	パリティビット出力 (RDCLK に同期)
FULL	出力	FIFO メモリのすべてのエントリがフル
ALMOSTFULL	出力	FIFO メモリのほぼすべてのエントリがフル。WRCLK に同期しています。値はユーザーが設定できます。
EMPTY	出力	FIFO が空です。この出力がアサートされると、読み出しは行われません。RDCLK に同期しています。
ALMOSTEMPTY	出力	FIFO のほぼすべての有効エントリが読み出されたことを示します。RDCLK に同期しています。値はユーザーが設定できます。
RDCOUNT	出力	FIFO データ読み出しポインタ。RDCLK に同期しています。最大読み出しポインタ値に達すると、0 に戻ります。
WRCOUNT	出力	FIFO データ書き込みポインタ。WRCLK に同期しています。最大書き込みポインタ値に達すると、0 に戻ります。
WRERR	出力	FIFO がフルのときに書き込みを行うと、アサートされます。WRCLK に同期しています。
RDERR	出力	FIFO が空のときに読み出しを行うと、アサートされます。RDCLK に同期しています。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
ALMOST_EMPTY_OFFSET	16 進数	12 ビット値	すべてゼロ	ほぼ空の状態を検出するしきい値を指定
ALMOST_FULL_OFFSET	16 進数	12 ビット値	すべてゼロ	ほぼフルの状態を検出するしきい値を指定
DATA_WIDTH	整数	4、9、18、36	36	データ幅を指定
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	FIFO FWFT をオン/オフに設定

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;

-- FIFO16: Virtex-4 4k deep x 4 wide BlockRAM Asynchronous FIFO
-- Xilinx HDL Libraries Guide, version 11.2

FIFO16_inst : FIFO16
generic map (
    ALMOST_FULL_OFFSET => X"080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"080", -- Sets the almost empty threshold
    DATA_WIDTH => 4, -- Sets data width to 4, 9, 18, or 36
    FIRST_WORD_FALL_THROUGH => FALSE) --Sets the FIFO FWFT to TRUE or FALSE
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO (31 DOWNTO 4) => unconnected (27 downto 0), -- Unused data output. Unconnected is a signal of 32 bits
    DO (3 DOWNTO 0) => DO, -- 4-bit data output
    DOP => unconnected (31 downto 28), -- 4-bit Unused parity data output. Unconnected is a signal of 32 bits
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 12-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 12-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI (31 DOWNTO 4) => X"0000000", -- Unused data inputs tied to ground
    DI (3 downto 0) => DI, -- 4-bit data input
    DIP => X"0", -- 4-bit Unused parity inputs tied to ground
    RDCLK => RDCLK, -- 1-bit read clock input
    RDEN => RDEN, -- 1-bit read enable input
    RST => RST, -- 1-bit reset input
    WRCLK => WRCLK, -- 1-bit write clock input
    WREN => WREN -- 1-bit write enable input
);

-- End of FIFO16_inst instantiation

```

Verilog 記述 (インスタンスエーション)

```
// FIFO16: Virtex-4 BlockRAM Asynchronous FIFO configured for 4k deep x 4 wide
// Xilinx HDL Libraries Guide, version 11.2

wire [27:0] unconnected;

FIFO16 #(
    .ALMOST_FULL_OFFSET(12'h080),      // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(12'h080),     // Sets the almost empty threshold
    .DATA_WIDTH(4),                    // Sets data width to 4, 9, 18, or 36
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO16_4kx4_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY),        // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL),          // 1-bit almost full output flag
    .DO({unconnected[27:0], DO}),      // 4-bit data output
    .EMPTY(EMPTY),                     // 1-bit empty output flag
    .FULL(FULL),                       // 1-bit full output flag
    .RDCOUNT(RDCOUNT),                 // 12-bit read count output
    .RDERR(RDERR),                     // 1-bit read error output
    .WRCOUNT(WRCOUNT),                 // 12-bit write count output
    .WRERR(WRERR),                     // 1-bit write error
    .DI({28'h0000000, DI}),            // 4-bit data input (rest tied to ground)
    .DIP(4'h0),                        // Parity bits tied to Ground
    .RDCLK(RDCLK),                     // 1-bit read clock input
    .RDEN(RDEN),                       // 1-bit read enable input
    .RST(RST),                         // 1-bit reset input
    .WRCLK(WRCLK),                     // 1-bit write clock input
    .WREN(WREN)                        // 1-bit write enable input
);

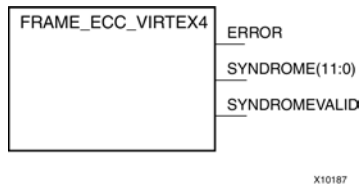
// End of FIFO16_4kx4_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

FRAME_ECC_VIRTEX4

プリミティブ : Reads a Single, Virtex®-4 Configuration Frame and Computes a Hamming, Single-Error Correction, Double-Error Detection Syndrome



概要

このデザイン エLEMENTは、Virtex®-4 の 1312 ビットのコンフィギュレーション フレームを 32 ビットずつ読み出します。その後、ハミング エラー訂正コード、ダブル エラー検出シンドローム値を算出します。これにより、フレーム ビットの 1 つにエラーがある場合に、訂正が必要なが示されます。2 ビット エラーがあることも検出されますが、これは訂正できません。FRAME_ECC_VIRTEX4 プリミティブでは、変更されたビットは修正されません。

ポートの説明

ポート名	方向	幅	機能
ERROR	出力	1	エラー出力。エラーが存在することを示します。
SYNDROME	出力	12	エラー ビットの位置。エラーの位置と、0、1、または 2 ビット エラーが存在することを示します。
SYNDROMEVALID	出力	1	High の場合、フレームに 0、1、または 2 ビット エラーがあることを示します。フレーム リードバックが終了すると High にアサートされます。

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

SYNDROME の値とエラー ステータスの関係

SYNDROME ビット 11	SYNDROME ビット 10 ~ 0	エラー ステータス
0	すべて 0	ビット エラーなし
0	0 以外	1 ビット エラーが存在 (SYNDROME 値はエラー ビットの位置を示す)
1	すべて 0	2 ビット エラーが存在 (訂正不可)

メモ : 上記の表の値は、SYNDROME_VALID が High の場合にのみ適用されます。

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;

-- FRAME_ECC_VIRTEX4: Configuration Frame Error Correction Circuitry
--                               Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

FRAME_ECC_VIRTEX4_inst : FRAME_ECC_VIRTEX4
port map (
    ERROR => ERROR,
    SYNDROME => SYNDROME,
    SYNDROMEVALID => SYNDROMEVALID
);

-- End of FRAME_ECC_VIRTEX4_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// FRAME_ECC_VIRTEX4: Configuration Frame Error Correction Circuitry
//                               Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

FRAME_ECC_VIRTEX4 FRAME_ECC_VIRTEX4_inst (
    .ERROR(ERROR), // 1-bit output indicating an error
    .SYNDROME(SYNDROME), // 12-bit output location of erroroneous bit
    .SYNDROMEVALID(SYNDROMEVALID) // 1-bit output indicating 0, 1 or 2 bit errors in frame
);

// End of FRAME_ECC_VIRTEX4_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

GT11_CUSTOM

プリミティブ : RocketIO MGTs with 622 Mb/s to 11.1 Gb/s Data Rates, 8 to 24 Transceivers per FPGA, and 2.5 GHz 5.55 GHz VCO, Less Than 1ns RMS Jitter

概要

このデザイン エLEMENTは、RocketIO™ MGT です。RocketIO MGT は、マルチギガビット シリアル トランシーバを Virtex®-4 デザインに簡単に組み込むための柔軟な機能を提供します。次の機能がサポートされています。

- ・ 10.3Gb/s データ レート
- ・ 1 つの FPGA に 8 ~ 24 個の トランシーバ を搭載
- ・ 2.5GHz ~ 5.55GHz VCO で動作し、RMS のジッタは 1ns 未満
- ・ トランスミッタのプリエンファシス
- ・ レシーバ連続時間の均等化
- ・ オンチップ AC カップリング レシーバ (オプションでバイパス可)
- ・ レシーバ信号の検出、信号損失インジケータ、バンド外信号レシーバ
- ・ バンド外信号に対し、Vcm の両方の出力にドライバ アイドル ステートを送信
- ・ 8B/10B または 64B/66B エンコード、あるいはデータ エンコードなし (パス スルー モード)
- ・ チャネル ボンディング
- ・ 柔軟な CRC (Cyclic Redundancy Check) の生成およびチェック
- ・ トランスミッタおよびレシーバ終端電圧用のピン
- ・ セカンダリ (ダイナミック) コンフィギュレーション バスを使用したリコンフィギュレーション
- ・ PMA RX-TX パスを含む複数のループバック パス

RocketIO MGT は、FX デバイスでのみ使用可能です。

論理表

入力	出力
CHBONDI [4:0]	DRDY
CSUPMARESET	RXBUFERR
DADDR [7:0]	RXCALFAIL
DCLK	RXCOMMADET
DEN	RXCYLELIMIT
DI [15:0]	RXLOCK
DWE	RXRealIGN
ENCHANSYNC	RXRECCLK1
ENMCOMMAALIGN	RXBCLK
ENPCOMMAALIGN	RXRECCLK2
GREFCLK	RXSIGDET
LOOPBACK [1:0]	TX1N
POWERDOWN	TX1P
REFCLK1	TXBUFERR

入力	出力
REFCLK2	TXCALFAIL
RX1N	TXCYCLELIMIT
RX1P	TXLOCK
RXBLOCKSYNC64B66BUSE	DO [15:0]
RXCLKSTABLE	RXLOSSOFSYNC [1:0]
RXCOMMADETUSE	RXCRCOUT [31:0]
RXCRCCLK	TXCRCOUT [31:0]
RXCRCDATAVALID	CHBONDO [4:0]
RXCRCDATAWIDTH [2:0]	RXSTATUS [5:0]
RXCRCIN [63:0]	RXDATA [63:0]
RXCRCINIT	RXCHARISCOMMA [7:0]
RXCRCINTCLK	RXCHARISK [7:0]
RXCRCPD	RXDISPERR [7:0]
RXCRCRESET	RXNOTINTABLE [7:0]
RXDATAWIDTH [1:0]	RXRUNDISP [7:0]
RXDEC64B66BUSE	TXRUNDISP [7:0]
RXDEC8B10BUSE	TXKERR [7:0]
RXDESCRAM64B66BUSE	
RXIGNOREBTF	
RXINTDATAWIDTH [1:0]	
RXPMARESET	
RXPOLARITY	
RXRESET	
RXSLIDE	
RXUSRCLK	
RXUSRCLK2	
TXBYPASS8B10B [7:0]	
TXCHARDISPMODE [7:0]	
TXCHARDISPVAL [7:0]	
TXCHARISK [7:0]	
TXCLKSTABLE	
TXCRCCLK	
TXCRCDATAVALID	
TXCRCDATAWIDTH [2:0]	
TXCRCIN [63:0]	
TXCRCINIT	
TXCRCINTCLK	

入力	出力
TXCRCPD	
TXCRCRESET	
TXDATA [63:0]	
TXDATAWIDTH [1:0]	
TXENC64B66BUSE	
TXENC8B10BUSE	
TXENOOB	
TXGEARBOX64B66BUSE	
TXINHIBIT	
TXINTDATAWIDTH [1:0]	
TXPMARESET	
TXPOLARITY	
TXRESET	
TXSCRAM64B66BUSE	
TXSYNC	
TXUSRCLK	
TXUSRCLK2	

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

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

GT11_DUAL

プリミティブ : RocketIO MGT Tile (contains 2 GT11_CUSTOM) with 622 Mb/s to 11.1 Gb/s data rates, 8 to 24 transceivers per FPGA, and 2.5 GHz 5.55 GHz VCO, less than 1ns RMS jitter

概要

RocketIO™ MGT は、マルチギガビット シリアル トランシーバを Virtex®-4 デザインに簡単に組み込むための柔軟な機能を提供します。RocketIO MGT では、次の機能がサポートされています。

- ・ 622Mb/s ~ 11.1Gb/s データ レート
- ・ 1 つの FPGA に 8 ~ 24 個の トランシーバ を搭載
- ・ 2.5GHz ~ 5.55GHz VCO で動作し、RMS のジッタは 1ns 未満
- ・ トランスミッタのプリエンファシス
- ・ レシーバ連続時間の均等化
- ・ オンチップ AC カップリング レシーバ
- ・ 2.5Gb/s までのデータ レート用のデジタル オーバーサンプリング レシーバ
- ・ レシーバ信号の検出、信号損失インジケータ、バンド外信号レシーバ
- ・ バンド外信号に対し、Vcm の両方の出力にドライバ アイドル ステートを送信
- ・ 8B/10B または 64B/66B エンコード、あるいはデータ エンコードなし (パス スルー モード)
- ・ チャネル ボンディング
- ・ 柔軟な CRC (Cyclic Redundancy Check) の生成およびチェック
- ・ トランスミッタおよびレシーバ終端電圧用のピン
- ・ セカンダリ (ダイナミック) コンフィギュレーション バスを使用したリコンフィギュレーション
- ・ PMA RX-TX パスを含む複数のループバック パス

論理表

入力	出力
[1:0] LOOPBACK_A;	[1:0] RXLOSSOFSYNC_A;
[1:0] LOOPBACK_B;	[1:0] RXLOSSOFSYNC_B;
[1:0] RXDATAWIDTH_A;	[15:0] DO_A;
[1:0] RXDATAWIDTH_B;	[15:0] DO_B;
[1:0] RXINTDATAWIDTH_A;	[31:0] RXCRCOUT_A;
[1:0] RXINTDATAWIDTH_B;	[31:0] RXCRCOUT_B;
[1:0] TXDATAWIDTH_A;	[31:0] TXCRCOUT_A;
[1:0] TXDATAWIDTH_B;	[31:0] TXCRCOUT_B;
[1:0] TXINTDATAWIDTH_A;	[4:0] CHBONDO_A;
[1:0] TXINTDATAWIDTH_B;	[4:0] CHBONDO_B;
[15:0] DL_A;	[5:0] RXSTATUS_A;
[15:0] DL_B;	[5:0] RXSTATUS_B;
[2:0] RXCRCDATAWIDTH_A;	[63:0] RXDATA_A;

入力	出力
[2:0] RXCRCDATAWIDTH_B;	[63:0] RXDATA_B;
[2:0] TXCRCDATAWIDTH_A;	[7:0] RXCHARISCOMMA_A;
[2:0] TXCRCDATAWIDTH_B;	[7:0] RXCHARISCOMMA_B;
[4:0] CHBONDI_A;	[7:0] RXCHARISK_A;
[4:0] CHBONDI_B;	[7:0] RXCHARISK_B;
[63:0] RXCRCIN_A;	[7:0] RXDISPERR_A;
[63:0] RXCRCIN_B;	[7:0] RXDISPERR_B;
[63:0] TXCRCIN_A;	[7:0] RXNOTINTABLE_A;
[63:0] TXCRCIN_B;	[7:0] RXNOTINTABLE_B;
[63:0] TXDATA_A;	[7:0] RXRUNDISP_A;
[63:0] TXDATA_B;	[7:0] RXRUNDISP_B;
[7:0] DADDR_A;	[7:0] TXKERR_A;
[7:0] DADDR_B;	[7:0] TXKERR_B;
[7:0] TXBYPASS8B10B_A;	[7:0] TXRUNDISP_A;
[7:0] TXBYPASS8B10B_B;	[7:0] TXRUNDISP_B;
[7:0] TXCHARDISPMODE_A;	DRDY_A;
[7:0] TXCHARDISPMODE_B;	DRDY_B;
[7:0] TXCHARDISPVAL_A;	RXBUFERR_A;
[7:0] TXCHARDISPVAL_B;	RXBUFERR_B;
[7:0] TXCHARISK_A;	RXCALFAIL_A;
[7:0] TXCHARISK_B;	RXCALFAIL_B;
DCLK_A;	RXCOMMADET_A;
DCLK_B;	RXCOMMADET_B;
DEN_A;	RXCYLELIMIT_A;
DEN_B;	RXCYLELIMIT_B;
DWE_A;	RXLOCK_A;
DWE_B;	RXLOCK_B;
ENCHANSYNC_A;	RXMCLK_A;
ENCHANSYNC_B;	RXMCLK_B;
ENMCOMMAALIGN_A;	RXPCSHCLKOUT_A;
ENMCOMMAALIGN_B;	RXPCSHCLKOUT_B;
ENPCOMMAALIGN_A;	RXRealIGN_A;
ENPCOMMAALIGN_B;	RXRealIGN_B;
GREFCLK_A;	RXRECCLK1_A;
GREFCLK_B;	RXRECCLK1_B;
POWERDOWN_A;	RXRECCLK2_A;
POWERDOWN_B;	RXRECCLK2_B;

入力	出力
REFCLK1_A;	RXSIGDET_A;
REFCLK1_B;	RXSIGDET_B;
REFCLK2_A;	TX1N_A;
REFCLK2_B;	TX1N_B;
RX1N_A;	TX1P_A;
RX1N_B;	TX1P_B;
RX1P_A;	TXBUFERR_A;
RX1P_B;	TXBUFERR_B;
RXBLOCKSYNC64B66BUSE_A;	TXCALFAIL_A;
RXBLOCKSYNC64B66BUSE_B;	TXCALFAIL_B;
RXCLKSTABLE_A;	TXCYCLELIMIT_A;
RXCLKSTABLE_B;	TXCYCLELIMIT_B;
RXCOMMADETUSE_A;	TXLOCK_A;
RXCOMMADETUSE_B;	TXLOCK_B;
RXCRCCLK_A;	TXOUTCLK1_A;
RXCRCCLK_B;	TXOUTCLK1_B;
RXCRCDATAVALID_A;	TXOUTCLK2_A;
RXCRCDATAVALID_B;	TXOUTCLK2_B;
RXCRCINIT_A;	TXPCSHCLKOUT_A;
RXCRCINIT_B;	TXPCSHCLKOUT_B;
RXCRCINTCLK_A;	
RXCRCINTCLK_B;	
RXCRCPD_A;	
RXCRCPD_B;	
RXCRCRESET_A;	
RXCRCRESET_B;	
RXDEC64B66BUSE_A;	
RXDEC64B66BUSE_B;	
RXDEC8B10BUSE_A;	
RXDEC8B10BUSE_B;	
RXDESCRAM64B66BUSE_A;	
RXDESCRAM64B66BUSE_B;	
RXIGNOREBTF_A;	
RXIGNOREBTF_B;	
RXPMARESET_A;	
RXPMARESET_B;	
RXPOLARITY_A;	
RXPOLARITY_B;	

入力	出力
RXRESET_A;	
RXRESET_B;	
RXSLIDE_A;	
RXSLIDE_B;	
RXSYNC_A;	
RXSYNC_B;	
RXUSCLK_A;	
RXUSCLK_B;	
RXUSCLK2_A;	
RXUSCLK2_B;	
TXCLKSTABLE_A;	
TXCLKSTABLE_B;	
TXCRCCLK_A;	
TXCRCCLK_B;	
TXCRCDATAVALID_A;	
TXCRCDATAVALID_B;	
TXCRCINIT_A;	
TXCRCINIT_B;	
TXCRCINTCLK_A;	
TXCRCINTCLK_B;	
TXCRCPD_A;	
TXCRCPD_B;	
TXCRCRESET_A;	
TXCRCRESET_B;	
TXENC64B66BUSE_A;	
TXENC64B66BUSE_B;	
TXENC8B10BUSE_A;	
TXENC8B10BUSE_B;	
TXENOOB_A;	
TXENOOB_B;	
TXGEARBOX64B66BUSE_A;	
TXGEARBOX64B66BUSE_B;	
TXINHIBIT_A;	
TXINHIBIT_B;	
TXPMARESET_A;	
TXPMARESET_B;	
TXPOLARITY_A;	
TXPOLARITY_B;	

入力	出力
TXRESET_A;	
TXRESET_B;	
TXSCRAM64B66BUSE_A;	
TXSCRAM64B66BUSE_B;	
TXSYNC_A;	
TXSYNC_B;	
TXUSRCLK_A;	
TXUSRCLK_B;	
TXUSRCLK2_A;	
TXUSRCLK2_B;	

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

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

GT11CLK

プリミティブ : A MUX That Can Select From Differential Package Input Clock, refclk From the Fabric, or rxblk to Drive the Two Vertical Reference Clock Buses for the Column of MGTs

概要

GT11CLK は、RocketIO™ クロックの専用パッケージピンを使用する場合にインスタンス化する必要があります。このブロックは、MGT の各列に 2 つずつあります。属性を使用して、このパッケージの入力で SYNCLK クロックツリーの 1 つまたは両方を駆動できます。詳細は、『Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide』を参照してください。

REFCLKSEL 属性を使用すると、MGTCLK、SYNCLK1IN、SYNCLK2IN、REFCLK、RXBCLK などのクロック オプションを指定できます。

ポートの説明

入力 : MGTCLKP、MGTCLKN

出力 : SYNCLK1OUT、SYNCLK2OUT

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

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

GT11CLK_MGT

プリミティブ : Allows Differential Package Input to Drive the Two Vertical Reference Clock Buses for the Column of MGTs

概要

GT11CLK は、RocketIO™ クロックの専用パッケージピンを使用する場合にインスタンス化する必要があります。このブロックは、MGT の各列に 2 つずつあります。属性を使用して、このパッケージの入力で SYNCLK クロックツリーの 1 つまたは両方を駆動できます。詳細は、『Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide』を参照してください。

ポートの説明

入力 : MGTCLKP、MGTCLKN

出力 : SYNCLK1OUT、SYNCLK2OUT

デザインの入力方法

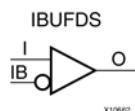
インスタンス化	可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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
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;

-- 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 (インスタンス化)

```
// 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)
);

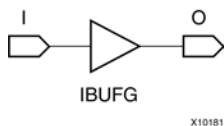
// End of IBUFDS_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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)
);

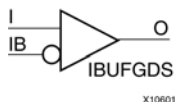
// End of IBUFG_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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
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
);

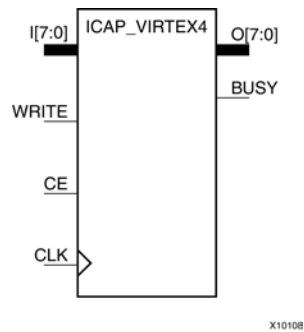
// End of IBUFGDS_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

ICAP_VIRTEX4

プリミティブ : Virtex-4 Internal Configuration Access Port



概要

このデザイン エLEMENTを使用すると、Virtex®-4 の内部コンフィギュレーション アクセス ポート (ICAP) にアクセスできます。

ポートの説明

ポート名	方向	幅	機能
BUSY	出力	1	BUSY 信号
O	出力	32	32 ビット データ バス出力
CE	入力	1	クロック イネーブル ピン
CLK	入力	1	クロック入力
WRITE	入力	1	書き込み信号
I	入力	32	32 ビット データ バス入力

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
ICAP_WIDTH	文字列	X8、X32	X8	ICAP コンポーネントのデータ幅を指定

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;

-- ICAP_VIRTEX4: Internal Configuration Access Port
--               Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

ICAP_VIRTEX4_inst : ICAP_VIRTEX4
generic map (
    ICAP_WIDTH => "X8") -- "X8" or "X32"
port map (
    BUSY => BUSY,      -- Busy output
    O => O,             -- 32-bit data output
    CE => CE,          -- Clock enable input
    CLK => CLK,         -- Clock input
    I => I,             -- 32-bit data input
    WRITE => WRITE     -- Write input
);

-- End of ICAP_VIRTEX4_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// ICAP_VIRTEX4: Internal Configuration Access Port
//               Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

ICAP_VIRTEX4 #(
    .ICAP_WIDTH("X8") // "X8" or "X32"
) ICAP_VIRTEX4_inst (
    .BUSY(BUSY),      // Busy output
    .O(O),            // 32-bit data output
    .CE(CE),          // Clock enable input
    .CLK(CLK),        // Clock input
    .I(I),            // 32-bit data input
    .WRITE(WRITE)     // Write input
);

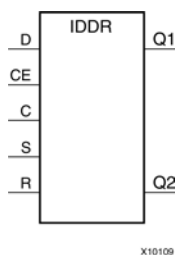
// End of ICAP_VIRTEX4_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

IDDR

プリミティブ : Input Dual Data-Rate Register



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。このプリミティブでは、データが取り込まれるクロック エッジごとにデータを転送するだけでなく、同じクロック エッジで同時にデータを転送することもできます。これにより、タイミングが複雑にならず、追加のリソースも必要ありません。

- OPPOSITE_EDGE モード** : 通常の DDR 方式でデータが受信されます。Q1 はクロック C の各立ち上がりエッジの後に変化し、Q2 は各立ち下がりエッジの後に変化します。
- SAME_EDGE モード** : データはクロック C の反対のエッジで受信されますが、立ち下がりエッジ データレジスタの前にレジスタが追加されるので、DDR データは同じクロック エッジで FPGA に送信されます。ただし、データ ペアは分離されているように見えます。Q1 と Q2 にはペア 1 および 2 が同時に送信されず、最初のペアがペア 1 とドントケアとなり、次のクロック サイクルでペア 2 と 3 が送信されます。
- SAME_EDGE_PIPELINED モード** : SAME_EDGE モードと同様にデータが受信されますが、SAME_EDGE モードでのデータ ペアの分離を回避するため、立ち上がりエッジ データレジスタの前にもレジスタが追加されます。これにより、データ ペアが Q1 と Q2 ピンに同時に送信されます。ただし、このモードを使用すると、Q1 と Q2 信号が変化するレイテンシが 1 サイクル分増加します。

IDDR は IODELAY などの SelectIO™ 機能とも使用できます。

メモ : 高速インターフェイスには、IDDR_2CLK コンポーネントを使用して データの取り込みに 2 つの独立したクロックを指定できます。このコンポーネントは、IDDR のパフォーマンス要件が不十分のときに使用します。IDDR_2CLK では、必要なクロック リソース数が増え、IDDR コンポーネントを使用するときには不要な配置制限が発生する可能性があります。

ポートの説明

ポート名	方向	幅	機能
Q1 ~ Q2	出力	1	FPGA に接続する IDDR 出力です。Q1 は最初のデータ ペア、Q2 は 2 番目のデータ ペアです。
C	入力	1	クロック入力ピンです。
CE	入力	1	Low になると、ポート O の出力クロックがディスエーブルになります。
D	入力	1	DDR データを IDDR モジュールに入力するピン。 このピンは、最上位の入力または双方向ポート、入力遅延が設定された IODELAY、または適切な入力または双方向バッファに接続します。
R	入力	1	アクティブ High のリセットで Q1 および Q2 を論理値 0 にします。SRATYPE 属性に基づき、同期または非同期に設定できます。
S	入力	1	アクティブ High のリセットで Q1 および Q2 を論理値 1 にします。SRATYPE 属性に基づき、同期または非同期に設定できます。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DDR_CLK_EDGE	文字列	OPPOSITE_EDGE、 SAME_EDGE、 SAME_EDGE_PIPELINED	OPPOSITE_EDGE	クロック エッジに相対させた IDDR 操作モードを選択します。
INIT_Q1	2 進数	0、1	0	コンフィギュレーションのスタートアップ後または GSR がアサートされたときの Q1 ピンの初期値を指定します。
INIT_Q2	2 進数	0、1	0	コンフィギュレーションのスタートアップ後または GSR がアサートされたときの Q2 ピンの初期値を指定します。
SRTYPE	文字列	SYNC、ASYNC	SYNC	セット/リセットのタイプを選択します。SYNC では、リセット (R) およびセット (S) ピンの動作が C クロック ピンの立ち上がりエッジに同期するように指定し、ASYNC では非同期のセット/リセット機能を指定します。

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;

-- IDDR: Double Data Rate Input Register with Set, Reset
--       and Clock Enable.
--       Virtex-4/5
-- Xilinx HDL Libraries Guide, version 11.2

IDDR_inst : IDDR
generic map (
    DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE", "SAME_EDGE"
                                         -- or "SAME_EDGE_PIPELINED"
    INIT_Q1 => '0', -- Initial value of Q1: '0' or '1'
    INIT_Q2 => '0', -- Initial value of Q2: '0' or '1'
    SRTYPE => "SYNC") -- Set/Reset type: "SYNC" or "ASYNC"
port map (
    Q1 => Q1, -- 1-bit output for positive edge of clock
    Q2 => Q2, -- 1-bit output for negative edge of clock
    C => C,   -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    D => D,   -- 1-bit DDR data input
    R => R,   -- 1-bit reset
    S => S    -- 1-bit set
);

-- End of IDDR_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// IDDR: Input Double Data Rate Input Register with Set, Reset
//      and Clock Enable.
//      Virtex-4/5/6
// Xilinx HDL Libraries Guide, version 11.2

IDDR #(
    .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE", "SAME_EDGE"
                                     // or "SAME_EDGE_PIPELINED"
    .INIT_Q1(1'b0), // Initial value of Q1: 1'b0 or 1'b1
    .INIT_Q2(1'b0), // Initial value of Q2: 1'b0 or 1'b1
    .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYNC"
) IDDR_inst (
    .Q1(Q1), // 1-bit output for positive edge of clock
    .Q2(Q2), // 1-bit output for negative edge of clock
    .C(C),   // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .D(D),   // 1-bit DDR data input
    .R(R),   // 1-bit reset
    .S(S)    // 1-bit set
);

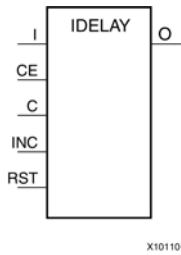
// End of IDDR_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

IDELAY

プリミティブ : Input Delay Element



概要

Virtex® および上記のデバイスには、各ユーザー I/O の入力パスに IDELAY モジュールがあります。このモジュールは、入力データを正しく取り込むためのスキュー調整アルゴリズムをインプリメントします。IDELAY は、データ信号、クロック信号、またはその両方に適用でき、その機能は 64 タップ遅延ラインで制御されます。IDELAYCTRL コンポーネントと共に使用すると、プロセス、電圧、温度の変化にかかわらず、正確に増分された遅延を追加できます。3 つの動作モードがあります。

- ・ **ゼロ ホールド タイム遅延モード** : このモードは、および デバイスのゼロ ホールド タイム遅延機能を使用して、あるデザインでも使用できます。このモードを使用する場合、IDELAYCTRL プリミティブをインスタンスエートする必要はありません。
- ・ **固定タップ遅延モード** : 遅延値は IOBDELAY_VALUE 属性で指定した値に固定されます。この値をランタイムで変更することはできません。このモードを使用する場合、IDELAYCTRL プリミティブをインスタンスエートする必要があります。
- ・ **可変タップ遅延モード** : 制御信号 CE と INC を変更することにより、遅延値をランタイムで変更できます。このモードを使用する場合、IDELAYCTRL プリミティブをインスタンスエートする必要があります。

ポートの説明

ポート名	方向	幅	機能
I	入力	1	IOB からのシリアル入力データ
C	入力	1	クロック入力
INC	入力	1	タップ遅延のインクリメント/デクリメント数
CE	入力	1	インクリメント/デクリメントをイネーブル
RST	入力	1	遅延チェーンをあらかじめ設定された値にリセット。値が設定されていない場合は、0 にリセット。
O	出力	1	組み合わせ出力

データ入力および出力 - I および O

IDELAY プリミティブは、3 種類の IOB ロケーションに配置されています。入力と出力の接続は、IOB ロケーションのタイプにより異なります。

- ・ **汎用 IOB** : 汎用 IOB にある IDELAY の入力は、入力バッファ IBUF から直接供給されます。IDELAY (O) の出力は、直接ユーザー ロジックに接続します。入力および出力データパスは組み合わせパスであり、クロック信号 (C) の影響は受けませんが、IOB の出力信号 (O) をレジスタに接続できます。

- ・ **リージョナル クロックを使用可能な IOB** : リージョナル クロックを使用可能な IOB は、HCLK IOB の上下にある I/O ペアに配置されています。この IOB にある IDELAY の入力は、入力バッファ IBUF から直接供給され、出力は次のコンポーネントに接続できます。

- ユーザー ロジック
- BUFIO (リージョナル クロック信号の場合)

リージョナル クロック バッファ BUFIO は、入力されるリージョナル クロック信号をリージョナル I/O クロック ツリー IOCLK に接続します。また、BUFIO は、リージョナル クロック バッファ BUFR に接続してリージョナル クロック ツリー rclk にも接続できます。入力および出力データパスは組み合わせパスであり、クロック信号 (C) の影響は受けませんが、IOB の出力信号 (O) をレジスタに接続できます。

- ・ **グローバル クロックを使用可能な IOB** : グローバル クロックを使用可能な IOB は、中央の I/O 列に配置されています。この IOB にある IDELAY の入力は、入力グローバル クロック バッファ IBUFG から直接供給され、出力は次のコンポーネントに接続できます。

- ユーザー ロジック
- BUFG (グローバル クロック信号の場合)

グローバル クロック バッファ BUFG は、入力されるリージョナル クロック信号をグローバル I/O クロック ツリー gclk に接続します。入力および出力データパスは組み合わせパスであり、クロック信号 (C) の影響は受けませんが、IOB の出力信号 (O) をレジスタに接続できます。

クロック入力 - C

IDELAY の制御入力 (RST、CE、INC) は、すべてクロック入力 (C) に同期しています。IDELAY のデータ入力および出力 (I および O) は、クロック信号の影響を受けません。このクロック入力は、ISERDES の CLKDIV 入力と同じです。このため、CLKDIV を駆動するために使用されるクロックソースは、すべて IDELAY クロック入力 (C) を駆動します。クロック入力 (C) で駆動可能なクロックソースは、次のとおりです。

- ・ 8 個の gclk (グローバル クロック ツリー)
- ・ 2 個の rclk (リージョナル クロック ツリー)

モジュール リセット - RST

IDELAY のリセット信号 RST は、タップ遅延ラインを IOBDELAY_VALUE 属性で設定された値にリセットします。IOBDELAY_VALUE 属性を指定しない場合は、タップ遅延ラインは 0 にリセットされます。

インクリメント/デクリメント信号 - CE、INC

インクリメント/デクリメント イネーブル信号 (CE) は、インクリメント/デクリメント信号 (INC) をイネーブルにします。INC は、タップ遅延ラインをインクリメントするか、デクリメントするかを指定します。CE = 0 の場合、INC の値にかかわらず、遅延は変化しません。CE = 1 の場合、INC の値に応じてタップ遅延値がインクリメントまたはデクリメントします。タップ遅延のインクリメント/デクリメントは、入力クロック (C) に同期して行われます。CE = 1 である限り、各クロック サイクルでタップ遅延が 1 ずつインクリメント/デクリメントされます。次の表に、インクリメント/デクリメントの動作を示します。

動作	RST	CE	INC
設定されたタップ数にリセット	1	X	X
タップ数を増分	0	1	1
タップ数を減分	0	1	0
変化なし	0	0	X

メモ :

1. RST は、遅延チェーンを IOBDELAY_VALUE 属性で指定された値にリセットします。値が設定されていない場合は、0 にリセットされます。
2. RST、CE、INC は、クロック入力 (C) に同期しています。

CE が High になると、次の立ち上がりクロックでインクリメント/デクリメントが開始します。CE が Low になると、次の立ち上がりクロックでインクリメント/デクリメントが停止します。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
IOBDELAY_TYPE	文字列	DEFAULT、FIXED、VARIABLE	DEFAULT	タップ遅延のタイプを指定
IOBDELAY_VALUE	整数	0 ~ 63	0	タップ遅延の初期値を指定

IOBDELAY_TYPE 属性

IOBDELAY_TYPE 属性は、使用する遅延のタイプを指定します。指定可能な値は DEFAULT、FIXED、または VARIABLE で、デフォルト値は DEFAULT です。DEFAULT に設定すると、ゼロ ホールド タイム遅延エレメントが選択されます。この遅延エレメントは、pad-to-pad ホールド タイムを削減し、遅延をデバイスの内部クロック分配遅延と一致させます。この遅延エレメントを使用すると、pad-to-pad ホールド タイムは 0 になります。

FIXED に設定した場合、タップ遅延値は IOBDELAY_VALUE 属性で指定したタップ数に固定されます。この値は、動作中に変更することはできません。

VARIABLE に設定した場合、タップ遅延を動作中に変更できます。CE = 1 および INC = 1 に設定するとタップ遅延がインクリメントされ、CE = 1 および INC = 0 に設定するとデクリメントされます。インクリメント/デクリメントは、入力クロック信号 C に同期して行われます。

IOBDELAY_VALUE 属性

タップ遅延の初期値を指定します。指定可能な値は 0 ~ 63 で、デフォルト値は 0 です。0 に設定すると、合計遅延は出力マルチプレクサの遅延 (約 400ps) になります。

タップ遅延がリセットされた場合 (RST = 1)、IOBDELAY_TYPE が FIXED の場合、タップ遅延は IOBDELAY_VALUE で設定した値になります。

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;

-- IDELAY: Input Delay Element
--      Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

IDELAY_inst : IDELAY
generic map (
  IOBDELAY_TYPE => "FIXED", -- "FIXED" or "VARIABLE"
  IOBDELAY_VALUE => 0) -- Any value from 0 to 63
port map (
  O => O,      -- 1-bit output
  C => C,      -- 1-bit clock input
  CE => CE,    -- 1-bit clock enable input
  I => I,      -- 1-bit data input
  INC => INC,  -- 1-bit increment input
  RST => RST   -- 1-bit reset input
);

-- End of IDELAY_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IDELAY: Input Delay Element
//      Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

IDELAY #(
  .IOBDELAY_TYPE("FIXED"), // "FIXED" or "VARIABLE"
  .IOBDELAY_VALUE(0)       // Any value from 0 to 63
) IDELAY_inst (
  .O(O),      // 1-bit output
  .C(C),      // 1-bit clock input
  .CE(CE),    // 1-bit clock enable input
  .I(I),      // 1-bit data input
  .INC(INC),  // 1-bit increment input
  .RST(RST)   // 1-bit reset input
);

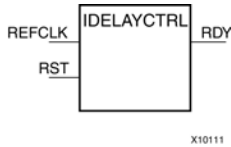
// End of IDELAY_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

IDELAYCTRL

プリミティブ : IDELAY Tap Delay Value Control



概要

このデザイン エLEMENTは、タップ遅延ラインを使用する場合にインスタンス化する必要があります。これは、IDELAY または ISERDES プリミティブがインスタンス化されており、IOBDELAY_TYPE 属性が FIXED または VARIABLE に設定されている場合です。このモジュールは、一定の周波数リファレンス クロック REFCLK を使用する場合に、プロセス、電圧、および温度の変化にかかわらず、タップ遅延ラインに電圧バイアスを供給します。これにより、正確な遅延調整が可能になります。

ポートの説明

ポート名	タイプ	幅	機能
RDY	出力	1	リファレンス クロック入力 が有効であることを示します。REFCLK.REFCLK が停止する (REFCLK が High または Low に 1 クロック周期以上保持される) と、RDY 信号がデassertされます。
REFCLK	入力	1	プロセス、電圧、温度の変化にかかわらず、タップ遅延ラインに電圧バイアスを供給します。タップ遅延をデータシートに記載された値にするには、REFCLK の周波数を 200MHz にする必要があります。
RST	入力	1	IDELAYCTRL 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。IDELAYCTRL をリセットするには、このポートを 50ns 以上 High にアassertする必要があります。

RST (モジュール リセット) : IDELAYCTRL 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。IDELAYCTRL をリセットするには、このポートを 50ns 以上 High にアassertする必要があります。

REFCLK (リファレンス クロック) : プロセス、電圧、温度の変化にかかわらず、タップ遅延ラインに電圧バイアスを供給します。タップ遅延をデータシートに記載された値にするには、REFCLK の周波数を 200MHz にする必要があります。

RDY (Ready 出力) : リファレンス クロック入力 REFCLK が有効になったことを示します。REFCLK が停止する (REFCLK が High または Low に 1 クロック周期以上保持される) と、RDY 信号がデassert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;

-- IDELAYCTRL: IDELAY Tap Delay Value Control
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

IDELAYCTRL_inst : IDELAYCTRL
generic map (
)
port map (
    RDY => RDY,          -- 1-bit Indicates the validity of the reference clock input, REFCLK. When REFCLK
                        -- disappears (i.e., REFCLK is held High or Low for one clock period or more), the RDY
                        -- signal is deasserted.

    REFCLK => REFCLK,    -- 1-bit Provides a voltage bias, independent of process, voltage, and temperature
                        -- variations, to the tap-delay lines in the IOBs. The frequency of REFCLK must be 200
                        -- MHz to guarantee the tap-delay value specified in the applicable data sheet.

    RST => RST           -- 1-bit Resets the IDELAYCTRL circuitry. The RST signal is an active-high asynchronous
                        -- reset. To reset the IDELAYCTRL, assert it High for at least 50 ns.
);

-- End of IDELAYCTRL_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// IDELAYCTRL: Input Delay Control Element (Must be used in conjunction with the IDELAY
//           when used in FIXED or VARIABLE tap-delay mode)
//           Virtex-4/5
// Xilinx HDL Libraries Guide, version 11.2

IDELAYCTRL IDELAYCTRL_inst (
    .RDY(RDY),          // 1-bit ready output
    .REFCLK(REFCLK),    // 1-bit reference clock input
    .RST(RST)           // 1-bit reset input
);

// End of IDELAYCTRL_inst instantiation

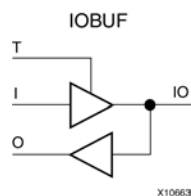
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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	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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTTL、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
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
);
```

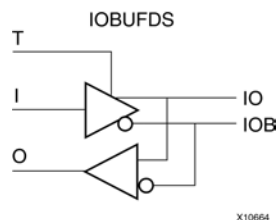
```
// End of IOBUF_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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
);

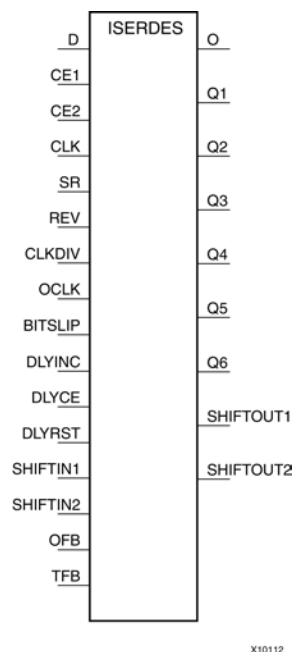
// End of IOBUFDS_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

ISERDES

プリミティブ : Dedicated I/O Buffer Input Deserializer



概要

このデザイン エLEMENTを使用することにより、同期ソリューションを簡単にインプリメントできます。ISERDES は専用のソース同期 I/O アーキテクチャです。このモジュールを使用すると、ソース同期アプリケーションで FPGA のロジックリソースを節約でき、タイミングが複雑になるのを防ぎます。

このモジュールには、シリアル/パラレル コンバータ、シリアル遅延チェーン、ワード アライメント ユニット (BITSLIP)、クロック イネーブル (CE) モジュールが含まれています。また、さまざまなアプリケーションに対応した複数のクロック入力があり、SelectIO™ 機能と共に使用できます。SERDES のサブモジュールの詳細は次のとおりです。

遅延チェーン モジュール

遅延チェーンは、入力データと転送されたクロックのタイミング関係を調整するための専用アーキテクチャです。このタイミング関係の調整は、ISERDES モジュール内に遅延を配置し、入力のスキューを調整することによって達成します。入力遅延チェーンは、あらかじめ設定しておくか (固定)、動作中に変更できます (可変)。また、このモジュールは IDELAYCTRL プリミティブと共に使用します。

遅延チェーン モジュールを使用するにはいくつかの属性が必要です。その属性は次のとおりです。

- ・ IOBDELAY_VALUE
- ・ IOBDELAY
- ・ IOBDELAY_TYPE

IOBDELAY_VALUE は、使用する遅延タップの数を指定します。指定可能な値は、0 ～ 63 で、デフォルト値は 0 です。

IOBDELAY 属性を IBUF、IFD、または BOTH に設定すると、遅延チェーンをそれぞれ組み合わせ出力 (O)、レジスタ付き出力 (Q1 ～ Q6)、または両方に使用できます。IOBDELAY 属性を NONE に設定すると、遅延チェーン モジュールがバイパスされます。

IOBDELAY_TYPE は、DEFAULT、FIXED、または VARIABLE に設定できます。DEFAULT に設定すると、ホールドタイムがゼロになります。FIXED に設定すると、遅延タップ値は IOBDELAY_VALUE で設定した値になります。このモードでは、デバイスがプログラムされた後に値を変更することはできません。VARIABLE に設定すると、遅延タップの初期値が IOBDELAY_VALUE の値に設定され、デバイスをプログラムした後に変更可能です。

遅延チェーン モードは、DLYRST、DLYCE、および DLYINC ピンを使用して制御します。これらのピンによる動作は、CLKDIV クロック信号に同期しています。DLYRST を High にアサートすると、遅延チェーンのタップ値が IOBDELAY_VALUE で設定した値にリセットされます。遅延タップ値をインクリメント/デクリメントするには、DLYCE および DLYINC を使用します。遅延タップ値を変更するには、DLYCE を High にアサートする必要があります。DLYINC を 1 に設定するとインクリメントされ、0 に設定するとデクリメントされます。

次の表に、遅延チェーン制御ピンの値による動作を示します。

動作	DLYRST	DLYCE	DLYINC
IOBDELAY_VALUE の値にリセット	1	X	X
タップ値をインクリメント	0	1	1
タップ値をデクリメント	0	1	0
変化なし	0	0	X

メモ： 遅延チェーンのすべての動作は、CLKDIV に同期しています。

シリアル/パラレル コンバータ

ISERDES モジュールのシリアル/パラレル コンバータは、シリアル データを取り込み、2 ～ 6 のデータ幅に変換します。データ幅拡張モードを使用すると、データ幅を 7、8、10 に拡張できます。データ幅を拡張するには、1 つの ISERDES をマスタ モードにし、もう 1 つの ISERDES をスレーブ モードに設定し、スレーブの SHIFTIN ポートをマスタの SHIFTOUT ポートに接続します。スレーブでは、出力として Q3 ～ Q6 ポートのみを使用します。シリアル/パラレル コンバータは、SDR または DDR モードの両方で使用できます。

このモジュールは、主に CLK および CLKDIV クロックで制御されます。次の表に、SDR および DDR の異なるモードにおける CLK と CLKDIV の関係を示します。

次の表に、シリアル/パラレル コンバーターの CLK と CLKDIV の関係を示します。

SDR のデータ幅	DDR のデータ幅	CLK	CLKDIV
2	4	2X	X
3	6	3X	X
4	8	4X	X
5	10	5X	X
6	–	6X	X
7	–	7X	X
8	–	8X	X

CE モジュール

CE モジュールは、基本的には 2:1 パラレル/シリアル コンバータです。このモジュールは CLKDIV クロック入力で制御され、シリアル/パラレル コンバータ モジュールのクロック イネーブル ポートを制御するのに使用されます。

BITSLIP モジュール

BITSLIP モジュールは、出力シーケンスを並べ替えるバレル シフタのような機能を実行します。BITSLIP を起動するたびに、出力パターンが変化します。BITSLIP による並べ替えの最大回数は、パターンに含まれるビット数から 1 を引いた数になります (DATA_WIDTH - 1)。BITSLIP は、SDR および DDR の両方でサポートされます。SDR と DDR の出力の並べ替えは、異なるので注意してください。

BITSLIP を使用するには BITSLIP_ENABLE 属性を ON に設定する必要があります。この属性を OFF に設定すると、BITSLIP モジュールはバイパスされます。

BITSLIP の動作は、CLKDIV クロック入力に同期しています。BITSLIP モジュールを起動するには、BITSLIP ポートを CLKDIV の 1 サイクル分だけ High にアサートします。BITSLIP ポートが CLKDIV の 1 サイクル分 High にアサートされると、BITSLIP の動作が完了します。DDR モードでは、CLKDIV の 2 サイクル分経過するまで、BITSLIP の動作が安定しない場合があります。BITSLIP の出力は、すべてレジスタ付き出力ポート (Q1 ~ Q6) に出力されます。

その他の機能

データ幅の拡張 : ISERDES モジュールでは、7 ビット幅以上のデータ幅を回復できます。この機能を使用するには、2 つの ISERDES モジュールをインスタンス化する必要があります。この 2 つの ISERDES は、隣接したマスタ/スレーブペアである必要があります。ISERDES_MODE 属性を MASTER または SLAVE に設定し、ISERDES のペアを区別する必要があります。また、マスタの SHIFTOUT ポートをスレーブの SHIFTIN ポートに接続します。SDR および DDR モードでは、データ幅 7、8、10 がサポートされています。次の表に、SDR および DDR モードで使用可能なデータ幅を示します。

モード	幅
SDR	2、3、4、5、6、7、8
DDR	4、6、8、10

ポートの説明

ポート名	方向	幅	機能
O	出力	1	組み合わせ出力。ISERDES モジュールのレジスタを介さない出力で、遅延チェーンの出力です。この出力ポートは、ISERDES モジュール内のすべてのサブモジュールをバイパスするようにコンフィギュレーションすることも可能です。BUFIO を駆動できます。
Q1:6	出力	1 (それぞれ)	レジスタ付き出力。ISERDES モジュールのレジスタ付き出力です。これらの出力を使用すると、ISERDES のサブモジュール パスの次の組み合わせを入力として使用できます。 <ul style="list-style-type: none"> 遅延チェーン → シリアル/パラレル コンバータ → BITSLIP モジュール 遅延チェーン → シリアル/パラレル コンバータ これらのポートは、2 ~ 6 ビットに設定できます。データ幅拡張モードでは、10 ビットまで拡張できます。
SHIFTOUT 1:2	出力	1 (それぞれ)	データ入力を拡張するためのキャリー出力です。スレーブの SHIFTIN1、SHIFTIN12 に接続します。
BITSLIP	入力	1	BITSLIP 動作を起動します。このピンを High にすると、BITSLIP モジュールがイネーブルになります。
CE 1:2	入力	1 (それぞれ)	CE モジュールに供給するクロック イネーブル入力です。

ポート名	方向	幅	機能
CLK	入力	1	<p>高速転送クロック入力。シリアル/パラレル コンバータおよび BITSLLIP モジュールを駆動するのに使用するクロック入力です。CLK ポートは、次のいずれかのクロックリソースで駆動します。</p> <ul style="list-style-type: none"> ・ クロック領域内の 8 個のグローバル クロック ライン ・ 2 個のリージョナル クロック ライン ・ 6 個のクロック I/O (隣接したクロック領域内) ・ FPGA (バイパスを介す)
CLKDIV	入力	1	<p>高速転送分周クロック入力。シリアル/パラレル コンバータ、遅延チェーン、BITSLLIP モジュール、および CE モジュールを駆動するのに使用するクロック入力です。CLK ポートに接続されたクロックよりも低周波数のクロックを入力する必要があります。CLKDIV ポートは、次のいずれかのクロックリソースで駆動します。</p> <ul style="list-style-type: none"> ・ クロック領域内の 8 個のグローバル クロック ライン ・ 2 個のリージョナル クロック ライン
D	入力	1	IOB からのシリアル入力データ。ISERDES モジュールにデータが入力されるポートです。SelectIO 機能を使用して、I/O 規格を指定できます。
DLYCE	入力	1	遅延チェーンのインクリメント/デクリメントをイネーブルにします。
DLYINC	入力	1	遅延チェーン インクリメント/デクリメントピン。DLYCE ピンを High にアサートすると、DLYINC ピンの値に応じて遅延チェーンのタップ値がインクリメント/デクリメントされます。DLYINC ピンが High の場合はタップ値がインクリメントされ、Low の場合はデクリメントされます。
DLYRST	入力	1	遅延チェーン リセット ピン。遅延チェーンのタップ数を IOBDELAY_VALUE で設定された値にリセットします。値が設定されていない場合は、0 にリセットします。
OCLK	入力	1	<p>メモリ インターフェイス アプリケーション用高速クロック。シリアル/パラレル コンバータを駆動するのに使用するクロック入力です。OCLK ポートは、次のいずれかのクロックリソースで駆動します。</p> <ul style="list-style-type: none"> ・ クロック領域内の 8 個のグローバル クロック ライン ・ 2 個のリージョナル クロック ライン ・ 6 個のクロック I/O (隣接したクロック領域内) ・ FPGA (バイパスを介す) <p>このクロックは、ストロブ信号が必要なメモリ インターフェイスに適しています。</p>
REV	入力	1	SR を反転。内部テスト用です。SR を使用した場合、REV ピンによりストレージ エLEMENTが反対の状態になります。リセットがセットよりも優先されます。REV ピンは、ISERDES ではサポートされていません。
SR	入力	1	<p>セット/リセット入力。ストレージ エLEMENTの状態をユーザー制約ファイル (UCF) で設定した SRVAL 属性で設定された状態にします。SRVAL = 1 の場合は 1、SRVAL = 0 の場合は 0 になります。SR を使用した場合、REV ピンによりストレージ エLEMENTが反対の状態になります。リセットがセットよりも優先されます。ISERDES コンポーネントのすべてのレジスタに対し、SR ピンはアクティブ High に非同期リセットします。</p>
SHIFTIN 1:2	入力	1 (それぞれ)	データ入力を拡張するためのキャリー入力です。マスタの SHIFTOUT1、SHIFTOUT2 に接続します。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
BITSLIP_ENABLE	ブール代数	FALSE、TRUE	FALSE	BITSLIP コントローラをイネーブル
DATA_RATE	文字列	SDR、DDR	DDR	データ レートを指定
DATA_WIDTH	文字列	DATA_RATE = DDR の場合は 4、6、8、10、 DATA_RATE = SDR の場合は 2、3、4、5、6、7、8	4	シリアル/パラレル コンバータの幅を指定。SDR と DDR のどちらを選択するかで値が異なります。
INTERFACE_TYPE	文字列	MEMORY、NETWORKING	MEMORY	ISERDES の使用モデルを指定
IOBDelay	文字列	NONE、IBUF、IFD、BOTH	NONE	遅延チェーンを使用する ISERDES の出力を指定
IOBDelay_TYPE	文字列	DEFAULT、FIXED、VARIABLE	DEFAULT	遅延チェーンを固定モードまたは可変モードに指定
IOBDelay_VALUE	整数	0 ～ 63	0	タップ遅延の初期値を指定
NUM_CE	整数	1、2	2	クロック イネーブルの数を指定

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;

-- ISERDES: Input SERDES
-- Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

ISERDES_inst : ISERDES
generic map (
    BITSLIP_ENABLE => FALSE, -- TRUE/FALSE to enable bitflip controller
                                -- Must be "FALSE" in interface type is "MEMORY"
    DATA_RATE => "DDR", -- Specify data rate of "DDR" or "SDR"
    DATA_WIDTH => 4, -- Specify data width - For DDR 4,6,8, or 10
                                -- For SDR 2,3,4,5,6,7, or 8
    INTERFACE_TYPE => "MEMORY", -- Use model - "MEMORY" or "NETWORKING"
    IOBDelay => "NONE", -- Specify outputs where delay chain will be applied
                                -- "NONE", "IBUF", "IFD", or "BOTH"
    IOBDelay_TYPE => "DEFAULT", -- Set tap delay "DEFAULT", "FIXED", or "VARIABLE"
    IOBDelay_VALUE => 0, -- Set initial tap delay to an integer from 0 to 63
    NUM_CE => 2, -- Define number of clock enables to an integer of 1 or 2
    SERDES_MODE => "MASTER") --Set SERDES mode to "MASTER" or "SLAVE"
port map (
    0 => 0, -- 1-bit output
    Q1 => Q1, -- 1-bit output
    Q2 => Q2, -- 1-bit output
    Q3 => Q3, -- 1-bit output
    Q4 => Q4, -- 1-bit output
    Q5 => Q5, -- 1-bit output
```

```

Q6 => Q6, -- 1-bit output
SHIFTOUT1 => SHIFTOUT1, -- 1-bit output
SHIFTOUT2 => SHIFTOUT2, -- 1-bit output
BITSLIP => BITSLIP, -- 1-bit input
CE1 => CE1, -- 1-bit input
CE2 => CE2, -- 1-bit input
CLK => CLK, -- 1-bit input
CLKDIV => CLKDIV, -- 1-bit input
D => D, -- 1-bit input
DLYCE => DLYCE, -- 1-bit input
DLYINC => DLYINC, -- 1-bit input
DLYRST => DLYRST, -- 1-bit input
OCLK => OCLK, -- 1-bit input
REV => '0', -- Must be tied to logic zero
SHIFTIN1 => SHIFTIN1, -- 1-bit input
SHIFTIN2 => SHIFTIN2, -- 1-bit input
SR => SR -- 1-bit input
);

-- End of ISERDES_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// ISERDES: Source Synchronous Input Deserializer
// Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

ISERDES #(
    .BITSLIP_ENABLE("FALSE"), // "TRUE"/"FALSE" to enable bitflip controller
                                // Must be "FALSE" if INTERFACE_TYPE set to "MEMORY"
    .DATA_RATE("DDR"), // Specify data rate of "DDR" or "SDR"
    .DATA_WIDTH(4), // Specify data width - For DDR 4,6,8, or 10
                        // For SDR 2,3,4,5,6,7, or 8
    .INTERFACE_TYPE("MEMORY"), // Use model - "MEMORY" or "NETWORKING"
    .IOBDelay("NONE"), // Specify outputs where delay chain will be applied
                        // "NONE", "IBUF", "IFD", or "BOTH"
    .IOBDelay_Type("DEFAULT"), // Set tap delay "DEFAULT", "FIXED", or "VARIABLE"
    .IOBDelay_Value(0), // Set initial tap delay to an integer from 0 to 63
    .NUM_CE(2), // Define number of clock enables to an integer of 1 or 2
    .SERDES_MODE("MASTER") // Set SERDES mode to "MASTER" or "SLAVE"
) ISERDES_inst (
    .O(0), // 1-bit combinatorial output
    .Q1(Q1), // 1-bit registered output
    .Q2(Q2), // 1-bit registered output
    .Q3(Q3), // 1-bit registered output
    .Q4(Q4), // 1-bit registered output
    .Q5(Q5), // 1-bit registered output
    .Q6(Q6), // 1-bit registered output
    .SHIFTOUT1(SHIFTOUT1), // 1-bit carry output
    .SHIFTOUT2(SHIFTOUT2), // 1-bit carry output
    .BITSLIP(BITSLIP), // 1-bit Bitflip input
    .CE1(CE1), // 1-bit clock enable input
    .CE2(CE2), // 1-bit clock enable input
    .CLK(CLK), // 1-bit clock input
    .CLKDIV(CLKDIV), // 1-bit divided clock input
    .D(D), // 1-bit serial data input
    .DLYCE(DLYCE), // 1-bit delay chain enable input
    .DLYINC(DLYINC), // 1-bit delay increment/decrement input
    .DLYRST(DLYRST), // 1-bit delay chain reset input
    .OCLK(OCLK), // 1-bit high-speed clock input
    .REV(1'b0), // Must be tied to logic zero
    .SHIFTIN1(SHIFTIN1), // 1-bit carry input
    .SHIFTIN2(SHIFTIN2), // 1-bit carry input
    .SR(SR) // 1-bit set/reset input
);

// End of ISERDES_inst instantiation

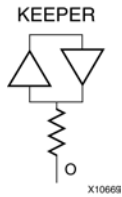
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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)
);

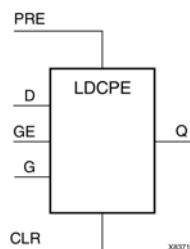
// End of KEEPER_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

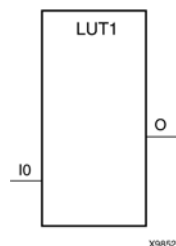
// End of LDCPE_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

LUT1

プリミティブ : 1-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 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
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: 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 (インスタンスレーション)

```
// 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
);

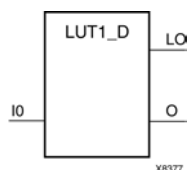
// End of LUT1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
    IO => IO  -- 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
    .IO(IO) // LUT input
);

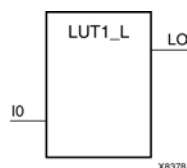
// End of LUT1_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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_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 (インスタンスレーション)

```
// 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
);

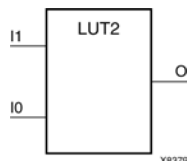
// End of LUT1_L_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

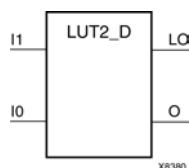
// End of LUT2_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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;

-- 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 (インスタンスレーション)

```
// 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
);

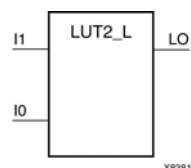
// End of LUT2_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

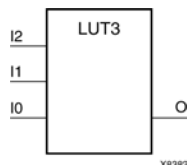
// End of LUT2_L_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

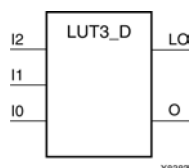
// End of LUT3_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

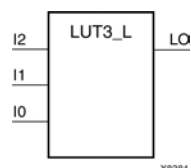
// End of LUT3_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

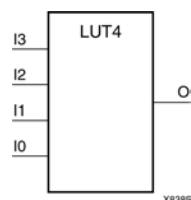
// End of LUT3_L_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

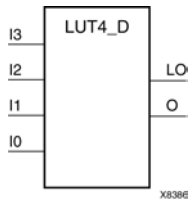
// End of LUT4_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

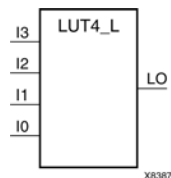
// End of LUT4_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

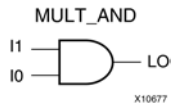
// End of LUT4_L_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

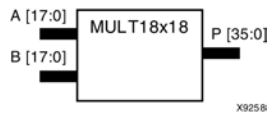
// End of MULT_AND_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

MULT18X18

プリミティブ : 18 x 18 Signed Multiplier



概要

MULT18X18 は、組み合わせ符号付き 18 X 18 ビット乗算器です。18 ビット入力 A の値に 18 ビット入力 B の値を掛け合わせた積が、出力 P に 36 ビットで出力されます。

論理表

入力		出力
A	B	P
A	B	A X B
A、B、および P は 2 の補数です。		

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

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;

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

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

-- End of MULT18X18_inst instantiation
```


Verilog 記述 (インスタンス化)

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

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

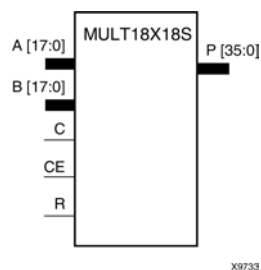
// End of MULT18X18_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

MULT18X18S

プリミティブ : 18 x 18 Signed Multiplier -- Registered Version



概要

MULT18X18S は、符号付き 18 X 18 乗算器 (MULT18X18) にレジスタを追加したもので、出力 (P)、データ入力 (A、B、C)、クロック イネーブル入力 (CE)、および同期リセット入力 (R) があります。レジスタは、GSR パルス後 0 に初期化されます。

18 ビット入力 A の値に 18 ビット入力 B の値を掛け合わせた積が、出力 P に 36 ビットで出力されます。

論理表

入力					出力
C	CE	Am	Bn	R	P
↑	X	X	X	1	0
↑	1	Am	Bn	0	A X B
X	0	X	X	0	変化なし

A、B、および P は 2 の補数です。

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

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;

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

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

-- End of MULT18X18S_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

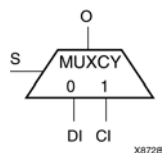
// End of MULT18X18S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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: 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 (インスタンスレーション)

```
// 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
);

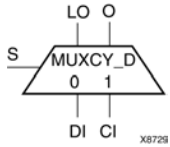
// End of MUXCY_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

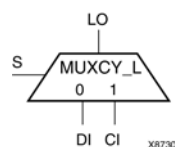
// End of MUXCY_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

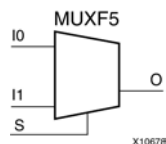
// End of MUXCY_L_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

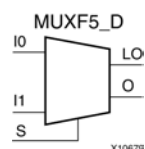
// End of MUXF5_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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_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,  -- Ouput of MUX to local routing
    O => O,    -- Output of MUX to general 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_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// 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), // Ouput of MUX to local routing
    .O(O),   // Output of MUX to general 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
);

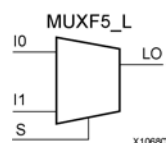
// End of MUXF5_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

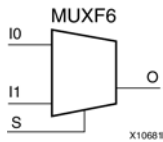
// End of MUXF5_L_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

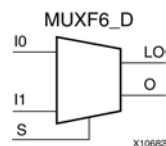
// End of MUXF6_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

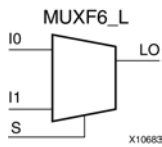
// End of MUXF6_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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 (インスタンスレーション)

```
// 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
);

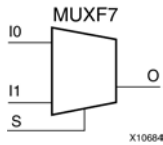
// End of MUXF6_L_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

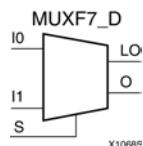
// End of MUXF7_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

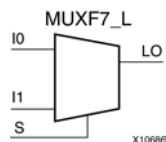
// End of MUXF7_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

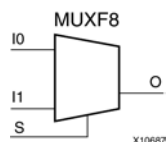
// End of MUXF7_L_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

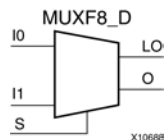
// End of MUXF8_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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。

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
);

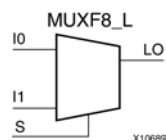
// End of MUXF8_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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。

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
);

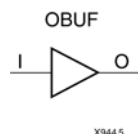
// End of MUXF8_L_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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、トライステートでない 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、LVTTL 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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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
);

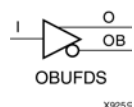
// End of OBUF_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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;

-- 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
);

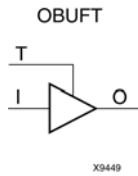
// End of OBUFDS_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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;

-- 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 (インスタンスレーション)

```
// 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
);

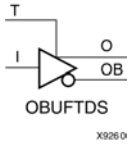
// End of OBUFT_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
CAPACITANCE	文字列	LOW、NORMAL、DONT_CARE	DONT_CARE	I/O を低いまたは通常の固有キャパシタンスと共に使用するかどうかを指定。
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;

-- 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
);

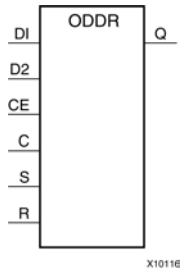
// End of OBUFTDS_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

ODDR

プリミティブ : Dedicated Dual Data Rate (DDR) Output Register



概要

このデザイン エレメントは、FPGA デバイスからデュアル データ レート (DDR) 信号を送信するための専用出力レジスタです。ODDR プリミティブでは、FPGA からのデータを送信するのに反対のクロック エッジだけではなく、同じクロック エッジを使用することも可能です。これにより、タイミングが複雑にならず、追加の CLB リソースも必要ありません。また ODDR は SelectIO™ 機能と共に使用されます。

ODDR のモード

このエレメントは 2 つのモードで動作します。これらのモードは、DDR_CLK_EDGE 属性で設定します。

- ・ **OPPOSITE_EDGE モード** : 通常の DDR 方式でデータを送信します。D1 はクロック C の立ち上がりエッジごとにサンプリングされ、D2 は立ち下がりエッジごとにサンプリングされます。Q は各クロック エッジで変化します。
- ・ **SAME_EDGE モード** : データはクロック C の反対のエッジで ODDR 出力から送信されますが、ODDR への 2 入力クロック信号 C の立ち上がりエッジで動作し、追加されたレジスタがクロック信号 C の立ち下がりエッジで動作します。この機能を使用すると、DDR データは同じクロック エッジで ODDR に取り込まれます。

ポートの説明

ポート名	タイプ	幅	機能
Q	出力	1	データ出力 (DDR)。IOB パッドに接続されます。
C	入力	1	クロック入力。クロック入力ピンです。
CE	入力	1	クロック イネーブル入力。High になると、ポート C のクロック入力がイネーブルになります。
D1 : D2	入力	1 (それぞれ)	データ入力。DDR データを ODDR モジュールに入力するピンです。
R	入力	1	リセット。SRTYPE の設定によって異なります。
S	入力	1	セット。アクティブ High の非同期セット ピンです。SRTYPE 属性の設定により、同期にもなります。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DDR_CLK_EDGE	文字列	OPPOSITE_EDGE、 SAME_EDGE	OPPOSITE_EDGE	DDR のデータ送信モードを選択します。
INIT	整数	0, 1	1	Q の初期値
SRTYPE	文字列	SYNC、ASYNC	SYNC	セット/リセットのタイプを選択

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;

-- ODDR: Output Double Data Rate Output Register with Set, Reset
--       and Clock Enable.
--       Virtex-4/5
-- Xilinx HDL Libraries Guide, version 11.2

ODDR_inst : ODDR
generic map(
  DDR_CLK_EDGE => "OPPOSITE_EDGE", -- "OPPOSITE_EDGE" or "SAME_EDGE"
  INIT => '0', -- Initial value for Q port ('1' or '0')
  SRTYPE => "SYNC") -- Reset Type ("ASYNC" or "SYNC")
port map (
  Q => Q, -- 1-bit DDR output
  C => C, -- 1-bit clock input
  CE => CE, -- 1-bit clock enable input
  D1 => D1, -- 1-bit data input (positive edge)
  D2 => D2, -- 1-bit data input (negative edge)
  R => R, -- 1-bit reset input
  S => S -- 1-bit set input
);

-- End of ODDR_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// ODDR: Output Double Data Rate Output Register with Set, Reset
//       and Clock Enable.
//       Virtex-4/5/6
// Xilinx HDL Libraries Guide, version 11.2

ODDR #(
  .DDR_CLK_EDGE("OPPOSITE_EDGE"), // "OPPOSITE_EDGE" or "SAME_EDGE"
  .INIT(1'b0), // Initial value of Q: 1'b0 or 1'b1
  .SRTYPE("SYNC") // Set/Reset type: "SYNC" or "ASYNC"
) ODDR_inst (
  .Q(Q), // 1-bit DDR output
  .C(C), // 1-bit clock input
  .CE(CE), // 1-bit clock enable input
  .D1(D1), // 1-bit data input (positive edge)
  .D2(D2), // 1-bit data input (negative edge)
  .R(R), // 1-bit reset
  .S(S) // 1-bit set
);

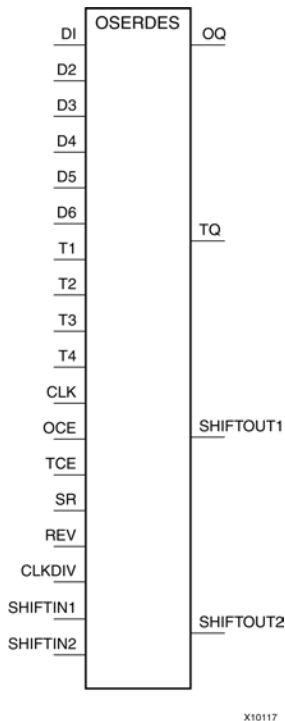
// End of ODDR_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

OSERDES

プリミティブ : Dedicated IOB Output Serializer



概要

このプリミティブを使用すると、同期インターフェイスを簡単にインプリメントできます。このモジュールを使用すると、FPGA のロジックリソースを節約でき、タイミングが複雑になるのを防ぎます。また、さまざまアプリケーションに対応した複数のクロック入力があり、SelectIO™ 機能と共に使用できます。

ポートの説明

ポート名	タイプ	幅	機能
OQ	出力	1	データパス出力。OSERDES モジュールのデータ出力です。このポートは、データ パラレル/シリアル コンバータの出力と IOB パッドのデータ入力を接続します。また、OSERDES モジュール内のすべてのサブモジュールをバイパスするようにコンフィギュレーションすることも可能です。
SHIFTOUT1、SHIFTOUT2	出力	1 (それぞれ)	データ入力を拡張するためのキャリー出力です。マスタの SHIFTIN1、SHIFTIN2 に接続します。
TQ	出力	1	トライステートパス出力。OSERDES モジュールのトライステート出力です。このポートは、トライステート パラレル/シリアル コンバータの出力と IOB パッドの制御入力を接続します。
CLK	入力	1	高速クロック入力。パラレル/シリアル コンバータを駆動するのに使用するクロック入力です。CLK ポートは、次のいずれかのクロックリソースで駆動します。 <ul style="list-style-type: none"> ・ クロック領域内の 10 個のグローバル クロック ライン ・ 4 個のリージョナル クロック ライン ・ 4 個のクロック I/O (隣接したクロック領域内)

ポート名	タイプ	幅	機能
			・ FPGA (パイパスを介す)
CLKDIV	入力	1	高速分周クロック入力。パラレル/シリアル コンバータを駆動するのに使用するクロック入力です。CLK ポートに接続されたクロックよりも低周波数に分周したクロックを入力する必要があります。CLKDIV のソースには、次のクロック リソースのいずれかを使用できます。 ・ クロック領域内の 10 個のグローバル クロック ライン ・ 4 個のリージョナル クロック ライン
D1 ~ D6	入力	1	パラレル データ入力。OSERDES モジュールにパラレル データが入力されるポート。このポートは FPGA に接続され、2 ~ 6 ビットにコンフィギュレーションできます。データ幅拡張モードでは、10 ビットまで拡張できます。
OCE	入力	1	パラレル/シリアル コンバータ (データ) クロック イネーブル。High の場合、データ パラレル/シリアル コンバータの出力がイネーブルになります。
SR	入力	1	セット/リセット入力。ストレージ エLEMENT の状態を SRVAL 属性で設定した状態にします。SRVAL = 1 の場合は 1、SRVAL = 0 の場合は 0 になります。リセットがセットよりも優先されます。
SHIFTIN1、SHIFTIN2	入力	1 (それぞれ)	データ入力を拡張するためのキャリー入力です。スレーブの SHIFTOUT1、SHIFTOUT2 に接続します。
T1 ~ T4	入力	1 (それぞれ)	パラレル トライステート入力。OSERDES モジュールにパラレル トライステート信号が入力されるポートです。このポートは FPGA に接続され、1 ~ 4 ビットにコンフィギュレーションできます。この機能は、データ幅拡張モードではサポートされません。
TCE	入力	1	パラレル/シリアル コンバータ (トライステート) クロック イネーブル。High の場合、トライステート信号パラレル/シリアル コンバータの出力がイネーブルになります。

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

OSERDES モジュールのデータ パラレル/シリアル コンバータは、2 ~ 6 ビットのパラレル データを取り込み、シリアル データに変換します。2 つの OSERDES モジュールをカスケード接続すると、データ入力幅を 7、8、または 10 ビットに拡張できます。拡張する場合は、1 つの OSERDES をマスタ モードにし、もう 1 つの OSERDES をスレーブ モードに設定して、スレーブの SHIFTOUT ポートをマスタの SHIFTIN ポートに接続します。スレーブでは、入力として D3 ~ D6 ポートのみを使用します。パラレル/シリアル コンバータは、SDR または DDR モードの両方で使用できます。

D1 ポートのデータ入力が、最初の出力ビットになります。このモジュールは、CLK および CLKDIV クロックで制御されます。次の表に、SDR および DDR の異なるモードにおける CLK と CLKDIV の関係を示します。

SDR のデータ幅	DDR のデータ幅	CLK	CLKDIV
2	4	2X	X
3	6	3X	X
4	8	4X	X
5	10	5X	X
6	–	6X	X
7	–	7X	X
8	–	8X	X

このブロックの出力は、FPGA の IOB パッドのデータ入力に接続されます。この IOB パッドには、SelectIO を使用して信号規格を設定できます。

パラレル/シリアル コンバータ (トライステート)

OSERDES モジュールのトライステート パラレル/シリアル コンバータは、4 ビットのパラレル トライステート信号を取り込み、シリアル トライステート信号に変換します。データ パラレル/シリアル コンバータとは異なり、トライステート パラレル/シリアル コンバータは信号幅を 5 ビット以上には拡張できません。このモジュールは、主に CLK および CLKDIV クロックで制御されます。このモードを使用するには、DATA_RATE_TQ および TRISTATE_WIDTH 属性を設定する必要があります。場合によっては、DATA_RATE_OQ および DATA_WIDTH を設定することも必要です。次の表に、使用する機能と属性の値を示します。

機能	DATA_RATE_TQ	TRISTATE_WIDTH
4 ビット DDR	DDR	4
1 ビット SDR	SDR	1
バッファ	BUF	1

このブロックの出力は、FPGA の IOB パッドのトライステート入力に接続されます。この IOB パッドには、SelectIO を使用して信号規格を設定できます。

データ幅の拡張

このエレメントでは、7 ビット幅以上のパラレル データを送信できます。ただし、トライステート出力は信号幅を拡張できません。7 ビット幅以上のデータを送信するには、エレメントを 2 つインスタンス化する必要があります。この 2 つは、隣接したマスタ/スレーブ ペアである必要があります。OSERDES_MODE 属性を MASTER または SLAVE に設定し、OSERDES のペアを区別する必要があります。また、マスタの SHIFTIN ポートをスレーブの SHIFTOUT ポートに接続します。SDR および DDR モードでは、データ幅 7、8、10 がサポートされています。次の表に、SDR および DDR モードで使用可能なデータ幅を示します。

モード	幅
SDR	2、3、4、5、6、7、8
DDR	4、6、8、10

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_RATE_OQ	文字列	SDR、DDR	DDR	データを CLK の各エッジで変化させるか、各立ち上がりエッジで変化させるかを指定
DATA_RATE_TQ	文字列	BUF、SDR、DDR	DDR	トライステート を CLK の各エッジで変化させるか、各立ち上がりエッジで変化させるか、バッファのコンフィギュレーションで変化させるかを指定
DATA_WIDTH	整数	2、3、4、5、6、7、8、10	4	DATA_RATE_OQ = DDR の場合は 4、6、8、10、DATA_RATE_OQ = SDR の場合は 2、3、4、5、6、7、8
INIT_OQ	2 進数	0、1	0	OQ 出力の初期値を指定
INIT_TQ	2 進数	0、1	0	TQ 出力の初期値を指定
SERDES_MODE	文字列	MASTER、SLAVE	MASTER	データ幅を拡張する場合に OSERDES モジュールがマスタかスレーブかを指定
SRVAL_OQ	2 進数	0、1	0	リセットをアサートした場合の OQ 出力の値を指定
SRVAL_TQ	2 進数	0、1	0	リセットをアサートした場合の TQ 出力の値を指定
TRISTATE_WIDTH	整数	1、2、4	4	DATA_RATE_TQ = DDR の場合は値は 2 または 4 です。DATA_RATE_TQ = SDR または BUF の場合は値は 1 です。

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;

-- OSERDES: Output SERDES
--      Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

OSERDES_inst : OSERDES
generic map (
    DATA_RATE_OQ => "DDR", -- Specify data rate to "DDR" or "SDR"
    DATA_RATE_TQ => "DDR", -- Specify data rate to "DDR", "SDR", or "BUF"
    DATA_WIDTH => 4, -- Specify data width - For DDR: 4,6,8, or 10
                        -- For SDR or BUF: 2,3,4,5,6,7, or 8
    INIT_OQ => '0', -- INIT for Q1 register - '1' or '0'
    INIT_TQ => '0', -- INIT for Q2 register - '1' or '0'
    SERDES_MODE => "MASTER", --Set SERDES mode to "MASTER" or "SLAVE"
    SRVAL_OQ => '0', -- Define Q1 output value upon SR assertion - '1' or '0'
    SRVAL_TQ => '0', -- Define Q1 output value upon SR assertion - '1' or '0'
    TRISTATE_WIDTH => 4) -- Specify parallel to serial converter width
                        -- When DATA_RATE_TQ = DDR: 2 or 4
                        -- When DATA_RATE_TQ = SDR or BUF: 1 "
port map (
    OQ => OQ, -- 1-bit output
    SHIFTOUT1 => SHIFTOUT1, -- 1-bit data expansion output
    SHIFTOUT2 => SHIFTOUT2, -- 1-bit data expansion output
    TQ => TQ, -- 1-bit 3-state control output
    CLK => CLK, -- 1-bit clock input
    CLKDIV => CLKDIV, -- 1-bit divided clock input
    D1 => D1, -- 1-bit parallel data input
    D2 => D2, -- 1-bit parallel data input
    D3 => D3, -- 1-bit parallel data input
    D4 => D4, -- 1-bit parallel data input

```

```

D5 => D5,      -- 1-bit parallel data input
D6 => D6,      -- 1-bit parallel data input
OCE => OCE,    -- 1-bit clcok enable input
REV => '0',    -- Must be tied to logic zero
SHIFTIN1 => SHIFTIN1, -- 1-bit data expansion input
SHIFTIN2 => SHIFTIN2, -- 1-bit data expansion input
SR => SR,      -- 1-bit set/reset input
T1 => T1,      -- 1-bit parallel 3-state input
T2 => T2,      -- 1-bit parallel 3-state input
T3 => T3,      -- 1-bit parallel 3-state input
T4 => T4,      -- 1-bit parallel 3-state input
TCE => TCE     -- 1-bit 3-state signal clock enable input
);

-- End of OSERDES_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// OSERDES: Source Synchronous Output Serializer
//          Virtex-4/5
// Xilinx HDL Libraries Guide, version 11.2

OSERDES #(
    .DATA_RATE_OQ("DDR"), // Specify data rate to "DDR" or "SDR"
    .DATA_RATE_TQ("DDR"), // Specify data rate to "DDR", "SDR", or "BUF"
    .DATA_WIDTH(4), // Specify data width - For DDR: 4,6,8, or 10
                        // For SDR or BUF: 2,3,4,5,6,7, or 8
    .INIT_OQ(1'b0), // INIT for OQ register - 1'b1 or 1'b0
    .INIT_TQ(1'b0), // INIT for OQ register - 1'b1 or 1'b0
    .SERDES_MODE("MASTER"), // Set SERDES mode to "MASTER" or "SLAVE"
    .SRVAL_OQ(1'b0), // Define OQ output value upon SR assertion - 1'b1 or 1'b0
    .SRVAL_TQ(1'b0), // Define TQ output value upon SR assertion - 1'b1 or 1'b0
    .TRISTATE_WIDTH(4) // Specify parallel to serial converter width
                        // When DATA_RATE_TQ = DDR: 2 or 4
                        // When DATA_RATE_TQ = SDR or BUF: 1
) OSERDES_inst (
    .OQ(OQ), // 1-bit data path output
    .SHIFTOUT1(SHIFTOUT1), // 1-bit data expansion output
    .SHIFTOUT2(SHIFTOUT2), // 1-bit data expansion output
    .TQ(TQ), // 1-bit 3-state control output
    .CLK(CLK), // 1-bit clock input
    .CLKDIV(CLKDIV), // 1-bit divided clock input
    .D1(D1), // 1-bit parallel data input
    .D2(D2), // 1-bit parallel data input
    .D3(D3), // 1-bit parallel data input
    .D4(D4), // 1-bit parallel data input
    .D5(D5), // 1-bit parallel data input
    .D6(D6), // 1-bit parallel data input
    .OCE(OCE), // 1-bit clock enable input
    .REV(1'b0), // Must be tied to logic zero
    .SHIFTIN1(SHIFTIN1), // 1-bit data expansion input
    .SHIFTIN2(SHIFTIN2), // 1-bit data expansion input
    .SR(SR), // 1-bit set/reset input
    .T1(T1), // 1-bit parallel 3-state input
    .T2(T2), // 1-bit parallel 3-state input
    .T3(T3), // 1-bit parallel 3-state input
    .T4(T4), // 1-bit parallel 3-state input
    .TCE(TCE) // 1-bit 3-state signal clock enable input
);

// End of OSERDES_inst instantiation

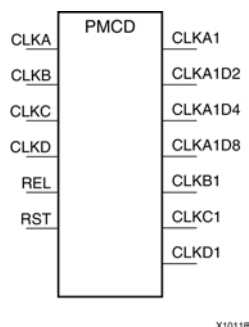
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

PMCD

プリミティブ : Phase-Matched Clock Divider



概要

このデザイン エLEMENTは、Virtex®-4 アーキテクチャに含まれているクロック リソースの 1 つで、次のようなクロック管理機能を備えています。

位相一致分周クロック

入力クロック CLKA の周波数を分周し、位相が一致したクロックを 4 つまで生成します。出力クロックの周波数は、入力クロック周波数を 1 (CLKA1)、2 (CLKA1D2)、4 (CLKA1D4)、および 8 (CLKA1D8) で分周した値です。出力クロック CLKA1、CLKA1D2、CLKA1D4、および CLKA1D8 は、立ち上がりエッジが揃っています。

位相一致クロック

入力クロック CLKA と PMCD のその他の入力クロック間のエッジ アライメント、位相関係、スキューが保持されます。CLKB、CLKC、および CLKD クロック入力に対しては、対応する CLKB1、CLKC1、および CLKD1 遅延出力があります。CLKA、CLKB、CLKC、CLKD には同じ遅延が挿入されるため、CLKA1、CLKB1、CLKC1、CLKD1 遅延出力では、エッジ アライメント、位相関係、および入力スキューが保持されます。

このデザイン エLEMENTは、グローバル バッファや DCM などその他のクロック リソースと共に使用できます。これらのクロック リソースを組み合わせることで、デザインの複雑なクロック ネットワークを柔軟に管理できます。

ポートの説明

ポート名	方向	機能
CLKA	入力	PMCD へのクロック入力。CLKA の周波数は、1、2、4、8 で分周できます。
CLKB、CLKC、CLKD	入力	PMCD へのクロック入力。これらのクロックは分周されませんが、クロック間の位相アライメントおよび位相関係を保持するために遅延が追加されます。
RST	入力	PMCD へのリセット入力。RST 信号がアサートされると、出力がすべて非同期で Low になります。RST 信号がディアサートされると、入力クロックに同期してすべての出力がトグルを開始します。
REL	入力	PMCD へのリリース入力。REL 信号がアサートされると、CLKA に同期して分周出力クロックがトグルを開始します。
CLKA1	出力	CLKA 入力と同じ周波数の出力で、遅延が追加されています。
CLKA1D2	出力	CLKA の周波数を 2 で分周した出力で、立ち上がりエッジは CLKA1 に揃えられています。
CLKA1D4	出力	CLKA の周波数を 4 で分周した出力で、立ち上がりエッジは CLKA1 に揃えられています。

ポート名	方向	機能
CLKA1D8	出力	CLKA の周波数を 8 で分周した出力で、立ち上がりエッジは CLKA1 に揃えられています。
CLKB1、CLKC1、CLKD1	出力	CLKB 入力と同じ周波数の出力で、遅延が追加されています。CLKB1 と CLKA1 間のスキューは、CLKB と CLKA 入力間のスキューと同じです。同様に、CLKC1 は CLKC に、CLKD1 は CLKD に遅延を追加したものです。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
EN_REL	ブール代数	FALSE、TRUE	FALSE	REL 信号のアサート時に、CLKA1D2、CLKA1D4、および CLKA1D8 出力のトグルを開始。 メモ: REL は CLKA 入力に同期。
RST_DEASSERT_CLK	文字列	CLKA、CLKB、CLKC、CLKD	CLKA	この属性で選択した PMCD 入力クロックに同期して、RST 信号のデアサートを認識

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;

-- PMCD: Phase-Matched Clock Divider Circuit
--      Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

PMCD_inst : PMCD
generic map (
    EN_REL => FALSE,           -- TRUE/FALSE to allow synchronous deassertion of RST
    RST_DEASSERT_CLK => "CLKA") -- Reset synchronization to which clock: CLKA, CLKB, CLKC or CLKD
port map (
    CLKA1 => CLKA1, -- Output CLKA divided by 1
    CLKA1D2 => CLKA1D2, -- Output CLKA divided by 2
    CLKA1D4 => CLKA1D4, -- Output CLKA divided by 4
    CLKA1D8 => CLKA1D8, -- Output CLKA divided by 8
    CLKB1 => CLKB1, -- Output phase matched CLKB
    CLKC1 => CLKC1, -- Output phase matched CLKC
    CLKD1 => CLKD1, -- Output phase matched CLKD
    CLKA => CLKA, -- Input CLKA
    CLKB => CLKB, -- Input CLKB
    CLKC => CLKC, -- Input CLKC
    CLKD => CLKD, -- Input CLKD
    REL => REL, -- PCMD release input
    RST => RST -- Active high reset input
);

-- End of PMCD_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// PMCD: Phase-Matched Clock Divider Circuit for Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

PMCD #(
    .EN_REL("FALSE"), // TRUE/FALSE to allow synchronous deassertion of RST
    .RST_DEASSERT_CLK("CLKA") // Reset synchronization to which clock: CLKA, CLKB, CLKC or CLKD
) PMCD_inst (
    .CLKA1(CLKA1), // Output CLKA divided by 1
    .CLKA1D2(CLKA1D2), // Output CLKA divided by 2
    .CLKA1D4(CLKA1D4), // Output CLKA divided by 4
    .CLKA1D8(CLKA1D8), // Output CLKA divided by 8
    .CLKB1(CLKB1), // Output phase matched CLKB
    .CLKC1(CLKC1), // Output phase matched CLKC
    .CLKD1(CLKD1), // Output phase matched CLKD
    .CLKA(CLKA), // Input CLKA
    .CLKB(CLKB), // Input CLKB
    .CLKC(CLKC), // Input CLKC
    .CLKD(CLKD), // Input CLKD
    .REL(REL), // PCMD release input
    .RST(RST) // Active high reset input
);

// End of PMCD_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

PPC405_ADV

プリミティブ : Primitive for the Power PC Core

概要

このデザイン エLEMENTは、PowerPC® アーキテクチャを基に開発された PowerPC エンベデッド環境アーキテクチャの 32 ビット インプリメンテーションで、Virtex®-4 用の PowerPC 405F6 プロセッサ コアです。このプロセッサ コアには、オンチップ メモリ ロジック (OCM)、APU コントローラ (Virtex-4 のみ)、およびバスケット ロジックおよびインターフェイスが含まれます。

PowerPC アーキテクチャでは、PowerPC ファミリのマイクロプロセッサのインプリメンテーション間で互換性を確実にするため、ソフトウェア モデルが提供されています。また、アプリケーション プログラム レベルでプロセッサのインプリメンテーションの互換性を保証するパラメータが定義されており、特定の要件を満たす PowerPC インプリメンテーションを柔軟に開発できます。

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

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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、トリステート出力、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)
);

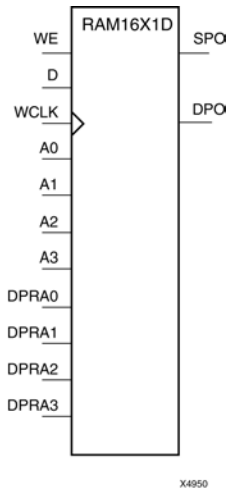
// End of PULLUP_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

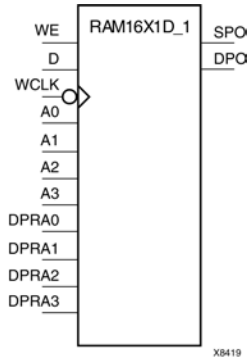
// End of RAM16X1D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM16X1D_1

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



概要

このエレメントは、クロックのネガティブ エッジで動作する 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) の独立した 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 High から Low に切り替わるときに、4 ビットの書き込みアドレスで選択されているワードにデータ入力 (D) の値がロードされます。書き込みを正しく行うには、WCLK が High から Low 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。

INIT 属性を使用すると、コンフィギュレーション中に RAM16X1D_1 を初期化できます。

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。

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				

ポートの説明

ポート名	方向	幅	機能
DPO	出力	1	読み出し専用 1 ビット データ出力
SPO	出力	1	読み出し/書き込み 1 ビット データ出力
A0	入力	1	読み出し/書き込み address[0] 入力
A1	入力	1	読み出し/書き込み address[1] 入力
A2	入力	1	読み出し/書き込み address[2] 入力
A3	入力	1	読み出し/書き込み address[3] 入力
D	入力	1	書き込み 1 ビット データ入力
DPRA0	入力	1	読み出し専用 address[0] 入力
DPRA1	入力	1	読み出し専用 address[1] 入力
DPRA2	入力	1	読み出し専用 address[2] 入力
DPRA3	入力	1	読み出し専用 address[3] 入力
WCLK	入力	1	書き込みクロック入力
WE	入力	1	書き込みイネーブル入力

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
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;

-- RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
--           All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X1D_1_inst : RAM16X1D_1
generic map (
    INIT => X"0000")
port map (
    DPO => DPO,      -- Read-only 1-bit data output for DPRA
    SPO => SPO,      -- R/W 1-bit data output for A0-A3
    A0 => A0,         -- R/W address[0] input bit
    A1 => A1,         -- R/W address[1] input bit
    A2 => A2,         -- R/W address[2] input bit
    A3 => A3,         -- R/W address[3] input bit
    D => D,           -- Write 1-bit data input
    DPRA0 => DPRA0,   -- Read-only address[0] input bit
    DPRA1 => DPRA1,   -- Read-only address[1] input bit
    DPRA2 => DPRA2,   -- Read-only address[2] input bit
    DPRA3 => DPRA3,   -- Read-only address[3] input bit
    WCLK => WCLK,     -- Write clock input
    WE => WE          -- Write enable input
);

-- End of RAM16X1D_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM16X1D_1: 16 x 1 negative edge write, asynchronous read dual-port distributed RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM16X1D_1 #(
    .INIT(16'h0000) // Initial contents of RAM
) RAM16X1D_1_inst (
    .DPO(DPO),      // Read-only 1-bit data output
    .SPO(SPO),      // R/W 1-bit data output
    .A0(A0),        // R/W address[0] input bit
    .A1(A1),        // R/W address[1] input bit
    .A2(A2),        // R/W address[2] input bit
    .A3(A3),        // R/W address[3] input bit
    .D(D),          // Write 1-bit data input
    .DPRA0(DPRA0),  // Read-only address[0] input bit
    .DPRA1(DPRA1),  // Read-only address[1] input bit
    .DPRA2(DPRA2),  // Read-only address[2] input bit
    .DPRA3(DPRA3),  // Read-only address[3] input bit
    .WCLK(WCLK),    // Write clock input
    .WE(WE)         // Write enable input
);

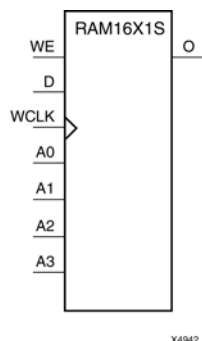
// End of RAM16X1D_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

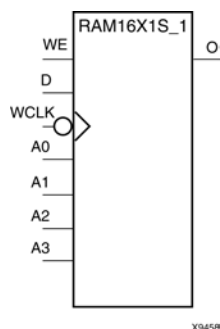
// End of RAM16X1S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM16X1S_1

プリミティブ : 16-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



概要

このエレメントはクロックの立ち下がりエッジで動作する 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 の場合、WCLK が High から Low 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。書き込みを正しく行うには、WCLK が High から Low 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 Low ですが、インバータを使用してアクティブ High 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エレメントを初期化できます。

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;

-- RAM16X1S_1: 16 x 1 negedge write distributed => LUT RAM
--           All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X1S_1_inst : RAM16X1S_1
generic map (
  INIT => X"0000")
port map (
  O => O,          -- RAM output
  A0 => A0,         -- RAM address[0] input
  A1 => A1,         -- RAM address[1] input
  A2 => A2,         -- RAM address[2] input
  A3 => A3,         -- RAM address[3] input
  D => D,          -- RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM16X1S_1_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAM16X1S_1: 16 x 1 negedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM16X1S_1 #(
  .INIT(16'h0000) // Initial contents of RAM
) RAM16X1S_1_inst (
  .O(O),          // RAM output
  .A0(A0),        // RAM address[0] input
  .A1(A1),        // RAM address[1] input
  .A2(A2),        // RAM address[2] input
  .A3(A3),        // RAM address[3] input
  .D(D),          // RAM data input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

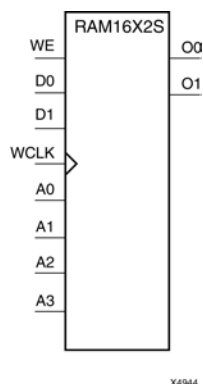
// End of RAM16X1S_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM16X2S

プリミティブ : 16-Deep by 2-Wide Static Synchronous RAM



概要

このエレメントは 16 ワード X 2 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 の場合、WCLK が Low から High に切り替わるときに、入力 (D1 ~ D0) 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。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 (O1 ~ O0) 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_xx 属性を使用すると、の RAM の初期値を指定できます。INIT_00 は出力 (O0) に対応する RAM のセルを初期化し、INIT_01 は出力 (O1) に対応するセルを初期化します。たとえば、RAM16X2S インスタンスは、INIT_00 および INIT_01 にそれぞれ 4 つの 16 進数値を指定して初期化します。RAM16X8S インスタンスは、INIT_00 ~ INIT_07 の 8 個の属性にそれぞれ 4 個の 16 進数値を指定して初期化します。RAM64X2S インスタンスは、INIT_00 および INIT_01 にそれぞれ 16 個の 16 進数値を指定して初期化します。

Virtex-4 デバイス以外では、このエレメントの初期値を直接指定することはできません。

論理表

入力			出力
WE (モード)	WCLK	D1 ~ D0	O1 ~ O0
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)	↑	D1 ~ D0	D1 ~ D0
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_00 ~ INIT_01	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;

-- RAM16X2S: 16 x 2 posedge write distributed => LUT RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X2S_inst : RAM16X2S
generic map (
  INIT_00 => X"0000", -- INIT for bit 0 of RAM
  INIT_01 => X"0000") -- INIT for bit 1 of RAM
port map (
  O0 => O0,      -- RAM data[0] output
  O1 => O1,      -- RAM data[1] output
  A0 => A0,      -- RAM address[0] input
  A1 => A1,      -- RAM address[1] input
  A2 => A2,      -- RAM address[2] input
  A3 => A3,      -- RAM address[3] input
  D0 => D0,      -- RAM data[0] input
  D1 => D1,      -- RAM data[1] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM16X2S_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAM16X2S: 16 x 2 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM16X2S #(
    .INIT_00(16'h0000), // Initial contents of bit 0 of RAM
    .INIT_01(16'h0000) // Initial contents of bit 1 of RAM
) RAM16X2S_inst (
    .O0(O0),           // RAM data[0] output
    .O1(O1),           // RAM data[1] output
    .A0(A0),           // RAM address[0] input
    .A1(A1),           // RAM address[1] input
    .A2(A2),           // RAM address[2] input
    .A3(A3),           // RAM address[3] input
    .D0(D0),           // RAM data[0] input
    .D1(D1),           // RAM data[1] input
    .WCLK(WCLK),       // Write clock input
    .WE(WE)            // Write enable input
);

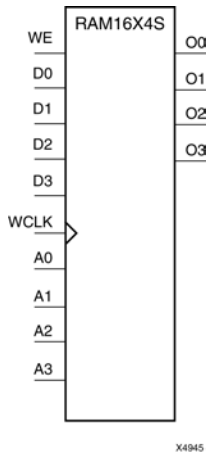
// End of RAM16X2S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM16X4S

プリミティブ : 16-Deep by 4-Wide Static Synchronous RAM



概要

このエレメントは 16 ワード X 4 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、入力 (D3 ~ D0) 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。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 (O3 ~ O0) 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。

論理表

入力			出力
WE (モード)	WCLK	D3:D0	O3:O0
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)	↑	D3:D0	D3:D0
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_00 ~ INIT_03	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;

-- RAM16X4S: 16 x 4 posedge write distributed => LUT RAM
--          Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X4S_inst : RAM16X4S
generic map (
  INIT_00 => X"0000", -- INIT for bit 0 of RAM
  INIT_01 => X"0000", -- INIT for bit 1 of RAM
  INIT_02 => X"0000", -- INIT for bit 2 of RAM
  INIT_03 => X"0000") -- INIT for bit 3 of RAM
port map (
  O0 => O0,      -- RAM data[0] output
  O1 => O1,      -- RAM data[1] output
  O2 => O2,      -- RAM data[2] output
  O3 => O3,      -- RAM data[3] output
  A0 => A0,      -- RAM address[0] input
  A1 => A1,      -- RAM address[1] input
  A2 => A2,      -- RAM address[2] input
  A3 => A3,      -- RAM address[3] input
  D0 => D0,      -- RAM data[0] input
  D1 => D1,      -- RAM data[1] input
  D2 => D2,      -- RAM data[2] input
  D3 => D3,      -- RAM data[3] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM16X4S_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAM16X4S: 16 x 4 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro, Spartan-3/3E/3A
// Xilinx HDL Libraries Guide, version 10.1.2

RAM16X4S #(
    .INIT_00(16'h0000), // INIT for bit 0 of RAM
    .INIT_01(16'h0000), // INIT for bit 1 of RAM
    .INIT_02(16'h0000), // INIT for bit 2 of RAM
    .INIT_03(16'h0000)  // INIT for bit 3 of RAM
) RAM16X4S_inst (
    .O0(O0),           // RAM data[0] output
    .O1(O1),           // RAM data[1] output
    .O2(O2),           // RAM data[2] output
    .O3(O3),           // RAM data[3] output
    .A0(A0),           // RAM address[0] input
    .A1(A1),           // RAM address[1] input
    .A2(A2),           // RAM address[2] input
    .A3(A3),           // RAM address[3] input
    .D0(D0),           // RAM data[0] input
    .D1(D1),           // RAM data[1] input
    .D2(D2),           // RAM data[2] input
    .D3(D3),           // RAM data[3] input
    .WCLK(WCLK),       // Write clock input
    .WE(WE)            // Write enable input
);

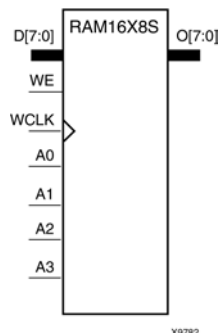
// End of RAM16X4S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM16X8S

プリミティブ : 16-Deep by 8-Wide Static Synchronous RAM



概要

このエレメントは 16 ワード X 8 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、入力 (D7 ~ D0) 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。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 (O7 ~ O0) 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。

論理表

入力			出力
WE (モード)	WCLK	D7:D0	O7:O0
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)	↑	D7:D0	D7:D0
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_00 ~ INIT_07	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;

-- RAM16X8S: 16 x 8 posedge write distributed => LUT RAM
--          Virtex-4 and Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM16X8S_inst : RAM16X8S
generic map (
    INIT_00 => X"0000", -- INIT for bit 0 of RAM
    INIT_01 => X"0000", -- INIT for bit 1 of RAM
    INIT_02 => X"0000", -- INIT for bit 2 of RAM
    INIT_03 => X"0000", -- INIT for bit 3 of RAM
    INIT_04 => X"0000", -- INIT for bit 4 of RAM
    INIT_05 => X"0000", -- INIT for bit 5 of RAM
    INIT_06 => X"0000", -- INIT for bit 6 of RAM
    INIT_07 => X"0000") -- INIT for bit 7 of RAM
port map (
    O => O,          -- 8-bit RAM data output
    A0 => A0,         -- RAM address[0] input
    A1 => A1,         -- RAM address[1] input
    A2 => A2,         -- RAM address[2] input
    A3 => A3,         -- RAM address[3] input
    D => D,          -- 8-bit RAM data input
    WCLK => WCLK,     -- Write clock input
    WE => WE         -- Write enable input
);

-- End of RAM16X8S_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM16X8S: 16 x 8 posedge write distributed (LUT) RAM
//          Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM16X8S #(
    .INIT_00(16'h0000), // INIT for bit 0 of RAM
    .INIT_01(16'h0000), // INIT for bit 1 of RAM
    .INIT_02(16'h0000), // INIT for bit 2 of RAM
    .INIT_03(16'h0000), // INIT for bit 3 of RAM
    .INIT_04(16'h0000), // INIT for bit 4 of RAM
    .INIT_05(16'h0000), // INIT for bit 5 of RAM
    .INIT_06(16'h0000), // INIT for bit 6 of RAM
    .INIT_07(16'h0000) // INIT for bit 7 of RAM
) RAM16X8S_inst (
    .O(O),           // 8-bit RAM data output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .D(D),           // 8-bit RAM data input
    .WCLK(WCLK),     // Write clock input
    .WE(WE)          // Write enable input
);

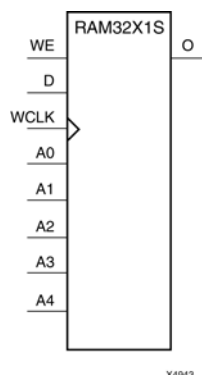
// End of RAM16X8S_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

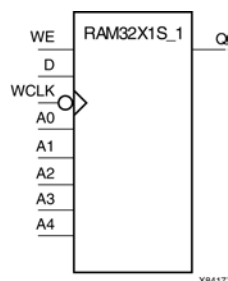
// End of RAM32X1S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM32X1S_1

プリミティブ : 32-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



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 High から Low 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 High から Low 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 Low ですが、インバータを使用してアクティブ High 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_1 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 = A4 ~ 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	32 ビット値	0	RAM、レジスタ、LUT 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_1: 32 x 1 negedge write distributed => LUT RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X1S_1_inst : RAM32X1S_1
generic map (
  INIT => X"00000000")
port map (
  O => O,          -- RAM output
  A0 => A0,         -- RAM address[0] input
  A1 => A1,         -- RAM address[1] input
  A2 => A2,         -- RAM address[2] input
  A3 => A3,         -- RAM address[3] input
  A4 => A4,         -- RAM address[4] input
  D => D,          -- RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X1S_1_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAM32X1S_1: 32 x 1 negedge write distributed (LUT) RAM
// All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM32X1S_1 #(
  .INIT(32'h00000000) // Initial contents of RAM
)RAM32X1S_1_inst (
  .O(O),              // RAM output
  .A0(A0),            // RAM address[0] input
  .A1(A1),            // RAM address[1] input
  .A2(A2),            // RAM address[2] input
  .A3(A3),            // RAM address[3] input
  .A4(A4),            // RAM address[4] input
  .D(D),              // RAM data input
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

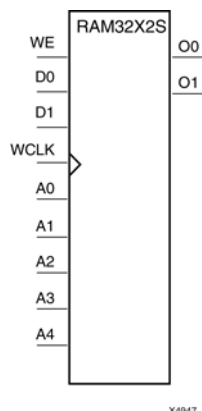
// End of RAM32X1S_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM32X2S

プリミティブ : 32-Deep by 2-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 2 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、入力 (D1 ~ D0) 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 (O1 ~ O0) 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_00 および INIT_01 属性を使用して RAM32X2S の初期値を指定できます。

論理表

入力			出力
WE (モード)	WCLK	D	O0-O1
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)	↑	D1:D0	D1:D0
1 (読み出し)	↓	X	データ
データ = A4 ~ 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_00	16 進数	32 ビット値	すべてゼロ	RAM の初期値を指定
INIT_01	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;

-- RAM32X2S: 32 x 2 posedge write distributed => LUT RAM
--           Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X2S_inst : RAM32X2S
generic map (
  INIT_00 => X"00000000", -- INIT for bit 0 of RAM
  INIT_01 => X"00000000") -- INIT for bit 1 of RAM
port map (
  O0 => O0,      -- RAM data[0] output
  O1 => O1,      -- RAM data[1] output
  A0 => A0,      -- RAM address[0] input
  A1 => A1,      -- RAM address[1] input
  A2 => A2,      -- RAM address[2] input
  A3 => A3,      -- RAM address[3] input
  A4 => A4,      -- RAM address[4] input
  D0 => D0,      -- RAM data[0] input
  D1 => D1,      -- RAM data[1] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM32X2S_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAM32X2S: 32 x 2 posedge write distributed (LUT) RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM32X2S #(
  .INIT_00(32'h00000000), // INIT for bit 0 of RAM
  .INIT_01(32'h00000000) // INIT for bit 1 of RAM
) RAM32X2S_inst (
  .O0(O0),      // RAM data[0] output
  .O1(O1),      // RAM data[1] output
  .A0(A0),      // RAM address[0] input
  .A1(A1),      // RAM address[1] input
  .A2(A2),      // RAM address[2] input
  .A3(A3),      // RAM address[3] input
  .A4(A4),      // RAM address[4] input
  .D0(D0),      // RAM data[0] input
  .D1(D1),      // RAM data[1] input
  .WCLK(WCLK),  // Write clock input
  .WE(WE)       // Write enable input
);

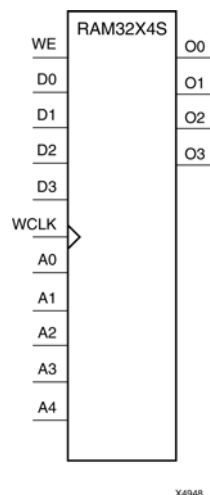
// End of RAM32X2S_inst instantiation
```


詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM32X4S

プリミティブ : 32-Deep by 4-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 4 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、入力 (D3 ~ D0) 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 (O3 ~ O0) 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。

論理表

入力			出力
WE	WCLK	D3 - D0	O3 - O0
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)	↑	D3:D0	D3:D0
1 (読み出し)	↓	X	データ
データ = A4 ~ 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_00	16 進数	32 ビット値	すべてゼロ	RAM の初期値を指定
INIT_01	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_02	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_03	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、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;

-- RAM32X4S: 32 x 4 posedge write distributed => LUT RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X4S_inst : RAM32X4S
generic map (
  INIT_00 => X"00000000", -- INIT for bit 0 of RAM
  INIT_01 => X"00000000", -- INIT for bit 1 of RAM
  INIT_02 => X"00000000", -- INIT for bit 2 of RAM
  INIT_03 => X"00000000") -- INIT for bit 3 of RAM
port map (
  O0 => O0,      -- RAM data[0] output
  O1 => O1,      -- RAM data[1] output
  O2 => O2,      -- RAM data[2] output
  O3 => O3,      -- RAM data[3] output
  A0 => A0,      -- RAM address[0] input
  A1 => A1,      -- RAM address[1] input
  A2 => A2,      -- RAM address[2] input
  A3 => A3,      -- RAM address[3] input
  A4 => A4,      -- RAM address[4] input
  D0 => D0,      -- RAM data[0] input
  D1 => D1,      -- RAM data[1] input
  D2 => D2,      -- RAM data[2] input
  D3 => D3,      -- RAM data[3] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM32X4S_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAM32X4S: 32 x 4 posedge write distributed (LUT) RAM
//          Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM32X4S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000), // INIT for bit 1 of RAM
    .INIT_02(32'h00000000), // INIT for bit 2 of RAM
    .INIT_03(32'h00000000)  // INIT for bit 3 of RAM
) RAM32X4S_inst (
    .O0(O0),           // RAM data[0] output
    .O1(O1),           // RAM data[1] output
    .O2(O2),           // RAM data[2] output
    .O3(O3),           // RAM data[3] output
    .A0(A0),           // RAM address[0] input
    .A1(A1),           // RAM address[1] input
    .A2(A2),           // RAM address[2] input
    .A3(A3),           // RAM address[3] input
    .A4(A4),           // RAM address[4] input
    .D0(D0),           // RAM data[0] input
    .D1(D1),           // RAM data[1] input
    .D2(D2),           // RAM data[2] input
    .D3(D3),           // RAM data[3] input
    .WCLK(WCLK),       // Write clock input
    .WE(WE)            // Write enable input
);

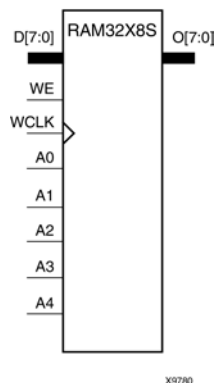
// End of RAM32X4S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM32X8S

プリミティブ : 32-Deep by 8-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 32 ワード X 8 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、入力 (D7 ~ D0) 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 (O7 ~ O0) 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。

論理表

入力			出力
WE (モード)	WCLK	D7:D0	O7:O0
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)	↑	D7:D0	D7:D0
1 (読み出し)	↓	X	データ
データ = A4 ~ 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_00	16 進数	32 ビット値	すべてゼロ	RAM の初期値を指定
INIT_01	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_02	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_03	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_04	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_05	16 進数	32 ビット値	すべてゼロ	ROM、RAM、レジスタ、ルックアップ テーブルの初期値を指定
INIT_06	16 進数	32 ビット値	すべてゼロ	RAM の初期値を指定
INIT_07	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;

-- RAM32X8S: 32 x 8 posedge write distributed => LUT RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X8S_inst : RAM32X8S
generic map (
  INIT_00 => X"00000000", -- INIT for bit 0 of RAM
  INIT_01 => X"00000000", -- INIT for bit 1 of RAM
  INIT_02 => X"00000000", -- INIT for bit 2 of RAM
  INIT_03 => X"00000000", -- INIT for bit 3 of RAM
  INIT_04 => X"00000000", -- INIT for bit 4 of RAM
  INIT_05 => X"00000000", -- INIT for bit 5 of RAM
  INIT_06 => X"00000000", -- INIT for bit 6 of RAM
  INIT_07 => X"00000000") -- INIT for bit 7 of RAM
port map (
  O => O,          -- 8-bit RAM data output
  A0 => A0,         -- RAM address[0] input
  A1 => A1,         -- RAM address[1] input
  A2 => A2,         -- RAM address[2] input
  A3 => A3,         -- RAM address[3] input
  A4 => A4,         -- RAM address[4] input
  D => D,          -- 8-bit RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X8S_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// RAM32X8S: 32 x 8 posedge write distributed (LUT) RAM
//           Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM32X8S #(
    .INIT_00(32'h00000000), // INIT for bit 0 of RAM
    .INIT_01(32'h00000000), // INIT for bit 1 of RAM
    .INIT_02(32'h00000000), // INIT for bit 2 of RAM
    .INIT_03(32'h00000000), // INIT for bit 3 of RAM
    .INIT_04(32'h00000000), // INIT for bit 4 of RAM
    .INIT_05(32'h00000000), // INIT for bit 5 of RAM
    .INIT_06(32'h00000000), // INIT for bit 6 of RAM
    .INIT_07(32'h00000000) // INIT for bit 7 of RAM
) RAM32X8S_inst (
    .O(O),           // 8-bit RAM data output
    .A0(A0),         // RAM address[0] input
    .A1(A1),         // RAM address[1] input
    .A2(A2),         // RAM address[2] input
    .A3(A3),         // RAM address[3] input
    .A4(A4),         // RAM address[4] input
    .D(D),           // 8-bit RAM data input
    .WCLK(WCLK),     // Write clock input
    .WE(WE)          // Write enable input
);

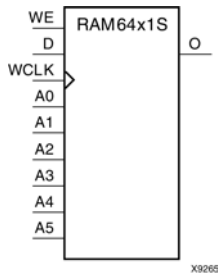
// End of RAM32X8S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

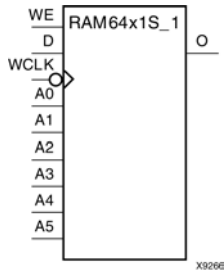
// End of RAM64X1S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM64X1S_1

プリミティブ : 64-Deep by 1-Wide Static Synchronous RAM with Negative-Edge Clock



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 High から Low 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。書き込みを正しく行うには、WCLK が High から Low 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 Low ですが、インバータを使用してアクティブ High 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

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;

-- RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
--           Spartan-3/3E/3A
-- Xilinx HDL Libraries Guide, version 11.2

RAM64X1S_1_inst : RAM64X1S_1
generic map (
    INIT => X"0000000000000000")
port map (
    O => O,          -- 1-bit data output
    A0 => A0,         -- Address[0] input bit
    A1 => A1,         -- Address[1] input bit
    A2 => A2,         -- Address[2] input bit
    A3 => A3,         -- Address[3] input bit
    A4 => A4,         -- Address[4] input bit
    A5 => A5,         -- Address[5] input bit
    D => D,           -- 1-bit data input
    WCLK => WCLK,     -- Write clock input
    WE => WE          -- Write enable input
);

-- End of RAM64X1S_1_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM64X1S_1: 64 x 1 negative edge write, asynchronous read single-port distributed RAM
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM64X1S_1 #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1S_1_inst (
    .O(O),          // 1-bit data output
    .A0(A0),        // Address[0] input bit
    .A1(A1),        // Address[1] input bit
    .A2(A2),        // Address[2] input bit
    .A3(A3),        // Address[3] input bit
    .A4(A4),        // Address[4] input bit
    .A5(A5),        // Address[5] input bit
    .D(D),          // 1-bit data input
    .WCLK(WCLK),    // Write clock input
    .WE(WE)         // Write enable input
);

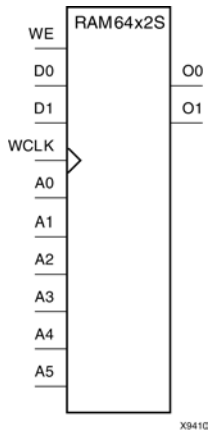
// End of RAM64X1S_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM64X2S

プリミティブ : 64-Deep by 2-Wide Static Synchronous RAM



概要

このデザイン エLEMENTは 64 ワード X 2 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、入力 (D1 ~ D0) 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。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 (O1 ~ O0) 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_00 および INIT_01 属性を使用してこのデザイン エLEMENTの初期値を指定できます。

論理表

入力			出力
WE (モード)	WCLK	D0 : D1	O0 : O1
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)	↑	D1 : D0	D1 : D0
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_00	16 進数	64 ビット値	すべてゼロ	RAM、レジスタ、LUT の初期値を指定
INIT_01	16 進数	64 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;

-- RAM64X2S: 64 x 2 positive edge write, asynchronous read single-port distributed RAM
--           Virtex-4/5
-- Xilinx HDL Libraries Guide, version 11.2

RAM64X2S_inst : RAM64X2S
generic map (
  INIT_00 => X"0000000000000000", -- INIT for bit 0 of RAM
  INIT_01 => X"0000000000000000") -- INIT for bit 1 of RAM
port map (
  O0 => O0,      -- Data[0] output
  O1 => O1,      -- Data[1] output bit
  A0 => A0,      -- Address[0] input bit
  A1 => A1,      -- Address[1] input bit
  A2 => A2,      -- Address[2] input bit
  A3 => A3,      -- Address[3] input bit
  A4 => A4,      -- Address[4] input bit
  A5 => A5,      -- Address[5] input bit
  D0 => D0,      -- Data[0] input
  D1 => D1,      -- Data[1] input
  WCLK => WCLK,  -- Write clock input
  WE => WE       -- Write enable input
);

-- End of RAM64X2S_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAM64X2S: 64 x 2 positive edge write, asynchronous read single-port distributed RAM
//           Virtex-II/II-Pro
// Xilinx HDL Libraries Guide, version 10.1.2

RAM64X2S #(
  .INIT_00(64'h0000000000000000), // INIT for RAM bit 0
  .INIT_01(64'h0000000000000000)) // INIT for RAM bit 1
) RAM64X2S_inst (
  .O0(O0),      // Data[0] output
  .O1(O1),      // Data[1] output bit
  .A0(A0),      // Address[0] input bit
  .A1(A1),      // Address[1] input bit
  .A2(A2),      // Address[2] input bit
  .A3(A3),      // Address[3] input bit
  .A4(A4),      // Address[4] input bit
  .A5(A5),      // Address[5] input bit
  .D0(D0),      // Data[0] input
  .D1(D1),      // Data[1] input
  .WCLK(WCLK),  // Write clock input
  .WE(WE)       // Write enable input
);

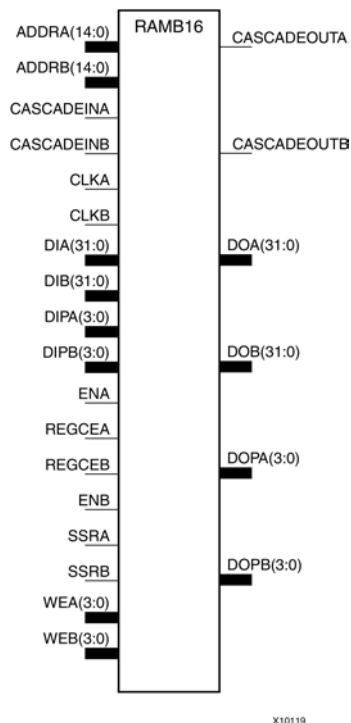
// End of RAM64X2S_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

RAMB16

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



概要

Virtex®-4 および上記のデバイスには、分散 RAM メモリに加え、18Kb のブロック RAM メモリが多数搭載されています。ブロック RAM メモリは真のデュアル ポート RAM であり、デバイス上で高速で離散型の大容量ブロック メモリとして使用できます。メモリは縦に並べられており、ブロック RAM メモリの総容量はデバイスのサイズによって異なります。18Kb ブロックをカスケード接続すると、タイミング遅延を最小限に抑えて、ビット数とワード数の多いメモリをインプリメントできます。

読み出し	読み出しは、クロック エッジに同期して行われます。読み出しアドレスが読み出しポートに取り込まれ、RAM アクセス時間後に保存されているデータが出力ラッチに送信されます。
書き込み	書き込みは、クロック エッジに同期して行われます。書き込みアドレスは書き込みポートに取り込まれ、入力データがメモリに保存されます。

書き込みの動作モード

書き込み中のポートでのデータ出力には、3 つのモードがあります。これらのモードでは、書き込み中に同じポートのデータ出力バスを使用できます。出力オプションは、コンフィギュレーションで指定します。オプションの選択により、各クロック サイクルでのブロック RAM メモリの効率が向上し、最大のバンド幅を使用したデザインが可能になります。

書き込みクロック エッジ後の出力ラッチのデータは、3 つのモードのいずれかを選択して指定します。

WRITE_FIRST (透過) モード	入力データがメモリに書き込まれると同時に、データ出力に送信されます (透過書き込み)。
READ_FIRST (書き込み前に読み出し) モード	書き込みアドレスに以前に保存されたデータが出力ラッチに送信され、それと同時に入力データがメモリに保存されます (書き込み前に読み出し)。
NO_CHANGE (変化なし) モード	書き込み中に出力ラッチは変化しません。

動作モードは、コンフィギュレーション中に設定します。各ポートに対し、属性を使用して 3 つのモードのいずれかを個別に設定できます。デフォルトでは、WRITE_FIRST に設定されています。

ポートの説明

出力ラッチの同期セット/リセット – SRVAL (SRVAL_A および SRVAL_B)

SRVAL_A および SRVAL_B (デュアル ポート) 属性は、SSR 入力のアサートした際の出力ラッチの値を定義します。次の表に示すように、SRVAL (SRVAL_A および SRVAL_B) 属性の幅がポート幅となります。

ポート幅およびデータ幅	DOP バス	DO バス	SRVAL
1	なし	<0>	1
2	なし	<1:0>	2
4	なし	<3:0>	4
9	<0>	<7:0>	(1+8) = 9
18	<1:0>	<15:0>	(2+16) = 18
36	<3:0>	<31:0>	(4 + 32) = 36

オプションの出力レジスタ切り替え – DO[A|B]_REG

RAMB16 の A/B 出力でのパイプライン レジスタの数を設定します。有効な値は 0 および 1 で、デフォルト値は 0 です。

出力レジスタのクロック反転 – INVERT_CLK_DO[A|B]_REG

TRUE に設定すると、RAMB16 の A/B 出力でのパイプライン レジスタのクロック入力が反転されます。デフォルト値は FALSE です。

拡張モード アドレス – RAM_EXTENSION_[A|B]

カスケード モードを使用する場合に、ブロック RAM の A ポートと B ポートを上位アドレス (UPPER) にするか、下位アドレス (LOWER) にするかを指定します。カスケード モードを使用するには、READ_WIDTH_[A/B] および WRITE_WIDTH_[A/B] を 1 に設定する必要があります。デフォルトでは NONE に設定されており、ブロック RAM はカスケード モードでは使用されません。

読み出し幅 – READ_WIDTH_[A|B]

ブロック RAM の A/B 読み出しポートの幅を指定します。有効な値は、0、1、2、4、9、18 および 36 で、デフォルト値は 0 です。両方のポートの READ_WIDTH_[A/B] 属性を同時に 0 に設定しないでください。

書き込み幅 – WRITE_WIDTH_[A|B]

ブロック RAM の A/B 書き込みポートの幅を指定します。有効な値は、0、1、2、4、9、18 および 36 で、デフォルト値は 0 です。

書き込みモード – WRITE_MODE_[A|B]

A/B 入力ポートの書き込みモードを指定します。有効な値は、WRITE_FIRST、READ_FIRST、および NO_CHANGE で、デフォルト値は WRITE_FIRST です。

RAMB16 のロケーション制約

ブロック RAM のインスタンスに LOC プロパティを指定すると、配置を制約できます。ブロック RAM を配置する位置の表記方法は、CLB 位置の表記方法とは異なっており、LOC プロパティを別のアレイでも簡単に使用できます。LOC プロパティは、次の形式で指定します。LOC = RAMB16_X#Y#

RAMB16_X0Y0 は、デバイスの左下にあるブロック RAM の位置を表します。

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	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
DOA_REG	整数	0、1	0	A ポート上の出力レジスタ (オプション)
DOB_REG	整数	0、1	0	B ポート上の出力レジスタ (オプション)
INIT_00 ~ INIT_39	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定
INIT_0A ~ INIT_0F	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定
INIT_1A ~ INIT_1F	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定
INIT_2A ~ INIT_2F	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定
INIT_3A ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定
INIT_A	16 進数	36 ビット値	すべてゼロ	A 出力ポートの初期値
INIT_B	16 進数	36 ビット値	すべてゼロ	B 出力ポートの初期値
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	パリティビットの初期値
INVERT_CLK_DOA_REG	ブール代数	FALSE、TRUE	FALSE	A ポート出力レジスタのクロックを反転
INVERT_CLK DOB_REG	ブール代数	FALSE、TRUE	FALSE	B ポート出力レジスタのクロックを反転
RAM_EXTENSION_A	文字列	LOWER、NONE、UPPER	NONE	カスケードする場合に上位アドレスにするか下位アドレスにするかを指定
RAM_EXTENSION_B	文字列	LOWER、NONE、UPPER	NONE	カスケードする場合に上位アドレスにするか下位アドレスにするかを指定
READ_WIDTH_A	整数	0、1、2、4、9、18、36	0	書き込みポート幅の指定
READ_WIDTH_B	整数	0、1、2、4、9、18、36	0	書き込みポート幅の指定

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	16 進数	36 ビット値	すべてゼロ	SSR をアサートした際の A ポート出力の値を指定
SRVAL_B	16 進数	36 ビット値	すべてゼロ	SSR をアサートした際の B ポート出力の値を指定
WRITE_MODE_A	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	デュアル ポート RAMB16 のポート A (Sn) の書き込みモードを指定
WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	デュアル ポート RAMB16 のポート B (Sn) の書き込みモードを指定
WRITE_WIDTH_A	整数	0、1、2、4、9、18、36	0	書き込みポート幅の指定
WRITE_WIDTH_B	整数	0、1、2、4、9、18、36	0	書き込みポート幅の指定

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;

-- RAMB16: 16k+2k Parity Paramatizable BlockRAM
--          Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2

RAMB16_inst : RAMB16
generic map (
    DOA_REG => 0, -- Optional output registers on the A port (0 or 1)

```

```
DOB_REG => 0, -- Optional output registers on the B port (0 or 1)
INIT_A => X"000000000", -- Initial values on A output port
INIT_B => X"000000000", -- Initial values on B output port
INVERT_CLK_DOA_REG => FALSE, -- Invert clock on A port output registers (TRUE or FALSE)
INVERT_CLK_DOB_REG => FALSE, -- Invert clock on B port output registers (TRUE or FALSE)
RAM_EXTENSION_A => "NONE", -- "UPPER", "LOWER" or "NONE" when cascaded
RAM_EXTENSION_B => "NONE", -- "UPPER", "LOWER" or "NONE" when cascaded
READ_WIDTH_A => 0, -- Valid values are 1,2,4,9,18 or 36
READ_WIDTH_B => 0, -- Valid values are 1,2,4,9,18 or 36
SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
-- "GENERATE_X_ONLY" or "NONE"
SRVAL_A => X"000000000", -- Port A output value upon SSR assertion
SRVAL_B => X"000000000", -- 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
WRITE_WIDTH_A => 0, -- Valid values are 1,2,4,9,18 or 36
WRITE_WIDTH_B => 0, -- Valid values are 1,2,4,9,18 or 36
-- The following INIT_xx declarations specify the initial contents of the RAM
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",
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",
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",
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 (
  CASCADEOUTA => CASCADEOUTA, -- 1-bit cascade output
  CASCADEOUTB => CASCADEOUTB, -- 1-bit cascade output
  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, -- 15-bit A port Address Input
  ADDRb => ADDRb, -- 15-bit B port Address Input
  CASCADEINA => CASCADEINA, -- 1-bit cascade A input
  CASCADEINB => CASCADEINB, -- 1-bit cascade B 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
  REGCEA => REGCEA, -- 1-bit A port register enable input
  REGCEB => REGCEB, -- 1-bit B port register 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 RAMB16_inst instantiation

```

Verilog 記述 (インスタネーション)

```

// RAMB16: Virtex-4 16k+2k Parity Paramatizable BlockRAM
// Xilinx HDL Libraries Guide, version 11.2

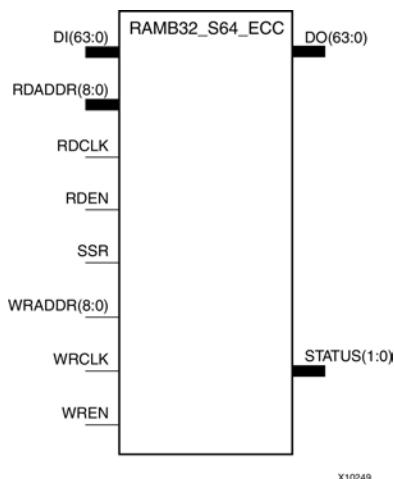
RAMB16 #(
  .DOA_REG(0), // Optional output registers on A port (0 or 1)
  .DOB_REG(0), // Optional output registers on B port (0 or 1)
  .INIT_A(36'h000000000), // Initial values on A output port
  .INIT_B(36'h000000000), // Initial values on B output port
  .INVERT_CLK_DOA_REG("FALSE"), // Invert clock on A port output registers ("TRUE" or "FALSE")
  .INVERT_CLK_DOB_REG("FALSE"), // Invert clock on B port output registers ("TRUE" or "FALSE")
  .RAM_EXTENSION_A("NONE"), // "UPPER", "LOWER" or "NONE" when cascaded
  .RAM_EXTENSION_B("NONE"), // "UPPER", "LOWER" or "NONE" when cascaded
  .READ_WIDTH_A(0), // Valid values are 1, 2, 4, 9, 18, or 36
  .READ_WIDTH_B(0), // Valid values are 1, 2, 4, 9, 18, or 36
  .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"
  .WRITE_WIDTH_A(2), // Valid values are 1, 2, 4, 9, 18, or 36
  .WRITE_WIDTH_B(0), // Valid values are 1, 2, 4, 9, 18, or 36

```

Virtex-4 ライブラリ ガイド (HDL 用)
UG619 (v11.3) 2009 年 9 月 16 日

RAMB32_S64_ECC

プリミティブ : 512 Deep by 64-Bit Wide Synchronous, Two-Port Block RAM with Built-In Error Correction



概要

縦に並ぶ 2 個のブロック RAM は、72 ビット幅 RAM の追加の 8 ビットを使用し、ビルトインのハミング エラー訂正コードを使用した 512 X 64 RAM としてコンフィギュレーションできます。この動作は自動的に行われます。書き込みで 8 個の保護ビットが生成され、これらのビットは読み出しでシングル エラーの訂正やダブル エラーの検知に使用されます。2 つのステータス出力で、エラーなし、シングル エラーの訂正、ダブル エラーの検知という 3 つの読み出し結果を表します。読み出しでは、メモリ アレイのエラーは訂正されず、DOUT に訂正されたデータが送信されるだけです。

エラー訂正コード (ECC) コンフィギュレーション オプションは、ほぼすべてのブロック RAM ペアで使用できますが、Virtex®-4 PowerPC® ブロックのすぐ上またはすぐ下にあるブロック RAM は使用できません。

ポートの説明

ポート名	方向	機能
DIN<63:0>	入力	データ入力バス
WRADDR<8:0>	入力	書き込みアドレス バス
RDADDR<8:0>	入力	読み出しアドレス バス
WREN	入力	ライト イネーブル。WREN = 1 の場合、データがメモリに書き込まれます。WREN = 0 の場合、書き込みはディスエーブルになります。
RDEN	入力	読み出しイネーブル。RDEN = 1 の場合、データがメモリから読み出されます。RDEN = 0 の場合、読み出しはディスエーブルになります。
SSR	入力	セット/リセット出力レジスタ (メモリの内容はセット/リセットしない)
WRCLK	入力	書き込みクロック
RDCLK	入力	読み出しクロック
DOUT<63:0>	出力	データ出力バス
STATUS<1:0>(1)	出力	エラー ステータス バス

メモ : ブロック RAM ECC ロジックにインプリメントされたハミング コードは、検知可能なエラーなし、DOUT でのシングルビット エラーの検知と訂正 (メモリでは訂正されない)、ダブルビット エラーの検知という 3 つの状態のうちいずれかを検知します。STATUS<1:0> は、これらの状態を表します。

STATUS[1:0]	機能
0	ビット エラーなし
1	シングルビット エラー。ブロック RAM ECC マクロは、シングルビット エラーを検知し、自動的に訂正します。
10	ダブルビット エラー。ブロック RAM ECC マクロは、ダブルビット エラーを検知します。
11	中間状態。予測されないステータスです。エラー ステータス バスを正しく機能させるには、データにシングルビット エラーまたはダブルビット エラー以外のエラーが含まれないようにする必要があります。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DO_REG	整数	0、1	0	A ポート上の出力レジスタ (オプション)
SIM_COLLISION_CHECK	文字列	ALL、NONE、WARNING_ONLY、GENERATE_X_ONLY	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>

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;

-- RAMB32_S64_ECC: Virtex-4 512 x 64 Error Correction BlockRAM
-- Xilinx HDL Libraries Guide, version 11.2

RAMB32_S64_ECC_inst: RAMB32_S64_ECC_inst
  generic map(
    DO_REG => 0, -- Optional output registers (0 or 1)
    SIM_COLLISION_CHECK => "ALL") -- Collision check enable "ALL", "WARNING_ONLY", "GENERATE_X_ONLY"
  port map (
    DO => DO,          -- 64-bit output data
    STATUS => STATUS,  -- 2-bit status output
    DI => DI,          -- 64-bit data input
    RDADDR => RDADDR,  -- 9-bit data address input
    RDCLK => RDCLK,    -- 1-bit read clock input
    RDEN => RDEN,      -- 1-bit read enable input
    SSR => '0',        -- Always tie to ground
    WRADDR => WRADDR,  -- 9-bit write address input
    WRCLK => WRCLK,    -- 1-bit write clock input
    WREN => WREN       -- 1-bit write enable input
  );

-- End of RAMB32_S64_ECC_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB32_S64_ECC: Virtex-4 512 x 64 Error Correction BlockRAM
// Xilinx HDL Libraries Guide, version 11.2

RAMB32_S64_ECC #(
  .DO_REG(0),          // Optional output registers (0 or 1)
  .SIM_COLLISION_CHECK("ALL") // Collision check enable "ALL",
                                // "WARNING_ONLY", "GENERATE_X_ONLY"
) RAMB32_S64_ECC_inst (
  .DO(DO),             // 64-bit output data
  .STATUS(STATUS),    // 2-bit status output
  .DI(DI),             // 64-bit data input
  .RDADDR(RDADDR),    // 9-bit data address input
  .RDCLK(RDCLK),       // 1-bit read clock input
  .RDEN(RDEN),         // 1-bit read enable input
  .SSR(1'b0),          // Always tie to ground
  .WRADDR(WRADDR),    // 9-bit write address input
  .WRCLK(WRCLK),       // 1-bit write clock input
  .WREN(WREN)          // 1-bit write enable input
);

// End of RAMB32_S64_ECC_inst instantiation

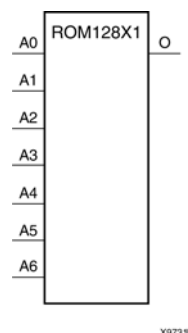
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

ROM128X1

プリミティブ : 128-Deep by 1-Wide ROM



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]
);

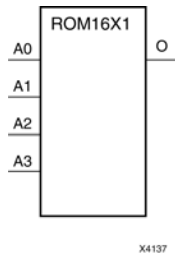
// End of ROM128X1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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]
);

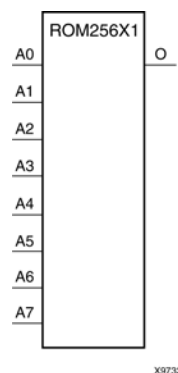
// End of ROM16X1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

ROM256X1

プリミティブ : 256-Deep by 1-Wide ROM



X9732

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]
);

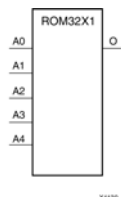
// End of ROM256X1_inst instantiation
```


詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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]
);

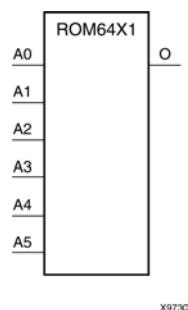
// End of ROM32X1_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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]
);

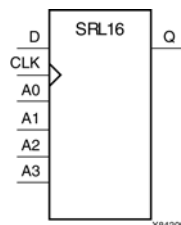
// End of ROM64X1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

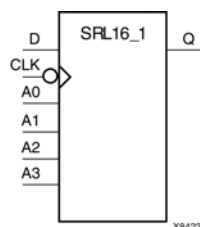
// End of SRL16_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

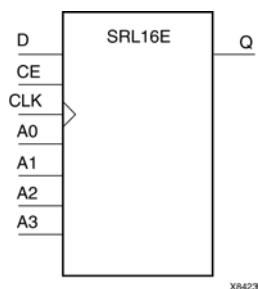
// End of SRL16_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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

インスタンシエーション	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

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 (インスタンシエーション)

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;

-- 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 (インスタンスレーション)

```
// 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
);

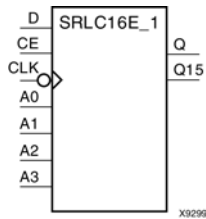
// End of SRL16E_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

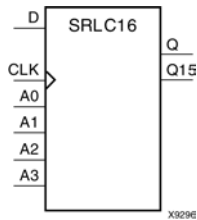
// End of SRL16E_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

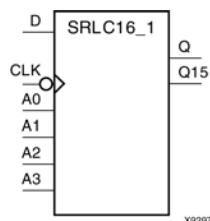
// End of SRLC16_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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	
Am	CLK	D	Q	Q15
Am	X	X	Q(Am)	変化なし
Am	↓	D	Q(Am - 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
);

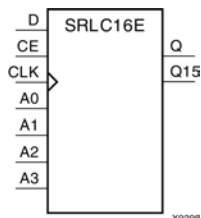
// End of SRLC16_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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
);

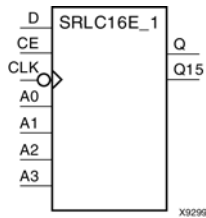
// End of SRLC16E_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

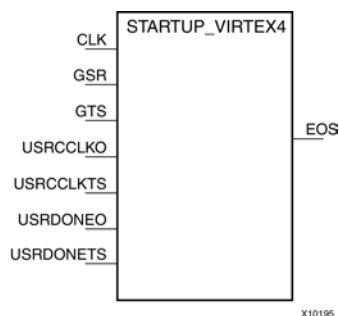
// End of SRLC16E_1_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

STARTUP_VIRTEX4

プリミティブ : Virtex®-4 User Interface to Configuration Clock, Global Reset, Global 3-State Controls, and Other Configuration Signals



概要

このデザイン エLEMENTでグローバル セット/リセット (GSR)、グローバル トライステート (GTS) 制御、コンフィギュレーション クロックをアクティベートし、また、コンフィギュレーション後の DONE および CLK ピンも制御できます。

ポートの説明

ポート名	方向	幅	機能
EOS	出力	1	EOS 信号
CLK	入力	1	クロック入力
GTS	入力	1	GTS 制御
GSR	入力	1	Global Set/Reset (GSR)
USRCCLKO	入力	1	外部 CCLK ピンを駆動します。
USRCCLKTS	入力	1	アサートされると、CCLK ピンがトライステートになります。
USRDONEO	入力	1	外部 DONE ピンを駆動します。
USRDONETS	入力	1	アサートされると、DONE ピンがトライステートになります。

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

メモ : ブロック RAM、LUT RAM、デジタル クロック マネージャ (DCM)、シフトレジスタ LUT (SRL16、SRL16_1、SRL16E、SRL16E_1、SRLC16、SRLC16_1、SRLC16E、SRLC16E_1) は、セットもリセットもされません。

BSCAN がディスエーブルで EXTEST 命令が実行されていない場合、コンフィギュレーションの後にグローバル トライステート信号 (GTS) が High になると、すべての IOB 出力がハイ インピーダンスになり、デバイスの出力が回路から切り離されます。ただし、入力はアクティブのままです。

CLK 入力を使用すると、JTAG の TCK または CCCLK ピンにクロックを供給する必要はなく、コンフィギュレーション スタートアップ シーケンスに指定の IO を使用してクロックを供給できます。この機能をイネーブルにするには、BitGen でビットストリームを生成する際に、スタートアップ クロックを userclk に設定しておく必要があります。

USRCLK0/TS および USRDONE0/TS は、外部 DONE および CCLK ピンを制御するために使用します。ROM データの FPGA への読み込みなど、さまざまなアプリケーションで、STARTUP_VIRTEX4 を USR_ACCESS_VIRTEX4 プリミティブと共に使用できます。詳細は、「USR_ACCESS_VIRTEX4」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;

-- STARTUP_VIRTEX4: Startup primitive for GSR, GTS or startup sequence
-- control. Virtex-4
-- Xilinx HDL Libraries Guide, version 11.2
STARTUP_VIRTEX4_inst : STARTUP_VIRTEX4
port map (
    EOS => EOS,          -- End of Startup 1-bit output
    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)
    USRCCLK0 => USRCCLK0, -- USRCCLK0 1-bit input
    USRCCLKTS => USRCCLKTS, -- USRCCLKTS 1-bit input
    USRDONE0 => USRDONE0, -- USRDONE0 1-bit input
    USRDONETS => USRDONETS -- USRDONETS 1-bit input
);

-- End of STARTUP_VIRTEX4_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// STARTUP_VIRTEX4: Startup primitive for GSR, GTS or startup sequence
// control. Virtex-4
// Xilinx HDL Libraries Guide, version 11.2

STARTUP_VIRTEX4 STARTUP_VIRTEX4_inst (
    .EOS(EOS), // End Of Startup 1-bit output
    .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)
    .USRCCLK0(USRCCLK0), // USERCLK0 1-bit input
    .USRCCLKTS(USRCCLKTS), // USERCLKTS 1-bit input
    .USRDONE0(USRDONE0), // USRDONE0 1-bit input
    .USRDONETS(USRDONETS) // USRDONETS 1-bit input
);

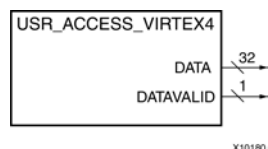
// End of STARTUP_VIRTEX4_inst instantiation
```

詳細情報

- [Virtex-4 FPGA ユーザー ガイド](#)
- [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

USR_ACCESS_VIRTEX4

プリミティブ : 32-Bit Register with a 32-Bit DATA Bus and a DATAVALID Port



概要

このデザイン エLEMENTは、ビットストリームからのデータに FPGA から直接アクセスできるようにする 32 ビットレジスタです。このモジュールには、32 ビット DATA バスと DATAVALID の 2 つの出力があります。コンフィギュレーション データソース クロックには、CCLK または TCK を使用します。

このブロックは、FPGA がコンフィギュレーションされた後に、ビットストリーム データ格納ソース (PROM など) のデータに FPGA からアクセスできるようにします。この機能を使用するには、STARTUP_VIRTEX4 ブロックもインスタンスエートする必要があります。STARTUP_VIRTEX4 ブロックには、EOS (End-Of-Startup) 信号がアサートされた後に CCLK および DONE ピンをユーザーが使用できるようにする入力があります。これらのピンは、USR_CCLK_O、USR_CCLK_TS、USR_DONE_O、および USR_DONE_TS です。DONE ピンが High になると PROM がリセットされてしまうので、BitGen の -g DONE_cycle: 7 オプションを使用して、DONE ピンが High にならないようにする必要があります。USR_CCLK_O ピンは、FPGA 内の制御クロックに接続します。PROM には、USR_ACCESS レジスタをターゲットとしたデータ パケットが含まれている必要があります。EOS がアサートされたら、USR_CCLK_TS を Low に保持した状態で、USR_CCLK_O ピンにクロックを供給するとデータ パケットを読み込むことができます。この使用法では、USR_CCLK_TS を Low に接続しておくこともできます。

また、BRAM または LUTRAM で定数を保存する代わりに、USR_ACCESS レジスタに 32 ビットの定数値 1 つを保存できます。

ポートの説明

ポート名	方向	幅	機能
DATA	出力	32	FPGA でビットストリーム データの格納ソースからのデータにアクセスできるようにする 32 ビットレジスタです。
DATAVALID	出力	1	DATA バスの値が有効 (新規) であるかを示す。真の場合、コンフィギュレーション データソース クロックの 1 サイクル間 High にアサートされます。

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

このモジュールを使用してビットストリーム データ格納ソース (PROM など) のデータに FPGA からアクセスできるようにする場合は、STARTUP_VIRTEX4 ブロックもインスタンスエートする必要があります。STARTUP_VIRTEX4 モジュールには、EOS (End-Of-Startup) 信号がアサートされた後に CCLK および DONE ピンをユーザーが使用できるようにする入力があります。これらのピンは、USR_CCLK_O、USR_CCLK_TS、USR_DONE_O、および USR_DONE_TS です。

USR_CCLK_O ピンは、FPGA 内の制御クロックに接続します。データ格納ソースには、USR_ACCESS_VIRTEX4 レジスタをターゲットとしたデータ パケットが含まれている必要があります。EOS がアサートされたら、USR_CCLK_TS を Low に保持した状態で、USR_CCLK_O ピンにクロックを供給するとデータ パケットを読み込むことができます。この使用法では、USR_CCLK_TS を Low に接続しておくこともできます。

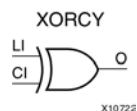
また、このモジュールを使用する場合は、BitGen の -g DONE_cycle: 7 オプションを使用して、DONE ピンが High にならないようにする必要があります。DONE が High にアサートされると、PROM がリセットされてしまいます。

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート : DC 特性およびスイッチ特性](#)

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
);

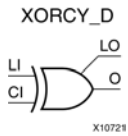
// End of XORCY_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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 (インスタンスレーション)

```
// 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
);

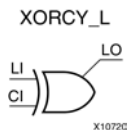
// End of XORCY_D_inst instantiation
```

詳細情報

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)

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

- ・ [Virtex-4 FPGA ユーザー ガイド](#)
- ・ [Virtex-4 FPGA データシート: DC 特性およびスイッチ特性](#)