

Virtex-5 ライブラリ ガイド (HDL 用)

UG621 (v11.3) 2009 年 9 月 16 日

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

ザイリンクス商標および著作権情報



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

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

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

目次

ザイリックス商標および著作権情報	2
1 : このマニュアルについて	15
デザイン エLEMENT	15
デザインの入力方法	16
2 : ターゲット デバイス変更後のデザイン エLEMENT	17
3 : UniMacro について	21
BRAM_SDP_MACRO	22
概要	22
ポートの説明	22
コンフィギュレーション表	23
デザインの入力方法	23
使用可能な属性	23
詳細情報	30
BRAM_SINGLE_MACRO	31
概要	31
ポートの説明	31
コンフィギュレーション表	32
デザインの入力方法	34
使用可能な属性	34
詳細情報	39
BRAM_TDP_MACRO	40
概要	40
ポートの説明	40
コンフィギュレーション表	41
デザインの入力方法	43
使用可能な属性	43
詳細情報	49
FIFO_DUALCLOCK_MACRO	50
概要	50
ポートの説明	50
コンフィギュレーション表	51
デザインの入力方法	51
使用可能な属性	52
詳細情報	53
FIFO_SYNC_MACRO	54
概要	54
ポートの説明	54
コンフィギュレーション表	55
デザインの入力方法	55
使用可能な属性	56
詳細情報	57
4 : ファンクション別分類	59
5 : デザイン エLEMENT	65
BSCAN_VIRTEX5	66
概要	66
ポートの説明	66
デザインの入力方法	66
使用可能な属性	67
詳細情報	67
BUFCF	68
概要	68
デザインの入力方法	68
詳細情報	69
BUFG	70
概要	70

ポートの説明	70
デザインの入力方法	70
詳細情報	71
BUFGCE	72
概要	72
論理表	72
デザインの入力方法	72
詳細情報	73
BUFGCTRL	74
概要	74
ポートの説明	74
デザインの入力方法	75
使用可能な属性	75
詳細情報	75
BUFGMUX_CTRL	76
概要	76
ポートの説明	76
デザインの入力方法	76
詳細情報	77
BUFIO	78
概要	78
ポートの説明	78
デザインの入力方法	78
詳細情報	79
BUFR	80
概要	80
ポートの説明	80
デザインの入力方法	80
使用可能な属性	80
詳細情報	81
CAPTURE_VIRTEX5	82
概要	82
ポートの説明	82
デザインの入力方法	82
使用可能な属性	82
詳細情報	83
CARRY4	84
概要	84
ポートの説明	84
デザインの入力方法	85
詳細情報	85
CFGLUT5	86
概要	86
ポートの説明	86
デザインの入力方法	87
使用可能な属性	87
詳細情報	88
CRC32	89
概要	89
ポートの説明	89
デザインの入力方法	90
使用可能な属性	90
詳細情報	90
CRC64	91
概要	91
ポートの説明	91
デザインの入力方法	92
使用可能な属性	92
詳細情報	92

DCIRESET	93
概要	93
ポートの説明	93
デザインの入力方法	93
使用可能な属性	93
詳細情報	94
DCM_ADV	95
概要	95
ポートの説明	95
デザインの入力方法	98
使用可能な属性	99
詳細情報	101
DSP48E	102
概要	102
ポートの説明	102
デザインの入力方法	105
使用可能な属性	105
詳細情報	109
FDCPE	110
概要	110
論理表	110
ポートの説明	110
デザインの入力方法	111
使用可能な属性	111
詳細情報	112
FDCPE_1	113
概要	113
論理表	113
ポートの説明	113
デザインの入力方法	114
使用可能な属性	114
詳細情報	115
FDRSE	116
概要	116
論理表	116
デザインの入力方法	116
使用可能な属性	116
詳細情報	117
FDRSE_1	118
概要	118
論理表	118
デザインの入力方法	118
使用可能な属性	118
詳細情報	119
FIFO18	120
概要	120
ポートの説明	120
デザインの入力方法	121
使用可能な属性	121
詳細情報	123
FIFO18_36	124
概要	124
ポートの説明	124
デザインの入力方法	125
使用可能な属性	125
詳細情報	127
FIFO36	128
概要	128
ポートの説明	128

デザインの入力方法.....	129
使用可能な属性.....	130
詳細情報	131
FIFO36_72	132
概要.....	132
ポートの説明	132
デザインの入力方法.....	133
使用可能な属性.....	134
詳細情報	135
FRAME_ECC_VIRTEX5.....	136
概要.....	136
ポートの説明	136
デザインの入力方法.....	136
詳細情報	137
GTP_DUAL	138
概要.....	138
デザインの入力方法.....	138
詳細情報	138
GTX_DUAL.....	139
概要.....	139
デザインの入力方法.....	139
詳細情報	139
IBUF.....	140
概要.....	140
ポートの説明	140
デザインの入力方法.....	140
使用可能な属性.....	140
詳細情報	141
IBUFDS.....	142
概要.....	142
論理表.....	142
ポートの説明	142
デザインの入力方法.....	142
使用可能な属性.....	143
詳細情報	143
IBUFG	144
概要.....	144
ポートの説明	144
デザインの入力方法.....	144
使用可能な属性.....	144
詳細情報	145
IBUFGDS	146
概要.....	146
論理表.....	146
ポートの説明	146
デザインの入力方法.....	146
使用可能な属性.....	147
詳細情報	147
ICAP_VIRTEX5.....	148
概要.....	148
ポートの説明	148
デザインの入力方法.....	148
使用可能な属性.....	148
詳細情報	149
IDDR	150
概要.....	150
ポートの説明	150
デザインの入力方法.....	151
使用可能な属性.....	151

詳細情報	152
IDDR_2CLK	153
概要	153
ポートの説明	153
デザインの入力方法	154
使用可能な属性	154
詳細情報	155
IDELAY	156
概要	156
ポートの説明	156
デザインの入力方法	158
使用可能な属性	158
詳細情報	159
IDELAYCTRL	160
概要	160
ポートの説明	160
デザインの入力方法	160
詳細情報	161
IOBUF	162
概要	162
論理表	162
ポートの説明	162
デザインの入力方法	162
使用可能な属性	163
詳細情報	164
IOBUFDS	165
概要	165
論理表	165
ポートの説明	165
デザインの入力方法	165
使用可能な属性	166
詳細情報	166
IODELAY	167
概要	167
ポートの説明	167
デザインの入力方法	168
使用可能な属性	168
詳細情報	170
ISERDES_NODELAY	171
概要	171
ポートの説明	171
デザインの入力方法	172
使用可能な属性	172
詳細情報	173
KEEPER	174
概要	174
ポートの説明	174
デザインの入力方法	174
詳細情報	175
KEY_CLEAR	176
概要	176
ポートの説明	176
デザインの入力方法	176
詳細情報	177
LDCPE	178
概要	178
論理表	178
ポートの説明	178
デザインの入力方法	179

使用可能な属性.....	179
詳細情報	180
LUT5	181
概要.....	181
論理表.....	182
ポートの説明	183
デザインの入力方法.....	183
使用可能な属性.....	183
詳細情報	184
LUT5_D.....	185
概要.....	185
論理表.....	186
ポートの説明	187
デザインの入力方法.....	187
使用可能な属性.....	187
詳細情報	188
LUT5_L.....	189
概要.....	189
論理表.....	190
ポートの説明	191
デザインの入力方法.....	191
使用可能な属性.....	191
詳細情報	192
LUT6	193
概要.....	193
論理表.....	193
ポートの説明	195
デザインの入力方法.....	196
使用可能な属性.....	196
詳細情報	197
LUT6_2	198
概要.....	198
論理表.....	198
ポートの説明	200
デザインの入力方法.....	200
使用可能な属性.....	201
詳細情報	201
LUT6_D.....	202
概要.....	202
論理表.....	202
ポートの説明	204
デザインの入力方法.....	205
使用可能な属性.....	205
詳細情報	206
LUT6_L.....	207
概要.....	207
論理表.....	207
ポートの説明	209
デザインの入力方法.....	210
使用可能な属性.....	210
詳細情報	211
MUXF7	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
詳細情報	213
MUXF7_D	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
詳細情報	215
MUXF7L	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
詳細情報	217
MUXF8	218
概要.....	218
論理表.....	218
ポートの説明	218
デザインの入力方法.....	218
詳細情報	219
MUXF8_D	220
概要.....	220
論理表.....	220
ポートの説明	220
デザインの入力方法.....	220
詳細情報	221
MUXF8_L	222
概要.....	222
論理表.....	222
ポートの説明	222
デザインの入力方法.....	222
詳細情報	223
OBUF	224
概要.....	224
ポートの説明	224
デザインの入力方法.....	224
使用可能な属性.....	224
詳細情報	225
OBUFDS	226
概要.....	226
論理表.....	226
ポートの説明	226
デザインの入力方法.....	226
使用可能な属性.....	226
詳細情報	227
OBUFT	228
概要.....	228
論理表.....	228
ポートの説明	228
デザインの入力方法.....	228
使用可能な属性.....	229
詳細情報	229
OBUFTDS	230
概要.....	230
論理表.....	230
ポートの説明	230
デザインの入力方法.....	230
使用可能な属性.....	230
詳細情報	231
ODDR.....	232
概要.....	232
ポートの説明	232
デザインの入力方法.....	232

使用可能な属性.....	233
詳細情報	233
OSERDES	234
概要	234
ポートの説明	234
デザインの入力方法.....	235
使用可能な属性.....	237
詳細情報	238
PLL_ADV.....	239
概要	239
ポートの説明	239
デザインの入力方法.....	240
使用可能な属性.....	241
詳細情報	245
PLL_BASE	246
概要	246
ポートの説明	246
デザインの入力方法.....	247
使用可能な属性.....	247
詳細情報	249
PPC440.....	250
概要	250
デザインの入力方法.....	250
詳細情報	250
PULLDOWN	251
概要	251
ポートの説明	251
デザインの入力方法.....	251
詳細情報	252
PULLUP.....	253
概要	253
ポートの説明	253
デザインの入力方法.....	253
詳細情報	254
RAM128X1D	255
概要	255
ポートの説明	255
デザインの入力方法.....	256
使用可能な属性.....	256
詳細情報	257
RAM16X1D_1	258
概要	258
論理表.....	258
ポートの説明	259
デザインの入力方法.....	259
使用可能な属性.....	259
詳細情報	260
RAM16X1S_1.....	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	262
RAM16X2S	263
概要	263
論理表.....	263
デザインの入力方法.....	264
使用可能な属性.....	264
詳細情報	265

RAM16X4S	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.....	267
詳細情報	268
RAM16X8S	269
概要.....	269
論理表.....	269
デザインの入力方法.....	269
使用可能な属性.....	269
詳細情報	270
RAM256X1S	271
概要.....	271
ポートの説明	271
デザインの入力方法.....	271
使用可能な属性.....	272
詳細情報	272
RAM32M	273
概要.....	273
ポートの説明	274
デザインの入力方法.....	274
使用可能な属性.....	275
詳細情報	276
RAM32X1D	277
概要.....	277
論理表.....	277
デザインの入力方法.....	277
使用可能な属性.....	278
詳細情報	279
RAM32X1S_1	280
概要.....	280
論理表.....	280
デザインの入力方法.....	280
使用可能な属性.....	281
詳細情報	281
RAM32X2S	282
概要.....	282
論理表.....	282
デザインの入力方法.....	282
使用可能な属性.....	283
詳細情報	284
RAM32X4S	285
概要.....	285
論理表.....	285
デザインの入力方法.....	285
使用可能な属性.....	286
詳細情報	287
RAM32X8S	288
概要.....	288
論理表.....	288
デザインの入力方法.....	288
使用可能な属性.....	289
詳細情報	290
RAM64M	291
概要.....	291
ポートの説明	292
デザインの入力方法.....	292
使用可能な属性.....	293

詳細情報	294
RAM64X1D	295
概要	295
論理表	295
デザインの入力方法	296
使用可能な属性	296
詳細情報	297
RAM64X1S	298
概要	298
論理表	298
デザインの入力方法	298
使用可能な属性	298
詳細情報	299
RAM64X1S_1	300
概要	300
論理表	300
デザインの入力方法	300
使用可能な属性	300
詳細情報	301
RAM64X2S	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	304
RAMB18	305
概要	305
ポートの説明	305
デザインの入力方法	306
使用可能な属性	306
詳細情報	311
RAMB18SDP	312
概要	312
ポートの説明	312
デザインの入力方法	313
使用可能な属性	313
詳細情報	317
RAMB36	318
概要	318
ポートの説明	319
デザインの入力方法	320
使用可能な属性	320
詳細情報	328
RAMB36SDP	329
概要	329
ポートの説明	329
デザインの入力方法	330
使用可能な属性	330
詳細情報	336
SRL16	337
概要	337
論理表	337
デザインの入力方法	337
使用可能な属性	338
詳細情報	338
SRL16_1	339
概要	339
論理表	339
デザインの入力方法	339

使用可能な属性.....	340
詳細情報	340
SRL16E	341
概要.....	341
論理表.....	341
ポートの説明	342
デザインの入力方法.....	342
使用可能な属性.....	342
詳細情報	343
SRL16E_1.....	344
概要.....	344
論理表.....	344
デザインの入力方法.....	344
使用可能な属性.....	345
詳細情報	345
SRLC32E	346
概要.....	346
ポートの説明	346
デザインの入力方法.....	347
使用可能な属性.....	347
詳細情報	348
STARTUP_VIRTEX5	349
概要.....	349
ポートの説明	349
デザインの入力方法.....	350
詳細情報	351
SYSMON	352
概要.....	352
ポートの説明	352
デザインの入力方法.....	353
使用可能な属性.....	353
詳細情報	356
TEMAC	357
概要.....	357
デザインの入力方法.....	357
詳細情報	357
USR_ACCESS_VIRTEX5	358
概要.....	358
ポートの説明	358
デザインの入力方法.....	358
詳細情報	359

このマニュアルについて

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

このマニュアルには、次の内容が含まれます。

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

デザイン エレメント

このバージョンのライブラリ ガイドでは、このアーキテクチャのザイリンクス ユニファイド ライブラリを構成するデザイン エレメントのカテゴリと、インスタンスエーション コード例が含まれます。次がそのカテゴリです。

- ・ **ターゲット デバイス変更後のエレメント**：このアーキテクチャで使用すると自動的に ISE ソフトウェア ツールにより変換される既存のデザイン エレメントです。ターゲット デバイスを変更することにより最新の回路設計技術を最大限に利用できるようになります。
- ・ **マクロ**：これらのエレメ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エーションしにくいようなプリミティブをインスタンスエーション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。
- ・ **プリミティブ**：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エーションして変換 (NGDBuild) プロセスを実行すると、変換後のファイルに含まれるのはまったく同じコンポーネントです。たとえば、ISERDES_NODELAY という Virtex-5 エレメントをユーザー プリミティブとしてインスタンスエーションし、変換 (NGDBuild) を実行すると、ISERDES_NODELAY がそのまま残ります。一方 Virtex-5 デバイスで ISERDES を使用していると、自動的に Virtex-5 用の ISERDES_NODELAY に変換されます。このため、「プリミティブ」の概念は、同じ分野でもユーザーによって異なります。

ザイリンクスでは、さまざまなデバイス アーキテクチャに対応した多数のデザイン エレメント (ユニマクロおよびプリミティブ) を含むソフトウェア ライブラリを提供しています。開発システム ソフトウェアのリリースごとに、新しいデザイン エレメントが組み込まれます。すべてのデザイン エレメントを含むユニファイド ライブラリに対し、このガイドにはアーキテクチャ固有のライブラリのみが含まれています。

デザインの入力方法

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

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

第 2 章

ターゲット デバイス変更後のデザイン エレメント

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

次の表に、そのエレメントと変更後のアドバンス エレメントを示します。

元のデザイン エレメント	変更後のエレメント
BUFGCE_1	BUFGCE + INV
BUFGMUX	BUFGMUX_CTRL
BUFGMUX_1	BUFGMUX_CTRL + INV
BUFGMUX_VIRTEX4	BUFGMUX_CTRL
BUFGP	BUFG
DCM_BASE	DCM_ADV
DCM_PS	DCM_ADV
DSP48	DSP48E
FD	FDCPE
FD_1	FDCPE + INV
FDC	FDCPE
FDC_1	FDCPE + INV
FDCE	FDCPE
FDCE_1	FDCPE + INV
FDCP	FDCPE
FDCP_1	FDCPE + INV
FDE	FDCPE
FDE_1	FDCPE + INV
FDPE	FDCPE
FDPE_1	FDCPE + INV
FDR	FDRSE
FDR_1	FDRSE + INV
FDRE	FDRSE
FDRE_1	FDRSE + INV

元のデザイン エLEMENT	変更後のELEMENT
FDRS	FDRSE
FDRS_1	FDRSE + INV
FDS	FDRSE
FDS_1	FDRSE + INV
FDSE	FDRSE
FDSE_1	FDRSE + INV
FIFO16	FIFO18
ISERDES	ISERDES_NODELAY
JTAGPPC	JTAG_PPC440
LD	LDCPE
LD_1	LDCPE + INV
LDC	LDCPE
LDC_1	LDCPE + INV
LDCE	LDCPE
LDCE_1	LDCPE + INV
LDCP	LDCPE
LDCP_1	LDCPE + INV
LDE	LDCPE
LDE_1	LDCPE + INV
LDP	LDCPE
LDP_1	LDCPE + INV
LDPE	LDCPE
LDPE_1	LDCPE + INV
LUT1	LUT5
LUT1_L	LUT5_L
LUT1_D	LUT5_D
LUT2	LUT5
LUT2_L	LUT5_L
LUT2_D	LUT5_D
LUT3	LUT5
LUT3_L	LUT5_L
LUT3_D	LUT56_D
LUT4	LUT5
LUT4_L	LUT5_L
LUT4_D	LUT5_D
MULT_AND	LUT6
MULT18X18	DSP48E
MULT18X18S	DSP48E
MUXCY	CARRY4

元のデザイン エLEMENT	変更後のELEMENT
MUXCY_D	CARRY4
MUXCY_L	CARRY4
MUXF5	LUT5
MUXF5_D	LUT5_D
MUXF5_L	LUT5_L
MUXF6	LUT6
MUXF6_D	LUT6_D
MUXF6_L	LUT6_L
PMCD	PLL_ADV
RAM16X1D	RAM64X1D
RAM16X1S	RAM64X1S
RAM32X1S	RAM64X1S
RAMB16	RAMB18
RAMB16BWE	RAMB18
ROM128X1	6 入力 LUT X 2 + MUXF7
ROM16X1	LUT5
ROM256X1	6 入力 LUT X 4 + MUXF6/7
ROM32X1	LUT5
ROM64X1	LUT6
SRLC16	SRLC32E
SRLC16_1	SRLC32E + INV
SRLC16E	SRLC32E
SRLC16E_1	SRLC32E + INV
XORCY	CARRY4
XORCY_D	CARRY4
XORCY_L	CARRY4

第 3 章

UniMacro について

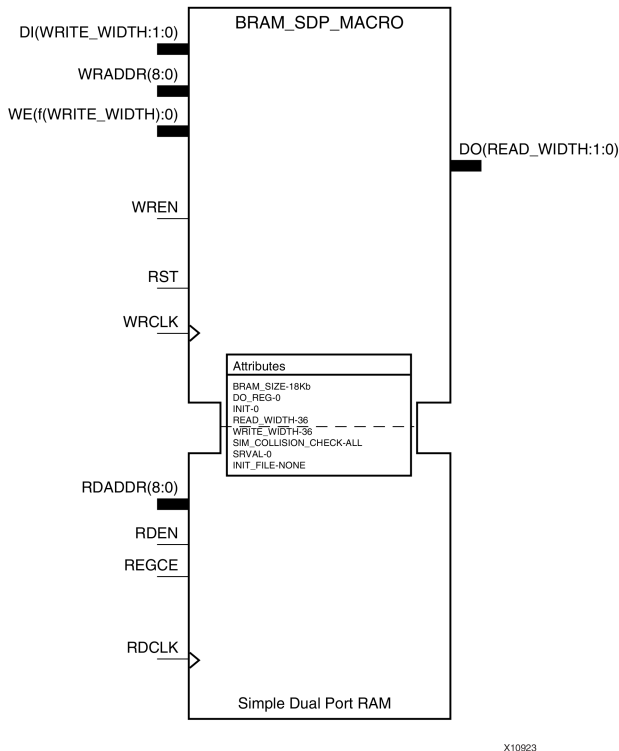
このセクションでは、このアーキテクチャで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

各 UniMacro について、次の情報が提供されています。

- ・ 名称
- ・ 説明
- ・ 回路図シンボル
- ・ 論理表 (該当するエレメントでのみ)
- ・ ポートの説明
- ・ デザインの入力方法
- ・ 使用可能な属性
- ・ インスタンス化コードの例
- ・ その他のリソース

BRAM_SDP_MACRO

マクロ : Simple Dual Port RAM



X10923

概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、READ と WRITE は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

メモ： このエレメントは、読み出しおよび書き込みのポートの幅が同じになるようにコンフィギュレーションする必要があります。

ポートの説明

属性	方向	幅	機能
出力ポート			
DO	出力	コンフィギュレーション表を参照	RDADDR で指定されたデータ出力バス
入力ポート			
DI	入力	コンフィギュレーション表を参照	WRADDR で指定されたデータ入力バス
WRADDR、RDADDR	入力	コンフィギュレーション表を参照	書き込み/読み出しアドレス入力バス
WE	入力	コンフィギュレーション表を参照	バイト幅ライト イネーブル
WREN、RDEN	入力	1	書き込み/読み出し イネーブル

属性	方向	幅	機能
SSR	入力	1	出力レジスタの同期リセット
REGCE	入力	1	出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
WRCLK、RDCLK	入力	1	書き込み/読み出しクロック入力

コンフィギュレーション表

DATA_WIDTH	BRAM_SIZE	ADDR	WE
72 ~ 37	36kb	9	8
36 ~ 19	36kb	10	4
	18kb	9	
18 ~ 10	36kb	11	2
	18kb	10	
9 ~ 5	36kb	12	1
	18kb	11	
4 ~ 3	36kb	13	1
	18kb	12	
2	36kb	14	1
	18kb	13	
1	36kb	15	1
	18kb	14	

デザインの入力方法

この UniMacro はインスタンスエーションのみが可能で、プリミティブにパラメータを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
BRAM_SIZE	文字列	18kb、36kb	18kb	RAM を 18kb または 36kb メモリとしてコンフィギュレーションします。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロックサイクルで読み出しが可能です。clock-to-out タイムが長くなります。

属性	タイプ	値	デフォルト	説明
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
READ_WIDTH、 WRITE_WIDTH	整数	1 ~ 72	36	DI/DO バスの幅を指定。READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
INIT_FILE	文字列	0 ビット STRING	NONE	初期値を含むファイルの名前
SIM_COLLISION_ CHECK	文字列	ALL、WARNING_ ONLY、 GENERATE_X_ ONLY、 NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。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>
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーションモデルがパフォーマンス重視モードで実行されます。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。
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_7F	16 進数	256 ビット値	すべてゼロ	16kb または 32kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべてゼロ	2kb または 4kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

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

```
library UNIMACRO;
```



```
use unimacro.Vcomponents.all;

-- BRAM_SDP_MACRO: Simple Dual Port RAM
--                Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BRAM_SDP_MACRO_inst : BRAM_SDP_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "VIRTEX5" -- Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  WRITE_WIDTH => 0,    -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH => 0,     -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  DO_REG => 0, -- Optional output register (0 or 1)
  INIT_FILE => "NONE",
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
                                -- "GENERATE_X_ONLY" or "NONE"
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
                                -- see "Synthesis and Simulation Design Guide" for details
  SRVAL => X"00000000000000000000", -- Set/Reset value for port output
  INIT => X"00000000000000000000", -- Initial values on output port
  -- 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",
```

[illegible]

```

INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => 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",

-- The next set of INIT_xx are valid when configured as 36Kb
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
DO => DO,          -- Output read data port
DI => DI,          -- Input write data port
RDADDR => RDADDR,  -- Input read address
RDCLK => RDCLK,    -- Input read clock
RDEN => RDEN,      -- Input read port enable
REGCE => REGCE,    -- Input read output register enable
RST => RST,        -- Input reset
WE => WE,          -- Input write enable
WRADDR => WRADDR,  -- Input write address
WRCLK => WRCLK,    -- Input write clock
WREN => WREN       -- Input write port enable
);
-- End of BRAM_SDP_MACRO_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// BRAM_SDP_MACRO: Simple Dual Port RAM
//               Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

BRAM_SDP_MACRO #(
    .BRAM_SIZE("18Kb"), // Target BRAM, "18Kb" or "36Kb"
    .DEVICE("VIRTEX5"), // Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .WRITE_WIDTH(0),    // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
    .READ_WIDTH(0),     // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
    .DO_REG(0),         // Optional output register (0 or 1)
    .INIT_FILE("NONE"),
    .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
                                // "GENERATE_X_ONLY" or "NONE"
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .SRVAL(72'h00000000000000000000), // Set/Reset value for port output
    .INIT(72'h00000000000000000000), // Initial values on output port
    .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

[illegible]

```
// The next set of INIT_xx are valid when configured as 36Kb
```

[illegible]

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

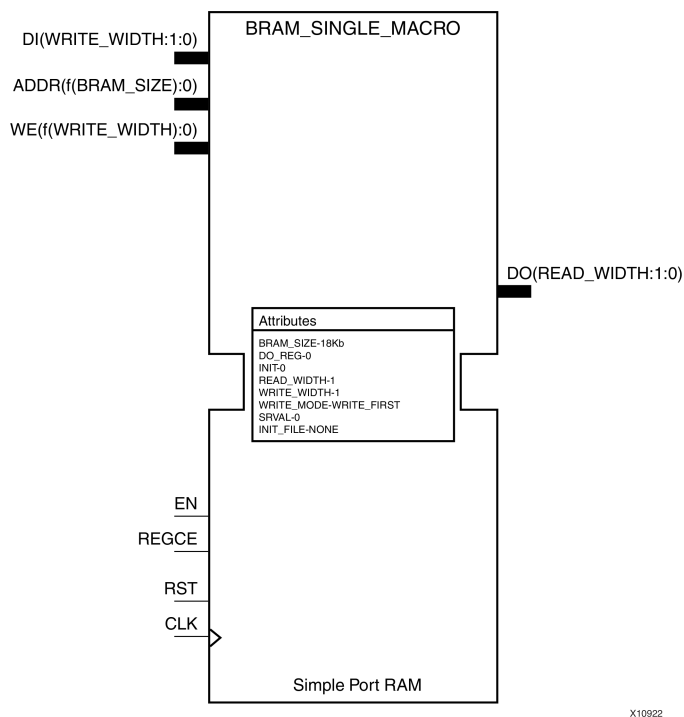
```
.WRADDR(WRADDR), // Input write address  
.WRCLK(WRCLK),   // Input write clock  
.WREN(WREN)      // Input write port enable  
);  
  
// End of BRAM_SDP_MACRO_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート：DC 特性およびスイッチ特性](#)

BRAM_SINGLE_MACRO

マクロ : Single Port RAM



概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのシングルポートのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。バイトイネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

ポートの説明

属性	方向	幅	機能
出力ポート			
DO	出力	コンフィギュレーション表を参照	ADDR で指定されたデータ出力バス
入力ポート			
DI	入力	コンフィギュレーション表を参照	ADDR で指定されたデータ入力バス
ADDR	入力	コンフィギュレーション表を参照	アドレス入力バス
WE	入力	コンフィギュレーション表を参照	バイト幅ライト イネーブル
EN	入力	1	書き込み/読み出しイネーブル
RST	入力	1	出力レジスタの同期リセット
REGCE	入力	1	出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
CLK	入力	1	クロック入力

コンフィギュレーション表

WRITE_WIDTH	READ_WIDTH	BRAM_SIZE	ADDR	WE
36 ~ 19	36 ~ 19	36kb	10	4
	18 ~ 10		11	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
18 ~ 10	36 ~ 19	36kb	11	2
	18 ~ 10		11	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
9 ~ 5	36 ~ 19	36kb	12	1
	18 ~ 10		12	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
4 ~ 3	36 ~ 19	36kb	13	1
	18 ~ 10		13	
	9 ~ 5		13	
	4 ~ 3		13	
	2		14	
	1		15	
2	36 ~ 19	36kb	14	1
	18 ~ 10		14	
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		15	

WRITE_WIDTH	READ_WIDTH	BRAM_SIZE	ADDR	WE
1	36 ~ 19	36kb	15	1
	18 ~ 10		15	
	9 ~ 5		15	
	3 ~ 4		15	
	2		15	
	1		15	
18 ~ 10	18 ~ 10	18kb	10	2
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
9 ~ 5	18 ~ 10	18kb	11	1
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
4 ~ 3	18 ~ 10	18kb	12	1
	9 ~ 5		12	
	4 ~ 3		12	
	2		13	
	1		14	
2	18 ~ 10	18kb	13	1
	9 ~ 5		13	
	4 ~ 3		13	
	2		13	
	1		14	
1	18 ~ 10	18kb	14	1
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		14	

デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
BRAM_SIZE	文字列	18kb、36kb	18kb	RAM を 18kb または 36kb メモリとしてコンフィギュレーションします。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
READ_WIDTH	整数	1 ～ 36	1	出力バスの幅を指定
WRITE_WIDTH	整数	1 ～ 36	1	入力バスの幅を指定
INIT_FILE	文字列	0 ビット STRING	NONE	初期値を含むファイルの名前
WRITE_MODE	文字列	READ_FIRST、WRITE_FIRST、NO_CHANGE	WRITE_FIRST	メモリへの書き込みモードを指定
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
INIT_00 ～ INIT_FF	16 進数	256 ビット値	すべてゼロ	16kb または 32kb のデータ メモリ アレイの初期値を指定
INITP_00 ～ INITP_0F	16 進数	256 ビット値	すべてゼロ	2kb または 4kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンス化)

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 UNIMACRO;
use unimacro.vcomponents.all;

-- BRAM_SINGLE_MACRO: Single Port RAM
--                      Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2
```

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

```
-- 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",

-- The next set of INIT_xx are valid when configured as 36Kb
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")

port map (
  DO => DO,      -- Output data
  ADDR => ADDR,   -- Input address
  CLK => CLK,     -- Input clock
  DI => DI,       -- Input data port
  EN => EN,       -- Input RAM enable
  REGCE => REGCE, -- Input output register enable
  RST => RST,     -- Input reset
  WE => WE        -- Input write enable
);

-- End of BRAM_SINGLE_MACRO_inst instantiation
```

Verilog 記述 (インスタネーション)

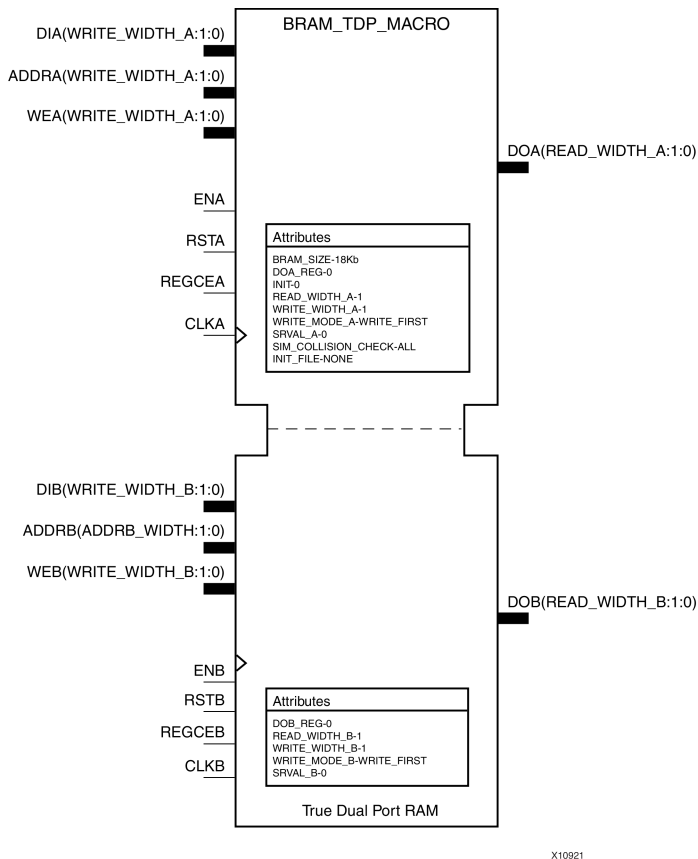
```
// BRAM_SINGLE_MACRO: Single Port RAM
//                               Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

BRAM_SINGLE_MACRO #(
  .BRAM_SIZE("18Kb"), // Target BRAM, "18Kb" or "36Kb"
  .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  .DO_REG(0), // Optional output register (0 or 1)
  .INIT(36'h00000000), // Initial values on output port
  .INIT_FILE("NONE"),
  .WRITE_WIDTH(0), // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  .READ_WIDTH(0), // Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
  .SRVAL(36'h00000000), // Set/Reset value for port output
  .WRITE_MODE("WRITE_FIRST"), // "WRITE_FIRST", "READ_FIRST", or "NO_CHANGE"
  .INIT_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_07(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_08(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_09(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0A(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0B(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0C(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0D(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0E(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_0F(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_10(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
```

[illegible]

BRAM_TDP_MACRO

マクロ : True Dual Port RAM



概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、READ と WRITE は互いに独立しており、同じメモリ アレイにアクセスする間は非同期になります。バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

ポートの説明

属性	方向	幅	機能
出力ポート			
DOA	出力	コンフィギュレーション表を参照	ADDRA で指定されたデータ出力バス
DOB	出力	コンフィギュレーション表を参照	ADDRB で指定されたデータ出力バス
入力ポート			
DIA	入力	コンフィギュレーション表を参照	ADDRA で指定されたデータ入力バス

属性	方向	幅	機能
DIB	入力	コンフィギュレーション表を参照	ADDRB で指定されたデータ入力バス
ADDRA, ADDRb	入力	コンフィギュレーション表を参照	ポート A およびポート B のアドレス入力バス
WEA, WEB	入力	コンフィギュレーション表を参照	ポート A およびポート B のライトイネーブル
ENA, ENB	入力	1	ポート A およびポート B のライト/リード イネーブル
RSTA, RSTB	入力	1	ポート A およびポート B の出力レジスタの同期リセット
REGCEA, REGCEB	入力	1	ポート A および B の出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
CLKA, CLKB	入力	1	ポート A および B の書き込み/読み出しクロック入力

コンフィギュレーション表

WRITE_WIDTH_A/B-DIA/DIB	READ_WIDTH_A/B-DOA/DOB	BRAM_SIZE	ADDRA/B	WEA/B
36 ~ 19	36 ~ 19	36Kb	10	4
	18 ~ 10		11	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
18 ~ 10	36 ~ 19	36Kb	11	2
	18 ~ 10		11	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
9 ~ 5	36 ~ 19	36Kb	12	1
	18 ~ 10		12	
	9 ~ 5		12	
	4 ~ 3		13	
	2		14	
	1		15	
4 ~ 3	36 ~ 19	36Kb	13	1
	18 ~ 10		13	
	9 ~ 5		13	
	4 ~ 3		13	
	2		14	
	1		15	

WRITE_WIDTH_A/B-DIA/DIB	READ_WIDTH_A/B-DOA/DOB	BRAM_SIZE	ADDRA/B	WEA/B
2	36 ~ 19	36Kb	14	1
	18 ~ 10		14	
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		15	
1	36 ~ 19	36Kb	15	1
	18 ~ 10		15	
	9 ~ 5		15	
	4 ~ 3		15	
	2		15	
	1		15	
18 ~ 10	18 ~ 10	18Kb	10	2
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
9 ~ 5	18 ~ 10	18Kb	11	1
	9 ~ 5		11	
	4 ~ 3		12	
	2		13	
	1		14	
4 ~ 3	18 ~ 10	18Kb	12	1
	9 ~ 5		12	
	4 ~ 3		12	
	2		13	
	1		14	
2	18 ~ 10	18Kb	13	1
	9 ~ 5		13	
	4 ~ 3		13	
	2		13	
	1		14	
1	18 ~ 10	18Kb	14	1
	9 ~ 5		14	
	4 ~ 3		14	
	2		14	
	1		14	
1	1	カスケード接続	16	1

デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
BRAM_SIZE	文字列	18Kb、36Kb	18Kb	RAM を 18Kb または 36Kb メモリとしてコンフィギュレーションします。
DO_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロックサイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
INIT_FILE	文字列	0 ビット STRING	NONE	初期値を含むファイルの名前
READ_WIDTH、WRITE_WIDTH	整数	1 ～72	36	DI/DO バスの幅を指定。READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
SIM_COLLISION_CHECK	文字列	ALL、WARNING_ONLY、GENERATE_X_ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>

属性	タイプ	値	デフォルト	説明
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
SRVAL_A、SRVAL_B	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_FF	16 進数	256 ビット値	すべてゼロ	16Kb または 32Kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべてゼロ	2Kb または 4Kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

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

```
library UNIMACRO;
use unimacro.Vcomponents.all;

-- BRAM_TDP_MACRO: True Dual Port RAM
--                Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "VIRTEX5", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  DOA_REG => 0, -- Optional port A output register (0 or 1)
  DOB_REG => 0, -- Optional port B output register (0 or 1)
  INIT_A => X"000000000", -- Initial values on A output port
  INIT_B => X"000000000", -- Initial values on B output port
  INIT_FILE => "NONE",
  READ_WIDTH_A => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH_B => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
  -- "GENERATE_X_ONLY" or "NONE"
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
  -- see "Synthesis and Simulation Design Guide" for details
  SRVAL_A => X"000000000", -- Set/Reset value for A port output
  SRVAL_B => X"000000000", -- 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 => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
  WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
  -- 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",
```

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

46 <http://japan.xilinx.com> Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

Verilog 記述 (インスタンス化)

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

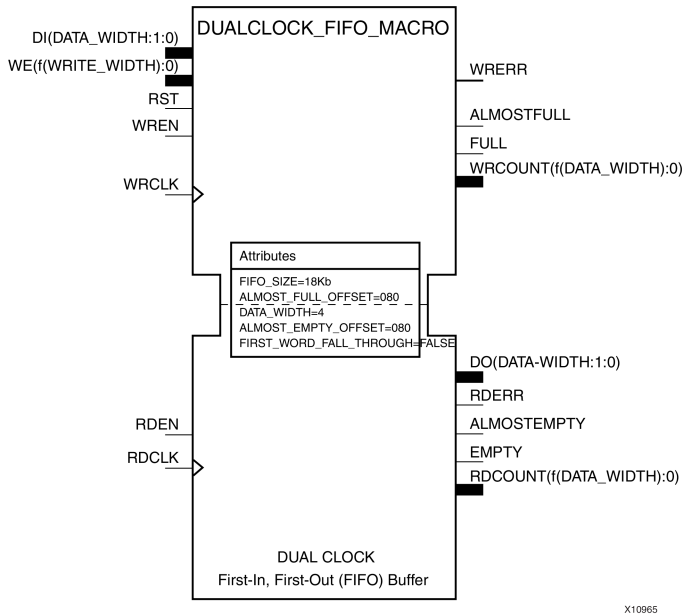
[illegible]

詳細情報

- ・ Virtex-5 FPGA ユーザー ガイド
- ・ Virtex-5 FPGA データシート：DC 特性およびスイッチ特性

FIFO_DUALCLOCK_MACRO

マクロ : Dual Clock First-In, First-Out (FIFO) RAM Buffer



概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。ブロック RAM に含まれる専用ロジックで FIFO を簡単にインプリメントできます。FIFO は 18kb または 36kb メモリとしてコンフィギュレーションできます。この UniMacro を使用すると、読み出しと書き込みに独立したクロックを使用するように FIFO がコンフィギュレーションされます。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。

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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 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。

ポートの説明

ポート名	方向	幅	機能
出力ポート			
ALMOSTEMPTY	出力	1	FIFO のほぼすべての有効エントリが読み出されています。
ALMOSTFULL	出力	1	FIFO メモリのほぼすべてのエントリがフル
DO	出力	コンフィギュレーション表を参照	ADDR で指定されたデータ出力バス
EMPTY	出力	1	FIFO が空です。
FULL	出力	1	FIFO メモリのすべてのエントリがフル
RDCOUNT	出力	コンフィギュレーション表を参照	FIFO データ読み出しポインタ

ポート名	方向	幅	機能
RDERR	出力	1	FIFO が空のときに読み出しを行うと、アサートされます。
WRCOUNT	出力	コンフィギュレーション表を参照	FIFO データ書き込みポインタ
WRERR	出力	1	FIFO がフルのときに書き込みを行うと、アサートされます。
入力ポート			
DI	入力	コンフィギュレーション表を参照	ADDR で指定されたデータ入力バス
RDCLK	入力	1	読み出しクロック
RDEN	入力	1	読み出しイネーブル
RST	入力	1	同期リセット
WRCLK	入力	1	書き込みクロック
WREN	入力	1	ライト イネーブル

コンフィギュレーション表

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。コンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

DATA_WIDTH	FIFO_SIZE	WRCOUNT	RDCOUNT
72 ～ 37	36kb	9	9
36 ～ 19	36kb	10	10
	18kb	9	9
18 ～ 10	36kb	11	11
	18kb	10	10
9 ～ 5	36kb	12	12
	18kb	11	11
1 ～ 4	36kb	13	13
	18kb	12	12

デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

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

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	13 ビット値	すべてゼロ	EMPTY と ALMOSTEMPTY の差を設定します。16 進数で指定します。
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべてゼロ	FULL と ALMOSTFULL の差を設定します。16 進数で指定します。
DATA_WIDTH	整数	1 ~ 72	4	DI/DO バスの幅
FIFO_SIZE	文字列	18kb、36kb	18kb	FIFO を 18kb または 36kb メモリとしてコンフィギュレーションします。
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	FALSE	TRUE に設定すると、RDEN をアサートしなくても、空の FIFO に書き込まれた最初のワードが出力されます。
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。

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 UNIMACRO;
use unimacro.vcomponents.all;

-- FIFO_DUALCLOCK_MACRO: Dual-Clock First-In, First-Out (FIFO) RAM Buffer
--                               Virtex-5, Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

FIFO_DUALCLOCK_MACRO_inst : FIFO_DUALCLOCK_MACRO
generic map (
    DEVICE => "VIRTEX5",           -- Target Device: "VIRTEX5", "VIRTEX6"
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 0,              -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    FIFO_SIZE => "18Kb",           -- Target BRAM, "18Kb" or "36Kb"
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation "SAFE" vs "FAST",
                        -- see "Synthesis and Simulation Design Guide" for details

port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,   -- Output almost full
    DO => DO,                   -- Output data
    EMPTY => EMPTY,             -- Output empty
    FULL => FULL,               -- Output full
    RDCOUNT => RDCOUNT,         -- Output read count
    RDERR => RDERR,             -- Output read error
    WRCOUNT => WRCOUNT,         -- Output write count
    WRERR => WRERR,             -- Output write error
    DI => DI,                   -- Input data
    RDCLK => RDCLK,             -- Input read clock
    RDEN => RDEN,               -- Input read enable
    RST => RST,                 -- Input reset
    WRCLK => WRCLK,             -- Input write clock
    WREN => WREN                -- Input write enable
);
-- End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// FIFO_DUALCLOCK_MACRO: Dual Clock First-In, First-Out (FIFO) RAM Buffer
//                               Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

FIFO_DUALCLOCK_MACRO #(
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(9'h080),  // Sets almost full threshold
    .DATA_WIDTH(0),               // Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    .DEVICE("VIRTEX5"),           // Target device: "VIRTEX5", "VIRTEX6"
    .FIFO_SIZE ("18Kb"),           // Target BRAM: "18Kb" or "36Kb"
    .FIRST_WORD_FALL_THROUGH (FALSE), // Sets the FIFO FWFT to "TRUE" or "FALSE"
    .SIM_MODE("SAFE") // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
) FIFO_DUALCLOCK_MACRO (
    .ALMOSTEMPTY(ALMOSTEMPTY), // Output almost empty
    .ALMOSTFULL(ALMOSTFULL),   // Output almost full
    .DO(DO),                   // Output data
    .EMPTY(EMPTY),             // Output empty
    .FULL(FULL),               // Output full
    .RDCOUNT(RDCOUNT),         // Output read count
    .RDERR(RDERR),             // Output read error
    .WRCOUNT(WRCOUNT),         // Output write count
    .WRERR(WRERR),             // Output write error
    .DI(DI),                   // Input data
    .RDCLK(RDCLK),             // Input read clock
    .RDEN(RDEN),               // Input read enable
    .RST(RST),                 // Input reset
    .WRCLK(WRCLK),             // Input write clock
    .WREN(WREN)                // Input write enable
);

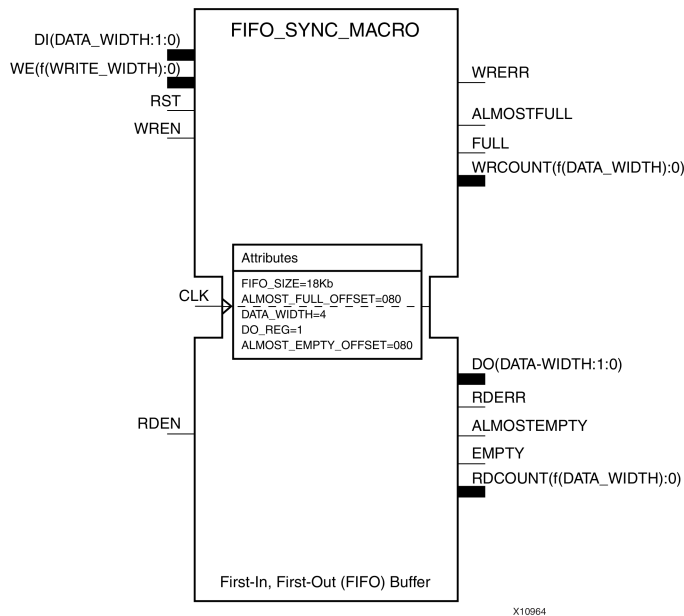
// End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

FIFO_SYNC_MACRO

マクロ : Synchronous First-In, First-Out (FIFO) RAM Buffer



概要

FPGA デバイスにはブロック RAM が数個含まれ、汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。ブロック RAM に含まれる専用ロジックで FIFO を簡単にインプリメントできます。FIFO は 18kb または 36kb メモリとしてコンフィギュレーションできます。この UniMacro を使用すると、読み出しおよび書き込みにクロックを使用するように FIFO がコンフィギュレーションされます。

ポートの説明

ポート名	方向	幅	機能
出力ポート			
ALMOSTEMPTY	出力	1	FIFO のほぼすべての有効エントリが読み出されています。
ALMOSTFULL	出力	1	FIFO メモリのほぼすべてのエントリがフル
DO	出力	コンフィギュレーション表を参照	ADDR で指定されたデータ出力バス
EMPTY	出力	1	FIFO が空です。
FULL	出力	1	FIFO メモリのすべてのエントリがフル
RDCOUNT	出力	コンフィギュレーション表を参照	FIFO データ読み出しポインタ
RDERR	出力	1	FIFO が空のときに読み出しを行うと、アサートされます。
WRCOUNT	出力	コンフィギュレーション表を参照	FIFO データ書き込みポインタ
WRERR	出力	1	FIFO がフルのときに書き込みを行うと、アサートされます。

ポート名	方向	幅	機能
入力ポート			
CLK	入力	1	読み出し/書き込みクロック
DI	入力	コンフィギュレーション表を参照	ADDR で指定されたデータ入力バス
RDEN	入力	1	読み出しイネーブル
RST	入力	1	同期リセット
WREN	入力	1	ライト イネーブル

コンフィギュレーション表

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。コンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

DATA_WIDTH	FIFO_SIZE	WRCOUNT	RDCOUNT
72 ~ 37	36kb	9	9
36 ~ 19	36kb	10	10
	18kb	9	9
18 ~ 10	36kb	11	11
	18kb	10	10
9 ~ 5	36kb	12	12
	18kb	11	11
1 ~ 4	36kb	13	13
	18kb	12	12

デザインの入力方法

この UniMacro はインスタンス化のみが可能で、プリミティブにパラメータを設定できるようにしたものです。上記のコンフィギュレーション表を参照し、デザイン要件を満たすように設定してください。

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

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	13 ビット値	すべてゼロ	EMPTY と ALMOSTEMPTY の差を設定します。16 進数で指定します。
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべてゼロ	FULL と ALMOSTFULL の差を設定します。16 進数で指定します。
DATA_WIDTH	整数	1 ~ 72	4	DI/DO バスの幅
DO_REG	2 進数	0、1	1	同期 FIFO では、標準の動作を実行するには 0 に設定します。 1 に設定した場合、同期 FIFO の出力にパイプラインレジスタが追加されます。このためデータに 1 クロックサイクルのレイテンシが発生しますが、clock-to-out タイミングは改善します。
FIFO_SIZE	文字列	18kb、36kb	18kb	FIFO を 18kb または 36kb メモリとしてコンフィギュレーションします。
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。

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 UNIMACRO;
use unimacro.Vcomponents.all;

-- FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIFO) RAM Buffer
--                      Virtex-5, Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

FIFO_SYNC_MACRO_inst : FIFO_SYNC_MACRO
generic map (
    DEVICE => "VIRTEX5",           -- Target Device: "VIRTEX5", "VIRTEX6"
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 0,              -- Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    FIFO_SIZE => "18Kb",           -- Target BRAM, "18Kb" or "36Kb"
    SIM_MODE => "SAFE") -- Simulation) "SAFE" vs "FAST",
                        -- see "Synthesis and Simulation Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY,    -- Output almost empty
    ALMOSTFULL => ALMOSTFULL,      -- Output almost full
    DO => DO,                      -- Output data
    EMPTY => EMPTY,               -- Output empty
    FULL => FULL,                 -- Output full
    RDCOUNT => RDCOUNT,           -- Output read count
    RDERR => RDERR,              -- Output read error
    WRCOUNT => WRCOUNT,          -- Output write count
    WRERR => WRERR,              -- Output write error
    CLK => CLK,                  -- Input clock
    DI => DI,                    -- Input data
    RDEN => RDEN,                -- Input read enable
    RST => RST,                  -- Input reset
    WREN => WREN,                -- Input write enable
);
-- End of FIFO_SYNC_MACRO_inst instantiation
```


Verilog 記述 (インスタンスエーション)

```
// FIFO_SYNC_MACRO: Synchronous First-In, First-Out (FIFO) RAM Buffer
//                               Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

FIFO_SYNC_MACRO #(
    .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5"
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(9'h080), // Sets almost full threshold
    .DATA_WIDTH(0), // Valid values are 1-72 (37-72 only valid when FIFO_SIZE="36Kb")
    .DEVICE("VIRTEX5"), // Target device: "VIRTEX5", "VIRTEX6"
    .DO_REG(0), // Optional output register (0 or 1)
    .FIFO_SIZE ("18Kb"), // Target BRAM: "18Kb" or "36Kb"
    .SIM_MODE("SAFE") // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
) FIFO_SYNC_MACRO_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // Output almost empty
    .ALMOSTFULL(ALMOSTFULL), // Output almost full
    .DO(DO), // Output data
    .EMPTY(EMPTY), // Output empty
    .FULL(FULL), // Output full
    .RDCOUNT(RDCOUNT), // Output read count
    .RDERR(RDERR), // Output read error
    .WRCOUNT(WRCOUNT), // Output write count
    .WRERR(WRERR), // Output write error
    .CLK(CLK), // Input clock
    .DI(DI), // Input data
    .RDEN(RDEN), // Input read enable
    .RST(RST), // Input reset
    .WREN(WREN) // Input write enable
);

// End of FIFO_SYNC_MACRO_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

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
CRC32	プリミティブ : Cyclic Redundancy Check Calculator for 32 bits
CRC64	プリミティブ : Cyclic Redundancy Check Calculator for 64 bits
SYSMON	プリミティブ : System Monitor
TEMAC	プリミティブ : Tri-mode Ethernet Media Access Controller (MAC)

四則演算ファンクション

デザイン エLEMENT	説明
DSP48E	プリミティブ : 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtractor/Accumulator or 2-Input Logic Unit

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
BUFGCTRL	プリミティブ : Global Clock MUX Buffer
BUFGMUX_CTRL	コンビニエンス プリミティブ : 2-to-1 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
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
PLL_ADV	プリミティブ : Advanced Phase Locked Loop Clock Circuit
PLL_BASE	プリミティブ : Basic Phase Locked Loop Clock Circuit

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_VIRTEX5	プリミティブ : Virtex®-5 JTAG Boundary-Scan Logic Access Circuit
CAPTURE_VIRTEX5	プリミティブ : Virtex®-5 Readback Register Capture Control
FRAME_ECC_VIRTEX5	プリミティブ : Virtex®-5 Configuration Frame Error Detection and Correction Circuitry
ICAP_VIRTEX5	プリミティブ : Internal Configuration Access Port
KEY_CLEAR	プリミティブ : Virtex-5 Configuration Encryption Key Erase
STARTUP_VIRTEX5	プリミティブ : Virtex®-5 Configuration Start-Up Sequence Interface
USR_ACCESS_VIRTEX5	プリミティブ : Virtex-5 User Access Register

ギガビット I/O

デザイン エLEMENT	説明
GTP_DUAL	プリミティブ : Dual Gigabit Transceiver
GTX_DUAL	プリミティブ : Dual Gigabit Transceiver

I/O コンポーネント

デザイン エLEMENT	説明
DCIRESET	プリミティブ : DCI State Machine Reset (After Configuration Has Been Completed)
IBUF	プリミティブ : Input Buffer
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
IOBUFGDS	プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable
IODELAY	プリミティブ : Input and Output Fixed or Variable Delay Element
ISERDES_NODELAY	プリミティブ : Input SERIAL/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
PPC440	プリミティブ : Power PC 440 CPU Core

RAM/ROM

デザイン エLEMENT	説明
FIFO18	プリミティブ : 18kb FIFO (First In, First Out) Block RAM Memory
FIFO18_36	プリミティブ : 36-bit Wide by 512 Deep 18kb FIFO (First In, First Out) Block RAM Memory
FIFO36	プリミティブ : 36kb FIFO (First In, First Out) Block RAM Memory
FIFO36_72	プリミティブ : 72-Bit Wide by 512 Deep 36kb FIFO (First In, First Out) Block RAM Memory with ECC (Error Detection and Correction Circuitry)
RAM128X1D	プリミティブ : 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)
RAM16X1D_1	プリミティブ : 16-Deep by 1-Wide Static Dual Port Synchronous RAM with Negative-Edge Clock
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
RAM256X1S	プリミティブ : 256-Deep by 1-Wide Random Access Memory (Select RAM)
RAM32M	プリミティブ : 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)
RAM32X1D	プリミティブ : 32-Deep by 1-Wide Static Dual Port 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
RAM64M	プリミティブ : 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)
RAM64X1D	プリミティブ : 64-Deep by 1-Wide Dual Port 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
RAMB18	プリミティブ : 18K-bit Configurable Synchronous True Dual Port Block RAM
RAMB18SDP	プリミティブ : 36-bit by 512 Deep, 18kb Synchronous Simple Dual Port Block RAM
RAMB36	プリミティブ : 36kb Configurable Synchronous True Dual Port Block RAM
RAMB36SDP	プリミティブ : 72-bit by 512 Deep, 36kb Synchronous Simple Dual Port Block RAM with ECC (Error Correction Circuitry)

レジスタおよびラッチ

デザイン エLEMENT	説明
FDCPE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset and Clear
FDCPE_1	プリミティブ : D Flip-Flop with Negative-Edge Clock, Clock Enable, and Asynchronous Preset and Clear
FDRSE	プリミティブ : D Flip-Flop with Synchronous Reset and Set and Clock Enable
FDRSE_1	プリミティブ : D Flip-Flop with Negative-Clock Edge, Synchronous Reset and Set, and Clock Enable
IDDR	プリミティブ : Input Dual Data-Rate Register
IDDR_2CLK	プリミティブ : Input Dual Data-Rate Register with Dual Clock Inputs
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
SRLC32E	プリミティブ : 32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with 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
CARRY4	プリミティブ : Fast Carry Logic with Look Ahead
CFGLUT5	プリミティブ : 5-input Dynamically Reconfigurable Look-Up Table (LUT)
LUT5	プリミティブ : 5-Input Lookup Table with General Output
LUT5_D	プリミティブ : 5-Input Lookup Table with General and Local Outputs
LUT5_L	プリミティブ : 5-Input Lookup Table with Local Output
LUT6	プリミティブ : 6-Input Lookup Table with General Output
LUT6_2	プリミティブ : Six-input, 2-output, Look-Up Table
LUT6_D	プリミティブ : 6-Input Lookup Table with General and Local Outputs
LUT6_L	プリミティブ : 6-Input Lookup Table 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

デザイン エLEMENT

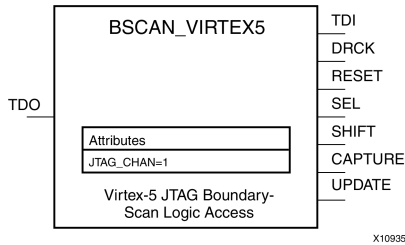
このセクションでは、このアーキテクチャで使えるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

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_VIRTEX5

プリミティブ : Virtex®-5 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_VIRTEX5: Boundary Scan primitive for connecting internal logic to
--                JTAG interface.
--                Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

BSCAN_VIRTEX5_inst : BSCAN_VIRTEX5
generic map (
  JTAG_CHAIN => 1) -- Value for USER command. 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_VIRTEX5_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// BSCAN_VIRTEX5: Boundary Scan primitive for connecting internal
//                logic to JTAG interface.
//                Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

BSCAN_VIRTEX5 #(
  .JTAG_CHAIN(1) // Value for USER command. Possible values: (1,2,3 or 4)
) BSCAN_VIRTEX5_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_VIRTEX5_inst instantiation
```

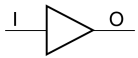
詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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



X1065.3

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
```

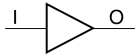
詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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

BUFG



X1065.4

概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFG_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

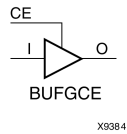
// End of BUFG_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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 ouptput
    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
);

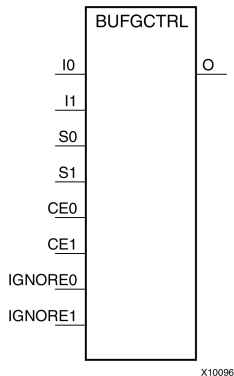
// End of BUFGCE_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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。

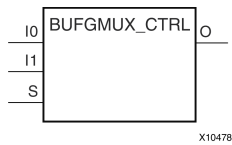
メモ : 2 つの PRESELECT 属性を同時に TRUE にすることはできません。

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFGMUX_CTRL

プリミティブ : 2-to-1 Global Clock MUX Buffer



概要

このデザイン エLEMENT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。このセレクト入力は、グローバル クロック リソースを駆動する 2 つのクロックのいずれかを選択するときに使用します。このコンポーネントは 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ELEMENTは、S ピンを 2:1 マルチプレクサのセレクト ピンとして使用します。この S ピンは、バッファの出力にグリッチを発生させることなく、いつでも切り替えることができます。

ポートの説明

ポート名	方向	幅	機能
O	出力	1 ビット	クロック出力
I0	入力	1 ビット	2 つのクロック入力の 1 つ
I1	入力	1 ビット	2 つのクロック入力の 1 つ
S	入力	1 ビット	I0 (S=0) または I1 (S=1) クロック出力の選択

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_CTRL: Global Clock Buffer 2-to-1 MUX
--               Virtex-5/6
-- Xilinx HDL Libraries Guide, version 11.2

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

-- End of BUFGMUX_CTRL_inst instantiation
```

Verilog 記述 (インスタンシエーション)

```
// BUFGMUX_CTRL: Global Clock Buffer 2-to-1 MUX
//                      Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

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

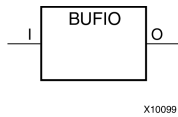
// End of BUFGMUX_CTRL_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

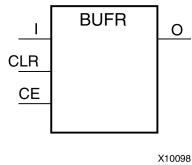
// End of BUFIO_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

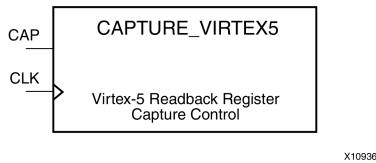
// End of BUFR_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

CAPTURE_VIRTEX5

プリミティブ : Virtex®-5 Readback Register Capture Control



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;

-- CAPTURE_VIRTEX5: Register State Capture for Bitstream Readback
--               Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

CAPTURE_VIRTEX5_inst : CAPTURE_VIRTEX5
generic map (
    ONESHOT => TRUE) -- TRUE or FALSE
port map (
    CAP => CAP,      -- Capture input
    CLK => CLK       -- Clock input
);
-- End of CAPTURE_VIRTEX5_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// CAPTURE_VIRTEX5: Register State Capture for Bitstream Readback
//               Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

CAPTURE_VIRTEX5 #(
    .ONESHOT("TRUE") // "TRUE" or "FALSE"
) CAPTURE_VIRTEX5_inst (
    .CAP(CAP),        // Capture input
    .CLK(CLK)         // Clock input
);

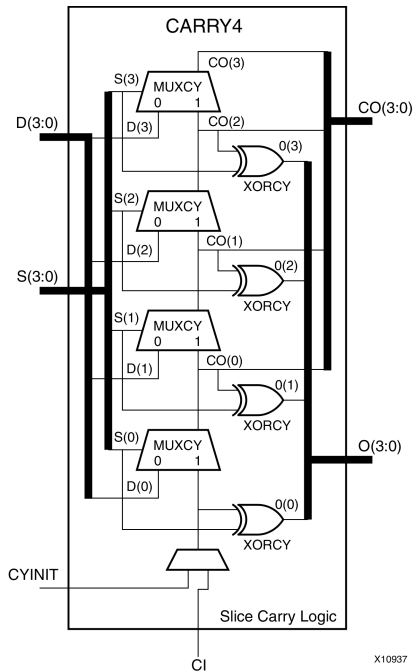
// End of CAPTURE_VIRTEX5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

CARRY4

プリミティブ : Fast Carry Logic with Look Ahead



概要

このデザイン エLEMENTは、スライスの高速キャリー ロジックです。キャリー チェーンには MUX および XOR がそれぞれ 4 個含まれています。これらの MUX および XOR はさらに複雑なファンクションを形成するために、専用配線を介してスライス内のその他のロジック (LUT) に接続されます。高速キャリー ロジックは、加算器、カウンタ、減算器、加減算器などの演算ファンクションの構築に加え、幅広のコンパレータ、アドレス デコーダ、ロジック ゲート (AND、OR、XOR など) などのその他のロジック ファンクションに使用できます。

ポートの説明

ポート名	方向	幅	機能
O	出力	4	キャリー チェーン XOR の通常データ出力ポート
CO	出力	4	キャリー チェーンの各段のキャリー出力ポート
DI	入力	4	キャリー MUX のデータ入力
S	入力	4	キャリー MUX のセレクト入力
CYINIT	入力	1	キャリー 初期化入力ポート
CI	入力	1	キャリー カスケード入力ポート

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;

-- CARRY4: Fast Carry Logic Component
--          Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

CARRY4_inst : CARRY4
port map (
    CO => CO,           -- 4-bit carry out
    O => O,             -- 4-bit carry chain XOR data out
    CI => CI,           -- 1-bit carry cascade input
    CYINIT => CYINIT,   -- 1-bit carry initialization
    DI => DI,           -- 4-bit carry-MUX data in
    S => S              -- 4-bit carry-MUX select input
);

-- End of CARRY4_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// CARRY4: Fast Carry Logic Component
//          Virtex-5, Virtex-5, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

CARRY4 CARRY4_inst (
    .CO(CO),           // 4-bit carry out
    .O(O),             // 4-bit carry chain XOR data out
    .CI(CI),           // 1-bit carry cascade input
    .CYINIT(CYINIT),  // 1-bit carry initialization
    .DI(DI),           // 4-bit carry-MUX data in
    .S(S)              // 4-bit carry-MUX select input
);

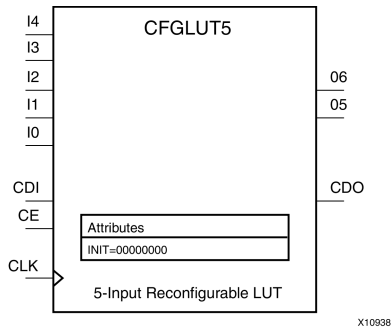
// End of CARRY4_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

CFGLUT5

プリミティブ : 5-input Dynamically Reconfigurable Look-Up Table (LUT)



概要

このデザイン エLEMENTは、ランタイムのダイナミック リコンフィギュレーションが可能な 5 入力ルックアップ テーブル (LUT) で、回路の動作中に LUT のロジック ファンクションを変更できます。CDI ピンを使用すると、クロックに同期して新しい INIT 値がシリアルにシフトされ、ロジック ファンクションが変更されます。O6 出力ピンでは、LUT に読み込まれた現在の INIT 値と現在選択されている I0 ~ I4 の入力ピンに基づいてロジック ファンクションが生成されます。オプションで O5 出力と O6 出力を使用して、同じ入力を共有する 4 入力ファンクションを 2 つ作成するか、または 5 入力ファンクション 1 つとその 5 入力ロジックのサブセットを使用する 4 入力ファンクションを作成できます (下の表を参照)。このELEMENTには、1 つのスライスに含まれる 6 入力 LUT 4 個のうちの 1 つが含まれます。

このELEMENTをカスケード接続する場合は、CDO ピンを次のELEMENTの CDI ピンに接続することで、1 つのシリアルチェーンのデータ (LUT ごとに 32 ビット) で複数のELEMENTをリコンフィギュレーションできます。

ポートの説明

ポート名	方向	幅	機能
O6	出力	1	5 入力 LUT 出力
O5	出力	1	4 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力
CDO	出力	1	リコンフィギュレーション データのカスケード出力 (オプションで後続の LUT の CDI 入力に接続)
CDI	入力	1	リコンフィギュレーション データ シリアル入力
CLK	入力	1	リコンフィギュレーション クロック
CE	入力	1	アクティブ High リコンフィギュレーション クロック イネーブル

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

- ・ CLK 入力をリコンフィギュレーション データを供給するのに使用するクロック ソースに接続します。
- ・ CDI 入力をリコンフィギュレーション データのソースに接続します。
- ・ CE ピンを LUT のリコンフィギュレーションをイネーブルまたはディスエーブルにするには、アクティブ High のロジックに接続します。
- ・ I4 ~ I0 ピンを論理式のソース入力に接続します。ロジック ファンクションは、O6 および O5 から出力されます。
- ・ このエレメントをカスケード接続する場合は、CDO ピンを次のエレメントの CDI ピンに接続することで、1 つのシリアル チェーンのデータで複数のエレメントをリコンフィギュレーションできます。

INIT 属性をこのデザイン エLEMENT に設定して、LUT の初期ロジック ファンクションを指定する必要があります。新しい INIT 値は、チェーンに含まれる LUT に 32 ビットずつシフトさせることで、回路の作動中いつでも読み込むことができます。O6 および O5 の出力値は、新しい 32 ビットの INIT 値が LUT に入力されるまで無視します。LUT のロジック ファンクションは、新しい INIT 値が LUT にシフトされるときに変化します。データは MSB (INIT[31]) から順に LSB (INIT[0]) までシフトされる必要があります。

次の表に示すように、O6 および O5 の論理値は、現在の INIT 値に基づいています。

I4 I3 I2 I1 I0	O6 値	O5 値
1 1 1 1 1	INIT[31]	INIT[15]
1 1 1 1 0	INIT[30]	INIT[14]
...
1 0 0 0 1	INIT[17]	INIT[1]
1 0 0 0 0	INIT[16]	INIT[0]
0 1 1 1 1	INIT[15]	INIT[15]
0 1 1 1 0	INIT[14]	INIT[14]
...
0 0 0 0 1	INIT[1]	INIT[1]
0 0 0 0 0	INIT[0]	INIT[0]

たとえば INIT 値が FFFF8000 の場合は、次の論理式を表します。

- ・ O6 = I4 または (I3、I2、I1、および I0)
- ・ O5 = I3、I2、I1、および I0

入力を共有するが機能は異なる 2 つの 4 入力 LUT として使用するには、I4 信号の論理値を 1 にします。INIT[31:16] が O6 出力の論理値に、INIT [15:0] の値が O5 出力の論理値に適用されます。

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	このエレメントの初期値を指定します。

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;

-- CFGLUT5: Reconfigurable 5-input LUT
--           Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

CFGLUT5_inst : CFGLUT5
generic map (
    INT => X"00000000")
port map (
    CDO => CDO, -- Reconfiguration cascade output
    O5 => O5,   -- 4-LUT output
    O6 => O6,   -- 5-LUT output
    CDI => CDI, -- Reconfiguration data input
    CE  => CE,  -- Reconfiguration enable input
    CLK => CLK, -- Clock input
    I0  => I0,  -- Logic data input
    I1  => I1,  -- Logic data input
    I2  => I2,  -- Logic data input
    I3  => I3,  -- Logic data input
    I4  => I4,  -- Logic data input
);

-- End of CFGLUT5_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// CFGLUT5: Reconfigurable 5-input LUT
//           Virtex-5, Virtex-5, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

CFGLUT5 #(
    .INIT(32'h00000000) // Specify initial LUT contents
) CFGLUT5_inst (
    .CDO(CDO), // Reconfiguration cascade output
    .O5(O5),   // 4-LUT output
    .O6(O6),   // 5-LUT output
    .CDI(CDI), // Reconfiguration data input
    .CE(CE),   // Reconfiguration enable input
    .CLK(CLK), // Clock input
    .I0(I0),   // Logic data input
    .I1(I1),   // Logic data input
    .I2(I2),   // Logic data input
    .I3(I3),   // Logic data input
    .I4(I4),   // Logic data input
);

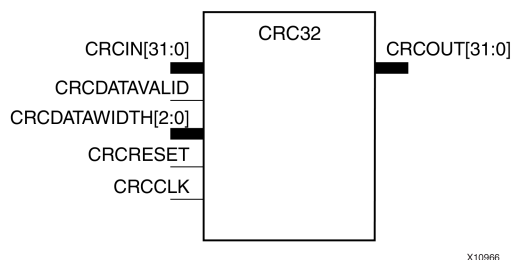
// End of CFGLUT5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

CRC32

プリミティブ : Cyclic Redundancy Check Calculator for 32 bits



概要

このデザイン エLEMENTは、フレームの内容を算出して、転送または格納前にフレームの最後に追加します。各 CRC ブロックでは、PCI EXPRESS®, ギガビット イーサネット、およびその他の汎用プロトコルに対して指定されている CRC-32 多項式を使用して 32 ビットの CRC が算出されます。32 ビットの CRC のプリミティブ CRC64 では、8、16、24、または 32 ビットの入力データを処理して 32 ビット CRC を生成できます。

ポートの説明

ポート名	方向	幅	機能		
CRCIN[31:0]	入力	32	CRC 入力データ。最大データ バス幅は 4 バイトです。		
CRCDATAVALID	入力	1	CRCIN 入力のデータが有効であることを示します。		
			1'b1: データは有効です。		
			1'b0: データは無効です。		
			この信号をディアサートすると、ディアサートされているクロック サイクル間、CRC の値が保持されます。		
CRCDATAWIDTH[2:0]	入力	3	有効な入力データ バイト数を示します。		
			CRCDATAWIDTH[2:0]	データ幅	CRC データ バスビット
			0	8 ビット	CRCIN[31:24]
			1	16 ビット	CRCIN[31:16]
			10	24 ビット	CRCIN[31:8]
			11	32 ビット	CRCIN[31:0]
CRCRESET	入力	1	CRC レジスタの同期リセット。アサートされると、CRC ブロックが CRC_INIT の値に初期化されます。		
CRCCLK	入力	1	CRC クロック		
CRCOUT[31:0]	出力	32	32 ビット CRC 出力。バイトが反転された、ビット反転 CRC 値で、直前のクロック サイクルの有効バイトおよび直前の CRC 値での CRC 計算に対応しています。CRCDATAVALIDA が 1 に設定されている必要があります。		

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

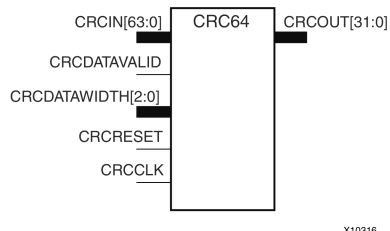
属性	タイプ	値	デフォルト	説明
CRC_INIT[31:0]	16 進数	32 ビット値	0xFFFFFFFF	CRC の内部レジスタの初期値を設定します。LX30T および LX50T ES シリコンでは、値が 0xFFFFFFFF に固定されています。

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

CRC64

プリミティブ : Cyclic Redundancy Check Calculator for 64 bits



X10316

概要

このデザイン エLEMENTは、フレームの内容を算出して、転送または格納前にフレームの最後に追加します。各 CRC ブロックでは、PCI EXPRESS®、ギガビット イーサネット、およびその他の汎用プロトコルに対して指定されている CRC-32 多項式を使用して 32 ビットの CRC が算出されます。64 ビットの CRC のプリミティブ CRC64 では、8、16、24、32、40、56、または 64 ビットの入力データを処理して 32 ビット CRC を生成できます。CRC64 プリミティブを使用すると、1 つのトランシーバ タイルで対になっている両方の CRC ハード ブロックが使用されます。

ポートの説明

ポート名	方向	幅	機能		
CRCIN[63:0]	入力	64	CRC 入力データ。最大データ パス幅は 8 バイトです。		
CRCDATAVALID	入力	1	CRCIN 入力のデータが有効であることを示します。		
			1'b1: データは有効です。		
			1'b0: データは無効です。		
			この信号をディアサートすると、ディアサートされているクロック サイクル間、CRC の値が保持されます。		
CRCDATAWIDTH[2:0]	入力	3	有効な入力データ バイト数を示します。		
			CRCDATAWIDTH[2:0]	データ幅	CRC データ バス ビット
			0	8 ビット	CRCIN[63:56]
			1	16 ビット	CRCIN[63:48]
			10	24 ビット	CRCIN[63:40]
			11	32 ビット	CRCIN[63:32]
			100	40 ビット	CRCIN[63:24]
			101	48 ビット	CRCIN[63:16]
			110	56 ビット	CRCIN[63:8]
			111	64 ビット	CRCIN[63:0]
CRCRESET	入力	1	CRC レジスタの同期リセット。アサートされると、CRC ブロックが CRC_INIT の値に初期化されます。		
CRCCLK	入力	1	CRC クロック		
CRCOUT[31:0]	出力	32	32 ビット CRC 出力。バイトが反転された、ビット反転 CRC 値で、直前のクロック サイクルの有効バイトおよび直前の CRC 値での CRC 計算に対応しています。CRCDATAVALID が 1 に設定されている必要があります。		

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

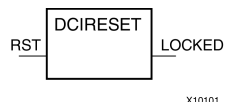
属性	タイプ	値	デフォルト	説明
CRC_INIT[31:0]	16 進数	32 ビット値	0xFFFFFFFF	CRC の内部レジスタの初期値を設定します。LX30T および LX50T ES シリコンでは、値が 0xFFFFFFFF に固定されています。

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

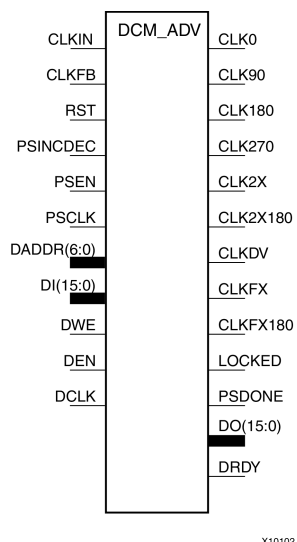
// End of DCIRESET_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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



X10102

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
);

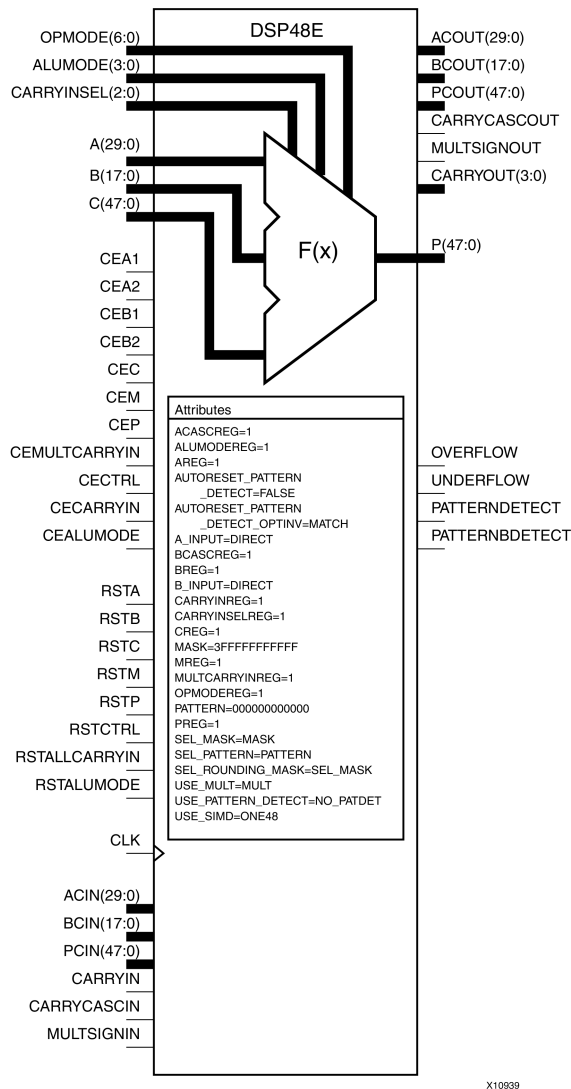
// End of DCM_ADV_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

DSP48E

プリミティブ : 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtractor/Accumulator or 2-Input Logic Unit



X10939

概要

このデザイン エLEMENTは、柔軟性が高い多用途のハード IP ブロックで、多くの DSP アルゴリズムで見られる小型で高速な演算処理を作成できます。このブロックでは、乗算、加算、減算、累積、シフト、論理処理、およびパターン検出などが実行できます。

ポートの説明

ポート名	方向	幅	機能
データ ポート			
A	入力	30	乗算器への 25 ビット データ入力または加算器/ロジック ユニット (LU) への 30 ビット MSB データ入力

ポート名	方向	幅	機能
B	入力	18	乗算器への 18 ビット データ入力または加算器/ロジック ユニット (LU) への 18 ビット LSB データ入力
C	入力	48	加算器/ロジック ユニット (LU) への 48 ビット データ入力および (または) パターン検出
CARRYIN	入力	1	加算器/ロジック ユニット (LU) への外部キャリー入力
P	出力	48	プライマリ データ出力
CARRYOUT	出力	4	<p>演算処理 (加算、減算など) のキャリー出力信号</p> <ul style="list-style-type: none"> USE_SIMD が FOUR12 の場合、CARRYOUT[3:0] は累積/加算/ロジック ユニット (LU) それぞれからの 12 ビットのキャリー出力を指します。 USE_SIMD が TWO24 の場合、CARRYOUT[3] および CARRYOUT[1] は累積/加算/ALU それぞれからの 24 ビットのキャリー出力を指します。 USE_SIMD が ONE48 の場合、CARRYOUT[3] は累積/加算/ロジック ユニット (LU) からの唯一の有効なキャリー出力です。
制御入力/ステータス ビット			
CLK	入力	1	DSP48E のクロック入力
OPMODE	入力	7	ALUMODE と共に DSP48E の演算処理を選択する制御入力です。
ALUMODE	入力	4	加算および減算を含むロジック ユニット (LU) ファンクションを選択する制御入力です。
CARRYINSEL	入力	3	DSP48E へのキャリー入力ソースを選択します。
OVERFLOW	出力	1	パターン検出器が使用され PREG=1 のときに、加算器/累積でオーバーフローを検出するアクティブ High の出力です。
UNDERFLOW	出力	1	パターン検出器が使用され PREG=1 のときに、加算器/累積でアンダーフローを検出するアクティブ High の出力です。
PATTERNDETECT	出力	1	アクティブ High のパターン検出で、MASK 属性で指定した箇所で PATTERN の反転した値と P の値が一致したときに High になります。結果は P と同じクロック サイクルで出力されます。
PATTERN BDETECT	出力	1	アクティブ High のパターン検出で、MASK 属性で指定した箇所で PATTERN の反転した値と P の値が一致したときに High になります。結果は P と同じクロック サイクルで出力されます。
リセット/クロック イネーブル入力			
RSTA	入力	1	A ポートレジスタ (AREG=1 または 2) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
RSTB	入力	1	B ポートレジスタ (BREG=1 または 2) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
RSTC	入力	1	C ポートレジスタ (CREG=1 または 2) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
RSTM	入力	1	乗算器レジスタ (MREG=1) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
RSTP	入力	1	P 出力、UNDERFLOW 出力、OVERFLOW 出力、PATTERNDETECT 出力、PATTERNBDETECT 出力、および CARRYOUT 出力レジスタ (PREG=1) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。

ポート名	方向	幅	機能
RSTCTRL	入力	1	OPMODE および CARRYINSEL レジスタ (OPMODEREG=1 および CARRYINSELREG=1) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
RSTALLCARRYIN	入力	1	全キャリー入力レジスタ (CARRYINREG=1 または MULTCARRYINREG=1) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
RSTALUMODE	入力	1	ALUMODE レジスタ (ALUMODEREG=1) のアクティブ High の同期リセットで、使用しない場合は論理値を 0 にします。
CEA1	入力	1	A ポートレジスタ (AREG=2) のアクティブ High のクロック イネーブルで、使用しない場合と AREG=2 の場合は論理値を 1 に、AREG=0 または 1 の場合は論理値を 0 にします。2 つのレジスタを使用する場合は、このレジスタが 1 段目になります。
CEA2	入力	1	A ポートレジスタのアクティブ High のクロック イネーブルで、使用しない場合と AREG=2 の場合は論理値を 1 に、AREG=0 または 1 の場合は論理値を 0 にします。2 つのレジスタを使用する場合は、このレジスタが 2 段目になります。
CEB1	入力	1	B ポートレジスタ (BREG=2) のアクティブ High のクロック イネーブルで、使用しない場合と BREG=2 の場合は論理値を 1 に、BREG=0 または 1 の場合は論理値を 0 にします。2 つのレジスタを使用する場合は、このレジスタが 1 段目になります。
CEB2	入力	1	B ポートレジスタのアクティブ High のクロック イネーブルで、使用しない場合と BREG=2 の場合は論理値を 1 に、BREG=0 または 1 の場合は論理値を 0 にします。2 つのレジスタを使用する場合は、このレジスタが 2 段目になります。
CEC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CEM	入力	1	乗算のレジスタ (MREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CEP	入力	1	出力ポートレジスタ (PREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CECTRL	入力	1	OPMODE およびキャリー入力セレクト レジスタ (CTRLREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CECARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CEMULTCARRYIN	入力	1	対称的な丸めを実行する乗算器の内部キャリー レジスタのクロック イネーブルです (MULTCARRYINREG=1)。
CEALUMODE	入力	1	ALUMODE 入力レジスタ (ALUMODEREG=1) のクロック イネーブルです。
カスケード ポート			
ACIN	入力	30	ポート A のカスケード入力で、カスケード接続されている上位の DSP48E の ACOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
BCIN	入力	18	ポート B のカスケード入力で、カスケード接続されている上位の DSP48E の BCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
PCIN	入力	48	ポート P のカスケード入力で、カスケード接続されている上位の DSP48E の PCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。

ポート名	方向	幅	機能
CARRYCASCIN	入力	1	上位の DSP48E の CARRYOUT[2] とカスケード接続されます。
MULTSIGNIN	入力	1	48 ビット以上の出力が必要な場合に、カスケード接続されている DSP48E の乗算器の符号出力を使用して、加算器/累積器の出力の符号を拡張します。MULTSIGNOUT 出力ピンにのみ接続します。
ACOUT	出力	30	ポート A のカスケード出力で、カスケード接続されている下位の DSP48E の ACIN に接続します。使用しない場合は未接続にします。
BCOUT	出力	18	ポート B のカスケード出力で、カスケード接続されている下位の DSP48E の BCIN に接続します。使用しない場合は未接続にします。
PCOUT	出力	48	ポート P のカスケード出力で、カスケード接続されている下位の DSP48E の PCIN に接続します。使用しない場合は未接続にします。
CARRYCASCOUT	出力	1	次の DSP48E にカスケード接続された CARRYOUT[3]
MULTSIGNOUT	出力	1	乗算器の符号出力を使用して、カスケード接続されている DSP48E で加算器/累積器の符号を拡張します。MULTSIGNIN 入力ピンにのみ接続します。

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
ACASCREG	整数	0、1、2	1	AREG 属性と組み合わせて使用し、ACIN カスケード入力の A 入力レジスタの数を指定します。AREG の値以下にする必要があります。
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。
ALUMODEREG	整数	0、1	1	ALUMODE 入力ピンにレジスタを付けるかを指定します。
AUTORESET_PATTERN_DETECT	ブール代数	TRUE、FALSE	FALSE	AUTORESET_PATTERN_DETECT_OPTINV で定義されたパターン検出イベントがこのクロックサイクルで発生した場合、DSP48E の P レジスタ (累積値またはカウンタ値) を次のクロックサイクルで自動的にリセットします。
AUTORESET_PATTERN_DETECT_OPTINV	文字列	MATCH、NOT_MATCH	MATCH	A) パターンが一致したとき、または B) パターンは現在のクロックサイクルでは一致しないがその直前のクロックサイクルで一致していた場合に、AUTORESET_PATTERN_DETECT により P レジスタを次のクロックサイクルで自動的にリセットするべきかを決定します。
A_INPUT	文字列	DIRECT、CASCADE	DIRECT	A (DIRECT) または ACIN (CASCADE) のいずれかを選択します。

属性	タイプ	値	デフォルト	説明
BCASCREG	整数	0、1、2	1	BREG 属性と組み合わせて使用し、BCIN カスケード入力の B 入力レジスタの数を指定します。
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。
B_INPUT	文字列	DIRECT、CASCADE	DIRECT	B (DIRECT) または BCIN (CASCADE) のいずれかを選択します。
CARRYINREG	整数	0、1	1	CARRYIN 入力にレジスタを付けるかを指定します。
CARRYINSELREG	整数	0、1	1	CARRYINSEL 入力にレジスタを付けるかを指定します。
CREG	整数	0、1	1	C 入力にレジスタを付けるかを指定します。
MASK	16 進数	48 ビット値	3FFF	パターン検出器で使用されるマスクを指定します。
MREG	整数	0、1	1	乗算器の段にレジスタを付けるかを指定します。 イネーブル = 1/ディスエーブル = 0
MULTCARRYINREG	整数	0、1	1	内部キャリーのレジスタ数を選択します (対称的な丸めを実行する乗算でのみ使用)。
OPMODEREG	整数	0、1	1	OPMODE 入力にレジスタを付けるかを指定します。
PATTERN	16 進数	48 ビット値	すべてゼロ	パターン検出器で使用されるパターンを指定します。
PREG	整数	0、1	1	P 入力にレジスタを付けるかを指定します。
SEL_MASK	文字列	MASK、C	MASK	パターン検出器のマスクに MASK を使用するか、C 入力を使用するかを指定します。
SEL_PATTERN	文字列	PATTERN、C	PATTERN	パターン検出器のパターンに PATTERN を使用するか、C 入力を使用するかを指定します。
SEL_ROUNDING_MASK	文字列	SEL_MASK、MODE1、MODE2	SEL_MASK	パターン検出で対称的な丸めと偶数丸めに使用するマスクを選択します。MODE1 または MODE2 に設定すると、SEL_MASK 属性は無視されます。これらは、偶数丸めに使用されます。
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
USE_MULT	文字列	MULT、MULT_S、NONE	MULT_S	乗算器の使用方法を選択します。NONE に設定すると、加算器/ロジックユニットのみを使用するときに電力を節約できます。MPEG が 0 の場合は MULT に、1 の場合は MULT_S に設定します。

属性	タイプ	値	デフォルト	説明
USE_SIMD	文字列	ONE48、 TWO24、 FOUR12	ONE48	SIMD (Single Instruction Multiple Data) 加算器/ロジックユニットの使用方法を選択します。48 ビットのロジックユニット 1 個、24 ビットのロジックユニット 2 個、または 12 ビットのロジックユニット 4 個から選択します。12 ビットのロジックユニット 4 個では、同じ命令が実行されることに注意してください。つまり、すべてのロジックユニットで減算または加算が同サイクルで実行されます。これにより、計算量の比較的少ないアプリケーション向けに 48 ビットの加算器を小型の加算器に分割できます。SIMD は、加算、累積、減算などの演算処理にのみに影響し、論理処理には影響しません。
USE_PATTERN_DETECT	文字列	PAT_DET、 NO_PAT_DET	NO_PAT_DET	パターン検出をイネーブルにします。シミュレーションおよびスピード ファイルのみに実行されます。

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;

-- DSP48E: DSP Function Block
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

DSP48E_inst : DSP48E
generic map (
  ACASCREG => 1,          -- Number of pipeline registers between
                           -- A/ACIN input and ACOUT output, 0, 1, or 2
  ALUMODEREG => 1,        -- Number of pipeline registers on ALUMODE input, 0 or 1
  AREG => 1,              -- Number of pipeline registers on the A input, 0, 1 or 2
  AUTORESET_PATTERN_DETECT => FALSE, -- Auto-reset upon pattern detect, TRUE or FALSE
  AUTORESET_PATTERN_DETECT_OPTINV => "MATCH", -- Reset if "MATCH" or "NOMATCH"
  A_INPUT => "DIRECT",    -- Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
  BCASCREG => 1,          -- Number of pipeline registers between B/BCIN input and BCOUT output, 0, 1, or 2
  BREG => 1,              -- Number of pipeline registers on the B input, 0, 1 or 2
  B_INPUT => "DIRECT",    -- Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
  CARRYINREG => 1,        -- Number of pipeline registers for the CARRYIN input, 0 or 1
  CARRYINSELREG => 1,     -- Number of pipeline registers for the CARRYINSEL input, 0 or 1
  CREG => 1,              -- Number of pipeline registers on the C input, 0 or 1
  MASK => X"3FFFFFFFFF",  -- 48-bit Mask value for pattern detect
  MREG => 1,              -- Number of multiplier pipeline registers, 0 or 1
  MULTCARRYINREG => 1,    -- Number of pipeline registers for multiplier carry in bit, 0 or 1
  OPMODEREG => 1,         -- Number of pipeline registers on OPMODE input, 0 or 1
  PATTERN => X"000000000000", -- 48-bit Pattern match for pattern detect
  PREG => 1,              -- Number of pipeline registers on the P output, 0 or 1
  SIM_MODE => "SAFE",     -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
                           -- Design Guide" for details
  SEL_MASK => "MASK",    -- Select mask value between the "MASK" value or the value on the "C" port
  SEL_PATTERN => "PATTERN", -- Select pattern value between the "PATTERN" value or the value on the "C" port
  SEL_ROUNDING_MASK => "SEL_MASK", -- "SEL_MASK", "MODE1", "MODE2"
  USE_MULT => "MULT_S",  -- Select multiplier usage, "MULT" (MREG => 0),
                           -- "MULT_S" (MREG => 1), "NONE" (not using multiplier)
  USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect, "PATDET", "NO_PATDET"
  USE_SIMD => "ONE48") -- SIMD selection, "ONE48", "TWO24", "FOUR12"
port map (
  ACOUT => ACOUT, -- 30-bit A port cascade output
  BCOUT => BCOUT, -- 18-bit B port cascade output
  CARRYCASCOUT => CARRYCASCOUT, -- 1-bit cascade carry output
  CARRYOUT => CARRYOUT, -- 4-bit carry output
  MULTSIGNOUT => MULTSIGNOUT, -- 1-bit multiplier sign cascade output
  OVERFLOW => OVERFLOW, -- 1-bit overflow in add/acc output
  P => P,          -- 48-bit output

```

```

PATTERNBDETECT => PATTERNBDETECT, -- 1-bit active high pattern bar detect output
PATTERNDETECT => PATTERNDETECT, -- 1-bit active high pattern detect output
PCOUT => PCOUT, -- 48-bit cascade output
UNDERFLOW => UNDERFLOW, -- 1-bit active high underflow in add/acc output
A => A, -- 30-bit A data input
ACIN => ACIN, -- 30-bit A cascade data input
ALUMODE => ALUMODE, -- 4-bit ALU control input
B => B, -- 18-bit B data input
BCIN => BCIN, -- 18-bit B cascade input
C => C, -- 48-bit C data input
CARRYCASCIN => CARRYCASCIN, -- 1-bit cascade carry input
CARRYIN => CARRYIN, -- 1-bit carry input signal
CARRYINSEL => CARRYINSEL, -- 3-bit carry select input
CEA1 => CEA1, -- 1-bit active high clock enable input for 1st stage A registers
CEA2 => CEA2, -- 1-bit active high clock enable input for 2nd stage A registers
CEALUMODE => CEALUMODE, -- 1-bit active high clock enable input for ALUMODE registers
CEB1 => CEB1, -- 1-bit active high clock enable input for 1st stage B registers
CEB2 => CEB2, -- 1-bit active high clock enable input for 2nd stage B registers
CEC => CEC, -- 1-bit active high clock enable input for C registers
CECARRYIN => CECARRYIN, -- 1-bit active high clock enable input for CARRYIN register
CECTRL => CECTRL, -- 1-bit active high clock enable input for OPMODE and carry registers
CEM => CEM, -- 1-bit active high clock enable input for multiplier registers
CEMULTCARRYIN => CEMULTCARRYIN, -- 1-bit active high clock enable for multiplier carry in register
CEP => CEP, -- 1-bit active high clock enable input for P registers
CLK => CLK, -- Clock input
MULTSIGNIN => MULTSIGNIN, -- 1-bit multiplier sign input
OPMODE => OPMODE, -- 7-bit operation mode input
PCIN => PCIN, -- 48-bit P cascade input
RSTA => RSTA, -- 1-bit reset input for A pipeline registers
RSTALLCARRYIN => RSTALLCARRYIN, -- 1-bit reset input for carry pipeline registers
RSTALUMODE => RSTALUMODE, -- 1-bit reset input for ALUMODE pipeline registers
RSTB => RSTB, -- 1-bit reset input for B pipeline registers
RSTC => RSTC, -- 1-bit reset input for C pipeline registers
RSTCTRL => RSTCTRL, -- 1-bit reset input for OPMODE pipeline registers
RSTM => RSTM, -- 1-bit reset input for multiplier registers
RSTP => RSTP, -- 1-bit reset input for P pipeline registers
);

-- End of DSP48E_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// DSP48E: DSP Function Block
//      Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

DSP48E #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ACASCREG(1), // Number of pipeline registers between A/ACIN input and ACOUT output, 0, 1, or 2
    .ALUMODEREG(1), // Number of pipeline registers on ALUMODE input, 0 or 1
    .AREG(1), // Number of pipeline registers on the A input, 0, 1 or 2
    .AUTORESET_PATTERN_DETECT("FALSE"), // Auto-reset upon pattern detect, "TRUE" or "FALSE"
    .AUTORESET_PATTERN_DETECT_OPTINV("MATCH"), // Reset if "MATCH" or "NOMATCH"
    .A_INPUT("DIRECT"), // Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
    .BCASCREG(1), // Number of pipeline registers between B/BCIN input and BCOUT output, 0, 1, or 2
    .BREG(1), // Number of pipeline registers on the B input, 0, 1 or 2
    .B_INPUT("DIRECT"), // Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
    .CARRYINREG(1), // Number of pipeline registers for the CARRYIN input, 0 or 1
    .CARRYINSELREG(1), // Number of pipeline registers for the CARRYINSEL input, 0 or 1
    .CREG(1), // Number of pipeline registers on the C input, 0 or 1
    .MASK(48'h3fffffff), // 48-bit Mask value for pattern detect
    .MREG(1), // Number of multiplier pipeline registers, 0 or 1
    .MULTCARRYINREG(1), // Number of pipeline registers for multiplier carry in bit, 0 or 1
    .OPMODEREG(1), // Number of pipeline registers on OPMODE input, 0 or 1
    .PATTERN(48'h000000000000), // 48-bit Pattern match for pattern detect
    .PREG(1), // Number of pipeline registers on the P output, 0 or 1
    .SEL_MASK("MASK"), // Select mask value between the "MASK" value or the value on the "C" port
    .SEL_PATTERN("PATTERN"), // Select pattern value between the "PATTERN" value or the value on the "C" port
    .SEL_ROUNDING_MASK("SEL_MASK"), // "SEL_MASK", "MODE1", "MODE2"
    .USE_MULT("MULT_S"), // Select multiplier usage, "MULT" (MREG => 0), "MULT_S" (MREG => 1), "NONE" (no multiplier)
    .USE_PATTERN_DETECT("NO_PATDET"), // Enable pattern detect, "PATDET", "NO_PATDET"
    .USE_SIMD("ONE48") // SIMD selection, "ONE48", "TWO24", "FOUR12"

```

```

) DSP48E_inst (
  .ACOUT(ACOUT), // 30-bit A port cascade output
  .BCOUT(BCOUT), // 18-bit B port cascade output
  .CARRYCASCOUT(CARRYCASCOUT), // 1-bit cascade carry output
  .CARRYOUT(CARRYOUT), // 4-bit carry output
  .MULTSIGNOUT(MULTSIGNOUT), // 1-bit multiplier sign cascade output
  .OVERFLOW(OVERFLOW), // 1-bit overflow in add/acc output
  .P(P), // 48-bit output
  .PATTERNBDETECT(PATTERNBDETECT), // 1-bit active high pattern bar detect output
  .PATTERNDETECT(PATTERNDETECT), // 1-bit active high pattern detect output
  .PCOUT(PCOUT), // 48-bit cascade output
  .UNDERFLOW(UNDERFLOW), // 1-bit active high underflow in add/acc output
  .A(A), // 30-bit A data input
  .ACIN(ACIN), // 30-bit A cascade data input
  .ALUMODE(ALUMODE), // 4-bit ALU control input
  .B(B), // 18-bit B data input
  .BCIN(BCIN), // 18-bit B cascade input
  .C(C), // 48-bit C data input
  .CARRYCASCIN(CARRYCASCIN), // 1-bit cascade carry input
  .CARRYIN(CARRYIN), // 1-bit carry input signal
  .CARRYINSEL(CARRYINSEL), // 3-bit carry select input
  .CEA1(CEA1), // 1-bit active high clock enable input for 1st stage A registers
  .CEA2(CEA2), // 1-bit active high clock enable input for 2nd stage A registers
  .CEALUMODE(CEALUMODE), // 1-bit active high clock enable input for ALUMODE registers
  .CEB1(CEB1), // 1-bit active high clock enable input for 1st stage B registers
  .CEB2(CEB2), // 1-bit active high clock enable input for 2nd stage B registers
  .CEC(CEC), // 1-bit active high clock enable input for C registers
  .CECARRYIN(CECARRYIN), // 1-bit active high clock enable input for CARRYIN register
  .CECTRL(CECTRL), // 1-bit active high clock enable input for OPMODE and carry registers
  .CEM(CEM), // 1-bit active high clock enable input for multiplier registers
  .CEMULTCARRYIN(CEMULTCARRYIN), // 1-bit active high clock enable for multiplier carry in register
  .CEP(CEP), // 1-bit active high clock enable input for P registers
  .CLK(CLK), // Clock input
  .MULTSIGNIN(MULTSIGNIN), // 1-bit multiplier sign input
  .OPMODE(OPMODE), // 7-bit operation mode input
  .PCIN(PCIN), // 48-bit P cascade input
  .RSTA(RSTA), // 1-bit reset input for A pipeline registers
  .RSTALLCARRYIN(RSTALLCARRYIN), // 1-bit reset input for carry pipeline registers
  .RSTALUMODE(RSTALUMODE), // 1-bit reset input for ALUMODE pipeline registers
  .RSTB(RSTB), // 1-bit reset input for B pipeline registers
  .RSTC(RSTC), // 1-bit reset input for C pipeline registers
  .RSTCTRL(RSTCTRL), // 1-bit reset input for OPMODE pipeline registers
  .RSTM(RSTM), // 1-bit reset input for multiplier registers
  .RSTP(RSTP) // 1-bit reset input for P pipeline registers
);

// End of DSP48E_inst instantiation

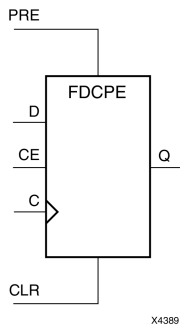
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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 入力の方が優先される)。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ことがあります。

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
);

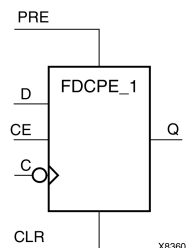
// End of FDCPE_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

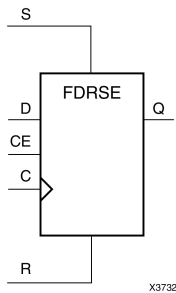
// End of FDCPE_1_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

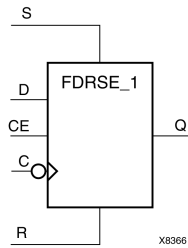
// End of FDRSE_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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

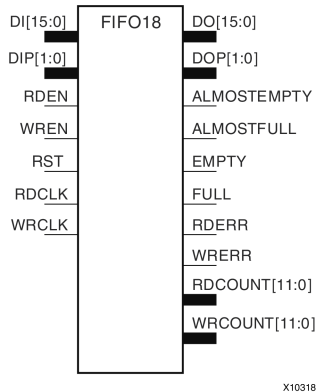
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

FIFO18

プリミティブ : 18kb FIFO (First In, First Out) Block RAM Memory



概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、これらの RAM を個別に FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36Kb または 18Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。FIFO18 では、FIFO 制御ロジックおよび 18Kb ブロック RAM が使用されます。このプリミティブは、4 ビット X 4K ワード、9 ビット X 2K ワード、または 18 ビット X 1K ワードのコンフィギュレーションで使用できます。また、このプリミティブは関連するすべての FIFO フラグおよびステータス信号と共に、同期モードまたはマルチレート (非同期) モードのいずれにもコンフィギュレーションできます。

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 サイクル後にディ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ィアサート レイテンシ サイクルのみが反映されます。

メモ : 36 ビット X 512 ワードの FIFO には、FIFO18_36 を使用します。ワード数が少なく、データ幅の狭いコンフィギュレーションには、FIFO36 または FIFO36_72 を使用します。エラー修正回路が必用な場合は、FIFO36_72 を使用します。

ポートの説明

ポート名	方向	幅	機能
DO	出力	4、8、16	FIFO データ出力バス
DOP	出力	0、1、2	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグのしきい値は ALMOST_FULL_OFFSET 属性で指定します。
EMPTY	出力	1	FIFO が空であることを示します。
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグのしきい値は ALMOST_EMPTY_OFFSET 属性で指定します。
WRERR、RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。
WRCOUNT、RDCOUNT	出力	12	FIFO 書き込み/読み出しポインタ
DI	入力	4、8、16	FIFO データ入力バス
DIP	入力	0、1、2	FIFO パリティ データ入力バス

ポート名	方向	幅	機能
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインタの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

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	不可
マクロのサポート	推奨

4 ビット幅でコンフィギュレーションされたプリミティブをインスタンス化するには、DIP ポートを論理値 0 にし、DOP ポートは未接続のままにします。DI[3:0] および DO[3:0] は適切な入力および出力信号に接続し、DI[15:4] は論理値 0 に、DO[15:4] は未接続のままにします。

9 ビット幅でコンフィギュレーションするときは、DIP[0] ポートを適切なデータ入力に接続し、DIP[1] は論理値 0 にします。DOP[0] は適切なデータ出力に接続し、DOP[1] は未接続のままにします。DI[7:0] および DO[7:0] は適切な入力および出力信号に接続し、DI[15:8] は論理値 0 に、DO[15:8] は未接続のままにします。

18 ビット幅をコンフィギュレーションするときは、DI、DIP、DO、および DOP 信号すべてを接続する必要があります。

いずれのコンフィギュレーションでも、未使用の DI または DIP 入力は論理値 0 に、未使用の DO または DOP ピンは未接続のままにする必要があります。EN_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。非同期モード (EN_SYN = FALSE) のときは、個別のクロック信号を使用できます。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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 サイクル後にディ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ィアサートレイテンシ サイクルのみが反映されます。

FIFO は電源投入後にリセットされる必要があります。FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

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	ALMOST_FULL フラグをトリガする RAM のデータ量を指定します。
ALMOST_EMPTY_OFFSET	16 進数	12 ビット値	すべてゼロ	ALMOST_EMPTY フラグをトリガする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。

属性	タイプ	値	デフォルト	説明
DATA_WIDTH	整数	4、9、18	4	FIFO に必要なデータ幅を指定
EN_SYN	ブール代数	TRUE、FALSE	FALSE	FIFO が非同期 (独立した 2 つのクロック) または同期 (1 つのクロック) のいずれで動作しているかを示します。マルチレートの場合は DO_REG=1 を設定する必要があります。
DO_REG	整数	0、1	1	EN_SYN のデータ パイプライン レジスタ
SIM_MODE	文字列	SAFE または FAST .	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。

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;

-- FIFO18: 16k+2k Parity Synchronous/Asynchronous BlockRAM FIFO BlockRAM Memory
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

FIFO18_inst : FIFO18
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, or 18
    DO_REG => 1, -- Enable output register ( 0 or 1)
    -- Must be 1 if the EN_SYN = FALSE
    EN_SYN => FALSE, -- Specified FIFO as Asynchronous (FALSE) or
    -- Synchronous (TRUE)
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 2-bit parity 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 => DI, -- 16-bit data input
    DIP => DIP, -- 2-bit parity input
    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 FIFO18_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// FIFO18: 16k+2k Parity Synchronous/Asynchronous BlockRAM FIFO
//          Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

FIFO18 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .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 or 18
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO18_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DO(DO), // 16-bit data output
    .DOP(DOP), // 2-bit parity 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(DI), // 16-bit data input
    .DIP(DIP), // 2-bit parity input
    .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
);

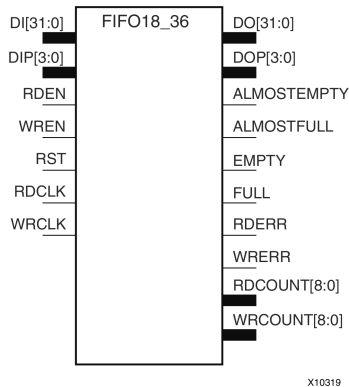
// End of FIFO18_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

FIFO18_36

プリミティブ : 36-bit Wide by 512 Deep 18kb FIFO (First In, First Out) Block RAM Memory



概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。FIFO18_36 を使用すると、幅広のデータパスが必要なときに 18Kb の FIFO ブロック RAM にアクセスできます。このコンポーネントは、36 ビット X 512 ワードの同期または非同期 FIFO RAM としてコンフィギュレーションできます。また、この FIFO RAM では関連するすべての FIFO フラグおよびステータス信号も提供されます。

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 サイクル後にディ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ィアサートレイテンシ サイクルのみが反映されます。

メモ : ワード数が多くデータ幅が狭いコンフィギュレーションには、代わりに FIFO18 を使用し、ワード数が多く、データ幅の広いコンフィギュレーションには、FIFO 36 または FIFO36_72 を使用します。エラー修正回路が必要な場合は FIFO36_72 を使用します。

ポートの説明

ポート名	方向	幅	機能
DO	出力	32	FIFO データ出力バス
DOP	出力	4	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグの値は ALMOST_FULL_OFFSET 属性で指定します。
EMPTY	出力	1	FIFO が空であることを示します。
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグの値は ALMOST_EMPTY_OFFSET 属性で指定します。
WRERR, RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。
WRCOUNT、 RDCOUNT	出力	9	FIFO 書き込み/読み出しポインタ

ポート名	方向	幅	機能
DI	入力	32	FIFO データ入力バス
DIP	入力	4	FIFO パリティデータ入力バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインタの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、 RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

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	不可
マクロのサポート	推奨

DI、DIP、DO、および DOP ピンは、対応する入力および出力データソースに接続する必要があります。36 ビット以下を使用するときは、未使用の DI または DIP 入力を論理値 0 に、未使用の DO または DOP ピンは未接続のままにする必要があります。EN_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。非同期モード (EN_SYN = FALSE) のときは、個別のクロック信号を使用します。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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 サイクル後にディ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ィアサートレイテンシ サイクルのみが反映されます。WREN および RDEN は、対応するライト イネーブルおよびリード イネーブル信号/ロジックに接続し、RST は適切なリセット信号/ロジックに接続するか、使用しない場合は論理値 0 にする必要があります。

FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

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	9 ビット値	すべてゼロ	ALMOST_FULL フラグをトリガする RAM のデータ量を指定します。
ALMOST_EMPTY_OFFSET	16 進数	9 ビット値	すべてゼロ	ALMOST_EMPTY フラグをトリガする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。
EN_SYN	ブール代数	TRUE、FALSE	FALSE	FALSE のときは非同期モード、TRUE のときは同期 (1 クロック) モードで FIFO が使用されることを示します。
DO_REG	整数	0、1	1	読み出しレイテンシ (パイプライン遅延 1 つ) 追加することで clock-to-out のタイミングを

属性	タイプ	値	デフォルト	説明
				向上するように、FIFO の出力レジスタをイネーブルします。EN_SYN が FALSE のときは DO_REG を 1 にする必要があります。
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。

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;

-- FIFO18_36: 36x18k Synchronous/Asynchronous BlockRAM FIFO
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

FIFO18_36_inst : FIFO18_36
generic map (
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DO_REG => 1, -- Enable output register (0 or 1)
    -- Must be 1 if EN_SYN = FALSE
    EN_SYN => FALSE, -- Specifies FIFO as Asynchronous (FALSE)
    -- or Synchronous (TRUE)
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 4-bit parity data output
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 9-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 9-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 32-bit data input
    DIP => DIP, -- 4-bit parity input
    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 FIFO18_36_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// FIFO18_36: 36x18k Synchronous/Asynchronous BlockRAM FIFO
//          Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

FIFO18_36 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(9'h080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO18_36_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DO(DO), // 32-bit data output
    .DOP(DOP), // 4-bit parity data output
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 9-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .WRCOUNT(WRCOUNT), // 9-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 32-bit data input
    .DIP(DIP), // 4-bit parity input
    .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
);

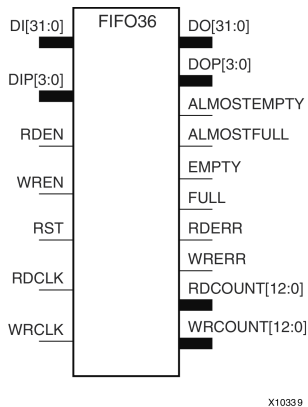
// End of FIFO18_36_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

FIFO36

プリミティブ : 36kb FIFO (First In, First Out) Block RAM Memory



概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。FIFO36 を使用すると、36Kb の FIFO のブロック RAM へアクセスできます。このコンポーネントは、関連 FIFO フラグとともに、4 ビット X 8K ワード、9 ビット X 4K ワード、18 ビット X 2K ワード、36 ビット X 1K ワードの同期またはマルチレート (非同期) FIFO RAM としてコンフィギュレーションできます。

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。

メモ : 72 ビット X 512 ワードの FIFO には、FIFO36_72 を使用します。ワード数が少なく、データ幅の狭いコンフィギュレーションには、FIFO18 または FIFO18_36 を使用します。エラー修正回路が必要な場合は、FIFO36_72 を使用します。

ポートの説明

ポート名	方向	幅	機能
DO	出力	4、8、16、32	FIFO データ出力バス
DOP	出力	0、1、2、4	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグのしきい値は ALMOST_FULL_OFFSET 属性で指定します。
EMPTY	出力	1	FIFO が空であることを示します。
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグのしきい値は ALMOST_EMPTY_OFFSET 属性で指定します。
WRERR、RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。
WRCOUNT、RDCOUNT	出力	13	FIFO 書き込み/読み出しポインタ

ポート名	方向	幅	機能
DI	入力	4、8、16、32	FIFO データ入力バス
DIP	入力	0、1、2、4	FIFO パリティ データ バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインタの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

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	不可
マクロのサポート	推奨

4 ビット幅でコンフィギュレーションされたプリミティブををインスタンス化するには、DIP ポートを論理値 0 にし、DOP ポートは未接続のままにします。DI[3:0] および DO[3:0] は適切な入力および出力信号に接続し、DI[31:4] は論理値 0 に、DO[31:4] は未接続のままにします。

9 ビット幅でコンフィギュレーションするときは、DIP[0] ポートを適切なデータ入力に接続し、DIP[3:1] は論理値 0 にします。DOP[0] は適切なデータ出力に接続し、DOP[3:1] は未接続のままにします。DI[7:0] および DO[7:0] は適切な入力および出力信号に接続し、DI[31:8] は論理値 0 に、DO[31:8] は未接続のままにします。

18 ビット幅でコンフィギュレーションするときは、DIP[1:0] ポートを適切なデータ入力に接続し、DIP[3:2] は論理値 0 にします。DOP[1:0] は適切なデータ出力に接続し、DOP[3:2] は未接続のままにします。DI[15:0] および DO[15:0] は適切な入力および出力信号に接続し、DI[31:16] は論理値 0 に、DO[31:16] は未接続のままにします。

36 ビット幅をコンフィギュレーションするときは、DI、DIP、DO、および DOP 信号すべてを接続する必要があります。

いずれのコンフィギュレーションでも、未使用の DI または DIP 入力は論理値 0 に、未使用の DO または DOP ピンは未接続のままにする必要があります。EN_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。

非同期モード (EN_SYN = FALSE) のときは、個別のクロック信号を使用します。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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 サイクル後にディassert 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ィassert レイテンシ サイクルのみが反映されます。WREN および RDEN は、対応するライト イネーブル およびリード イネーブル信号/ロジックに接続し、RST は適切なリセット信号/ロジックに接続するか、使用しない場合は論理値 0 にする必要があります。FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

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	13 ビット値	すべてゼロ	ALMOST_FULL フラグをトリガする RAM のデータ量を指定します。
ALMOST_EMPTY_OFFSET	16 進数	13 ビット値	すべてゼロ	ALMOST_EMPTY フラグをトリガする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。
DATA_WIDTH	整数	4 ~ 36	4	FIFO に必要なデータ幅を指定
EN_SYN	ブール代数	TRUE、FALSE	FALSE	FIFO が非同期 (独立した 2 つのクロック) または同期 (1 つのクロック) のいずれかで動作しているかを示します。マルチレートの場合は DO_REG=1 を設定する必要があります。
DO_REG	整数	0、1	1	EN_SYN のデータ パイプラインレジスタ
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。

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;

-- FIFO36: 32k+4k Parity Synchronous/Asynchronous BlockRAM FIFO BlockRAM Memory
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

FIFO36_inst : FIFO36
generic map (
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DATA_WIDTH => 4, -- Sets data width to 4, 9, 18, or 36
    DO_REG => 1, -- Enable output register ( 0 or 1)
    -- Must be 1 if the EN_SYN = FALSE
    EN_SYN => FALSE, -- Specified FIFO as Asynchronous (FALSE) or
    -- Synchronous (TRUE)
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DO => DO, -- 32-bit data output
    DOP => DOP, -- 4-bit parity data output
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 13-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 13-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 32-bit data input
    DIP => DIP, -- 4-bit parity input
    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 FIFO36_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// FIFO36: 32k+4k Parity Synchronous/Asynchronous BlockRAM FIFO
//      Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

FIFO36 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(13'h0080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(13'h0080), // Sets the almost empty threshold
    .DATA_WIDTH(4), // Sets data width to 4, 9, 18 or 36
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO36_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DO(DO), // 32-bit data output
    .DOP(DOP), // 4-bit parity data output
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 13-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .WRCOUNT(WRCOUNT), // 13-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 32-bit data input
    .DIP(DIP), // 4-bit parity input
    .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
);

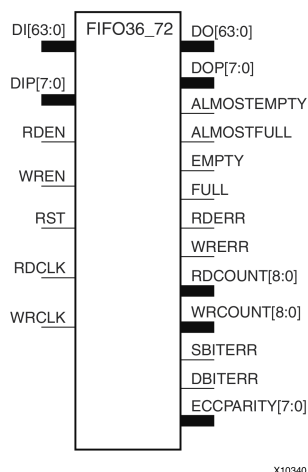
// End of FIFO36_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

FIFO36_72

プリミティブ : 72-Bit Wide by 512 Deep 36kb FIFO (First In, First Out) Block RAM Memory with ECC (Error Detection and Correction Circuitry)



概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。このエレメントを使用すると、幅広のデータパスが必要なときに 36Kb の FIFO ブロック RAM にアクセスできます。このコンポーネントは、72 ビット X 512 ワードの同期または非同期 FIFO RAM としてコンフィギュレーションできます。エラー検出と訂正回路をイネーブルにすると、メモリ破損を検出し、修正することもできます。また、この FIFO RAM では関連するすべての FIFO フラグおよびステータス信号も提供されます。

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 サイクル後にディ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ィアサートレイテンシ サイクルのみが反映されます。

メモ : ワード数が多くデータ幅が狭いコンフィギュレーションには、代わりに FIFO36 を使用し、ワード数が少なく、データ幅の狭いコンフィギュレーションには、FIFO18 または FIFO18_36 を使用します。

ポートの説明

ポート名	方向	幅	機能
DO	出力	64	FIFO データ出力バス
DOP	出力	8	FIFO パリティ データ出力バス
FULL	出力	1	FIFO がフルであることを示します。
ALMOSTFULL	出力	1	FIFO がほぼフルであることを示します。このフラグの値は ALMOST_FULL_OFFSET 属性で指定します。
EMPTY	出力	1	FIFO が空であることを示します。
ALMOSTEMPTY	出力	1	FIFO がほぼ空であることを示します。このフラグの値は ALMOST_EMPTY_OFFSET 属性で指定します。
WRERR, RDERR	出力	1	WRERR は FIFO がフルの間に書き込みが実行されたことを示し、RDERR は FIFO が空の間に読み出しが実行されたことを示します。

ポート名	方向	幅	機能
WRCOUNT、 RDCOUNT	出力	9	FIFO 書き込み/読み出しポインタ
SBITERR	出力	1	シングル ビット エラーが検出されたことを示す ECC ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
DBITERR	出力	1	ダブル ビット エラーが検出されたことを示す ECC ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
ECCPARITY	出力	8	メモリ エラー検出と訂正を行う ECC デコーダで使用される ECC エンコーダから生成された 8 ビット データ
DI	入力	64	FIFO データ入力バス
DIP	入力	8	FIFO パリティ データ入力バス
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
RST	入力	1	FIFO 機能、フラグ、ポインタの非同期リセット (アクティブ High)。RESET は 3 クロック サイクル間アサートする必要があります。
WRCLK、 RDCLK	入力	1	FIFO リード クロックおよびライト クロック (立ち上がりエッジで動作)

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	不可
マクロのサポート	推奨

DI、DIP、DO、および DOP ピンは、FIFO が ECC モードで動作していない場合は、対応する入力および出力データソースに接続する必要があります。ECC モードで動作している場合は、パリティビットが ECC の機能で必用なため、DI および DO ポートのみを使用する必要があります。使用可能なビット数以下を使用するときは、未使用の DI または DIP 入力を論理値 0 に、未使用の DO または DOP ピンは未接続のままにする必要があります。EN_SYM 属性を TRUE に設定して FIFO を同期に設定するときは、同じクロックソースを WRCLK および RDCLK に接続する必要があります。

非同期モード (EN_SYN = FALSE) のときは、個別のクロック信号を使用します。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、ALMOSTEMPTY フラグと ALMOSTFULL フラグが 1 サイクル後にディא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ィアサートレイテンシ サイクルのみが反映されます。WREN および RDEN は、対応するライト イネーブルおよびリード イネーブル信号/ロジックに接続し、RST は適切なりセット信号/ロジックに接続するか、使用しない場合は論理値 0 にする必要があります。

FULL、ALMOSTFULL、EMPTY、および ALMOSTEMPTY 出力フラグは、適切なデスティネーション ロジックに接続するか、使用しない場合は未接続のままにする必要があります。WRERR、RDERR、WRCOUNT、および RDCOUNT はオプションの出力で、使用しない場合は未接続のままにできます。ECC 機能を使用するには、EN_ECC_READ および EN_ECC_WRITE 属性を TRUE に設定する必要があります。エラー検出回路の動作を監視する場合は、SBITERR、DBITERR、および ECCPARITY 信号を適切なロジックに接続します。インスタンス化コードに含まれるジェネリック (VHDL) またはインライン defparam (Verilog) を変更することですべての属性を設定し、目的どおりに FIFO が動作するようにします。

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	9 ビット値	080	ALMOST_FULL フラグをトリガする RAM のデータ量を指定します。
ALMOST_EMPTY_OFFSET	16 進数	9 ビット値	080	ALMOST_EMPTY フラグをトリガする RAM のデータ量を指定します。
FIRST_WORD_FALL_THROUGH	ブール代数	TRUE、FALSE	FALSE	TRUE に設定すると、RDEN がアサートされずに FIFO に最初に書き込まれた値が DO に出力されます。
EN_SYN	ブール代数	TRUE、FALSE	FALSE	FALSE のときは非同期モード、TRUE のときは同期 (1 クロック) モードで FIFO が使用されることを示します。
DO_REG	整数	0, 1	1	読み出しレイテンシ (パイプライン遅延 1 つ) 追加することで clock-to-out のタイミングを向上するように、FIFO の出力レジスタをイネーブルします。EN_SYN が FALSE のときは DO_REG を 1 にする必要があります。
EN_ECC_READ	ブール代数	TRUE、FALSE	FALSE	ECC デコーダ回路をイネーブルにします。
EN_ECC_WRITE	ブール代数	TRUE、FALSE	FALSE	ECC エンコーダ回路をイネーブルにします。
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。

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;

-- FIFO36_72: 72x36k Synchronous/Asynchronous BlockRAM FIFO /w ECC
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

FIFO36_72_inst : FIFO36_72
generic map (
    ALMOST_FULL_OFFSET => X"0080", -- Sets almost full threshold
    ALMOST_EMPTY_OFFSET => X"0080", -- Sets the almost empty threshold
    DO_REG => 1, -- Enable output register (0 or 1)
    -- Must be 1 if EN_SYN = FALSE
    EN_ECC_READ => FALSE, -- Enable ECC decoder, TRUE or FALSE
    EN_ECC_WRITE => FALSE, -- Enable ECC encoder, TRUE or FALSE
    EN_SYN => FALSE, -- Specifies FIFO as Asynchronous (FALSE)
    -- or Synchronous (TRUE)
    FIRST_WORD_FALL_THROUGH => FALSE, -- Sets the FIFO FWFT to TRUE or FALSE
    SIM_MODE => "SAFE") -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
port map (
    ALMOSTEMPTY => ALMOSTEMPTY, -- 1-bit almost empty output flag
    ALMOSTFULL => ALMOSTFULL, -- 1-bit almost full output flag
    DBITERR => DBITERR -- 1-bit double bit error status output
    DO => DO, -- 64-bit data output
    DOP => DOP, -- 4-bit parity data output
    ECCPARITY => ECCPARITY -- 8-bit generated error correction parity
    EMPTY => EMPTY, -- 1-bit empty output flag
    FULL => FULL, -- 1-bit full output flag
    RDCOUNT => RDCOUNT, -- 9-bit read count output
    RDERR => RDERR, -- 1-bit read error output
    WRCOUNT => WRCOUNT, -- 9-bit write count output
    WRERR => WRERR, -- 1-bit write error
    DI => DI, -- 64-bit data input
    DIP => DIP, -- 4-bit parity input

```

```

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 FIFO36_72_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// FIFO36_72: 72x36k Synchronous/Asynchronous BlockRAM FIFO /w ECC
//           Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

FIFO36_72 #(
    .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
    .ALMOST_FULL_OFFSET(9'h080), // Sets almost full threshold
    .ALMOST_EMPTY_OFFSET(9'h080), // Sets the almost empty threshold
    .DO_REG(1), // Enable output register (0 or 1)
    // Must be 1 if EN_SYN = "FALSE"
    .EN_ECC_READ("FALSE"), // Enable ECC decoder, "TRUE" or "FALSE"
    .EN_ECC_WRITE("FALSE"), // Enable ECC encoder, "TRUE" or "FALSE"
    .EN_SYN("FALSE"), // Specifies FIFO as Asynchronous ("FALSE")
    // or Synchronous ("TRUE")
    .FIRST_WORD_FALL_THROUGH("FALSE") // Sets the FIFO FWFT to "TRUE" or "FALSE"
) FIFO36_72_inst (
    .ALMOSTEMPTY(ALMOSTEMPTY), // 1-bit almost empty output flag
    .ALMOSTFULL(ALMOSTFULL), // 1-bit almost full output flag
    .DBITERR(DBITERR), // 1-bit double bit error status output
    .DO(DO), // 64-bit data output
    .DOP(DOP), // 4-bit parity data output
    .ECCPARITY(ECCPARITY), // 8-bit generated error correction parity
    .EMPTY(EMPTY), // 1-bit empty output flag
    .FULL(FULL), // 1-bit full output flag
    .RDCOUNT(RDCOUNT), // 9-bit read count output
    .RDERR(RDERR), // 1-bit read error output
    .SBITERR(SBITERR), // 1-bit single bit error status output
    .WRCOUNT(WRCOUNT), // 9-bit write count output
    .WRERR(WRERR), // 1-bit write error
    .DI(DI), // 64-bit data input
    .DIP(DIP), // 4-bit parity input
    .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 FIFO36_72_inst instantiation

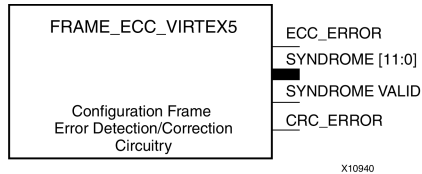
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

FRAME_ECC_VIRTEX5

プリミティブ : Virtex®-5 Configuration Frame Error Detection and Correction Circuitry



概要

このデザイン エLEMENTでは、FPGA のコンフィギュレーション メモリ向けの専用ビルトイン ECC (エラー検出および修正回路) が提供されます。このELEMENTには、ECC 回路のステータスおよびリードバック CRC 回路のステータスを監視する出力が含まれています。

ポートの説明

ポート名	方向	幅	機能
ECCERROR	出力	1	フレーム ECC エラーが見つかったことを示します。SYNDROME が 0 以外の場合は 1、SYNDROME がすべて 0 のときは 0 になり、エラーが検出されなかったことを示します。
SYNDROME	出力	12	フレーム ECC エラーの発生場所を示します。 <ul style="list-style-type: none"> エラーなし : すべてゼロ 1 ビット エラー : SYNDROME[11]=0、SYNDROME[10:0]= フレームでのエラーの位置が示されます。 2 ビット エラー : SYNDROME[11]=1、SYNDROME[10:0]= ドントケア 2 ビット以上のエラー : 出力が不明です。
SYNDROMEVALID	出力	1	SYNDROME の値が有効であることを示します。
CRCERROR	出力	1	リードバック CRC エラーを示します。

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

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;

-- FRAME_ECC_VIRTEX5: Configuration Frame Error Correction Circuitry
--                               Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

FRAME_ECC_VIRTEX5_inst : FRAME_ECC_VIRTEX5
port map (
    CRCERROR => CRCERROR,    -- 1-bit output indicating a CRC error
    ECCERROR => ECCERROR,    -- 1-bit output indicating an ECC error
    SYNDROME => SYNDROME,    -- 12-bit output location of erroneous bit
    SYNDROMEVALID => SYNDROMEVALID -- 1-bit output indicating the
                                -- SYNDROME output is valid
);

-- End of FRAME_ECC_VIRTEX5_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// FRAME_ECC_VIRTEX5: Configuration Frame Error Correction Circuitry
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

FRAME_ECC_VIRTEX5 FRAME_ECC_VIRTEX5_inst (
    .CRCERROR(CRCERROR), // 1-bit output indicating a CRC error
    .ECCERROR(ECCERROR), // 1-bit output indicating an ECC error
    .SYNDROME(SYNDROME), // 12-bit output location of erroneous bit
    .SYNDROMEVALID(SYNDROMEVALID) // 1-bit output indicating the
                                // SYNDROME output is valid
);

// End of FRAME_ECC_VIRTEX5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

GTP_DUAL

プリミティブ : Dual Gigabit Transceiver

概要

このエレメントは、Virtex®-5 FPGA 用の消費電力を抑えたトランシーバです。高度なコンフィギュレーションが可能で、FPGA のプログラマブル ロジック リソースに完全に統合されています。

デザインの入力方法

インスタンス化	不可
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-5 FPGA データシート : DC 特性およびスイッチ特性](#)
- ・ [Virtex-5 FPGA ユーザー ガイド](#)

GTX_DUAL

プリミティブ : Dual Gigabit Transceiver

概要

このELEMENTは、Virtex®-5 FPGA 用の消費電力を抑えたトランシーバです。高度なコンフィギュレーションが可能で、FPGA のプログラマブル ロジック リソースに完全に統合されています。

デザインの入力方法

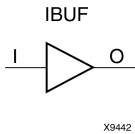
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

詳細情報

- ・ [Virtex-5 FPGA RocketIO GTX トランシーバ ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)
- ・ [Virtex-5 FPGA ユーザー ガイド](#)

IBUF

プリミティブ : Input Buffer



概要

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

ポートの説明

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

デザインの入力方法

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

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of IBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

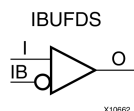
// End of IBUF_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
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 (インスタンスレーション)

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

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

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

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

-- End of IBUFDS_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

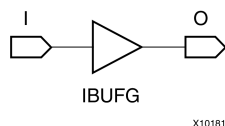
// End of IBUFDS_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
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;

-- 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 (インスタンス化)

```
// 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(0), // Clock buffer output
    .I(I) // Clock buffer input (connect directly to top-level port)
);

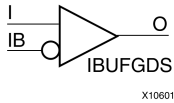
// End of IBUFG_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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 ポートをこの入力をソースとする DCM、BUFG、またはロジックに接続してください。一部の合成ツールでは、IBUFG を FPGA のクロック リソースに接続すると、必要に応じて BUFG が自動的に推論されます。generic/defparam 値を設定し、バッファのビヘイビアを適切に設定してください。

使用可能な属性

属性	タイプ	値	デフォルト	説明
DIFF_TERM	ブール 代数	TRUE、FALSE	FALSE	ビルトインの差動終端抵抗をイネーブル
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;

-- 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 (インスタンスエーション)

```
// 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
);

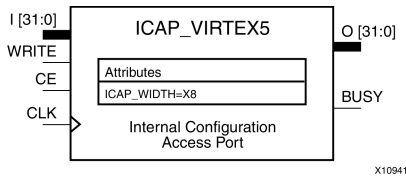
// End of IBUFGDS_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

ICAP_VIRTEX5

プリミティブ : Internal Configuration Access Port



概要

このデザイン エLEMENTからは、FPGA ファブリックから FPGA のコンフィギュレーション機能にアクセスできます。このコンポーネントを使用すると、FPGA アレイのコンフィギュレーション ロジックにコマンドおよびデータを書き込んだり、コンフィギュレーション ロジックからデータを読み出したりすることができます。このファンクションを正しく使用しないと FPGA の機能および信頼性に悪い影響を与えるため、この機能に精通していない限りこのELEMENTは使用しないでください。

ポートの説明

ポート名	方向	幅	機能
O	出力	32	コンフィギュレーション データ出力バス
Busy	出力	1	Busy/Ready 出力
I	入力	32	コンフィギュレーション データ入力バス
WRITE	入力	1	アクティブ Low のライト入力
CE	入力	1	アクティブ Low のイネーブル入力
CLK	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

パラレル バスのビット順の詳細は、『Virtex-5 FPGA コンフィギュレーション ユーザー ガイド』を参照してください。

使用可能な属性

属性	タイプ	値	デフォルト	説明
ICAP_WIDTH	文字列	X8, X16, X32	X8	ICAP_VIRTEX5 で使用する入力および出力データ幅を指定します。

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;

-- ICAP_VIRTEX5: Internal Configuration Access Port
--           Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

ICAP_VIRTEX5_inst : ICAP_VIRTEX5
generic map (
    ICAP_WIDTH => "X8") -- "X8", "X16" 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_VIRTEX5_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// ICAP_VIRTEX5: Internal Configuration Access Port
//           Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

ICAP_VIRTEX5 #(
    .ICAP_WIDTH("X8") // "X8", "X16" or "X32"
) ICAP_VIRTEX5_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
);

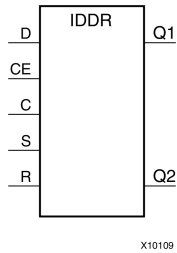
// End of ICAP_VIRTEX5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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。SRTYPE 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。SRTYPE 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
);

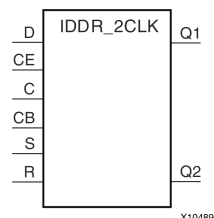
// End of IDDR_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

IDDR_2CLK

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



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。通常は、高速インターフェイスにのみこのプリミティブを使用します。このプリミティブでは、必要なクロック リソース数および消費電力が増加し、IDDR コンポーネントを使用するときには不要な配置配線での制限も発生するためです。代わりに IDDR コンポーネントを使用すると、動作速度は多少遅くなるものの、簡単に使用でき、リソース数も抑えられ、制限も低減します。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 はクロック CB の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、各クロックの立ち上がりエッジで受信されますが、CB クロック データレジスタの前にレジスタが 1 つ追加されます。この追加されたレジスタにもクロック信号 C の立ち上がりエッジでクロックが供給されるため、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、C のクロック 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。

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	立ち上がりエッジのデータをキャプチャするプライマリ クロック入力ピン
CB	入力	1	立ち下がりエッジのデータをキャプチャするセカンダリ クロック入力ピン (通常プライマリ クロックと 180 度位相がずれています)
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。SRTYPE 属性に基づき、同期または非同期に設定できます。

ポート名	方向	幅	機能
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。SRTYPE 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

- ・ C ピンを立ち上がりクロック エッジを示すクロック ソースへ、CB ピンを立ち下がりクロック エッジを示すクロック ソースへ接続します。
- ・ D ピンは最上位の入力または双方向ポート、IODELAY、またはインスタンス化された入力または双方向バッファへ接続します。
- ・ Q1 および Q2 ピンは、適切なデータ ソースへ接続する必要があります。
- ・ CE ピンは、未使用のときに High にするか、または適切なクロック イネーブル ロジックに接続します。
- ・ R および S ピンは未使用のときに Low にするか、または適切なセット/リセット生成ロジックに接続します。
- ・ 目的の動作になるように、コンポーネントに属性を設定します。
- ・ このペアのコンポーネントは同じクロックを使用してインスタンス化し、使用可能な I/O リソースを無駄にしないように、I/O ペアの P および N に LOC 制約を使用して固定します。
- ・ このコンポーネントは常にほかの I/O コンポーネントと共にコードの最上位の階層にインスタンス化します。これにより、階層デザイン フローが促進されます。
- ・ CLK スキューを最小限に抑えるには、CLK および CLKB の両方が、ローカル反転ではなく、グローバル配線 (DCM / MMCM) から来るようにする必要があります。ローカル反転はクロックにスキューを追加しますが、DCM / PLL はスキューを抑えます。

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	DDR のデータ送信モードを選択します。詳細は、「概要」を参照してください。
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;

-- IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
--           Set, Reset and Clock Enable.
--           Virtex-5/6
-- Xilinx HDL Libraries Guide, version 11.2

IDDR_2CLK_inst : IDDR_2CLK
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 "ASYN"
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 primary clock input
    CB => CB, -- 1-bit secondary 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_2CLK_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// IDDR_2CLK: Dual-Clock, Input Double Data Rate Input Register with
//           Set, Reset and Clock Enable.
//           Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

IDDR_2CLK #(
    .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 "ASYN"
) IDDR_2CLK_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 primay clock input
    .CB(CB), // 1-bit secondary 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
);

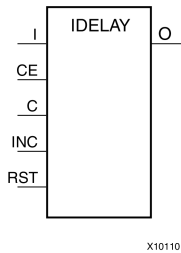
// End of IDDR_2CLK_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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、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化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化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化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は受けません。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

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
);

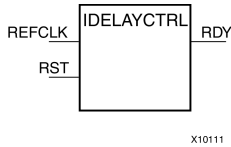
// End of IDELAY_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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;

-- 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 (インスタンスレーション)

```
// 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
);

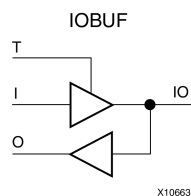
// End of IDELAYCTRL_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
DRIVE	整数	2、4、6、8、12、16、24	12	I/O 規格として LVTTL、LVCMOS12、LVCMOS15、LVCMOS18、LVCMOS25 または LVCMOS33 を使用する SelectIO™ バッファの出力の駆動電流 (mA) を選択
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;

-- 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 (インスタンスレーション)

```
// 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
);

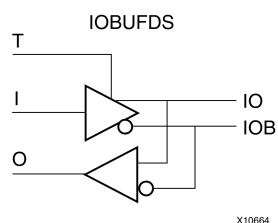
// End of IOBUF_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
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;

-- 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 (インスタンス化)

```
// 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
);

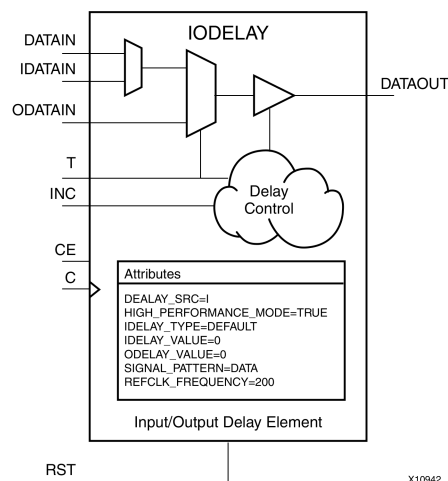
// End of IOBUFDS_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

IODELAY

プリミティブ : Input and Output Fixed or Variable Delay Element



概要

このデザイン エLEMENTは、FPGA の入力パスに固定遅延または可変遅延を、出力パスに固定遅延を追加します。この遅延は、チップへ入力されるデータ、またはチップから出力されるデータをアライメントするのに有効で、プロセス、温度、電圧の影響を受けずにデータ アライメントを監視できます。IODELAY は FPGA の I/O すべてに使用でき、IDELAYCTRL 回路と共に使用すると、正確に増分された遅延を追加できます。VARIABLE モードでは、入力パスを調整して遅延の量を増分または減分できます。出力遅延パスは、固定遅延でのみ使用できます。IODELAY は、FPGA の内部パスに一定の遅延または可変遅延を追加する場合にも使用できます。ただし、このように使用する場合は、入力パス遅延または出力パス遅延に関連する I/O に使用できなくなります。

ポートの説明

ポート名	方向	幅	機能
DATAOUT	出力	1	入力ポートで遅延されたデータ出力 (入力データ パス ロジックに接続)
IDATAIN	入力	1	I/O からのデータ入力 (ポートの I/O バッファに直接接続)。IDATAIN を使用する場合、DATAIN は論理値 0 (グラウンドに接続) にする必要があります。
ODATAIN	入力	1	出力データ パスのデータ入力 (出力データ ソースに接続)。ODATAIN を使用する場合、DATAIN は論理値 0 (グラウンドに接続) にする必要があります。
DATAIN	入力	1	内部データ パス遅延のデータ入力。DATAIN を使用する場合、IDATAIN および ODATAIN は論理値 0 (グラウンドに接続) にする必要があります。
T	入力	1	トライステート入力制御ピン。入力のみまたは内部遅延の場合は High に、出力のみの場合は Low にします。
CE	入力	1	アクティブ High になるとインクリメント/デクリメントを有効にします。
INC	入力	1	インクリメント/デクリメント タップ遅延
C	入力	1	クロック入力 (必ず VARIABLE モードに設定)
RST	入力	1	アクティブ High の同期リセット。遅延チェーンを IDELAY_VALUE/ODELAY_VALUE タップにリセットします。値を指定しない場合は、デフォルトは 0 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

入力遅延処理の場合は、最上位の I/O ポート、入力バッファ、または I/O バッファのいずれかに直接 IDATAIN ピンを接続します。出力遅延処理の場合は、遅延させる出力データを供給するロジックに ODATAIN 入力を接続します。内部パス遅延処理の場合は、FPGA 内の適切なソースおよびデスティネーション ロジックに DATAIN ピンを接続します。IODELAY を内部信号遅延に使用する場合は、IDATAIN および ODATAIN の論理値を 0 (グラウンドに接続) にする必要があります。

どの場合でも、DATAOUT は遅延させたデータが供給される I/O またはロジックに接続する必要があります。IODELAY を使用する場合、トライステート出力にするために T ピンを制御信号に接続します。出力遅延のみに IODELAY を使用する場合は、T ピンを論理値 0 (グラウンドに接続) にします。入力のみ IODELAY を使用する場合、または内部信号を遅延させる場合は、T ピンを論理値 1 (VCC に接続) にします。IODELAY を VARIABLE モードに設定する場合は、CE ピン、INC ピン、C ピン、および RST ピンを適切な遅延制御信号に接続します。FIXED モードでは、これらのピンは論理値 0 (グラウンドに接続) にする必要があります。

使用可能な属性

属性	タイプ	値	デフォルト	説明
HIGH_PERFORMANCE_MODE	ブール代数	TRUE、FALSE	FALSE	TRUE の場合は、出力ジッタを減少させます。
DELAY_SRC	文字列	I、O、IO、DATAIN	I	IODELAY コンポーネントのソースを指定します。I の場合は入力ポート (IBUF、入力モード) に直接接続し、O の場合は出力ポート (OBUF、出力モード)、IO の場合はポートに接続し、DATAIN の場合はポートに接続しません (内部モード)。
IDELAY_TYPE	文字列	DEFAULT、FIXED、VARIABLE	DEFAULT	入力遅延タイプに DEFAULT (ホールドタイムの削除)、FIXED、VARIABLE のいずれかを指定
IDELAY_VALUE	整数	0 ～ 63	0	FIXED モードでは入力パス遅延のタップ数、VARIABLE モードでは初期遅延タップ値を指定
ODELAY_VALUE	整数	0 ～ 63	0	出力パス遅延のタップ数を指定
REFCLK_FREQUENCY	実数	190.00 ～ 210.00	200.00	IDELAYCTRL を使用する場合は、IODELAY の入力リファレンス周波数を指定
SIGNAL_PATTERN	文字列	CLOCK、DATA	DATA	ディレイ カリキュレータで使用され、設定に基づいて IODELAY ブロックを介した伝搬遅延が決定されます。DATA を指定すると、タップあたりの遅延およびジッタが追加されます。クロック同様の信号では、ジッタは増加しません。

VHDL 記述 (インスタンス化)

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

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

-- IODELAY: Input and/or Output Fixed/Variable Delay Element
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

IODELAY_inst : IODELAY
generic map (
    DELAY_SRC => "I", -- Specify which input port to be used
                    -- "I"=IDATAIN, "O"=ODATAIN, "DATAIN"=DATAIN, "IO"=Bi-directional
    HIGH_PERFORMANCE_MODE => TRUE, -- TRUE specifies lower jitter
                                -- at expense of more power
    IDELAY_TYPE => "FIXED", -- "FIXED" or "VARIABLE"
    IDELAY_VALUE => 0, -- 0 to 63 tap values
    ODELAY_VALUE => 0, -- 0 to 63 tap values
    REFCLK_FREQUENCY => 200.0, -- Frequency used for IDELAYCTRL
                        -- 175.0 to 225.0
    SIGNAL_PATTERN => "DATA") -- Input signal type, "CLOCK" or "DATA"
port map (
    DATAOUT => DATAOUT, -- 1-bit delayed data output
    C => C, -- 1-bit clock input
    CE => CE, -- 1-bit clock enable input
    DATAIN => DATAIN, -- 1-bit internal data input
    IDATAIN => IDATAIN, -- 1-bit input data input (connect to port)
    INC => INC, -- 1-bit increment/decrement input
    ODATAIN => ODATAIN, -- 1-bit output data input
    RST => RST, -- 1-bit active high, synch reset input
    T => T -- 1-bit 3-state control input
);

-- End of IODELAY_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// IODELAY: Input and/or Output Fixed/variable Delay Element
//      Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

IODELAY # (
    .DELAY_SRC("I"), // Specify which input port to be used, "I"=IDATAIN,
                    // "O"=ODATAIN, "DATAIN"=DATAIN, "IO"=Bi-directional
    .HIGH_PERFORMANCE_MODE("TRUE"), // "TRUE" specifies lower jitter
                                // at expense of more power
    .IDELAY_TYPE("FIXED"), // "FIXED" or "VARIABLE"
    .IDELAY_VALUE(0), // 0 to 63 tap values
    .ODELAY_VALUE(0), // 0 to 63 tap values
    .REFCLK_FREQUENCY(200.0), // Frequency used for IDELAYCTRL
                        // 175.0 to 225.0
    .SIGNAL_PATTERN("DATA") // Input signal type, "CLOCK" or "DATA"
) IODELAY_INST (
    .DATAOUT(DATAOUT), // 1-bit delayed data output
    .C(C), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .DATAIN(DATAIN), // 1-bit internal data input
    .IDATAIN(IDATAIN), // 1-bit input data input (connect to port)
    .INC(INC), // 1-bit increment/decrement input
    .ODATAIN(ODATAIN), // 1-bit output data input
    .RST(RST), // 1-bit active high, synch reset input
    .T(T) // 1-bit 3-state control input
);

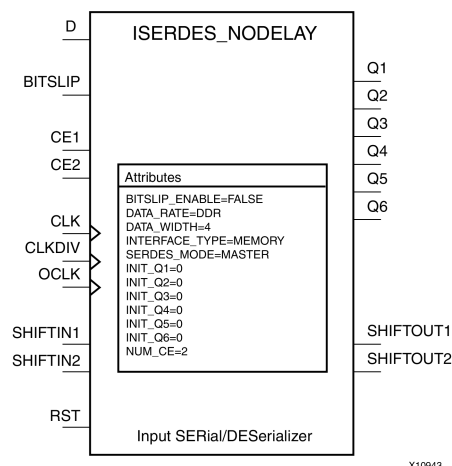
// End of IODELAY_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

ISERDES_NODELAY

プリミティブ : Input SERial/DESerializer



X10943

概要

ISERDES_NODELAY は、シリアル/パラレル コンバータで、高速のソース同期シリアル データを取り込みます。ISERDES_NODELAY には、クロック管理およびデータ アライメントを行うロジックが含まれており、シングル データ レート (SDR) またはダブル データ レート (DDR) のデータを、単一のインスタンス (MASTER) では 2 ～ 6 ビットのデータ幅に、カスケード接続された 2 つの ISERDES_NODELAY (MASTER/SLAVE) では、7 ～ 10 ビットのデータ幅にできます。ISERDES_NODELAY はメモリに使用でき、さまざまなタイプ of データ インターフェイス アプリケーションに接続できます。ISERDES_NODELAY は IODELAY と共に使用して、入力シリアル データのアライメントを実行できます。DDR モードでは、データを取り込むために ISERDES_NODELAY に 1 クロックまたは 2 クロックでクロックを供給できます。2 クロック モードを使用すると、パフォーマンスが向上する場合がありますが、必要なクロック リソース数および消費電力が増加し、配置配線での制限も発生する可能性があります。高速の I/O パフォーマンスが不要な場合は、1 クロック モードを使用してください。

ポートの説明

ポート名	方向	幅	機能
Q1 ～ Q6	出力	1	レジスタ付きパラレル入力データ
SHIFTOUT1/SHIFTOUT2	出力	1	ISERDES_MODE が MASTER で、2 つの ISERDES_NODELAY をカスケード接続する場合は、スレーブの SHIFTIN1 と SHIFTIN2 入力に接続します。
D	入力	1	追加の入力遅延が必要な場合に、デザイン最上位の入力ポート、I/O ポート、または IODELAY に直接接続する入力データ
BITSLIP	入力	1	入力データ BITSLIP イネーブル
CE1 / CE2	入力	1	入力データレジスタのクロック イネーブル
CLK	入力	1	プライマリ クロック入力
CLKB	入力	1	OSERDES の入力でのビット順序は、ISERDES_NODELAY ブロックの出力でのものとは逆になります。詳細は、該当デバイスのユーザーガイドを参照してください。
CLKDIV	入力	1	パラレル データに使用する分周クロック

ポート名	方向	幅	機能
OCLK	入力	1	通常メモリ インターフェイスに使用される高速の出力クロック
SHIFTIN1 / SHIFTIN2	入力	1	ISERDES_MODE が SLAVE の場合は、マスタの SHIFTOUT1 と SHIFTOUT2 出力に接続します。このピンはグラウンドに接続する必要があります。
RST	入力	1	SERDES のレジスタのアクティブ High の非同期リセット

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	TRUE、FALSE	FALSE	BITSLIP をイネーブルにします。 NETWORKING モードでのみ使用できます。
DATA_RATE	文字列	SDR、DDR	DDR	シングル データ レートまたはダブル データ レートを指定
DATA_WIDTH	整数	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	パラレル データ幅を選択
INTERFACE_TYPE	文字列	MEMORY、 NETWORKING	MEMORY	メモリ インターフェイスまたはネットワーク インターフェイスを指定
SERDES_MODE	文字列	MASTER、 SLAVE	MASTER	カスケード接続してデータ幅を拡張する場合に ISERDES をマスタ モードかスレーブ モードに設定するかどうかを指定
NUM_CE	整数	1、2	2	ISERDES_NODELAY に使用するクロック 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_NODELAY: Input Serial / DESerializer
--               Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

ISERDES_NODELAY_inst : ISERDES_NODELAY
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 -
                        -- NETWORKING SDR: 2, 3, 4, 5, 6, 7, 8 : DDR 4, 6, 8, 10
                        -- MEMORY SDR N/A : DDR 4
    INTERFACE_TYPE => "MEMORY", -- Use model - "MEMORY" or "NETWORKING"
    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 (
  Q1 => Q1, -- 1-bit registered SERDES output
  Q2 => Q2, -- 1-bit registered SERDES output
  Q3 => Q3, -- 1-bit registered SERDES output
  Q4 => Q4, -- 1-bit registered SERDES output
  Q5 => Q5, -- 1-bit registered SERDES output
  Q6 => Q6, -- 1-bit registered SERDES output
  SHIFTOUT1 => SHIFTOUT1, -- 1-bit cascade Master/Slave output
  SHIFTOUT2 => SHIFTOUT2, -- 1-bit cascade Master/Slave output
  BITSLIP => BITSLIP, -- 1-bit Bitslip enable input
  CE1 => CE1, -- 1-bit clock enable input
  CE2 => CE2, -- 1-bit clock enable input
  CLK => CLK, -- 1-bit master clock input
  CLKB => CLKB, -- 1-bit secondary clock input for DATA_RATE=DDR
  CLKDIV => CLKDIV, -- 1-bit divided clock input
  D => D, -- 1-bit data input, connects to IODELAY or input buffer
  OCLK => OCLK, -- 1-bit fast output clock input
  RST => RST, -- 1-bit asynchronous reset input
  SHIF TIN1 => SHIF TIN1, -- 1-bit cascade Master/Slave input
  SHIF TIN2 => SHIF TIN2 -- 1-bit cascade Master/Slave input
);

-- End of ISERDES_NODELAY_inst instantiation

```

Verilog 記述 (インスタンスエーション)

```

// ISERDES_NODELAY: Input Serial / DESerializer
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

ISERDES_NODELAY #(
  .BITSLIP_ENABLE("FALSE"), // "TRUE"/"FALSE" to enable bitslip 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 -
                             // NETWORKING SDR: 2, 3, 4, 5, 6, 7, 8 : DDR 4, 6, 8, 10
                             // MEMORY SDR N/A : DDR 4
  .INTERFACE_TYPE("MEMORY"), // Use model - "MEMORY" or "NETWORKING"
  .NUM_CE(2),                // Number of clock enables used, 1 or 2
  .SERDES_MODE("MASTER")    // Set SERDES mode to "MASTER" or "SLAVE"
) ISERDES_NODELAY_inst (
  .Q1(Q1), // 1-bit registered SERDES output
  .Q2(Q2), // 1-bit registered SERDES output
  .Q3(Q3), // 1-bit registered SERDES output
  .Q4(Q4), // 1-bit registered SERDES output
  .Q5(Q5), // 1-bit registered SERDES output
  .Q6(Q6), // 1-bit registered SERDES output
  .SHIFTOUT1(SHIFTOUT1), // 1-bit cascade Master/Slave output
  .SHIFTOUT2(SHIFTOUT2), // 1-bit cascade Master/Slave output
  .BITSLIP(BITSLIP), // 1-bit Bitslip enable input
  .CE1(CE1), // 1-bit clock enable input
  .CE2(CE2), // 1-bit clock enable input
  .CLK(CLK), // 1-bit master clock input
  .CLKB(CLKB), // 1-bit secondary clock input for DATA_RATE=DDR
  .CLKDIV(CLKDIV), // 1-bit divided clock input
  .D(D), // 1-bit data input, connects to IODELAY or input buffer
  .OCLK(OCLK), // 1-bit fast output clock input
  .RST(RST), // 1-bit asynchronous reset input
  .SHIF TIN1(SHIF TIN1), // 1-bit cascade Master/Slave input
  .SHIF TIN2(SHIF TIN2) // 1-bit cascade Master/Slave input
);

// End of ISERDES_NODELAY_inst instantiation

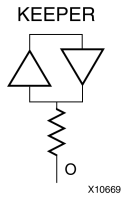
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 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)
);

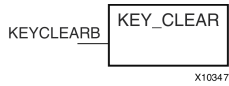
// End of KEEPER_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

KEY_CLEAR

プリミティブ : Virtex-5 Configuration Encryption Key Erase



概要

このデザイン エLEMENTでは、内部ロジックからコンフィギュレーション暗号回路キー レジスタの内容を消去できます。

ポートの説明

ポート名	方向	幅	機能
KEYCLEARB	入力	1	アクティブ Low の入力で、コンフィギュレーション暗号キーを消去します。

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

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;

-- KEY_CLEAR: Startup primitive for GSR, GTS or startup sequence control
--             Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

KEY_CLEAR_inst : KEY_CLEAR
  port map (
    KEYCLEARB => KEYCLEARB -- Active low key reset 1-bit input
  );

-- End of KEY_CLEAR_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// KEY_CLEAR: Startup primitive for GSR, GTS or startup sequence control
//             Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

KEY_CLEAR KEY_CLEAR_inst (
  .KEYCLEARB(KEYCLEARB) // Active low key reset 1-bit input
);

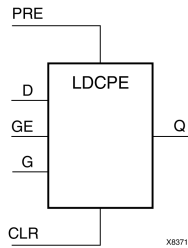
// End of KEY_CLEAR_inst instantiation
```


詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

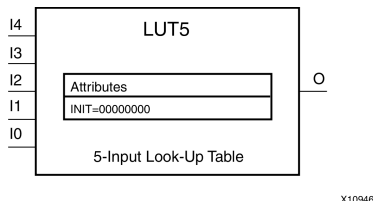
// End of LDCPE_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

LUT5

プリミティブ : 5-Input Lookup Table with General Output



概要

このデザイン エLEMENTは、入力 5 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) として動作するか、5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。LUT5、LUT5_L、および LUT5_D の機能は同じですが、LUT5_L および LUT5_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 32 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。たとえば Verilog で INIT 値が 32'h8000000000000000 (VHDL では X"8000000000000000") の場合、入力すべてが 1 ではない限り、出力が 0 になります (5 入力の AND ゲート)。Verilog で 32'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") の場合、入力がすべてゼロではない限り、出力は 1 になります (5 入力 OR ゲート)。

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
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値					

ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 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	32 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

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;

-- LUT5: 5-input Look-Up Table with general output
--      Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT5_inst : LUT5
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4 -- LUT input
);

-- End of LUT5_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT5: 5-input Look-Up Table with general output
//      Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2
```

```
LUT5 #(
    .INIT(32'h00000000) // Specify LUT Contents
) LUT5_inst (
    .O(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4) // LUT input
);

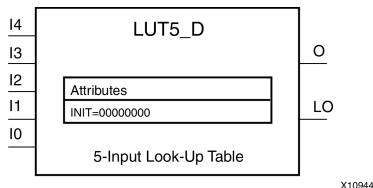
// End of LUT5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT5_D

プリミティブ : 5-Input Lookup Table with General and Local Outputs



概要

このデザイン エLEMENTは、入力 5 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) として動作するか、5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。LUT5、LUT5_L、および LUT5_D の機能は同じですが、LUT5_L および LUT5_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 32 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力が適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。たとえば Verilog で INIT 値が 32'h8000000000000000 (VHDL では X"8000000000000000") の場合は、入力すべてが 1 ではない限り、出力が 0 になります (5 入力の AND ゲート)。また、Verilog で INIT 値が 32'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") の場合は、入力がすべてゼロではない限り、出力は 1 になります (5 入力 OR ゲート)。

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	
I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	1	1	INIT[3]	INIT[3]
0	0	1	0	0	INIT[4]	INIT[4]
0	0	1	0	1	INIT[5]	INIT[5]
0	0	1	1	0	INIT[6]	INIT[6]
0	0	1	1	1	INIT[7]	INIT[7]
0	1	0	0	0	INIT[8]	INIT[8]
0	1	0	0	1	INIT[9]	INIT[9]
0	1	0	1	0	INIT[10]	INIT[10]
0	1	0	1	1	INIT[11]	INIT[11]
0	1	1	0	0	INIT[12]	INIT[12]
0	1	1	0	1	INIT[13]	INIT[13]
0	1	1	1	0	INIT[14]	INIT[14]
0	1	1	1	1	INIT[15]	INIT[15]
1	0	0	0	0	INIT[16]	INIT[16]
1	0	0	0	1	INIT[17]	INIT[17]
1	0	0	1	0	INIT[18]	INIT[18]
1	0	0	1	1	INIT[19]	INIT[19]
1	0	1	0	0	INIT[20]	INIT[20]
1	0	1	0	1	INIT[21]	INIT[21]
1	0	1	1	0	INIT[22]	INIT[22]
1	0	1	1	1	INIT[23]	INIT[23]
1	1	0	0	0	INIT[24]	INIT[24]
1	1	0	0	1	INIT[25]	INIT[25]
1	1	0	1	0	INIT[26]	INIT[26]
1	1	0	1	1	INIT[27]	INIT[27]
1	1	1	0	0	INIT[28]	INIT[28]
1	1	1	0	1	INIT[29]	INIT[29]
1	1	1	1	0	INIT[30]	INIT[30]
1	1	1	1	1	INIT[31]	INIT[31]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
L0	出力	1	内部 CLB 接続用の 5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 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	32 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

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;

-- LUT5: 5-input Look-Up Table with general output
--       Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT5_inst : LUT5
generic map (
  INIT => X"00000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4 -- LUT input
);

-- End of LUT5_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// LUT5: 5-input Look-Up Table with general output
//      Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

LUT5 #(
    .INIT(32'h00000000) // Specify LUT Contents
) LUT5_inst (
    .O(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4) // LUT input
);

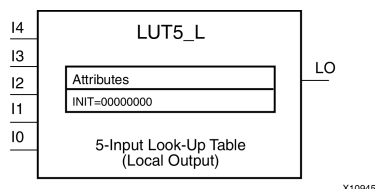
// End of LUT5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT5_L

プリミティブ : 5-Input Lookup Table with Local Output



概要

このデザイン エLEMENTは、入力 5 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) として動作するか、5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。LUT5、LUT5_L、および LUT5_D の機能は同じですが、LUT5_L および LUT5_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗黙的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 32 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。たとえば Verilog で INIT 値が 32'h8000000000000000 (VHDL では X"8000000000000000") の場合、入力すべてが 1 ではない限り、出力が 0 になります (5 入力の AND ゲート)。Verilog で 32'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") の場合、入力がすべてゼロではない限り、出力は 1 になります (5 入力 OR ゲート)。

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。バイナリの真理値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 : リストされた論理表の値に対応する 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
I4	I3	I2	I1	I0	LO
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値					

ポートの説明

ポート名	方向	幅	機能
L0	出力	1	内部 CLB 接続用の 6/5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 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	32 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

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;

-- LUT5: 5-input Look-Up Table with general output
--       Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT5_inst : LUT5
generic map (
  INIT => X"00000000" -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4 -- LUT input
);

-- End of LUT5_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT5: 5-input Look-Up Table with general output
//      Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

LUT5 #(
    .INIT(32'h00000000) // Specify LUT Contents
) LUT5_inst (
    .O(0), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4) // LUT input
);

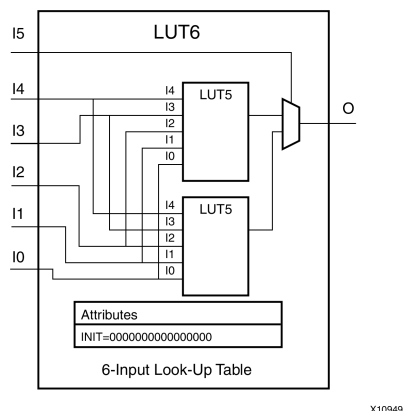
// End of LUT5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT6

プリミティブ : 6-Input Lookup Table with General Output



概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) として動作するか、6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT6 はルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。LUT6、LUT6_L、および LUT6_D の機能は同じですが、LUT6_L および LUT6_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT6_L では LUT6 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT6_D では LUT6 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 64 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力が適用されるときに、対応する INIT ビット値に 1 を割り当てることで算出されます。たとえば Verilog で INIT 値が `64'h8000000000000000` (VHDL では `X"8000000000000000"`) の場合は、入力すべてが 1 ではない限り、出力が 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値が `64'hffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) の場合は、入力がすべてゼロではない限り、出力は 1 になります (6 入力 OR ゲート)。

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
I5	I4	I3	I2	I1	I0	O
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]

入力						出力
I5	I4	I3	I2	I1	I0	O
0	0	0	0	1	0	INIT[2]
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]

入力						出力
I5	I4	I3	I2	I1	I0	O
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

ポートの説明

ポート名	方向	幅	機能
O	出力	1	6/5 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 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	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

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;

-- LUT6: 6-input Look-Up Table with general output
--       Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT6_inst : LUT6
generic map (
    INIT => X"0000000000000000") -- Specify LUT Contents
port map (
    O => O, -- LUT general output
    I0 => I0, -- LUT input
    I1 => I1, -- LUT input
    I2 => I2, -- LUT input
    I3 => I3, -- LUT input
    I4 => I4, -- LUT input
    I5 => I5 -- LUT input
);

-- End of LUT6_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT6: 6-input Look-Up Table with general output
//       Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

LUT6 #(
    .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_inst (
    .O(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4), // LUT input
    .I5(I5) // LUT input
);

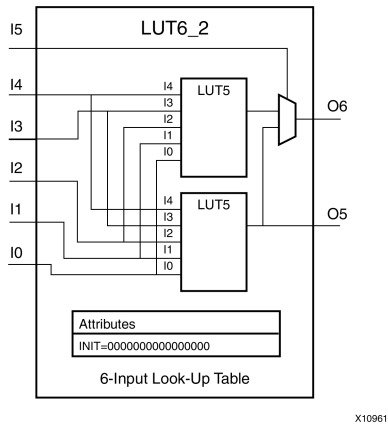
// End of LUT6_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

LUT6_2

プリミティブ : Six-input, 2-output, Look-Up Table



概要

このデザイン エLEMENTは、入力 6 個、出力 2 個のルックアップ テーブル (LUT) で、非同期 32 ビット デュアル ROM (5 ビットのアドレス指定) として動作するか、共有入力の付き 5 入力のロジック ファンクションのいずれか 2 つをインプリメントできます。または、共有入力とロジック値の付いた 6 入力および 5 入力のロジック ファンクションをインプリメントすることもできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT6_2 は、Virtex-5 スライスに含まれるルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。

LUT のロジック ファンクションを指定するために 64 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力 that 適用されるときに、対応する INIT ビット値に 1 を割り当てることで算出されます。たとえば、Verilog で INIT 値が 64'h00000000 (VHDL では X"FFFFFFFFFFFFFFFE") の場合は、入力がすべてゼロではない限り、O6 出力は 1 になり、I[4:0] がすべてゼロでない限り O5 出力は 1 になります (5 または 6 入力の OR ゲート)。INIT 値の下半分 (ビット 31:0) は O5 出力のロジック ファンクションに適用されます。

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	
I5	I4	I3	I2	I1	I0	O5	O6
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]

入力						出力	
0	0	0	1	0	0	INIT[4]	INIT[4]
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[0]	INIT[32]
1	0	0	0	0	1	INIT[1]	INIT[33]
1	0	0	0	1	0	INIT[2]	INIT[34]
1	0	0	0	1	1	INIT[3]	INIT[35]
1	0	0	1	0	0	INIT[4]	INIT[36]
1	0	0	1	0	1	INIT[5]	INIT[37]
1	0	0	1	1	0	INIT[6]	INIT[38]
1	0	0	1	1	1	INIT[7]	INIT[39]
1	0	1	0	0	0	INIT[8]	INIT[40]

入力						出力	
1	0	1	0	0	1	INIT[9]	INIT[41]
1	0	1	0	1	0	INIT[10]	INIT[42]
1	0	1	0	1	1	INIT[11]	INIT[43]
1	0	1	1	0	0	INIT[12]	INIT[44]
1	0	1	1	0	1	INIT[13]	INIT[45]
1	0	1	1	1	0	INIT[14]	INIT[46]
1	0	1	1	1	1	INIT[15]	INIT[47]
1	1	0	0	0	0	INIT[16]	INIT[48]
1	1	0	0	0	1	INIT[17]	INIT[49]
1	1	0	0	1	0	INIT[18]	INIT[50]
1	1	0	0	1	1	INIT[19]	INIT[51]
1	1	0	1	0	0	INIT[20]	INIT[52]
1	1	0	1	0	1	INIT[21]	INIT[53]
1	1	0	1	1	0	INIT[22]	INIT[54]
1	1	0	1	1	1	INIT[23]	INIT[55]
1	1	1	0	0	0	INIT[24]	INIT[56]
1	1	1	0	0	1	INIT[25]	INIT[57]
1	1	1	0	1	0	INIT[26]	INIT[58]
1	1	1	0	1	1	INIT[27]	INIT[59]
1	1	1	1	0	0	INIT[28]	INIT[60]
1	1	1	1	0	1	INIT[29]	INIT[61]
1	1	1	1	1	0	INIT[30]	INIT[62]
1	1	1	1	1	1	INIT[31]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

デザインの入力方法

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

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	LUT5/6 の出力ファンクションを指定

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;

-- LUT6_2: 6-input 2 output Look-Up Table
--          Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT6_2_inst : LUT6_2
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O6 => O6, -- 6/5-LUT output (1-bit)
  O5 => O5, -- 5-LUT output (1-bit)
  I0 => I0, -- LUT input (1-bit)
  I1 => I1, -- LUT input (1-bit)
  I2 => I2, -- LUT input (1-bit)
  I3 => I3, -- LUT input (1-bit)
  I4 => I4, -- LUT input (1-bit)
  I5 => I5  -- LUT input (1-bit)
);

-- End of LUT6_2_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT6_2: 6-input, 2 output Look-Up Table
//          Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

LUT6_2 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_2_inst (
  .O6(O6), // 6/5-LUT output (1-bit)
  .O5(O5), // 5-LUT output (1-bit)
  .I0(I0), // LUT input (1-bit)
  .I1(I1), // LUT input (1-bit)
  .I2(I2), // LUT input (1-bit)
  .I3(I3), // LUT input (1-bit)
  .I4(I4), // LUT input (1-bit)
  .I5(I5)  // LUT input (1-bit)
);

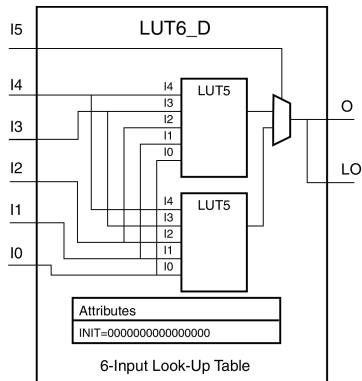
// End of LUT6_2_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT6_D

プリミティブ : 6-Input Lookup Table with General and Local Outputs



概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) として動作するか、6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT6 はルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。LUT6、LUT6_L、および LUT6_D の機能は同じですが、LUT6_L および LUT6_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT6_L では LUT6 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT6_D では LUT6 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 64 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力が適用されるときに、対応する INIT ビット値に 1 を割り当てることで算出されます。たとえば Verilog で INIT 値が 64'h8000000000000000 (VHDL では X"8000000000000000") の場合は、入力すべてが 1 ではない限り、出力が 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値が 64'hffffffff (VHDL では X"FFFFFFFFFFFFFFFF") の場合は、入力がすべてゼロではない限り、出力は 1 になります (6 入力 OR ゲート)。

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	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[32]	INIT[32]
1	0	0	0	0	1	INIT[33]	INIT[33]
1	0	0	0	1	0	INIT[34]	INIT[34]
1	0	0	0	1	1	INIT[35]	INIT[35]
1	0	0	1	0	0	INIT[36]	INIT[36]
1	0	0	1	0	1	INIT[37]	INIT[37]

入力						出力	
I5	I4	I3	I2	I1	I0	O	LO
1	0	0	1	1	0	INIT[38]	INIT[38]
1	0	0	1	1	1	INIT[39]	INIT[39]
1	0	1	0	0	0	INIT[40]	INIT[40]
1	0	1	0	0	1	INIT[41]	INIT[41]
1	0	1	0	1	0	INIT[42]	INIT[42]
1	0	1	0	1	1	INIT[43]	INIT[43]
1	0	1	1	0	0	INIT[44]	INIT[44]
1	0	1	1	0	1	INIT[45]	INIT[45]
1	0	1	1	1	0	INIT[46]	INIT[46]
1	0	1	1	1	1	INIT[47]	INIT[47]
1	1	0	0	0	0	INIT[48]	INIT[48]
1	1	0	0	0	1	INIT[49]	INIT[49]
1	1	0	0	1	0	INIT[50]	INIT[50]
1	1	0	0	1	1	INIT[51]	INIT[51]
1	1	0	1	0	0	INIT[52]	INIT[52]
1	1	0	1	0	1	INIT[53]	INIT[53]
1	1	0	1	1	0	INIT[54]	INIT[54]
1	1	0	1	1	1	INIT[55]	INIT[55]
1	1	1	0	0	0	INIT[56]	INIT[56]
1	1	1	0	0	1	INIT[57]	INIT[57]
1	1	1	0	1	0	INIT[58]	INIT[58]
1	1	1	0	1	1	INIT[59]	INIT[59]
1	1	1	1	0	0	INIT[60]	INIT[60]
1	1	1	1	0	1	INIT[61]	INIT[61]
1	1	1	1	1	0	INIT[62]	INIT[62]
1	1	1	1	1	1	INIT[63]	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値							

ポートの説明

ポート名	方向	幅	機能
O6	出力	1	6/5 LUT 出力
O5	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4、I5	入力	1	LUT 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	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

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;

-- LUT6: 6-input Look-Up Table with general output
--       Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT6_inst : LUT6
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT6: 6-input Look-Up Table with general output
//       Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

LUT6 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_inst (
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

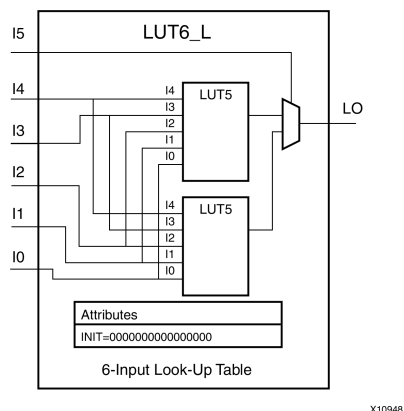
// End of LUT6_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

LUT6_L

プリミティブ : 6-Input Lookup Table with Local Output



概要

このデザイン エLEMENTは、入力 6 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 64 ビット ROM (6 ビットのアドレス指定) として動作するか、6 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT6 はルックアップ テーブル (LUT) 4 個のうちの 1 つにマップされます。LUT6、LUT6_L、および LUT6_D の機能は同じですが、LUT6_L および LUT6_D では、LUT6 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT6_L では LUT6 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT6_D では LUT6 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT6 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗示的に指定する必要がない限り、いつでも使用できます。

LUT のロジック ファンクションを指定するために 64 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力が適用されるときに、対応する INIT ビット値に 1 を割り当てることで計算されます。たとえば Verilog で INIT 値が `64'h8000000000000000` (VHDL では `X"8000000000000000"`) の場合は、入力すべてが 1 ではない限り、出力が 0 になります (6 入力の AND ゲート)。また、Verilog で INIT 値が `64'hffffffff` (VHDL では `X"FFFFFFFFFFFFFFFF"`) の場合は、入力がすべてゼロではない限り、出力は 1 になります (6 入力 OR ゲート)。

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。バイナリの真理値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
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]

入力						出力
I5	I4	I3	I2	I1	I0	LO
0	0	0	0	1	0	INIT[2]
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]

入力						出力
I5	I4	I3	I2	I1	I0	LO
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]
INIT = INIT 属性で指定された 16 進数値を 2 進数で表した値						

ポートの説明

ポート名	方向	幅	機能
LO	出力	1	6/5 入力 LUT 出力または内部 CLB 接続
I0、I1、I2、I3、I4、I5	入力	1	LUT 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	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

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;

-- LUT6: 6-input Look-Up Table with general output
--       Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

LUT6_inst : LUT6
generic map (
  INIT => X"0000000000000000") -- Specify LUT Contents
port map (
  O => O, -- LUT general output
  I0 => I0, -- LUT input
  I1 => I1, -- LUT input
  I2 => I2, -- LUT input
  I3 => I3, -- LUT input
  I4 => I4, -- LUT input
  I5 => I5 -- LUT input
);

-- End of LUT6_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// LUT6: 6-input Look-Up Table with general output
//       Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

LUT6 #(
  .INIT(64'h0000000000000000) // Specify LUT Contents
) LUT6_inst (
  .O(O), // LUT general output
  .I0(I0), // LUT input
  .I1(I1), // LUT input
  .I2(I2), // LUT input
  .I3(I3), // LUT input
  .I4(I4), // LUT input
  .I5(I5) // LUT input
);

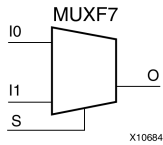
// End of LUT6_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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 エレメントは、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
);

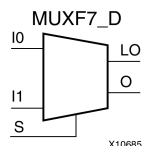
// End of MUXF7_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 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 エレメントは、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
);

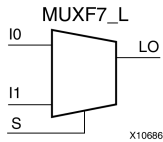
// End of MUXF7_D_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

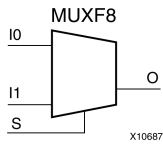
// End of MUXF7_L_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 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
);

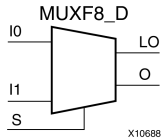
// End of MUXF8_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 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
);

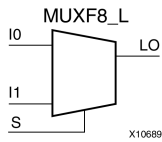
// End of MUXF8_D_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

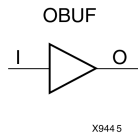
// End of MUXF8_L_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 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、トリステートでない FPGA デバイス ピンに駆動するために使用します。デザインのすべての出力ポートに OBUF、OBUFT、OBUFDS、OBUFTDS のいずれかを接続する必要があります。

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of OBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

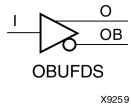
// End of OBUF_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
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;

-- 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 (インスタンス化)

```
// 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
);

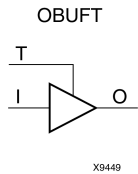
// End of OBUFDS_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
DRIVE	整数	2、4、6、8、12、16、24	12	出力の駆動電流を指定します。許容範囲で最も低い値を使用してください。
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をエレメントに割り当て
SLEW	文字列	SLOW、FAST	SLOW	出力ドライバのスループートを指定。この属性の最適な設定方法は、データシートを参照してください。

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of OBUFT_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

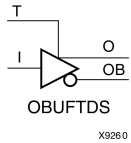
// End of OBUFT_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 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	トリステート イネーブル入力

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
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
);

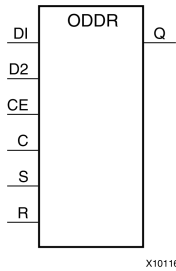
// End of OBUFTDS_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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 のクロック入力がい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
);

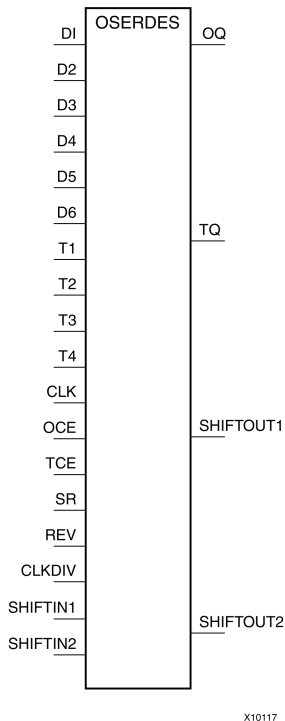
// End of ODDR_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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 (バイパスを介す)
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

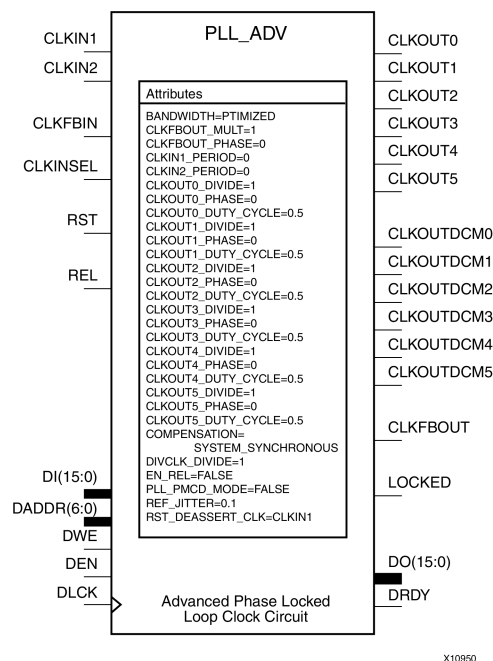
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

PLL_ADV

プリミティブ : Advanced Phase Locked Loop Clock Circuit



概要

このデザイン エLEMENTは、FPGA 内部と外部回路の両方に対するクロック合成およびクロック管理機能を備えたエンベデッド位相ロック ループ クロック回路です。PLL 回路を使用すると、クロックの多重化、位相一致、位相シフト、通倍、分周が可能になり、またデューティ サイクルやジッタ フィルタを変更できます。PLL は、DCM (デジタル クロック マネージャ) と共に、または DCM の代わりに使用でき、FPGA などの回路のクロック供給を制御します。

ポートの説明

ポート名	タイプ	幅	機能
CLKFBDCM	出力	1	DCM を駆動する場合に調整に使用する PLL フィードバック
CLKFBIN	入力	1	クロック フィードバック入力。このポートの接続は、フィードバック モード属性設定と一致している必要があります。
CLKFBOUT	出力	1	クロック ネットワークの遅延調整方法を指定するために使用する専用 PLL フィードバック出力。ユーザー コンフィギュレーション可能なクロック出力で、1 (バイパス) から 128 までの VCO 位相出力 (ユーザー制御可能) の分割値であることが可能です。デザインにより、位相は入力クロックおよび出力クロックに相対しています。
CLKINSEL	入力	1	入力マルチプレクサのステートを制御するコア配線からの信号。High = CLKIN1、Low = CLKIN2 です。
CLKIN1	入力	1	入力クロック
CLKIN2	入力	1	セカンダリ クロック入力で、リファレンス クロックを PLL にダイナミックに切り替えられるようにします。
CLKOUTDCM0	出力	1	ユーザー コンフィギュレーション可能なクロックで、PLL と同じ CMT 内で DCM のみに接続することができます。

ポート名	タイプ	幅	機能
CLKOUTDCM1 ~ CLKOUTDCM5	出力	1	ユーザー コンフィギュレーション可能なクロック出力で、1 (バイパス) から 128 までの VCO 位相出力 (ユーザー制御可能) の分割値であることが可能です。デザインにより、位相は入力クロックおよび出力クロックに相対しています。
CLKOUT0 ~ CLKOUT5	出力	1	ユーザー コンフィギュレーション可能なクロック出力で、1 (バイパス) から 128 までの VCO 位相出力 (ユーザー制御可能) の分割値であることが可能です。デザインにより、位相は入力クロックおよび出力クロックに相対しています。
DADDR[4:0]	入力	5	ダイナミック リコンフィギュレーションのアドレスを提供します。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。
DCLK	入力	1	ダイナミック リコンフィギュレーション ポートのリファレンス クロック
DEN	入力	1	ダイナミック リコンフィギュレーション機能にアクセスするための制御信号を提供します。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場合は、DEN が Low に接続されます。DEN が Low の場合 DO 出力がステータス信号を反映します。
DI[15:0]	入力	16	リコンフィギュレーション データを提供。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。
DO[15:0]	出力	16	ダイナミック リコンフィギュレーションを使用する場合、PLL ステータスまたはデータ出力を提供します。DO バスが PLL ステータスを表すには、DEN および DWE がそれぞれ GND に接続されていて、DADDR バスおよび DI バスがすべてゼロである必要があります。
DRDY	出力	1	PLL ダイナミック リコンフィギュレーション用に DEN 信号への応答を提供します。
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。
LOCKED	出力	1	PLL からの同期出力で、PLL の位相アライメントが定義されていたウィンドウ内で行われ、定義されていた PPM 範囲内の周波数で達成されたことを示します。
REL	入力	1	未使用
RST	入力	1	リセット

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
BANDWIDTH	文字列	OPTIMIZED、 HIGH、LOW	OPTIMIZED	PLL バンド幅制御。PLL バンド幅を設定します。 ・ HIGH に設定すると、良好な PLL パフォーマンスを維持しながら PLL プログラミング アルゴリズムが最も高い設定で実行されます (低ジッタ、低ジッタ ピーク、最大位相マージンなど)。 ・ LOW に設定すると、良好な PLL パフォーマンスを維持しながら PLL プログラミング アルゴリズムが最も低い設定で実行されます。
CLKFBOUT_ DESKEW_ ADJUST	文字列	NONE、 1、2、3、4、5、6、7、 8、9、10、11、12、 13、14、15、16、17、 18、19、20、21、22、 23、24、25、26、27、 28、29、30、31	NONE	該当ユーザー ガイドを参照。
CLKFBOUT_MULT	整数	1 ~ 64	1	カウンタ乗算値。常に 50% デューティサイクルにコンフィギュレーションされています。
CLKFBOUT_PHASE	1 上位ビット浮動 小数点	0.0 ~ 360.0	0.0	出力クロックに相対したカウンタ出力位相
CLK_FEEDBACK	文字列	CLKFBOUT、 CLKOUT0	CLKFBOUT	CLKFB_IN を駆動するクロック ソース。
CLKIN1_PERIOD ~ CLKIN2_PERIOD	文字列	0 ビット文字列	0	LOCK 検出回路をコンフィギュレーションし、VCO が有効範囲内で動作していることを確認するために必要なリファレンス クロック周波数です。値が指定されていない場合、エラー チェックが行われないことを知らせる警告が出されます。単位ナノ秒 (0.001ns と小数点第 3 桁まで)。

属性	タイプ	値	デフォルト	説明
CLKOUT0_DESKEW_ADJUST ~ CLKOUT5_DESKEW_ADJUST	文字列	NONE、 1、2、3、4、5、6、7、 8、9、10、11、12、 13、14、15、16、17、 18、19、20、21、22、 23、24、25、26、27、 28、29、30、31	NONE	後方互換性のためで、Spartan®-6 では使用されていません。
CLKOUT0_DIVIDE ~ CLKOUT5_DIVIDE	整数	1 ~ 128	1	カウンタ除算値
CLKOUT0_DUTY_CYCLE ~ CLKOUT5_DUTY_CYCLE	2 上位ビット浮動 小数点	0.01 ~ 0.99	0.50	カウンタ出力デューティ サイクル
CLKOUT0_PHASE ~ CLKOUT5_PHASE	1 上位ビット浮動 小数点	0.0 ~ 360.0	0.0	出力クロックに相対したカウンタ出力位相
COMPENSATION	文字列	SYSTEM_ SYNCHRONOUS、 SOURCE_ SYNCHRONOUS	SYSTEM_ SYNCHRONOUS	入力クロックの PLL 位相調整を指定します。 ・ SYSTEM_SYNCHRONOUS は、 ホールド タイム 0 ですべてのクロック遅延の調整を試みます。 ・ SOURCE_SYNCHRONOUS は、 クロックがデータと共に供給されて位相が揃っている場合に使用します。
DIVCLK_DIVIDE	整数	1 ~ 52	1	カウンタ除算値。常に 50% デューティ サイクルにコンフィギュレーションされています。
EN_REL	ブール代数	FALSE、TRUE	FALSE	後方互換性のためで、Spartan-6 では使用されていません。
PLL_PMCD_MODE	ブール代数	FALSE、TRUE	FALSE	後方互換性のためで、Spartan-6 では使用されていません。
REF_JITTER	3 上位ビット浮動 小数点	0.000 ~ 1.000	0.100	リファレンス クロック ジッタは、リファレンス クロックの割合で示した UI (ユニット インターバル) で指定します。この値は、入力クロックの最大ピークトゥピーク値にします。デフォルトは 0.1UI、または REF_CLK_FREQ、DIVCLK_DIVIDE、および REF_CLK_JITTER_MAX (PLL_SPEED_FILE.xls) の値に基づいた等価の UI。このパラメータは小数 3 桁までサポートします。
RESET_ON_LOSS_OF_LOCK	ブール代数	FALSE、TRUE	FALSE	後方互換性のためで、Spartan-6 では使用されていません。
RST_DEASSERT_CLK	文字列	CLKIN1、CLKFBIN	CLKIN1	後方互換性のためで、Spartan-6 では使用されていません。

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;

-- PLL_ADV: Phase-Lock Loop Clock Circuit
--      Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

PLL_ADV_inst : PLL_ADV
generic map (
    BANDWIDTH => "OPTIMIZED", -- "HIGH", "LOW" or "OPTIMIZED"
    CLKFBOUT_MULT => 1, -- Multiplication factor for all output clocks
    CLKFBOUT_PHASE => 0.0, -- Phase shift (degrees) of all output clocks
    CLKIN1_PERIOD => 0.000, -- Clock period (ns) of input clock on CLKIN1
    CLKIN2_PERIOD => 0.000, -- Clock period (ns) of input clock on CLKIN2
    CLKOUT0_DIVIDE => 1, -- Division factor for CLKOUT0 (1 to 128)
    CLKOUT0_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT0 (0.01 to 0.99)
    CLKOUT0_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    CLKOUT1_DIVIDE => 1, -- Division factor for CLKOUT1 (1 to 128)
    CLKOUT1_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT1 (0.01 to 0.99)
    CLKOUT1_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    CLKOUT2_DIVIDE => 1, -- Division factor for CLKOUT2 (1 to 128)
    CLKOUT2_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT2 (0.01 to 0.99)
    CLKOUT2_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    CLKOUT3_DIVIDE => 1, -- Division factor for CLKOUT3 (1 to 128)
    CLKOUT3_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT3 (0.01 to 0.99)
    CLKOUT3_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    CLKOUT4_DIVIDE => 1, -- Division factor for CLKOUT4 (1 to 128)
    CLKOUT4_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT4 (0.01 to 0.99)
    CLKOUT4_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    CLKOUT5_DIVIDE => 1, -- Division factor for CLKOUT5 (1 to 128)
    CLKOUT5_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT5 (0.01 to 0.99)
    CLKOUT5_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    COMPENSATION => "SYSTEM_SYNCHRONOUS", -- "SYSTEM_SYNCHRONOUS",
                                           -- "SOURCE_SYNCHRONOUS", "INTERNAL",
                                           -- "EXTERNAL", "DCM2PLL", "PLL2DCM"
    DIVCLK_DIVIDE => 1, -- Division factor for all clocks (1 to 52)
    EN_REL => FALSE, -- Enable release (PMCD mode only)
    PLL_PMCD_MODE => FALSE, -- PMCD Mode, TRUE/FALSE
    REF_JITTER => 0.100, -- Input reference jitter (0.000 to 0.999 UI%)
    RST_DEASSERT_CLK => "CLKIN1") -- In PMCD mode, clock to synchronize RST release
port map (
    CLKFBDCM => CLKFBDCM, -- Output feedback signal used when PLL feeds a DCM
    CLKFBOUT => CLKFBOUT, -- General output feedback signal
    CLKOUT0 => CLKOUT0, -- One of six general clock output signals
    CLKOUT1 => CLKOUT1, -- One of six general clock output signals
    CLKOUT2 => CLKOUT2, -- One of six general clock output signals
    CLKOUT3 => CLKOUT3, -- One of six general clock output signals
    CLKOUT4 => CLKOUT4, -- One of six general clock output signals
    CLKOUT5 => CLKOUT5, -- One of six general clock output signals
    CLKOUTDCM0 => CLKOUTDCM0, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM1 => CLKOUTDCM1, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM2 => CLKOUTDCM2, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM3 => CLKOUTDCM3, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM4 => CLKOUTDCM4, -- One of six clock outputs to connect to the DCM
    CLKOUTDCM5 => CLKOUTDCM5, -- One of six clock outputs to connect to the DCM
    DO => DO, -- Dynamic reconfig data output (16-bits)
    DRDY => DRDY, -- Dynamic reconfig ready output
    LOCKED => LOCKED, -- Active high PLL lock signal
    CLKFBIN => CLKFBIN, -- Clock feedback input
    CLKIN1 => CLKIN1, -- Primary clock input
    CLKIN2 => CLKIN2, -- Secondary clock input
    CLKINSEL => CLKINSEL, -- Selects CLKIN1 or CLKIN2
    DADDR => DADDR, -- Dynamic reconfig address input (5-bits)
    DCLK => DCLK, -- Dynamic reconfig clock input
    DEN => DEN, -- Dynamic reconfig enable input
    DI => DI, -- Dynamic reconfig data input (16-bits)
    DWE => DWE, -- Dynamic reconfig write enable input
    REL => REL, -- Clock release input (PMCD mode only)
    RST => RST -- Asynchronous PLL reset
);

-- End of PLL_ADV_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// PLL_ADV: Phase-Lock Loop Clock Circuit
//      Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

PLL_ADV #(
    .BANDWIDTH("OPTIMIZED"), // "HIGH", "LOW" or "OPTIMIZED"
    .CLKFBOUT_MULT(1),       // Multiplication factor for all output clocks
    .CLKFBOUT_PHASE(0.0),    // Phase shift (degrees) of all output clocks
    .CLKIN1_PERIOD(0.000),   // Clock period (ns) of input clock on CLKIN1
    .CLKIN2_PERIOD(0.000),   // Clock period (ns) of input clock on CLKIN2
    .CLKOUT0_DIVIDE(1),      // Division factor for CLKOUT0 (1 to 128)
    .CLKOUT0_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT0 (0.01 to 0.99)
    .CLKOUT0_PHASE(0.0),     // Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    .CLKOUT1_DIVIDE(1),      // Division factor for CLKOUT1 (1 to 128)
    .CLKOUT1_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT1 (0.01 to 0.99)
    .CLKOUT1_PHASE(0.0),     // Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    .CLKOUT2_DIVIDE(1),      // Division factor for CLKOUT2 (1 to 128)
    .CLKOUT2_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT2 (0.01 to 0.99)
    .CLKOUT2_PHASE(0.0),     // Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    .CLKOUT3_DIVIDE(1),      // Division factor for CLKOUT3 (1 to 128)
    .CLKOUT3_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT3 (0.01 to 0.99)
    .CLKOUT3_PHASE(0.0),     // Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    .CLKOUT4_DIVIDE(1),      // Division factor for CLKOUT4 (1 to 128)
    .CLKOUT4_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT4 (0.01 to 0.99)
    .CLKOUT4_PHASE(0.0),     // Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    .CLKOUT5_DIVIDE(1),      // Division factor for CLKOUT5 (1 to 128)
    .CLKOUT5_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT5 (0.01 to 0.99)
    .CLKOUT5_PHASE(0.0),     // Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    .COMPENSATION("SYSTEM_SYNCHRONOUS"), // "SYSTEM_SYNCHRONOUS",
                                         // "SOURCE_SYNCHRONOUS", "INTERNAL", "EXTERNAL",
                                         // "DCM2PLL", "PLL2DCM"
    .DIVCLK_DIVIDE(1),       // Division factor for all clocks (1 to 52)
    .EN_REL("FALSE"),        // Enable release (PMCD mode only)
    .PLL_PMCD_MODE("FALSE"), // PMCD Mode, TRUE/FALSE
    .REF_JITTER(0.100),      // Input reference jitter (0.000 to 0.999 UI%)
    .RST_DEASSERT_CLK("CLKIN1") // In PMCD mode, clock to synchronize RST release
) PLL_ADV_inst (
    .CLKFBDCM(CLKFBDCM), // Output feedback signal used when PLL feeds a DCM
    .CLKFBOUT(CLKFBOUT), // General output feedback signal
    .CLKOUT0(CLKOUT0),   // One of six general clock output signals
    .CLKOUT1(CLKOUT1),   // One of six general clock output signals
    .CLKOUT2(CLKOUT2),   // One of six general clock output signals
    .CLKOUT3(CLKOUT3),   // One of six general clock output signals
    .CLKOUT4(CLKOUT4),   // One of six general clock output signals
    .CLKOUT5(CLKOUT5),   // One of six general clock output signals
    .CLKOUTDCM0(CLKOUTDCM0), // One of six clock outputs to connect to the DCM
    .CLKOUTDCM1(CLKOUTDCM1), // One of six clock outputs to connect to the DCM
    .CLKOUTDCM2(CLKOUTDCM2), // One of six clock outputs to connect to the DCM
    .CLKOUTDCM3(CLKOUTDCM3), // One of six clock outputs to connect to the DCM
    .CLKOUTDCM4(CLKOUTDCM4), // One of six clock outputs to connect to the DCM
    .CLKOUTDCM5(CLKOUTDCM5), // One of six clock outputs to connect to the DCM
    .DO(DO),              // Dynamic reconfig data output (16-bits)
    .DRDY(DRDY),          // Dynamic reconfig ready output
    .LOCKED(LOCKED),      // Active high PLL lock signal
    .CLKFBIN(CLKFBIN),    // Clock feedback input
    .CLKIN1(CLKIN1),      // Primary clock input
    .CLKIN2(CLKIN2),      // Secondary clock input
    .CLKINSEL(CLKINSEL),  // Selects '1' = CLKIN1, '0' = CLKIN2
    .DADDR(DADDR),        // Dynamic reconfig address input (5-bits)
    .DCLK(DCLK),          // Dynamic reconfig clock input
    .DEN(DEN),            // Dynamic reconfig enable input
    .DI(DI),              // Dynamic reconfig data input (16-bits)
    .DWE(DWE),            // Dynamic reconfig write enable input
    .REL(REL),            // Clock release input (PMCD mode only)
    .RST(RST)             // Asynchronous PLL reset
);

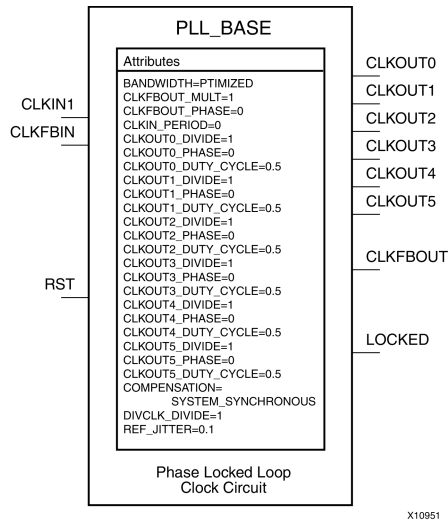
// End of PLL_ADV_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

PLL_BASE

プリミティブ : Basic Phase Locked Loop Clock Circuit



X10951

概要

このデザイン エLEMENTは、FPGA 内部と外部回路の両方に対するクロック合成およびクロック管理機能を備えたエンベデッド位相ロック ループ クロック回路で、PLL_ADV デザイン エLEMENTのサブセットです。PLL_BASE を使用すると、ほとんどの PLL クロック回路において統合が簡単になります。このコンポーネントには PLL で提供可能なすべての機能は備わっていませんが、入力クロックの位相をシフト、通倍、分周でき、またデューティ サイクルやジッタ フィルタを変更する機能があります。

ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLKOUT0 ~ 5	出力	1	位相が制御される 6 個の出力クロックの 1 つ
CLKFBOUT	出力	1	クロック ネットワークの遅延調整方法を指定するために使用する専用 PLL フィードバック出力。この出力の接続の有無は、調整方法によって異なります。
CLKIN	入力	1	PLL のクロック ソース入力。FPGA の専用クロック ピン、DCM 出力クロック ピン、または BUFG 出力ピンによって駆動されます。
CLKFBIN	入力	1	クロック フィードバック入力。CLKFBOUT ポートからのみ接続できます。
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	非同期リセット

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
COMPENSATION	文字列	SYSTEM_ SYNCHRONOUS、SOURCE_ SYNCHRONOUS	SYSTEM_ SYNCHRONOUS	入力クロックの PLL 位相調整を指定します。すべてのクロック遅延を調整する場合は SYSTEM_ SYNCHRONOUS を、クロックがデータと共に供給されて位相が揃っているときには SOURCE_ SYNCHRONOUS を使用します。
BANDWIDTH	文字列	HIGH、LOW、OPTIMIZED	OPTIMIZED	ジッタ、位相マージンなどの PLL 特性に影響する PLL プログラム アルゴリズムを指定
CLKOUT0_DIVIDE、CLKOUT1_DIVIDE、CLKOUT2_DIVIDE、CLKOUT3_DIVIDE、CLKOUT4_DIVIDE、CLKOUT5_DIVIDE	整数	1 ~ 128	1	別の周波数を使用する場合に、CLKOUT クロック出力を分周する値を指定します。この値と FBCLKOUT_MULT 値から出力周波数が決まります。
CLKOUT0_PHASE、CLKOUT1_PHASE、CLKOUT2_PHASE、CLKOUT3_PHASE、CLKOUT4_PHASE、CLKOUT5_PHASE	実数	0.01 ~ 360.0	0.0	CLKOUT クロック出力との位相オフセットを度数で指定します。90 は 90 度または 4 分の 1 サイクルの位相オフセット、180 は 180 度または 2 分の 1 サイクルの位相オフセットを示します。
CLKOUT0_DUTY_CYCLE、CLKOUT1_DUTY_CYCLE、CLKOUT2_DUTY_CYCLE、CLKOUT3_DUTY_CYCLE、CLKOUT4_DUTY_CYCLE、CLKOUT5_DUTY_CYCLE	実数	0.01 ~ 0.99	0.50	CLKOUT クロック出力のデューティサイクルをパーセントで指定します。0.50 の場合、デューティサイクルは 50% になります。
CLKFBOUT_MULT	整数	1 ~ 64	1	別の周波数を使用する場合に、すべての CLKOUT クロック出力を過倍する値を指定します。この値と CLKOUT#_DIVIDE 値から出力周波数が決まります。
DIVCLK_DIVIDE	整数	1 ~ 52	1	すべての出力クロックの分周比を指定
CLKFBOUT_PHASE	実数	0.0 ~ 360	0.0	クロックフィードバック出力の位相オフセットを度数で指定します。
REF_JITTER	実数	0.000 ~ 0.999	0.100	リファレンス クロック ジッタは、リファレンス クロックの割合で示した UI (ユニット インターバル) で指定します。この値は、入力クロックの最大ピークトゥピーク値にします。
CLKIN_PERIOD	実数	1.000 ~ 52.630	0.000	PLL CLKIN 入力への入力周期を指定 (ns)

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;

-- PLL_BASE: Phase-Lock Loop Clock Circuit
--           Virtex-5, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

PLL_BASE_inst : PLL_BASE
generic map (
    BANDWIDTH => "OPTIMIZED", -- "HIGH", "LOW" or "OPTIMIZED"
    CLKFBOUT_MULT => 1, -- Multiplication factor for all output clocks
    CLKFBOUT_PHASE => 0.0, -- Phase shift (degrees) of all output clocks
    CLKIN_PERIOD => 0.000, -- Clock period (ns) of input clock on CLKIN
    CLKOUT0_DIVIDE => 1, -- Division factor for CLKOUT0 (1 to 128)
    CLKOUT0_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT0 (0.01 to 0.99)
    CLKOUT0_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    CLKOUT1_DIVIDE => 1, -- Division factor for CLKOUT1 (1 to 128)
    CLKOUT1_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT1 (0.01 to 0.99)
    CLKOUT1_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    CLKOUT2_DIVIDE => 1, -- Division factor for CLKOUT2 (1 to 128)
    CLKOUT2_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT2 (0.01 to 0.99)
    CLKOUT2_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    CLKOUT3_DIVIDE => 1, -- Division factor for CLKOUT3 (1 to 128)
    CLKOUT3_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT3 (0.01 to 0.99)
    CLKOUT3_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    CLKOUT4_DIVIDE => 1, -- Division factor for CLKOUT4 (1 to 128)
    CLKOUT4_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT4 (0.01 to 0.99)
    CLKOUT4_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    CLKOUT5_DIVIDE => 1, -- Division factor for CLKOUT5 (1 to 128)
    CLKOUT5_DUTY_CYCLE => 0.5, -- Duty cycle for CLKOUT5 (0.01 to 0.99)
    CLKOUT5_PHASE => 0.0, -- Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    COMPENSATION => "SYSTEM_SYNCHRONOUS", -- "SYSTEM_SYNCHRONOUS",
                                           -- "SOURCE_SYNCHRONOUS", "INTERNAL",
                                           -- "EXTERNAL", "DCM2PLL", "PLL2DCM"
    DIVCLK_DIVIDE => 1, -- Division factor for all clocks (1 to 52)
    REF_JITTER => 0.100) -- Input reference jitter (0.000 to 0.999 UI%)
port map (
    CLKFBOUT => CLKFBOUT, -- General output feedback signal
    CLKOUT0 => CLKOUT0, -- One of six general clock output signals
    CLKOUT1 => CLKOUT1, -- One of six general clock output signals
    CLKOUT2 => CLKOUT2, -- One of six general clock output signals
    CLKOUT3 => CLKOUT3, -- One of six general clock output signals
    CLKOUT4 => CLKOUT4, -- One of six general clock output signals
    CLKOUT5 => CLKOUT5, -- One of six general clock output signals
    LOCKED => LOCKED, -- Active high PLL lock signal
    CLKFBIN => CLKFBIN, -- Clock feedback input
    CLKIN => CLKIN, -- Clock input
    RST => RST -- Asynchronous PLL reset
);

-- End of PLL_BASE_inst instantiation
```


Verilog 記述 (インスタンス化)

```
// PLL_BASE: Phase-Lock Loop Clock Circuit
//          Virtex-5, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

PLL_BASE #(
    .BANDWIDTH("OPTIMIZED"), // "HIGH", "LOW" or "OPTIMIZED"
    .CLKFBOUT_MULT(1),       // Multiplication factor for all output clocks
    .CLKFBOUT_PHASE(0.0),    // Phase shift (degrees) of all output clocks
    .CLKIN_PERIOD(0.000),    // Clock period (ns) of input clock on CLKIN
    .CLKOUT0_DIVIDE(1),      // Division factor for CLKOUT0 (1 to 128)
    .CLKOUT0_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT0 (0.01 to 0.99)
    .CLKOUT0_PHASE(0.0),     // Phase shift (degrees) for CLKOUT0 (0.0 to 360.0)
    .CLKOUT1_DIVIDE(1),      // Division factor for CLKOUT1 (1 to 128)
    .CLKOUT1_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT1 (0.01 to 0.99)
    .CLKOUT1_PHASE(0.0),     // Phase shift (degrees) for CLKOUT1 (0.0 to 360.0)
    .CLKOUT2_DIVIDE(1),      // Division factor for CLKOUT2 (1 to 128)
    .CLKOUT2_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT2 (0.01 to 0.99)
    .CLKOUT2_PHASE(0.0),     // Phase shift (degrees) for CLKOUT2 (0.0 to 360.0)
    .CLKOUT3_DIVIDE(1),      // Division factor for CLKOUT3 (1 to 128)
    .CLKOUT3_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT3 (0.01 to 0.99)
    .CLKOUT3_PHASE(0.0),     // Phase shift (degrees) for CLKOUT3 (0.0 to 360.0)
    .CLKOUT4_DIVIDE(1),      // Division factor for CLKOUT4 (1 to 128)
    .CLKOUT4_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT4 (0.01 to 0.99)
    .CLKOUT4_PHASE(0.0),     // Phase shift (degrees) for CLKOUT4 (0.0 to 360.0)
    .CLKOUT5_DIVIDE(1),      // Division factor for CLKOUT5 (1 to 128)
    .CLKOUT5_DUTY_CYCLE(0.5), // Duty cycle for CLKOUT5 (0.01 to 0.99)
    .CLKOUT5_PHASE(0.0),     // Phase shift (degrees) for CLKOUT5 (0.0 to 360.0)
    .COMPENSATION("SYSTEM_SYNCHRONOUS"), // "SYSTEM_SYNCHRONOUS",
                                         // "SOURCE_SYNCHRONOUS", "INTERNAL", "EXTERNAL",
                                         // "DCM2PLL", "PLL2DCM"
    .DIVCLK_DIVIDE(1),       // Division factor for all clocks (1 to 52)
    .REF_JITTER(0.100)       // Input reference jitter (0.000 to 0.999 UI%)
) PLL_BASE_inst (
    .CLKFBOUT(CLKFBOUT), // General output feedback signal
    .CLKOUT0(CLKOUT0),   // One of six general clock output signals
    .CLKOUT1(CLKOUT1),   // One of six general clock output signals
    .CLKOUT2(CLKOUT2),   // One of six general clock output signals
    .CLKOUT3(CLKOUT3),   // One of six general clock output signals
    .CLKOUT4(CLKOUT4),   // One of six general clock output signals
    .CLKOUT5(CLKOUT5),   // One of six general clock output signals
    .LOCKED(LOCKED),     // Active high PLL lock signal
    .CLKFBIN(CLKFBIN),   // Clock feedback input
    .CLKIN(CLKIN),        // Clock input
    .RST(RST)             // Asynchronous PLL reset
);

// End of PLL_BASE_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

PPC440

プリミティブ : Power PC 440 CPU Core

概要

このELEMENTは 2 命令同時発行可能なスーパースカラ プロセッサで、旧型の PowerPC® 405 と同じ命令セット アーキテクチャのインプリメントのパフォーマンスが大幅に改善されています。

デザインの入力方法

インスタンス化	不可
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

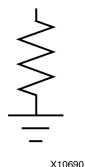
詳細情報

- ・ [IBM PPC440x5 CPU Core User's Manual](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)
- ・ [Virtex-5 FPGA ユーザー ガイド](#)

PULLDOWN

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

PULLDOWN



X10690

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
```

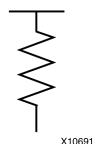
詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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

PULLUP



概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

-- PULLUP: I/O Buffer Weak Pull-up
--      All FPGA, CoolRunner-II
-- Xilinx HDL Libraries Guide, version 11.2

PULLUP_inst : PULLUP
port map (
  O => O      -- Pullup output (connect directly to top-level port)
);

-- End of PULLUP_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// PULLUP: I/O Buffer Weak Pull-up
//      All FPGA, CoolRunner-II
// Xilinx HDL Libraries Guide, version 11.2

PULLUP PULLUP_inst (
  .O(O)      // Pullup output (connect directly to top-level port)
);

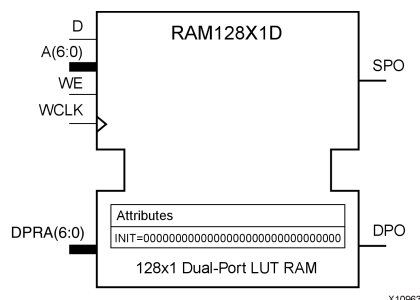
// End of PULLUP_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM128X1D

プリミティブ : 128-Deep by 1-Wide Dual Port Random Access Memory (Select RAM)



概要

このデザイン エLEMENTは 128 ワード X 1 ビットの RAM で読み出し/書き込みポートがあり、ライト イネーブル (WE) が High のときにアドレス バス A で指定されたロケーションに D 入力データ ピンの値が書き込まれます。この書き込みは WCLK の立ち上がりエッジの直後に実行され、同じ値が SPO に出力されます。WE が Low のときは非同期読み出しが実行され、アドレス バス A で指定されたメモリ ロケーションの値が SPO に非同期で出力されます。アドレス バス DPRA の値を変更することにより、読み出しポートでは非同期読み出しを実行できます。DPO にその値が出力されます。

ポートの説明

ポート名	方向	幅	機能
SPO	出力	1	アドレス バス A で指定された読み出し/書き込みポートのデータ出力
DPO	出力	1	アドレス バス DPRA で指定された読み出しポートのデータ出力
D	入力	1	アドレス バス A で指定された書き込みデータ入力
A	入力	7	読み出し/書き込みポートのアドレス バス
DPRA	入力	7	読み出しポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

インスタンスエートする場合は、このコンポーネントを次のように接続します。

- ・ WCLK 入力をクロックソースに、D 入力を格納するデータソースに、DPO 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- ・ オプションで、SPO 出力を適切なデスティネーションに接続するか、または未接続にすることもできます。
- ・ クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- ・ 7 ビット バス A は読み出し/書き込みアドレスに、7 ビット バス DPRA は読み出しアドレスに接続する必要があります。
- ・ 128 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべてゼロになります。

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	16 進数	128 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;

-- RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
--           dual-port distributed LUT RAM
--           Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAM128X1D_inst : RAM128X1D
generic map (
  INIT => X"00000000000000000000000000000000"
)
port map (
  DPO => DPO,      -- Read/Write port 1-bit output
  SPO => SPO,      -- Read port 1-bit output
  A => A,          -- Read/Write port 7-bit address input
  D => D,          -- RAM data input
  DPRA => DPRA,    -- Read port 7-bit address input
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- RAM data input
);

-- End of RAM128X1D_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// RAM128X1D: 128-deep by 1-wide positive edge write, asynchronous read
//           dual-port distributed LUT RAM
//           Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

RAM128X1D #(
  .INIT(128'h00000000000000000000000000000000)
) RAM128X1D_inst (
  .DPO(DPO),      // Read port 1-bit output
  .SPO(SPO),      // Read/Write port 1-bit output
  .A(A),          // Read/Write port 7-bit address input
  .D(D),          // RAM data input
  .DPRA(DPRA),    // Read port 7-bit address input
  .WCLK(WCLK),    // Write clock input
  .WE(WE)         // Write enable input
);

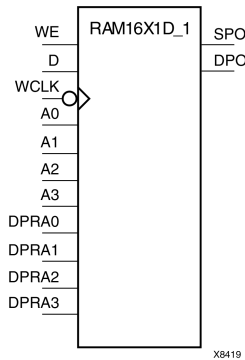
// End of RAM128X1D_inst instantiation
```


詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

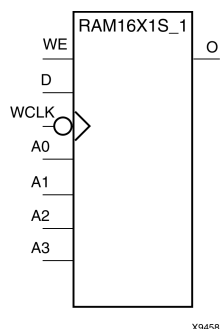
// End of RAM16X1D_1_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

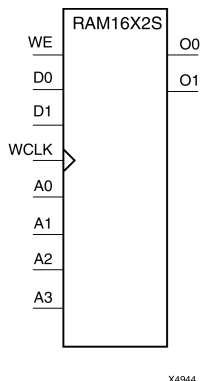
// End of RAM16X1S_1_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

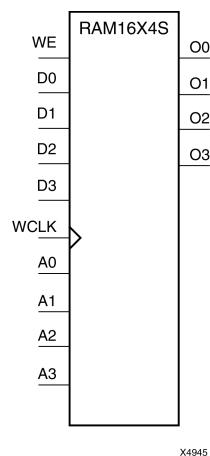
// End of RAM16X2S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

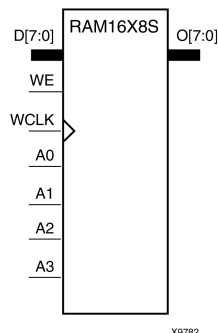
// End of RAM16X4S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

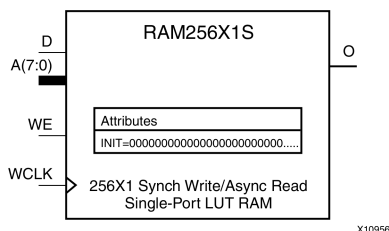
// End of RAM16X8S_inst instantiation
```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM256X1S

プリミティブ : 256-Deep by 1-Wide Random Access Memory (Select RAM)



概要

このデザイン エLEMENTは、256 ワード X 1 ビットの RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (Select RAM と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースを使用しません。同期読み出しを行う場合は、出力にレジスタを付けて同じスライスに配置できます。ただし、この場合は RAM とレジスタで同じクロックを使用する必要があります。RAM256X1S には、アクティブ High のライト イネーブル (WE) があり、この信号が High になると、WCLK ピンの立ち上がりエッジで D 入力データピンの値がメモリ アレイに書き込まれます。出力 O は、WE の値にかかわらず、アドレス バス A で指定されたメモリ ロケーションの値を出力します。書き込みが実行されると、出力の値が更新されます。

ポートの説明

ポート名	方向	幅	機能
O	出力	1	アドレス バス A で指定された読み出し/書き込みポートのデータ出力
D	入力	1	アドレス バス A で指定された書き込みデータ入力
A	入力	8	読み出し/書き込みポートのアドレス バス
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

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

インスタンス化する場合、このコンポーネントを次のように接続します。

- WCLK 入力をクロック ソースに、D 入力を格納するデータ ソースに、O 出力を FDCE の D 入力などの適切なデスティネーションに接続します。
- クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。
- 8 ビット バス A は、読み出し/書き込みのソースに接続します。
- 256 ビットの 16 進数で構成される INIT 属性で、RAM の初期値を指定できます。

指定しない場合は、初期値はすべてゼロになります。

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	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;

-- RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
--             single-port distributed LUT RAM
--             Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAM256X1S_inst : RAM256X1S
generic map (
  INIT => X"0000000000000000000000000000000000000000000000000000000000000000"
)
port map (
  O => O,  -- Read/Write port 1-bit output
  A => A,  -- Read/Write port 8-bit address input
  D => D,  -- RAM data input
  WCLK => WCLK, -- Write clock input
  WE => WE -- Write enable input
);

-- End of RAM256X1S_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAM256X1S: 256-deep by 1-wide positive edge write, asynchronous read
//             single-port distributed LUT RAM
//             Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

RAM256X1S #(
  .INIT(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAM256X1S_inst (
  .O(O),      // Read/Write port 1-bit output
  .A(A),      // Read/Write port 8-bit address input
  .WE(WE),    // Write enable input
  .WCLK(WCLK), // Write clock input
  .D(D)       // RAM data input
);

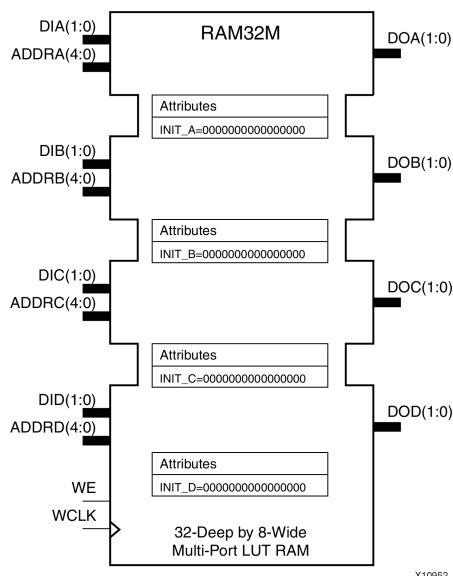
// End of RAM256X1S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM32M

プリミティブ : 32-Deep by 8-bit Wide Multi Port Random Access Memory (Select RAM)



概要

このデザイン エLEMENTは、32 ワード X 8 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™) を使用してインプリメントされるため、デバイスのブロック RAM リソースを使用しません。RAM32M は、1 つのスライスにインプリメントされます。含まれるのは、8 ビット書き込み、2 ビット読み出しのポート 1 つと、同じメモリからの 2 ビットの読み出しポートが 3 つです。これにより、バイト幅の書き込みと独立した 2 ビットの読み出しが可能な RAM ができます。DIA、DIB、DIC、および DID 入力すべてが同じデータ入力に接続されると、この RAM は読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 32x2 クワッドポートメモリになります。DID がグランドに接続される場合、DOD は使用されません。ADDRA、ADDR B、ADDRC が同じアドレスに接続されると、この RAM は 32x6 の単純なデュアルポート RAM になります。ADDR D が ADDRA、ADDR B、ADDRC に接続されると、32x8 のシングルポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

ポートの説明

ポート名	方向	幅	機能
DOA	出力	2	アドレス バス ADDRA で指定された読み出しポートのデータ出力
DOB	出力	2	アドレス バス ADDRb で指定された読み出しポートのデータ出力
DOC	出力	2	アドレス バス ADDRC で指定された読み出しポートのデータ出力
DOD	出力	2	アドレス バス ADDRd で指定された読み出し/書き込みポートのデータ出力
DIA	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRb で指定)
DIC	入力	2	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRC で指定)
DID	入力	2	アドレス バス ADDRd で指定された書き込みデータ入力
ADDRA	入力	5	読み出しアドレス バス A
ADDRB	入力	5	読み出しアドレス バス B
ADDRC	入力	5	読み出しアドレス バス C
ADDRD	入力	5	8 ビットのデータ書き込みポート、2 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

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

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、合成ツールで推論できるようになることがあります。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM32M のインスタンス化は、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しを行う場合は、RAM32M の出力を FDRSE に接続してファンクションの出力タイミングを向上させることも可能ですが、通常の RAM の操作では不要です。

インバータをこのコンポーネントのクロック入力に追加すると、クロックの立ち下がりエッジでデータを入力できます。このインバータはブロック内に組み込まれ、クロックの立ち下がりエッジで RAM への書き込みを実行できます。

インスタンス化する場合、このコンポーネントは、次のように接続します。WCLK 入力をクロックソースに、DIA、DIB、DIC、DID 入力を格納するデータソースに、DOA、DOB、DOC、DOD 出力を FDCE の D 入力などの適切なデスティネーションに接続するか、使用しない場合は未接続のままにします。クロックイネーブルピン (WE) は、適切なライトイネーブルソースに接続します。5 ビットバス ADDR_D は読み出し/書き込みアドレスに、5 ビットバス ADDR_A、ADDR_B、ADDR_C は読み出しアドレスに接続する必要があります。オプションで INIT_A、INIT_B、INIT_C、INIT_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、 $ADDR_y[z] = INIT_y[2*z+1:2*z]$ で計算されます。たとえば、RAM の ADDR_C ポートが 00001 の場合、INIT_C[3:2] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべてゼロになります。

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべてゼロ	A ポートの RAM の初期値を指定
INIT_B	16 進数	64 ビット値	すべてゼロ	B ポートの RAM の初期値を指定
INIT_C	16 進数	64 ビット値	すべてゼロ	C ポートの RAM の初期値を指定
INIT_D	16 進数	64 ビット値	すべてゼロ	D ポートの RAM 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;

-- RAM32M: 32-deep by 8-wide Multi Port LUT RAM
--          Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAM32M_inst : RAM32M
generic map (
  INIT_A => X"0000000000000000", -- Initial contents of A port
  INIT_B => X"0000000000000000", -- Initial contents of B port
  INIT_C => X"0000000000000000", -- Initial contents of C port
  INIT_D => X"0000000000000000") -- Initial contents of D port
port map (
  DOA => DOA, -- Read port A 2-bit output
  DOB => DOB, -- Read port B 2-bit output
  DOC => DOC, -- Read port C 2-bit output
  DOD => DOD, -- Read/Write port D 2-bit output
  ADDR_A => ADDR_A, -- Read port A 5-bit address input
  ADDR_B => ADDR_B, -- Read port B 5-bit address input
  ADDR_C => ADDR_C, -- Read port C 5-bit address input
  ADDR_D => ADDR_D, -- Read/Write port D 5-bit address input
  DIA => DIA, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_A
  DIB => DIB, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_B
  DIC => DIC, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_C
  DID => DID, -- RAM 2-bit data write input addressed by ADDR_D,
               -- read addressed by ADDR_D
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);
-- End of RAM32M_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM32M: 32-deep by 8-wide Multi Port LUT RAM
//           Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

RAM32M #(
    .INIT_A(64'h0000000000000000), // Initial contents of A Port
    .INIT_B(64'h0000000000000000), // Initial contents of B Port
    .INIT_C(64'h0000000000000000), // Initial contents of C Port
    .INIT_D(64'h0000000000000000) // Initial contents of D Port
) RAM32M_inst (
    .DOA(DOA),           // Read port A 2-bit output
    .DOB(DOB),           // Read port B 2-bit output
    .DOC(DOC),           // Read port C 2-bit output
    .DOD(DOD),           // Read/Write port D 2-bit output
    .ADDRA(ADDRA),       // Read port A 5-bit address input
    .ADDRB(ADDRB),       // Read port B 5-bit address input
    .ADDRC(ADDRC),       // Read port C 5-bit address input
    .ADDRD(ADDRD),       // Read/Write port D 5-bit address input
    .DIA(DIA),           // RAM 2-bit data write input addressed by ADDRD,
                        //   read addressed by ADDRA
    .DIB(DIB),           // RAM 2-bit data write input addressed by ADDRD,
                        //   read addressed by ADDRb
    .DIC(DIC),           // RAM 2-bit data write input addressed by ADDRD,
                        //   read addressed by ADDRc
    .DID(DID),           // RAM 2-bit data write input addressed by ADDRD,
                        //   read addressed by ADDRd
    .WCLK(WCLK),         // Write clock input
    .WE(WE)              // Write enable input
);

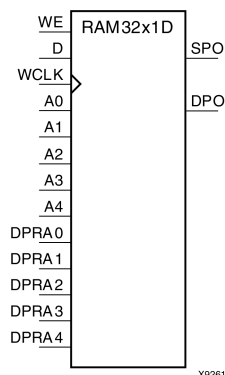
// End of RAM32M_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM32X1D

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



概要

このデザイン エLEMENTは 32 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 (DPRA4 ~ DPRA0) と書き込みアドレス (A4 ~ 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 ポートは完全に非同期です。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 5 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。INIT 属性を使用すると、コンフィギュレーション中に RAM32X1D を初期化できます。モード選択を次の論理表に示します。

SPO 出力には、A4 ~ 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、DPRA4 ~ DPRA0 で指定されたメモリ セルの値が出力されます。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

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

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、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;

-- RAM32X1D: 32 x 1 positive edge write, asynchronous read dual-port distributed RAM
--           Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X1D_inst : RAM32X1D
generic map (
  INIT => X"00000000")
port map (
  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
  A4 => A4,        -- R/W address[4] 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
  DPRA4 => DPRA4,  -- Read-only address[4] input bit
  WCLK => WCLK,    -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X1D_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAM32X1D: 32 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

RAM32X1D #(
  .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1D_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
  .A4(A4),            // R/W address[4] 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
  .DPRA4(DPRA4),      // Read-only address[4] input bit
  .WCLK(WCLK),        // Write clock input
  .WE(WE)             // Write enable input
);

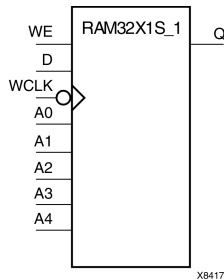
// End of RAM32X1D_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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;

-- 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 (インスタンスレーション)

```
// 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
);

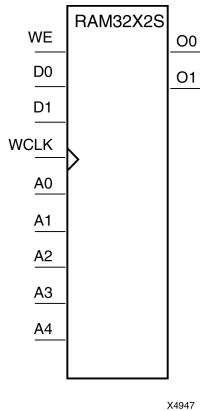
// End of RAM32X1S_1_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

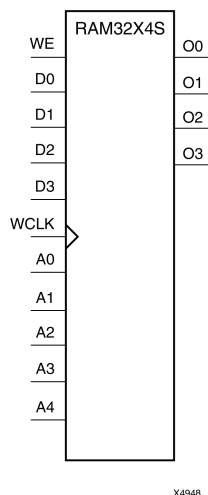
// End of RAM32X2S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

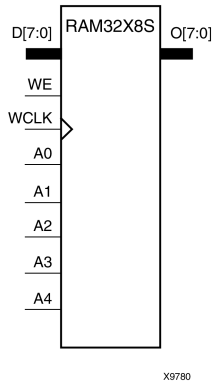
// End of RAM32X4S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

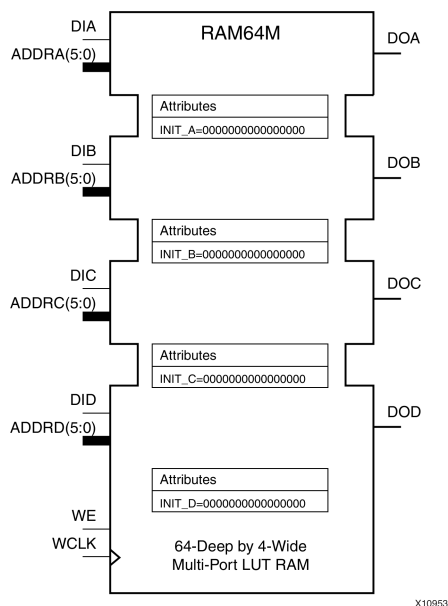
// End of RAM32X8S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM64M

プリミティブ : 64-Deep by 4-bit Wide Multi Port Random Access Memory (Select RAM)



概要

このデザイン エLEMENTは、64 ワード X 4 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™ と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースを使用しません。RAM64M は、1 つのスライスにインプリメントされます。含まれるのは、4 ビット書き込み、1 ビット読み出しのポート 1 つと、同じメモリからの 1 ビットの読み出しポートが 3 つです。DIA、DIB、DIC、および DID 入力すべてが同じデータ入力に接続されると、この RAM は読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 64x1 クワッド ポートメモリになります。DID がグランドに接続される場合、DOD は使用されません。ADDRA、ADDRB、ADDRC が同じアドレスに接続されると、この RAM は 64x3 の単純なデュアル ポート RAM になります。ADDRD が ADDR A、ADDRB、ADDRC に接続されると、64x4 のシングル ポート RAM になります。この RAM には、ほかにも可能なコンフィギュレーションがあります。

ポートの説明

ポート名	方向	幅	機能
DOA	出力	1	アドレス バス ADDRA で指定された読み出しポートのデータ出力
DOB	出力	1	アドレス バス ADDRb で指定された読み出しポートのデータ出力
DOC	出力	1	アドレス バス ADDRC で指定された読み出しポートのデータ出力
DOD	出力	1	アドレス バス ADDRd で指定された読み出し/書き込みポートのデータ出力
DIA	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRb で指定)
DIC	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRC で指定)
DID	入力	1	アドレス バス ADDRd で指定された書き込みデータ入力
ADDRA	入力	6	読み出しアドレス バス A
ADDRb	入力	6	読み出しアドレス バス B
ADDRC	入力	6	読み出しアドレス バス C
ADDRd	入力	6	4 ビットのデータ書き込みポート、1 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

デザインの入力方法

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

このエレメントは、同期書き込みと非同期読み出し機能を備えた RAM を記述することにより、合成ツールで推論できるようになることがあります。RAM の推論およびコード例の詳細は、合成ツールのマニュアルを参照してください。RAM64M のインスタンスエーションは、RAM ファンクションを暗示的に指定する必要がある場合、コンポーネントを手動でまたは相対的に配置する必要がある場合に実行することをお勧めします。同期読み出しを行う場合は、RAM64M の出力を FDRSE に接続してファンクションの出力タイミングを向上させることも可能ですが、通常の RAM の操作では不要です。インバータをこのコンポーネントのクロック入力に追加すると、クロックの立ち下がりエッジでデータを入力できます。このインバータはブロック内に組み込まれ、クロックの立ち下がりエッジで RAM への書き込みを実行できます。

インスタンスエートする場合、このコンポーネントは、次のように接続します。WCLK 入力をクロックソースに、DIA、DIB、DIC、DID 入力を格納するデータソースに、DOA、DOB、DOC、DOD 出力を FDCE の D 入力などの適切なデスティネーションに接続するか、使用しない場合は未接続のままにします。クロック イネーブル ピン (WE) は、適切なライト イネーブル ソースに接続します。5 ビット バス ADDRd は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRb、ADDRC は読み出しアドレスに接続する必要があります。オプションで INIT_A、INIT_B、INIT_C、INIT_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、ADDRy[z] = INIT_y[z] で計算されます。

たとえば、RAM の ADDRC ポートが 00001 の場合、INIT_C[1] 値がそのアドレスで最初の書き込みが行われる前の DOC ポートの初期値になります。指定しない場合は、初期値はすべてゼロになります。

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_A	16 進数	64 ビット値	すべてゼロ	A ポートの RAM の初期値を指定
INIT_B	16 進数	64 ビット値	すべてゼロ	B ポートの RAM の初期値を指定
INIT_C	16 進数	64 ビット値	すべてゼロ	C ポートの RAM の初期値を指定
INIT_D	16 進数	64 ビット値	すべてゼロ	D ポートの 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;

-- RAM64M: 64-deep by 4-wide Multi Port LUT RAM
--          Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAM64M_inst : RAM64M
generic map (
  INIT_A => X"0000000000000000", -- Initial contents of A port
  INIT_B => X"0000000000000000", -- Initial contents of B port
  INIT_C => X"0000000000000000", -- Initial contents of C port
  INIT_D => X"0000000000000000") -- Initial contents of D port
port map (
  DOA => DOA, -- Read port A 1-bit output
  DOB => DOB, -- Read port B 1-bit output
  DOC => DOC, -- Read port C 1-bit output
  DOD => DOD, -- Read/Write port D 1-bit output
  ADDRA => ADDRA, -- Read port A 6-bit address input
  ADDRb => ADDRb, -- Read port B 6-bit address input
  ADDRc => ADDRc, -- Read port C 6-bit address input
  ADDRd => ADDRd, -- Read/Write port D 6-bit address input
  DIA => DIA, -- RAM 1-bit data write input addressed by ADDRd,
               -- read addressed by ADDRA
  DIB => DIB, -- RAM 1-bit data write input addressed by ADDRd,
               -- read addressed by ADDRb
  DIC => DIC, -- RAM 1-bit data write input addressed by ADDRd,
               -- read addressed by ADDRc
  DID => DID, -- RAM 1-bit data write input addressed by ADDRd,
               -- read addressed by ADDRd
  WCLK => WCLK, -- Write clock input
  WE => WE      -- Write enable input
);
-- End of RAM64M_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// RAM64M: 64-deep by 4-wide Multi Port LUT RAM
//          Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

RAM64M #(
  .INIT_A(64'h0000000000000000), // Initial contents of A Port
  .INIT_B(64'h0000000000000000), // Initial contents of B Port
  .INIT_C(64'h0000000000000000), // Initial contents of C Port
  .INIT_D(64'h0000000000000000) // Initial contents of D Port
) RAM64M_inst (
  .DOA(DOA), // Read port A 1-bit output
  .DOB(DOB), // Read port B 1-bit output
  .DOC(DOC), // Read port C 1-bit output
  .DOD(DOD), // Read/Write port D 1-bit output
  .DIA(DIA), // RAM 1-bit data write input addressed by ADDRDRD,
              // read addressed by ADDRRA
  .DIB(DIB), // RAM 1-bit data write input addressed by ADDRDRD,
              // read addressed by ADDRDB
  .DIC(DIC), // RAM 1-bit data write input addressed by ADDRDRD,
              // read addressed by ADDRRC
  .DID(DID), // RAM 1-bit data write input addressed by ADDRDRD,
              // read addressed by ADDRDRD
  .ADDRA(ADDRRA), // Read port A 6-bit address input
  .ADDRB(ADDRRB), // Read port B 6-bit address input
  .ADDRC(ADDRRC), // Read port C 6-bit address input
  .ADDRD(ADDRDRD), // Read/Write port D 6-bit address input
  .WE(WE), // Write enable input
  .WCLK(WCLK) // Write clock input
);

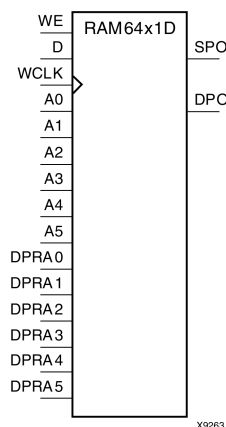
// End of RAM64M_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM64X1D

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



概要

このデザイン エLEMENTは 64 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 (DPRA5 ~ DPRA0) と書き込みアドレス (A5 ~ 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 ポートは完全に非同期です。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 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 ビットの書き込みアドレス (A0 ~ A5) 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、A5 ~ 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、DPRA5 ~ 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

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 = A5 ~ A0 で指定されたワード				
data_d = DPRA5 ~ 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	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;

-- RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
--           Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAM64X1D_inst : RAM64X1D
generic map (
    INIT => X"0000000000000000")
port map (
    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
    A4 => A4,        -- R/W address[4] input bit
    A5 => A5,        -- R/W address[5] input bit
    D => D,          -- Write 1-bit data input
    DPRA0 => DPRA0,  -- 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
    DPRA4 => DPRA4,  -- Read-only address[4] input bit
    DPRA5 => DPRA5,  -- Read-only address[5] input bit
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);

-- End of RAM64X1D_inst instantiation
```


Verilog 記述 (インスタンス化)

```
// RAM64X1D: 64 x 1 positive edge write, asynchronous read dual-port distributed RAM
//           Virtex-5/6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

RAM64X1D #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1D_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
    .A4(A4),             // R/W address[4] input bit
    .A5(A5),             // R/W address[5] 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
    .DPRA4(DPRA4),       // Read-only address[4] input bit
    .DPRA5(DPRA5),       // Read-only address[5] input bit
    .WCLK(WCLK),         // Write clock input
    .WE(WE)              // Write enable input
);

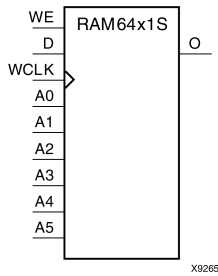
// End of RAM64X1D_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

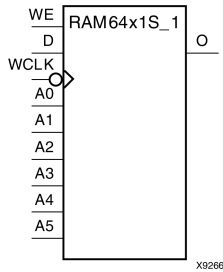
// End of RAM64X1S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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

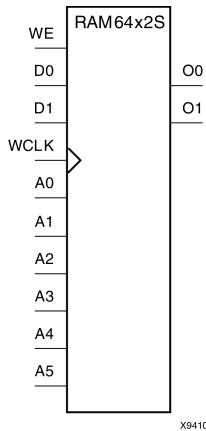
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

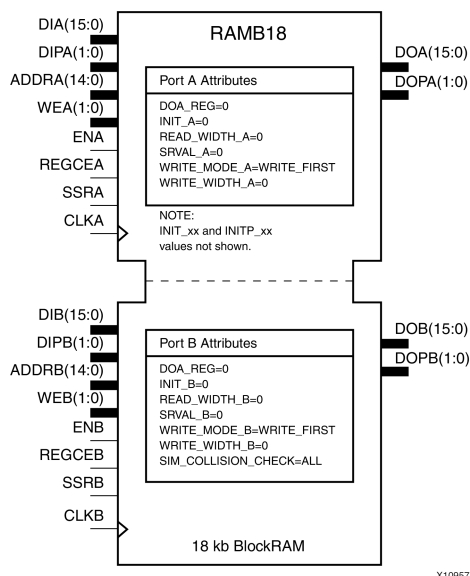
// End of RAM64X2S_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAMB18

プリミティブ : 18K-bit Configurable Synchronous True Dual Port Block RAM



X10957

概要

Virtex®-5 デバイスにはブロック RAM が数個含まれ、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36 Kb または 18 Kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。RAMB18 を使用すると、18kb の FIFO のブロック RAM へアクセスできます。このエレメントをカスケード 接続すると、大型の RAM を作成できます。このエレメントは、1 ビット X 16K ワード ~ 18 ビット X 1029 ワードの完全なデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、READ と WRITE は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。広いデータ幅でコンフィギュレーションすると、バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

ポートの説明

ポート名	方向	幅	機能
DOA, DOB	出力	1、2、4、8、16	ポート A/B のデータ出力バス
DOPA, DOPB	出力	0、1、2	ポート A/B のパリティ データ出力バス
DIA, DIB	入力	1、2、4、8、16	ポート A/B のデータ入力バス
DIPA, DIPB	入力	0、1、2	ポート A/B のパリティ データ入力バス
ADDRA, ADDRb	入力	14	ポート A/B のアドレス入力バス
WEA	入力	2	ポート A のバイト幅ライト イネーブル
WEB	入力	2	ポート B のバイト幅ライト イネーブル
ENA, ENB	入力	1	ポート A/B のイネーブル
SSRA, SSRB	入力	1	ポート A/B の出力レジスタの同期セット/リセット。High になると、関連ポートのプリセット/リセットが SRVAL_A/SRVAL_B で指定した値になります。

ポート名	方向	幅	機能
REGCEA、REGCEB	入力	1	ポート A/B の出力レジスタのクロック イネーブル入力
CLKA、CLKB	入力	1	ポート A/B のクロック入力

デザインの入力方法

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

次の表に、さまざまなポート幅のデータ、アドレスおよびライトイネーブル接続を、ポート A またはポート B の DATA_WIDTH の値ごとに示します。同じポートの読み出しと書き込みで異なる幅を使用する場合は、アドレス接続を選択できるようにこの表のデータ幅が大きい方の 2 つを使用してください。

特定のコンフィギュレーションで必要にならないデータ ポートおよびアドレス ポートは、次の例外を除き、すべて未接続にするかまたはグラウンドに接続する必要があります。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1	DI[0]	ADDR[14:0]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[14:1]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[14:2]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[14:3]	WE[1:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[14:4]	WE[1:0] および WE[1] をユーザー WE 信号に接続	DO[15:0]、DOP[1:0]

使用可能な属性

属性	タイプ	値	デフォルト	説明
DOA_REG、DOB_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT_A	16 進数	18 ビット値	すべてゼロ	コンフィギュレーション後のポート A の出力の初期値を指定します。
INIT_B	16 進数	18 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
READ_WIDTH_A	整数	0、1、2、4、9、18	0	ポート A の読み出しのデータ幅を指定します (パリティビットを含む)。0 の場合は、ポートが使用されません。

属性	タイプ	値	デフォルト	説明
READ_WIDTH_B	整数	0、1、2、4、9、18	0	ポート B の読み出しのデータ幅を指定します (パリティビットを含む)。0 の場合は、ポートが使用されません。
SIM_COLLISION_CHECK	文字列	ALL、 WARNING_ONLY、 GENERATE_X_ONLY、 NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。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>
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
SRVAL_A	16 進数	18 ビット値	すべてゼロ	同期リセット信号 (SSRA) がアサートされたときのポート A の出力値を指定
SRVAL_B	16 進数	18 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	<p>書き込みが実行されるときポートの動作を指定します。</p> <ul style="list-style-type: none"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
WRITE_WIDTH_A	整数	0、1、2、4、9、18	0	ポート A への書き込みのデータ幅を指定します (パリティビットを含む)。0 の場合は、ポートが使用されません。
WRITE_WIDTH_B	整数	0、1、2、4、9、18	0	ポート B への書き込みのデータ幅を指定します (パリティビットを含む)。0 の場合は、ポートが使用されません。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定

属性	タイプ	値	デフォルト	説明
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティデータメモリアレイの初期値を指定

INIT_A、INIT_B、SRVAL_A、SRVAL_B のマップ

INIT_A、INIT_B、SRVAL_A および SRVAL_B の属性はすべて 18 ビットですが、特定のポートで READ_WIDTH が 18 ビット未満の値に設定されていると、そのビットのサブセットのみが使用されます。

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;

-- RAMB18: 16k+2k Parity Paramatizable True Dual-Port BlockRAM
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

RAMB18_inst : RAMB18
generic map (
    DOA_REG => 0, -- Optional output register on A port (0 or 1)
    DOB_REG => 0, -- Optional output register on B port (0 or 1)
    INIT_A => X"00000", -- Initial values on A output port
    INIT_B => X"00000", -- Initial values on B output port
    READ_WIDTH_A => 0, -- Valid values are 1, 2, 4, 9, or 18
    READ_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, or 18
    SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
    -- "GENERATE_X_ONLY" or "NONE"
    SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
    -- Design Guide" for details
    SRVAL_A => X"00000", -- Set/Reset value for A port output
    SRVAL_B => X"00000", -- 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 => 0, -- Valid values are 1, 2, 4, 9, or 18
    WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, or 18
    -- 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 (
  DOA => DOA, -- 16-bit A port data output
  DOB => DOB, -- 16-bit B port data output
  DOPA => DOPA, -- 2-bit A port parity data output
  DOPB => DOPB, -- 2-bit B port parity data output
  ADDRA => ADDRA, -- 14-bit A port address input
  ADDRb => ADDRb, -- 14-bit B port address input
  CLKA => CLKA, -- 1-bit A port clock input
  CLKB => CLKB, -- 1 bit B port clock input
  DIA => DIA, -- 16-bit A port data input
  DIB => DIB, -- 16-bit B port data input
  DIPA => DIPA, -- 2-bit A port parity data input
  DIPB => DIPB, -- 2-bit B port parity data 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 set/reset input
  SSRB => SSRB, -- 1-bit B port set/reset input
  WEA => WEA, -- 2-bit A port write enable input
  WEB => WEB -- 2-bit B port write enable input
);

-- End of RAMB18_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB18: 16k+2k Parity Paramatizable True Dual-Port BlockRAM
// Virtex-5

```

```
// Xilinx HDL Libraries Guide, version 11.2
```

[illegible]

```
.INIT_35(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_36(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_37(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_38(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_39(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_3F(256'h0000000000000000000000000000000000000000000000000000000000000000),

// The next set of INITP_xx are for the parity bits
.INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_04(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_05(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_06(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INITP_07(256'h0000000000000000000000000000000000000000000000000000000000000000)
) RAMB18_inst (
    .DOA(DOA),           // 16-bit A port data output
    .DOB DOB),          // 16-bit B port data output
    .DOPA(DOPA),         // 2-bit A port parity data output
    .DOPB(DOPB),         // 2-bit B port parity data output
    .ADDRA(ADDRA),       // 14-bit A port address input
    .ADDRB(ADDRB),       // 14-bit B port address input
    .CLKA(CLKA),         // 1-bit A port clock input
    .CLKB(CLKB),         // 1-bit B port clock input
    .DIA(DIA),           // 16-bit A port data input
    .DIB(DIB),           // 16-bit B port data input
    .DIPA(DIPA),         // 2-bit A port parity data input
    .DIPB(DIPB),         // 2-bit B port parity data 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 set/reset input
    .SSRB(SSRB),         // 1-bit B port set/reset input
    .WEA(WEA),           // 2-bit A port write enable input
    .WEB(WEB)            // 2-bit B port write enable input
);

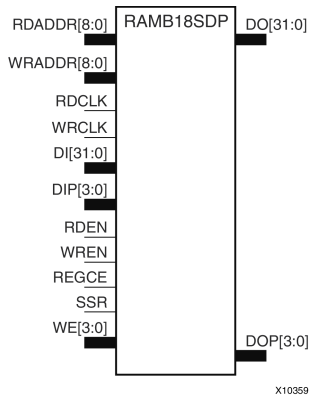
// End of RAMB18_inst instantiation
```

詳細情報

- ・ Virtex-5 FPGA ユーザー ガイド
- ・ Virtex-5 FPGA データシート：DC 特性およびスイッチ特性

RAMB18SDP

プリミティブ : 36-bit by 512 Deep, 18kb Synchronous Simple Dual Port Block RAM



概要

このデザイン エLEMENTはブロック RAM メモリの 1 つで、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。RAMB18SDP を使用すると、18kb のブロック RAM へアクセスできます。このコンポーネントは、36 ビット幅 X 512 ワードの単純なデュアル ポート RAM に設定されています。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、READ と WRITE は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。

メモ : このELEMENTは、読み出しおよび書き込みのポートの幅が同じになるようにコンフィギュレーションする必要があります。

ポートの説明

ポート名	方向	幅	機能
DO	出力	32	RDADDR で指定されたデータ出力バス
DOP	出力	4	RDADDR で指定されたデータ パリティ出力バス
DI	入力	32	WRADDR で指定されたデータ入力バス
DIP	入力	4	WRADDR で指定されたデータ パリティ入力バス
WRDDRA、 RDDDRB	入力	9	書き込み/読み出しアドレス入力バス
WE	入力	4	ライト イネーブル
WREN、RDEN	入力	1	書き込み/読み出しイネーブル
SSR	入力	1	出力レジスタの同期リセット
REGCE	入力	1	出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
WRCLK、RDCLK	入力	1	書き込み/読み出しクロック入力

デザインの入力方法

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

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	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロック サイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
SIM_COLLISION_CHECK	文字列	ALL、WARNING_ONLY、GENERATE_X_ONLY、NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。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>
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
SRVAL	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSR) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスーション)

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。

[illegible]

```

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 (
  DO => DO,          -- 32-bit Data Output
  DOP => DOP,         -- 4-bit Parity Output
  RDCLK => RDCLK,     -- 1-bit read port clock
  RDEN => RDEN,       -- 1-bit read port enable
  REGCE => REGCE,     -- 1-bit register enable input
  SSR => SSR,         -- 1-bit synchronous output set/reset input
  WRCLK => WRCLK,     -- 1-bit write port clock
  WREN => WREN,       -- 1-bit write port enable
  WRADDR => WRADDR,   -- 9-bit write port address input
  RDADDR => RDADDR,   -- 9-bit read port address input
  DI => DI,           -- 32-bit data input
  DIP => DIP,         -- 4-bit parity data input
  WE => WE            -- 4-bit write enable input
);

-- End of RAMB18SDP_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB18SDP: 36x512 Simple Dual-Port BlockRAM
// Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

RAMB18SDP #(
  .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
  .DO_REG(0), // Optional output register (0 or 1)
  .INIT(36'h00000000), // Initial values on output port
  .SIM_COLLISION_CHECK("ALL"), // Collision check enable "ALL", "WARNING_ONLY",
  // "GENERATE_X_ONLY" or "NONE"
  .SRVAL(36'h00000000), // Set/Reset value for port output

  // The following INIT_xx declarations specify the initial contents of the RAM
  .INIT_00(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_01(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_02(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_03(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_04(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_05(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_06(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_07(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_08(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_09(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_0A(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_0B(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_0C(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_0D(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_0E(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_0F(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),
  .INIT_10(256'h00000000_00000000_00000000_00000000_00000000_00000000_00000000_00000000),

```

[illegible]

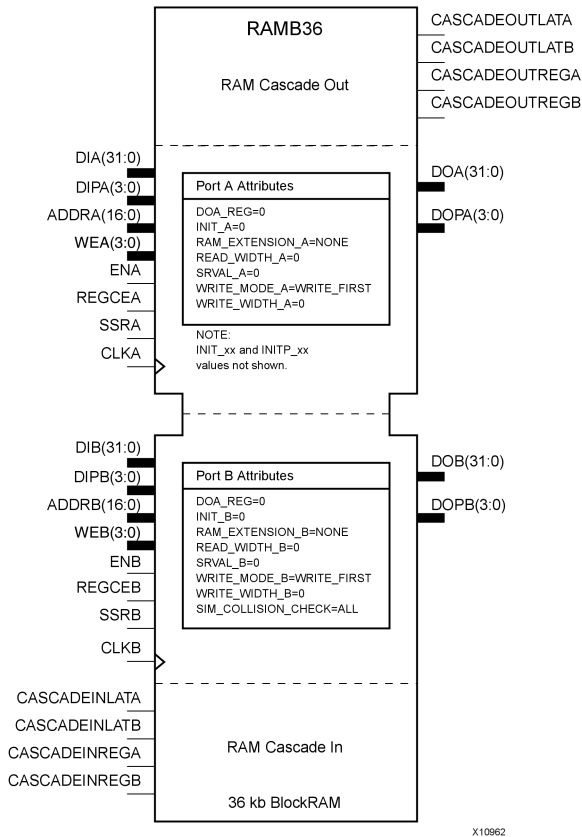
```
// End of RAMB18SDP_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAMB36

プリミティブ : 36kb Configurable Synchronous True Dual Port Block RAM



X10962

概要

このデザイン エLEMENTはブロック RAM メモリの 1 つで、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。またこのデザイン エLEMENTを使用すると、36kb の FIFO のブロック RAM へアクセスできます。このコンポーネントは、1 ビット X 32K ワード ~ 36 ビット X 1K ワードの完全なデュアル ポート RAM としてコンフィギュレーションできます。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。広いデータ幅でコンフィギュレーションすると、バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できますが、レイテンシは増加します。

このデザイン エLEMENTは単純なデュアル ポート (72 ビット幅 X 512 ワード、オプションで ECC スクラブ機能あり) に使用してください。このELEMENTは、カスケード接続された RAMB18 を使用して作成できます。

RAMB36 に配置可能なELEMENTの組み合わせを次に示します。

- ・ RAMB18/RAMB18
- ・ RAMB18/FIFO18
- ・ RAMB18SDP/RAMB18SDP
- ・ RAMB18SDP/FIFO18_36

ポートの説明

ポート名	方向	幅	機能
DOA	出力	1、2、4、8、16、32	ポート A のデータ出力バス
DOB	出力	1、2、4、8、16、32	ポート B のデータ出力バス
DOPA、DOPB	出力	0、1、2、4	ポート A/B のパリティ データ出力バス
CASCADEOUTLATA、 CASCADEOUTLATB	出力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の出力 (下位にある RAMB36 の CASCADEINLATA、CASCADEINLATB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が LOWER に設定されていない場合は未接続)。
CASCADEOUTREGA、 CASCADEOUTREGB	出力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の出力 (下位にある RAMB36 の CASCADEINREGA、CASCADEINREGB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が LOWER に設定されていない場合は未接続)。
CASCADEINLATA、 CASCADEINLATB	入力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の入力 (上位にある RAMB36 の CASCADEOUTLATA、CASCADEOUTLATB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が UPPER に設定されていない場合は未接続)。
CASCADEINREGA、 CASCADEINREGB	入力	1	2 つのブロック RAM をカスケード接続して 64K ワード X 1 ビットのメモリを作成するときに使用するポート A とポート B の入力 (上位にある RAMB36 の CASCADEOUTREGA、CASCADEOUTREGB に接続。64K X 1 RAM を作成しない場合または RAM_EXTENSION_A/B が UPPER に設定されていない場合は未接続)。
DIA	入力	1、2、4、8、16、32	ポート A のデータ入力バス
DIB	入力	1、2、4、8、16、32	ポート B のデータ入力バス
DIPA、DIPB	入力	0、1、2、4	ポート A/B のパリティ データ入力バス
ADDRA、ADDRB	入力	16	ポート A/B のアドレス入力バス。CASC モードの場合は 16。
WEA	入力	4	ポート A のバイト幅ライト イネーブル
WEB	入力	4	ポート B のバイト幅ライト イネーブル
ENA、ENB	入力	1	ポート A/B のイネーブル。High になると関連するポートの読み出しまたは書き込みが実行されます。
SSRA、SSRB	入力	1	ポート A/B の出力レジスタの同期セット/リセット。High になると、関連ポートのプリセット/リセットが SRVAL_A/SRVAL_B で指定した値になります。
REGCEA、REGCEB	入力	1	ポート A/B の出力レジスタのクロック イネーブル入力。High になると、関連するポートの出力レジスタが有効になります。

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	可

次の表に、さまざまなポート幅のデータ、アドレスおよびライトイネーブル接続を、ポート A またはポート B の DATA_WIDTH の値ごとに示します。同じポートの読み出しと書き込みで異なる幅を使用する場合は、アドレス接続を選択できるようにこの表のデータ幅が大きい方の 2 つを使用してください。

特定のコンフィギュレーションで必要にならないデータ ポートおよびアドレス ポートは、次の例外を除き、すべて未接続にするかまたはグランドに接続する必要があります。

- ・ アドレス ビット 15 はカスケード可能なブロック RAM でのみ使用します。カスケードしないブロック RAM の場合は、High に接続します。
- ・ ADDR ピンは 16 ビット幅である必要があります。ただし、カスケード可能でない RAM の有効アドレスはピン 14 (15 - アドレス幅) のみです。残りのピンは、ピン 15 を含め High に接続しておく必要があります。

DATA_WIDTH 値	DI、DIP 接続	ADDR 接続	WE 接続	DO、DOP 接続
1 (カスケードあり)	DI[0]	ADDR[15:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
1 (カスケードなし)	DI[0]	ADDR[14:0]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[0]
2	DI[1:0]	ADDR[14:1]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[1:0]
4	DI[3:0]	ADDR[14:2]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[3:0]
9	DI[7:0]、DIP[0]	ADDR[14:3]	WE[3:0] をシングル ユーザー WE 信号に接続	DO[7:0]、DOP[0]
18	DI[15:0]、DIP[1:0]	ADDR[14:4]	WE[0] および WE[2] をユーザー WE[0] に、WE[1] および WE[3] をユーザー WE[1] に接続	DO[15:0]、DOP[1:0]
36	DI[31:0]、DIP[3:0]	ADDR[14:5]	各 WE[3:0] 信号を関連したバイトライト イネーブルに接続	DO[31:0]、DOP[3:0]

使用可能な属性

属性	タイプ	値	デフォルト	説明
DOA_REG、 DOB_REG	整数	0、1	0	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロック サイクルで読み出しが可能です。clock-to-out タイムが長くなります。
INIT_A	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート A の出力の初期値を指定。
INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定

属性	タイプ	値	デフォルト	説明
READ_WIDTH_A	整数	0、1、4、9、18、36	0	ポート A の読み出しのデータ幅を指定します (パリティビットを含む)。ポート B を使用しない場合は、ポート幅を 0 にする必要があります。それ以外の場合は、任意のポート幅に設定してください。
READ_WIDTH_B	整数	0、1、4、9、18、36	0	ポート B の読み出しのデータ幅を指定します (パリティビットを含む)。ポート B を使用しない場合は、ポート幅を 0 にする必要があります。それ以外の場合は、任意のポート幅に設定してください。
SIM_COLLISION_CHECK	文字列	ALL、WARNING_ONLY、GENERATE_X_ONLY、NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。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>
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
SRVAL_A	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRA) がアサートされたときのポート A の出力値を指定
SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (SSRB) がアサートされたときのポート B の出力値を指定します。

属性	タイプ	値	デフォルト	説明
WRITE_MODE_A、 WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_ FIRST	書き込みが実行されるときのパートの動作を指定します。 <ul style="list-style-type: none"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリ ロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
WRITE_WIDTH_A	整数	0、1、2、4、9、18、 36	0	ポート B への書き込みのデータ幅を指定します (パリティビットを含む)。ポートを使用しない場合は、ポート幅を 0 にする必要があります。それ以外の場合は、任意のデータ幅に設定してください。
WRITE_WIDTH_B	整数	0、1、2、4、9、18、 36	0	ポート B への書き込みのデータ幅を指定します (パリティビットを含む)。ポートを使用しない場合は、ポート幅を 0 にする必要があります。それ以外の場合は、任意のデータ幅に設定してください。
RAM_EXTENTION_A、 RAM_EXTENTION_B	文字列	UPPER、LOWER、 NONE	NONE	2 つのブロック RAM をカスケード接続して 72K X 1 RAM を作成しない場合は、NONE に設定します。カスケード接続する場合は、RAM を正しくコンフィギュレーションするために、RAM の相対位置を UPPER または LOWER で指定します。
INIT_00 ~ INIT_7F	16 進数	256 ビット値	すべてゼロ	72kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべてゼロ	4kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- RAMB36: 32k+4k Parity Paramatizable True Dual-Port BlockRAM
--          Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

RAMB36_inst : RAMB36
generic map (
    DOA_REG => 0, -- Optional output register on A port (0 or 1)
    DOB_REG => 0, -- Optional output register on 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
    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"
SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST", see "Synthesis and Simulation
                    -- Design Guide" for details
SRVAL_A => X"00000000", -- Set/Reset value for A port output
SRVAL_B => X"00000000", -- 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 => 0, -- Valid values are 1, 2, 3, 4, 9, 18, 36
WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 3, 4, 9, 18, 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",
```

[illegible]

```

INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000")
port map (
  CASCADEOUTLATA => CASCADEOUTLATA, -- 1-bit cascade A latch output
  CASCADEOUTLATB => CASCADEOUTLATB, -- 1-bit cascade B latch output
  CASCADEOUTREGA => CASCADEOUTREGA, -- 1-bit cascade A register output
  CASCADEOUTREGB => CASCADEOUTREGB, -- 1-bit cascade B register output
  DOA => DOA, -- 32-bit A port data output
  DOB => DOB, -- 32-bit B port data output
  DOPA => DOPA, -- 4-bit A port parity data output
  DOPB => DOPB, -- 4-bit B port parity data output
  ADDRA => ADDRA, -- 16-bit A port address input
  ADDRb => ADDRb, -- 16-bit B port address input
  CASCADEINLATA => CASCADEINLATA, -- 1-bit cascade A latch input
  CASCADEINLATB => CASCADEINLATB, -- 1-bit cascade B latch input
  CASCADEINREGA => CASCADEINREGA, -- 1-bit cascade A register input
  CASCADEINREGB => CASCADEINREGB, -- 1-bit cascade B register input
  CLKA => CLKA, -- 1-bit A port clock input
  CLKB => CLKB, -- 1 bit B port clock input
  DIA => DIA, -- 32-bit A port data input
  DIB => DIB, -- 32-bit B port data input
  DIPA => DIPA, -- 4-bit A port parity data input
  DIPB => DIPB, -- 4-bit B port parity data input
  ENA => ENA, -- 1-bit A port enable input
  ENB => ENB, -- 1-bit B port enable input
  REGCEA => REGCEA, -- 1-bit A port register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  SSRA => SSRA, -- 1-bit A port set/reset input
  SSRB => SSRB, -- 1-bit B port set/reset input
  WEA => WEA, -- 4-bit A port write enable input
  WEB => WEB -- 4-bit B port write enable input
);

-- End of RAMB36_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// RAMB36: 32k+4k Parity Paramatizable True Dual-Port BlockRAM
//          Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

RAMB36 #(
  .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
  .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
  .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(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(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
  .INIT_02(256'h0000000000000000000000000000000000000000000000000000000000000000),

```

[illegible]

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

```

.CASCADEOUTREGA(CASCADEOUTREGA), // 1-bit cascade A register output
.CASCADEOUTREGB(CASCADEOUTREGB), // 1-bit cascade B register output
.DOA(DOA), // 32-bit A port data output
.DOB(DOB), // 32-bit B port data output
.DOPA(DOPA), // 4-bit A port parity data output
.DOPB(DOPB), // 4-bit B port parity data output
.ADDRA(ADDRB), // 16-bit A port address input
.ADDRB(ADDRB), // 16-bit B port address input
.CASCADEINLATA(CASCADEINLATA), // 1-bit cascade A latch input
.CASCADEINLATB(CASCADEINLATB), // 1-bit cascade B latch input
.CASCADEINREGA(CASCADEINREGA), // 1-bit cascade A register input
.CASCADEINREGB(CASCADEINREGB), // 1-bit cascade B register input
.CLKA(CLKA), // 1-bit A port clock input
.CLKB(CLKB), // 1-bit B port clock input
.DIA(DIA), // 32-bit A port data input
.DIB(DIB), // 32-bit B port data input
.DIPA(DIPA), // 4-bit A port parity data input
.DIPB(DIPB), // 4-bit B port parity data input
.ENA(ENA), // 1-bit A port enable input
.ENB(ENB), // 1-bit B port enable input
.REGCEA(REGCEA), // 1-bit A port register enable input
.REGCEB(REGCEB), // 1-bit B port register enable input
.SSRA(SSRA), // 1-bit A port set/reset input
.SSRB(SSRB), // 1-bit B port set/reset input
.WEA(WEA), // 4-bit A port write enable input
.WEB(WEB) // 4-bit B port write enable input
);

// End of RAMB36_inst instantiation

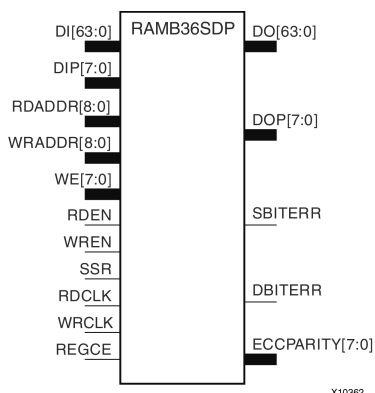
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

RAMB36SDP

プリミティブ : 72-bit by 512 Deep, 36kb Synchronous Simple Dual Port Block RAM with ECC (Error Correction Circuitry)



概要

このデザイン エLEMENTはブロック RAM メモリの 1 つで、FIFO、自動エラー訂正 RAM、または汎用 RAM/ROM (36kb または 18kb) としてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。RAMB36SDP を使用すると、36kb のブロック RAM へアクセスできます。このコンポーネントは、72 ビット幅 X 512 ワードの単純なデュアル ポート RAM に設定されています。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、READ と WRITE は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。バイト イネーブルの書き込みが可能になり、オプションの出力レジスタを使用して RAM の clock-to-out タイムを短縮できます。エラー検出と訂正回路をイネーブルにすると、メモリ破損を検出し、修正することもできます。

メモ : このELEMENTは、読み出しおよび書き込みのポートの幅が同じになるようにコンフィギュレーションする必要があります。

ポートの説明

ポート名	方向	幅	機能
DO	出力	64	RDADDR で指定されたデータ出力バス
DOP	出力	8	RDADDR で指定されたデータ パリティ出力バス
SBITERR	出力	1	シングル ビット エラーが検出されたことを示す ECC ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
DBITERR	出力	1	ダブル ビット エラーが検出されたことを示す ECC ファンクションからのステータス出力。使用する場合は、EN_ECC_READ を TRUE にする必要があります。
ECCPARITY	出力	8	メモリ エラー検出と訂正を行う ECC デコーダで使用される ECC エンコーダから生成された 8 ビット データ
DI	入力	64	WRADDR で指定されたデータ入力バス
DIP	入力	8	WRADDR で指定されたデータ パリティ入力バス
WRADDR, RDADDR	入力	9	書き込み/読み出しアドレス入力バス
WE	入力	8	書き込みイネーブル
WREN, RDEN	入力	1	書き込み/読み出しイネーブル

ポート名	方向	幅	機能
SSR	入力	1	出力レジスタの同期リセット
REGCE	入力	1	出力レジスタのクロック イネーブル入力 (DO_REG=1 の場合にのみ有効)
WRCLK、RDCLK	入力	1	書き込み/読み出しクロック入力

デザインの入力方法

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

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	値を 1 にすると、RAM の出力レジスタがイネーブルになり、RAM からの clock-to-out タイムが短縮されます。ただし、読み出しレイテンシのクロックサイクルは増加します。値を 0 にすると、1 クロックサイクルで読み出しが可能ですが、clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
EN_ECC_READ	ブール代数	TRUE、FALSE	FALSE	ECC デコーダ回路をイネーブルにします。
EN_ECC_WRITE	ブール代数	TRUE、FALSE	FALSE	ECC エンコーダ回路をイネーブルにします。
EN_ECC_SCRUB	ブール代数	TRUE、FALSE	FALSE	RAM の内容の ECC スクラブ機能をイネーブルにします。
SIM_COLLISION_CHECK	文字列	ALL、WARNING_ONLY、GENERATE_X_ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>

属性	タイプ	値	デフォルト	説明
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーション モデルがパフォーマンス重視モードで実行されます。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。
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (SSR) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティデータ メモリ アレイの初期値を指定

VHDL 記述 (インスタンス化)

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

[illegible]

[illegible]

```

INIT_69 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_6F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_70 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_71 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_72 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_73 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_74 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_75 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_76 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_77 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_78 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_79 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7A => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_7F => 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",
INITP_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
port map (
  DBITERR => DBITERR, -- 1-bit double bit error status output
  SBITERR => SBITERR, -- 1-bit single bit error status output
  DO => DO,           -- 64-bit Data Output
  DOP => DOP,         -- 8-bit Parity Output
  ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
  RDCLK => RDCLK,     -- 1-bit read port clock
  RDEN => RDEN,       -- 1-bit read port enable
  REGCE => REGCE,     -- 1-bit register enable input
  SSR => SSR,         -- 1-bit synchronous output set/reset input
  WRCLK => WRCLK,     -- 1-bit write port clock
  WREN => WREN,       -- 1-bit write port enable
  WRADDR => WRADDR,   -- 9-bit write port address input
  RDADDR => RDADDR,   -- 9-bit read port address input
  DI => DI,           -- 64-bit data input
  DIP => DIP,         -- 8-bit parity data input
  WE => WE            -- 8-bit write enable input
);

-- End of RAMB36SDP_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// RAMB36SDP: 72x512 Simple Dual-Port BlockRAM /w ECC
//           Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

RAMB36SDP #(
  .SIM_MODE("SAFE"), // Simulation: "SAFE" vs. "FAST", see "Synthesis and Simulation Design Guide" for details
  .DO_REG(0),        // Optional output register (0 or 1)
  .EN_ECC_READ("FALSE"), // Enable ECC decoder, "TRUE" or "FALSE"
  .EN_ECC_WRITE("FALSE"), // Enable ECC encoder, "TRUE" or "FALSE"

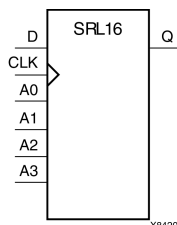
```

[illegible]

Virtex-5 ライブラリ ガイド (HDL 用)
UG621 (v11.3) 2009 年 9 月 16 日

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
);

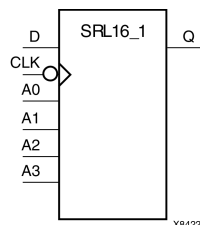
// End of SRL16_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

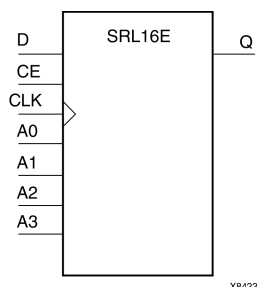
// End of SRL16_1_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	16 ビット値	すべてゼロ	コンフィギュレーション後のシフトレジスタと出力の初期値を指定

VHDL 記述 (インスタンス化)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
--      All FPGAs
-- Xilinx HDL Libraries Guide, version 11.2

SRL16E_inst : SRL16E
generic map (
  INIT => X"0000")
port map (
  Q => Q,          -- SRL data output
  A0 => A0,        -- Select[0] input
  A1 => A1,        -- Select[1] input
  A2 => A2,        -- Select[2] input
  A3 => A3,        -- Select[3] input
  CE => CE,        -- Clock enable input
  CLK => CLK,      -- Clock input
  D => D           -- SRL data input
);

-- End of SRL16E_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// SRL16E: 16-bit shift register LUT with clock enable operating on posedge of clock
//      All FPGAs
// Xilinx HDL Libraries Guide, version 11.2

SRL16E #(
    .INIT(16'h0000) // Initial Value of Shift Register
) SRL16E_inst (
    .Q(Q),           // SRL data output
    .A0(A0),         // Select[0] input
    .A1(A1),         // Select[1] input
    .A2(A2),         // Select[2] input
    .A3(A3),         // Select[3] input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

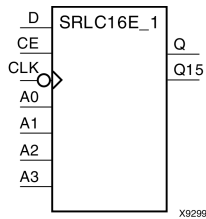
// End of SRL16E_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 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
);

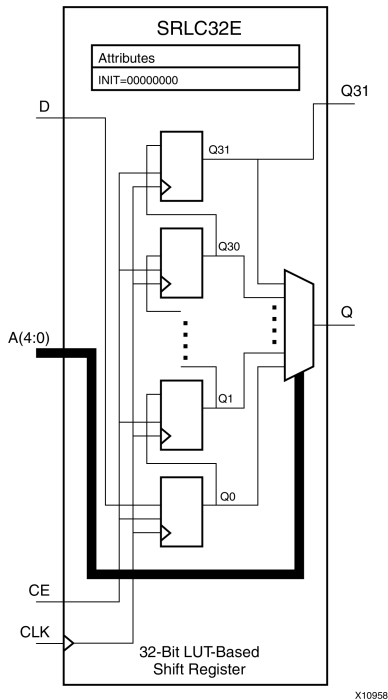
// End of SRL16E_1_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

SRLC32E

プリミティブ : 32 Clock Cycle, Variable Length Shift Register Look-Up Table (LUT) with Clock Enable



概要

このデザイン エLEMENTは、1 つのルックアップ テーブル (LUT) にインプリメントされている、可変長で 1 ～ 32 クロック サイクルのシフトレジスタです。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。このELEMENTは、アクティブ High のクロック イネーブルおよびカスケード機能も備えているため、複数の SRLC32E 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。

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
Q31	出力	1	シフトレジスタ カスケード出力 (後続 SRLC32E の D 入力に接続)
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	5	SRL のワード数のダイナミック選択 A=00000 ==> 1 ビット シフト長 A=11111 ==> 32 ビット シフト長

デザインの入力方法

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

インスタンス化する場合、このコンポーネントを次のように接続します。

- ・ CLK 入力を任意のクロックソースに、D 入力をシフト/格納するデータソースに、Q 出力を FDCPE 入力または FDRSE 入力などの適切なデスティネーションに接続します。
- ・ クロック イネーブル ピン (CE) はクロック イネーブル信号に接続するか、使用しない場合は論理値を 1 にします。
- ・ 5 ビット バス A は、一定の値 (0 ~ 31) にしてシフトレジスタの長さを 1 ~ 32 ビットに固定するか、または適切な論理値にしてシフトレジスタの長さを 1 ~ 32 ビットの範囲で変更することもできます。
- ・ シフトレジスタの長さを 32 ビットより大きくする場合は、Q31 出力ピンを後続の SRLC32E の D 入力に接続してカスケード接続します。
- ・ Q31 出力を SRLC32E 以外に接続することはできません。
- ・ Q 出力は、カスケード モードでも使用できます。
- ・ 32 ビットの 16 進数の INIT 属性で、シフトレジスタの初期シフトパターンを指定できます。
- ・ INIT[0] は、シフトアウトされる最初の値です。

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	SRLC32E の初期のシフトパターンを指定

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;

-- SRLC32E: 32-bit variable length shift register LUT
--      with clock enable
--      Virtex-5/6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

SRLC32E_inst : SRLC32E
generic map (
  INIT => X"00000000")
port map (
  Q => Q,          -- SRL data output
  Q31 => Q31,      -- SRL cascade output pin
  A => A,          -- 5-bit shift depth select input
  CE => CE,        -- Clock enable input
  CLK => CLK,      -- Clock input
  D => D           -- SRL data input
);

-- End of SRLC32E_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// SRLC32E: 32-bit variable length shift register LUT
//          with clock enable
//          Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

SRLC32E #(
    .INIT(32'h00000000) // Initial Value of Shift Register
) SRLC32E_inst (
    .Q(Q),           // SRL data output
    .Q31(Q31),       // SRL cascade output pin
    .A(A),           // 5-bit shift depth select input
    .CE(CE),         // Clock enable input
    .CLK(CLK),       // Clock input
    .D(D)            // SRL data input
);

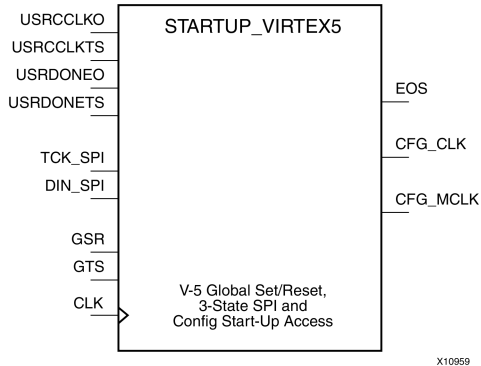
// End of SRLC32E_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)

STARTUP_VIRTEX5

プリミティブ : Virtex®-5 Configuration Start-Up Sequence Interface



X10959

概要

このデザイン エLEMENTは、グローバル非同期セット/リセット (GSR) 信号、グローバルトライステート (GTS) 専用配線、内部コンフィギュレーション信号、SPI PROM が使用される場合は SPI PROM の入力ピンなどへのロジックとデバイスピンの接続に使用されます。デバイスのコンフィギュレーションの終わりにスタートアップ シーケンスで別のクロックを使用するのを指定したり、コンフィギュレーション クロックを内部ロジックにアクセスさせるのにも使用されます。

ポートの説明

ポート名	方向	幅	機能
EOS	出力	1	コンフィギュレーションが終了するとアクティブ High になります。
CFGCLK	出力	1	コンフィギュレーションのメイン クロック出力
CFGMCLK	出力	1	コンフィギュレーションの内部オシレータのクロック出力
USRCCLKO	入力	1	内部ユーザー CCLK
USRCCLKTS	入力	1	内部ユーザー CCLK トライステート イネーブル
USRDONEO	入力	1	内部ユーザー DONE ピンの出力を制御
USRDONETS	入力	1	ユーザー DONE トライステート イネーブル
TCK_SPI	出力	1	SPI PROM コンフィギュレーションを使用した場合の TCK コンフィギュレーション ピンへの内部アクセス
DIN_SPI	出力	1	SPI PROM コンフィギュレーションを使用した場合の DIN コンフィギュレーション ピンへの内部アクセス
GSR	入力	1	アクティブ High GSR 信号
GTS	入力	1	アクティブ High GTS 信号
CLK	入力	1	ユーザー スタートアップ 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

専用のグローバルトライステートが使用される場合は、適切なソース ピンまたはロジックをこのプリミティブの GTS 入力ピンに接続します。コンフィギュレーションのスタートアップ シーケンスのクロックを指定するには、デザインからのクロックをこのデザイン エLEMENTの CLK ピンに接続します。CFGCLK および CFGCLK を使用すると、内部コンフィギュレーション クロックにアクセスでき、EOS 信号はコンフィギュレーション スタートアップ シーケンスの終了を伝えます。

SPI PROM を使用してデバイスをコンフィギュレーションする場合に、コンフィギュレーション後の SPI PROM へのアクセスが必要であれば、このコンポーネントの TCK_SPI ピンと DIN_SPI ピンを使用して、ほかの専用コンフィギュレーション入力ピンにアクセスできるようにします。

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;

-- STARTUP_VIRTEX5: Startup primitive for GSR, GTS or startup sequence control,
--                  SPI PROM pins, configuration clock and start-up status
--                  Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

STARTUP_VIRTEX5_inst : STARTUP_VIRTEX5
port map (
    CFGCLK => CFGCLK, -- Config logic clock 1-bit output
    CFGMCLK => CFGMCLK, -- Config internal osc clock 1-bit output
    DINSPI => DINSPI, -- DIN SPI PROM access 1-bit output
    EOS => EOS, -- End of Startup 1-bit output
    TCKSPI => TCKSPI, -- TCK SPI PROM access 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)
    USRCCLKO => USRCCLKO, -- User CCLK 1-bit input
    USRCCLKTS => USRCCLKTS, -- User CCLK 3-state, 1-bit input
    USRDONEO => USRDONEO, -- User Done 1-bit input
    USRDONETS => USRDONETS -- User Done 3-state, 1-bit input
);

-- End of STARTUP_VIRTEX5_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// STARTUP_VIRTEX5: Startup primitive for accessing GSR, GTS, startup sequence
//                      control, SPI PROM pins, configuration clock and start-up status
//                      Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

STARTUP_VIRTEX5 STARTUP_VIRTEX5_inst (
    .CFGCLK(CFGCLK),           // Config logic clock 1-bit output
    .CFGMCLK(CFGMCLK),        // Config internal osc clock 1-bit output
    .DINSPI(DINSPI),          // DIN SPI PROM access 1-bit output
    .EOS(EOS),                // End Of Startup 1-bit output
    .TCKSPI(TCKSPI),          // TCK SPI PROM access 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)
    .USRCCLKO(USRCCLKO),      // User CCLK 1-bit input
    .USRCCLKTS(USRCCLKTS),    // User CCLK 3-state 1-bit input
    .USRDONEO(USRDONEO),      // User Done 1-bit input
    .USRDONETS(USRDONETS)     // User Done 3-state 1-bit input
);

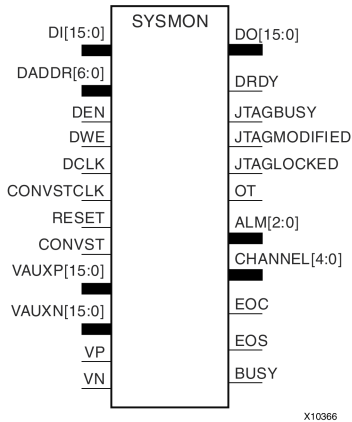
// End of STARTUP_VIRTEX5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)
- ・ [Virtex-5 FPGA ユーザー ガイド](#)

SYSMON

プリミティブ : System Monitor



概要

このデザイン エLEMENTは、10 ビット、200kSPS (キロサンプル/秒) の Analog-to-Digital Converter (ADC) をベースに構築されています。ADC は、大量のオンチップ センサーと組み合わせて、オンチップ電源電圧およびチップ温度などの FPGA の物理的な動作パラメータを計測するのに使用されます。外部電圧には、専用のアナログ入力ペア (VP/VN) と 16 のユーザーが選択可能なアナログ入力 (補助アナログ入力 (VAUXP[15:0]、VAUXN[15:0])) を介します。外部アナログ入力を使用すると、ADC でボードやエンクロージャの物理的環境を監視できます。

ポートの説明

ポート名	タイプ	幅	機能
ALM[2:0]	出力	3	温度、Vccint および Vccaux の 3 ビット出力アラーム
BUSY	出力	1	1 ビット出力 ADC ビジー信号
CHANNEL[4:0]	出力	5	5 ビット出力チャネル選択
CONVST	入力	1	1 ビット入力変換開始
CONVSTCLK	入力	1	1 ビット入力変換開始クロック
DADDR[6:0]	入力	7	ダイナミック リコンフィギュレーションの 7 ビット入力アドレス バス
DCLK	入力	1	ダイナミック リコンフィギュレーションの 1 ビット入力クロック
DEN	入力	1	ダイナミック リコンフィギュレーションの 1 ビット入力イネーブル
DI[15:0]	入力	16	ダイナミック リコンフィギュレーションの 16 ビット入力データ バス
DO[15:0]	出力	16	ダイナミック リコンフィギュレーションの 16 ビット出力データ バス
DRDY	出力	1	ダイナミック リコンフィギュレーションの 1 ビット出力データレディ
DWE	入力	1	ダイナミック リコンフィギュレーションの 1 ビット入力ライト イネーブル

ポート名	タイプ	幅	機能
EOC	出力	1	変換の 1 ビット出力エンド
EOS	出力	1	シーケンスの 1 ビット出力エンド
JTAGBUSY	出力	1	1 ビット出力 JTAG DRP ビジー
JTAGLOCKED	出力	1	1 ビット出力 DRP ポート ロック
JTAGMODIFIED	出力	1	DRP への 1 ビット出力 JTAG 書き込み
OT	出力	1	温度アラームの 1 ビット出力
RESET	入力	1	1 ビット入力アクティブ High リセット
VAUXN[15:0]	入力	16	16 ビット入力 N 側補助アナログ入力
VAUXP[15:0]	入力	16	16 ビット入力 P 側補助アナログ入力
VN	入力	1	1 ビット入力 N 側アナログ入力
VP	入力	1	1 ビット入力 P 側アナログ入力

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	不可
マクロのサポート	不可

該当する入力および出力ポートをすべて接続し、このコンポーネントの該当するビヘイビアの属性を設定します。シミュレーションには、アナログおよび温度情報をモデルに渡すためにテキストファイルを使用します。テキストファイルのフォーマットは次のとおりです。

```
// Must use valid headers on all columns
// Comments can be added to the stimulus file using '///'
TIME TEMP VCCAUX VCCINT VP VN VAUXP[0] VAUXN[0]
00000 45 2.5 1.0 0.5 0.0 0.7 0.0
05000 85 2.45 1.1 0.3 0.0 0.2 0.0
// Time stamp data is in nano seconds (ns)
// Temperature is recorded in C (degrees centigrade)
// All other channels are recorded as V (Volts)
// Valid column headers are:
// TIME, TEMP, VCCAUX, VCCINT, VP, VN,
// VAUXP[0], VAUXN[0],.....VAUXP[15], VAUXN[15]
// External analog inputs are differential so VP = 0.5 and VN = 0.0 the
// input on channel VP/VN is 0.5 - 0.0 = 0.5V
```

メモ: このコードをコンパイルする場合は、テキストに余分なスペースを追加しないでください。コンパイル エラーが発生する場合があります。

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_40	16 進数	16'h0000 ~ 16'hffff	16'h0000	コンフィギュレーション レジスタ 0
INIT_41	16 進数	16'h0000 ~ 16'hffff	16'h0000	コンフィギュレーション レジスタ 1

属性	タイプ	値	デフォルト	説明
INIT_42	16 進数	16'h0000 ~ 16'hffff	16'h0800	コンフィギュレーション レジスタ 2
INIT_43	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 0
INIT_44	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 1
INIT_45	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 2
INIT_46	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 3
INIT_47	16 進数	16'h0000 ~ 16'hffff	16'h0000	テスト レジスタ 4
INIT_48	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 0
INIT_49	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 1
INIT_4A	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 2
INIT_4B	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 3
INIT_4C	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 4
INIT_4D	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 5
INIT_4E	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 6
INIT_4F	16 進数	16'h0000 ~ 16'hffff	16'h0000	シーケンス レジスタ 7
INIT_50	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 0
INIT_51	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 1
INIT_52	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 2
INIT_53	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 3
INIT_54	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 4
INIT_55	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 5
INIT_56	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 6
INIT_57	16 進数	16'h0000 ~ 16'hffff	16'h0000	アラーム制限レジスタ 7

属性	タイプ	値	デフォルト	説明
SIM_DEVICE	文字列	VIRTEX5、 VIRTEX6	VIRTEX5	DRP クロック (DCLK) と低周波数の ADC クロック (ADCCLOCK) との最小分周率を 8 から 2 に変更したものをシミュレーション モデルに反映させることができる。
SIM_MONITOR_FILE	文字列	0 ビット文字列	design.txt	シミュレーション アナログ入力ファイル

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;

-- SYSMON: System Monitor
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

SYSMON_inst : SYSMON
generic map (
    INIT_40 => X"0000",      -- Configuration register 0
    INIT_41 => X"0000",      -- Configuration register 1
    INIT_42 => X"0800",      -- Configuration register 2
    INIT_43 => X"0000",      -- Test register 0
    INIT_44 => X"0000",      -- Test register 1
    INIT_45 => X"0000",      -- Test register 2
    INIT_46 => X"0000",      -- Test register 3
    INIT_47 => X"0000",      -- Test register 4
    INIT_48 => X"0000",      -- Sequence register 0
    INIT_49 => X"0000",      -- Sequence register 1
    INIT_4A => X"0000",      -- Sequence register 2
    INIT_4B => X"0000",      -- Sequence register 3
    INIT_4C => X"0000",      -- Sequence register 4
    INIT_4D => X"0000",      -- Sequence register 5
    INIT_4E => X"0000",      -- Sequence register 6
    INIT_4F => X"0000",      -- Sequence register 7
    INIT_50 => X"0000",      -- Alarm limit register 0
    INIT_51 => X"0000",      -- Alarm limit register 1
    INIT_52 => X"0000",      -- Alarm limit register 2
    INIT_53 => X"0000",      -- Alarm limit register 3
    INIT_54 => X"0000",      -- Alarm limit register 4
    INIT_55 => X"0000",      -- Alarm limit register 5
    INIT_56 => X"0000",      -- Alarm limit register 6
    INIT_57 => X"0000",      -- Alarm limit register 7
    SIM_DEVICE => "VIRTEX5", -- Device family selection
    SIM_MONITOR_FILE => "design.txt" -- Simulation analog entry file
)
port map (
    ALM => ALM,      -- 3-bit output alarm for temp, Vccint and Vccaux
    BUSY => BUSY,    -- 1-bit output ADC busy signal
    CHANNEL => CHANNEL, -- 5-bit output channel selection
    DO => DO,        -- 16-bit output data bus for dynamic reconfig
    DRDY => DRDY,    -- 1-bit output data ready for dynamic reconfig
    EOC => EOC,      -- 1-bit output end of conversion
    EOS => EOS,      -- 1-bit output end of sequence
    JTAGBUSY => JTAGBUSY, -- 1-bit output JTAG DRP busy
    JTAGLOCKED => JTAGLOCKED, -- 1-bit output DRP port lock
    JTAGMODIFIED => JTAGMODIFIED, -- 1-bit output JTAG write to DRP
    OT => OT,        -- 1-bit output over temperature alarm
    CONVST => CONVST, -- 1-bit input convert start
    CONVSTCLK => CONVSTCLK, -- 1-bit input convert start clock
    DADDR => DADDR,  -- 7-bit input address bus for dynamic reconfig
    DCLK => DCLK,    -- 1-bit input clock for dynamic reconfig
    DEN => DEN,      -- 1-bit input enable for dynamic reconfig
    DI => DI,        -- 16-bit input data bus for dynamic reconfig
    DWE => DWE,      -- 1-bit input write enable for dynamic reconfig
    RESET => RESET,  -- 1-bit input active high reset
    VAUXN => VAUXN,  -- 16-bit input N-side auxiliary analog input
    VAUXP => VAUXP,  -- 16-bit input P-side auxiliary analog input

```

```

VN => VN,                -- 1-bit input N-side analog input
VP => VP                -- 1-bit input P-side analog input
);

-- End of SYSMON_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// SYSMON: System Monitor
//      Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

SYSMON #(
    .INIT_40(16'h0), // Configuration register 0
    .INIT_41(16'h0), // Configuration register 1
    .INIT_42(16'h0), // Configuration register 2
    .INIT_43(16'h0), // Test register 0
    .INIT_44(16'h0), // Test register 1
    .INIT_45(16'h0), // Test register 2
    .INIT_46(16'h0), // Test register 3
    .INIT_47(16'h0), // Test register 4
    .INIT_48(16'h0), // Sequence register 0
    .INIT_49(16'h0), // Sequence register 1
    .INIT_4A(16'h0), // Sequence register 2
    .INIT_4B(16'h0), // Sequence register 3
    .INIT_4C(16'h0), // Sequence register 4
    .INIT_4D(16'h0), // Sequence register 5
    .INIT_4E(16'h0), // Sequence register 6
    .INIT_4F(16'h0), // Sequence register 7
    .INIT_50(16'h0), // Alarm limit register 0
    .INIT_51(16'h0), // Alarm limit register 1
    .INIT_52(16'h0), // Alarm limit register 2
    .INIT_53(16'h0), // Alarm limit register 3
    .INIT_54(16'h0), // Alarm limit register 4
    .INIT_55(16'h0), // Alarm limit register 5
    .INIT_56(16'h0), // Alarm limit register 6
    .INIT_57(16'h0), // Alarm limit register 7
    .STM_MONITOR_FILE("design.txt") // Simulation analog entry file
) SYSMON_inst (
    .ALM(ALM),                // 3-bit output for temp, Vccint and Vccaux
    .BUSY(BUSY),              // 1-bit output ADC busy signal
    .CHANNEL(CHANNEL),        // 5-bit output channel selection
    .DO(DO),                  // 16-bit output data bus for dynamic reconfig
    .DRDY(DRDY),              // 1-bit output data ready for dynamic reconfig
    .EOC(EOC),                // 1-bit output end of conversion
    .EOS(EOS),                // 1-bit output end of sequence
    .JTAGBUSY(JTAGBUSY),      // 1-bit output JTAG DRP busy
    .JTAGLOCKED(JTAGLOCKED), // 1-bit output DRP port lock
    .JTAGMODIFIED(JTAGMODIFIED), // 1-bit output JTAG write to DRP
    .OT(OT),                  // 1-bit output over temperature alarm
    .CONVST(CONVST),          // 1-bit input convert start
    .CONVSTCLK(CONVSTCLK),    // 1-bit input convert start clock
    .DADDR(DADDR),            // 7-bit input address bus for dynamic reconfig
    .DCLK(DCLK),              // 1-bit input clock for dynamic reconfig
    .DEN(DEN),                // 1-bit input enable for dynamic reconfig
    .DI(DI),                  // 16-bit input data bus for dynamic reconfig
    .DWE(DWE),                // 1-bit input write enable for dynamic reconfig
    .RESET(RESET),            // 1-bit input active high reset
    .VAUXN(VAUXN),            // 16-bit input N-side auxiliary analog input
    .VAUXP(VAUXP),            // 16-bit input P-side auxiliary analog input
    .VN(VN),                  // 1-bit input N-side analog input
    .VP(VP),                  // 1-bit input P-side analog input
);

// End of SYSMON_inst instantiation

```

詳細情報

- [Virtex-5 FPGA ユーザー ガイド](#)
- [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)

TEMAC

プリミティブ : Tri-mode Ethernet Media Access Controller (MAC)

概要

このデザイン エLEMENTには、イーサネットシステムの接続の条件を満たすため、別々にコンフィギュレーション可能なエンベデッド イーサネット MAC が 1 ペア含まれます。

デザインの入力方法

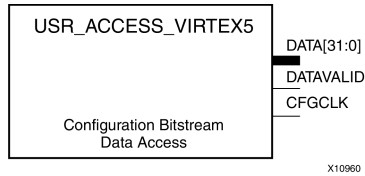
インスタンス化	不可
推論	不可
CORE Generator™ およびウィザード	推奨
マクロのサポート	不可

詳細情報

- ・ [Virtex-5 エンベデッドトライモード イーサネット MAC ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート : DC 特性およびスイッチ特性](#)
- ・ [Virtex-5 FPGA ユーザー ガイド](#)

USR_ACCESS_VIRTEX5

プリミティブ : Virtex-5 User Access Register



概要

このデザイン エLEMENTを使用すると、コンフィギュレーション ロジック内の 32 ビットのレジスタにアクセスでき、ビットストリームからのデータを読み出すことができます。たとえば、コンフィギュレーション後に FPGA デザインからビットストリーム格納ソースに保存されたデータにアクセスすることができます。

ポートの説明

ポート名	方向	幅	機能
DATA	出力	32	コンフィギュレーション出力データ
DATAVALID	出力	1	有効なデータを含むアクティブ High の DATA ポート
CFGCLK	出力	1	コンフィギュレーション 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

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;

-- USR_ACCESS_VIRTEX5: Configuration Data Memory Access Port
-- Virtex-5
-- Xilinx HDL Libraries Guide, version 11.2

USR_ACCESS_VIRTEX5_inst : USR_ACCESS_VIRTEX5
port map (
    CFGCLK => CFGCLK, -- 1-bit configuration clock output
    DATA => DATA,    -- 32-bit config data output
    DATAVALID => DATAVALID -- 1-bit data valid output
);

-- End of USR_ACCESS_VIRTEX5_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// USR_ACCESS_VIRTEX5: Configuration Data Memory Access Port
//                               Virtex-5
// Xilinx HDL Libraries Guide, version 11.2

USR_ACCESS_VIRTEX5 USR_ACCESS_VIRTEX5_inst (
    .CFGCLK(CFGCLK),           // 1-bit configuration clock output
    .DATA(DATA),               // 32-bit config data output
    .DATAVALID(DATAVALID) // 1-bit data valid output
);

// End of USR_ACCESS_VIRTEX5_inst instantiation
```

詳細情報

- ・ [Virtex-5 FPGA ユーザー ガイド](#)
- ・ [Virtex-5 FPGA データシート: DC 特性およびスイッチ特性](#)