

# Virtex-6 ライブラリ ガイド (HDL 用)

UG623 (v11.4) 2009 年 12 月 2 日

## 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.

本資料は英語版 (v.11.4) を翻訳したもので、内容に相違が生じる場合には原文を優先します。  
資料によっては英語版の更新に対応していないものがあります。  
日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

# 目次

ザイリンクス商標および著作権情報 .....	2
1 : このマニュアルについて .....	15
デザインの入力方法 .....	15
2 : UniMacro について .....	17
BRAM_SDP_MACRO .....	18
概要 .....	18
ポートの説明 .....	18
コンフィギュレーション表 .....	19
デザインの入力方法 .....	19
使用可能な属性 .....	19
詳細情報 .....	26
BRAM_SINGLE_MACRO .....	27
概要 .....	27
ポートの説明 .....	27
コンフィギュレーション表 .....	28
デザインの入力方法 .....	30
使用可能な属性 .....	30
詳細情報 .....	35
BRAM_TDP_MACRO .....	36
概要 .....	36
ポートの説明 .....	36
コンフィギュレーション表 .....	37
デザインの入力方法 .....	39
使用可能な属性 .....	39
詳細情報 .....	45
FIFO_DUALCLOCK_MACRO .....	46
概要 .....	46
ポートの説明 .....	46
コンフィギュレーション表 .....	47
デザインの入力方法 .....	47
使用可能な属性 .....	48
詳細情報 .....	49
FIFO_SYNC_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
ADDMACC_MACRO .....	54
概要 .....	54
ポートの説明 .....	54
デザインの入力方法 .....	55
使用可能な属性 .....	55
詳細情報 .....	56
ADDSUB_MACRO .....	57
概要 .....	57
ポートの説明 .....	57
デザインの入力方法 .....	57
使用可能な属性 .....	58
詳細情報 .....	59
COUNTER_LOAD_MACRO .....	60
概要 .....	60
ポートの説明 .....	60
デザインの入力方法 .....	60

使用可能な属性.....	61
詳細情報 .....	62
COUNTER_TC_MACRO .....	63
概要.....	63
ポートの説明 .....	63
デザインの入力方法.....	63
使用可能な属性.....	64
詳細情報 .....	65
EQ_COMPARE_MACRO.....	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 .....	68
MACC_MACRO .....	69
概要.....	69
ポートの説明 .....	69
デザインの入力方法.....	70
使用可能な属性.....	70
詳細情報 .....	71
DSP_MULT_MACRO.....	72
概要.....	72
ポートの説明 .....	72
デザインの入力方法.....	72
使用可能な属性.....	73
詳細情報 .....	74
3 : ファンクション別分類.....	75
4 : デザイン エレメント .....	81
AND2B1L .....	82
概要.....	82
論理表.....	82
ポートの説明 .....	82
デザインの入力方法.....	82
詳細情報 .....	82
BSCAN_VIRTEX6.....	83
概要.....	83
ポートの説明 .....	84
デザインの入力方法.....	84
使用可能な属性.....	84
詳細情報 .....	85
BUFG .....	86
概要.....	86
ポートの説明 .....	86
デザインの入力方法.....	86
詳細情報 .....	87
BUFGCE .....	88
概要.....	88
論理表.....	88
デザインの入力方法.....	88
詳細情報 .....	89
BUFGCE_1 .....	90
概要.....	90
論理表.....	90
デザインの入力方法.....	90
詳細情報 .....	91
BUFGCTRL .....	92
概要.....	92
ポートの説明 .....	92
デザインの入力方法.....	93

使用可能な属性.....	93
詳細情報 .....	93
BUFGMUX .....	94
概要.....	94
論理表.....	94
ポートの説明 .....	94
デザインの入力方法.....	94
使用可能な属性.....	95
詳細情報 .....	95
BUFGMUX_1.....	96
概要.....	96
論理表.....	96
デザインの入力方法.....	96
詳細情報 .....	97
BUFGMUX_CTRL .....	98
概要.....	98
ポートの説明 .....	98
デザインの入力方法.....	98
詳細情報 .....	99
BUFGP.....	100
概要.....	100
デザインの入力方法.....	100
詳細情報 .....	100
BUFH .....	101
概要.....	101
ポートの説明 .....	101
デザインの入力方法.....	101
詳細情報 .....	101
BUFHCE .....	102
概要.....	102
ポートの説明 .....	102
デザインの入力方法.....	102
使用可能な属性.....	102
詳細情報 .....	103
BUFIO .....	104
概要.....	104
ポートの説明 .....	104
デザインの入力方法.....	104
詳細情報 .....	105
BUFIODQS.....	106
概要.....	106
ポートの説明 .....	106
デザインの入力方法.....	106
使用可能な属性.....	106
詳細情報 .....	107
BUFR.....	108
概要.....	108
ポートの説明 .....	108
デザインの入力方法.....	108
使用可能な属性.....	108
詳細情報 .....	109
CAPTURE_VIRTEX6 .....	110
概要.....	110
ポートの説明 .....	110
デザインの入力方法.....	110
使用可能な属性.....	110
詳細情報 .....	111
CARRY4.....	112
概要.....	112

ポートの説明 .....	112
デザインの入力方法 .....	113
詳細情報 .....	113
CFG_LUT5 .....	114
概要 .....	114
ポートの説明 .....	114
デザインの入力方法 .....	115
使用可能な属性 .....	115
詳細情報 .....	116
DCI_RESET .....	117
概要 .....	117
ポートの説明 .....	117
デザインの入力方法 .....	117
使用可能な属性 .....	117
詳細情報 .....	118
DNA_PORT .....	119
概要 .....	119
ポートの説明 .....	119
デザインの入力方法 .....	119
使用可能な属性 .....	120
詳細情報 .....	120
DSP48E1 .....	121
概要 .....	121
ポートの説明 .....	122
デザインの入力方法 .....	124
使用可能な属性 .....	124
詳細情報 .....	129
EFUSE_USR .....	130
概要 .....	130
ポートの説明 .....	130
デザインの入力方法 .....	130
使用可能な属性 .....	130
詳細情報 .....	131
FDCE .....	132
概要 .....	132
論理表 .....	132
デザインの入力方法 .....	132
使用可能な属性 .....	132
詳細情報 .....	133
FDPE .....	134
概要 .....	134
論理表 .....	134
デザインの入力方法 .....	134
使用可能な属性 .....	134
詳細情報 .....	134
FDRE .....	135
概要 .....	135
論理表 .....	135
デザインの入力方法 .....	135
使用可能な属性 .....	135
詳細情報 .....	135
FDSE .....	136
概要 .....	136
論理表 .....	136
デザインの入力方法 .....	136
使用可能な属性 .....	136
詳細情報 .....	136
FIFO18E1 .....	137
概要 .....	137

ポートの説明 .....	137
デザインの入力方法 .....	138
使用可能な属性 .....	138
詳細情報 .....	140
FIFO36E1 .....	141
概要 .....	141
ポートの説明 .....	141
デザインの入力方法 .....	142
使用可能な属性 .....	143
詳細情報 .....	145
FRAME_ECC_VIRTEX6 .....	146
概要 .....	146
ポートの説明 .....	146
デザインの入力方法 .....	147
使用可能な属性 .....	147
詳細情報 .....	148
GTHE1_QUAD .....	149
概要 .....	150
デザインの入力方法 .....	150
詳細情報 .....	150
GTXE1 .....	151
概要 .....	152
デザインの入力方法 .....	152
詳細情報 .....	152
IBUF .....	153
概要 .....	153
ポートの説明 .....	153
デザインの入力方法 .....	153
使用可能な属性 .....	153
詳細情報 .....	154
IBUFDS .....	155
概要 .....	155
論理表 .....	155
ポートの説明 .....	155
デザインの入力方法 .....	155
使用可能な属性 .....	155
詳細情報 .....	156
IBUFDS_GTHE1 .....	157
概要 .....	157
デザインの入力方法 .....	157
詳細情報 .....	157
IBUFDS_GTXE1 .....	158
概要 .....	158
デザインの入力方法 .....	158
詳細情報 .....	158
IBUFG .....	159
概要 .....	159
ポートの説明 .....	159
デザインの入力方法 .....	159
使用可能な属性 .....	159
詳細情報 .....	160
IBUFGDS .....	161
概要 .....	161
論理表 .....	161
ポートの説明 .....	161
デザインの入力方法 .....	161
使用可能な属性 .....	162
詳細情報 .....	162
ICAP_VIRTEX6 .....	163

概要.....	163
ポートの説明 .....	163
デザインの入力方法.....	163
使用可能な属性.....	163
詳細情報 .....	164
IDDR .....	165
概要.....	165
ポートの説明 .....	165
デザインの入力方法.....	166
使用可能な属性.....	166
詳細情報 .....	167
IDDR_2CLK .....	168
概要.....	168
ポートの説明 .....	168
デザインの入力方法.....	169
使用可能な属性.....	169
詳細情報 .....	170
IOBUF .....	171
概要.....	171
論理表.....	171
ポートの説明 .....	171
デザインの入力方法.....	171
使用可能な属性.....	172
詳細情報 .....	173
IOBUFDS .....	174
概要.....	174
論理表.....	174
ポートの説明 .....	174
デザインの入力方法.....	174
使用可能な属性.....	175
詳細情報 .....	175
IODELAYE1.....	176
概要.....	176
ポートの説明 .....	176
デザインの入力方法.....	177
使用可能な属性.....	177
詳細情報 .....	179
ISERDESE1 .....	180
概要.....	180
ポートの説明 .....	180
デザインの入力方法.....	181
使用可能な属性.....	181
詳細情報 .....	185
KEEPER.....	186
概要.....	186
ポートの説明 .....	186
デザインの入力方法.....	186
詳細情報 .....	187
LDCE.....	188
概要.....	188
論理表.....	188
デザインの入力方法.....	188
使用可能な属性.....	188
詳細情報 .....	188
LDPE.....	189
概要.....	189
論理表.....	189
デザインの入力方法.....	189
使用可能な属性.....	189



詳細情報 .....	189
LUT5 .....	190
概要 .....	190
論理表 .....	191
ポートの説明 .....	192
デザインの入力方法 .....	192
使用可能な属性 .....	192
詳細情報 .....	193
LUT5_D .....	194
概要 .....	194
論理表 .....	195
ポートの説明 .....	196
デザインの入力方法 .....	196
使用可能な属性 .....	196
詳細情報 .....	197
LUT5_L .....	198
概要 .....	198
論理表 .....	199
ポートの説明 .....	200
デザインの入力方法 .....	200
使用可能な属性 .....	200
詳細情報 .....	201
LUT6 .....	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 .....	205
LUT6_2 .....	206
概要 .....	206
論理表 .....	206
ポートの説明 .....	208
デザインの入力方法 .....	208
使用可能な属性 .....	209
詳細情報 .....	209
LUT6_D .....	210
概要 .....	210
論理表 .....	210
ポートの説明 .....	212
デザインの入力方法 .....	213
使用可能な属性 .....	213
詳細情報 .....	213
LUT6_L .....	214
概要 .....	214
論理表 .....	214
ポートの説明 .....	216
デザインの入力方法 .....	217
使用可能な属性 .....	217
詳細情報 .....	217
MMCM_ADV .....	218
概要 .....	218
ポートの説明 .....	218
デザインの入力方法 .....	220
使用可能な属性 .....	220
詳細情報 .....	224
MMCM_BASE .....	225
概要 .....	225
ポートの説明 .....	225

デザインの入力方法.....	226
使用可能な属性.....	226
詳細情報 .....	229
MUXF7 .....	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
詳細情報 .....	231
MUXF7_D .....	232
概要.....	232
論理表.....	232
ポートの説明 .....	232
デザインの入力方法.....	232
詳細情報 .....	233
MUXF7_L .....	234
概要.....	234
論理表.....	234
ポートの説明 .....	234
デザインの入力方法.....	234
詳細情報 .....	235
MUXF8 .....	236
概要.....	236
論理表.....	236
ポートの説明 .....	236
デザインの入力方法.....	236
詳細情報 .....	237
MUXF8_D .....	238
概要.....	238
論理表.....	238
ポートの説明 .....	238
デザインの入力方法.....	238
詳細情報 .....	239
MUXF8_L .....	240
概要.....	240
論理表.....	240
ポートの説明 .....	240
デザインの入力方法.....	240
詳細情報 .....	241
OBUF .....	242
概要.....	242
ポートの説明 .....	242
デザインの入力方法.....	242
使用可能な属性.....	242
詳細情報 .....	243
OBUFDS .....	244
概要.....	244
論理表.....	244
ポートの説明 .....	244
デザインの入力方法.....	244
使用可能な属性.....	244
詳細情報 .....	245
OBUFTDS .....	246
概要.....	246
論理表.....	246
ポートの説明 .....	246
デザインの入力方法.....	246
使用可能な属性.....	246
詳細情報 .....	247

ODDR.....	248
概要.....	248
ポートの説明.....	248
デザインの入力方法.....	248
使用可能な属性.....	249
詳細情報.....	249
OR2L.....	250
概要.....	250
論理表.....	250
ポートの説明.....	250
デザインの入力方法.....	250
詳細情報.....	250
OSERDESE1.....	251
概要.....	251
ポートの説明.....	251
デザインの入力方法.....	253
使用可能な属性.....	253
詳細情報.....	255
PCIE_2.0.....	256
概要.....	257
デザインの入力方法.....	257
詳細情報.....	257
PULLDOWN.....	258
概要.....	258
ポートの説明.....	258
デザインの入力方法.....	258
詳細情報.....	259
PULLUP.....	260
概要.....	260
ポートの説明.....	260
デザインの入力方法.....	260
詳細情報.....	261
RAM128X1D.....	262
概要.....	262
ポートの説明.....	262
デザインの入力方法.....	263
使用可能な属性.....	263
詳細情報.....	264
RAM256X1S.....	265
概要.....	265
ポートの説明.....	265
デザインの入力方法.....	265
使用可能な属性.....	266
詳細情報.....	266
RAM32M.....	267
概要.....	267
ポートの説明.....	268
デザインの入力方法.....	268
使用可能な属性.....	269
詳細情報.....	270
RAM32X1S.....	271
概要.....	271
論理表.....	271
デザインの入力方法.....	271
使用可能な属性.....	272
詳細情報.....	272
RAM64M.....	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
RAM64X1D .....	277
概要.....	277
論理表.....	277
デザインの入力方法.....	278
使用可能な属性.....	278
詳細情報 .....	279
RAM64X1S .....	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.....	280
詳細情報 .....	281
RAMB18E1 .....	282
概要.....	282
ポートの説明 .....	282
デザインの入力方法.....	284
使用可能な属性.....	284
詳細情報 .....	291
RAMB36E1 .....	292
概要.....	292
ポートの説明 .....	293
デザインの入力方法.....	294
使用可能な属性.....	295
詳細情報 .....	304
SRL16E .....	305
概要.....	305
論理表.....	305
ポートの説明 .....	306
デザインの入力方法.....	306
使用可能な属性.....	306
詳細情報 .....	307
SRLC32E .....	308
概要.....	308
ポートの説明 .....	308
デザインの入力方法.....	309
使用可能な属性.....	309
詳細情報 .....	310
STARTUP_VIRTEX6 .....	311
概要.....	311
ポートの説明 .....	312
デザインの入力方法.....	312
使用可能な属性.....	313
詳細情報 .....	314
SYSMON .....	315
概要.....	315
ポートの説明 .....	315
デザインの入力方法.....	316
使用可能な属性.....	316
詳細情報 .....	319
TEMAC_SINGLE .....	320
概要.....	320
デザインの入力方法.....	321
詳細情報 .....	321
USR_ACCESS_VIRTEX6 .....	322
概要.....	322
ポートの説明 .....	322

デザインの入力方法.....	322
詳細情報 .....	323



# 第 1 章

## このマニュアルについて

---

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

このマニュアルには、次の内容が含まれます。

- ・ 概要
- ・ このアーキテクチャでサポートされるプリミティブとマクロのファンクション別リスト
- ・ 各プリミティブの詳細説明

## デザインの入力方法

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。

- ・ **インスタンスエーション**：デザインにコンポーネントが直接インスタンスエートされます。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エートしにくいようなプリミティブをインスタンスエート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

# UniMacro について

---

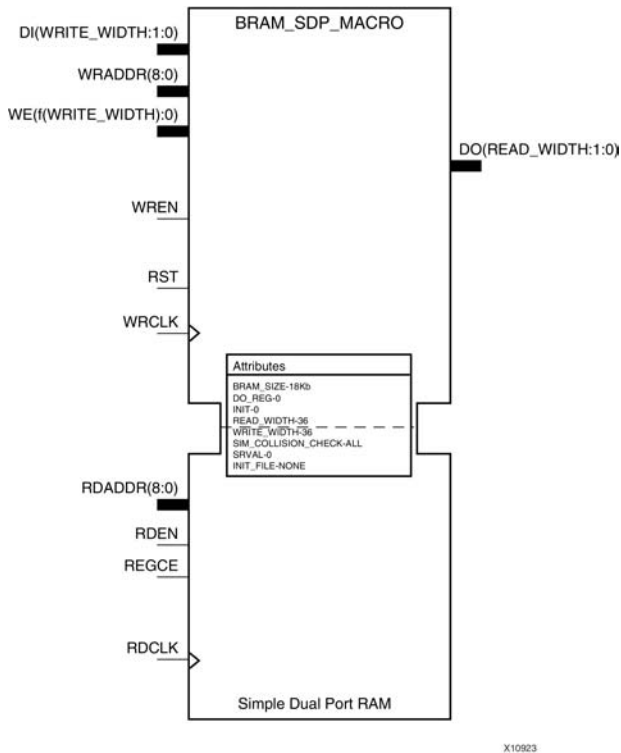
このセクションでは、このアーキテクチャで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

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



## 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。

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"> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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。</li> </ul> <p><b>メモ:</b> 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-6 ライブラリ ガイド (HDL 用)  
UG623 (v11.4) 2009 年 12 月 2 日

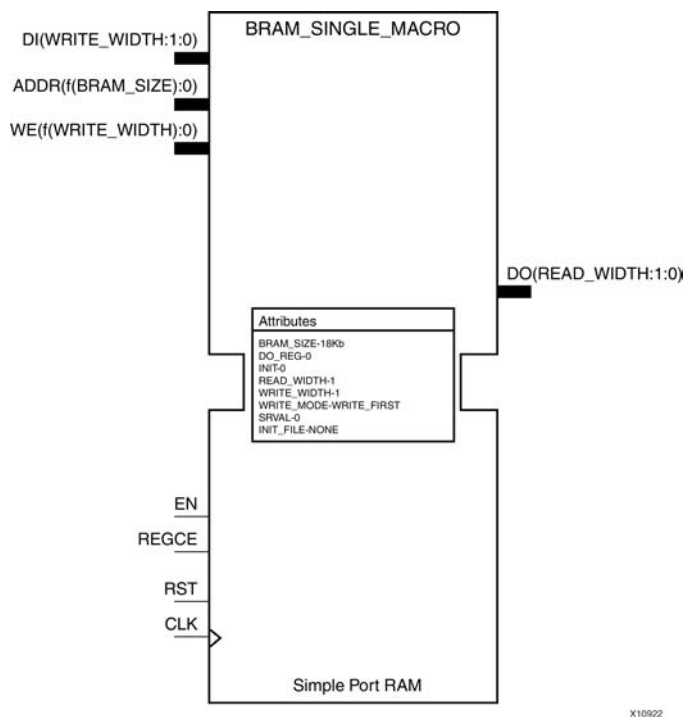
```
.WRADDR(WRADDR), // Input write address  
.WRCLK(WRCLK),   // Input write clock  
.WREN(WREN)      // Input write port enable  
);  
  
// End of BRAM_SDP_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BRAM\_SINGLE\_MACRO

マクロ : Single Port RAM



X10922

## 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-6 ライブラリ ガイド (HDL 用)  
UG623 (v11.4) 2009 年 12 月 2 日

[illegible]



```
-- 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

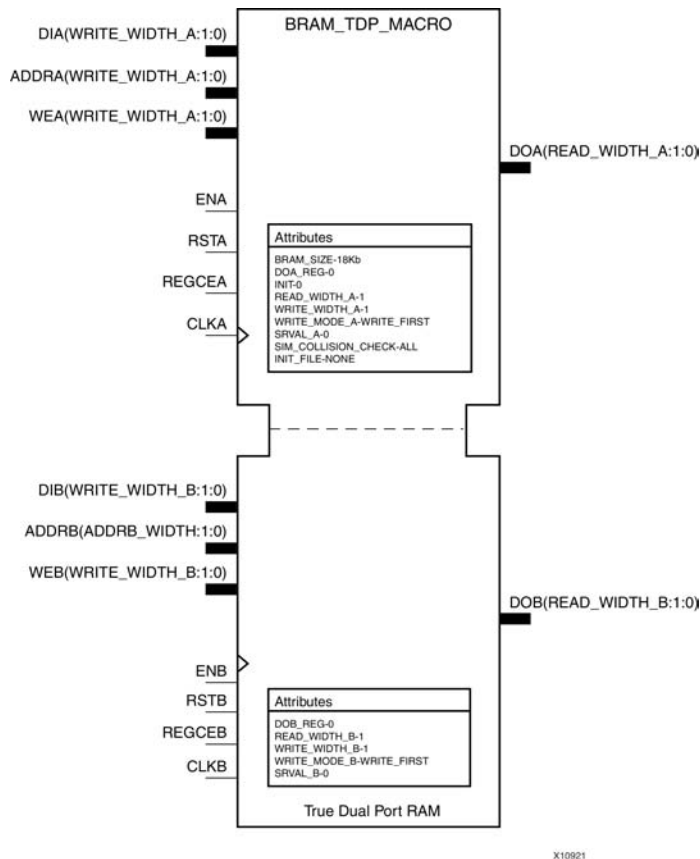
BRAM_SINGLE_MACRO #(
  .BRAM_SIZE("18Kb"), // Target BRAM, "18Kb" or "36Kb"
  .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  .DO_REG(0), // Optional output register (0 or 1)
  .INIT(36'h0000000000), // Initial values on output port
  .INIT_FILE("NONE"),
  .WRITE_WIDTH(0), // 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'h0000000000), // 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"> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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。</li> </ul> <p><b>メモ:</b> 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",
```



41

```
INIT_5B => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5C => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5D => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5E => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_5F => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_60 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_61 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_62 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_63 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_64 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_65 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_66 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_67 => X"0000000000000000000000000000000000000000000000000000000000000000",
INIT_68 => X"0000000000000000000000000000000000000000000000000000000000000000",
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",

-- 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 (
  DOA => DOA,      -- Output port-A data
  DOB => DOB,      -- Output port-B data
  ADDRA => ADDRA,   -- Input port-A address
  ADDRb => ADDRb,   -- Input port-B address
  CLKA => CLKA,     -- Input port-A clock
  CLKB => CLKB,     -- Input port-B clock
  DIA => DIA,      -- Input port-A data
  DIB => DIB,      -- Input port-B data
  ENA => ENA,      -- Input port-A enable
  ENB => ENB,      -- Input port-B enable
  REGCEA => REGCEA, -- Input port-A output register enable
  REGCEB => REGCEB, -- Input port-B output register enable
  RSTA => RSTA,     -- Input port-A reset
  RSTB => RSTB,     -- Input port-B reset
  WEA => WEA,      -- Input port-A write enable
```

## Verilog 記述 (インスタンス化)

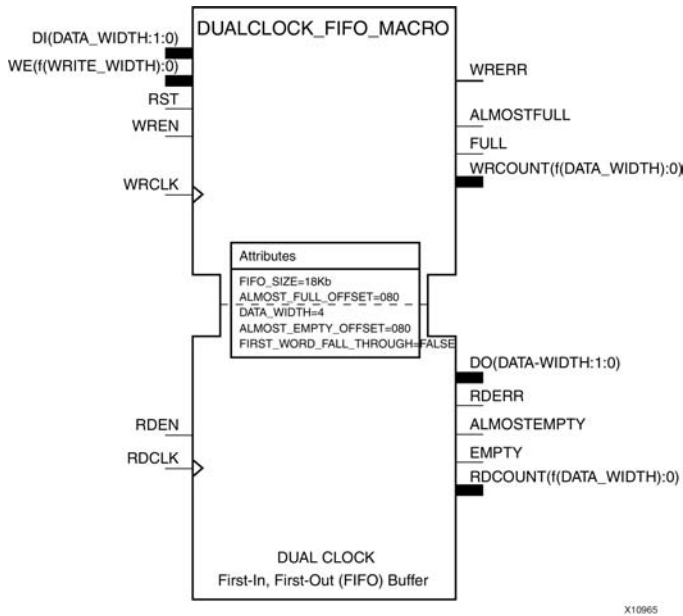
Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v11.4) 2009 年 12 月 2 日

[illegible]

Virtex-6 FPGA の資料 (ユーザー ガイドおよびデータシート)

## 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 (インスタンスエーション)

```
// 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
);

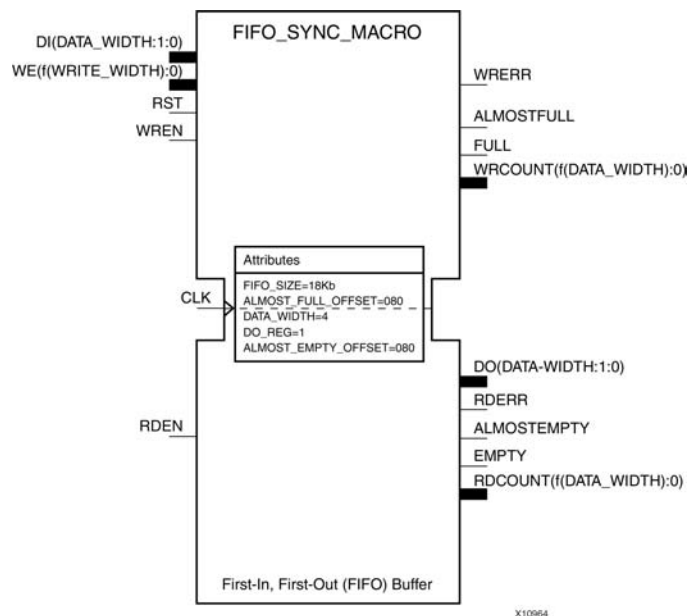
// End of FIFO_DUALCLOCK_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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ト
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ト
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
);

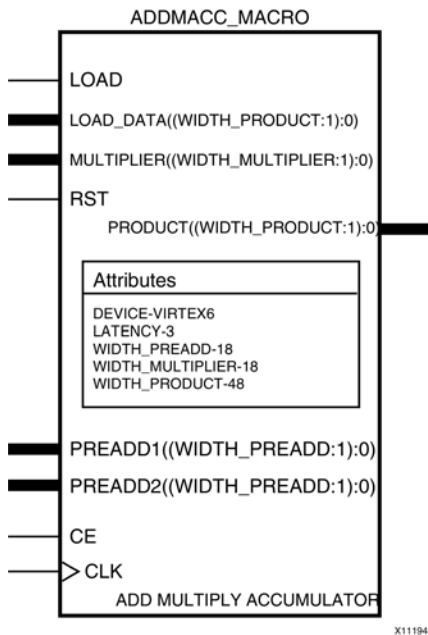
// End of FIFO_SYNC_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ADDMACC\_MACRO

マクロ : Adder/Multiplier/Accumulator



### 概要

前置加算器、乗算/累積ファンクションとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
PRODUCT	出力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	プライマリ データ出力
入力ポート			
PREADD1	入力	可変値。WIDTH_PREADD 属性を参照。	前置加算データ入力
PREADD2	入力	可変値。WIDTH_PREADD 属性を参照。	前置加算データ入力
MULTIPLIER	入力	可変値。WIDTH_MULTIPLIER 属性を参照。	乗算器データ入力
CARRYIN	入力	1	キャリー入力
CLK	入力	1	クロック
CE	入力	1	クロック イネーブル
LOAD	入力	1	ロード
LOAD_DATA	入力	可変値。WIDTH_PRODUCT 属性を参照。	DSP スライスでは、LOAD がアサートされると P は $A*B+LOAD\_DATA$ でロードされます。
RST	入力	1	同期リセット

## 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。

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
WIDTH_PREADD	整数	1 ~ 24	24	PREADD1 および PREADD2 入力の幅を制御します。
WIDTH_MULTIPLIER	整数	1 ~ 18	18	MULTIPLIER 入力の幅を制御します。
WIDTH_PRODUCT	整数	1 ~ 48	48	MULTIPLIER 出力の幅を制御します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数 <ul style="list-style-type: none"> <li>1 - MREG == 1</li> <li>2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>3 - AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>4 - AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	VIRTEX6、SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ

## 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;

-- ADDMACC_MACRO: Add and Multiple Accumulate Function implemented in a DSP48E
--                Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

ADDMACC_MACRO_inst : ADDMACC_MACRO
generic map (
  DEVICE => "VIRTEX6", -- Target Device: "VIRTEX6", "SPARTAN6"
  LATENCY => 3,         -- Desired clock cycle latency, 1-4
  WIDTH_PREADD => 18,    -- Pre-Adder input bus width, 1-25
  WIDTH_MULTIPLIER => 18, -- Multiplier input bus width, 1-18
  WIDTH_PRODUCT => 48)  -- Product output bus width, 1-48
port map (
  PRODUCT => PRODUCT,    -- ADDMACC output bus, width determined by WIDTH_PRODUCT generic
  MULTIPLIER => MULTIPLIER, -- MULTIPLIER input bus, width determined by WIDTH_MULTIPLIER generic
  PREADDER1 => PREADDER1, -- 1st Pre-Adder input bus, width determined by WIDTH_PREADDER generic
  PREADDER2 => PREADDER2, -- 2nd Pre-Adder input bus, width determined by WIDTH_PREADDER generic
  CARRYIN => CARRYIN,    -- 1-bit carry-in input to accumulator
  CE => CE,              -- 1-bit active high input clock enable
  CLK => CLK,            -- 1-bit positive edge clock input
```

```
LOAD => LOAD, -- 1-bit active high input load accumulator enable
LOAD_DATA => LOAD_DATA, -- Load accumulator input data,
                        -- width determined by WIDTH_PRODUCT generic
RST => RST      -- 1-bit input active high reset
);
-- End of ADDMACC_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ADDMACC_MACRO: Variable width & latency - Pre-Add -> Multiplier -> Accumulate
//                function implemented in a DSP48E
//                Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

ADDMACC_MACRO #(
    .DEVICE("VIRTEX5"),    // Target Device: "VIRTEX6", "SPARTAN6"
    .LATENCY(2),           // Desired clock cycle latency, 0-2
    .WIDTH_PREADD(25),     // Pre-adder inputwidth, 1-25
    .WIDTH_MULTIPLIER(18), // Multiplier input width, 1-18
    .WIDTH_PRODUCT(48)     // MACC output width, 1-48
) ADDMACC_MACRO_inst (
    .PRODUCT(PRODUCT),    // MACC result output, width defined by WIDTH_PRODUCT parameter
    .CARRYIN(CARRYIN),    // 1-bit carry-in input
    .CLK(CLK),            // 1-bit clock input
    .CE(CE),              // 1-bit clock enable input
    .LOAD(LOAD),          // 1-bit accumulator load input
    .LOAD_DATA(LOAD_DATA), // Accumulator load data input, width defined by WIDTH_PRODUCT parameter
    .MULTIPLIER(MULTIPLIER), // Multiplier data input, width defined by WIDTH_MULTIPLIER parameter
    .PREADD2(PREADD2),    // Preadder data input, width defined by WIDTH_PREADD parameter
    .PREADD1(PREADD1),    // Preadder data input, width defined by WIDTH_PREADD parameter
    .RST(RST)             // 1-bit active high synchrnous reset
);

// End of ADDMACC_MACRO_inst instantiation
```

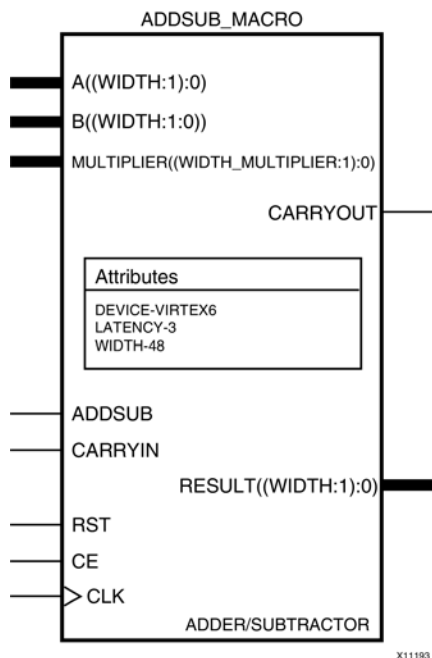
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## ADDSUB\_MACRO

マクロ : Adder/Subtractor



### 概要

単純な加算器/減算器として使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
CARRYOUT	出力	1	キャリー出力
RESULT	出力	可変値。WIDTH 属性を参照。	RDADDR で指定されたデータ出力バス
入力ポート			
ADDSUB	入力	1	High の場合、RESULT は加算器です。Low の場合、RESULT は減算器です。
A	入力	可変値。WIDTH 属性を参照。	加算器/減算器へのデータ入力
B	入力	可変値。WIDTH_B 属性を参照。	加算器/減算器へのデータ入力
CE	入力	1	クロック イネーブル
CARRYIN	入力	1	キャリー入力
CLK	入力	1	クロック
RST	入力	1	同期リセット

### 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。

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
DEVICE	文字列	VIRTEX6、 SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
LATENCY	整数	0、1、2	2	パイプライン レジスタの数 <ul style="list-style-type: none"> <li>1 - PREG == 1</li> <li>2 - AREG == BREG == CREG == PREG</li> </ul>
WIDTH	整数	1 ～ 48	48	A、B、RESULT ポート幅。B および RESULT ポート幅をほかのパラメータ使 用し上書きできます。
WIDTH_B	整数	1 ～ 48	48	ポート B 幅上書き
WIDTH_RESULT	整数	1 ～ 48	48	RESULT ポート幅上書き

## 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;

-- ADDSUB_MACRO: Variable width & latency - Adder / Subtractor implemented in a DSP48E
--                Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

ADDSUB_MACRO_inst : ADDSUB_MACRO
generic map (
    DEVICE => "VIRTEX5", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    LATENCY => 2,         -- Desired clock cycle latency, 0-2
    WIDTH => 48)          -- Input / Output bus width, 1-48
port map (
    CARRYOUT => CARRYOUT, -- 1-bit carry-out output signal
    RESULT => RESULT,     -- Add/sub result output, width defined by WIDTH generic
    A => A,                -- Input A bus, width defined by WIDTH generic
    ADD_SUB => ADD_SUB,    -- 1-bit add/sub input, high selects add, low selects subtract
    B => B,                -- Input B bus, width defined by WIDTH generic
    CARRYIN => CARRYIN,    -- 1-bit carry-in input
    CE => CE,              -- 1-bit clock enable input
    CLK => CLK,            -- 1-bit clock input
    RST => RST             -- 1-bit active high synchronous reset
);
-- End of ADDSUB_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ADDSUB_MACRO: Variable width & latency - Adder / Subtrator implemented in a DSP48E
//                               Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

ADDSUB_MACRO #(
    .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(2),         // Desired clock cycle latency, 0-2
    .WIDTH(48)           // Input / output bus width, 1-48
) ADDSUB_MACRO_inst (
    .CARRYOUT(CARRYOUT), // 1-bit carry-out output signal
    .RESULT(RESULT),     // Add/sub result output, width defined by WIDTH parameter
    .A(A),               // Input A bus, width defined by WIDTH parameter
    .ADD_SUB(ADD_SUB),   // 1-bit add/sub input, high selects add, low selects subtract
    .B(B),               // Input B bus, width defined by WIDTH parameter
    .CARRYIN(CARRYIN),   // 1-bit carry-in input
    .CE(CE),             // 1-bit clock enable input
    .CLK(CLK),           // 1-bit clock input
    .RST(RST)            // 1-bit active high synchronous reset
);

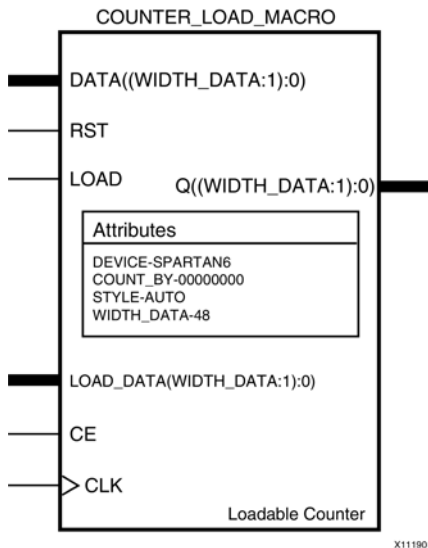
// End of ADDSUB_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## COUNTER\_LOAD\_MACRO

マクロ : Loadable Counter



### 概要

ダイナミック ロード アップ/ダウン カウンタとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な出力幅およびカウント値があり、DSP48 ブロックを HDL に統合しやすくします。

### ポートの説明

属性	方向	幅	機能
出力ポート			
Q	出力	可変値。WIDTH_DATA 属性を参照。	カウンタ出力
入力ポート			
DATA	入力	可変値。WIDTH_DATA 属性を参照。	データ入力 (可変データに対し 2 クロックのレイテンシ)
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
LOAD	入力	可変値。WIDTH_DATA 属性を参照。	アサートされると、LOAD_DATA からカウンタをロードします (2 クロックのレイテンシ)。
LOAD_DATA	入力	可変値。WIDTH_DATA 属性を参照。	DSP スライスでは、LOAD ピンをアサートすると、このデータが P レジスタに入力されます (2 クロックのレイテンシ)。
DIRECTION	入力	1	High の場合はアップ、Low の場合はダウンです (2 クロックのレイテンシ)。
RST	入力	1	同期リセット

### 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。

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
DEVICE	文字列	VIRTEX6、 SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
COUNT_BY	16 進数	48 ビット値	000000000001	N ごとにカウント。WIDTH_DATA より優先されます。
WIDTH_DATA	整数	1 ~ 48	48	カウンタ幅を指定します。

## 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 UNIMACRO;
use unimacro.Vcomponents.all;

-- COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
--                               Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

COUNTER_LOAD_MACRO_inst : COUNTER_LOAD_MACRO
generic map (
    COUNT_BY => X"000000000001", -- Count by value
    DEVICE => "VIRTEX5",          -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    WIDTH_DATA => 48)             -- Counter output bus width, 1-48
port map (
    Q => Q,                        -- Counter output, width determined by WIDTH_DATA generic
    CLK => CLK,                    -- 1-bit clock input
    CE => CE,                      -- 1-bit clock enable input
    DIRECTION => DIRECTION,        -- 1-bit up/down count direction input, high is count up
    LOAD => LOAD,                  -- 1-bit active high load input
    LOAD_DATA => LOAD_DATA,        -- Counter load data, width determined by WIDTH_DATA generic
    RST => RST                     -- 1-bit active high synchronous reset
);
-- End of COUNTER_LOAD_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// COUNTER_LOAD_MACRO: Loadable variable counter implemented in a DSP48E
//                               Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

COUNTER_LOAD_MACRO #(
    .COUNT_BY(48'h00000000000001), // Count by value
    .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .WIDTH_DATA(48) // Counter output bus width, 1-48
) COUNTER_LOAD_MACRO_inst (
    .Q(Q), // Counter output, width determined by WIDTH_DATA parameter
    .CLK(CLK), // 1-bit clock input
    .CE(CE), // 1-bit clock enable input
    .DIRECTION(DIRECTION), // 1-bit up/down count direction input, high is count up
    .LOAD(LOAD), // 1-bit active high load input
    .LOAD_DATA(LOAD_DATA), // Counter load data, width determined by WIDTH_DATA parameter
    .RST(RST) // 1-bit active high synchronous reset
);

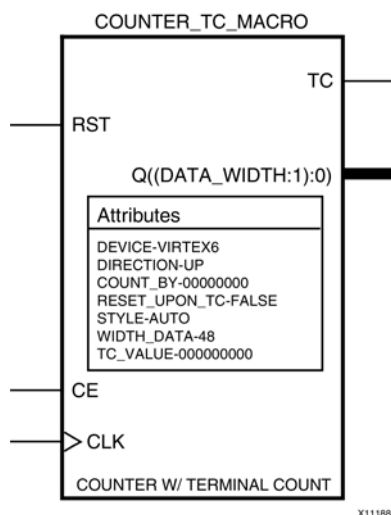
// End of COUNTER_LOAD_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## COUNTER\_TC\_MACRO

マクロ : Counter with Terminal Count



### 概要

ターミナル アップ/ダウン カウンタとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な出力幅、ターミナル カウンタ値、カウント設定、カウント方向があり、DSP48 ブロックを HDL に統合しやすくします。

### ポートの説明

属性	方向	幅	機能
出力ポート			
TC	出力	1	TC_VALUE が得られると、ターミナル カウントが High になります。
Q	出力	可変値。WIDTH_DATA 属性を参照。	カウンタ出力
入力ポート			
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
RST	入力	1	同期リセット

### 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。

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
RESET_UPON_TC	ブール代数	TRUE、FALSE	FALSE	ターミナル カウント達成時にカウンタをリセットするかどうかを指定します。
DEVICE	文字列	VIRTEX6、SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
DIRECTION	文字列	UP、DOWN	UP	カウンタ方向を指定します。
COUNT_BY	16 進数	48 ビット値	000000000001	N ごとにカウント。WIDTH_DATA より優先されます。
TC_VALUE	16 進数	48 ビット値	すべてゼロ	ターミナル カウント値
WIDTH_DATA	整数	1 ～ 48	48	カウンタ幅を指定します。

## VHDL 記述 (インスタンスエーション)

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

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

-- COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
--                      Virtex-5, Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

COUNTER_TC_MACRO_inst : COUNTER_TC_MACRO
generic map (
  COUNT_BY => X"000000000001", -- Count by value
  DEVICE   => "VIRTEX5",       -- Target Device: "VIRTEX5", "VIRTEX6"
  DIRECTION => "UP",           -- Counter direction "UP" or "DOWN"
  RESET_UPON_TC => FALSE,      -- Reset counter upon terminal count, TRUE or FALSE
  TC_VALUE  => X"000000000000", -- Terminal count value
  WIDTH_DATA => 48)            -- Counter output bus width, 1-48
port map (
  Q => Q,                      -- Counter output, width determined by WIDTH_DATA generic
  TC => TC,                    -- 1-bit terminal count output, high = terminal count is reached
  CLK => CLK,                  -- 1-bit clock input
  CE => CE,                    -- 1-bit clock enable input
  RST(RST)                    -- 1-bit active high synchronous reset
);
-- End of COUNTER_TC_MACRO_inst instantiation
```



## Verilog 記述 (インスタンスレーション)

```
// COUNTER_TC_MACRO: Counter with terminal count implemented in a DSP48E
//                               Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

COUNTER_TC_MACRO #(
    .COUNT_BY(48'h00000000000001), // Count by value
    .DEVICE("VIRTEX5"),             // Target Device: "VIRTEX5", "VIRTEX6"
    .DIRECTION("UP"),               // Counter direction, "UP" or "DOWN"
    .RESET_UPON_TC("FALSE"), // Reset counter upon terminal count, "TRUE" or "FALSE"
    .TC_VALUE(48'h00000000000000), // Terminal count value
    .WIDTH_DATA(48)                 // Counter output bus width, 1-48
) COUNTER_TC_MACRO_inst (
    .Q(Q), // Counter output bus, width determined by WIDTH_DATA parameter
    .TC(TC), // 1-bit terminal count output, high = terminal count is reached
    .CLK(CLK), // 1-bit positive edge clock input
    .CE(CE), // 1-bit active high clock enable input
    .RST(RST) // 1-bit active high synchronous reset
);

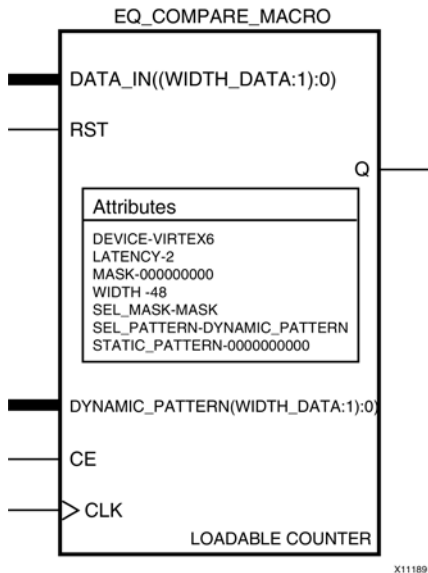
// End of COUNTER_TC_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## EQ\_COMPARE\_MACRO

マクロ : Equality Comparator



### 概要

等価コンパレータとして使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシ、マスク、入力ソースがあり、DSP48 ブロックを HDL に統合しやすくします。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
Q	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 DYNAMIC_PATTERN の値と DATA_IN 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。
入力ポート			
DATA_IN	入力	可変幅。WIDTH 属性値に等しい。	比較する入力データ
DYNAMIC_PATTERN	入力	可変幅。WIDTH 属性値に等しい。	DATA_IN と比較するダイナミック データ
CLK	入力	1	クロック
CE	入力	1	クロック イネーブル
RST	入力	1	同期リセット

### 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。

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
DEVICE	文字列	VIRTEX6、 SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ
SEL_PATTERN	整数	1 ~ 24	24	PREADD1 および PREADD2 入力の幅を制御します。
MASK	16 進数	48 ビットの 16 進数	すべて 1	パターン検出器で使用するマスクを指定します。
STATIC_PATTERN	16 進数	48 ビットの 16 進数	すべて 1	パターン検出器で使用するパターンを指定します。
SEL_MASK	文字列	MASK、 DYNAMIC_PATTERN	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。
WIDTH	整数	1 ~ 48	48	DATA_IN および DYNAMIC_PATTERN の幅
LATENCY	整数	0、1、2、3	2	パイプライン レジスタの数 <ul style="list-style-type: none"> <li>1: QREG == 1</li> <li>2: AREG == BREG == CREG == QREG == 1</li> <li>3: AREG == BREG == 2 and CREG == QREG == 1</li> </ul>

## 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;

-- EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
--                      Virtex-5, Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

EQ_COMPARE_MACRO_inst : EQ_COMPARE_MACRO
generic map (
  DEVICE => "VIRTEX5",           -- Target Device: "VIRTEX5", "VIRTEX6"
  LATENCY => 2,                   -- Desired clock cycle latency, 0-2
  MASK => X"000000000000",       -- Select bits to be masked, must set
  SEL_MASK => "MASK",            -- SEL_MASK = "MASK"
  SEL_PATTERN => "DYNAMIC_PATTERN", -- "MASK" = use MASK generic,
  STATIC_PATTERN => X"000000000000", -- "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
  WIDTH => 48)                   -- "STATIC_PATTERN" = use STATIC_PATTERN generic
-- Specify static pattern,
-- must set SEL_PATTERN = "STATIC_PATTERN"
-- Comparator output bus width, 1-48
```

```

port map (
    Q => Q,          -- 1-bit output indicating a match
    CE => CE,        -- 1-bit active high input clock enable input
    CLK => CLK,      -- 1-bit positive edge clock input
    DATA_IN => DATA_IN -- Input Data Bus, width determined by WIDTH generic
    DYNAMIC_PATTERN => DYNAMIC_PATTERN, -- Input Dynamic Match/Mask Bus, width determined by WIDTH generic
    RST(RST)        -- 1-bit input active high reset
);
-- End of EQ_COMPARE_MACRO_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// EQ_COMPARE_MACRO: Equality Comparator implemented in a DSP48E
//                               Virtex-5, Virtex-6
// Xilinx HDL Libraries Guide, version 11.2

EQ_COMPARE_MACRO #(
    .DEVICE("VIRTEX5"),          // Target Device: "VIRTEX5", "VIRTEX6"
    .LATENCY(2),                 // Desired clock cycle latency, 0-2
    .MASK(48'h000000000000),    // Select bits to be masked, must set SEL_MASK="MASK"
    .SEL_MASK("MASK"),          // "MASK" = use MASK parameter,
                                // "DYNAMIC_PATTERN" = use DYNAMIC_PATTE input bus
    .SEL_PATTERN("STATIC_PATTERN"), // "STATIC_PATTERN" = use STATIC_PATTERN parameter,
                                // "DYNAMIC_PATTERN" = use DYNAMIC_PATTERN input bus
    .STATIC_PATTERN(48'h000000000000), // Specify static pattern, must set SEL_PATTERN = "STATIC_PATTERN"
    .WIDTH(48)                   // Comparator output bus width, 1-48
) EQ_COMPARE_MACRO_inst (
    .Q(Q),          // 1-bit output indicating a match
    .CE(CE),        // 1-bit active high input clock enable
    .CLK(CLK),      // 1-bit positive edge clock input
    .DATA_IN(DATA_IN), // Input Data Bus, width determined by WIDTH parameter
    .DYNAMIC_PATTERN(DYNAMIC_PATTERN), // Input Dynamic Match/Mask Bus, width determined by WIDTH parameter
    .RST(RST)       // 1-bit input active high reset
);

// End of EQ_COMPARE_MACRO_inst instantiation

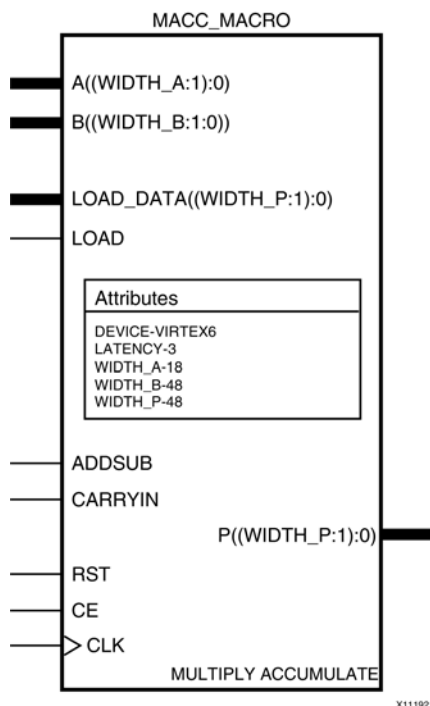
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MACC\_MACRO

マクロ : Multiplier/Accumulator



## 概要

単純な符号付乗算器/アキュムレータ モードで使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

## ポートの説明

ポート名	方向	幅	機能
出力ポート			
P	出力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	プライマリ データ出力
入力ポート			
A	入力	可変値。WIDTH_A 属性を参照。	乗算器データ入力
B	入力	可変値。WIDTH_B 属性を参照。	乗算器データ入力
CARRYIN	入力	1	キャリー入力
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
LOAD	入力	1	ロード
LOAD_DATA	入力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	DSP スライスでは、LOAD がアサートされると P は $A*B + \text{LOAD\_DATA}$ でロードされます。

ポート名	方向	幅	機能
RST	入力	1	同期リセット
ADDSUB	入力	1	High の場合はアキュムレータを乗算モードに、Low の場合は減算モードに設定します。

## 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。

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
WIDTH_A	整数	1 ~ 25	25	A 入力の幅を制御します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を制御します。
LATENCY	整数	0、1、2、3、4	3	パイプライン レジスタの数 <ul style="list-style-type: none"> <li>1 - MREG == 1</li> <li>2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>3 - AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>4 - AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	VIRTEX6、SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ

## 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;

-- MACC_MACRO: Multiple Accumulate Function implemented in a DSP48E
--             Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

MACC_MACRO_inst : MACC_MACRO
generic map (
  DEVICE => "VIRTEX5",    -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  LATENCY => 3,           -- Desired clock cycle latency, 1-4
  WIDTH_A => 25,          -- Multiplier A-input bus width, 1-25
  WIDTH_B => 18,          -- Multiplier B-input bus width, 1-18
  WIDTH_P => 48)         -- Accumulator output bus width, 1-48
port map (
```

```

P => P,      -- MACC output bus, width determined by WIDTH_P generic
A => A,      -- MACC input A bus, width determined by WIDTH_A generic
ADDSUB => ADDSUB, -- 1-bit add/sub input, high selects add, low selects subtract
B => B,      -- MACC input B bus, width determined by WIDTH_B generic
CARRYIN => CARRYIN, -- 1-bit carry-in input to accumulator
CE => CE,    -- 1-bit active high input clock enable
CLK => CLK,  -- 1-bit positive edge clock input
LOAD => LOAD, -- 1-bit active high input load accumulator enable
LOAD_DATA => LOAD_DATA, -- Load accumulator input data,
                        -- width determined by WIDTH_P generic
RST => RST    -- 1-bit input active high reset
);

-- End of MACC_MACRO_inst instantiation

```

## Verilog 記述 (インスタンス化)

```

// MACC_MACRO: Multiply Accumulate Function implemented in a DSP48E
//               Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

MACC_MACRO #(
    .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(3),        // Desired clock cycle latency, 1-4
    .WIDTH_A(25),       // Multiplier A-input bus width, 1-25
    .WIDTH_B(18),       // Multiplier B-input bus width, 1-18
    .WIDTH_P(48)        // Accumulator output bus width, 1-48
) MACC_MACRO (
    .P(P),              // MACC output bus, width determined by WIDTH_P parameter
    .A(A),              // MACC input A bus, width determined by WIDTH_A parameter
    .ADDSUB(ADDSUB),    // 1-bit add/sub input, high selects add, low selects subtract
    .B(B),              // MACC input B bus, width determined by WIDTH_B parameter
    .CARRYIN(CARRYIN),  // 1-bit carry-in input to accumulator
    .CE(CE),            // 1-bit active high input clock enable
    .CLK(CLK),          // 1-bit positive edge clock input
    .LOAD(LOAD),        // 1-bit active high input load accumulator enable
    .LOAD_DATA(LOAD_DATA), // Load accumulator input data, width determined by WIDTH_P parameter
    .RST(RST)           // 1-bit input active high reset
);

// End of MACC_MACRO_inst instantiation

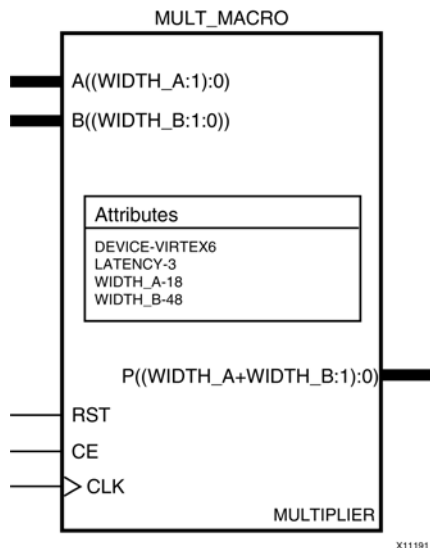
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## DSP\_MULT\_MACRO

マクロ : Multiplier



### 概要

単純な乗算器として使用すると、DSP48 ブロックのインスタンス化が簡単になります。パラメータ設定可能な入力幅および出力幅、レイテンシがあり、DSP48 ブロックを HDL に統合しやすくします。

### ポートの説明

ポート名	方向	幅	機能
出力ポート			
P	出力	可変幅。WIDTH_A 属性値 + WIDTH_B 属性値に等しい。	プライマリ データ出力
入力ポート			
A	入力	可変値。WIDTH_A 属性を参照。	乗算器データ入力
B	入力	可変値。WIDTH_B 属性を参照。	乗算器データ入力
CE	入力	1	クロック イネーブル
CLK	入力	1	クロック
RST	入力	1	同期リセット

### 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。

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
WIDTH_A	整数	1 ~ 25	25	A 入力の幅を制御します。
WIDTH_B	整数	1 ~ 18	18	B 入力の幅を制御します。
LATENCY	整数	0, 1, 2, 3, 4	3	パイプライン レジスタの数 <ul style="list-style-type: none"> <li>・ 1 - MREG == 1</li> <li>・ 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1</li> <li>・ 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1</li> <li>・ 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1</li> </ul>
DEVICE	文字列	VIRTEX6、SPARTAN6	VIRTEX6	ターゲットのハードウェア アーキテクチャ

## 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;

-- MULT_MACRO: Multiply Function implemented in a DSP48E
--               Virtex-5, Virtex-6, Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

MULT_MACRO_inst : MULT_MACRO
generic map (
  DEVICE => "VIRTEX5",    -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  LATENCY => 3,           -- Desired clock cycle latency, 0-4
  WIDTH_A => 18,          -- Multiplier A-input bus width, 1-25
  WIDTH_B => 18)          -- Multiplier B-input bus width, 1-18
port map (
  P => P,                -- Multiplier output bus, width determined by WIDTH_P generic
  A => A,                -- Multiplier input A bus, width determined by WIDTH_A generic
  B => B,                -- Multiplier input B bus, width determined by WIDTH_B generic
  CE => CE,             -- 1-bit active high input clock enable
  CLK => CLK,           -- 1-bit positive edge clock input
  RST => RST             -- 1-bit input active high reset
);
-- End of MULT_MACRO_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// MULT_MACRO: Multiply Function implemented in a DSP48E
//                Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

MULT_MACRO #(
    .DEVICE("VIRTEX5"), // Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    .LATENCY(3),         // Desired clock cycle latency, 0-4
    .WIDTH_A(18),        // Multiplier A-input bus width, 1-25
    .WIDTH_B(18),        // Multiplier B-input bus width, 1-18
) MULT_MACRO (
    .P(P),               // Multiplier output bus, width determined by WIDTH_P parameter
    .A(A),               // Multiplier input A bus, width determined by WIDTH_A parameter
    .B(B),               // Multiplier input B bus, width determined by WIDTH_B parameter
    .CE(CE),             // 1-bit active high input clock enable
    .CLK(CLK),           // 1-bit positive edge clock input
    .RST(RST)            // 1-bit input active high reset
);

// End of MULT_MACRO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

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

### アドバンス

デザイン エLEMENT	説明
PCIE_2_0	プリミティブ : PCI Express version 2.0 Compliant.
SYSMON	プリミティブ : System Monitor
TEMAC_SINGLE	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
DSP48E1	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
BUFH	コンビニエンス プリミティブ : Clock buffer for a single clocking region
BUFHCE	プリミティブ : Clock buffer for a single clocking region with clock enable
BUFIO	プリミティブ : Local Clock Buffer for I/O
BUFIODQS	プリミティブ : Differential Clock Input for Transceiver Reference Clocks
BUFR	プリミティブ : Regional Clock Buffer for I/O and Logic Resources
IBUFDS	プリミティブ : Differential Signaling Input Buffer
IBUFDS_GTHE1	プリミティブ : Differential Clock Input for the GTH Transceiver Reference Clocks
IBUFDS_GTXE1	プリミティブ : Differential Clock Input for the Transceiver Reference Clocks
MMCM_ADV	プリミティブ : MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.

## 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_VIRTEX6	プリミティブ : Virtex®-6 JTAG Boundary-Scan Logic Access Circuit
CAPTURE_VIRTEX6	プリミティブ : Virtex®-6 Readback Register Capture Control
DNA_PORT	プリミティブ : Device DNA Data Access Port
EFUSE_USR	プリミティブ : 32-bit non-volatile design ID
FRAME_ECC_VIRTEX6	プリミティブ : Virtex®-6 Configuration Frame Error Detection and Correction Circuitry
ICAP_VIRTEX6	プリミティブ : Internal Configuration Access Port
STARTUP_VIRTEX6	プリミティブ : Virtex®-6 Configuration Start-Up Sequence Interface
USR_ACCESS_VIRTEX6	プリミティブ : Virtex-6 User Access Register

## コンビニエンス プリミティブ

デザイン エLEMENT	説明
BUFG	コンビニエンス プリミティブ : Global Clock Buffer
BUFGCE	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable
BUFGCE_1	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable and Output State 1
BUFGMUX	コンビニエンス プリミティブ : Global Clock MUX Buffer
BUFGMUX_1	コンビニエンス プリミティブ : Global Clock MUX Buffer with Output State 1
BUFGMUX_CTRL	コンビニエンス プリミティブ : 2-to-1 Global Clock MUX Buffer
BUFGP	コンビニエンス プリミティブ : Primary Global Buffer for Driving Clocks or Longlines
BUFH	コンビニエンス プリミティブ : Clock buffer for a single clocking region
MMCM_BASE	コンビニエンス プリミティブ : Mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.

## ギガビット I/O

デザイン エLEMENT	説明
GTHE1_QUAD	プリミティブ : Gigabit Transceiver
GTXE1	プリミティブ : 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
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
IODELAYE1	プリミティブ : Input and Output Fixed or Variable Delay Element
ISERDESE1	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
OBUFGDS	プリミティブ : Differential Signaling Output Buffer
OBUFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
OSERDESE1	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

## ロジック

デザイン エLEMENT	説明
AND2B1L	プリミティブ : Two input AND gate implemented in place of a Slice Latch
OR2L	プリミティブ : Two input OR gate implemented in place of a Slice Latch

## RAM/ROM

デザイン エLEMENT	説明
FIFO18E1	プリミティブ : 18 k-bit FIFO (First In, First Out) Block RAM Memory
FIFO36E1	プリミティブ : 36 kb FIFO (First In, First Out) Block RAM Memory
RAM128X1D	プリミティブ : 128-Deep by 1-Wide Dual Port Random Access Memory (Select 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)
RAM32X1S	プリミティブ : 32-Deep by 1-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
RAMB18E1	プリミティブ : 18K-bit Configurable Synchronous Block RAM
RAMB36E1	プリミティブ : 36K-bit Configurable Synchronous Block RAM

## レジスタおよびラッチ

デザイン エLEMENT	説明
FDCE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Clear
FDPE	プリミティブ : D Flip-Flop with Clock Enable and Asynchronous Preset
FDRE	プリミティブ : D Flip-Flop with Clock Enable and Synchronous Reset
FDSE	プリミティブ : D Flip-Flop with Clock Enable and Synchronous Set
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
LDCE	プリミティブ : Transparent Data Latch with Asynchronous Clear and Gate Enable
LDPE	プリミティブ : Transparent Data Latch with Asynchronous 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
SRL16E	プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with 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
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

---

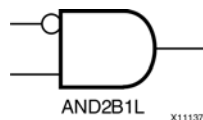
このセクションでは、このアーキテクチャで使えるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

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

## AND2B1L

プリミティブ : Two input AND gate implemented in place of a Slice Latch



### 概要

このデザイン エLEMENTでは、コンフィギュレーション可能なスライス ラッチで 1 入力反転される 2 入力 AND ゲートのファンクションが使用されます。このELEMENTを使用すると、ロジックのレジスタ/ラッチ リソース数をトレードオフにすることで、ロジック レベルを削減して、デバイスのロジック集積度を高めることができます。このELEMENTはレジスタのパックおよび集積度に影響を与えるので注意してください。AND2B1L または OR2L ELEMENTをスライスに指定すると、残りのレジスタおよびラッチが使用できなくなります。

### 論理表

入力		出力
DI	SRI	O
0	0	0
0	1	0
1	0	1
1	1	0

### ポートの説明

ポート名	タイプ	幅	機能
O	出力	1	AND ゲートの出力
DI	入力	1	同じスライスにあるソース LUT に通常接続されるアクティブ High の入力
SRI	入力	1	通常スライス外からソースされるアクティブ Low の入力  <b>メモ :</b> 複数の AND2B1L または OR2B1L を 1 つのスライスにパックするには、この入力に共通の信号を接続する必要があります。

### デザインの入力方法

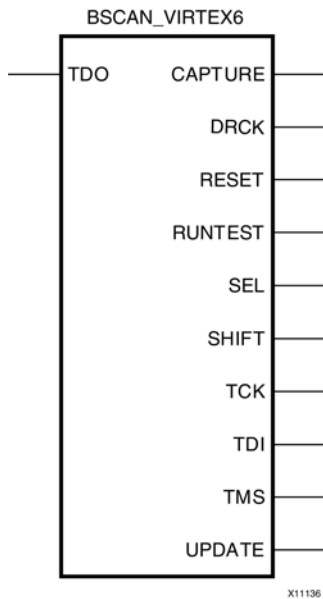
インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BSCAN\_VIRTEX6

プリミティブ : Virtex®-6 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 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。
RUNTEST	出力	1	JTAG が Run Test/Idle ステートであることを示します。JTAG TAP コントローラが RTI ステート マシンになると 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。
TCK	出力	1	FGPA への TCK 入力値と同じ値を出力します。
TDI	出力	1	TDI ピンと同じ値を出力します。
TMS	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
DISABLE_JTAG	ブール 代数	TRUE、FALSE	FALSE	BSCAN_VIRTEX6 インスタンス 1 に付けられます。
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;

-- BSCAN_VIRTEX6: Virtex-6 JTAG Boundary-Scan Logic Access
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

BSCAN_VIRTEX6_inst : BSCAN_VIRTEX6
generic map (
    DISABLE_JTAG => FALSE,
    JTAG_CHAIN => 1           -- Chain number.
)
port map (
    CAPTURE => CAPTURE, -- 1-bit Scan Data Register Capture instruction.
    DRCK => DRCK,       -- 1-bit Scan Clock instruction. DRCK is a gated version of TCTCK, it toggles during
                        -- the CAPTUREDR and SHIFTDTR states.

    RESET => RESET,     -- 1-bit Scan register reset instruction.
    RUNTEST => RUNTEST, -- 1-bit Asserted when TAP controller is in Run Test Idle state. Make sure is the
                        -- same name as BSCAN primitive used in Spartan products.

    SEL => SEL,          -- 1-bit Scan mode Select instruction.
    SHIFT => SHIFT,     -- 1-bit Scan Chain Shift instruction.
    TCK => TCK,          -- 1-bit Scan Clock. Fabric connection to TAP Clock pin.
    TDI => TDI,          -- 1-bit Scan Chain Output. Mirror of TDI input pin to FPGA.
    TMS => TMS,          -- 1-bit Test Mode Select. Fabric connection to TAP.
    UPDATE => UPDATE,    -- 1-bit Scan Register Update instruction.
    TDO => TDO           -- 1-bit Scan Chain Input.
);

-- End of BSCAN_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// BSCAN_VIRTEX6: Virtex-6 JTAG Boundary-Scan Logic Access
//           Virtex-6
// Xilinx HDL Language Template, version 11.1

BSCAN_VIRTEX6 #(
    .DISABLE_JTAG("FALSE"),
    .JTAG_CHAIN(1)           // Chain number.
)
BSCAN_VIRTEX6_inst (
    .CAPTURE(CAPTURE), // 1-bit Scan Data Register Capture instruction.
    .DRCK(DRCK),       // 1-bit Scan Clock instruction. DRCK is a gated version of TCTCK, it toggles during
                        // the CAPTUREDR and SHIFTDTR states.

    .RESET(RESET),     // 1-bit Scan register reset instruction.
    .RUNTEST(RUNTEST), // 1-bit Asserted when TAP controller is in Run Test Idle state. Make sure is the same
                        // name as BSCAN primitive used in Spartan products.

    .SEL(SEL),          // 1-bit Scan mode Select instruction.
    .SHIFT(SHIFT),     // 1-bit Scan Chain Shift instruction.
    .TCK(TCK),          // 1-bit Scan Clock. Fabric connection to TAP Clock pin.
    .TDI(TDI),          // 1-bit Scan Chain Output. Mirror of TDI input pin to FPGA.
    .TMS(TMS),          // 1-bit Test Mode Select. Fabric connection to TAP.
    .UPDATE(UPDATE),   // 1-bit Scan Register Update instruction.
    .TDO(TDO)           // 1-bit Scan Chain Input.
);

// End of BSCAN_VIRTEX6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

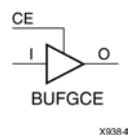
// End of BUFG_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGCE

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



### 概要

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

### 論理表

入力		出力
I	CE	O
X	0	0
I	1	I

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFGCE_inst instantiation
```



## Verilog 記述 (インスタンス化)

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

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

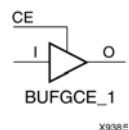
// End of BUFGCE_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGCE\_1

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



### 概要

このデザイン エLEMENTは、クロック イネーブル付きグローバル クロック バッファです。O 出力は、クロック イネーブル (CE) が Low (非アクティブ) のときに High (1) になります。CE が High になると、I 入力の値が O に出力されます。

### 論理表

入力		出力
I	CE	O
X	0	1
I	1	I

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFGCE_1_inst instantiation
```

## Verilog 記述 (インスタンス化)

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

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

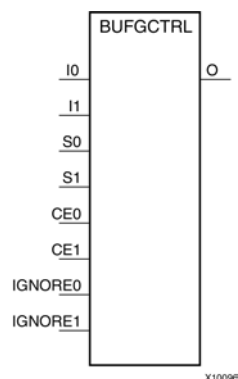
// End of BUFGCE_1_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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。

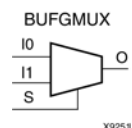
メモ : 2 つの PRESELECT 属性を同時に TRUE にすることはできません。

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGMUX

プリミティブ : Global Clock MUX Buffer



### 概要

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

BUFGMUX および BUFGMUX\_1 では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX\_1 は出力ステートが 1 に保持されます。

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスエーション)

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;

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

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

-- End of BUFGMUX_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

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

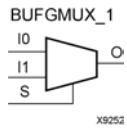
BUFGMUX BUFGMUX_inst (
    .O(O),    // Clock MUX output
    .I0(I0),  // Clock0 input
    .I1(I1),  // Clock1 input
    .S(S)     // Clock select input
);

// End of BUFGMUX_inst instantiation
```

## 詳細情報

## BUFGMUX\_1

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



### 概要

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

このデザイン エLEMENTと BUFGMUX では、S の値が変化した後クロックが切り替わるまでに保持される出力ステートが異なります。BUFGMUX は出力ステートが 0 に、BUFGMUX\_1 は出力ステートが 1 に保持されます。

### 論理表

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

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFGMUX_1_inst instantiation
```



## Verilog 記述 (インスタンス化)

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

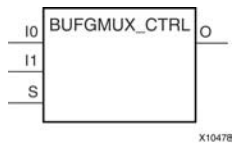
BUFGMUX_1 BUFGMUX_1_inst (
    .O(O),      // Clock MUX output
    .I0(I0),    // Clock0 input
    .I1(I1),    // Clock1 input
    .S(S)       // Clock select input
);

// End of BUFGMUX_1_inst instantiation
```

## 詳細情報

## 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 (インスタンス化)

```
// 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
);

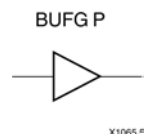
// End of BUFGMUX_CTRL_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFGP

コンビニエンス プリミティブ : Primary Global Buffer for Driving Clocks or Longlines



### 概要

このデザイン エLEMENTはプライマリ グローバル バッファであり、FPGA デバイス内でファンアウトの高いクロックまたは制御信号を分配するために使用されます。これは BUFG を駆動する IBUFG と同等です。

このELEMENTを使用すると、コンフィギャブル ロジック ブロック (CLB) および I/O ブロック (IOB) のクロック ピンに直接アクセスでき、ほかの CLB 入力には制限付きでアクセスできます。BUFGP への入力は、専用 IOB からのみ供給されます。構造上、このELEMENTは常にクロック ピンに直接アクセスできるようになっています。ただし、BUFGP がデバイスのどのエッジに配置されているかによって、F3、G1、C3、C1 ピンのいずれか 1 つにしかアクセスできません。必要なピンに垂直ラインから直接アクセスできない場合、PAR で別の CLB を通して信号が送られ、一般配線を使用してそのロード ピンにアクセスします。

### デザインの入力方法

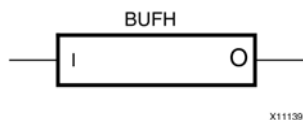
このELEMENTは、回路図でのみ使用できます。

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFH

コンビニエンス プリミティブ : Clock buffer for a single clocking region



### 概要

インスタンス化で HCLK クロック バッファ リソースを使用するための機能を提供します。

### ポートの説明

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

### 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

### VHDL 記述 (インスタンス化)

```
-- BUFH: (Clock buffer for a single clocking region)
--      Spartan-6
-- Xilinx HDL Libraries Guide, version 11.4

BUFH_inst : BUFH
generic map (
)
port map (
  O => O, -- 1-bit Clock Output
  I => I  -- 1-bit Clock Input
);

-- End of BUFH_inst instantiation
```

### Verilog 記述 (インスタンス化)

```
// BUFH: Clock buffer for a single clocking region
//      Virtex-6
// Xilinx HDL Language Template, version 11.4

BUFH BUFH_inst (
  .O(O), // 1-bit The output to the BUFH
  .I(I)  // 1-bit The input to the BUFH
);

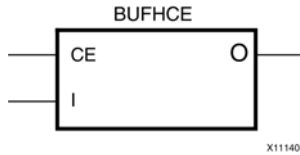
// End of BUFH_inst instantiation
```

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFHCE

**プリミティブ** : Clock buffer for a single clocking region with clock enable



### 概要

このデザイン エLEMENTでは、インスタンス化で HCLK クロック バッファリソースを使用するための機能を提供します。また、CE を介してクロック イネーブル (CE) をディスエーブルにする電力消費量低減機能もあります。

### ポートの説明

ポート名	タイプ	幅	機能
CE	入力	1	I から O まで信号を伝播します。Low の場合出力を 0 に設定します。
I	入力	1	BUFH への入力
O	出力	1	BUFH の出力

### 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

### 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT_OUT	10 進数	0、1	0	初期出力値で、停止が Low か High かを示します。

### VHDL 記述 (インスタンス化)

```
-- BUFHCE: Clock buffer for a single clocking region with clock enable
--      Virtex-6
-- Xilinx HDL Libraries Guide, version 11.4

BUFHCE_inst : BUFHCE
generic map (
  INIT_OUT => 0 -- Initial output value, also indicates stop low vs stop high behavior
)
port map (
  O => O,    -- 1-bit The output of the BUFH
  CE => CE,  -- 1-bit Enables propagation of signal from I to O. When low, sets output to 0.
  I => I     -- 1-bit The input to the BUFH
);

-- End of BUFHCE_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFHCE: Clock buffer for a single clocking region with clock enable
//          Virtex-6
// Xilinx HDL Language Template, version 11.4

BUFHCE #(
    .INIT_OUT(0) // Initial output value, also indicates stop low vs stop high behavior
)
BUFHCE_inst (
    .O(0), // 1-bit The output of the BUFH
    .CE(CE), // 1-bit Enables propagation of signal from I to O. When low, sets output to 0.
    .I(I) // 1-bit The input to the BUFH
);

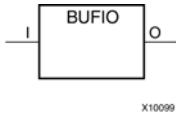
// End of BUFHCE_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

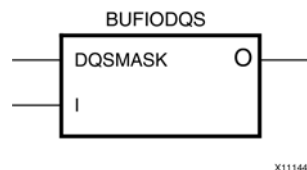
// End of BUFIO_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## BUFIODQS

プリミティブ : Differential Clock Input for Transceiver Reference Clocks



### 概要

メモリ アプリケーションの使用に理想的な追加専用回路付き BUFIO と同じクロック バッファで、オプションで余分な BUFIO 遅延を削除し、ストロブからのバースト長の後に I/O クロックをスケルチします。

### ポートの説明

ポート名	タイプ	幅	機能
DQSMASK	入力	1	ストロブからのバースト長の後に I/O クロックをスケルチします。
I	入力	1	クロック入力ポート
O	出力	1	クロック出力ポート

### 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
DQSMASK_ENABLE	ブール代数	FALSE、TRUE	FALSE	スケルチ回路を有効にします。

### VHDL 記述 (インスタンシエーション)

```
-- BUFIODQS: Differential Clock Input for Transceiver Reference Clocks
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFIODQS_inst : BUFIODQS
generic map (
    DQSMASK_ENABLE => FALSE -- Enables the squelch circuitry
)
port map (
    O => O,                -- 1-bit Clock output port
    DQSMASK => DQSMASK,    -- 1-bit "squelch" the I/O clock after a given burst length from strobe
    I => I                  -- 1-bit Clock input port
);

-- End of BUFIODQS_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// BUFIODQS: Differential Clock Input for Transceiver Reference Clocks
//          Virtex-6
// Xilinx HDL Language Template, version 11.1

BUFIODQS #(
    .DQSMASK_ENABLE("FALSE") // Enables the squelch circuitry
)
BUFIODQS_inst (
    .O(0), // 1-bit Clock output port
    .DQSMASK(DQSMASK), // 1-bit "squelch" the I/O clock after a given burst length from strobe
    .I(I) // 1-bit Clock input port
);

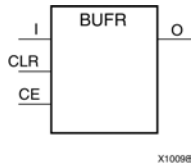
// End of BUFIODQS_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

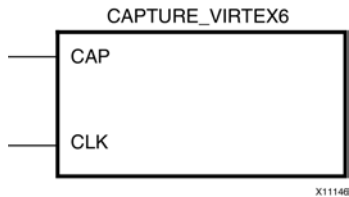
// End of BUFR_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## CAPTURE\_VIRTEX6

プリミティブ : Virtex®-6 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

正しく動作するように、入力および出力をすべてデザインに接続します。

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

-- CAPTURE_VIRTEX6: Virtex-6 Readback Register Capture Control
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

CAPTURE_VIRTEX6_inst : CAPTURE_VIRTEX6
generic map (
    ONESHOT => TRUE    -- Specifies the procedure for performing single readback per CAP trigger.
)
port map (
    CAP => CAP, -- 1-bit Readback capture trigger
    CLK => CLK  -- 1-bit Readback capture clock
);

-- End of CAPTURE_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// CAPTURE_VIRTEX6: Virtex-6 Readback Register Capture Control
//           Virtex-6
// Xilinx HDL Language Template, version 11.1

CAPTURE_VIRTEX6 #(
    .ONESHOT("TRUE") // Specifies the procedure for performing single readback per CAP trigger.
)
CAPTURE_VIRTEX6_inst (
    .CAP(CAP), // 1-bit Readback capture trigger
    .CLK(CLK) // 1-bit Readback capture clock
);

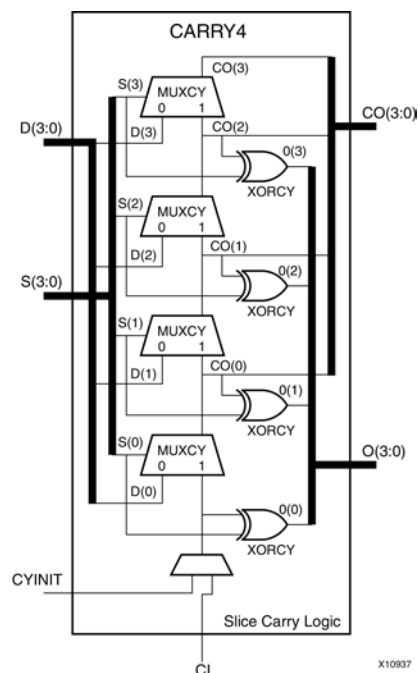
// End of CAPTURE_VIRTEX6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

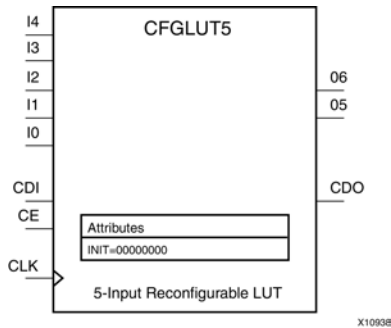
// End of CARRY4_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

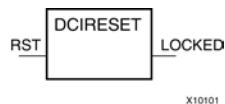
// End of CFGLUT5_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

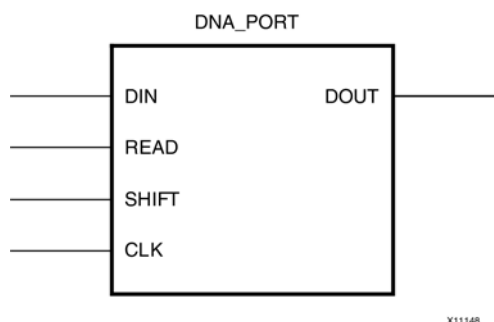
// End of DCIRESET_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## DNA\_PORT

プリミティブ : Device DNA Data Access Port



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

正しく動作するように、入力および出力をすべてデザインに接続します。

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

## 使用可能な属性

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

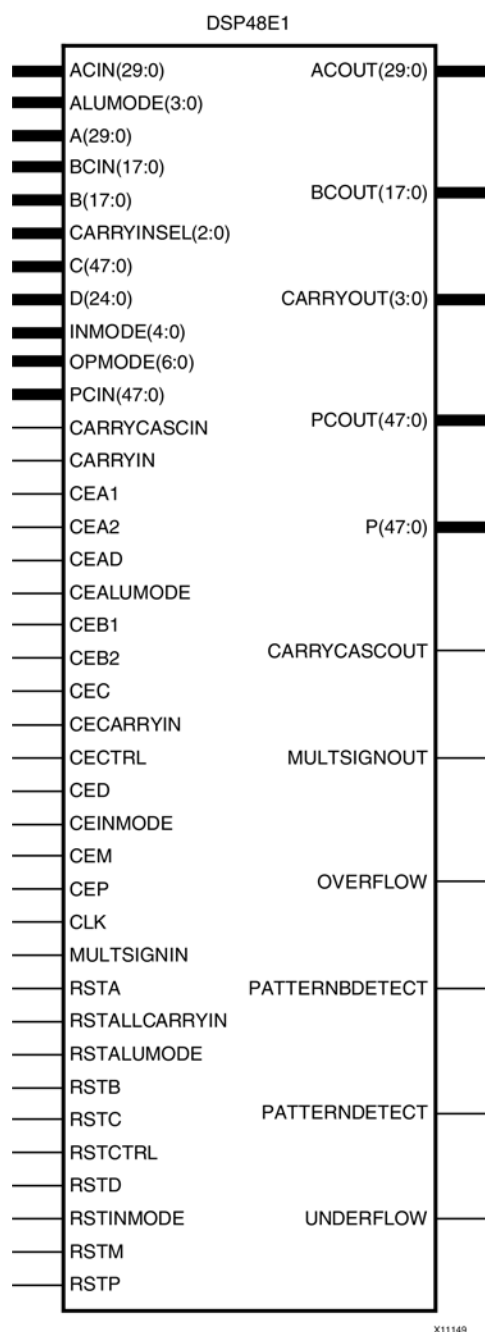
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## DSP48E1

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



### 概要

このデザイン エLEMENTは、Virtex®-6 に含まれる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（前置加算器を含む）、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	タイプ	幅	機能
A[29:0]	入力	30	乗算器/前置加算器への 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
ACIN[29:0]	入力	30	ポート A のカスケード入力、カスケード接続されている上位の DSP スライスの 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。
ACOUT[29:0]	出力	30	ポート A のカスケード出力、カスケード接続されている下位の DSP スライスの ACIN に接続します。使用しない場合は未接続にします。
ALUMODE[3:0]	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。
B[17:0]	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
BCIN[17:0]	入力	18	ポート B のカスケード入力、カスケード接続されている上位の DSP スライスの 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。
BCOUT[17:0]	出力	18	ポート B のカスケード出力、カスケード接続されている下位の DSP スライスの BCIN に接続します。使用しない場合は未接続にします。
C[47:0]	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
CARRYCASCIN	入力	1	上位 DSP スライスからのカスケード キャリー出力
CARRYCASCOUT	出力	1	下位 DSP スライスへのカスケード キャリー出力
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
CARRYINSEL[2:0]	入力	3	DSP スライスへのキャリー入力ソースを選択します。
CARRYOUT[3:0]	出力	4	演算処理 (加算、減算など) のキャリー出力信号。 <ul style="list-style-type: none"> <li>USE_SIMD が FOUR12 の場合、CARRYOUT は累積器/加算器/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。</li> <li>USE_SIMD が TWO24 の場合、CARRYOUT は累積器/加算器それぞれからの 24 ビットのキャリー出力を指します。</li> <li>USE_SIMD が ONE48 の場合、CARRYOUT は累積器/加算器/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。</li> </ul>
CEAD	入力	1	前置加算器出力 AD パイプライン レジスタのアクティブ High のクロック イネーブルです。使用しない場合と ADREG=1 の場合は論理値を 1 に、ADREG=0 の場合は論理値を 0 にします。
CEALUMODE	入力	1	ALUMODE 入力レジスタ (ALUMODEREG=1) のアクティブ High のクロック イネーブルです。使用しない場合は論理値を 1 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。ダイナミック AB アクセスが使用されている場合、このクロック イネーブルは INMODE=1 に適用されます。
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。レジスタが使用されている場合 (AREG=1)、CEA2 がクロック イネーブルです。

ポート名	タイプ	幅	機能
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。ダイナミック AB アクセスが使用されている場合、このクロック イネーブルは INMODE=1 に適用されます。
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。レジスタが使用されている場合 (BREG=1)、CEB2 がクロック イネーブルです。
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。
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。
CECTRL	入力	1	OPMODE および CARRYINSEL レジスタのアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CED	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のクロック イネーブルで、使用しない場合は論理値を 1 にします。
CEINMODE	入力	1	INMODE 入力レジスタ (INMODEREG=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。
CLK	入力	1	DSP スライスのクロック入力
D[24:0]	入力	25	前置加算器への 25 ビット データ入力、または乗算器への入力
INMODE[4:0]	入力	5	ALUMODE および OPMODE と共に使用する DSP スライスの演算処理を選択する制御入力です。INMODE 信号は乗算器 (前置加算器を含む) の前にある信号およびブロックの機能を制御します。
MULTSIGNIN	入力	1	カスケードされた上位 DSP スライスからの乗算器符号出力。48 ビットを超える出力が必要な場合に加算器/累積器 (MACC) の出力の符号拡張のために使用します。MULTSIGNOUT 出力ピンにのみ接続します。
MULTSIGNOUT	出力	1	カスケードされた下位 DSP スライスへの乗算器符号出力。48 ビットを超える出力が必要な場合に加算器/累積器 (MACC) の出力の符号拡張のために使用します。MULTSIGNIN 入力ピンにのみ接続します。
OPMODE[6:0]	入力	7	ALUMODE および INMODE と共に使用する DSP スライスの演算処理を選択する制御入力です。
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。
P[47:0]	出力	48	プライマリ データ出力
PATTERNBDETECT	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。
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。

ポート名	タイプ	幅	機能
PCIN[47:0]	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 DSP スライスの PCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
PCOUT[47:0]	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 DSP スライスの PCIN に接続します。使用しない場合は未接続にします。
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。
RSTALLCARRYIN	入力	1	全キャリー入力レジスタ (CARRYINREG=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。
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。
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。
RSTD	入力	1	D ポートレジスタ (DREG=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。
RSTINMODE	入力	1	INMODE レジスタ (INMODEREG=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。
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	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。
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。

## 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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
ACASCREG	整数	1、0、2	1	AREG 属性と組み合わせて使用し、A カスケード ACOUT 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。
ADREG	整数	1、0	1	前置加算器出力 (AD) パイプライン レジスタの使用を選択します。AD パイプライン レジスタを使用する場合は 1 に設定します。

属性	タイプ	値	デフォルト	説明
A_INPUT	文字列	DIRECT、 CASCADE	DIRECT	A または ACIN のいずれかを選択します。
ALUMODEREG	整数	1、0	1	ALUMODE 入力をレジスタに格納する場合は 1 に設定します。
AREG	整数	1、0、2	1	A 入力にレジスタを付けるかを指定します。
AUTORESET_ PATDET	文字列	NO_RESET、 RESET_MATCH、 RESET_NOT_ MATCH	NO_RESET	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、DSP スライスの R 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。  RESET_MATCH および RESET_NOT_MATCH 設定で、1) パターンが一致したとき、2) パターンが現在のサイクルで一致しないが、前のサイクルでは一致していたとき、DSP スライスで 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。
BCASCREG	整数	1、0、2	1	BREG 属性と組み合わせて使用し、B カスケード BCOUT の B 入力レジスタの数を指定します。ただし、BREG の値以下にする必要があります。
B_INPUT	文字列	DIRECT、 CASCADE	DIRECT	B または BCIN のいずれかを選択します。A 入力にレジスタを付けるかを指定します。
BREG	整数	1、0、2	1	B 入力にレジスタを付けるかを指定します。
CARRYINREG	整数	1、0	1	CARRYIN 入力をレジスタに格納する場合は 1 に設定します。
CARRYINSELREG	整数	1、0	1	CARRYINSEL 入力をレジスタに格納する場合は 1 に設定します。
CREG	整数	1、0	1	C 入力にレジスタを付けるかを指定します。
DREG	整数	1、0	1	D 入力にレジスタを付けるかを指定します。
INMODEREG	整数	1、0	1	INMODE 入力をレジスタに格納する場合は 1 に設定します。
MASK	16 進数	48'h000000 000000 ~ 48'hffffffff	48'h3fff ffffff	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	1、0	1	乗算器の段にレジスタを付けるかを指定します。イネーブル = 1/ディスエーブル = 0
OPMODEREG	整数	1、0	1	OPMODE 入力をレジスタに格納する場合は 1 に設定します。
PATTERN	16 進数	48'h0000000 00000 ~ 48'hffffffff	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	1、0	1	P 入力にレジスタを付けるかを指定します。

属性	タイプ	値	デフォルト	説明
SEL_MASK	文字列	MASK、C、ROUNDING_MODE1、ROUNDING_MODE2	MASK	パターン検出器で使用されるマスクを指定します。C および MASK はパターン検出器で標準的に使用される値です (カウンタ、オーバーフロー検出など)。ROUNDING_MODE1 (C バー 1 つ左にシフト) および ROUNDING_MODE2 (C バー 2 つ左にシフト) は、オプションでレジスタを付けた C ポートに基づき、特別マスクを選択します。これらの丸めモードは、Virtex-6 FPGA DSP48E1 ブロック ユーザー ガイドにあるようにパターン検出器を使用して DSP スライスでの収束丸め機能をインプリメントするのに使用します。
SEL_PATTERN	文字列	PATTERN、C	PATTERN	パターン検出器で使用されるパターンを指定します。
USE_DPORT	ブール代数	FALSE、TRUE	FALSE	前置加算器および D ポートの使用を選択します。
USE_MULT	文字列	MULTIPLY、DYNAMIC、NONE	MULTIPLY	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。DYNAMIC は、ユーザーが A*B と A:B をダイナミックに切り替えていて、この 2 つのパスのワーストケース タイミングを取得することが必要であることを示します。
USE_PATTERN_DETECT	文字列	NO_PATDET、PATDET	NO_PATDET	PATDET を設定するとパターン検出器がシミュレーション モデルおよびスピード ファイルで有効になります。
USE_SIMD	文字列	ONE48、FOUR12、TWO24	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、計算量の比較的小さ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のみに影響し、論理処理には影響しません。

## VHDL 記述 (インスタンスエーション)

```
-- DSP48E1: 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtractor/Accumulator or 2-Input Logic Unit
-- Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

DSP48E1_inst : DSP48E1
generic map (
    ACASCREG => 1,
    ADREG => 1,
    ALUMODEREG => 1,
    AREG => 1,
    AUTORESET_PATDET => "NO_RESET",
    A_INPUT => "DIRECT",
    BCASCREG => 1,
    BREG => 1,
    B_INPUT => "DIRECT",
    CARRYINREG => 1,
    CARRYINSELREG => 1,
    CREG => 1,
    DREG => 1,
    INMODEREG => 1,
    MASK => X"3fffffff",
    -- Number of pipeline registers between A/ACIN input and ACOUT output,
    -- 0, 1, or 2
    -- Number of pipeline registers on pre-adder output, 0 or 1
    -- Number of pipeline registers on ALUMODE input, 0 or 1
    -- Number of pipeline registers on the A input, 0, 1 or 2
    -- NO_RESET, RESET_MATCH, RESET_NOT_MATCH
    -- Selects A input used, "DIRECT" (A port) or "CASCADE" (ACIN port)
    -- Number of pipeline registers between B/BCIN input and BCOUT output,
    -- 0, 1, or 2
    -- Number of pipeline registers on the B input, 0, 1 or 2
    -- Selects B input used, "DIRECT" (B port) or "CASCADE" (BCIN port)
    -- Number of pipeline registers for the CARRYIN input, 0 or 1
    -- Number of pipeline registers for the CARRYINSEL input, 0 or 1
    -- Number of pipeline registers on the C input, 0 or 1
    -- Number of pipeline registers on the D input, 0 or 1
    -- Number of pipeline registers on INMODE input, 0 or 1
    -- 48-bit Mask value for pattern detect
```



```

MREG => 1, -- Number of multiplier pipeline registers, 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
SEL_MASK => "MASK", -- "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
SEL_PATTERN => "PATTERN", -- Select pattern value between the "PATTERN" value or the value on
-- the "C" port
USE_DPORT => FALSE, -- Select D port usage, TRUE or FALSE
USE_MULT => "MULTIPLY", -- Select multiplier usage, "MULTIPLY", "DYNAMIC", or "NONE" (no
-- multiplier)
USE_PATTERN_DETECT => "NO_PATDET", -- Enable pattern detect, "PATDET", "NO_PATDET"
USE_SIMD => "ONE48" -- SIMD selection, "ONE48", "TWO24", "FOUR12"
)
port map (
-- Cascade: 30-bit (each) Cascade Ports
ACOUT => ACOUT, -- 30-bit A port cascade output
BCOUT => BCOUT, -- 18-bit B port cascade output
CARRYCASCOUT => CARRYCASCOUT, -- 1-bit cascade carry output
MULTSIGNOUT => MULTSIGNOUT, -- 1-bit multiplier sign cascade output
PCOUT => PCOUT, -- 48-bit cascade output
-- Control: 1-bit (each) Control Inputs/Status Bits
OVERFLOW => OVERFLOW, -- 1-bit overflow in add/acc output
PATTERNBDETECT => PATTERNBDETECT, -- 1-bit active high pattern bar detect output
PATTERNDETECT => PATTERNDETECT, -- 1-bit active high pattern detect output
UNDERFLOW => UNDERFLOW, -- 1-bit active high underflow in add/acc output
-- Data: 4-bit (each) Data Ports
CARRYOUT => CARRYOUT, -- 4-bit carry output
P => P, -- 48-bit output
-- Cascade: 30-bit (each) Cascade Ports
ACIN => ACIN, -- 30-bit A cascade data input
BCIN => BCIN, -- 18-bit B cascade input
CARRYCASCIN => CARRYCASCIN, -- 1-bit cascade carry input
MULTSIGNIN => MULTSIGNIN, -- 1-bit multiplier sign input
PCIN => PCIN, -- 48-bit P cascade input
-- Control: 4-bit (each) Control Inputs/Status Bits
ALUMODE => ALUMODE, -- 4-bit ALU control input
CARRYINSEL => CARRYINSEL, -- 3-bit carry select input
CEINMODE => CEINMODE, -- 1-bit active high clock enable input for INMODE registers
CLK => CLK, -- 1-bit Clock input
INMODE => INMODE, -- 5-bit INMODE control input
OPMODE => OPMODE, -- 7-bit operation mode input
RSTINMODE => RSTINMODE, -- 1-bit reset input for INMODE pipeline registers
-- Data: 30-bit (each) Data Ports
A => A, -- 30-bit A data input
B => B, -- 18-bit B data input
C => C, -- 48-bit C data input
CARRYIN => CARRYIN, -- 1-bit carry input signal
D => D, -- 25-bit D data input
-- Reset/Clock Enable: 1-bit (each) Reset/Clock Enable Inputs
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
CEAD => CEAD, -- 1-bit active high clock enable input for pre-adder output 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
CED => CED, -- 1-bit active high clock enable input for D registers
CEM => CEM, -- 1-bit active high clock enable input for multiplier registers
CEP => CEP, -- 1-bit active high clock enable input for P registers
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
RSTD => RSTD, -- 1-bit reset input for D pipeline registers
RSTM => RSTM, -- 1-bit reset input for multiplier registers
RSTP => RSTP, -- 1-bit reset input for P pipeline registers
);
-- End of DSP48E1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// DSP48E1: 25x18 Two's Complement Multiplier with Integrated 48-Bit, 3-Input Adder/Subtractor/Accumulator or 2-Input Logic Unit
//          Virtex-6
// Xilinx HDL Language Template, version 11.1

DSP48E1 #(
    .ACASCREG(1),                // Number of pipeline registers between A/ACIN input and ACOUT output,
                                // 0, 1, or 2
    .ADREG(1),                  // Number of pipeline registers on pre-adder output, 0 or 1
    .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_PATDET("NO_RESET"), // NO_RESET, RESET_MATCH, RESET_NOT_MATCH
    .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
    .DREG(1),                  // Number of pipeline registers on the D input, 0 or 1
    .INMODEREG(1),             // Number of pipeline registers on INMODE input, 0 or 1
    .MASK(48'hffffffff),       // 48-bit Mask value for pattern detect
    .MREG(1),                  // Number of multiplier pipeline registers, 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"),         // "C", "MASK", "ROUNDING_MODE1", "ROUNDING_MODE2"
    .SEL_PATTERN("PATTERN"),   // Select pattern value between the "PATTERN" value or the value on the
                                // "C" port
    .USE_DPORT("FALSE"),       // Select D port usage, TRUE or FALSE
    .USE_MULT("MULTIPLY"),     // Select multiplier usage, "MULTIPLY", "DYNAMIC", or "NONE" (no
                                // multiplier)
    .USE_PATTERN_DETECT("NO_PATDET"), // Enable pattern detect, "PATDET", "NO_PATDET"
    .USE_SIMD("ONE48")         // SIMD selection, "ONE48", "TWO24", "FOUR12"
)
DSP48E1_inst (
    // Cascade: 30-bit (each) Cascade Ports
    .ACOUT(ACOUT),              // 30-bit A port cascade output
    .BCOUT(BCOUT),              // 18-bit B port cascade output
    .CARRYCASCOUT(CARRYCASCOUT), // 1-bit cascade carry output
    .MULTSIGNOUT(MULTSIGNOUT),  // 1-bit multiplier sign cascade output
    .PCOUT(PCOUT),              // 48-bit cascade output
    // Control: 1-bit (each) Control Inputs/Status Bits
    .OVERFLOW(OVERFLOW),        // 1-bit overflow in add/acc output
    .PATTERNBDETECT(PATTERNBDETECT), // 1-bit active high pattern bar detect output
    .PATTERNDETECT(PATTERNDETECT), // 1-bit active high pattern detect output
    .UNDERFLOW(UNDERFLOW),      // 1-bit active high underflow in add/acc output
    // Data: 4-bit (each) Data Ports
    .CARRYOUT(CARRYOUT),        // 4-bit carry output
    .P(P),                      // 48-bit output
    // Cascade: 30-bit (each) Cascade Ports
    .ACIN(ACIN),                // 30-bit A cascade data input
    .BCIN(BCIN),                // 18-bit B cascade input
    .CARRYCASCIN(CARRYCASCIN),  // 1-bit cascade carry input
    .MULTSIGNIN(MULTSIGNIN),    // 1-bit multiplier sign input
    .PCIN(PCIN),                // 48-bit P cascade input
    // Control: 4-bit (each) Control Inputs/Status Bits
    .ALUMODE(ALUMODE),          // 4-bit ALU control input
    .CARRYINSEL(CARRYINSEL),    // 3-bit carry select input
    .CEINMODE(CEINMODE),        // 1-bit active high clock enable input for INMODE registers
    .CLK(CLK),                  // 1-bit Clock input
    .INMODE(INMODE),            // 5-bit INMODE control input
    .OPMODE(OPMODE),            // 7-bit operation mode input
    .RSTINMODE(RSTINMODE),      // 1-bit reset input for INMODE pipeline registers
    // Data: 30-bit (each) Data Ports
    .A(A),                      // 30-bit A data input
    .B(B),                      // 18-bit B data input
    .C(C),                      // 48-bit C data input
    .CARRYIN(CARRYIN),          // 1-bit carry input signal

```



```

.D(D), // 25-bit D data input
// Reset/Clock Enable: 1-bit (each) Reset/Clock Enable Inputs
.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
.CEAD(CEAD), // 1-bit active high clock enable input for pre-adder output 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
.CED(CED), // 1-bit active high clock enable input for D registers
.CEM(CEM), // 1-bit active high clock enable input for multiplier registers
.CEP(CEP), // 1-bit active high clock enable input for P registers
.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
.RSTD(RSTD), // 1-bit reset input for D pipeline registers
.RSTM(RSTM), // 1-bit reset input for multiplier registers
.RSTP(RSTP) // 1-bit reset input for P pipeline registers
);

// End of DSP48E1_inst instantiation

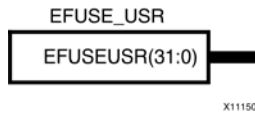
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## EFUSE\_USR

プリミティブ : 32-bit non-volatile design ID



### 概要

このデザイン エLEMENTは、デザインに固有のビット (各デザインに関連した ID など) を格納することができる 32 の不揮発ヒューズに内部アクセスするために使用します。これらのヒューズは JTAG を介して外部に書き込む必要があります。

### ポートの説明

ポート名	タイプ	幅	機能
EFUSEUSR[31:0]	出力	32	ユーザー E ヒューズ レジスタの値

### デザインの入力方法

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

### 使用可能な属性

属性	タイプ	値	デフォルト	説明
SIM_EFUSE_VALUE	16 進数	32'h00000000 ~ 32'hffffff	32'h00000000	INIT が High になった後にシミュレーション モデルでスタティック値がピンで駆動されます。

### VHDL 記述 (インスタンス化)

```
-- EFUSE_USR: 32-bit non-volatile design ID
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

EFUSE_USR_inst : EFUSE_USR
generic map (
    SIM_EFUSE_VALUE => X"00000000" -- Causes simulation model to drive a static value onto these pins after
                                   -- INIT goes high.
)
port map (
    EFUSEUSR => EFUSEUSR -- 32-bit User E-Fuse register value
);

-- End of EFUSE_USR_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// EFUSE_USR: 32-bit non-volatile design ID
//           Virtex-6
// Xilinx HDL Language Template, version 11.1

EFUSE_USR #(
    .SIM_EFUSE_VALUE(32'h00000000) // Causes simulation model to drive a static value onto these pins after
                                   // INIT goes high.
)
EFUSE_USR_inst (
    .EFUSEUSR(EFUSEUSR) // 32-bit User E-Fuse register value
);

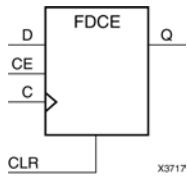
// End of EFUSE_USR_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FDCE

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



### 概要

このデザイン エLEMENTは、クロック イネーブルと非同期クリアがある単一の D タイプ フリップフロップです。クロック イネーブル (CE) が High、非同期クリア (CLR) が Low の場合、クロック (C) が Low から High に切り替わるときにデータ入力 (D) の値がデータ出力 (Q) に送られます。CLR が High になると、ほかのすべての入力は無視され、出力 (Q) の値が Low にリセットされます。CE が Low の場合、クロック遷移は無視されます。

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

### 論理表

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

### デザインの入力方法

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

### 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	2 進数	0	0	コンフィギュレーション後の Q 出力の初期値を指定。  Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDCE の場合、INIT 値は 0 である必要があります。1 に設定する場合は、この動作を表す非同期回路を作成する必要がありますが、ザイリンクスでは推奨されていません。

## VHDL 記述 (インスタンスエーション)

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

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

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

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

-- End of FDCE_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

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

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

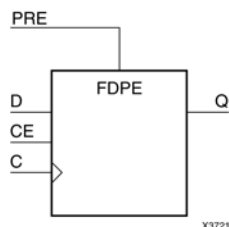
// End of FDCE_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FDPE

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



### 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) 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。非同期の PRE が High になると、ほかのすべての入力は無視され、Q 出力が High にセットされます。PRE が Low、CE が High の場合、クロック (C) 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。CE が Low の場合、クロック遷移は無視されます。

FPGA では、電力を供給すると、フリップフロップは非同期にプリセットされ、出力が High になります。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
PRE	CE	D	C	Q
1	X	X	X	1
0	0	X	X	変化なし
0	1	D	↑	D

### デザインの入力方法

このELEMENTは、回路図でのみ使用できます。

### 使用可能な属性

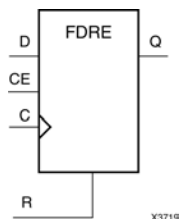
属性	タイプ	値	デフォルト	説明
INIT	2 進数	1	1	コンフィギュレーション後の Q 出力の初期値を指定。  Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDPE の場合、INIT 値は 1 である必要があります。0 に設定する場合は、この動作を表す非同期回路を作成する必要がありますが、ザイリンクスでは推奨されていません。

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FDRE

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



### 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)、同期リセット (R) 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。R 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	CE	D	C	Q
1	X	X	↑	0
0	0	X	X	変化なし
0	1	D	↑	D

### デザインの入力方法

このELEMENTは、回路図でのみ使用できます。

### 使用可能な属性

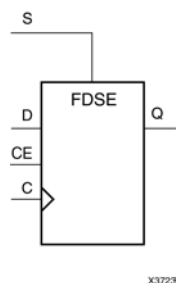
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0	0	コンフィギュレーション後の Q 出力の初期値を指定  Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDRE の場合、INIT 値を 0 に設定する必要があります。1 に設定すると余分なロジックが挿入されます。

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FDSE

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



### 概要

FDSE は、データ (D)、クロック イネーブル (CE)、同期セット (S) 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。同期セット (S) 入力が高レベルになると、クロック イネーブル (CE) 入力は無視され、クロック (C) が Low から High に切り替わるときに Q 出力が高レベルにセットされます。S が Low、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。

FPGA では、電力を供給すると、フリップフロップは非同期にプリセットされ、出力が高レベルになります。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
S	CE	D	C	Q
1	X	X	↑	1
0	0	X	X	変化なし
0	1	D	↑	D

### デザインの入力方法

このエレメントは、回路図でのみ使用できます。

### 使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	2 進数	1	1	コンフィギュレーション後の Q 出力の初期値を指定。  Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDSE の場合、INIT 値を 1 に設定する必要があります。0 に設定すると余分なロジックが挿入されます。

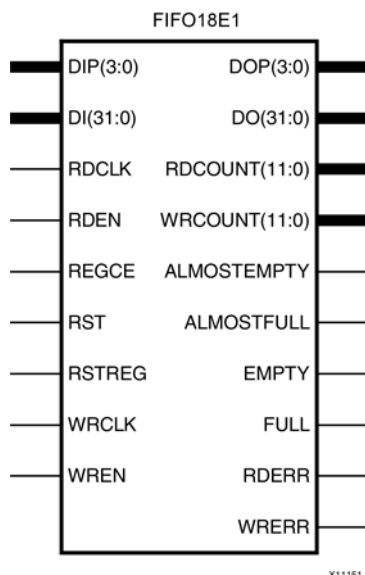
### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## FIFO18E1

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



### 概要

Virtex®-6 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。FIFO18E1 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、9 ビット X 2K、18 ビット X 1K、または 36 ビット X 512 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、同期モードまたはデュアルレート (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 を使用する必要があります。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、FIFO36E1 を使用できます。ただし、エラー修正回路が必用な場合は、FIFO36\_72 の FIFO36E1 を使用する必要があります。

### ポートの説明

ポート名	タイプ	幅	機能
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。
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。
DI[31:0]	入力	32	FIFO データ入力バス
DIP[3:0]	入力	4	FIFO パリティ データ入力バス
DO[31:0]	出力	32	FIFO データ出力バス

ポート名	タイプ	幅	機能
DOP[3:0]	出力	4	FIFO パリティ データ出力バス
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO がフルであることを示します。
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
REGCE	入力	1	パイプライン化された同期 FIFO の出力レジスタ クロック イネーブル
RST	入力	1	3 クロック サイクル間アクティブ Highの (FIFO ロジック) の非同期リセット (デュアル レートの FIFO 向け)、同期リセット (同期 FIFO)
RSTREG	入力	1	出力レジスタの同期セット/リセット
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)
WRCOUNT、RDCOUNT	出力	12	FIFO 書き込み/読み出しポインタ
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
WRERR、RDERR	出力	1	<ul style="list-style-type: none"> <li>WRERR は FIFO がフルの間に書き込みが実行されたことを示します。</li> <li>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。</li> </ul>

## 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
ALMOST_EMPTY_OFFSET	16 進数	13'h0000 ~ 13'h8191	13'h0080	ALMOST_EMPTY フラグをトリガする RAM のデータ量を指定します。
ALMOST_FULL_OFFSET	16 進数	13'h0000 ~ 13'h8191	13'h0080	ALMOST_FULL フラグをトリガする RAM のデータ量を指定します。
DATA_WIDTH	整数	4、9、18、36	4	FIFO に必要なデータ幅を指定
DO_REG	整数	1、0	1	EN_SYN のデータ バイプライン レジスタ
EN_SYN	ブール代数	FALSE、TRUE	FALSE	FIFO がデュアル クロック (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。デュアル クロックの場合は DO_REG=1 である必要があります。
FIFO_MODE	文字列	FIFO18、FIFO18_36	FIFO18	FIFO18 または FIFO18_36 モードを選択します。

属性	タイプ	値	デフォルト	説明
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE、TRUE	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。
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 DO 出力の初期値を指定します。
SRVAL	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの FIFO の出力値を指定します。DO_REG=1 の場合のみ有効です。

## VHDL 記述 (インスタンスレーション)

```
-- FIFO18E1: 18KB FIFO (First In, First Out) Block RAM Memory
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

FIFO18E1_inst : FIFO18E1
generic map (
    ALMOST_EMPTY_OFFSET => X"00080",    -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET  => X"00080",    -- Sets almost full 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 dual-clock ("FALSE") or Synchronous ("TRUE")
    FIFO_MODE => "FIFO18",              -- Sets mode to FIFO18 or FIFO18_36
    FIRST_WORD_FALL_THROUGH => FALSE,    -- Sets the FIFO FWFT to "TRUE" or "FALSE"
    INIT => X"000000000",                -- Initial values on output port
    SRVAL => X"000000000"                -- Set/Reset value for output port
)
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
    -- WRCOUNT, RDCOUNT: 12-bit (each) FIFO write/read pointer.
    RDCOUNT => RDCOUNT,          -- 12-bit read count output
    WRCOUNT => WRCOUNT,        -- 12-bit write count output
    -- WRERR, RDERR: 1-bit (each) WRERR indicates that a write occurred while the FIFO was full and RDERR
    -- indicates that a read occurred while the FIFO was empty
    RDERR => RDERR,              -- 1-bit read error output
    WRERR => WRERR,              -- 1-bit write error
    DI => DI,                    -- 32-bit data input
    DIP => DIP,                  -- 4-bit parity input
    RDEN => RDEN,                -- 1-bit read enable input
    REGCE => REGCE,              -- 1-bit clock enable input
    RST => RST,                  -- 1-bit reset input
    RSTREG => RSTREG,            -- 1-bit output register set/reset
    -- WRCLK, RDCLK: 1-bit (each) FIFO read and write clocks (positive edge
    -- triggered).
    RDCLK => RDCLK,              -- 1-bit read clock input
    WRCLK => WRCLK,              -- 1-bit write clock input
    WREN => WREN                 -- 1-bit write enable input
);

-- End of FIFO18E1_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// FIFO18E1: 18KB FIFO (First In, First Out) Block RAM Memory
//          Virtex-6
// Xilinx HDL Language Template, version 11.1

FIFO18E1 #(
    .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full 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 dual-clock ("FALSE") or Synchronous ("TRUE")
    .FIFO_MODE("FIFO18"),              // Sets mode to FIFO18 or FIFO18_36
    .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to "TRUE" or "FALSE"
    .INIT(36'h0000000000),             // Initial values on output port
    .SRVAL(36'h0000000000)             // Set/Reset value for output port
)
FIFO18E1_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
    // WRCOUNT, RDCOUNT: 12-bit (each) FIFO write/read pointer.
    .RDCOUNT(RDCOUNT),          // 12-bit read count output
    .WRCOUNT(WRCOUNT),          // 12-bit write count output
    // WRERR, RDERR: 1-bit (each) WRERR indicates that a write occurred while the FIFO was full and RDERR
    // indicates that a read occurred while the FIFO was empty
    .RDERR(RDERR),              // 1-bit read error output
    .WRERR(WRERR),              // 1-bit write error
    .DI(DI),                    // 32-bit data input
    .DIP(DIP),                  // 4-bit parity input
    .RDEN(RDEN),                // 1-bit read enable input
    .REGCE(REGCE),              // 1-bit clock enable input
    .RST(RST),                  // 1-bit reset input
    .RSTREG(RSTREG),            // 1-bit output register set/reset
    // WRCLK, RDCLK: 1-bit (each) FIFO read and write clocks (positive edge
    // triggered).
    .RDCLK(RDCLK),              // 1-bit read clock input
    .WRCLK(WRCLK),              // 1-bit write clock input
    .WREN(WREN)                 // 1-bit write enable input
);

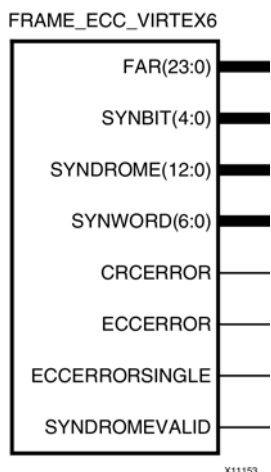
// End of FIFO18E1_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FIFO36E1

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



### 概要

Virtex®-6 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。FIFO36E1 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 フラグと共に、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、72 ビット X 512 ワードの同期またはデュアル クロック (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** : 72 ビット X 512 ワードの FIFO には、FIFO36\_72 を使用する必要があります。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、FIFO18E1 を使用します。ただし、エラー修正回路が必要な場合は、FIFO36\_72 を使用する必要があります。

### ポートの説明

ポート名	タイプ	幅	機能
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。
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。
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。
DI[63:0]	入力	64	FIFO データ入力バス
DIP[7:0]	入力	8	FIFO パリティ データ入力バス
DO[63:0]	出力	64	FIFO データ出力バス

ポート名	タイプ	幅	機能
DOP[7:0]	出力	8	FIFO パリティ データ出力バス
ECCPARITY[7:0]	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
EMPTY	出力	1	FIFO が空であることを示します。
FULL	出力	1	FIFO がフルであることを示します。
INJECTDBITERR	入力	1	ECC 機能が使用されている場合はダブル ビット エラーが挿入されます。
INJECTSBITERR	入力	1	ECC 機能が使用されている場合はシングル ビット エラーが挿入されます。
RDEN	入力	1	アクティブ High の FIFO リード イネーブル
REGCE	入力	1	パイプライン化された同期 FIFO の出力レジスタ クロック イネーブル
RST	入力	1	3 クロック サイクル間アクティブ High の (FIFO ロジック) の非同期リセット (デュアル レートの FIFO 向け)、同期リセット (同期 FIFO)
RSTREG	入力	1	出力レジスタの同期セット/リセット
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。
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)
WRCOUNT、RDCOUNT	出力	13	FIFO 書き込み/読み出しポインタ
WREN	入力	1	アクティブ High の FIFO ライト イネーブル
WRERR、RDERR	出力	1	<ul style="list-style-type: none"> <li>WRERR は FIFO がフルの間に書き込みが実行されたことを示します。</li> <li>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。</li> </ul>

## 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
ALMOST_EMPTY_OFFSET	16 進数	13'h0000 ~ 13'h8191	13'h0080	ALMOST_EMPTY フラグをトリガする RAM のデータ量を指定します。
ALMOST_FULL_OFFSET	16 進数	13'h0000 ~ 13'h8191	13'h0080	ALMOST_FULL フラグをトリガする RAM のデータ量を指定します。
DATA_WIDTH	整数	4, 9, 18, 36, 72	4	FIFO に必要なデータ幅を指定
DO_REG	整数	1, 0	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	FALSE, TRUE	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	FALSE, TRUE	FALSE	ECC エンコーダ回路をイネーブルにします。
EN_SYN	ブール代数	FALSE, TRUE	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。
FIFO_MODE	文字列	FIFO36, FIFO36_72	FIFO36	FIFO36 または FIFO36_72 モードを選択します。
FIRST_WORD_FALL_THROUGH	ブール代数	FALSE, TRUE	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。
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 DO 出力の初期値を指定します。
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの FIFO の出力値を指定します。DO_REG=1 の場合のみ有効です。

## VHDL 記述 (インスタンスエーション)

```

-- FIFO36E1: 36KB FIFO (First In, First Out) Block RAM Memory
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

FIFO36E1_inst : FIFO36E1
generic map (
    ALMOST_EMPTY_OFFSET => X"00080",    -- Sets the almost empty threshold
    ALMOST_FULL_OFFSET  => X"00080",    -- Sets almost full threshold
    DATA_WIDTH         => 4,           -- Sets data width to 4, 9, 18, 36, or 72
    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")
    FIFO_MODE           => "FIFO36",    -- Sets mode to FIFO36 or FIFO36_72
    FIRST_WORD_FALL_THROUGH => FALSE,   -- Sets the FIFO FWFT to "TRUE" or "FALSE"
    INIT                => X"0000000000000000", -- Initial values on output port
    SRVAL               => X"0000000000000000" -- Set/Reset value for output port
)
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,         -- 8-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
SBITERR => SBITERR,          -- 1-bit single bit error status output
-- WRCOUNT, RDCOUNT: 13-bit (each) FIFO write/read pointer.
RDCOUNT => RDCOUNT,          -- 9-bit read count output
WRCOUNT => WRCOUNT,        -- 9-bit write count output
-- WRERR, RDERR: 1-bit (each) WRERR indicates that a write occurred while the FIFO was full and RDERR
-- indicates that a read occurred while the FIFO was empty
RDERR => RDERR,              -- 1-bit read error output
WRERR => WRERR,              -- 1-bit write error
DI => DI,                    -- 64-bit data input
DIP => DIP,                  -- 8-bit parity input
INJECTDBITERR => INJECTDBITERR, -- 1-bit Inject a double bit error if ECC feature is used.
INJECTSBITERR => INJECTSBITERR, -- 1-bit Inject a single bit error if ECC feature is used.
RDEN => RDEN,                -- 1-bit read enable input
REGCE => REGCE,              -- 1-bit clock enable input
RST => RST,                   -- 1-bit reset input
RSTREG => RSTREG,            -- 1-bit output register set/reset
-- WRCLK, RDCLK: 1-bit (each) FIFO read and write clocks (positive edge
triggered).
RDCLK => RDCLK,              -- 1-bit read clock input
WRCLK => WRCLK,              -- 1-bit write clock input
WREN => WREN                  -- 1-bit write enable input
);

-- End of FIFO36E1_inst instantiation
```



## Verilog 記述 (インスタンス化)

```
// FIFO36E1: 36KB FIFO (First In, First Out) Block RAM Memory
//      Virtex-6
// Xilinx HDL Language Template, version 11.1

FIFO36E1 #(
    .ALMOST_EMPTY_OFFSET(13'h0080),    // Sets the almost empty threshold
    .ALMOST_FULL_OFFSET(13'h0080),     // Sets almost full threshold
    .DATA_WIDTH(4),                    // Sets data width to 4, 9, 18, 36, or 72
    .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")
    .FIFO_MODE("FIFO36"),               // Sets mode to FIFO36 or FIFO36_72
    .FIRST_WORD_FALL_THROUGH("FALSE"), // Sets the FIFO FWFT to "TRUE" or "FALSE"
    .INIT(72'h00000000000000000000),   // Initial values on output port
    .SRVAL(72'h00000000000000000000)   // Set/Reset value for output port
)
FIFO36E1_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),                         // 8-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
    .SBITERR(SBITERR),                 // 1-bit single bit error status output
    // WRCOUNT, RDCOUNT: 13-bit (each) FIFO write/read pointer.
    .RDCOUNT(RDCOUNT),                 // 9-bit read count output
    .WRCOUNT(WRCOUNT),                 // 9-bit write count output
    // WRERR, RDERR: 1-bit (each) WRERR indicates that a write occurred while the FIFO was full and RDERR
    // indicates that a read occurred while the FIFO was empty
    .RDERR(RDERR),                     // 1-bit read error output
    .WRERR(WRERR),                     // 1-bit write error
    .DI(DI),                           // 64-bit data input
    .DIP(DIP),                         // 8-bit parity input
    .INJECTDBITERR(INJECTDBITERR),     // 1-bit Inject a double bit error if ECC feature is used.
    .INJECTSBITERR(INJECTSBITERR),     // 1-bit Inject a single bit error if ECC feature is used.
    .RDEN(RDEN),                       // 1-bit read enable input
    .REGCE(REGCE),                     // 1-bit clock enable input
    .RST(RST),                         // 1-bit reset input
    .RSTREG(RSTREG),                   // 1-bit output register set/reset
    // WRCLK, RDCLK: 1-bit (each) FIFO read and write clocks (positive edge
    // triggered).
    .RDCLK(RDCLK),                     // 1-bit read clock input
    .WRCLK(WRCLK),                     // 1-bit write clock input
    .WREN(WREN)                        // 1-bit write enable input
);

// End of FIFO36E1_inst instantiation
```

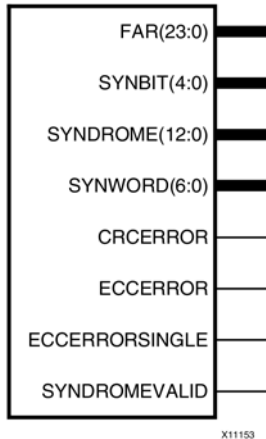
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## FRAME\_ECC\_VIRTEX6

プリミティブ : Virtex®-6 Configuration Frame Error Detection and Correction Circuitry

FRAME\_ECC\_VIRTEX6



X11153

### 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。

SEU 修正機能では、シングル ビット エラーの自動修正を行うためのハードウェア バージョンが提供されます。この修正機能で使用する追加出力には、ソフト コアで使用するハミング コード シンドロームのデコードが含まれます。

### ポートの説明

ポート名	タイプ	幅	機能
CRCERROR	出力	1	リードバック CRC エラーを示します。
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 以外のときは 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。
ECCERRORSINGLE	出力	1	シングル ビット フレーム ECC エラが検出されたことを示します。
FAR[23:0]	出力	24	SEU 修正/投入および ICAP アプリケーションでは FAR レジスタを確認できます。この出力は、FARSRC 属性により EFAR または FAR コンフィギュレーション レジスタをポイントします。
SYNBIT[4:0]	出力	5	フレーム内の SYNWORD でポイントされているワード内のエラーのあるビットのインデックス (0 → 31) が検出されたことを示します。ECCERRORSINGLE が High のとき有効です。
SYNDROME[12:0]	出力	13	フレーム ECC エラーの発生場所を示します。 <ul style="list-style-type: none"> <li>S[12] = 0, S[11:0] = 0 : エラーなし</li> <li>S[12] = 1, S[11:0] ≠ 0 : シングル ビット (SED) エラー。S[11:0] でエラー位置が示されます。</li> <li>S[12] = 1, S[11:0] = 0 : シングル ビット エラー。全体パリティビット p[11] がエラー</li> <li>S[12] = 0, S[11:0] ≠ 0 : ダブルビット エラー、修正不可能</li> </ul>

ポート名	タイプ	幅	機能
SYNDROMEVALID	出力	1	SYNDROME の値が有効であることを示します。
SYNWORD[6:0]	出力	7	ECC エラーが検出されたフレームでの 32 ビットワードのインデックス (0 → 80)。SYNDROME からデコードされます。ECCERRORSINGLE が High のとき有効です。

## デザインの入力方法

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

## 使用可能な属性

属性	タイプ	値	デフォルト	説明
FARSRC	文字列	EFAR、FAR	EFAR	FAR または EFAR コンフィギュレーションレジスタのどちらを FAR[23:0] の出力でポイントするかを決定します。コンフィギュレーション オプションレジスタビット CTL0[7] を設定します。

## VHDL 記述 (インスタンス化)

```
-- FRAME_ECC_VIRTEX6: Virtex-6 Configuration Frame Error Detection and Correction Circuitry
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

FRAME_ECC_VIRTEX6_inst : FRAME_ECC_VIRTEX6
generic map (
    FARSRC => "EFAR" -- Determines if the output of FAR[23:0] is pointed to the FAR or EFAR configuration
                      -- register. Sets config option register bit REG_TBD[bit].
)
port map (
    CRCERROR => CRCERROR,          -- 1-bit Readback CRC error.
    ECCERROR => ECCERROR,          -- 1-bit Frame ECC error found. Value is a one when SYNDROME is
    -- non-zero and a zero when SYNDROME is all zeroes indicating no errors
    -- detected.

    ECCERRORSINGLE => ECCERRORSINGLE, -- 1-bit Indicates single-bit Frame ECC error detected.
    FAR => FAR,                     -- 24-bit SEU Correction/Injection and ICAP applications can benefit
    -- from being able to see the FAR register This output can point to the
    -- EFAR or FAR configuration register depending on the FARSRC
    -- attribute.

    SYNBIT => SYNBIT,              -- 5-bit The index (0 → 31) of the bit w/ error in the word pointed to
    -- by SYNWORD in the frame detected. Valid when ECCERRORSINGLE is high.

    SYNDROME => SYNDROME,          -- 13-bit Frame ECC error where: No errors: All zeros One bit error:
    -- SYNDROME[11]=0, SYNDROME[10:0]= location of error in FRAME Two bit
    -- errors: SYNDROME[11]=1, SYNDROME[10:0]=dont care More than two bit
    -- errors: Unknown output.

    SYNDROMEVALID => SYNDROMEVALID, -- 1-bit Frame ECC output indicating that the value on SYNDROME is
    -- valid.

    SYNWORD => SYNWORD             -- 7-bit The index (0 → 80) of the 32-bit word in the frame where an
    -- ECC error has been detected. Decoded from SYNDROME. Valid when
    -- ECCERRORSINGLE is high.
);
```

```
-- End of FRAME_ECC_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// FRAME_ECC_VIRTEX6: Virtex-6 Configuration Frame Error Detection and Correction Circuitry
//                               Virtex-6
// Xilinx HDL Language Template, version 11.1

FRAME_ECC_VIRTEX6 #(
    .FARSRC("EFAR") // Determines if the output of FAR[23:0] is pointed to the FAR or EFAR configuration
                    // register. Sets config option register bit REG_TBD[bit].
)
FRAME_ECC_VIRTEX6_inst (
    .CRCERROR(CRCERROR), // 1-bit Readback CRC error.
    .ECCERROR(ECCERROR), // 1-bit Frame ECC error found. Value is a one when SYNDROME is non-zero
                        // and a zero when SYNDROME is all zeroes indicating no errors detected.

    .ECCERRORSINGLE(ECCERRORSINGLE), // 1-bit Indicates single-bit Frame ECC error detected.
    .FAR(FAR), // 24-bit SEU Correction/Injection and ICAP applications can benefit
                // from being able to see the FAR register This output can point to the
                // EFAR or FAR configuration register depending on the FARSRC attribute.

    .SYNBIT(SYNBIT), // 5-bit The index (0 -> 31) of the bit w/ error in the word pointed to
                    // by SYNWORD in the frame detected. Valid when ECCERRORSINGLE is high.

    .SYNDROME(SYNDROME), // 13-bit Frame ECC error where: No errors: All zeros One bit error:
                        // SYNDROME[11]=0, SYNDROME[10:0]= location of error in FRAME Two bit
                        // errors: SYNDROME[11]=1, SYNDROME[10:0]=dont care More than two bit
                        // errors: Unknown output.

    .SYNDROMEVALID(SYNDROMEVALID), // 1-bit Frame ECC output indicating that the value on SYNDROME is valid.
    .SYNWORD(SYNWORD) // 7-bit The index (0 -> 80) of the 32-bit word in the frame where an
                    // ECC error has been detected. Decoded from SYNDROME. Valid when
                    // ECCERRORSINGLE is high.
);

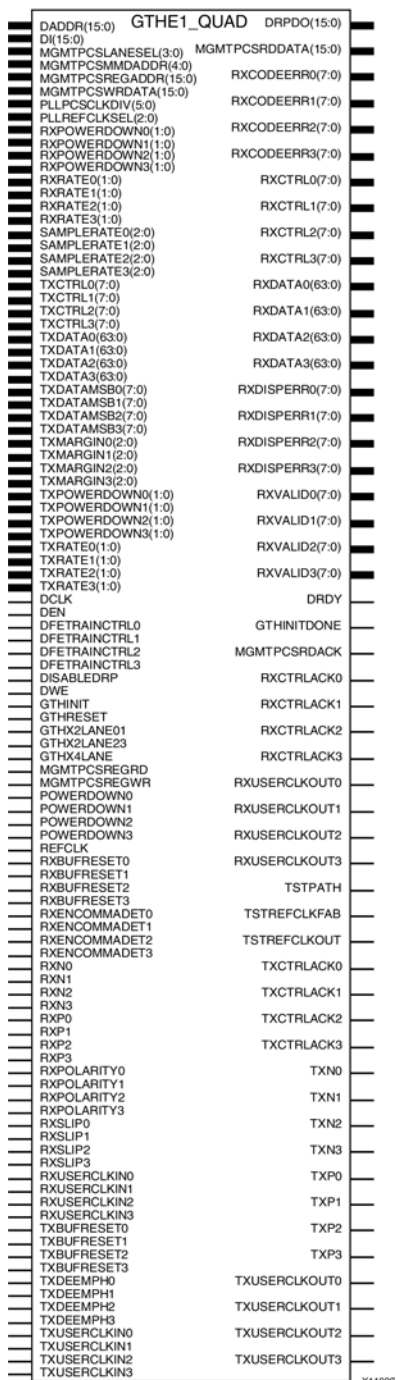
// End of FRAME_ECC_VIRTEX6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## GTHE1\_QUAD

ブリミティブ : Gigabit Transceiver



X11000

## 概要

Virtex®-6 FPGA GTH トランシーバのデザイン エLEMENT です。GTH は HTX ファミリの Virtex-6 FPGA の中で最高速、10G で最適化されたコンフィギュレーション可能なトランシーバです。このELEMENTの詳細は、Virtex-6 FPGA GTH トランシーバ ユーザー ガイドを参照してください。Virtex-6 FPGA GTX Transceiver Wizard は、GTHE1\_QUAD プリミティブをインスタンスエートするラッパの生成に使用されるツールです。このウィザードは、ザイリンクス CORE Generator™ ツールに含まれています。

## デザインの入力方法

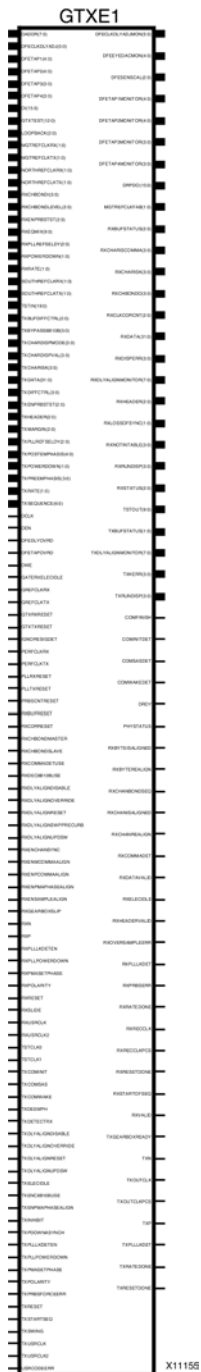
このELEMENTをインスタンスエートするには、Virtex-6 FPGA GTH Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンスエートしないでください。

## 詳細情報

- ・ [Virtex-6 FPGA GTH トランシーバ ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

# GTXE1

プリミティブ : Gigabit Transceiver



## 概要

このデザイン エLEMENTでは、効率性に優れ詳細にコンフィギュレーション可能なトランシーバである Virtex®-6 FPGA RocketIO™ GTX トランシーバが表現されます。このELEMENTの詳細は、Virtex-6 FPGA RocketIO GTX トランシーバ ユーザー ガイドを参照してください。Virtex-6 FPGA RocketIO GTX Transceiver Wizard は、GTXE1 プリミティブをインスタンス化してラッパの生成に使用されるツールです。このウィザードは、ザイリンクス CORE Generator™ ツールに含まれています。

## デザインの入力方法

このELEMENTをインスタンス化するには、Virtex-6 FPGA RocketIO GTX Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンス化しないでください。

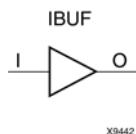
## 詳細情報

- ・ [Virtex-6 FPGA RocketIO GTX トランシーバ ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## 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)
);

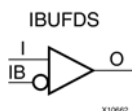
// End of IBUF_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS

**プリミティブ** : Differential Signaling Input Buffer



### 概要

このデザイン エLEMENT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 ポートをこの入力に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
IOSTANDARD	文字列	データシートを参照	DEFAULT	I/O 規格をELEMENTに割り当て

## VHDL 記述 (インスタンスレーション)

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

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

-- IBUFDS: 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)
);

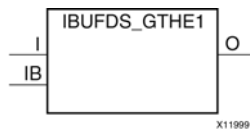
// End of IBUFDS_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_GTHE1

プリミティブ : Differential Clock Input for the GTH Transceiver Reference Clocks



### 概要

GTH トランシーバ リファレンス クロックの専用差動クロック入力です。4 区画ある GTH の区画ごとに IBUFGDS\_GTHE1 コンポーネントが 1 つあり、GTHE1\_QUAD プリミティブの REFCLK ピンに直接接続されます。

### デザインの入力方法

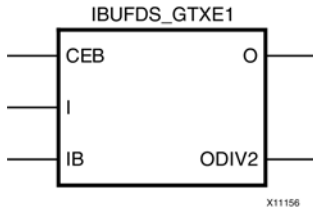
このエレメントをインスタンス化するには、Virtex-6 FPGA GTH Transceiver Wizard またはこのエレメントを含む関連コアを使用します。このエレメントは直接インスタンス化しないでください。

### 詳細情報

- ・ [Virtex-6 FPGA GTH トランシーバ ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFDS\_GTXE1

プリミティブ : Differential Clock Input for the Transceiver Reference Clocks



### 概要

GT のリファレンス クロック入力だけでなく BUFG/MMCM などのクロック リソースを駆動することもできる、トランシーバ リファレンス クロックの差動クロック入力です。IBUFDS\_GTXE1 に関連したクワッドの 4 GTXE1にある MGTREFCLKRX/TX ピン、その上のクワッドの 4 GTXE1にある NORTHREFCLKRX/TX ピン、または下のクワッドの 4 GTXE1 にある SOUTHREFCLKRX/TX ピンに接続します。

IBUFDS\_GTXE1 エLEMENTが接続できるデスティネーション ピンは Virtex®-6 には複数あります。GT のリファレンス クロ ックが接続されている場合、最適化された配線に基づき、どのピンの GT にもソフトウェアによるフル コントロールのもと配線 接続ができます。複数のクロックが GT に接続されている場合、ソフトウェアにより各 IBUFDS が GT 上の指定ピンに配線さ れます。つまり、IBUFDS\_GTXE1 の O ピンは、GT の MGTREFCLKRX/TX ピンまたは NORTH/SOUTHREFCLKRX/TX ピンに接続されます。

**メモ :** RX および TX のマルチプレクサはそれぞれ別を選択できますが、配線はシリコン上で共有されます。

### デザインの入力方法

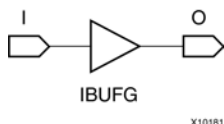
このELEMENTをインスタンスエートするには、RocketIO™ Wizard またはこのELEMENTを含む関連コアを使用します。こ のELEMENTは直接インスタンスエートしないでください。

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IBUFG

**プリミティブ** : Dedicated Input Clock Buffer



### 概要

IBUFG は、FPGA への入力クロックをグローバル クロック配線リソースに接続するために使用する専用入力です。IBUFG は最上位ポートから MMCM または 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 ケーパブル (CC) ピンまたはグローバル クロック (GC) ピンでのみ駆動できます。

### 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
```

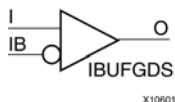
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## IBUFGDS

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



### 概要

このデザイン エLEMENT は、クロック バッファ (BUFG) または MMCM 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 MMCM、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
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
);

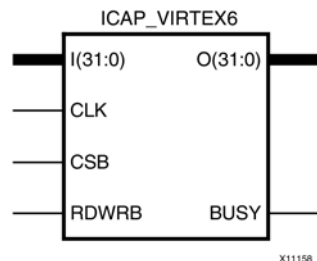
// End of IBUFGDS_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ICAP\_VIRTEX6

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	タイプ	幅	機能
BUSY	出力	1	Busy/Ready 出力
CLK	入力	1	クロック入力
CSB	入力	1	アクティブ Low の ICAP イネーブル
I[31:0]	入力	32	コンフィギュレーション データ入力バス
O[31:0]	出力	32	コンフィギュレーション データ出力バス
RDWRB	入力	1	読み出し/書き込みの選択

### デザインの入力方法

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

### 使用可能な属性

属性	タイプ	値	デフォルト	説明
ICAP_WIDTH	文字列	X8、X16、X32	X8	ICAP_VIRTEX6 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)

```
-- ICAP_VIRTEX6: Internal Configuration Access Port
--                      Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

ICAP_VIRTEX6_inst : ICAP_VIRTEX6
generic map (
    ICAP_AUTO_SWITCH => "DISABLE",
    ICAP_WIDTH => "X8"           -- Specifies the input and output data width to be used with the
                                -- ICAP_VIRTEX6.
)
port map (
    BUSY => BUSY,    -- 1-bit Busy/Ready output
    O => O,          -- 32-bit Configuration data output bus
    CLK => CLK,      -- 1-bit Clock Input
    CSB => CSB,      -- 1-bit Active-Low ICAP Enable
    I => I,          -- 32-bit Configuration data input bus
    RDWRB => RDWRB   -- 1-bit Read/Write Select
);

-- End of ICAP_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

```
// ICAP_VIRTEX6: Internal Configuration Access Port
//                      Virtex-6
// Xilinx HDL Language Template, version 11.1

ICAP_VIRTEX6 #(
    .ICAP_AUTO_SWITCH("DISABLE"),
    .ICAP_WIDTH("X8")           // Specifies the input and output data width to be used with the
                                // ICAP_VIRTEX6.
)
ICAP_VIRTEX6_inst (
    .BUSY(BUSY),    // 1-bit Busy/Ready output
    .O(O),          // 32-bit Configuration data output bus
    .CLK(CLK),      // 1-bit Clock Input
    .CSB(CSB),      // 1-bit Active-Low ICAP Enable
    .I(I),          // 32-bit Configuration data input bus
    .RDWRB(RDWRB)   // 1-bit Read/Write Select
);

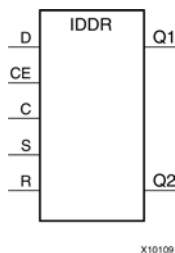
// End of ICAP_VIRTEX6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

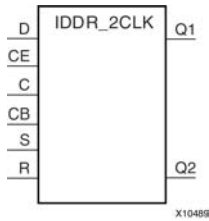
// End of IDDR_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、必用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
);

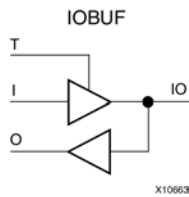
// End of IDDR_2CLK_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUF

プリミティブ : Bi-Directional Buffer



## 概要

このデザイン エLEMENT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
);

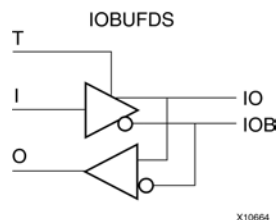
// End of IOBUF_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IOBUFDS

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



### 概要

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

### 論理表

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

### ポートの説明

ポート名	方向	幅	機能
O	出力	1	バッファの出力
I/O	入出力	1	Diff_p 入出力
IOB	入出力	1	Diff_n 入出力
I	入力	1	バッファの入力
T	入力	1	トライステート 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;

-- 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 (インスタンスエーション)

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

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

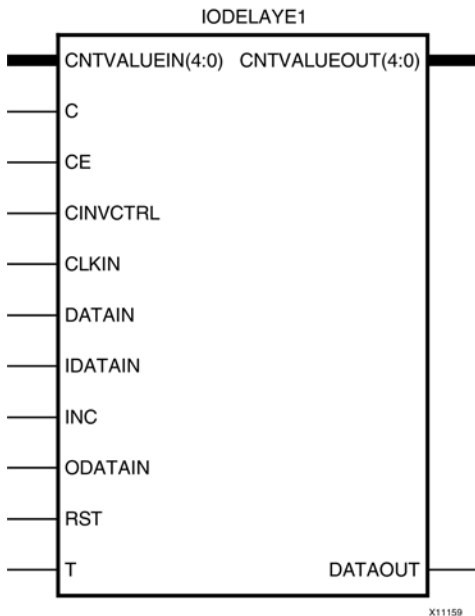
// End of IOBUFDS_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## IODELAYE1

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、Virtex®-6 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。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、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	タイプ	幅	機能
C	入力	1	クロック入力 (必ず VARIABLE モードに設定)
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。
CINVCTRL	入力	1	クロック (C) の極性をダイナミックに反転させます。
CLKIN	入力	1	クロックが IODELAY にアクセスします (IO CLKMUX から)。
CNTVALUEIN[4:0]	入力	5	ロード可能なカウンタ アプリケーションに対するデバイスからのカウンタ値
CNTVALUEOUT[4:0]	出力	5	監視目的のデバイスへのカウンタ値
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。
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。



ポート名	タイプ	幅	機能
INC	入力	1	インクリメント/デクリメント タップ遅延
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) する必要があります。
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。
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。

## 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
CINVCTRL_SEL	ブール代数	FALSE、TRUE	FALSE	クロック (C) の極性をダイナミックに反転させます。
DELAY_SRC	文字列	I、CLKIN、DATAIN、IO、O	I	IODELAY コンポーネントのソースを指定します。 <ul style="list-style-type: none"> <li>I は入力ポートまたは IBUF (入力モード) に直接接続します。</li> <li>O は出力ポートまたは OBUF (出力モード) に接続します。</li> <li>IO はポートに接続します。</li> <li>DATAIN はどのポート (内部モード) にも接続しません。</li> </ul>
HIGH_PERFORMANCE_MODE	ブール代数	TRUE、FALSE	TRUE	TRUE の場合は、出力ジッタを減少させます。
IDELAY_TYPE	文字列	DEFAULT、FIXED、VARIABLE、VAR_LOADABLE	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、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	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_TYPE	文字列	FIXED、VARIABLE、VAR_LOADABLE	FIXED	出力遅延タイプに DEFAULT (ホールドタイムの削除)、FIXED、VARIABLE のいずれかを指定

属性	タイプ	値	デフォルト	説明
ODELAY_VALUE	整数	0、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	0	出力パス遅延のタップ数を指定
REFCLK_FREQUENCY	1 上位ビット FLOAT	175.0 ~ 225.0	200.0	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	DATA、CLOCK	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 (インスタンスエーション)

```
-- IDELAYE1: Input and Output Fixed or Variable Delay Element
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

IDELAYE1_inst : IDELAYE1
generic map (
    CINVCTRL_SEL => FALSE,          -- Dynamically inverts the Clock (C) polarity.
    DELAY_SRC => "I",              -- Specifies the source to the IDELAY component. "I" means it will be
                                -- connected directly to an input port or IBUF (input mode), "O" means
                                -- it will be connected to an output port or OBUF (output mode), "IO"
                                -- means it will be connected to a port, and "DATAIN" means it will not
                                -- be connected to any port (internal mode).
    HIGH_PERFORMANCE_MODE => TRUE,  -- When TRUE, this attribute reduces the output jitter.
    IDELAY_TYPE => "DEFAULT",       -- Specifies a fixed, variable or default (eliminate hold time) input
                                -- delay.
    IDELAY_VALUE => 0,              -- Specifies the number of taps of delay for the input path when in
                                -- fixed mode or the initial delay tap value for variable mode.
    ODELAY_TYPE => "FIXED",         -- Specifies the number of taps of delay for the output path.
    ODELAY_VALUE => 0,              -- When using an associated IDELAYCTRL, specifies the input reference
    REFCLK_FREQUENCY => 200.0,      -- frequency to the component.
    SIGNAL_PATTERN => "DATA"        -- Used by the delay calculator to determine different propagation
                                -- delays through the IDELAY block based on the setting. DATA will be
                                -- the addition of per tap delay and per tap jitter. No jitter is
                                -- introduced for clock-like signals.
)
port map (
    CNTVALUEOUT => CNTVALUEOUT,    -- 5-bit Counter value going to fabric for monitoring purpose
    DATAOUT => DATAOUT,          -- 1-bit Delayed data output from input port (connect to input datapath logic)
    C => C,                        -- 1-bit Clock input (Must be connected for variable mode)
    CE => CE,                      -- 1-bit Active high enable increment/decrement function
    CINVCTRL => CINVCTRL,          -- 1-bit Dynamically inverts the Clock (C) polarity
    CLKIN => CLKIN,                -- 1-bit Clock Access into the IDELAY (from the IO CLKMUX)
    CNTVALUEIN => CNTVALUEIN,      -- 5-bit Counter value from fabric for loadable counter application
    DATAIN => DATAIN,            -- 1-bit Data input for the internal datapath delay. When DATAIN is used,
                                -- IDATAIN and ODATAIN must be tied to a logic zero (ground).

    IDATAIN => IDATAIN,            -- 1-bit Data input to device from the I/O (connect directly to port, I/O
                                -- Buffer). When IDATAIN is used, DATAIN must be tied to a logic zero
                                -- (ground).

    INC => INC,                    -- 1-bit Increment / Decrement tap delay
    ODATAIN => ODATAIN,            -- 1-bit Data input for the output datapath from the device (connect to
                                -- output data source). When ODATAIN is used, DATAIN must be tied to a logic
                                -- zero (ground).

    RST => RST,                    -- 1-bit Active high, synchronous reset, resets delay chain to IDELAY_VALUE/
                                -- ODELAY_VALUE tap. If no value is specified, the default is 0.

    T => T                          -- 1-bit 3-state input control. Tie high for input-only or internal delay or
                                -- tie low for output only.
```

```
);

-- End of IODELAYE1_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// IODELAYE1: Input and Output Fixed or Variable Delay Element
//          Virtex-6
// Xilinx HDL Language Template, version 11.1

IODELAYE1 #(
    .CINVCTRL_SEL("FALSE"),           // Dynamically inverts the Clock (C) polarity.
    .DELAY_SRC("I"),                  // Specifies the source to the IODELAY component. "I" means it will be
                                      // connected directly to an input port or IBUF (input mode), "O" means it
                                      // will be connected to an output port or OBUF (output mode), "IO" means
                                      // it will be connected to a port, and "DATAIN" means it will not be
                                      // connected to any port (internal mode).
    .HIGH_PERFORMANCE_MODE("TRUE"),   // When TRUE, this attribute reduces the output jitter.
    .IDELAY_TYPE("DEFAULT"),          // Specifies a fixed, variable or default (eliminate hold time) input
                                      // delay.
    .IDELAY_VALUE(0),                 // Specifies the number of taps of delay for the input path when in fixed
                                      // mode or the initial delay tap value for variable mode.
    .ODELAY_TYPE("FIXED"),            // Specifies the number of taps of delay for the output path.
    .ODELAY_VALUE(0),                // When using an associated IDELAYCTRL, specifies the input reference
    .REFCLK_FREQUENCY(200.0),         // frequency to the component.
    .SIGNAL_PATTERN("DATA")          // Used by the delay calculator to determine different propagation delays
                                      // through the IODELAY block based on the setting. DATA will be the
                                      // addition of per tap delay and per tap jitter. No jitter is introduced
                                      // for clock-like signals.
)
IODELAYE1_inst (
    .CNTVALUEOUT(CNTVALUEOUT), // 5-bit Counter value going to fabric for monitoring purpose
    .DATAOUT(DATAOUT),         // 1-bit Delayed data output from input port (connect to input datapath logic)
    .C(C),                     // 1-bit Clock input (Must be connected for variable mode)
    .CE(CE),                   // 1-bit Active high enable increment/decrement function
    .CINVCTRL(CINVCTRL),       // 1-bit Dynamically inverts the Clock (C) polarity
    .CLKIN(CLKIN),             // 1-bit Clock Access into the IODELAY (from the IO CLKMUX)
    .CNTVALUEIN(CNTVALUEIN),    // 5-bit Counter value from fabric for loadable counter application
    .DATAIN(DATAIN),           // 1-bit Data input for the internal datapath delay. When DATAIN is used,
    // IDATAIN and ODATAIN must be tied to a logic zero (ground).

    .IDATAIN(IDATAIN),         // 1-bit Data input to device from the I/O (connect directly to port, I/O
    // Buffer). When IDATAIN is used, DATAIN must be tied to a logic zero
    // (ground).

    .INC(INC),                 // 1-bit Increment / Decrement tap delay
    .ODATAIN(ODATAIN),         // 1-bit Data input for the output datapath from the device (connect to output
    // data source). When ODATAIN is used, DATAIN must be tied to a logic zero
    // (ground).

    .RST(RST),                 // 1-bit Active high, synchronous reset, resets delay chain to IDELAY_VALUE/
    // ODELAY_VALUE tap. If no value is specified, the default is 0.

    .T(T)                      // 1-bit 3-state input control. Tie high for input-only or internal delay or
    // tie low for output only.
);

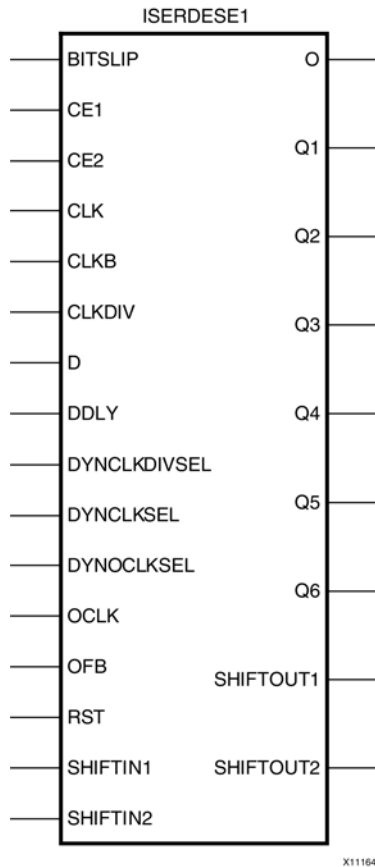
// End of IODELAYE1_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ISERDESE1

プリミティブ : Input SERial/DESerializer



### 概要

このエレメントは、高速ソース同期アプリケーションのインプリメンテーションに特化したクロックおよびロジック機能を持つ、専用シリアル/パラレル コンバータです。FPGA でデシリアライザを設計する際の複雑なタイミング問題を避け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
BITSLLIP	入力	1	入力データ BITSLLIP イネーブル
CE1	入力	1	入力データレジスタのクロック イネーブル
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	セカンダリ クロック入力 <ul style="list-style-type: none"> <li>1 クロック DDR モード (DATA_RATE="DDR") を使用する場合は、CLK ピンに接続されているクロックを反転し、CLKB ピンに接続します。</li> </ul>

ポート名	タイプ	幅	機能
			<ul style="list-style-type: none"> <li>2 クロック DDR モードを使用する場合は、独立した位相シフトクロックを CLKB ピンに接続します。</li> <li>シングル データレート モード (DATA_RATE="SDR") を使用する場合は、このピンを未接続にするかグランドの接続します。</li> </ul>
CLKDIV	入力	1	パラレル データに使用する分周クロック
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
DDLY	入力	1	IODELAY からのシリアル入力
DYNCLKDIVSEL	入力	1	オプションの反転を使用し CLKDIV または CLKDIV_B をダイナミックに選択します。
DYNCLKSEL	入力	1	オプションの反転を使用し CLK または CLK_B をダイナミックに選択します。
O	出力	1	組み合わせ出力
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
OCLKB	入力	1	非同同期オーバーサンプリングに使用されます。
OFB	入力	1	OLOGIC/OSERDES 出力 (ODELAY があるものまたはないもの) からのフィードバック バス
Q1 ~ Q6	出力	1	レジスタ付きパラレル入力データ
RST	入力	1	SERDES のレジスタのアクティブ High の非同同期リセット
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。このピンはグランドに接続する必要があります。
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。

## 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
DATA_RATE	文字列	DDR、SDR	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	4、2、3、5、6、7、8、10	4	パラレル データ幅を選択
DYN_CLKDIV_INV_EN	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、メモリ インターフェイスのダイナミック CLKDIV 反転をオン
DYN_CLK_INV_EN	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、メモリ インターフェイスのダイナミック CLK 反転をオン

属性	タイプ	値	デフォルト	説明
DYN_OCLK_INV_EN	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、メモリ インターフェイスのダイナミック OCLK 反転をオン
INIT_Q1	2 進数	1'b0 ~ 1'b1	1'b0	Q1 出力の初期値を指定
INIT_Q2	2 進数	1'b0 ~ 1'b1	1'b0	Q2 出力の初期値を指定
INIT_Q3	2 進数	1'b0 ~ 1'b1	1'b0	Q3 出力の初期値を指定
INIT_Q4	2 進数	1'b0 ~ 1'b1	1'b0	Q4 出力の初期値を指定
INTERFACE_TYPE	文字列	MEMORY、MEMORY_DDR3、MEMORY_QDR、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
IOBDelay	文字列	NONE、BOTH、IBUF、IFD	NONE	レジスタを介した (Q1 ~ Q6) 出力または組み合わせパス (O) 出力への入力に遅延のあるもの、または遅延のないものを使用することができます。
NUM_CE	整数	2、1	2	ISERDES_NODELAY に使用するクロック イネーブルの数を指定
OFB_USED	ブール代数	FALSE、TRUE	FALSE	OLOGIC、OSERDES からのスニーク パスを選択
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
SRVAL_Q1	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の Q1 出力の値を指定
SRVAL_Q2	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の Q2 出力の値を指定
SRVAL_Q3	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の Q3 出力の値を指定
SRVAL_Q4	2 進数	1'b0 ~ 1'b1	1'b0	SR をアサートした場合の Q4 出力の値を指定

## VHDL 記述 (インスタンスエーション)

```

-- ISERDESE1: (Input Serial/DESerializer)
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

ISERDESE1_inst : ISERDESE1
generic map (
    DATA_RATE => "DDR",           -- Single Data Rate or Double Data Rate operation.
    DATA_WIDTH => 4,               -- Parallel data width selection.
    DYN_CLKDIV_INV_EN => FALSE,     -- Enables dynamic CLKDIV inversion for Memory interfaces when TRUE.
    DYN_CLK_INV_EN => FALSE,       -- Enables dynamic CLK inversion for Memory interfaces when TRUE.
    INIT_Q1 => "0",                 -- Defines the initial value of Q1 output.
    INIT_Q2 => "0",                 -- Defines the initial value of Q2 output.
    INIT_Q3 => "0",                 -- Defines the initial value of Q3 output.
    INIT_Q4 => "0",                 -- Defines the initial value of Q4 output.
    INTERFACE_TYPE => "MEMORY",    -- Memory or Networking interface type.
    IOBDelay => "NONE",
    NUM_CE => 2,                    -- Specifies the number of clock enables used for the ISERDES_NODELAY.
    OFB_USED => FALSE,              -- Will select the sneak path from the OLOGIC, OSERDES.
    SERDES_MODE => "MASTER",       -- Specify whether the ISERDES is operating in master or slave modes when
                                   -- cascaded width expansion.
    SRVAL_Q1 => "0",                -- Defines the value of Q1 output when the SR is invoked.
    SRVAL_Q2 => "0",                -- Defines the value of Q2 output when the SR is invoked.
    SRVAL_Q3 => "0",                -- Defines the value of Q3 output when the SR is invoked.
    SRVAL_Q4 => "0"                -- Defines the value of Q4 output when the SR is invoked.
)

```

```

)
port map (
    O => O, -- 1-bit Combinatorial output.
    -- Q1 - Q6: 1-bit (each) Registered parallelized input data.
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    Q5 => Q5,
    Q6 => Q6,
    -- SHIFTOUT1/SHIFTOUT2: 1-bit (each) If ISERDES_MODE="MASTER" and two ISERDES_NODELAY are to be cascaded,
    -- connect to the slave ISERDES_NODELAY IDATASHIFTIN1/2 inputs.
    SHIFTOUT1 => SHIFTOUT1,
    SHIFTOUT2 => SHIFTOUT2,
    BITSLIP => BITSLIP, -- 1-bit Input data bitslip function enable.
    CE1 => CE1, -- 1-bit Input data register clock enables.
    CE2 => CE2, -- 1-bit DITTO
    CLK => CLK, -- 1-bit Primary clock input pin used.
    CLKB => CLKB, -- 1-bit Secondary clock input. If using in single clock DDR mode
    -- (DATA_RATE="DDR"), invert the clock connected to the CLK pin and connect
    -- to the CLKB pin. If using in dual clock mode DDR mode, connect a unique,
    -- phase shifted clock to the CLKB pin. If using in single data-rate mode
    -- (DATA_RATE="SDR"), leave this pin unconnected or connect to ground.

    CLKDIV => CLKDIV, -- 1-bit Divided clock to be used for parallelized data.
    D => D, -- 1-bit Input data to be connected directly to the top-level input or I/O
    -- port of the design or to an IODELAY component if additional input delay
    -- control is desired.

    DDLY => DDLY,
    DYNCLKDIVSEL => DYNCLKDIVSEL, -- 1-bit Dynamically select CLKDIV or CLKDIV_B (via an optional inversion).
    DYNCLKSEL => DYNCLKSEL, -- 1-bit Dynamically select CLK or CLK_B (via an optional inversion).
    OCLK => OCLK, -- 1-bit High speed output clock typically used for memory interfaces.
    OFB => OFB, -- 1-bit Feedback Path from the OLOGIC/OSERDES Output (w/ or w/o ODELAY).
    RST => RST, -- 1-bit Active high asynchronous reset signal for the registers of the
    -- SERDES.

    -- SHIFTIN1/SHIFTIN2: 1-bit (each) If ISERDES_MODE="SLAVE" connect to the master ISERDES_NODELAY
    -- IDATASHIFTOUT1/2 outputs. This pin must be grounded.
    SHIFTIN1 => SHIFTIN1,
    SHIFTIN2 => SHIFTIN2
);

-- End of ISERDESE1_inst instantiation

```

## Verilog 記述 (インスタンス化)

```
// ISERDESE1: (Input Serial/DESerializer)
//          Virtex-6
// Xilinx HDL Language Template, version 11.1

ISERDESE1 #(
    .DATA_RATE("DDR"),           // Single Data Rate or Double Data Rate operation.
    .DATA_WIDTH(4),              // Parallel data width selection.
    .DYN_CLKDIV_INV_EN("FALSE"), // Enables dynamic CLKDIV inversion for Memory interfaces when TRUE.
    .DYN_CLK_INV_EN("FALSE"),    // Enables dynamic CLK inversion for Memory interfaces when TRUE.
    .INIT_Q1(1'b0),              // Defines the initial value of Q1 output.
    .INIT_Q2(1'b0),              // Defines the initial value of Q2 output.
    .INIT_Q3(1'b0),              // Defines the initial value of Q3 output.
    .INIT_Q4(1'b0),              // Defines the initial value of Q4 output.
    .INTERFACE_TYPE("MEMORY"),   // Memory or Networking interface type.
    .IOBDELAY("NONE"),           // Specifies the number of clock enables used for the ISERDES_NODELAY.
    .NUM_CE(2),                  // Will select the sneak path from the OLOGIC, OSERDES.
    .OFB_USED("FALSE"),          // Specify whether the ISERDES is operating in master or slave modes when
    .SERDES_MODE("MASTER"),      // cascaded width expansion.
    .SRVAL_Q1(1'b0),             // Defines the value of Q1 output when the SR is invoked.
    .SRVAL_Q2(1'b0),             // Defines the value of Q2 output when the SR is invoked.
    .SRVAL_Q3(1'b0),             // Defines the value of Q3 output when the SR is invoked.
    .SRVAL_Q4(1'b0)             // Defines the value of Q4 output when the SR is invoked.
)
ISERDESE1_inst (
    .O(0),                      // 1-bit Combinatorial output.
    // Q1 - Q6: 1-bit (each) Registered parallelized input data.
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
    // SHIFTOUT1/SHIFTOUT2: 1-bit (each) If ISERDES_MODE="MASTER" and two ISERDES_NODELAY are to be cascaded,
    // connect to the slave ISERDES_NODELAY IDATASHIFTIN1/2 inputs.
    .SHIFTOUT1(SHIFTOUT1),
    .SHIFTOUT2(SHIFTOUT2),
    .BITSLIP(BITSLIP),          // 1-bit Input data bitslip function enable.
    .CE1(CE1),                  // 1-bit Input data register clock enables.
    .CE2(CE2),                  // 1-bit DITTO
    .CLK(CLK),                  // 1-bit Primary clock input pin used.
    .CLKB(CLKB),                // 1-bit Secondary clock input. If using in single clock DDR mode
    // (DATA_RATE="DDR"), invert the clock connected to the CLK pin and connect
    // to the CLKB pin. If using in dual clock mode DDR mode, connect a unique,
    // phase shifted clock to the CLKB pin. If using in single data-rate mode
    // (DATA_RATE="SDR"), leave this pin unconnected or connect to ground.

    .CLKDIV(CLKDIV),            // 1-bit Divided clock to be used for parallelized data.
    .D(D),                      // 1-bit Input data to be connected directly to the top-level input or I/O
    // port of the design or to an IODELAY component if additional input delay
    // control is desired.

    .DDL(DDL),
    .DYNCLKDIVSEL(DYNCLKDIVSEL), // 1-bit Dynamically select CLKDIV or CLKDIV_B (via an optional inversion).
    .DYNCLKSEL(DYNCLKSEL),       // 1-bit Dynamically select CLK or CLK_B (via an optional inversion).
    .OCLK(OCLK),                 // 1-bit High speed output clock typically used for memory interfaces.
    .OFB(OFB),                   // 1-bit Feedback Path from the OLOGIC/OSERDES Output (w/ or w/o ODELAY).
    .RST(RST),                   // 1-bit Active high asynchronous reset signal for the registers of the
    // SERDES.

    // SHIFTIN1/SHIFTIN2: 1-bit (each) If ISERDES_MODE="SLAVE" connect to the master ISERDES_NODELAY
    // IDATASHIFTOUT1/2 outputs. This pin must be grounded.
    .SHIFTIN1(SHIFTIN1),
    .SHIFTIN2(SHIFTIN2)
);

// End of ISERDESE1_inst instantiation
```



## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## KEEPER

プリミティブ : KEEPER Symbol



## 概要

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

## ポートの説明

属性	方向	幅	機能
O	出力	1 ビット	キーパ出力

## デザインの入力方法

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

## VHDL 記述 (インスタンス化)

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

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

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

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

-- End of KEEPER_inst instantiation
```

## Verilog 記述 (インスタンス化)

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

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

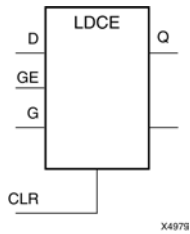
// End of KEEPER_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LDCE

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



### 概要

このデザイン エLEMENTは、非同期クリア (CLR) およびゲートイネーブル (GE) 付き透過データラッチです。非同期クリア入力 (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。ゲート入力 (G) およびゲートイネーブル (GE) が High で、CLR が Low のとき、Q にはデータ入力 (D) が使用されます。GE が Low の場合、D の値は不定値になります。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	GE	G	D	Q
1	X	X	X	0
0	0	X	X	変化なし
0	1	1	D	D
0	1	0	X	変化なし
0	1	↓	D	D

### デザインの入力方法

このELEMENTは、回路図でのみ使用できます。

### 使用可能な属性

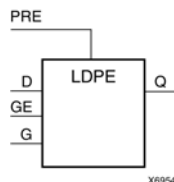
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後の Q 出力の初期値を指定

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LDPE

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



### 概要

このデザイン エLEMENTは、非同期プリセット (PRE) およびゲート イネーブル (GE) 付き透過データ ラッチです。PRE が High になると、ほかの入力は無視され、データ出力 (Q) が High にプリセットされます。ゲート入力 (G) およびゲート イネーブル (GE) が High の場合、Q にデータ入力 (D) の値が出力されます。D 入力の値は、G が High から Low に切り替わる時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。

電力が供給されると、ラッチは非同期にプリセットされ、出力が High 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
PRE	GE	G	D	Q
1	X	X	X	1
0	0	X	X	変化なし
0	1	1	D	D
0	1	0	X	変化なし
0	1	↓	D	D

### デザインの入力方法

このELEMENTは、回路図でのみ使用できます。

### 使用可能な属性

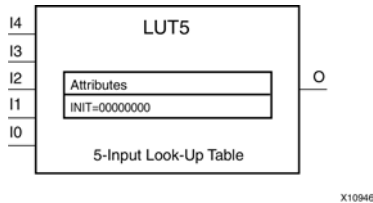
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	1	電源投入時または Q ポートに対する GSR のアサート時の初期値を指定

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## LUT5

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



### 概要

このデザイン エレメントは、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"FFFFFFFFFFFFFFFE") 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
);

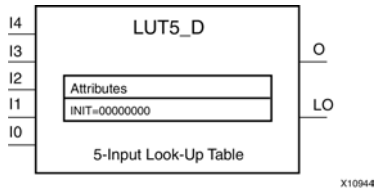
// End of LUT5_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 エレメントは、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(0), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4) // LUT input
);

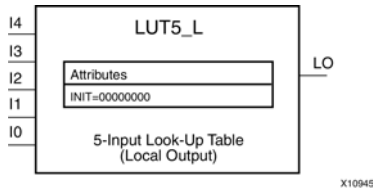
// End of LUT5_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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(O), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4) // LUT input
);

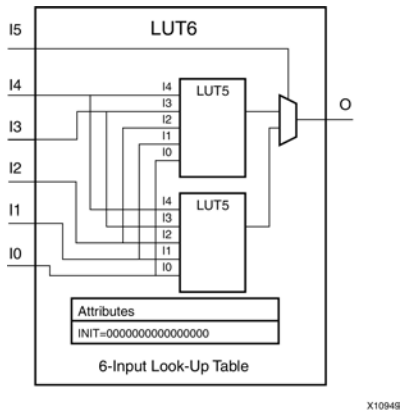
// End of LUT5_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

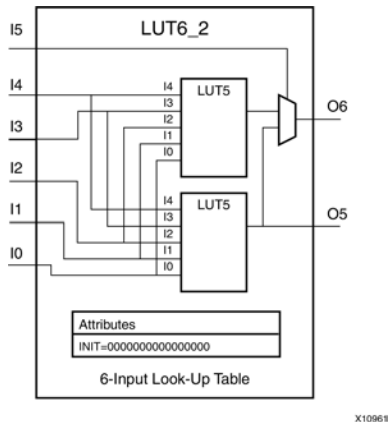
// End of LUT6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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;

-- 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 (インスタンスエーション)

```
// 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)
);

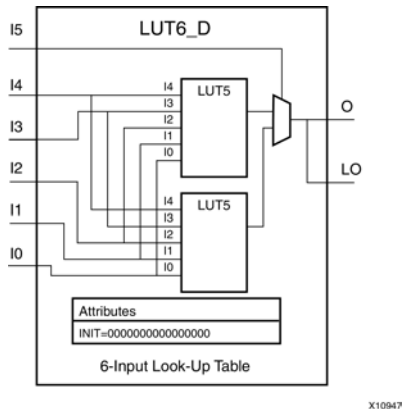
// End of LUT6_2_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

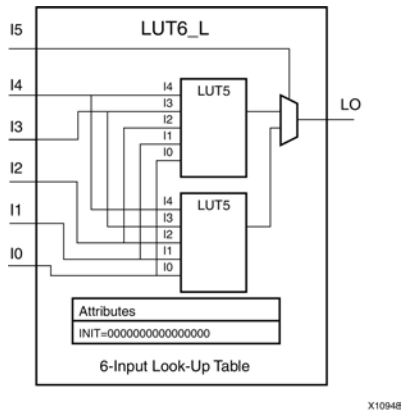
// End of LUT6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

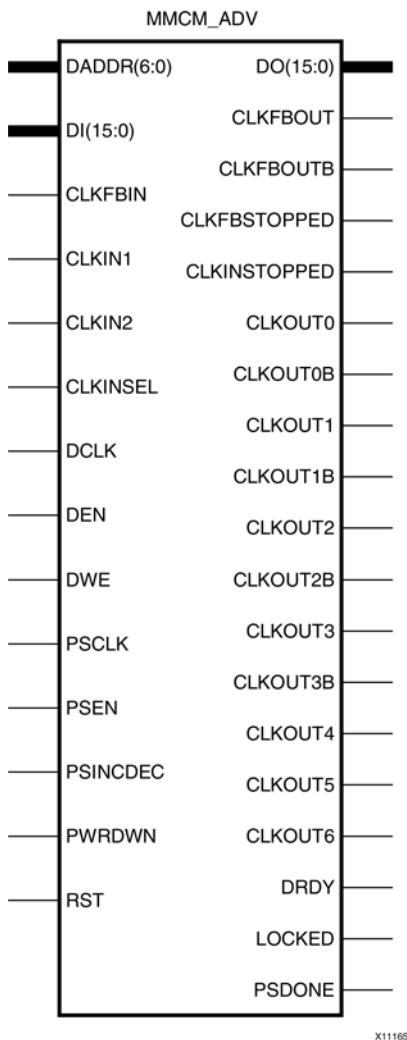
// End of LUT6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MMCM\_ADV

**プリミティブ :** MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.



### 概要

このコンポーネントは混合信号ブロックで、クロック ネットワーク デスキュー、周波数合成、ジッタ低減をサポートするために設計されています。7 つの O カウンタを独立してプログラムすることができます。つまり O0 は 2、O1 は 3 で分周するようプログラムできます。ただし、1 つの VCO ですべてのカウンタを駆動するため、VCO 動作周波数をすべての出力カウンタと同じにする必要があります。CLKFBOUT および CLKFBOUTB はロジックを駆動するのに使用できますが、DIVCLK\_DIVIDE の値で分周される CLKin 周波数と同じにする必要があります。

### ポートの説明

ポート名	タイプ	幅	機能
CLKFBIN	入力	1	クロック フィードバック入力
CLKFBOUT	出力	1	専用 MMCM フィードバック出力
CLKFBOUTB	出力	1	反転した MMCM フィードバック クロック出力

ポート名	タイプ	幅	機能
CLKFBSTOPPED	出力	1	フィードバック クロックが停止したことを示すステータス ピン
CLKINSEL	入力	1	クロック入力マルチプレクサのステートを制御する信号で、High = CLKIN1、Low = CLKIN2 です。MMCM リファレンス クロックをダイナミックに切り替えます。
CLKINSTOPPED	出力	1	入力クロックが停止したことを示すステータス ピン
CLKIN1	入力	1	汎用クロック入力
CLKIN2	入力	1	MMCM リファレンス クロックのセカンダリ クロック入力
CLKOUT[0:6]	出力	7、1 ビット	ユーザー コンフィギュラブル クロック出力 (0 ~ 6) で、1 (バイパスされたもの) から 128 までの VCO 位相出力 (ユーザー制御可能) を除算したものです。出力クロックは、位相シフトしない限り相互に位相アラインし、正しいフィードバック コンフィギュレーションの入力クロックにアラインしています。
CLKOUT[0:3]B	出力	4、1 ビット	CLKOUT[0:3] を反転したものです。
DADDR[6:0]	入力	7	ダイナミック リコンフィギュレーション アドレス (DADDR) 入力バスは、ダイナミック リコンフィギュレーション用のリコンフィギュレーション 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 クロックです。
DEN	入力	1	ダイナミック リコンフィギュレーション イネーブル (DEN) は、ダイナミック リコンフィギュレーション機能へのアクセスを提供します。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。
DI[15:0]	入力	16	ダイナミック リコンフィギュレーション データ入力 (DI) バスはリコンフィギュレーション データを提供します。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。
DO[15:0]	出力	16	ダイナミック リコンフィギュレーション出力バスは、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に MMCM データ出力を提供します。
DRDY	出力	1	ダイナミック リコンフィギュレーション準備完了 (DRDY) 出力は、MMCM のダイナミック リコンフィギュレーション機能用の DEN 信号への応答を提供します。
DWE	入力	1	ダイナミック リコンフィギュレーション ライト イネーブル (DWE) 入力ピンは、DADDR アドレスへの DI データの書き込みを制御するライト イネーブル信号を提供します。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	MMCM からの出力で、MMCM で、定義されている時間枠内で位相アライメントが完了したこと、PPM 範囲内で周波数が一致したことを示します。MMCM は、電源投入時に自動的にロックします。余分なリセットは不要です。入力クロックが停止した場合、または位相アライメントに違反が起きた場合 (入力クロック位相シフトなど) に、LOCKED がディアサートされます。MMCM は LOCKED がディアサートされた後に自動的にロックします。
PSCLK	入力	1	位相シフト クロック
PSDONE	出力	1	位相シフト終了
PSEN	入力	1	位相シフト イネーブル
PSINCDEC	入力	1	位相シフト インクリメント/デクリメント制御
PWRDWN	入力	1	パワー ダウンがインスタンス化されますが、MMCM は未使用です。

ポート名	タイプ	幅	機能
RST	入力	1	非同期リセット信号。RST 信号は、MMCM の非同期信号です。この信号が解放されると、MMCM は同期して再イネーブルになります。入力クロックのコンディション (周波数など) が変わる場合、リセットは不要です。

## 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
BANDWIDTH	文字列	OPTIMIZED、HIGH、LOW	OPTIMIZED	ジッタ、位相マージンなどの MMCM 特性に影響する MMCM プログラム アルゴリズムを指定します。
CLKFBOUT_MULT_F	3 上位ビット 浮動小数点	1.000 ~ 64.000	1.000	別の周波数を使用する場合に、すべての CLKOUT クロック出力を通倍する値を指定します。この値、CLKOUT#_DIVIDE 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKFBOUT_USE_FINE_PS	ブール代数	FALSE、TRUE	FALSE	CLKFBOUT カウンタ ファイン可変位相シフトイネーブル。CLKFBOUT_MULT は整数である必要があり、小数の除算はできません。
CLKIN1_PERIOD	フロート (nS)	1.000 ~ 1000.000	0.000	MMCM CLKIN1 入力の入力周期を指定します。ps まで指定できます。この値は必ず設定する必要があります。
CLKIN2_PERIOD	フロート (nS)	1.000 ~ 1000.000	0.000	MMCM CLKIN2 入力の入力周期を指定します。ps まで指定できます。この値は必ず設定する必要があります。
CLKOUT0_DIVIDE_F	3 上位ビット 浮動小数点	1.000 ~ 128.000	1.000	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、CLKFBOUT_MULT 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT[0:6]_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ~ 0.999	0.500	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。
CLKOUT[0:6]_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。

属性	タイプ	値	デフォルト	説明
CLKFBOUT_USE_FINE_PS	ブール代数	FALSE、TRUE	FALSE	CLKOUT0 カウンタ ファイン可変位相シフトイネーブル。CLKOUT0_DIVIDE は整数である必要があります、小数の除算はできません。
CLKOUT[1:6]_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、CLKFBOUT_MULT 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLOCK_HOLD	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、VCO 周波数を CLKIN を損失する前の近似値に保持します。
DIVCLK_DIVIDE	整数	1 ~ 128	1	入力クロックに対するすべての出力クロックの分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER2	3 上位ビット 浮動小数点	0.000 ~ 0.999	0.010	リファレンス クロック 2 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 です。このパラメータは小数 3 桁をサポートします。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、DONE が High になる前にロックするため、この属性が設定されている MMCM を待ちます。

## VHDL 記述 (インスタンスエーション)

```
-- MMCM_ADV: MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.
--           Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

MMCM_ADV_inst : MMCM_ADV
generic map (
    BANDWIDTH => "OPTIMIZED",
    CLKFBOUT_MULT_F => 1.0,
    CLKFBOUT_PHASE => 0.0,
    CLKFBOUT_USE_FINE_PS => FALSE,
    CLKIN1_PERIOD => 0.0,
    CLKIN2_PERIOD => 0.0,
    CLKOUT0_DIVIDE_F => 1.0,
    CLKOUT0_DUTY_CYCLE => 0.5,
    CLKOUT0_PHASE => 0.0,
    CLKOUT0_USE_FINE_PS => FALSE,
    CLKOUT1_DIVIDE => 1,
    CLKOUT1_DUTY_CYCLE => 0.5,
    CLKOUT1_PHASE => 0.0,
    CLKOUT1_USE_FINE_PS => FALSE,
    -- PLL bandwidth control, sets the MMCM bandwidth. Selecting HIGH sets
    -- the MMCM programming algorithm to the highest setting possible while
    -- maintaining good MMCM performance (i.e. low jitter, low jitter
    -- peaking, maximum phase margin). Selecting LOW sets the MMCM algorithm
    -- to the lowest setting possible while maintaining good MMCM
    -- performance. Selecting Optimized allows the MMCM programming
    -- algorithm to configure the MMCM in the best performance mode without
    -- trying to reach any particular bandwidth.
    -- Counter multiply value, Now supports non-integer values
    -- Only supported when CLKFBOUT_MULT is an integer value
    -- Counter fine variable phase shift enable, only supported when
    -- CLKFBOUT_MULT is an integer value
    -- The reference clock frequency is required for properly configuring
    -- the LOCK detect circuit and checking to make sure the VCO is
    -- operating within the allowed range. If no value is specified, a
    -- warning should be issued stating it was not provided so no error
    -- checking will be done.
    -- The reference clock frequency is required for properly configuring
    -- the LOCK detect circuit and checking to make sure the VCO is
    -- operating within the allowed range. If no value is specified, a
    -- warning should be issued stating it was not provided so no error
    -- checking will be done.
    -- Counter divide value, Now supports non-integer values but you lose
    -- CLKOUT5
    -- Counter output duty cycle
    -- Counter output phase relative to the output clock, only supported
    -- when CLKOUT0_DIVIDE is an integer value.
    -- Counter fine variable phase shift enable.
    -- Counter divide value
    -- Counter output duty cycle
    -- Counter output phase relative to the output clock
    -- Counter fine variable phase shift enable.
```

```

CLKOUT2_DIVIDE => 1,          -- Counter divide value
CLKOUT2_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT2_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT2_USE_FINE_PS => FALSE, -- Counter fine variable phase shift enable.
CLKOUT3_DIVIDE => 1,          -- Counter divide value
CLKOUT3_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT3_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT3_USE_FINE_PS => FALSE, -- Counter fine variable phase shift enable.
CLKOUT4_CASCADE => FALSE,
CLKOUT4_DIVIDE => 1,          -- Counter divide value
CLKOUT4_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT4_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT4_USE_FINE_PS => FALSE, -- Counter fine variable phase shift enable.
CLKOUT5_DIVIDE => 1,          -- Counter divide value
CLKOUT5_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT5_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT5_USE_FINE_PS => FALSE, -- Counter fine variable phase shift enable.
CLKOUT6_DIVIDE => 1,          -- Counter divide value
CLKOUT6_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT6_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT6_USE_FINE_PS => FALSE, -- Counter fine variable phase shift enable.
CLOCK_HOLD => FALSE,
COMPENSATION => "ZHOLD",      -- Defines how the PLL feedback is configured. SYSTEM_SYNCHRONOUS
                                -- indicates the MMCM is configured to provide a negative hold time
                                -- SOURCE_SYNCHRONOUS indicates the MMCM is configured to provide a zero
                                -- hold time SYSTEM_SYNCHRONOUS also does the AUTO detect of the
                                -- feedback and it indicates the MMCM is configured based on how a
                                -- customer has connected it. This can be any of the following: INTERNAL
                                -- indicates the MMCM is using its own internal feedback path so no
                                -- delay is being compensated. EXTERNAL indicates a network external to
                                -- the FPGA is being compensated. HROWCLK indicates the path between the
                                -- MMCM and the HROWCLK is being compensated CASCADE indicates cascading
                                -- of 2 MMCM's.
DIVCLK_DIVIDE => 1,          -- Counter divide value, always configured for 50% duty cycle
REF_JITTER1 => 0.0,         -- The reference clock2 jitter is specified in terms of the UI which is
                                -- a percentage of the reference clock. The number provided should be
                                -- the maximum peak to peak value our part can expect on the input
                                -- clock. Default is 0.1UI. This parameter should support out to three
                                -- decimal places.
REF_JITTER2 => 0.0,         -- The reference clock1 jitter is specified in terms of the UI which is
                                -- a percentage of the reference clock. The number provided should be
                                -- the maximum peak to peak value our part can expect on the input
                                -- clock. Default is 0.1UI. This parameter should support out to three
                                -- decimal places.
STARTUP_WAIT => FALSE      -- When TRUE, the MMCM locks before DONE goes high
)
port map (
  CLKFBOUT => CLKFBOUT,      -- 1-bit MMCM Feedback clock output
  CLKFBOUTB => CLKFBOUTB,    -- 1-bit Inverted MMCM feedback clock output
  CLKFBSTOPPED => CLKFBSTOPPED, -- 1-bit Status pin indicating that the feedback clock has stopped.
  CLKINSTOPPED => CLKINSTOPPED, -- 1-bit Status pin indicating that the input clock has stopped
  CLKOUT0 => CLKOUT0,        -- 1-bit MMCM clock output 0
  CLKOUT0B => CLKOUT0B,      -- 1-bit Inverted MMCM clock output 0
  CLKOUT1 => CLKOUT1,        -- 1-bit MMCM clock output 1
  CLKOUT1B => CLKOUT1B,      -- 1-bit Inverted MMCM clock output 1
  CLKOUT2 => CLKOUT2,        -- 1-bit MMCM clock output 2
  CLKOUT2B => CLKOUT2B,      -- 1-bit Inverted MMCM clock output 2
  CLKOUT3 => CLKOUT3,        -- 1-bit MMCM clock output 3
  CLKOUT3B => CLKOUT3B,      -- 1-bit Inverted MMCM clock output 3
  CLKOUT4 => CLKOUT4,        -- 1-bit MMCM clock output 4
  CLKOUT5 => CLKOUT5,        -- 1-bit MMCM clock output 5, not used if CLKOUT0 is not an integer
  CLKOUT6 => CLKOUT6,        -- 1-bit MMCM clock output 6, not used if CLKFBOUT_MULT is not an integer
  DO => DO,                  -- 16-bit DRP output signals
  DRDY => DRDY,              -- 1-bit DRP ready signal
  LOCKED => LOCKED,          -- 1-bit MMC locked signal
  PSDONE => PSDONE,          -- 1-bit Phase shift done
  CLKFBIN => CLKFBIN,        -- 1-bit Feedback clock pin to the MMCM
  CLKIN1 => CLKIN1,          -- 1-bit Reference clock pin 1 to the MMCM
  CLKIN2 => CLKIN2,          -- 1-bit Reference clock pin 2 to the MMCM
  CLKINSEL => CLKINSEL,      -- 1-bit Clock select control pin to the MMCM; 1 = CLKIN1; 0=CLKIN2
  DADDR => DADDR,            -- 7-bit DRP address control signals
  DCLK => DCLK,              -- 1-bit DRP clock signal (connects to interconnect clock)
  DEN => DEN,                -- 1-bit DRP enable signal

```

```

DI => DI,                -- 16-bit DRP data inputs
DWE => DWE,              -- 1-bit DRP write enable
PSCLK => PSCLK,          -- 1-bit Phase shift clock
PSEN => PSEN,            -- 1-bit Phase shift enable
PSINCDEC => PSINCDEC,    -- 1-bit Phase shift Increment/Decrement control
PWRDWN => PWRDWN,        -- 1-bit Power down
RST => RST               -- 1-bit MMCM global reset pin
);

-- End of MMCM_ADV_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// MMCM_ADV: MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.
//          Virtex-6
// Xilinx HDL Language Template, version 11.1

MMCM_ADV #(
    .BANDWIDTH("OPTIMIZED"), // PLL bandwidth control, sets the MMCM bandwidth. Selecting HIGH sets
                              // the MMCM programming algorithm to the highest setting possible while
                              // maintaining good MMCM performance (i.e. low jitter, low jitter
                              // peaking, maximum phase margin). Selecting LOW sets the MMCM algorithm
                              // to the lowest setting possible while maintaining good MMCM
                              // performance. Selecting Optimized allows the MMCM programming algorithm
                              // to configure the MMCM in the best performance mode without trying to
                              // reach any particular bandwidth.
    .CLKFBOUT_MULT_F(1.0),   // Counter multiply value, Now supports non-integer values
    .CLKFBOUT_PHASE(0.0),    // Only supported when CLKFBOUT_MULT is an integer value
    .CLKFBOUT_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable, only supported when
                              // CLKFBOUT_MULT is an integer value
    .CLKIN1_PERIOD(0.0),     // The reference clock frequency is required for properly configuring the
                              // LOCK detect circuit and checking to make sure the VCO is operating
                              // within the allowed range. If no value is specified, a warning should
                              // be issued stating it was not provided so no error checking will be
                              // done.
    .CLKIN2_PERIOD(0.0),     // The reference clock frequency is required for properly configuring the
                              // LOCK detect circuit and checking to make sure the VCO is operating
                              // within the allowed range. If no value is specified, a warning should
                              // be issued stating it was not provided so no error checking will be
                              // done.
    .CLKOUT0_DIVIDE_F(1.0),  // Counter divide value, Now supports non-integer values but you lose
                              // CLKOUT5
    .CLKOUT0_DUTY_CYCLE(0.5), // Counter output duty cycle
    .CLKOUT0_PHASE(0.0),      // Counter output phase relative to the output clock, only supported when
                              // CLKOUT0_DIVIDE is an integer value.
    .CLKOUT0_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLKOUT1_DIVIDE(1),        // Counter divide value
    .CLKOUT1_DUTY_CYCLE(0.5),  // Counter output duty cycle
    .CLKOUT1_PHASE(0.0),       // Counter output phase relative to the output clock
    .CLKOUT1_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLKOUT2_DIVIDE(1),        // Counter divide value
    .CLKOUT2_DUTY_CYCLE(0.5),  // Counter output duty cycle
    .CLKOUT2_PHASE(0.0),       // Counter output phase relative to the output clock
    .CLKOUT2_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLKOUT3_DIVIDE(1),        // Counter divide value
    .CLKOUT3_DUTY_CYCLE(0.5),  // Counter output duty cycle
    .CLKOUT3_PHASE(0.0),       // Counter output phase relative to the output clock
    .CLKOUT3_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLKOUT4_CASCADE("FALSE"),
    .CLKOUT4_DIVIDE(1),        // Counter divide value
    .CLKOUT4_DUTY_CYCLE(0.5),  // Counter output duty cycle
    .CLKOUT4_PHASE(0.0),       // Counter output phase relative to the output clock
    .CLKOUT4_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLKOUT5_DIVIDE(1),        // Counter divide value
    .CLKOUT5_DUTY_CYCLE(0.5),  // Counter output duty cycle
    .CLKOUT5_PHASE(0.0),       // Counter output phase relative to the output clock
    .CLKOUT5_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLKOUT6_DIVIDE(1),        // Counter divide value
    .CLKOUT6_DUTY_CYCLE(0.5),  // Counter output duty cycle
    .CLKOUT6_PHASE(0.0),       // Counter output phase relative to the output clock
    .CLKOUT6_USE_FINE_PS("FALSE"), // Counter fine variable phase shift enable.
    .CLOCK_HOLD("FALSE"),

```



```

.COMPENSATION("ZHOLD"), // Defines how the PLL feedback is configured. SYSTEM_SYNCHRONOUS
                          // indicates the MMCM is configured to provide a negative hold time
                          // SOURCE_SYNCHRONOUS indicates the MMCM is configured to provide a zero
                          // hold time SYSTEM_SYNCHRONOUS also does the AUTO detect of the feedback
                          // and it indicates the MMCM is configured based on how a customer has
                          // connected it. This can be any of the following: INTERNAL indicates the
                          // MMCM is using its own internal feedback path so no delay is being
                          // compensated. EXTERNAL indicates a network external to the FPGA is
                          // being compensated. HROWCLK indicates the path between the MMCM and the
                          // HROWCLK is being compensated CASCADE indicates cascading of 2 MMCM's.
.DIVCLK_DIVIDE(1), // Counter divide value, always configured for 50% duty cycle
.REF_JITTER1(0.0), // The reference clock2 jitter is specified in terms of the UI which is a
                  // percentage of the reference clock. The number provided should be the
                  // maximum peak to peak value our part can expect on the input clock.
                  // Default is 0.1UI. This parameter should support out to three decimal
                  // places.
.REF_JITTER2(0.0), // The reference clock1 jitter is specified in terms of the UI which is a
                  // percentage of the reference clock. The number provided should be the
                  // maximum peak to peak value our part can expect on the input clock.
                  // Default is 0.1UI. This parameter should support out to three decimal
                  // places.
.STARTUP_WAIT("FALSE") // When TRUE, the MMCM locks before DONE goes high
)
MMCM_ADV_inst (
  .CLKFBOUT(CLKFBOUT), // 1-bit MMCM Feedback clock output
  .CLKFBOUTB(CLKFBOUTB), // 1-bit Inverted MMCM feedback clock output
  .CLKFBSTOPPED(CLKFBSTOPPED), // 1-bit Status pin indicating that the feedback clock has stopped.
  .CLKINSTOPPED(CLKINSTOPPED), // 1-bit Status pin indicating that the input clock has stopped
  .CLKOUT0(CLKOUT0), // 1-bit MMCM clock output 0
  .CLKOUT0B(CLKOUT0B), // 1-bit Inverted MMCM clock output 0
  .CLKOUT1(CLKOUT1), // 1-bit MMCM clock output 1
  .CLKOUT1B(CLKOUT1B), // 1-bit Inverted MMCM clock output 1
  .CLKOUT2(CLKOUT2), // 1-bit MMCM clock output 2
  .CLKOUT2B(CLKOUT2B), // 1-bit Inverted MMCM clock output 2
  .CLKOUT3(CLKOUT3), // 1-bit MMCM clock output 3
  .CLKOUT3B(CLKOUT3B), // 1-bit Inverted MMCM clock output 3
  .CLKOUT4(CLKOUT4), // 1-bit MMCM clock output 4
  .CLKOUT5(CLKOUT5), // 1-bit MMCM clock output 5, not used if CLKOUT0 is not an integer
  .CLKOUT6(CLKOUT6), // 1-bit MMCM clock output 6, not used if CLKFBOUT_MULT is not an integer
  .DO(DO), // 16-bit DRP output signals
  .DRDY(DRDY), // 1-bit DRP ready signal
  .LOCKED(LOCKED), // 1-bit MMC locked signal
  .PSDONE(PSDONE), // 1-bit Phase shift done
  .CLKFBIN(CLKFBIN), // 1-bit Feedback clock pin to the MMCM
  .CLKIN1(CLKIN1), // 1-bit Reference clock pin 1 to the MMCM
  .CLKIN2(CLKIN2), // 1-bit Reference clock pin 2 to the MMCM
  .CLKINSEL(CLKINSEL), // 1-bit Clock select control pin to the MMCM; 1 = CLKIN1; 0=CLKIN2
  .DADDR(DADDR), // 7-bit DRP address control signals
  .DCLK(DCLK), // 1-bit DRP clock signal (connects to interconnect clock)
  .DEN(DEN), // 1-bit DRP enable signal
  .DI(DI), // 16-bit DRP data inputs
  .DWE(DWE), // 1-bit DRP write enable
  .PSCLK(PSCLK), // 1-bit Phase shift clock
  .PSEN(PSEN), // 1-bit Phase shift enable
  .PSINCDEC(PSINCDEC), // 1-bit Phase shift Increment/Decrement control
  .PWRDWN(PWRDWN), // 1-bit Power down
  .RST(RST) // 1-bit MMCM global reset pin
);

// End of MMCM_ADV_inst instantiation

```

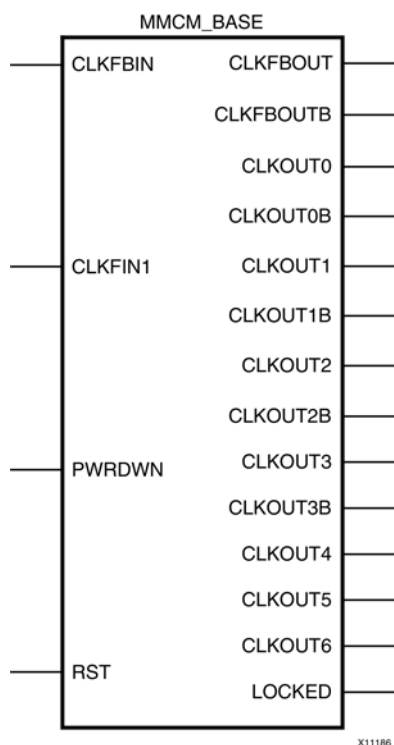
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## MMCM\_BASE

**コンビニエンス プリミティブ** : Mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.



### 概要

このコンポーネントは混合信号ブロックで、クロック ネットワーク デスキュー、周波数合成、ジッタ低減をサポートするために設計されています。7 つの O カウンタを独立してプログラムすることができます。つまり O0 は 2、O1 は 3 で分周するようプログラムできます。ただし、1 つの VCO ですべてのカウンタを駆動するため、VCO 動作周波数をすべての出力カウンタと同じにする必要があります。CLKFBOUT および CLKFBOUTB はロジックを駆動するのに使用できますが、CLKin 周波数と同じにする必要があります。

### ポートの説明

ポート名	タイプ	幅	機能
CLKFBIN	入力	1	クロック フィードバック入力
CLKFBOUT	出力	1	専用 MMCM フィードバック出力
CLKFBOUTB	出力	1	反転した MMCM フィードバック クロック出力
CLKIN1	入力	1	汎用クロック入力
CLKOUT[0:6]	出力	7、1 ビット	ユーザー コンフィギュラブル クロック出力 (0 ~ 6) で、1 (バイパスされたもの) から 128 までの VCO 位相出力 (ユーザー制御可能) を除算したものです。出力クロックは、位相シフトしない限り相互に位相アラインし、正しいフィードバック コンフィギュレーションの入力クロックにアラインしています。
CLKOUT[0:3]B	出力	4、1 ビット	CLKOUT[0:3] を反転したものです。

ポート名	タイプ	幅	機能
LOCKED	出力	1	MMCM からの出力で、MMCM で、定義されている時間枠内で位相アライメントが完了したこと、PPM 範囲内で周波数が一致したことを示します。MMCM は、電源投入時に自動的にロックします。余分なリセットは不要です。入力クロックが停止した場合、または位相アライメントに違反が起きた場合（入力クロック位相シフトなど）に、LOCKED がディASSERTされます。MMCM は LOCKED がディASSERTされた後に自動的にロックします。
PWRDWN	入力	1	パワー ダウンがインスタンスシートされますが、MMCM は未使用です。
RST	入力	1	非同期リセット信号。RST 信号は、MMCM の非同期信号です。この信号が解放されると、MMCM は同期して再イネーブルになります。入力クロックのコンディション（周波数など）が変わる場合、リセットは不要です。

## 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
BANDWIDTH	文字列	OPTIMIZED、HIGH、LOW	OPTIMIZED	ジッタ、位相マージンなどの MMCM 特性に影響する MMCM プログラム アルゴリズムを指定します。
CLKFBOUT_MULT_F	3 上位ビット 浮動小数点	1.000 ~ 64.000	1.000	別の周波数を使用する場合に、すべての CLKOUT クロック出力を通倍する値を指定します。この値、CLKOUT#_DIVIDE 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKFBOUT_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。フィードバック クロックをシフトすると、MMCM の出力クロックがすべて負の値に位相シフトします。
CLKIN1_PERIOD	フロート (nS)	1.000 ~ 1000.000	0.000	MMCM CLKIN1 入力の入力周期を指定します。ps まで指定できます。この値は必ず設定する必要があります。
CLKOUT0_DIVIDE_F	3 上位ビット 浮動小数点	1.000 ~ 128.000	1.000	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、CLKFBOUT_MULT 値および DIVCLK_DIVIDE 値により出力周波数が決まります。
CLKOUT[0:6]_DUTY_CYCLE	3 上位ビット 浮動小数点	0.001 ~ 0.999	0.500	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。
CLKOUT[0:6]_PHASE	3 上位ビット 浮動小数点	-360.000 ~ 360.000	0.000	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。

属性	タイプ	値	デフォルト	説明
CLOCK_HOLD	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、VCO 周波数を CLKIN を損失する前の近似値に保持します。
DIVCLK_DIVIDE	整数	1 ~ 128	1	入力クロックに対するすべての出力クロックの分周比を指定し、PFD に入力される CLKIN を分周します。
REF_JITTER1	3 上位ビット 浮動小数点	0.000 ~ 0.999	0.010	MMCM パフォーマンスを最適化するため、リファレンス クロックの期待ジッタ値を指定します。バンド幅設定が OPTIMIZED になっていると、値が既知のものでない場合は入力クロックに最適なパラメータが選択されます。値が既知のものである場合、値は、入力クロックの期待ジッタの UI パーセンテージ (最大ピークトゥピーク) により指定される必要があります。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	TRUE の場合、DONE が High になる前にロックするため、この属性が設定されている MMCM を待ちます。

## VHDL 記述 (インスタンスエーション)

```
-- MMCM_BASE: MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.
-- Virtex-6
-- Xilinx HDL Language Template, version 11.1

MMCM_BASE_inst : MMCM_BASE
generic map (
    BANDWIDTH => "OPTIMIZED", -- PLL bandwidth control, sets the MMCM bandwidth. Selecting HIGH sets the
                                -- MMCM programming algorithm to the highest setting possible while
                                -- maintaining good MMCM performance (i.e. low jitter, low jitter peaking,
                                -- maximum phase margin). Selecting LOW sets the MMCM algorithm to the lowest
                                -- setting possible while maintaining good MMCM performance. Selecting
                                -- Optimized allows the MMCM programming algorithm to configure the MMCM in
                                -- the best performance mode without trying to reach any particular
                                -- bandwidth.
    CLKFBOUT_MULT_F => 1.0, -- Counter multiply value, Now supports non-integer values
    CLKFBOUT_PHASE => 0.0, -- Only supported when CLKFBOUT_MULT is an integer value
    CLKIN1_PERIOD => 0.0, -- The reference clock frequency is required for properly configuring the
                                -- LOCK detect circuit and checking to make sure the VCO is operating within
                                -- the allowed range. If no value is specified, a warning should be issued
                                -- stating it was not provided so no error checking will be done.
    CLKOUT0_DIVIDE_F => 1.0, -- Counter divide value, Now supports non-integer values but you lose CLKOUT5
    CLKOUT0_DUTY_CYCLE => 0.5, -- Counter output duty cycle
    CLKOUT0_PHASE => 0.0, -- Counter output phase relative to the output clock, only supported when
                                -- CLKOUT0_DIVIDE is an integer value.
    CLKOUT1_DIVIDE => 1, -- Counter divide value
    CLKOUT1_DUTY_CYCLE => 0.5, -- Counter output duty cycle
    CLKOUT1_PHASE => 0.0, -- Counter output phase relative to the output clock
    CLKOUT2_DIVIDE => 1, -- Counter divide value
    CLKOUT2_DUTY_CYCLE => 0.5, -- Counter output duty cycle
    CLKOUT2_PHASE => 0.0, -- Counter output phase relative to the output clock
    CLKOUT3_DIVIDE => 1, -- Counter divide value
    CLKOUT3_DUTY_CYCLE => 0.5, -- Counter output duty cycle
    CLKOUT3_PHASE => 0.0, -- Counter output phase relative to the output clock
    CLKOUT4_CASCADE => FALSE,
```

```

CLKOUT4_DIVIDE => 1,          -- Counter divide value
CLKOUT4_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT4_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT5_DIVIDE => 1,          -- Counter divide value
CLKOUT5_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT5_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLKOUT6_DIVIDE => 1,          -- Counter divide value
CLKOUT6_DUTY_CYCLE => 0.5,    -- Counter output duty cycle
CLKOUT6_PHASE => 0.0,        -- Counter output phase relative to the output clock
CLOCK_HOLD => FALSE,
DIVCLK_DIVIDE => 1,           -- Counter divide value, always configured for 50% duty cycle
REF_JITTER1 => 0.0,          -- The reference clock2 jitter is specified in terms of the UI which is a
                                -- percentage of the reference clock. The number provided should be the
                                -- maximum peak to peak value our part can expect on the input clock. Default
                                -- is 0.1UI. This parameter should support out to three decimal places.
STARTUP_WAIT => FALSE        -- When TRUE, the MMCM locks before DONE goes high
)
port map (
  CLKFBOUT => CLKFBOUT,      -- 1-bit MMCM Feedback clock output
  CLKFBOUTB => CLKFBOUTB,    -- 1-bit Inverted MMCM feedback clock output
  CLKOUT0 => CLKOUT0,        -- 1-bit MMCM clock output 0
  CLKOUT0B => CLKOUT0B,      -- 1-bit Inverted MMCM clock output 0
  CLKOUT1 => CLKOUT1,        -- 1-bit MMCM clock output 1
  CLKOUT1B => CLKOUT1B,      -- 1-bit Inverted MMCM clock output 1
  CLKOUT2 => CLKOUT2,        -- 1-bit MMCM clock output 2
  CLKOUT2B => CLKOUT2B,      -- 1-bit Inverted MMCM clock output 2
  CLKOUT3 => CLKOUT3,        -- 1-bit MMCM clock output 3
  CLKOUT3B => CLKOUT3B,      -- 1-bit Inverted MMCM clock output 3
  CLKOUT4 => CLKOUT4,        -- 1-bit MMCM clock output 4
  CLKOUT5 => CLKOUT5,        -- 1-bit MMCM clock output 5, not used if CLKOUT0 is not an integer
  CLKOUT6 => CLKOUT6,        -- 1-bit MMCM clock output 6, not used if CLKFBOUT_MULT is not an integer
  LOCKED => LOCKED,          -- 1-bit MMC locked signal
  CLKFBIN => CLKFBIN,        -- 1-bit Feedback clock pin to the MMCM
  CLKIN1 => CLKIN1,          -- 1-bit Reference clock pin 1 to the MMCM
  PWRDWN => PWRDWN,          -- 1-bit Power down
  RST => RST                 -- 1-bit MMCM global reset pin
);

-- End of MMCM_BASE_inst instantiation

```

## Verilog 記述 (インスタンスエーション)

```

// MMCM_BASE: MMCM is a mixed signal block designed to support clock network deskew, frequency synthesis, and jitter reduction.
// Virtex-6
// Xilinx HDL Language Template, version 11.1

MMCM_BASE #(
  .BANDWIDTH("OPTIMIZED"),    // PLL bandwidth control, sets the MMCM bandwidth. Selecting HIGH sets the
                                // MMCM programming algorithm to the highest setting possible while
                                // maintaining good MMCM performance (i.e. low jitter, low jitter peaking,
                                // maximum phase margin). Selecting LOW sets the MMCM algorithm to the lowest
                                // setting possible while maintaining good MMCM performance. Selecting
                                // Optimized allows the MMCM programming algorithm to configure the MMCM in
                                // the best performance mode without trying to reach any particular bandwidth.
  .CLKFBOUT_MULT_F(1.0),      // Counter multiply value, Now supports non-integer values
  .CLKFBOUT_PHASE(0.0),       // Only supported when CLKFBOUT_MULT is an integer value
  .CLKIN1_PERIOD(0.0),        // The reference clock frequency is required for properly configuring the LOCK
                                // detect circuit and checking to make sure the VCO is operating within the
                                // allowed range. If no value is specified, a warning should be issued stating
                                // it was not provided so no error checking will be done.
  .CLKOUT0_DIVIDE_F(1.0),      // Counter divide value, Now supports non-integer values but you lose CLKOUT5
  .CLKOUT0_DUTY_CYCLE(0.5),   // Counter output duty cycle
  .CLKOUT0_PHASE(0.0),        // Counter output phase relative to the output clock, only supported when
                                // CLKOUT0_DIVIDE is an integer value.
  .CLKOUT1_DIVIDE(1),          // Counter divide value
  .CLKOUT1_DUTY_CYCLE(0.5),    // Counter output duty cycle
  .CLKOUT1_PHASE(0.0),         // Counter output phase relative to the output clock
  .CLKOUT2_DIVIDE(1),          // Counter divide value
  .CLKOUT2_DUTY_CYCLE(0.5),    // Counter output duty cycle
  .CLKOUT2_PHASE(0.0),         // Counter output phase relative to the output clock
  .CLKOUT3_DIVIDE(1),          // Counter divide value
  .CLKOUT3_DUTY_CYCLE(0.5),    // Counter output duty cycle

```

```

.CLKOUT3_PHASE(0.0),          // Counter output phase relative to the output clock
.CLKOUT4_CASCADE("FALSE"),
.CLKOUT4_DIVIDE(1),           // Counter divide value
.CLKOUT4_DUTY_CYCLE(0.5),     // Counter output duty cycle
.CLKOUT4_PHASE(0.0),          // Counter output phase relative to the output clock
.CLKOUT5_DIVIDE(1),           // Counter divide value
.CLKOUT5_DUTY_CYCLE(0.5),     // Counter output duty cycle
.CLKOUT5_PHASE(0.0),          // Counter output phase relative to the output clock
.CLKOUT6_DIVIDE(1),           // Counter divide value
.CLKOUT6_DUTY_CYCLE(0.5),     // Counter output duty cycle
.CLKOUT6_PHASE(0.0),          // Counter output phase relative to the output clock
.CLOCK_HOLD("FALSE"),
.DIVCLK_DIVIDE(1),            // Counter divide value, always configured for 50% duty cycle
.REF_JITTER1(0.0),            // The reference clock2 jitter is specified in terms of the UI which is a
                                // percentage of the reference clock. The number provided should be the
                                // maximum peak to peak value our part can expect on the input clock. Default
                                // is 0.1UI. This parameter should support out to three decimal places.
.STARTUP_WAIT("FALSE")       // When TRUE, the MMCM locks before DONE goes high
)
MMCM_BASE_inst (
.CLKFBOUT(CLKFBOUT),          // 1-bit MMCM Feedback clock output
.CLKFBOUTB(CLKFBOUTB),        // 1-bit Inverted MMCM feedback clock output
.CLKOUT0(CLKOUT0),            // 1-bit MMCM clock output 0
.CLKOUT0B(CLKOUT0B),          // 1-bit Inverted MMCM clock output 0
.CLKOUT1(CLKOUT1),            // 1-bit MMCM clock output 1
.CLKOUT1B(CLKOUT1B),          // 1-bit Inverted MMCM clock output 1
.CLKOUT2(CLKOUT2),            // 1-bit MMCM clock output 2
.CLKOUT2B(CLKOUT2B),          // 1-bit Inverted MMCM clock output 2
.CLKOUT3(CLKOUT3),            // 1-bit MMCM clock output 3
.CLKOUT3B(CLKOUT3B),          // 1-bit Inverted MMCM clock output 3
.CLKOUT4(CLKOUT4),            // 1-bit MMCM clock output 4
.CLKOUT5(CLKOUT5),            // 1-bit MMCM clock output 5, not used if CLKOUT0 is not an integer
.CLKOUT6(CLKOUT6),            // 1-bit MMCM clock output 6, not used if CLKFBOUT_MULT is not an integer
.LOCKED(LOCKED),              // 1-bit MMC locked signal
.CLKFBIN(CLKFBIN),            // 1-bit Feedback clock pin to the MMCM
.CLKIN1(CLKIN1),              // 1-bit Reference clock pin 1 to the MMCM
.PWRDWN(PWRDWN),              // 1-bit Power down
.RST(RST)                     // 1-bit MMCM global reset pin
);

// End of MMCM_BASE_inst instantiation

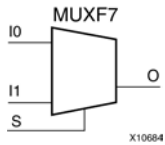
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MUXF7

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



### 概要

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

このほか、ローカル出力を持つ MUXF7\_D および MUXF7\_L があり、異なるタイミング モデルでレイアウト前のタイミング予測をより正確に行う必要がある場合に使用できます。

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

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

MUXF7_inst : MUXF7
port map (
    O => O,      -- Output of MUX to general routing
    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_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
    .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
);

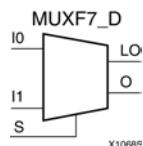
// End of MUXF7_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MUXF7\_D

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



### 概要

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

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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



## VHDL 記述 (インスタンスレーション)

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

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

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

MUXF7_D_inst : MUXF7_D
port map (
    LO => LO,  -- Ouptut of MUX to local routing
    O => O,    -- Output of MUX to general routing
    IO => IO,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    I1 => I1,  -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
    S => S     -- Input select to MUX
);

-- End of MUXF7_D_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

MUXF7_D MUXF7_D_inst (
    .LO(LO), // Ouptut of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .I1(I1), // Input (tie to MUXF6 LO out or LUT6 O6 pin)
    .S(S)    // Input select to MUX
);

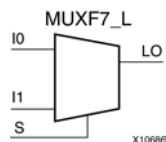
// End of MUXF7_D_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MUXF7\_L

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



### 概要

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

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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_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 (インスタンスレーション)

```
// 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
);

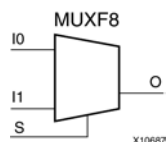
// End of MUXF7_L_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MUXF8

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



### 概要

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

## VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF8_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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

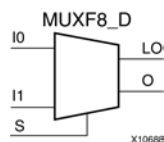
// End of MUXF8_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MUXF8\_D

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



### 概要

このデザイン エレメントは、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
);

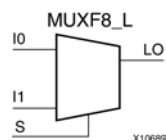
// End of MUXF8_D_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## MUXF8\_L

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



### 概要

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

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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



## VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF8_L_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

MUXF8_L MUXF8_L_inst (
    .LO(LO), // Output of MUX to local routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

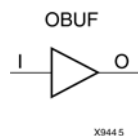
// End of MUXF8_L_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OBUF

プリミティブ : Output Buffer



### 概要

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

このエレメント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。このエレメントでは、LVTTL 規格が使用され、DRIVE 制約と SLOW または FAST 制約を使用して駆動電流とスルー レートを選択できます。デフォルトでは、DRIVE=12mA、スルー レートは SLOW に設定されています。

### ポートの説明

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

### デザインの入力方法

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

### 使用可能な属性

属性	タイプ	値	デフォルト	説明
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;

-- 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
);

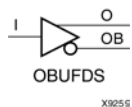
// End of OBUF_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OBUFDS

プリミティブ : Differential Signaling Output Buffer



### 概要

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

### 論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

### ポートの説明

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of OBUFDS_inst instantiation
```

## Verilog 記述 (インスタンスレーション)

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

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

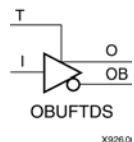
// End of OBUFDS_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OBUFTDS

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



### 概要

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

### 論理表

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

### ポートの説明

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

### デザインの入力方法

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

### 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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;

-- 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 (インスタンス化)

```
// 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
);

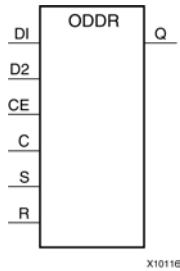
// End of OBUFTDS_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## ODDR

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



### 概要

このデザイン エLEMENT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ELEMENTは 2 つのモードで動作します。これらのモードは、DDR\_CLK\_EDGE 属性で設定します。

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

### ポートの説明

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

### デザインの入力方法

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



## 使用可能な属性

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

## VHDL 記述 (インスタンス化)

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;

-- 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 (インスタンス化)

```
// 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
);

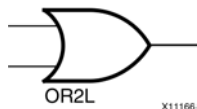
// End of ODDR_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## OR2L

プリミティブ : Two input OR gate implemented in place of a Slice Latch



### 概要

このデザイン エLEMENTでは、コンフィギュレーション可能なスライス ラッチで 2 入力 OR ゲートのファンクションが使用されます (論理表を参照)。このELEMENTを使用すると、ロジックのレジスタ/ラッチ リソース数をトレードオフにすることで、ロジック レベルを削減して、デバイスのロジック集積度を高めることができます。このELEMENTはレジスタのパックおよび集積度に影響を与えるので注意してください。AND2B1L または OR2L ELEMENTをスライスに指定すると、残りのレジスタおよびラッチが使用できなくなります。

### 論理表

入力		出力
DI	SRI	O
0	0	0
0	1	1
1	0	1
1	1	1

### ポートの説明

ポート名	タイプ	幅	機能
O	出力	1	OR ゲートの出力
DI	入力	1	同じスライスにあるソース LUT に通常接続されるアクティブ High の入力
SRI	入力	1	通常スライス外からソースされるアクティブ Low の入力  <b>メモ:</b> 複数の AND2B1L または OR2B1L を 1 つのスライスにパックするには、この入力に共通の信号を接続する必要があります。

### デザインの入力方法

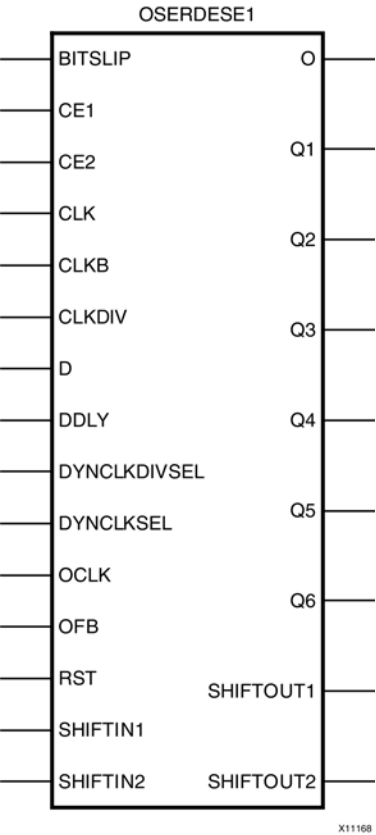
インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

### 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

# OSERDESE1

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、ソース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、ザイリンクス FPGA の 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
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します。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"><li>・ クロック領域内の 10 個のグローバル クロック ライン</li><li>・ 4 個のリージョナル クロック ライン</li><li>・ 4 個のクロック I/O (隣接したクロック領域内)</li><li>・ FPGA (バイパスを介す)</li></ul>
CLKB	入力	1	高速クロック入力

ポート名	タイプ	幅	機能
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します。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。 <ul style="list-style-type: none"> <li>・ クロック領域内の 10 個のグローバル クロック ライン</li> <li>・ 4 個のリージョナル クロック ライン</li> </ul>
CLKPERF	入力	1	MMCM からのパフォーマンス パスの入力
CLKPERFDELAY	入力	1	IODELAY からの BUFO の遅延バージョン
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。
IOCLKGLITCH	出力	1	高速クロックでグリッチが発生したことを示します。アクティブ High です。
OCBEXTEND	出力	1	DDR3 モードで使い、CLK を CLKPERF または CLKPERFDELAY に一致させるために 出力巡回バッファ (OCB) がレイテンシを拡大したことを示します。
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。
ODV	入力	1	DDR3 に使用。ODELAY 値がクロック周期の 180 度を越える場合、1 に設定されます。
OFB	出力	1	データ出力のフィードバック パス
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。
RST	入力	1	リセット
SHIFTIN1/ SHIFTIN2	入力	1	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。
SHIFTOUT1/ SHIFTOUT2	出力	1	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。
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。
TFB	出力	1	トライステート制御出力。トライステート パラレル/シリアル コンバータの出力を IODELAY の制御/トライステート入力に接続します。
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。
T1 ~ T4	入力	1	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。
WC	入力	1	DDR3 に使用。FIFO カウンタをリセットし、IDELAY から ODELAY へと IODELAY を切り替えます。

## 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
DATA_RATE_OQ	文字列	DDR、SDR	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します。
DATA_RATE_TQ	文字列	DDR、BUF、SDR	DDR	トライステート 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します。
DATA_WIDTH	整数	4、2、3、5、6、7、8、10	4	<ul style="list-style-type: none"> <li>DATA_RATE_OQ = DDR の場合、値は 4、6、8、または 10 になります。</li> <li>DATA_RATE_OQ = SDR の場合、値は 2、3、4、5、6、7、または 8 になります。</li> </ul>
DDR3_DATA	整数	1、0	1	DDR3 では、I/O が DQ または DQS ピンの場合は 1 に、制御、アドレス、クロックなどの場合は 0 に設定されます。
INIT_OQ	2 進数	1'b0 ~ 1'b1	1'b0	OQ 出力の初期値を指定
INIT_TQ	2 進数	1'b0 ~ 1'b1	1'b0	TQ 出力の初期値を指定
INTERFACE_TYPE	文字列	DEFAULT、MEMORY_DDR3	DEFAULT	DDR3 回路をバイパスします。
ODELAY_USED	整数	0、1	0	書き込みまたは BUFO アライメントのために BUFO は IODELAY を駆動します。
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	1'b0 ~ 1'b1	1'b0	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	1'b0 ~ 1'b1	1'b0	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	4、1	4	DATA_RATE_TQ = DDR、DATA_WIDTH = 4、および DATA_RATE_OQ = DDR の場合、値は 1 または 4 に制限されます。DATA_RATE_TQ、DATA_WIDTH、および DATA_RATE_OQ がそれ以外の値に設定されている場合、値は 1 になります。

## VHDL 記述 (インスタンス化)

```
-- OSERDESE1: Dedicated IOB Output Serializer
-- Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2
```

```
OSERDESE1_inst : OSERDESE1
generic map (
  DATA_RATE_OQ => "DDR",      -- Defines whether the data changes at every clock edge or every positive
                                -- clock edge with respect to CLK.
  DATA_RATE_TQ => "DDR",      -- Defines whether the 3-state changes at every clock edge, every positive
                                -- clock edge, or buffer configuration with respect to CLK.
  DATA_WIDTH => 4,            -- If DATA_RATE_OQ = DDR, value is limited to 4, 6, 8, or 10. If
                                -- DATA_RATE_OQ = SDR, value is limited to 2, 3, 4, 5, 6, 7, or 8.
```

```

DDR3_DATA => 1,          -- For DDR3, if the I/O is a DQ or DQS pin, set to 1. If control, address,
                           -- clock, etc. set to 0.
INIT_OQ => "0",          -- Defines the initial value of OQ output.
INIT_TQ => "0",          -- Defines the initial value of TQ output.
INTERFACE_TYPE => "DEFAULT", -- To bypass DDR3 circuitry
ODELAY_USED => 0,        -- BUFO drives IODELAY for write leveling or BUFO alignment.
SERDES_MODE => "MASTER", -- Defines whether the OSERDES module is a master or slave when width
                           -- expansion is used.
SRVAL_OQ => "0",         -- Defines the value of OQ output when reset is invoked.
SRVAL_TQ => "0",         -- Defines the value of TQ output when reset is invoked.
TRISTATE_WIDTH => 4      -- If DATA_RATE_TQ = DDR, DATA_WIDTH = 4, and DATA_RATE_OQ = DDR, value is
                           -- limited to 1 or 4. For all other settings of DATA_RATE_TQ, DATA_WIDTH,
                           -- and DATA_RATE_OQ, value is limited to 1.
)
port map (
  OCBEXTEND => OCBEXTEND,
  OFB => OFB,          -- 1-bit Feedback path for Data Output
  OQ => OQ,            -- 1-bit Data Path Output
  -- SHIFTOUT1/SHIFTOUT2: 1-bit (each) Carry Out for data input expansion. Connect to SHIFTIN1/2 of
  master.
  SHIFTOUT1 => SHIFTOUT1,
  SHIFTOUT2 => SHIFTOUT2,
  TFB => TFB,
  TQ => TQ,            -- 1-bit 3-State Path Output
  CLK => CLK,          -- 1-bit High Speed Clock Input - This clock input is used to drive the
                           -- parallel-to-serial converters. The possible source for the CLK port is
                           -- from one of the following clock resources: - Ten global clock lines in a
                           -- clock region - Four regional clock lines - Four clock capable I/Os
                           -- (within adjacent clock region) - Fabric (through bypass).

  CLKDIV => CLKDIV,     -- 1-bit Divided High Speed Clock Input
  CLKPERF => CLKPERF,   -- 1-bit Input for performance path from PLL.
  CLKPERFDELAY => CLKPERFDELAY, -- 1-bit Delayed version of BUFO from IODELAY
  -- D1 - D6: 1-bit (each) 1-bit parallel Data Input
  D1 => D1,
  D2 => D2,
  D3 => D3,
  D4 => D4,
  D5 => D5,
  D6 => D6,
  OCE => OCE,          -- 1-bit Parallel to serial converter (data) clock enable
  ODV => ODV,          -- 1-bit Used for DDR3. Set to 1 if the ODELAY values exceeds 180 degrees
                           -- of the clock period.

  RST => RST,
  -- SHIFTIN1/SHIFTIN2: 1-bit (each) 1-bit Data Input Expansion
  SHIFTIN1 => SHIFTIN1,
  SHIFTIN2 => SHIFTIN2,
  -- T1 - T4: 1-bit (each) Parallel 3-State Inputs
  T1 => T1,
  T2 => T2,
  T3 => T3,
  T4 => T4,
  TCE => TCE,          -- 1-bit Parallel to serial converter (3-state) clock enable
  WC => WC             -- 1-bit Used for DDR3. Resets FIFO counters and turns IODELAY from IDELAY
                           -- to ODELAY.
);

-- End of OSERDESE1_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```

// OSERDESE1: Dedicated IOB Output Serializer
//      Virtex-6
// Xilinx HDL Language Template, version 11.1

OSERDESE1 #(
  .DATA_RATE_OQ("DDR"), // Defines whether the data changes at every clock edge or every positive
                        // clock edge with respect to CLK.
  .DATA_RATE_TQ("DDR"), // Defines whether the 3-state changes at every clock edge, every positive
                        // clock edge, or buffer configuration with respect to CLK.

```

```

.DATA_WIDTH(4),           // If DATA_RATE_OQ = DDR, value is limited to 4, 6, 8, or 10. If DATA_RATE_OQ
                          // = SDR, value is limited to 2, 3, 4, 5, 6, 7, or 8.
.DDR3_DATA(1),           // For DDR3, if the I/O is a DQ or DQS pin, set to 1. If control, address,
                          // clock, etc. set to 0.
.INIT_OQ(1'b0),          // Defines the initial value of OQ output.
.INIT_TQ(1'b0),          // Defines the initial value of TQ output.
.INTERFACE_TYPE("DEFAULT"), // To bypass DDR3 circuitry
.ODELAY_USED(0),         // BUFO drives IODELAY for write leveling or BUFO alignment.
.SERDES_MODE("MASTER"), // Defines whether the OSERDES module is a master or slave when width
                          // expansion is used.
.SRVAL_OQ(1'b0),         // Defines the value of OQ output when reset is invoked.
.SRVAL_TQ(1'b0),         // Defines the value of TQ output when reset is invoked.
.TRISTATE_WIDTH(4)       // If DATA_RATE_TQ = DDR, DATA_WIDTH = 4, and DATA_RATE_OQ = DDR, value is
                          // limited to 1 or 4. For all other settings of DATA_RATE_TQ, DATA_WIDTH, and
                          // DATA_RATE_OQ, value is limited to 1.
)
OSERDESE1_inst (
  .OCBEXTEND(OCBEXTEND),
  .OFB(OFB),              // 1-bit Feedback path for Data Output
  .OQ(OQ),               // 1-bit Data Path Output
  // SHIFTOUT1/SHIFTOUT2: 1-bit (each) Carry Out for data input expansion. Connect to SHIFTIN1/2 of
  ter.
  .SHIFTOUT1(SHIFTOUT1),
  .SHIFTOUT2(SHIFTOUT2),
  .TFB(TFB),
  .TQ(TQ),               // 1-bit 3-State Path Output
  .CLK(CLK),             // 1-bit High Speed Clock Input - This clock input is used to drive the
                          // parallel-to-serial converters. The possible source for the CLK port is
                          // from one of the following clock resources: - Ten global clock lines in a
                          // clock region - Four regional clock lines - Four clock capable I/Os
                          // (within adjacent clock region) - Fabric (through bypass).

  .CLKDIV(CLKDIV),       // 1-bit Divided High Speed Clock Input
  .CLKPERF(CLKPERF),     // 1-bit Input for performance path from PLL.
  .CLKPERFDELAY(CLKPERFDELAY), // 1-bit Delayed version of BUFO from IODELAY
  // D1 - D6: 1-bit (each) 1-bit parallel Data Input
  .D1(D1),
  .D2(D2),
  .D3(D3),
  .D4(D4),
  .D5(D5),
  .D6(D6),
  .OCE(OCE),             // 1-bit Parallel to serial converter (data) clock enable
  .ODV(ODV),             // 1-bit Used for DDR3. Set to 1 if the ODELAY values exceeds 180 degrees of
                          // the clock period.

  .RST(RST),
  // SHIFTIN1/SHIFTIN2: 1-bit (each) 1-bit Data Input Expansion
  .SHIFTIN1(SHIFTIN1),
  .SHIFTIN2(SHIFTIN2),
  // T1 - T4: 1-bit (each) Parallel 3-State Inputs
  .T1(T1),
  .T2(T2),
  .T3(T3),
  .T4(T4),
  .TCE(TCE),             // 1-bit Parallel to serial converter (3-state) clock enable
  .WC(WC),               // 1-bit Used for DDR3. Resets FIFO counters and turns IODELAY from IDELAY
                          // to ODELAY.
);

// End of OSERDESE1_inst instantiation

```

## 詳細情報

Virtex-6 FPGA の資料 (ユーザー ガイドおよびデータシート)

## PCIE\_2\_0

プリミティブ : PCI Express version 2.0 Compliant.





## 概要

このELEMENTは、RocketIO™ トランシーバ、ブロック RAM、さまざまなクロックリソースなど、FPGA のほかのリソースと併用します。エンドポイント、ルートポート、またはカスタム PCI EXPRESS® デザインを PCIe\_2\_0 を使用してインプリメントするには、必ず CORE Generator™ (ISE® Design Suite に含まれる) を使用して PCI EXPRESS デザイン用の LogiCORE™ IP コアを作成してください。LogiCORE は、PCIE\_2\_0 ソフトウェア プリミティブをインスタンス化し、インターフェイスを FPGA リソースに接続し、すべての属性を設定して、シンプルでユーザーにとって使いやすいインターフェイスを提供します。

## デザインの入力方法

このELEMENTをインスタンス化するには、PCI EXPRESS コアまたはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンス化しないでください。

## 詳細情報

- ・ [Virtex-6 FPGA RocketIO GTP トランシーバ ユーザー ガイド](#)
- ・ [Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PULLDOWN

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

PULLDOWN



### 概要

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

### ポートの説明

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

### デザインの入力方法

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

### VHDL 記述 (インスタンス化)

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

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

-- PULLDOWN: I/O Buffer Weak Pull-down
--           All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

PULLDOWN_inst : PULLDOWN
port map (
  O => O      -- Pulldown output (connect directly to top-level port)
);

-- End of PULLDOWN_inst instantiation
```

### Verilog 記述 (インスタンス化)

```
// PULLDOWN: I/O Buffer Weak Pull-down
//           All FPGA
// Xilinx HDL Libraries Guide, version 11.2

PULLDOWN PULLDOWN_inst (
  .O(O)       // Pulldown output (connect directly to top-level port)
);

// End of PULLDOWN_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## PULLUP

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



### 概要

このデザイン エLEMENTは、1 つの入力、トリステスト出力、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)
);

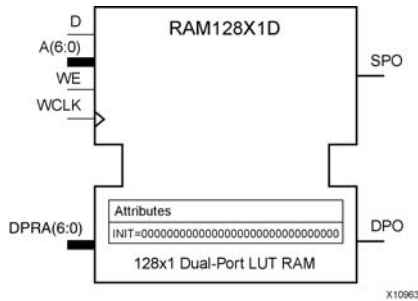
// End of PULLUP_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、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
```

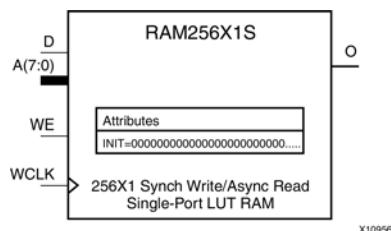
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## 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
);

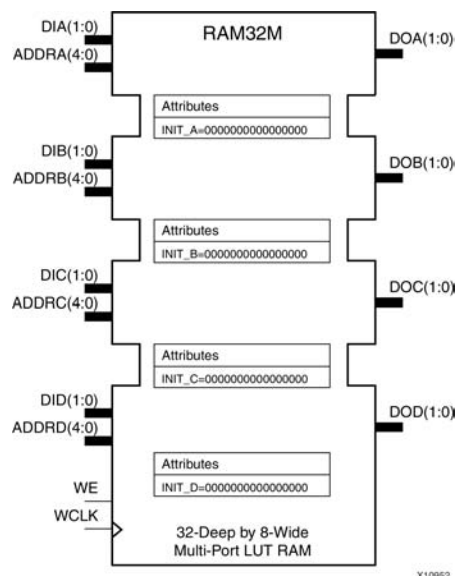
// End of RAM256X1S_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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、ADDRb、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。ADDRd が ADDRA、ADDRb、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
);

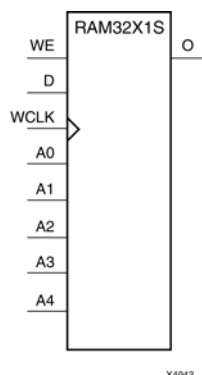
// End of RAM32M_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAM32X1S

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



### 概要

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

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

### 論理表

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

### デザインの入力方法

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

## 使用可能な属性

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

## VHDL 記述 (インスタンスエーション)

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

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

-- RAM32X1S: 32 x 1 posedge write distributed => LUT RAM
-- All FPGA
-- Xilinx HDL Libraries Guide, version 11.2

RAM32X1S_inst : RAM32X1S
generic map (
  INIT => X"00000000")
port map (
  O => O,          -- RAM output
  A0 => A0,         -- RAM address[0] input
  A1 => A1,         -- RAM address[1] input
  A2 => A2,         -- RAM address[2] input
  A3 => A3,         -- RAM address[3] input
  A4 => A4,         -- RAM address[4] input
  D => D,          -- RAM data input
  WCLK => WCLK,     -- Write clock input
  WE => WE         -- Write enable input
);

-- End of RAM32X1S_inst instantiation
```

## Verilog 記述 (インスタンスエーション)

```
// RAM32X1S: 32 x 1 posedge write distributed (LUT) RAM
// All FPGA
// Xilinx HDL Libraries Guide, version 11.2

RAM32X1S #(
  .INIT(32'h00000000) // Initial contents of RAM
) RAM32X1S_inst (
  .O(O),             // RAM output
  .A0(A0),           // RAM address[0] input
  .A1(A1),           // RAM address[1] input
  .A2(A2),           // RAM address[2] input
  .A3(A3),           // RAM address[3] input
  .A4(A4),           // RAM address[4] input
  .D(D),             // RAM data input
  .WCLK(WCLK),       // Write clock input
  .WE(WE)            // Write enable input
);

// End of RAM32X1S_inst instantiation
```

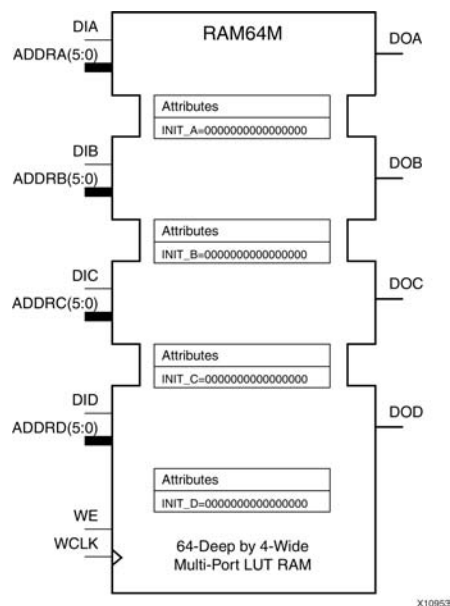
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## 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 メモリにできます。DID がグランドに接続される場合、DOD は使用されません。ADDR A、ADDR B、ADDR C 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。ADDR D が ADDR A、ADDR B、ADDR C 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 ADDRDB 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 ADDRDRD 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	ADDRDRD 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	ADDRDRD で指定された書き込みデータ入力 (読み出し出力は ADDRDB で指定)
DIC	入力	1	ADDRDRD 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 ADDRDRD 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
ADDRDB	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
ADDRDRD	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。RAM64M 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、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 ADDRDRD は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRDB、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、 $\text{ADDRy}[z] = \text{INIT\_y}[z]$  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 ADDRDR,
                // read addressed by ADDRRA
    .DIB(DIB), // RAM 1-bit data write input addressed by ADDRDR,
                // read addressed by ADDRBR
    .DIC(DIC), // RAM 1-bit data write input addressed by ADDRDR,
                // read addressed by ADDRRC
    .DID(DID), // RAM 1-bit data write input addressed by ADDRDR,
                // read addressed by ADDRDR
    .ADDRRA(ADDRRA), // Read port A 6-bit address input
    .ADDRBR(ADDRBR), // Read port B 6-bit address input
    .ADDRRC(ADDRRC), // Read port C 6-bit address input
    .ADDRDR(ADDRDR), // Read/Write port D 6-bit address input
    .WE(WE), // Write enable input
    .WCLK(WCLK) // Write clock input
);

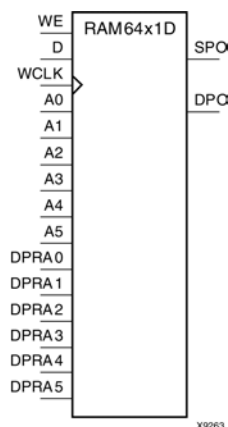
// End of RAM64M_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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 (インスタンスレーション)

```
// 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
);

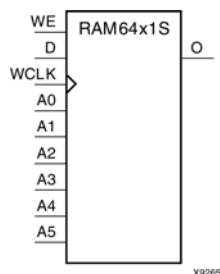
// End of RAM64X1D_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAM64X1S

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



### 概要

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

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

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

### 論理表

モード選択を次の論理表に示します。

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

### デザインの入力方法

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

### 使用可能な属性

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



## VHDL 記述 (インスタンスレーション)

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: 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 (インスタンスレーション)

```

// 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

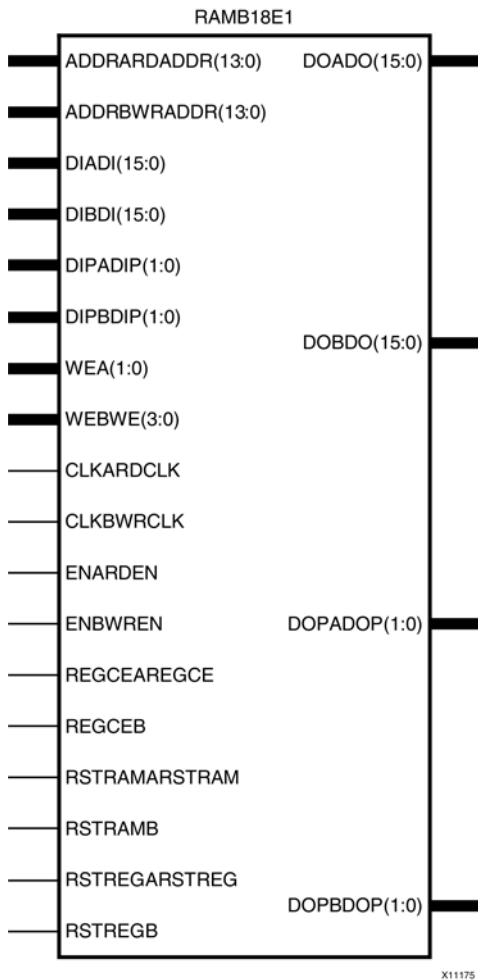
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAMB18E1

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



### 概要

Virtex®-6 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。またこのデザイン エLEMENT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ELEMENT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。また 36 ビット X 512 ワードの単純デュアル ポート RAM にコンフィギュレーションすることもできます。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	タイプ	幅	機能
ADDRARDADDR[13:0]	入力	14	ポート A アドレス入力バス/読み出しアドレス入力バス
ADDRBRWADDR[13:0]	入力	14	ポート B アドレス入力バス/書き込みアドレス入力バス

ポート名	タイプ	幅	機能
CLKARDCLK	入力	1	ポート A クロック入力/読み出しクロック入力
CLKBWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力
DIADI[15:0]	入力	16	ポート A データ入力バス/WRADDR でアドレス指定されるデータ入力バス RAM_MODE=SDP の場合、DIADI の論理値は DI[15:0] です。
DIBDI[15:0]	入力	16	ポート B データ入力バス/WRADDR でアドレス指定されるデータ入力バス RAM_MODE=SDP の場合、DIBDI の論理値は DI[31:16] です。
DIPADIP[1:0]	入力	2	ポート A パリティ データ入力バス/WRADDR でアドレス指定されるデータパリティ入力バス RAM_MODE=SDP の場合、DIPADIP の論理値は DIP[1:0] です。
DIPBDIP[1:0]	入力	2	ポート B パリティ データ入力バス/WRADDR でアドレス指定されるデータパリティ入力バス RAM_MODE=SDP の場合、DIPBDIP の論理値は DIP[3:2] です。
DOADO[15:0]	出力	16	ポート A データ出力バス/RDADDR でアドレス指定されるデータ出力バス RAM_MODE=SDP の場合、DOADO の論理値は DO[15:0] です。
DOBDO[15:0]	出力	16	ポート B データ出力バス/RDADDR でアドレス指定されるデータ出力バス RAM_MODE=SDP の場合、DOBDO の論理値は DO[31:16] です。
DOPADOP[1:0]	出力	2	ポート A パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス RAM_MODE=SDP の場合、DOPADOP の論理値は DOP[1:0] です。
DOPBDOP[1:0]	出力	2	ポート B パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス RAM_MODE=SDP の場合、DOPBDOP の論理値は DOP[3:2] です。
ENARDEN	入力	1	ポート A RAM イネーブル/リード イネーブル
ENBWREN	入力	1	ポート B RAM イネーブル/ライト イネーブル
REGCEAREGCE	入力	1	ポート A 出力レジスタ クロック イネーブル入力/出力レジスタ クロック イネーブル入力 (DO_REG=1 の場合のみ有効)
REGCEB	入力	1	ポート B 出力レジスタ クロック イネーブル (DO_REG=1 および RAM_MODE=TDP の場合のみ有効)
RSTRAMARSTRAM	入力	1	SRVAL_A で示される値に同期データ ラッチ セット/リセット。RSTRAMARSTRAM は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMARSTRAM でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。この信号は、RAM_MODE=TDP の場合はポート A の RSTRAMA、RAM_MODE=SDP の場合は RSTRAM です。
RSTRAMB	入力	1	SRVAL_B で示される値に同期データ ラッチ セット/リセット。RSTRAMB は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMB でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE=SDP の場合は使用されません。
RSTREGARSTREG	入力	1	SRVAL_A で示される値に同期出力レジスタ セット/リセット。RSTREGARSTREG は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_A は、この信号の優先度が REGCEAREGCE よりも高いかどうかを決定します。この信号は、RAM_MODE=TDP の場合はポート A の RSTREGA、RAM_MODE=SDP の場合は RSTREG です。

ポート名	タイプ	幅	機能
RSTREGB	入力	1	SRVAL_B で示される値に同期出力レジスタ セット/リセット。 RSTREGB は、DO_REG=1 のとき出力レジスタをセット/リセットします。 RSTREG_PRIORITY_B は、この信号の優先度が REGCEB よりも高いかどうかを決定します。RAM_MODE=SDP の場合は使用されません。
WEA[1:0]	入力	2	ポート A のバイト幅ライト イネーブル。RAM_MODE=SDP の場合は使用されません。異なるポート幅の WEA マッピングについてはユーザー ガイドを参照してください。
WEBWE[3:0]	入力	4	ポート B のバイト幅ライト イネーブル/ライト イネーブル。異なるポート幅の WEBWE マッピングについてはユーザー ガイドを参照してください。

## 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
DOA_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。TDP でポート A に、SDP では低いほうの 18 ビットまで (パリティビットを含む) に適用されます。
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が、clock-to-out タイムが長くなります。TDP でポート B に、SDP では高いほうの 18 ビット (パリティビットを含む) に適用されます。
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。TDP でポート A に、SDP では低いほうの 18 ビットまで (パリティビットを含む) に適用されます。
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。TDP でポート B に、SDP では高いほうの 18 ビット (パリティビットを含む) に適用されます。
INIT_FILE	文字列	0 ビット文字列	None	初期 RAM の内容を指定するファイル名

属性	タイプ	値	デフォルト	説明
INITIALDATA	16 進数	256'h00000000 000000000000 000000000000 000000000000 000000000000 0000 ~ 256'h00000000 000000000000 000000000000 0000000001.157 92089237316e+77	すべてゼロ	16kb のデータ メモリ アレイの初期値を指定
INITIALPARITY	16 進数	256'h00000000 000000000000 000000000000 000000000000 000000000000 000 ~ 256'h00000000 000000000000 000000000000 0000000001.157 92089237316e+77	すべてゼロ	2kb のパリティ メモリ アレイの初期値を指定
RAM_MODE	文字列	TDP、SDP	TDP	シングル デュアル ポート (SDP) または真のデュアル ポート (TDP) を選択します。
READ_WIDTH_A	整数	0、1、2、4、9、18、36、72	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 A 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。SDP の場合は、パリティビットを含む読み出し幅です。
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)。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。SDP では使用されません。
RSTREG_PRIORITY_A	文字列	RSTREG、REGCE	RSTREG	RSTREG または REGCE のレジスタ優先順位を選択します。TDP でポート A に、SDP では低いほうの 18 ビットまで (パリティビットを含む) に適用されます。
RSTREG_PRIORITY_B	文字列	RSTREG、REGCE	RSTREG	RSTREG または REGCE のレジスタ優先順位を選択します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。

属性	タイプ	値	デフォルト	説明
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。</p> <ul style="list-style-type: none"> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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。</li> <li>・ NONE に設定すると、エラーが無視され、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。</li> </ul> <p><b>メモ:</b> ALL に設定しておくとしミュレーション中に発生した問題に気付かない可能性があります。この値を変更する場合は注意が必要です。</p>
SRVAL_A	16 進数	18 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。TDP でポート A に、SDP では低いほうの 18 ビットまで (パリティビットを含む) に適用されます。
SRVAL_B	16 進数	18 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
WRITEMODE	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"> <li>・ 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。</li> <li>・ 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。</li> <li>・ 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。</li> </ul>

属性	タイプ	値	デフォルト	説明
				デフォルトは TDP の WRITE_FIRST です。SDP の READ_FIRST と同じにする必要があります。
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)。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。SDP では使用されません。
WRITE_WIDTH_B	整数	0、1、2、4、9、18、36、72	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。SDP の場合は、パリティビットを含む書き込み幅です。

## VHDL 記述 (インスタンスエーション)

```
-- RAMB18E1: 18K-bit Configurable Synchronous Block RAM
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2
```

```
RAMB18E1_inst : RAMB18E1
generic map (
  DOA_REG => 0,
  DOB_REG => 0,

  -- INITP_00 to INITP_07: Allows specification of the initial contents of the 2KB parity data memory
  -- array.
  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",
  -- INIT_00 to INIT_3F: Allows specification of the initial contents of the 16KB data memory array.
  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",
INIT_A => X"00000000",
INIT_B => X"00000000",
INIT_FILE => "NONE",

RAM_MODE => "TDP",
READ_WIDTH_A => 0,

READ_WIDTH_B => 0,
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
SIM_COLLISION_CHECK => "ALL",

SRVAL_A => X"00000000",
SRVAL_B => X"00000000",

WRITE_WIDTH_A => 0,
WRITE_WIDTH_B => 0,

-- WriteMode: Specifies output behavior of the port being written to: "WRITE_FIRST" = written value
-- appears on output port of the RAM "READ_FIRST" = previous RAM contents for that memory location appear
-- on the output port "NO_CHANGE" = previous value on the output port remains the same. Default is
-- WRITE_FIRST in TDP mode. Must equal READ_FIRST in SDP mode.
WRITE_MODE_A => "WRITE_FIRST",

```

```

-- Initial values on A
-- output port
-- Initial values on B
-- output port
-- File name of file
-- used to specify
-- initial RAM contents.
-- SDP or TDP
-- 0, 1, 2, 4, 9, 18, or
-- 36
-- 0, 1, 2, 4, 9, 18
-- RSTREG or REGCE
-- RSTREG or REGCE
-- Collision check
-- enable "ALL",
-- "WARNING_ONLY",
-- "GENERATE_X_ONLY" or
-- "NONE"
-- Set/Reset value for A
-- port output
-- Set/Reset value for B
-- port output
-- 0, 1, 2, 4, 9, or 18
-- 0, 1, 2, 4, 9, 18, or
-- 36
-- "WRITE_FIRST",
-- "READ_FIRST", or
-- "NO_CHANGE"

```



## Verilog 記述 (インスタンス化)

## Verilog 記述 (インスタンス化)

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v11.4) 2009 年 12 月 2 日

```
.INIT_11(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_12(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_13(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_14(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_15(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_16(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_17(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_18(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_19(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_1F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_20(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_21(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_22(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_23(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_24(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_25(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_26(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_27(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_28(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_29(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2A(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2B(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2C(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2D(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2E(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_2F(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_30(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_31(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_32(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_33(256'h0000000000000000000000000000000000000000000000000000000000000000),
.INIT_34(256'h0000000000000000000000000000000000000000000000000000000000000000),
.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),
.INIT_A(18'h00000), // Initial values on A
// output port
.INIT_B(18'h00000), // Initial values on B
// output port
.INIT_FILE("NONE"), // File name of file
// used to specify
// initial RAM
// contents.
.RAM_MODE("TDP"), // SDP or TDP
.READ_WIDTH_A(0), // 0, 1, 2, 4, 9, 18,
// or 36
.READ_WIDTH_B(0), // 0, 1, 2, 4, 9, 18
.RSTREG_PRIORITY_A("RSTREG"), // RSTREG or REGCE
.RSTREG_PRIORITY_B("RSTREG"), // RSTREG or REGCE
.SIM_COLLISION_CHECK("ALL"), // Collision check
// enable "ALL",
// "WARNING_ONLY",
// "GENERATE_X_ONLY" or
// "NONE"
.SRVAL_A(18'h00000), // Set/Reset value for
// A port output
.SRVAL_B(18'h00000), // Set/Reset value for
// B port output
.WRITE_WIDTH_A(0), // 0, 1, 2, 4, 9, or 18
.WRITE_WIDTH_B(0), // 0, 1, 2, 4, 9, 18,
// or 36
```

```

// WriteMode: Specifies output behavior of the port being written to: "WRITE_FIRST" = written value
// appears on output port of the RAM "READ_FIRST" = previous RAM contents for that memory location appear
// on the output port "NO_CHANGE" = previous value on the output port remains the same. Default is
// WRITE_FIRST in TDP mode. Must equal READ_FIRST in SDP mode.
.WRITE_MODE_A("WRITE_FIRST"), // "WRITE_FIRST",
                                // "READ_FIRST", or
                                // "NO_CHANGE"
.WRITE_MODE_B("WRITE_FIRST") // "WRITE_FIRST",
                                // "READ_FIRST", or
                                // "NO_CHANGE"
)
RAMB18E1_inst (
    .DOADO(DOADO), // 16-bit A port data/LSB data output
    .DOBDO(DOBDO), // 16-bit B port data/MSB data output
    .DOPADOP(DOPADOP), // 2-bit A port parity/LSB parity output
    .DOPBDOP(DOPBDOP), // 2-bit B port parity/MSB parity output
    .ADDRARDADDR(ADDRARDADDR), // 14-bit A port address/Read address input
    .ADDRBWRADDR(ADDRBWRADDR), // 14-bit B port address/Write address input
    .CLKARDCLK(CLKARDCLK), // 1-bit A port clock/Read clock input
    .CLKBWRCLK(CCLKBWRCLK), // 1-bit B port clock/Write clock input
    .DIADI(DIADI), // 16-bit A port data/LSB data input
    .DIBDI(DIBDI), // 16-bit B port data/MSB data input
    .DIPADIP(DIPADIP), // 2-bit A port parity/LSB parity input
    .DIPBDIP(DIPBDIP), // 2-bit B port parity/MSB parity input
    .ENARDEN(ENARDEN), // 1-bit A port enable/Read enable input
    .ENBWREN(ENBWREN), // 1-bit B port enable/Write enable input
    .REGCEAREGCE(REGCEAREGCE), // 1-bit A port register enable/Register enable input
    .REGCEB(REGCEB), // 1-bit B port register enable input
    .RSTRAMARSTRAM(RSTRAMARSTRAM), // 1-bit A port set/reset input
    .RSTRAMB(RSTRAMB), // 1-bit B port set/reset input
    .RSTREGARSTREG(RSTREGARSTREG), // 1-bit A port register set/reset input
    .RSTREGB(RSTREGB), // 1-bit B port register set/reset input
    .WEA(WEA), // 2-bit A port write enable input
    .WEBWE(WEBWE) // 4-bit B port write enable/Write enable input
);

// End of RAMB18E1_inst instantiation

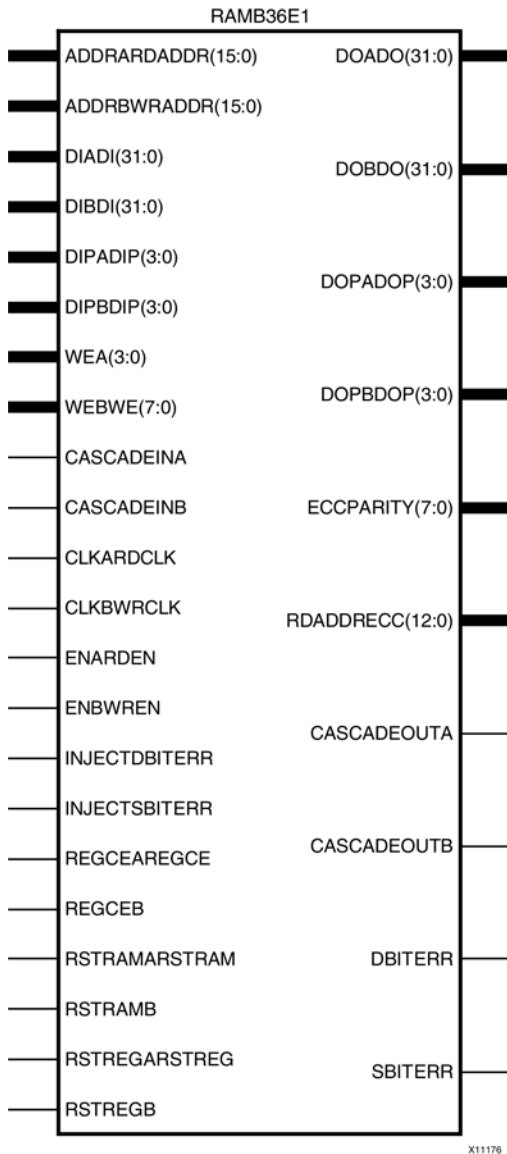
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## RAMB36E1

プリミティブ : 36K-bit Configurable Synchronous Block RAM



### 概要

Virtex®-6 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。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ELEMENTをカスケード 接続すると、大型の RAM を作成できます。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、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。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。

## ポートの説明

ポート名	タイプ	幅	機能
ADDRARDADDR[15:0]	入力	16	ポート A アドレス入力バス/読み出しアドレス入力バス
ADDRBWRADDR[15:0]	入力	16	ポート B アドレス入力バス/書き込みアドレス入力バス
CASCADEINA	入力	1	ポート A カスケード入力。RAM_MODE=SDP の場合は使用されません。
CASCADEINB	入力	1	ポート B カスケード入力。RAM_MODE=SDP の場合は使用されません。
CASCADEOUTA	出力	1	ポート A カスケード出力。RAM_MODE=SDP の場合は使用されません。
CASCADEOUTB	出力	1	ポート B カスケード出力。RAM_MODE=SDP の場合は使用されません。
CLKARDCLK	入力	1	ポート A クロック入力/読み出しクロック入力
CLKBWRCLK	入力	1	ポート B クロック入力/書き込みクロック入力
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。この機能を使用するには EN_ECC_READ を TRUE に設定します。RAM_MODE=TDP の場合は使用されません。
DIADI[31:0]	入力	32	ポート A データ入力バス/WRADDR でアドレス指定されるデータ入力バス RAM_MODE=SDP の場合、DIADI の論理値は DI[31:0] です。
DIBDI[31:0]	入力	32	ポート B データ入力バス/WRADDR でアドレス指定されるデータ入力バス RAM_MODE=SDP の場合、DIBDI の論理値は DI[63:32] です。
DIPADIP[3:0]	入力	4	ポート A パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス RAM_MODE=SDP の場合、DIPADIP の論理値は DIP[3:0] です。
DIPBDIP[3:0]	入力	4	ポート B パリティ データ入力バス/WRADDR でアドレス指定されるデータ パリティ入力バス RAM_MODE=SDP の場合、DIPBDIP の論理値は DIP[7:4] です。
DOADO[31:0]	出力	32	ポート A データ出力バス/RDADDR でアドレス指定されるデータ出力バス RAM_MODE=SDP の場合、DOADO の論理値は DO[31:0] です。
DOBDO[31:0]	出力	32	ポート B データ出力バス/RDADDR でアドレス指定されるデータ出力バス RAM_MODE=SDP の場合、DOBDO の論理値は DO[63:32] です。
DOPADOP[3:0]	出力	4	ポート A パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス RAM_MODE=SDP の場合、DOPADOP の論理値は DOP[3:0] です。
DOPBDOP[3:0]	出力	4	ポート B パリティ データ出力バス/RDADDR でアドレス指定されるパリティ データ出力バス RAM_MODE=SDP の場合、DOPBDOP の論理値は DOP[7:4] です。
ECCPARITY[7:0]	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 RAM_MODE=TDP の場合は使用されません。
ENARDEN	入力	1	ポート A RAM イネーブル/リード イネーブル
ENBWREN	入力	1	ポート B RAM イネーブル/ライト イネーブル
INJECTDBITERR	入力	1	ECC 機能が使用されている場合はダブル ビット エラーが挿入されます。
INJECTSBITERR	入力	1	ECC 機能が使用されている場合はシングル ビット エラーが挿入されます。
RDADDRECC[8:0]	出力	9	9 ビット ECC 読み出しアドレス。RAM_MODE=TDP の場合は使用されません。
REGCEAREGCE	入力	1	ポート A 出力レジスタ クロック イネーブル入力/出力レジスタ クロック イネーブル入力 (DO_REG=1 の場合のみ有効)
REGCEB	入力	1	ポート B 出力レジスタ クロック イネーブル (DO_REG=1 および RAM_MODE=TDP の場合のみ有効)

ポート名	タイプ	幅	機能
RSTRAMARSTRAM	入力	1	SRVAL_A で示される値に同期データ ラッチ セット/リセット。RSTRAMARSTRAM は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMARSTRAM でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。この信号は、RAM_MODE=TDP の場合はポート A の RSTRAMA、RAM_MODE=SDP の場合は RSTRAM です。
RSTRAMB	入力	1	SRVAL_B で示される値に同期データ ラッチ セット/リセット。RSTRAMB は、DO_REG=0 または 1 のとき BRAM データ出力ラッチをセット/リセットします。DO_REG=1 の場合は、RSTRAMB でリセットされる内部データ ラッチ ノードと BRAM の DO 出力の間に 1 サイクルのレイテンシがあります。RAM_MODE=SDP の場合は使用されません。
RSTREGARSTREG	入力	1	SRVAL_A で示される値に同期出力レジスタ セット/リセット。RSTREGARSTREG は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_A は、この信号の優先度が REGCEAREGCE よりも高いかどうかを決定します。この信号は、RAM_MODE=TDP の場合はポート A の RSTREGA、RAM_MODE=SDP の場合は RSTREG です。
RSTREGB	入力	1	SRVAL_B で示される値に同期出力レジスタ セット/リセット。RSTREGB は、DO_REG=1 のとき出力レジスタをセット/リセットします。RSTREG_PRIORITY_B は、この信号の優先度が REGCEB よりも高いかどうかを決定します。RAM_MODE=SDP の場合は使用されません。
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。RAM_MODE=TDP の場合は使用されません。
WEA[3:0]	入力	4	ポート A のバイト幅ライト イネーブル。RAM_MODE=SDP の場合は使用されません。異なるポート幅の WEA マッピングについてはユーザー ガイドを参照してください。
WEBWE[7:0]	入力	8	ポート B のバイト幅ライト イネーブル/ライト イネーブル。異なるポート幅の WEBWE マッピングについては『Virtex®-6ユーザー ガイド』を参照してください。

## 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
DOA_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。TDP でポート A に、SDP では低いほうの 36 ビットまで (パリティビットを含む) に適用されます。
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が、clock-to-out タイムが長くなります。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
EN_ECC_READ	ブール代数	FALSE、TRUE	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	FALSE、TRUE	FALSE	ECC エンコーダ回路をイネーブルにします。
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します。TDP でポート A に、SDP では低いほうの 36 ビットまで (パリティビットを含む) に適用されます。
INIT_B	16 進数	36 ビット値	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
INIT_FILE	文字列	0 ビット文字列	NONE	初期 RAM の内容を指定するファイル名
INIT_00 ~ INIT_7F	16 進数	256'h00000000 000000000000 000000000000 000000000000 000000000000 0000 256'h00000000 000000000000 000000000000 000000000000 000000000001.1 57920892373 16e+77	すべてゼロ	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'h00000000 000000000000 000000000000 000000000000 000000000000 0000 256'h00000000 000000000000 000000000000 000000000000 000000000001.1 57920892373 16e+77	すべてゼロ	4kb のパリティ 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
RAM_EXTENSION_A	文字列	LOWER、 NONE、 UPPER	NONE	ポート A カスケード モードを選択します。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。RAM_MODE=SDP の場合は使用されません。
RAM_EXTENSION_B	文字列	LOWER、 NONE、 UPPER	NONE	ポート B カスケード モードを選択します。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。RAM_MODE=SDP の場合は使用されません。
RAM_MODE	文字列	TDP、SDP	TDP	シンプル デュアル ポート (SDP) または真のデュアル ポート (TDP) を選択します。
READ_WIDTH_A	整数	0、1、2、4、9、18、 36、72	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 A 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。SDP の場合は、パリティビットを含む読み出し幅です。
READ_WIDTH_B	整数	0、1、2、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。SDP では使用されません。
RSTREG_PRIORITY_A	文字列	RSTREG、 REGCE	RSTREG	RSTREG または REGCE のレジスタ優先順位を選択します。TDP でポート A に、SDP では低いほうの 36 ビットまで (パリティビットを含む) に適用されます。
RSTREG_PRIORITY_B	文字列	RSTREG、 REGCE	RSTREG	RSTREG または REGCE のレジスタ優先順位を選択します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
SIM_COLLISION_CHECK	文字列	ALL、 GENERATE_X_ONLY、 NONE、または WARNING_ONLY	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> <li>NONE に設定すると、エラーが無視され、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。</li> </ul> <p><b>メモ:</b> ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>



属性	タイプ	値	デフォルト	説明
SRVAL_A	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。TDP でポート A に、SDP では低いほうの 36 ビットまで (パリティビットを含む) に適用されます。
SRVAL_B	16 進数	36 ビット値	すべてゼロ	同期リセット信号 (RSTREG) がアサートされたときの RAM の出力値を指定します。TDP でポート B に、SDP では高いほうのビット (パリティビットを含む) に適用されます。
WRITEMODE	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	書き込みが実行されるときポートの動作を指定します。 <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul> デフォルトは TDP の WRITE_FIRST です。SDP の READ_FIRST と同じにする必要があります。
WRITE_WIDTH_A	整数	0、1、2、4、9、18、 36	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)。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。SDP では使用されません。
WRITE_WIDTH_B	整数	0、1、2、4、9、18、 36、72	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。SDP の場合は、パリティビットを含む書き込み幅です。

## VHDL 記述 (インスタンスエーション)

```
-- RAMB36E1: 36K-bit Configurable Synchronous Block RAM
--          Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

RAMB36E1_inst : RAMB36E1
generic map (
    DOA_REG => 0,
    DOB_REG => 0,
    EN_ECC_READ => FALSE,
    EN_ECC_WRITE => FALSE,
    -- INITP_00 to INITP_0F: Allows specification of the initial contents of the 4KB parity data memory
    -- array.
    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",
    -- Optional output
    -- register on A port (0
    -- or 1)
    -- Optional output
    -- register on B port (0
    -- or 1)
    -- Enable ECC decoder,
    -- TRUE or FALSE
    -- Enable ECC encoder,
    -- TRUE or FALSE
```

[illegible]

```
-- Initial values on A
-- output port
-- Initial values on B
-- output port
-- File name of file
-- used to specify
-- initial RAM contents.
-- "UPPER", "LOWER", or
```

```

RAM_EXTENSION_B => "NONE",
RAM_MODE => "TDP",
READ_WIDTH_A => 0,
READ_WIDTH_B => 0,
RSTREG_PRIORITY_A => "RSTREG",
RSTREG_PRIORITY_B => "RSTREG",
SIM_COLLISION_CHECK => "ALL",

SRVAL_A => X"000000000",
SRVAL_B => X"000000000",
WRITE_WIDTH_A => 0,
WRITE_WIDTH_B => 0,

-- WriteMode: Specifies output behavior of the port being written to: "WRITE_FIRST" = written value
-- appears on output port of the RAM "READ_FIRST" = previous RAM contents for that memory location appear
-- on the output port "NO_CHANGE" = previous value on the output port remains the same. Default is
-- WRITE_FIRST in TDP mode. Must equal READ_FIRST in SDP mode.
WRITE_MODE_A => "WRITE_FIRST",

WRITE_MODE_B => "WRITE_FIRST"

)
port map (
  CASCADEOUTA => CASCADEOUTA, -- 1-bit A port cascade output
  CASCADEOUTB => CASCADEOUTB, -- 1-bit B port cascade output
  DBITERR => DBITERR, -- 1-bit double bit error status output
  DOADO => DOADO, -- 32-bit A port data/LSB data output
  DOBDO => DOBDO, -- 32-bit B port data/MSB data output
  DOPADOP => DOPADOP, -- 4-bit A port parity/LSB parity output
  DOPBDOP => DOPBDOP, -- 4-bit B port parity/MSB parity output
  ECCPARITY => ECCPARITY, -- 8-bit generated error correction parity
  RDADDRECC => RDADDRECC, -- 9-bit ECC read address. Not used when RAM_MODE=TDP.
  SBITERR => SBITERR, -- 1-bit single bit error status output
  ADDRARDADDR => ADDRARDADDR, -- 16-bit A port address/Read address input
  ADDRBRWADDR => ADDRBRWADDR, -- 16-bit B port address/Write address input
  CASCADEINA => CASCADEINA, -- 1-bit A port cascade input
  CASCADEINB => CASCADEINB, -- 1-bit B port cascade input
  CLKARDCLK => CLKARDCLK, -- 1-bit A port clock/Read clock input
  CLKBWRCLK => CLKBWRCLK, -- 1-bit B port clock/Write clock input
  DIADI => DIADI, -- 32-bit A port data/LSB data input
  DIBDI => DIBDI, -- 32-bit B port data/MSB data input
  DIPADIP => DIPADIP, -- 4-bit A port parity/LSB parity input
  DIPBDIP => DIPBDIP, -- 4-bit B port parity/MSB parity input
  ENARDEN => ENARDEN, -- 1-bit A port enable/Read enable input
  ENBWREN => ENBWREN, -- 1-bit B port enable/Write enable input
  INJECTDBITERR => INJECTDBITERR, -- 1-bit Inject a double bit error if ECC feature is used.
  INJECTSBITERR => INJECTSBITERR, -- 1-bit Inject a single bit error if ECC feature is used.
  REGCEAREGCE => REGCEAREGCE, -- 1-bit A port register enable/Register enable input
  REGCEB => REGCEB, -- 1-bit B port register enable input
  RSTRAMARSTRAM => RSTRAMARSTRAM, -- 1-bit A port set/reset input
  RSTRAMB => RSTRAMB, -- 1-bit B port set/reset input
  RSTREGARSTREG => RSTREGARSTREG, -- 1-bit A port register set/reset input
  RSTREGB => RSTREGB, -- 1-bit B port register set/reset input
  WEA => WEA, -- 4-bit A port write enable input
  WEBWE => WEBWE -- 8-bit B port write enable/Write enable input
);

-- End of RAMB36E1_inst instantiation

```

[illegible]

[illegible]

Virtex-6 ライブラリ ガイド (HDL 用)  
UG623 (v11.4) 2009 年 12 月 2 日

```

.ADDRBWRADDR(ADDRBWRADDR),      // 16-bit B port address/Write address input
.CASCADEINA(CASCADEINA),        // 1-bit A port cascade input
.CASCADEINB(CASCADEINB),        // 1-bit B port cascade input
.CLKARDCLK(CLKARDCLK),           // 1-bit A port clock/Read clock input
.CLKBWRCLK(CLKBWRCLK),           // 1-bit B port clock/Write clock input
.DIADI(DIADI),                   // 32-bit A port data/LSB data input
.DIBDI(DIBDI),                   // 32-bit B port data/MSB data input
.DIPADIP(DIPADIP),               // 4-bit A port parity/LSB parity input
.DIPBDIP(DIPBDIP),               // 4-bit B port parity/MSB parity input
.ENARDEN(ENARDEN),               // 1-bit A port enable/Read enable input
.ENBWREN(ENBWREN),               // 1-bit B port enable/Write enable input
.INJECTDBITERR(INJECTDBITERR),   // 1-bit Inject a double bit error if ECC feature is used.
.INJECTSBITERR(INJECTSBITERR),   // 1-bit Inject a single bit error if ECC feature is used.
.REGCEAREGCE(REGCEAREGCE),       // 1-bit A port register enable/Register enable input
.REGCEB(REGCEB),                 // 1-bit B port register enable input
.RSTRAMARSTRAM(RSTRAMARSTRAM),   // 1-bit A port set/reset input
.RSTRAMB(RSTRAMB),               // 1-bit B port set/reset input
.RSTREGARSTREG(RSTREGARSTREG),   // 1-bit A port register set/reset input
.RSTREGB(RSTREGB),               // 1-bit B port register set/reset input
.WEA(WEA),                       // 4-bit A port write enable input
.WEBWE(WEBWE)                    // 8-bit B port write enable/Write enable input
);

// End of RAMB36E1_inst instantiation

```

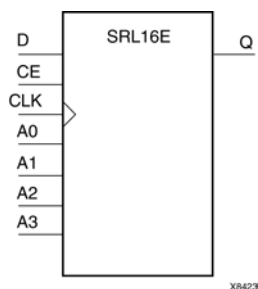
## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)



## SRL16E

プリミティブ : 16-Bit Shift Register Look-Up Table (LUT) with Clock Enable



### 概要

このデザイン エLEMENTは、シフトレジスタ ルックアップ テーブル (LUT) です。シフトレジスタの長さは、入力 A3、A2、A1、A0 の値によって決定されます。

シフトレジスタの長さは、固定することも、変動させることもできます。

- ・ **固定長のシフトレジスタを作成するには :** 入力 A3 ~ A0 の値を一定の値にします。シフトレジスタは 1 ~ 16 ビットの長さに設定できます。アドレス入力の値によるシフトレジスタの長さは、長さ =  $(8 \times A3) + (4 \times A2) + (2 \times A1) + A0 + 1$  という式で算出できます。A3、A2、A1、A0 がすべてゼロの場合 (0000) はシフトレジスタの長さは 1 ビットになり、すべて 1 の場合 (1111) は 16 ビットになります。
- ・ **シフトレジスタ長を動的に変化させるには :** 入力 A3 ~ A0 の値を変化させます。たとえば、A2、A1、A0 がすべて 1 の場合 (111) に A3 を 1 から 0 に切り替えると、シフトレジスタの長さは 16 ビットから 8 ビットに変化します。内部的には、シフトレジスタの長さは常に 16 ビットで、どのビットの値が出力されるかは入力 A3 ~ A0 の値によって決定されます。

シフトレジスタ LUT の初期値を指定するには、INIT 属性に 4 桁の 16 進数を割り当てます。一番左の桁が最上位ビットになります。INIT の値を指定しない場合は、シフトレジスタ LUT の内容はコンフィギュレーション中にゼロ (0000) にクリアされます。

CE が High の場合、クロック (CLK) が Low から High に切り替わるときに、D の値がシフトレジスタの第 1 ビットにロードされます。次にクロックが Low から High に切り替わるときに CE が High の場合、シフトレジスタの値は次の高位ビットにシフトされ、新しい値がロードされます。アドレス入力の値によってシフトレジスタの長さが決まり、Q にその値が出力されます。CE が Low の場合、クロック遷移は無視されます。

### 論理表

入力				出力
Am	CE	CLK	D	Q
Am	0	X	X	Q(Am)
Am	1	↑	D	Q(Am - 1)
m = 0、1、2、3				

## ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	4	SRL のワード数のダイナミック選択 <ul style="list-style-type: none"> <li>・ A=0000 ==&gt; 1 ビット シフト長</li> <li>・ A=1111 ==&gt; 16 ビット シフト長</li> </ul>

## 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
);

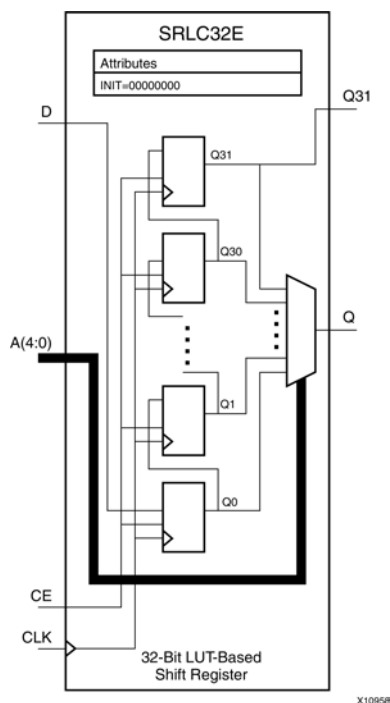
// End of SRL16E_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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
);

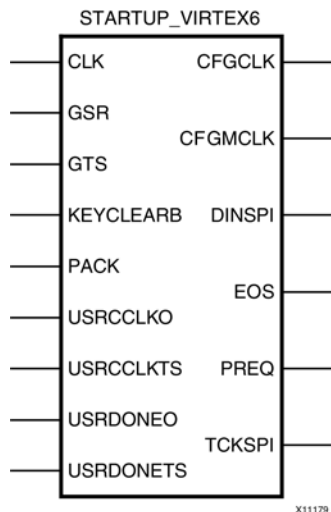
// End of SRLC32E_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## STARTUP\_VIRTEX6

プリミティブ : Virtex®-6 Configuration Start-Up Sequence Interface



### 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	タイプ	幅	機能
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
CLK	入力	1	ユーザー スタートアップ クロック
DINSPI	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
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。
GSR	入力	1	アクティブ High GSR 信号
GTS	入力	1	アクティブ High GSR 信号
KEYCLEARB	入力	1	バッテリー充電 RAM (BBRAM) からのクリア AES デクリプタ。BBRAM の内容を消去するには KEYCLEARB を約 200ns 間以上 Low にします。
PACK	入力	1	PROGRAM 確認応答
PREQ	出力	1	デバイスへの PROGRAM リクエスト
TCKSPI	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
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

## 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。CFGMCLK 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。



## 使用可能な属性

属性	タイプ	値	デフォルト	説明
PROG_USR	ブール代数	FALSE、TRUE	FALSE	プログラム イベント セキュリティ機能をアクティベート
CFGCLK_EN	ブール代数	FALSE、TRUE	FALSE	コンフィギュレーション ロジック メイン クロックを有効にします。
CFGMCLK_EN	ブール代数	FALSE、TRUE	FALSE	コンフィギュレーション内部オシレータ クロックを有効にします。

## VHDL 記述 (インスタンスエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- STARTUP_VIRTEX6: Virtex-6 Configuration Start-Up Sequence Interface
-- Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

STARTUP_VIRTEX6_inst : STARTUP_VIRTEX6
generic map (
    PROG_USR => FALSE           -- Activate program event security feature
)
port map (
    CFGCLK => CFGCLK,           -- 1-bit Configuration main clock output
    CFGMCLK => CFGMCLK,         -- 1-bit Configuration internal oscillator clock output
    DINSPI => DINSPI,           -- 1-bit Internal access to the DIN configuration pin when using SPI PROM
                                -- configuration

    EOS => EOS,                 -- 1-bit Active high signal indicates the End Of Configuration.
    PREQ => PREQ,               -- 1-bit PROGRAM request to fabric
    TCKSPI => TCKSPI,           -- 1-bit Internal access to the TCK configuration pin when using SPI PROM
                                -- configuration

    CLK => CLK,                 -- 1-bit User start-up clock
    GSR => GSR,                 -- 1-bit Active high Global Set/Reset signal
    GTS => GTS,                 -- 1-bit Active high Global 3-State signal
    KEYCLEARB => KEYCLEARB,     -- 1-bit Clear AES Decrypter Key from Battery-Backed RAM (BDRAM). To erase the
                                -- battery-backed-RAM contents, pull KEYCLEARB low for more than ~200ns.

    PACK => PACK,               -- 1-bit PROGRAM acknowledge
    USRCCLKO => USRCCLKO,       -- 1-bit Internal user CCLK
    USRCCLKTS => USRCCLKTS,     -- 1-bit Internal user CCLK 3-state enable
    USRDONEO => USRDONEO,       -- 1-bit Internal user DONE pin output control
    USRDONETS => USRDONETS     -- 1-bit User DONE 3-state enable
);

-- End of STARTUP_VIRTEX6_inst instantiation

```

## Verilog 記述 (インスタンスレーション)

```
// STARTUP_VIRTEX6: Virtex-6 Configuration Start-Up Sequence Interface
//                               Virtex-6
// Xilinx HDL Language Template, version 11.1

STARTUP_VIRTEX6 #(
    .PROG_USR("FALSE") // Activate program event security feature
)
STARTUP_VIRTEX6_inst (
    .CFGCLK(CFGCLK), // 1-bit Configuration main clock output
    .CFGMCLK(CFGMCLK), // 1-bit Configuration internal oscillator clock output
    .DINSPI(DINSPI), // 1-bit Internal access to the DIN configuration pin when using SPI PROM
                    // configuration

    .EOS(EOS), // 1-bit Active high signal indicates the End Of Configuration.
    .PREQ(PREQ), // 1-bit PROGRAM request to fabric
    .TCKSPI(TCKSPI), // 1-bit Internal access to the TCK configuration pin when using SPI PROM
                    // configuration

    .CLK(CLK), // 1-bit User start-up clock
    .GSR(GSR), // 1-bit Active high Global Set/Reset signal
    .GTS(GTS), // 1-bit Active high Global 3-State signal
    .KEYCLEARB(KEYCLEARB), // 1-bit Clear AES Decrypter Key from Battery-Backed RAM (BBRAM). To erase the
                           // battery-backed-RAM contents, pull KEYCLEARB low for more than ~200ns.

    .PACK(PACK), // 1-bit PROGRAM acknowledge
    .USRCCLKO(USRCCLKO), // 1-bit Internal user CCLK
    .USRCCLKTS(USRCCLKTS), // 1-bit Internal user CCLK 3-state enable
    .USRDONEO(USRDONEO), // 1-bit Internal user DONE pin output control
    .USRDONETS(USRDONETS) // 1-bit User DONE 3-state enable
);

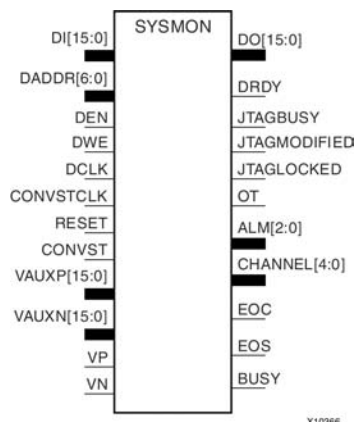
// End of STARTUP_VIRTEX6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## 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])) を紹介します。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;

-- 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 (インスタンス化)

```

// 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
    .SIM_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

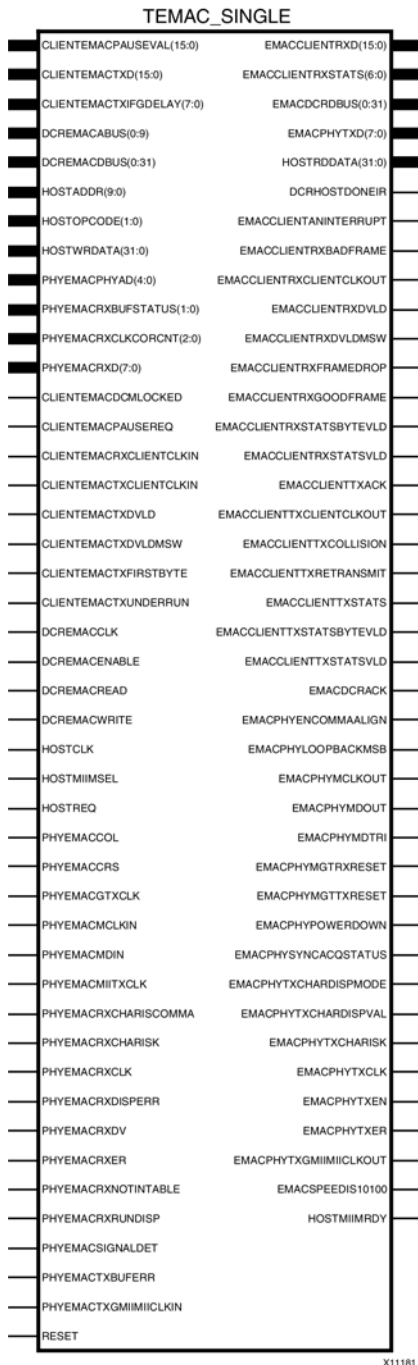
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## TEMAC\_SINGLE

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



## 概要

Virtex®-6 FPGA エンベデッドトライモードイーサネット MAC をインスタンス化するために必要なポートおよび属性を提供します。SecureIP 暗号化 HDL を含むため、論理およびタイミング シミュレーションにも使用されます。このプリミティブは、イーサネット MAC ラップを作成するために、CORE Generator™ を使用して特定ニーズに合わせて変更することができます。



## デザインの入力方法

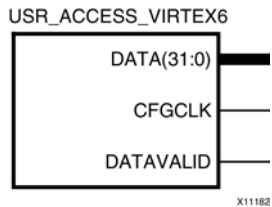
このELEMENTをインスタンスエートするには、エンベデッド開発キット (EDK) またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンスエートしないでください。

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

## USR\_ACCESS\_VIRTEX6

プリミティブ : Virtex-6 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	タイプ	幅	機能
CFGCLK	出力	1	コンフィギュレーション クロック
DATA[31:0]	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

### 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_VIRTEX6: Virtex-6 User Access Register
--                               Virtex-6
-- Xilinx HDL Libraries Guide, version 11.2

USR_ACCESS_VIRTEX6_inst : USR_ACCESS_VIRTEX6
generic map (
)
port map (
  CFGCLK => CFGCLK,      -- 1-bit Configuration Clock
  DATA  => DATA,        -- 32-bit Configuration Output Data
  DATAVALID => DATAVALID -- 1-bit Active high DATA port contains valid data
);

-- End of USR_ACCESS_VIRTEX6_inst instantiation
```

## Verilog 記述 (インスタンス化)

```
// USR_ACCESS_VIRTEX6: Virtex-6 User Access Register
//                               Virtex-6
// Xilinx HDL Language Template, version 11.1

USR_ACCESS_VIRTEX6 USR_ACCESS_VIRTEX6_inst (
    .CFGCLK(CFGCLK),           // 1-bit Configuration Clock
    .DATA(DATA),               // 32-bit Configuration Output Data
    .DATAVALID(DATAVALID)     // 1-bit Active high DATA port contains valid data
);

// End of USR_ACCESS_VIRTEX6_inst instantiation
```

## 詳細情報

[Virtex-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)