

Spartan-6 ライブラリ ガイド (HDL 用)

UG615 (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	13
デザインの入力方法	13
2 : UniMacro について	15
BRAM_SDP_MACRO	16
概要	16
ポートの説明	16
コンフィギュレーション表	17
デザインの入力方法	17
使用可能な属性	17
詳細情報	24
BRAM_SINGLE_MACRO	25
概要	25
ポートの説明	25
コンフィギュレーション表	26
デザインの入力方法	28
使用可能な属性	28
詳細情報	33
BRAM_TDP_MACRO	34
概要	34
ポートの説明	34
コンフィギュレーション表	35
デザインの入力方法	37
使用可能な属性	37
詳細情報	43
ADDMACC_MACRO	44
概要	44
ポートの説明	44
デザインの入力方法	45
使用可能な属性	45
詳細情報	46
ADDSUB_MACRO	47
概要	47
ポートの説明	47
デザインの入力方法	47
使用可能な属性	48
詳細情報	49
COUNTER_LOAD_MACRO	50
概要	50
ポートの説明	50
デザインの入力方法	50
使用可能な属性	51
詳細情報	52
MACC_MACRO	53
概要	53
ポートの説明	53
デザインの入力方法	54
使用可能な属性	54
詳細情報	55
DSP_MULT_MACRO	56
概要	56
ポートの説明	56
デザインの入力方法	56
使用可能な属性	57
詳細情報	58

3 : ファンクション別分類	59
4 : デザイン エレメント	65
AND2B1L	66
概要	66
論理表	66
ポートの説明	66
デザインの入力方法	66
詳細情報	66
BSCAN_SPARTAN6	67
概要	67
ポートの説明	68
デザインの入力方法	68
使用可能な属性	68
詳細情報	69
BUFG	70
概要	70
ポートの説明	70
デザインの入力方法	70
詳細情報	71
BUFGCE	72
概要	72
論理表	72
デザインの入力方法	72
詳細情報	73
BUFGCE_1	74
概要	74
論理表	74
デザインの入力方法	74
詳細情報	75
BUFGMUX	76
概要	76
論理表	76
ポートの説明	76
デザインの入力方法	76
使用可能な属性	77
詳細情報	77
BUFGMUX_1	78
概要	78
論理表	78
デザインの入力方法	78
詳細情報	79
BUFGP	80
概要	80
デザインの入力方法	80
詳細情報	80
BUFH	81
概要	81
ポートの説明	81
デザインの入力方法	81
詳細情報	81
BUFIO2	82
概要	82
ポートの説明	82
デザインの入力方法	82
使用可能な属性	82
詳細情報	83
BUFIO2FB	84
概要	84
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
BUFPLL	86
概要.....	86
ポートの説明	86
デザインの入力方法.....	86
使用可能な属性.....	86
詳細情報	87
BUFPLL_MCB	88
概要.....	88
詳細情報	88
CARRY4.....	89
概要.....	89
ポートの説明	89
デザインの入力方法.....	90
詳細情報	90
CFGLUT5	91
概要.....	91
ポートの説明	91
デザインの入力方法.....	92
使用可能な属性.....	92
詳細情報	93
DCM_CLKGEN	94
概要.....	94
ポートの説明	94
デザインの入力方法.....	95
使用可能な属性.....	95
詳細情報	97
DCM_SP	98
概要.....	98
ポートの説明	98
デザインの入力方法.....	99
使用可能な属性.....	99
詳細情報	102
DNA_PORT	103
概要.....	103
ポートの説明	103
デザインの入力方法.....	103
使用可能な属性.....	104
詳細情報	104
DSP48A1	105
概要.....	106
ポートの説明	106
デザインの入力方法.....	108
使用可能な属性.....	108
詳細情報	110
FDCE	111
概要.....	111
論理表.....	111
デザインの入力方法.....	111
使用可能な属性.....	111
詳細情報	112
FDPE.....	113
概要.....	113
論理表.....	113
デザインの入力方法.....	113
使用可能な属性.....	113
詳細情報	113

FDRE.....	114
概要.....	114
論理表.....	114
デザインの入力方法.....	114
使用可能な属性.....	114
詳細情報	114
FDSE	115
概要.....	115
論理表.....	115
デザインの入力方法.....	115
使用可能な属性.....	115
詳細情報	115
GTPA1_DUAL	116
概要.....	117
デザインの入力方法.....	117
詳細情報	117
IBUF.....	118
概要.....	118
ポートの説明	118
デザインの入力方法.....	118
使用可能な属性.....	118
詳細情報	119
IBUFDS.....	120
概要.....	120
論理表.....	120
ポートの説明	120
デザインの入力方法.....	120
使用可能な属性.....	120
詳細情報	121
IBUFG	122
概要.....	122
ポートの説明	122
デザインの入力方法.....	122
使用可能な属性.....	122
詳細情報	123
IBUFGDS	124
概要.....	124
論理表.....	124
ポートの説明	124
デザインの入力方法.....	124
使用可能な属性.....	125
詳細情報	125
ICAP_SPARTAN6	126
概要.....	126
ポートの説明	126
デザインの入力方法.....	126
詳細情報	127
IDDR2.....	128
概要.....	128
論理表.....	128
デザインの入力方法.....	128
使用可能な属性.....	129
詳細情報	130
IOBUF	131
概要.....	131
論理表.....	131
ポートの説明	131
デザインの入力方法.....	131
使用可能な属性.....	132

詳細情報	133
IOBUFDS	134
概要	134
論理表	134
ポートの説明	134
デザインの入力方法	134
使用可能な属性	135
詳細情報	135
IODELAY2	136
概要	136
ポートの説明	136
デザインの入力方法	137
使用可能な属性	137
詳細情報	140
IODRP2	141
概要	141
詳細情報	141
IODRP2_MCB	142
概要	142
詳細情報	142
ISERDES2	143
概要	143
ポートの説明	143
デザインの入力方法	144
使用可能な属性	145
詳細情報	147
KEEPER	148
概要	148
ポートの説明	148
デザインの入力方法	148
詳細情報	149
LDCE	150
概要	150
論理表	150
デザインの入力方法	150
使用可能な属性	150
詳細情報	150
LDPE	151
概要	151
論理表	151
デザインの入力方法	151
使用可能な属性	151
詳細情報	151
LUT5	152
概要	152
論理表	153
ポートの説明	154
デザインの入力方法	154
使用可能な属性	154
詳細情報	155
LUT5_D	156
概要	156
論理表	157
ポートの説明	158
デザインの入力方法	158
使用可能な属性	158
詳細情報	159
LUT5_L	160
概要	160

論理表.....	161
ポートの説明	162
デザインの入力方法.....	162
使用可能な属性.....	162
詳細情報	163
LUT6	164
概要.....	164
論理表.....	164
ポートの説明	166
デザインの入力方法.....	167
使用可能な属性.....	167
詳細情報	168
LUT6_2	169
概要.....	169
論理表.....	169
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	172
LUT6_D.....	173
概要.....	173
論理表.....	173
ポートの説明	175
デザインの入力方法.....	176
使用可能な属性.....	176
詳細情報	177
LUT6_L.....	178
概要.....	178
論理表.....	178
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	182
MUXF7	183
概要.....	183
論理表.....	183
ポートの説明	183
デザインの入力方法.....	183
詳細情報	184
MUXF7_D.....	185
概要.....	185
論理表.....	185
ポートの説明	185
デザインの入力方法.....	185
詳細情報	186
MUXF7_L.....	187
概要.....	187
論理表.....	187
ポートの説明	187
デザインの入力方法.....	187
詳細情報	188
MUXF8	189
概要.....	189
論理表.....	189
ポートの説明	189
デザインの入力方法.....	189
詳細情報	190
MUXF8_D.....	191
概要.....	191

論理表.....	191
ポートの説明	191
デザインの入力方法.....	191
詳細情報	192
MUXF8L	193
概要.....	193
論理表.....	193
ポートの説明	193
デザインの入力方法.....	193
詳細情報	194
OBUF	195
概要.....	195
ポートの説明	195
デザインの入力方法.....	195
使用可能な属性.....	195
詳細情報	196
OBUFDS	197
概要.....	197
論理表.....	197
ポートの説明	197
デザインの入力方法.....	197
使用可能な属性.....	197
詳細情報	198
OBUFT	199
概要.....	199
論理表.....	199
ポートの説明	199
デザインの入力方法.....	199
使用可能な属性.....	199
詳細情報	200
OBUFTDS	201
概要.....	201
論理表.....	201
ポートの説明	201
デザインの入力方法.....	201
使用可能な属性.....	201
詳細情報	202
ODDR2	203
概要.....	203
論理表.....	203
デザインの入力方法.....	203
使用可能な属性.....	204
詳細情報	205
OR2L	206
概要.....	206
論理表.....	206
ポートの説明	206
デザインの入力方法.....	206
詳細情報	206
OSERDES2	207
概要.....	207
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	212
PCIE_A1	213
概要.....	214
デザインの入力方法.....	214
詳細情報	214

PLL_BASE	215
概要.....	215
ポートの説明	215
デザインの入力方法.....	216
使用可能な属性.....	216
詳細情報	218
POST_CRC_INTERNAL	219
概要.....	219
ポートの説明	219
デザインの入力方法.....	219
詳細情報	219
PULLDOWN	220
概要.....	220
ポートの説明	220
デザインの入力方法.....	220
詳細情報	221
PULLUP	222
概要.....	222
ポートの説明	222
デザインの入力方法.....	222
詳細情報	223
RAM128X1D	224
概要.....	224
ポートの説明	224
デザインの入力方法.....	225
使用可能な属性.....	225
詳細情報	226
RAM256X1S.....	227
概要.....	227
ポートの説明	227
デザインの入力方法.....	227
使用可能な属性.....	228
詳細情報	228
RAM32M	229
概要.....	229
ポートの説明	230
デザインの入力方法.....	230
使用可能な属性.....	231
詳細情報	232
RAM32X1S	233
概要.....	233
論理表.....	233
デザインの入力方法.....	233
使用可能な属性.....	234
詳細情報	234
RAM64M	235
概要.....	235
ポートの説明	236
デザインの入力方法.....	236
使用可能な属性.....	237
詳細情報	238
RAM64X1D.....	239
概要.....	239
論理表.....	239
デザインの入力方法.....	240
使用可能な属性.....	240
詳細情報	241
RAM64X1S	242
概要.....	242

論理表.....	242
デザインの入力方法.....	242
使用可能な属性.....	242
詳細情報	243
RAMB16BWER.....	244
概要.....	244
ポートの説明	244
デザインの入力方法.....	245
使用可能な属性.....	246
詳細情報	247
RAMB8BWER.....	248
概要.....	248
ポートの説明	248
デザインの入力方法.....	250
使用可能な属性.....	250
詳細情報	255
SRL16E	256
概要.....	256
論理表.....	256
ポートの説明	257
デザインの入力方法.....	257
使用可能な属性.....	257
詳細情報	258
SRLC32E	259
概要.....	259
ポートの説明	259
デザインの入力方法.....	260
使用可能な属性.....	260
詳細情報	261
STARTUP_SPARTAN6	262
概要.....	262
ポートの説明	262
デザインの入力方法.....	262
詳細情報	263
SUSPEND_SYNC	264
概要.....	264
ポートの説明	264
デザインの入力方法.....	264
詳細情報	265

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

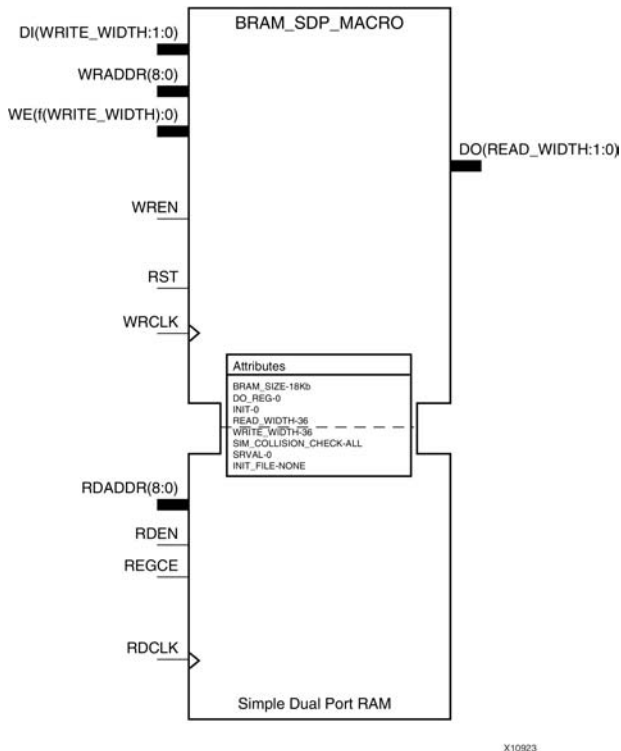
このセクションでは、このアーキテクチャで利用できる UniMacro について説明します。UniMacro は、アルファベット順に並べられています。

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

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

使用可能な属性

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

属性	タイプ	値	デフォルト	説明
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
READ_WIDTH、 WRITE_WIDTH	整数	1 ~ 72	36	DI/DO バスの幅を指定。READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
INIT_FILE	文字列	0 ビット STRING	NONE	初期値を含むファイルの名前
SIM_COLLISION_ CHECK	文字列	ALL、WARNING_ ONLY、 GENERATE_X_ ONLY、 NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>
SIM_MODE	文字列	SAFE または FAST	SAFE	シミュレーションのみの属性です。FAST に設定すると、シミュレーションモデルがパフォーマンス重視モードで実行されます。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。
SRVAL	16 進数	72 ビット値	すべてゼロ	同期リセット信号 (RST) がアサートされたときの DO ポートの出力値を指定します。
INIT_00 ~ INIT_7F	16 進数	256 ビット値	すべてゼロ	16kb または 32kb のデータ メモリ アレイの初期値を指定
INITP_00 ~ INITP_0F	16 進数	256 ビット値	すべてゼロ	2kb または 4kb のパリティ データ メモリ アレイの初期値を指定

VHDL 記述 (インスタンスエーション)

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

```
library UNIMACRO;
```

```
use unimacro.Vcomponents.all;

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

BRAM_SDP_MACRO_inst : BRAM_SDP_MACRO
generic map (
  BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
  DEVICE => "VIRTEX5" -- Target device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
  WRITE_WIDTH => 0,    -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  READ_WIDTH => 0,     -- Valid values are 1-72 (37-72 only valid when BRAM_SIZE="36Kb")
  DO_REG => 0, -- Optional output register (0 or 1)
  INIT_FILE => "NONE",
  SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
                                -- "GENERATE_X_ONLY" or "NONE"
  SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
                    -- see "Synthesis and Simulation Design Guide" for details
  SRVAL => X"00000000000000000000", -- Set/Reset value for port output
  INIT => X"00000000000000000000", -- Initial values on output port
  -- The following INIT_xx declarations specify the initial contents of the RAM
  INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
  INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

[illegible]

```

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

-- The next set of INITP_xx are for the parity bits
INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
INITP_07 => X"0000000000000000000000000000000000000000000000000000000000000000",

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

```

Verilog 記述 (インスタンスレーション)

```

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

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

```



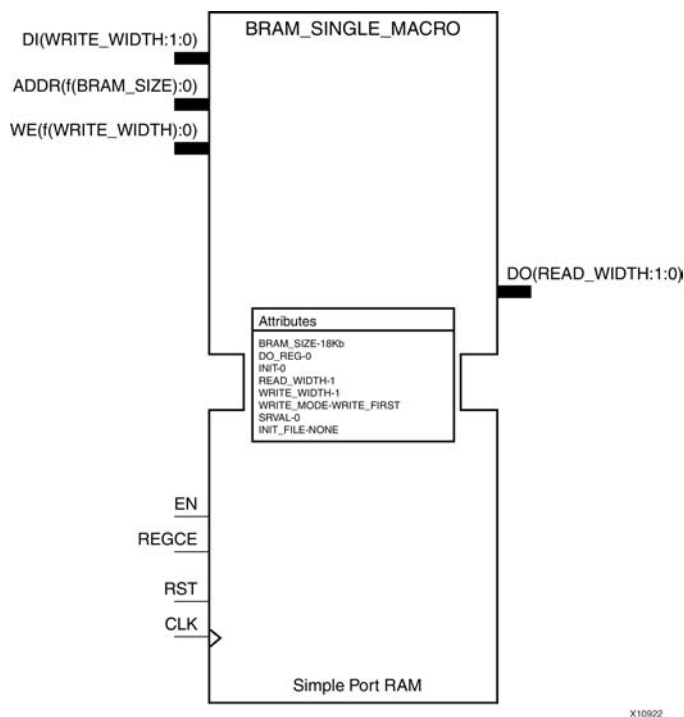
```
.WRADDR(WRADDR), // Input write address  
.WRCLK(WRCLK),   // Input write clock  
.WREN(WREN)      // Input write port enable  
);  
  
// End of BRAM_SDP_MACRO_inst instantiation
```

詳細情報

[Spartan-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
```

```
BRAM_SINGLE_MACRO_inst : BRAM_SINGLE_MACRO
generic map (
    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_A => X"000000000", -- 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 => X"000000000", -- Set/Reset value for port output
    WRITE_MODE => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
    -- The following INIT_xx declarations specify the initial contents of the RAM
    INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_14 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_15 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_16 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_17 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_18 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_19 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_1F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_20 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_21 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_22 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_23 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_24 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_25 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_26 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_27 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_28 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_29 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_2F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_30 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_31 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_32 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_33 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_34 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_35 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_36 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_37 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_38 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_39 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

[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)

```

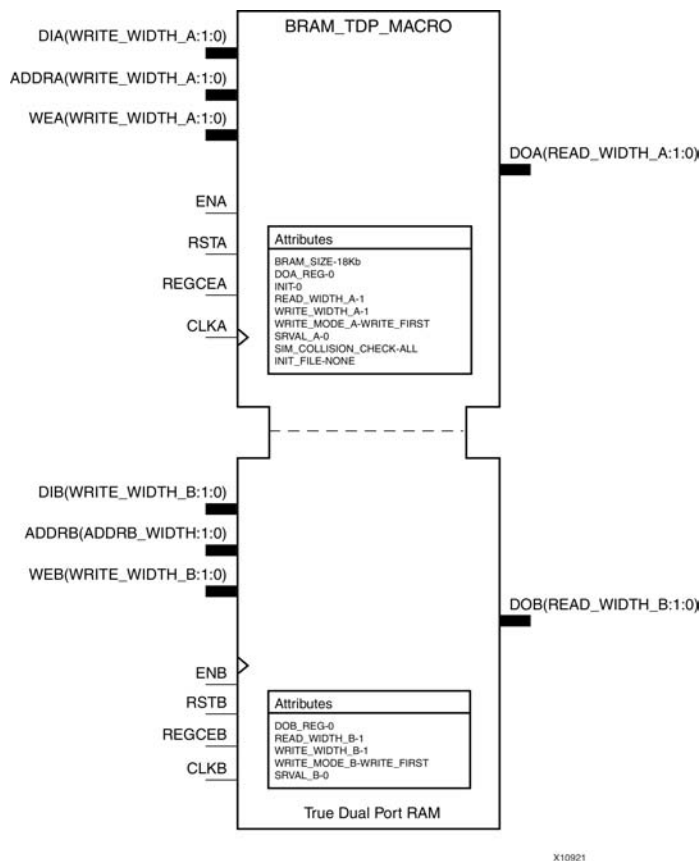
// BRAM_SINGLE_MACRO: Single Port RAM
//                               Virtex-5, Virtex-6, Spartan-6
// Xilinx HDL Libraries Guide, version 11.2

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),

```


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。clock-to-out タイムが長くなります。
INIT	16 進数	72 ビット値	すべてゼロ	コンフィギュレーション後の出力の初期値を指定
INIT_FILE	文字列	0 ビット STRING	NONE	初期値を含むファイルの名前
READ_WIDTH、WRITE_WIDTH	整数	1 ～72	36	DI/DO バスの幅を指定。READ_WIDTH と WRITE_WIDTH に同じ値を指定する必要があります。
SIM_COLLISION_CHECK	文字列	ALL、WARNING_ONLY、GENERATE_X_ONLY、または NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。詳細は次のとおりです。</p> <ul style="list-style-type: none"> ・ ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 ・ WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 ・ GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 ・ NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。詳細は、『合成/シミュレーション デザイン ガイド』を参照してください。</p>

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

VHDL 記述 (インスタンスエーション)

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

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

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

BRAM_TDP_MACRO_inst : BRAM_TDP_MACRO
generic map (
    BRAM_SIZE => "18Kb", -- Target BRAM, "18Kb" or "36Kb"
    DEVICE => "VIRTEX5", -- Target Device: "VIRTEX5", "VIRTEX6", "SPARTAN6"
    DOA_REG => 0, -- Optional port A output register (0 or 1)
    DOB_REG => 0, -- Optional port B output register (0 or 1)
    INIT_A => X"000000000", -- Initial values on A output port
    INIT_B => X"000000000", -- Initial values on B output port
    INIT_FILE => "NONE",
    READ_WIDTH_A => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
    READ_WIDTH_B => 0, -- Valid values are 1-36 (19-36 only valid when BRAM_SIZE="36Kb")
    SIM_COLLISION_CHECK => "ALL", -- Collision check enable "ALL", "WARNING_ONLY",
    -- "GENERATE_X_ONLY" or "NONE"
    SIM_MODE => "SAFE", -- Simulation: "SAFE" vs "FAST",
    -- see "Synthesis and Simulation Design Guide" for details
    SRVAL_A => X"000000000", -- Set/Reset value for A port output
    SRVAL_B => X"000000000", -- Set/Reset value for B port output
    WRITE_MODE_A => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
    WRITE_MODE_B => "WRITE_FIRST", -- "WRITE_FIRST", "READ_FIRST" or "NO_CHANGE"
    WRITE_WIDTH_A => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
    WRITE_WIDTH_B => 0, -- Valid values are 1, 2, 4, 9, 18 or 36 (36 only valid when BRAM_SIZE="36Kb")
    -- The following INIT_xx declarations specify the initial contents of the RAM
    INIT_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_04 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_05 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_06 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_07 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_08 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_09 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0A => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0B => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0C => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0D => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0E => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_0F => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_10 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_11 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_12 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INIT_13 => X"0000000000000000000000000000000000000000000000000000000000000000",
```

39

[illegible]

Verilog 記述 (インスタンス化)

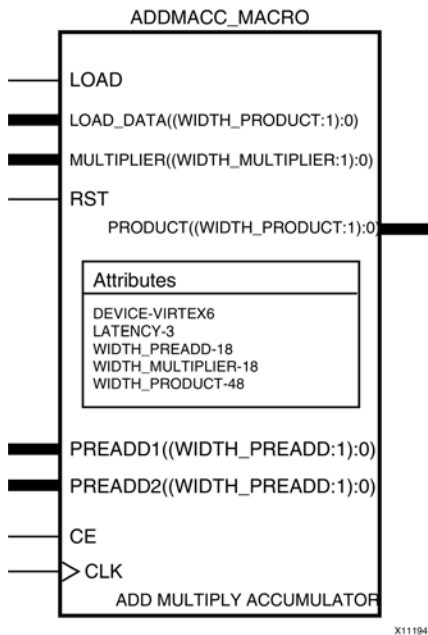
Spartan-6 ライブラリ ガイド (HDL 用)
UG615 (v11.4) 2009 年 12 月 2 日

[illegible]

Spartan-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"> 1 - MREG == 1 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1
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
);

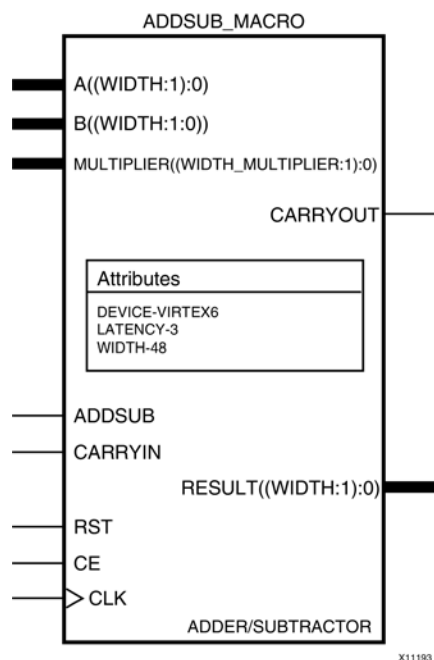
// End of ADDMACC_MACRO_inst instantiation
```

詳細情報

[Spartan-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"> 1 - PREG == 1 2 - AREG == BREG == CREG == PREG
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
);

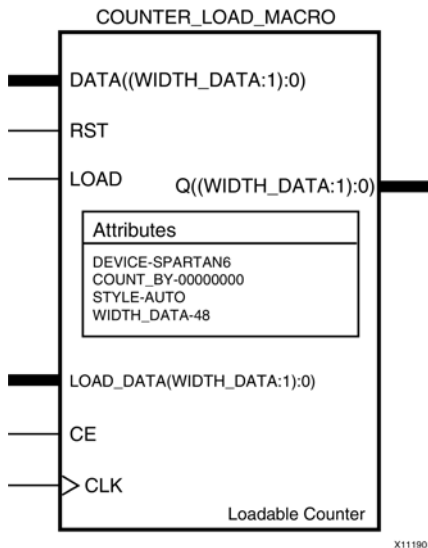
// End of ADDSUB_MACRO_inst instantiation
```

詳細情報

[Spartan-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
);

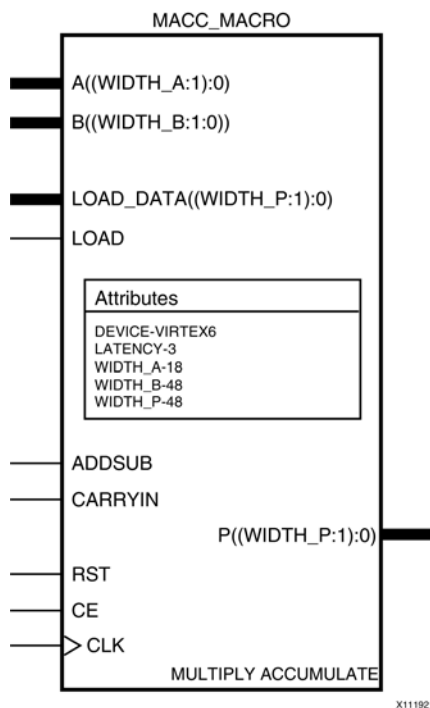
// End of COUNTER_LOAD_MACRO_inst instantiation
```

詳細情報

[Spartan-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"> 1 - MREG == 1 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1
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

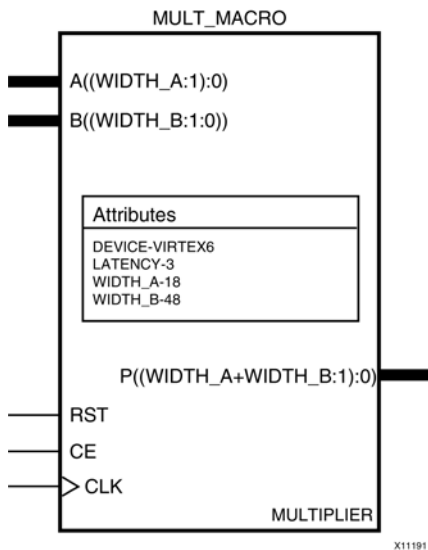
```

詳細情報

[Spartan-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"> ・ 1 - MREG == 1 ・ 2 - AREG == BREG == 1 および MREG == 1、または MREG == 1 および PREG == 1 ・ 3 - AREG == BREG == 1 および MREG == 1 および PREG == 1 ・ 4 - AREG == BREG == 2 および MREG == 1 および PREG == 1
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

[Spartan-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_A1	プリミティブ : PCI Express

四則演算ファンクション

デザイン エLEMENT	説明
DSP48A1	プリミティブ : Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block

クロック コンポーネント

デザイン エLEMENT	説明
BUFG	コンビニエンス プリミティブ : Global Clock Buffer
BUFGCE	コンビニエンス プリミティブ : Global Clock Buffer with Clock Enable
BUFGMUX	プリミティブ : Global Clock MUX Buffer
BUFGMUX_1	プリミティブ : Global Clock MUX Buffer with Output State 1
BUFH	プリミティブ : Clock buffer for a single clocking region
BUFIO2	プリミティブ : Dual Clock Buffer and Strobe Pulse
BUFIO2FB	プリミティブ : Feedback Clock Buffer.
BUFPLL	プリミティブ : PLL Buffer
BUFPLL_MCB	プリミティブ : PLL Buffer for the Memory Controller Block
DCM_CLKGEN	プリミティブ : Digital Clock Manager.
DCM_SP	プリミティブ : Digital Clock Manager
PLL_BASE	プリミティブ : Basic Phase Locked Loop Clock Circuit

コンフィギュレーション/BSCAN コンポーネント

デザイン エLEMENT	説明
BSCAN_SPARTAN6	プリミティブ : Spartan®-6 JTAG Boundary Scan Logic Control Circuit
DNA_PORT	プリミティブ : Device DNA Data Access Port
ICAP_SPARTAN6	プリミティブ : Internal Configuration Access Port
POST_CRC_INTERNAL	プリミティブ : Post-configuration CRC error detection
STARTUP_SPARTAN6	プリミティブ : Spartan®-6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface
SUSPEND_SYNC	プリミティブ : Suspend Mode Access

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

ギガビット I/O

デザイン エLEMENT	説明
GTPA1_DUAL	プリミティブ : Dual Gigabit Transceiver

I/O コンポーネント

デザイン エLEMENT	説明
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
IOBUFDS	プリミティブ : 3-State Differential Signaling I/O Buffer with Active Low Output Enable
IODELAY2	プリミティブ : Input and Output Fixed or Variable Delay Element
IODRP2	プリミティブ : I/O Control Port
IODRP2_MCB	プリミティブ : I/O Control Port for the Memory Controller Block
ISERDES2	プリミティブ : Input SERIAL/DESerializer.
KEEPER	プリミティブ : KEEPER Symbol
OBUF	プリミティブ : Output Buffer
OBUFDS	プリミティブ : Differential Signaling Output Buffer
OBUFFT	プリミティブ : 3-State Output Buffer with Active Low Output Enable
OBUFFTDS	プリミティブ : 3-State Output Buffer with Differential Signaling, Active-Low Output Enable
OSERDES2	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
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
RAMB16BWER	プリミティブ : 16K-bit Data and 2K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers
RAMB8BWER	プリミティブ : 8K-bit Data and 1K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers

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
IDDR2	プリミティブ : Double Data Rate Input D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset
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
ODDR2	プリミティブ : Dual Data Rate Output D Flip-Flop with Optional Data Alignment, Clock Enable and Programmable Synchronous or Asynchronous Set/Reset

シフト レジスタ

デザイン エLEMENT	説明
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

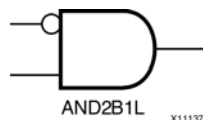
このセクションでは、このアーキテクチャで使えるデザイン エLEMENTについて説明します。デザイン エLEMENTは、アルファベット順に並べられています。

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 が反転される 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 メモ : 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

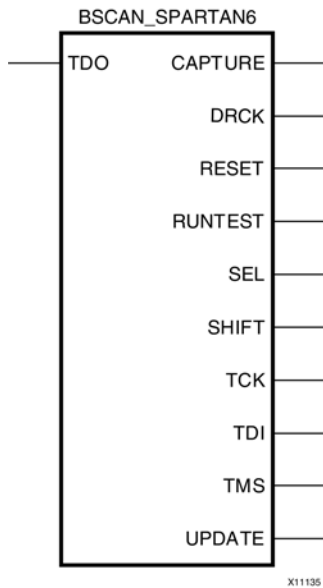
インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

BSCAN_SPARTAN6

プリミティブ : Spartan®-6 JTAG Boundary Scan Logic Control Circuit



概要

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

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

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。
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	FPGA の TCK 入力ピンの値
TDI	出力	1	TDI ピンと同じ値を出力します。
TMS	出力	1	FPGA の TMS 入力ピンの値
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 TDO ピンへのデータ入力の値が示されます。

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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_SPARTAN6: Spartan-6 JTAG Boundary-Scan Logic Access
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BSCAN_SPARTAN6_inst : BSCAN_SPARTAN6
generic map (
    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 SHIFTR 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_SPARTAN6_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BSCAN_SPARTAN6: Spartan-6 JTAG Boundary-Scan Logic Access
//           Spartan-6
// Xilinx HDL Language Template, version 11.1

BSCAN_SPARTAN6 #(
    .JTAG_CHAIN(1) // Chain number.
)
BSCAN_SPARTAN6_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 SHIFTR 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.
);

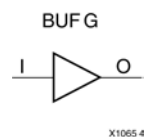
// End of BSCAN_SPARTAN6_inst instantiation
```

詳細情報

- [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFG

プリミティブ : Global Clock Buffer



概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFG_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

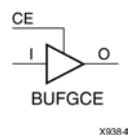
// End of BUFG_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFGCE

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



概要

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

論理表

入力		出力
I	CE	O
X	0	0
I	1	I

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFGCE_inst instantiation
```


Verilog 記述 (インスタンス化)

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

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

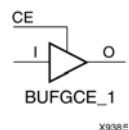
// End of BUFGCE_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGCE_1

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



概要

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

論理表

入力		出力
I	CE	O
X	0	1
I	1	I

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

BUFGCE_1_inst : BUFGCE_1
port map (
    O => O,    -- Clock buffer ouptput
    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
);

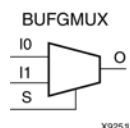
// End of BUFGCE_1_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGMUX

プリミティブ : Global Clock MUX Buffer



概要

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

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

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

論理表

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFGMUX_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

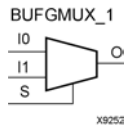
// End of BUFGMUX_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

BUFGMUX_1

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



概要

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

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

論理表

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of BUFGMUX_1_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

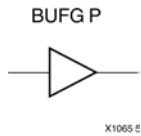
// End of BUFGMUX_1_inst instantiation
```

詳細情報

- [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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

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

BUFH

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
);

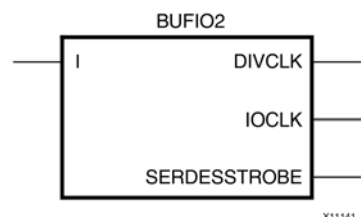
// End of BUFH_inst instantiation
```

詳細情報

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

BUFIO2

プリミティブ : Dual Clock Buffer and Strobe Pulse



概要

このプリミティブでは、1 つのクロック入力から 2 つのクロック出力およびストロブ パルスを生成します。IOCLK 出力は、入力クロックにバッファが付いたものです。DIVCLK 出力の周期およびデューティ サイクルは、属性設定によって異なります。DIVIDE_BYPASS が TRUE に設定されると、DIVCLK 出力は入力にバッファが付けられたものになり、SERDESSTROBE 出力が 1 に駆動されます。DIVIDE_BYPASS が FALSE に設定されると、DIVCLK および SERDESSTROBE 出力は 属性の設定に従って入力クロックが分周されたものになります。

ポートの説明

ポート名	タイプ	幅	機能
DIVCLK	出力	1	分周クロック
I	入力	1	クロック入力
IOCLK	出力	1	クロック出力
SERDESSTROBE	出力	1	SERDES ストロブ (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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DIVIDE	10 進数	1、2、3、4、5、6、7、8	1	DIVCLK 分周器の分周値を設定します。
DIVIDE_BYPASS	ブール代数	TRUE、FALSE	TRUE	DIVCLK のソースを分周器 (FALSE) または分周器をバイパスして I 入力 (TRUE) にします。

VHDL 記述 (インスタンスレーション)

```
-- BUFIO2: Dual Clock Buffer and Strobe Pulse
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFIO2_inst : BUFIO2
generic map (
    DIVIDE => 1,                -- Set the DIVCLK divider divide-by value.
    DIVIDE_BYPASS => TRUE,      -- DIVCLK output sourced from Divider (FALSE) or from I input, bypassing Divider
                                -- (TRUE).
    I_INVERT => FALSE,
    USE_DOUBLER => FALSE
)
port map (
    DIVCLK => DIVCLK,           -- 1-bit Output divided clock
    IOCLK => IOCLK,             -- 1-bit Output clock
    SERDESSTROBE => SERDESSTROBE, -- 1-bit Output SERDES strobe (Clock Enable)
    I => I                      -- 1-bit Clock input
);

-- End of BUFIO2_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFIO2: Dual Clock Buffer and Strobe Pulse
//           Spartan-6
// Xilinx HDL Language Template, version 11.1

BUFIO2 #(
    .DIVIDE(1),                // Set the DIVCLK divider divide-by value.
    .DIVIDE_BYPASS("TRUE"),    // DIVCLK output sourced from Divider (FALSE) or from I input, bypassing Divider
                                // (TRUE).
    .I_INVERT("FALSE"),
    .USE_DOUBLER("FALSE")
)
BUFIO2_inst (
    .DIVCLK(DIVCLK),           // 1-bit Output divided clock
    .IOCLK(IOCLK),             // 1-bit Output clock
    .SERDESSTROBE(SERDESSTROBE), // 1-bit Output SERDES strobe (Clock Enable)
    .I(I)                      // 1-bit Clock input
);

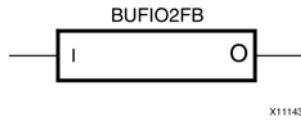
// End of BUFIO2_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFIO2FB

プリミティブ : Feedback Clock Buffer.



概要

このデザイン エLEMENTは、出力の遅延を決定する属性を持つ単純なバッファです。DIVIDE_BYPASS を TRUE に設定すると、遅延は BUFIO2 のバイパス遅延と同一になります。FALSE に設定すると、遅延は BUFIO2 と BUFIO2FB 出力の位相が揃えられるように BUFIO2 DIVCLK 出力と類似した値になります。

ポートの説明

ポート名	タイプ	幅	機能
I	入力	1	入力フィードバック クロック
O	出力	1	出力フィードバック 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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DIVIDE_BYPASS	ブール代数	TRUE、FALSE	TRUE	DIVCLK のソースを分周器 (FALSE) または分周器をバイパスして I 入力 (TRUE) にします。FALSE の場合は、CLKDIV DIVIDE も 1 に設定する必要があります。

VHDL 記述 (インスタンシエーション)

```
-- BUFIO2FB: Feedback Clock Buffer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFIO2FB_inst : BUFIO2FB
generic map (
    DIVIDE_BYPASS => TRUE      -- DIVCLK output sourced from Divider (FALSE) or from I input, bypassing Divider
                                -- (TRUE) Note: if FALSE, also need to set CLKDIV DIVIDE value to 1.
)
port map (
    O => O, -- 1-bit Output feedback clock
    I => I  -- 1-bit Feedback clock input
);

-- End of BUFIO2FB_inst instantiation
```

Verilog 記述 (インスタンシエーション)

```
// BUFIO2FB: Feedback Clock Buffer
//          Spartan-6
// Xilinx HDL Language Template, version 11.1

BUFIO2FB #(
    .DIVIDE_BYPASS("TRUE") // DIVCLK output sourced from Divider (FALSE) or from I input, bypassing Divider
                          // (TRUE) Note: if FALSE, also need to set CLKDIV DIVIDE value to 1.
)
BUFIO2FB_inst (
    .O(0), // 1-bit Output feedback clock
    .I(I)  // 1-bit Feedback clock input
);

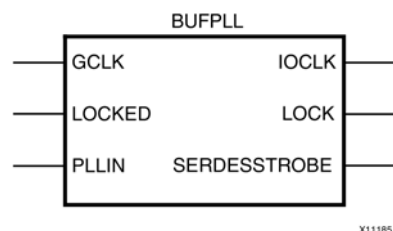
// End of BUFIO2FB_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFPLL

プリミティブ : PLL Buffer



概要

このデザイン エLEMENTは、SERDES クロックおよびストロブ パルスを生成し、また PLL の LOCKED 出力を GCLK、SERDES ストロブ、および PLL クロックと揃えるためにするために高速 IO 配線で使用されます。IOCLK 出力は、入力クロックにバッファが付いたものです。LOCK 出力はどのクロックにも揃えられていません。LOCKED 出力の機能は PLL のロック信号と同じですが、PLL がロックして BUFPLL で SERDESSTROBE 信号が適切に揃えられるまで High にならない点が異なります。

ポートの説明

ポート名	タイプ	幅	機能
GCLK	入力	1	GCLK クロック入力
IOCLK	出力	1	PLL クロック出力
LOCK	出力	1	同期化された LOCK 出力
LOCKED	入力	1	PLL から入力される LOCKED 通知
PLLIN	入力	1	PLL クロック入力
SERDESSTROBE	出力	1	SERDES ストロブ (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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DIVIDE	整数	1、2、3、4、5、6、7、8	1	SERDESSTROBE の対する PLLIN 分周器の分周値を設定します。

VHDL 記述 (インスタンスレーション)

```
-- BUFPLL: (MISSING DESCRIPTION)
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

BUFPLL_inst : BUFPLL
generic map (
    DIVIDE => 1 -- Set the PLLIN divider divide-by value for SERDESSTROBE.
)
port map (
    IOCLK => IOCLK,           -- 1-bit Output PLL clock
    LOCK => LOCK,             -- 1-bit Synchronized LOCK output
    SERDESSTROBE => SERDESSTROBE, -- 1-bit Output SERDES strobe (clock enable)
    GCLK => GCLK,             -- 1-bit GCLK clock input
    LOCKED => LOCKED,         -- 1-bit LOCKED sign from PLL input
    PLLIN => PLLIN            -- 1-bit PLL clock input
);

-- End of BUFPLL_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// BUFPLL: (MISSING DESCRIPTION)
//          Spartan-6
// Xilinx HDL Language Template, version 11.1

BUFPLL #(
    .DIVIDE(1) // Set the PLLIN divider divide-by value for SERDESSTROBE.
)
BUFPLL_inst (
    .IOCLK(IOCLK),           // 1-bit Output PLL clock
    .LOCK(LOCK),             // 1-bit Synchronized LOCK output
    .SERDESSTROBE(SERDESSTROBE), // 1-bit Output SERDES strobe (clock enable)
    .GCLK(GCLK),             // 1-bit GCLK clock input
    .LOCKED(LOCKED),         // 1-bit LOCKED sign from PLL input
    .PLLIN(PLLIN)            // 1-bit PLL clock input
);

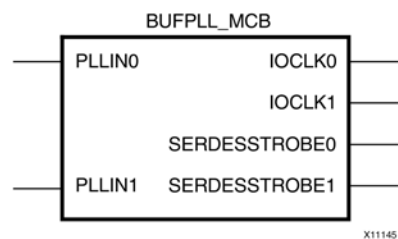
// End of BUFPLL_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

BUFPLL_MCB

プリミティブ : PLL Buffer for the Memory Controller Block



概要

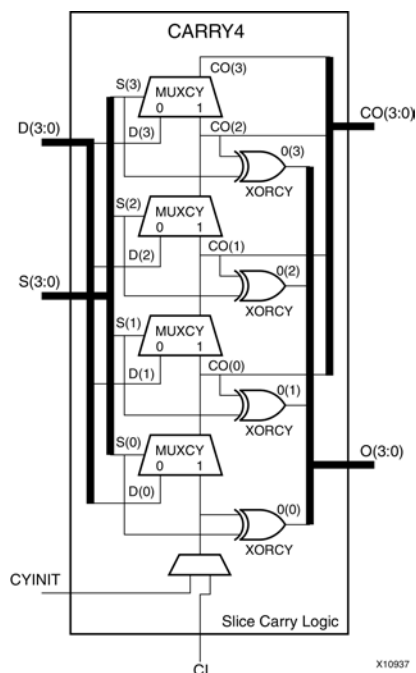
このデザイン エLEMENTは、外部メモリ インターフェイスをインプリメントするために MIG (Memory Interface Generator) コアで MCB ブロックと組み合わせて使用されます。MIG の外部での使用はサポートされていません。

詳細情報

- ・ [ザイリンクス メモリ インターフェイス ジェネレータ \(MIG\) ユーザー ガイド](#)
- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

CARRY4

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



概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of CARRY4_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

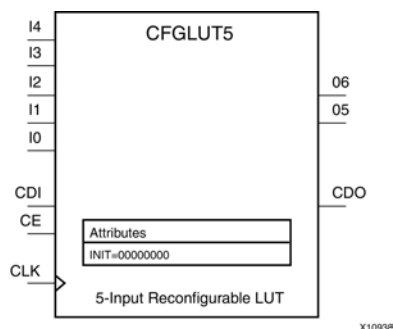
// End of CARRY4_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

CFGLUT5

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



概要

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

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

ポートの説明

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

デザインの入力方法

インスタンシエーション	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

- ・ 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
);

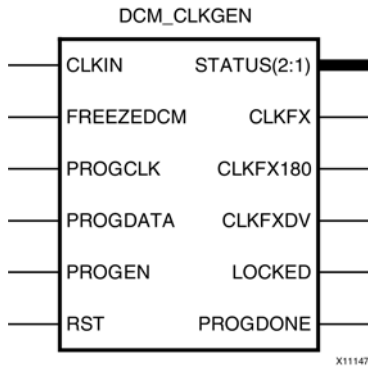
// End of CFGLUT5_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

DCM_CLKGEN

プリミティブ : Digital Clock Manager.



概要

Spartan®-6 デバイスには、8 ～ 10 個の DCM がデバイス中央列に PLL と共に配置されています。DCM (デジタル クロック マネージャ) では、柔軟性のある、完全なクロック周波数制御が供給され、クロック ネットワーク デスキュー、周波数合成、および位相シフトを含むアドバンス クロック機能を使用できます。

ポートの説明

ポート名	タイプ	幅	機能
CLKFX	出力	1	生成された出力クロック
CLKFXDV	出力	1	CLKFXDV_DIV 属性の分周値により分周された出力クロック。 メモ : CLKFX および CLKFXDV の位相は揃えられません。
CLKFX180	出力	1	CLKFX から 180 度位相がシフトされた出力クロック
CLKIN	入力	1	DCM にソース クロックを供給します。 フリーランニング オシレータ モードの場合は、DCM がロックしてフリーズしたら動作クロックの接続を解除することができます。その他のモードでは、フリーランニング クロックを供給し続ける必要があります。
FREEZEDCM	入力	1	CLKIN 入力失われたときのタップ調整の変動を防ぎます。
LOCKED	出力	1	DCM 操作が開始可能であることを示す DCM の同期出力
PROGCLK	入力	1	M および/または D リコンフィギュレーションのクロック入力
PROGDATA	入力	1	DCM の M (通倍) および/または D (分周) 値の再プログラム用データを供給するシリアル データ入力。この入力は、PROGCLK 入力に同期させる必要があります。
PROGDONE	出力	1	M または D 値の再プログラミングが正しく完了したことを示すアクティブ High 出力
PROGEN	入力	1	M/D 値を再プログラムする アクティブ High のイネーブル入力。この入力は、PROGCLK 入力に同期させる必要があります。
RST	入力	1	DCM 回路をリセットします。RST 信号は、アクティブ High の非同期リセットです。RST 信号をアサートすると、すべての DCM 出力 (LOCKED 信号、ステータス信号、出力クロック) がソースクロックの 4 サイクル以内に Low になります。リセットは非同期であるため、ディアサート中にクロックの最後のサイクルが短いパルスになったり、デューティ サイクルが崩れたり、クロック間のスキューが調整されなる可能性があります。このため、デバイスをリコンフィギュレーションする場合や入力周波数を変更する場合は、RST ピンを使用する必要があります。RST 信号をディアサートすると、次の CLKIN サイクルに同期してクロックをロック状態にするプロセスが開始しま

ポート名	タイプ	幅	機能
			す。DCM がリセット後に正しくロックされるようにするには、CLKIN 信号が供給され、3 クロック サイクル以上安定するまで RST 信号をディassertしておく必要があります。どのデザインでも、クロックが安定するまで DCM をリセットに保持する必要があります。コンフィギュレーションでは、GWE が解除されるまで DCM は自動的にリセット状態に保持されます。GSR が解除されたときにクロックが安定していれば、コンフィギュレーション後に DCM をリセットする必要はありません。
STATUS[2:1]	出力	2	クロック ステータス出力 <ul style="list-style-type: none"> STATUS[1] : CLKIN の停止 STATUS[2] : CLKFX または CLKFX180 の停止

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
CLKFX_DIVIDE	整数	1 ~ 256	1	この値と入力周波数および CLKFX_MULTIPLY の値を組み合わせ、CLKFX および CLKFX180 の出力周波数が決定されます。
CLKFXDV_DIVIDE	整数	2、4、8、16、32	2	CLKFXDV の分周値を指定します。
CLKFX_MD_MAX	3 上位ビット 浮動小数点	0 ~ 256.000	0.000	DCM_CLKGEN を変数 M および D 値で使用する とき、スタティック タイミング解析中に使用する M と D の最大比率を指定します。
CLKFX_MULTIPLY	整数	2 ~ 256	4	この値と入力周波数および CLKFX_DIVIDE の 値を組み合わせ、CLKFX および CLKFX180 の出力周波数が決定されます。
CLKIN_PERIOD	文字列	0 ビット文字列	10.0	CLKFX/CLKFX180 出力のために DCM の調整 を補助しロックにかかる時間を短縮するために使用 するソース クロックの周期を指定します。
DFS_BANDWIDTH	文字列	OPTIMIZED、 HIGH、 LOW	OPTIMIZED	プロセス、電圧、温度 (PVT) に対する DCM の 周波数調整バンド幅を指定します。
PROG_MD_ BANDWIDTH	文字列	OPTIMIZED、 HIGH、 LOW	OPTIMIZED	M および D 値のプログラミング変更に対する DCM の周波数調整バンド幅を指定します。
SPREAD_SPECTRUM	文字列	NONE、 CUSTOM、 LOW_EMISSION_ 24_DOWN_SPREAD、 VIDEO_LINK_75_ CENTER_SPREAD、 VIDEO_LINK_90_ CENTER_SPREAD	NONE	スペクトラム拡散でサポートされるモードを指定し ます。周波数ホッピングを実現するには、適切 な IP と共に使用する必要があります。
STARTUP_WAIT	ブール代数	FALSE、TRUE	FALSE	コンフィギュレーション DONE 信号を DCM LOCKED 信号が High になるまで遅らせます。

VHDL 記述 (インスタンスレーション)

```

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

DCM_CLKGEN_inst : DCM_CLKGEN
generic map (
    CLKFXDV_DIVIDE => 2,          -- Specifies divide value for CLKFXDV.
    CLKFX_DIVIDE   => 1,          -- This value in conjunction with the input frequency and
                                -- CLKFX_MULTIPLY value determine the resultant output frequency for
                                -- the CLKFX and CLKFX180 outputs.
    CLKFX_MD_MAX   => 0.0,        -- When using the DCM_CLKGEN with variable M and D values, this would
                                -- specify the maximum ratio of M and D used during static timing
                                -- analysis to ensure proper timing of the DCM output.
    CLKFX_MULTIPLY => 4,          -- This value in conjunction with the input frequency and CLKFX_DIVIDE
                                -- value determine the resultant output frequency for the CLKFX and
                                -- CLKFX180 outputs.
    CLKIN_PERIOD   => "10.0",    -- This attribute specifies the source clock period which is used to
                                -- help the DCM adjust for the optimum CLKFX/CLKFX180 outputs and also
                                -- result in faster locking time.
    DFS_BANDWIDTH  => "OPTIMIZED", -- Specifies the frequency adjust bandwidth of the DCM due to PVT
    PROG_MD_BANDWIDTH => "OPTIMIZED", -- Specifies the frequency adjust bandwidth of the DCM due to change of
                                -- programming of the M and/o D values.
    SPREAD_SPECTRUM => "NONE",    -- Specify a supported mode for Spread Spectrum. Must be used in
                                -- conjunction with the appropriate IP in order to fully realize the
                                -- frequency hopping.
    STARTUP_WAIT   => FALSE      -- Delays configuration DONE signal until DCM LOCKED signal goes high.
)
port map (
    CLKFX => CLKFX,              -- 1-bit Generated output clock.
    CLKFX180 => CLKFX180,        -- 1-bit Generated output clock 180 degree out of phase from CLKFX.
    CLKFXDV => CLKFXDV,          -- 1-bit Divided output clock, Divide value derived from CLKFXDV_DIV attribute.
                                -- There is no phase alignment between CLKFX and CLKFXDV.

    LOCKED => LOCKED,            -- 1-bit Synchronous output from the DCM that provides the user with an
                                -- indication the DCM is ready for operation.

    PROGDONE => PROGDONE,        -- 1-bit Active high output to indicate the successful re-programming of an M
                                -- and/or D value.

    STATUS => STATUS,            -- 2-bit Clock Status lines.
    CLKIN => CLKIN,              -- 1-bit The source clock (CLKIN) input pin provides the source clock to the DCM.
                                -- In the case of Free-running oscillator mode, running clock needs to be
                                -- connected until DCM is locked and DCM is frozen, then clock can be removed. In
                                -- the other modes, a free running clock needs to be provided and remain.

    FREEZEDCM => FREEZEDCM,      -- 1-bit Prevents tap adjustment drift in the event of a lost CLKIN input
    PROGCLK => PROGCLK,          -- 1-bit Clock input for M and/or D reconfiguration.
    PROGDATA => PROGDATA,        -- 1-bit Serial data input to supply information for the reprogramming of M
                                -- and/or D values of the DCM. This input must be applied synchronous to the
                                -- PROGCLK input.

    PROGEN => PROGEN,            -- 1-bit Active high enable input for the reprogramming of M/D values. This input
                                -- must be applied synchronous to the PROGCLK input.

    RST => RST                   -- 1-bit Reset pin
);

-- End of DCM_CLKGEN_inst instantiation

```


Verilog 記述 (インスタンス化)

```
// DCM_CLKGEN: Digital Clock Manager
//          Spartan-6
// Xilinx HDL Language Template, version 11.1

DCM_CLKGEN #(
    .CLKFXDV_DIVIDE(2),           // Specifies divide value for CLKFXDV.
    .CLKFX_DIVIDE(1),            // This value in conjunction with the input frequency and CLKFX_MULTIPLY
                                // value determine the resultant output frequency for the CLKFX and
                                // CLKFX180 outputs.
    .CLKFX_MD_MAX(0.0),          // When using the DCM_CLKGEN with variable M and D values, this would
                                // specify the maximum ratio of M and D used during static timing
                                // analysis to ensure proper timing of the DCM output.
    .CLKFX_MULTIPLY(4),           // This value in conjunction with the input frequency and CLKFX_DIVIDE
                                // value determine the resultant output frequency for the CLKFX and
                                // CLKFX180 outputs.
    .CLKIN_PERIOD("10.0"),       // This attribute specifies the source clock period which is used to
                                // help the DCM adjust for the optimum CLKFX/CLKFX180 outputs and also
                                // result in faster locking time.
    .STARTUP_WAIT("FALSE")       // Delays configuration DONE signal until DCM LOCKED signal goes high.
)
DCM_CLKGEN_inst (
    .CLKFX(CLKFX),               // 1-bit Generated output clock.
    .CLKFX180(CLKFX180),         // 1-bit Generated output clock 180 degree out of phase from CLKFX.
    .CLKFXDV(CLKFXDV),           // 1-bit Divided output clock, Divide value derived from CLKFXDV_DIV attribute.
                                // There is no phase alignment between CLKFX and CLKFXDV.

    .LOCKED(LOCKED),             // 1-bit Synchronous output from the DCM that provides the user with an indication
                                // the DCM is ready for operation.

    .PROGDONE(PROGDONE),         // 1-bit Active high output to indicate the successful re-programming of an M
                                // and/or D value.

    .STATUS(STATUS),             // 2-bit Clock Status lines.
    .CLKIN(CLKIN),               // 1-bit The source clock (CLKIN) input pin provides the source clock to the DCM.
                                // In the case of Free-running oscillator mode, running clock needs to be
                                // connected until DCM is locked and DCM is frozen, then clock can be removed. In
                                // the other modes, a free running clock needs to be provided and remain.

    .FREEZEDCM(FREEZEDCM),       // 1-bit Prevents tap adjustment drift in the event of a lost CLKIN input
    .PROGCLK(PROGCLK),           // 1-bit Clock input for M and/or D reconfiguration.
    .PROGDATA(PROGDATA),         // 1-bit Serial data input to supply information for the reprogramming of M and/or
                                // D values of the DCM. This input must be applied synchronous to the PROGCLK
                                // input.

    .PROGEN(PROGEN),             // 1-bit Active high enable input for the reprogramming of M/D values. This input
                                // must be applied synchronous to the PROGCLK input.

    .RST(RST)                    // 1-bit Reset pin
);

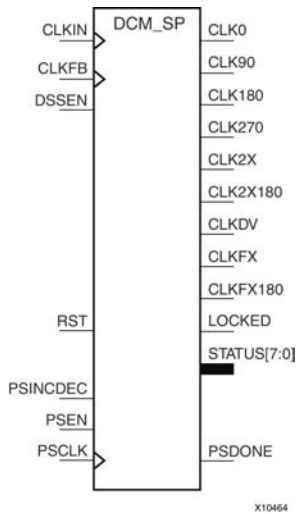
// End of DCM_CLKGEN_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

DCM_SP

プリミティブ : Digital Clock Manager



概要

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

ポートの説明

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

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

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
CLKDV_DIVIDE	1 上位ビット FLOAT	2.0、1.5、2.5、3.0、 3.5、4.0、4.5、5.0、 5.5、6.0、6.5、7.0、 7.5、8.0、9.0、10.0、 11.0、12.0、13.0、 14.0、15.0、16.0	2.0	CLKDLL、CLKDLLE、CLKDLLHF、 DCM_SP のクロック分周器 (CLKDV 出力) の分周比を指定
CLK_FEEDBACK	文字列	1X、2X、NONE	1X	DCM フィードバック モードを定義します。 <ul style="list-style-type: none"> 1X : CLK0 をフィードバックにします。 2X : CLK2X をフィードバックにします。
CLKFX_DIVIDE	整数	1 ~ 32	1	CLKFX 出力の分周比を指定
CLKFX_MULTIPLY	整数	2 ~ 32	4	CLKFX 出力の通倍比を指定
CLKIN_DIVIDE_BY_2	ブール代数	FALSE、TRUE	FALSE	CLKIN を 2 で分周
CLKIN_PERIOD	文字列	0 ビット文字列	10.0	CLKIN 入力への入力周期を指定 (ns)

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

VHDL 記述 (インスタンス化)

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;

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

DCM_SP_inst : DCM_SP
generic map (
    CLKDV_DIVIDE => 2.0,          -- Specifies the extent to which the CLKDLL, CLKDLLE, CLKDLLHF, or
    -- DCM_SP clock divider (CLKDV output) is to be frequency divided.
    CLKFX_DIVIDE => 1,           -- Specifies the frequency divider value for the CLKFX output.
    CLKFX_MULTIPLY => 4,         -- Specifies the frequency multiplier value for the CLKFX output.
    CLKIN_DIVIDE_BY_2 => FALSE,  -- Enables CLKIN divide by two features.
    CLKIN_PERIOD => "10.0",      -- Specifies the input period to the DCM_SP CLKIN input in ns.
    CLKOUT_PHASE_SHIFT => "NONE", -- This attribute specifies the phase shift mode. NONE = No phase
    -- shift capability. Any set value has no effect. FIXED = DCM
    -- outputs are a fixed phase shift from CLKIN. Value is specified
    -- by PHASE_SHIFT attribute. VARIABLE = Allows the DCM outputs to
    -- be shifted in a positive and negative range relative to CLKIN.
    -- Starting value is specified by PHASE_SHIFT.
    CLK_FEEDBACK => "1X",        -- Defines the DCM feedback mode. 1X: CLK0 as feedback 2X: CLK2X
    -- as feedback.
    DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS", -- Sets configuration bits affecting the clock delay alignment
    -- between the DCM_SP output clocks and an FPGA clock input pin.
    DLL_FREQUENCY_MODE => "LOW", -- AUTO mode allows DLL to do automatic frequency search to decide
    -- whether DLL will operate in LOW or HIGH mode. This is a legacy
    -- attribute where the high and low value has no affect, it is
    -- always in auto mode.
    DSS_MODE => "NONE",
    DUTY_CYCLE_CORRECTION => TRUE, -- Corrects the duty cycle of the CLK0, CLK90, CLK180, and CLK270
    -- outputs.
    PHASE_SHIFT => 0,           -- Defines the amount of fixed phase shift from -255 to 255
    STARTUP_WAIT => FALSE      -- Delays configuration DONE until DCM LOCK.
)
port map (
    CLK0 => CLK0,               -- 1-bit Same frequency as CLKIN, 0 degree phase shift.
    CLK180 => CLK180,           -- 1-bit Same frequency as CLKIN, 180 degree phase shift.
    CLK270 => CLK270,           -- 1-bit Same frequency as CLKIN, 180 degree phase shift.
    CLK2X => CLK2X,             -- 1-bit Two times CLKIN frequency clock, aligned with CLK0.
    CLK2X180 => CLK2X180,      -- 1-bit 180 degree shifted version of the CLK2X clock.
    CLK90 => CLK90,             -- 1-bit Same frequency as CLKIN, 90 degree phase shift.
    CLKDV => CLKDV,             -- 1-bit Divided version of CLK0. Divide value is programmable.
    CLKFX => CLKFX,             -- 1-bit Digital Frequency Synthesizer output (DFS).
    CLKFX180 => CLKFX180,      -- 1-bit 180 degree shifted version of the CLKFX clock.
    LOCKED => LOCKED,          -- 1-bit Signal indicating when the DCM has LOCKed.
    PSDONE => PSDONE,           -- 1-bit Output signal that indicates variable phase shift is done.
    STATUS => STATUS,           -- 8-bit DCM Status Bits
    CLKFB => CLKFB,             -- 1-bit Feedback clock input to DCM. The feedback input is required unless the DFS
    -- is used stand-alone. The source of CLKFB must be CLK0 or CLK2X output from the
    -- DCM.
    CLKIN => CLKIN,             -- 1-bit Clock input for the DCM.
    DSEN => DSEN,
    PSCLK => PSCLK,             -- 1-bit Phase shift clock input. The PSCLK input pin provides the source clock for
    -- the DCM phase shift.
    PSEN => PSEN,               -- 1-bit Variable Phase Shift enable signal, synchronous with PSCLK.
    PSINCDEC => PSINCDEC,      -- 1-bit The phase shift increment/decrement (PSINCDEC) input signal must be
    -- synchronous with PSCLK. The PSINCDEC signal is used to increment or decrement
    -- the phase shift factor when PSEN is activated. The PSINCDEC is asserted HIGH for
    -- increment and LOW for decrement.
    RST => RST                  -- 1-bit The reset input pin (RST) resets the DCM circuitry. The RST signal is an
    -- active HIGH asynchronous reset.
);
```

```
-- End of DCM_SP_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

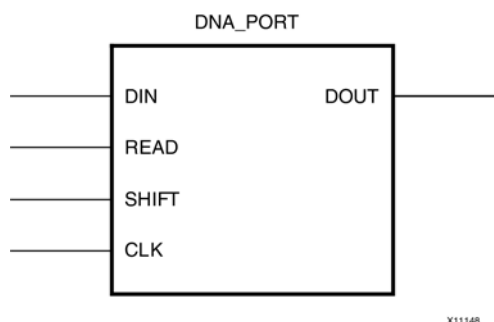
// End of DCM_SP_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA クロック リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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

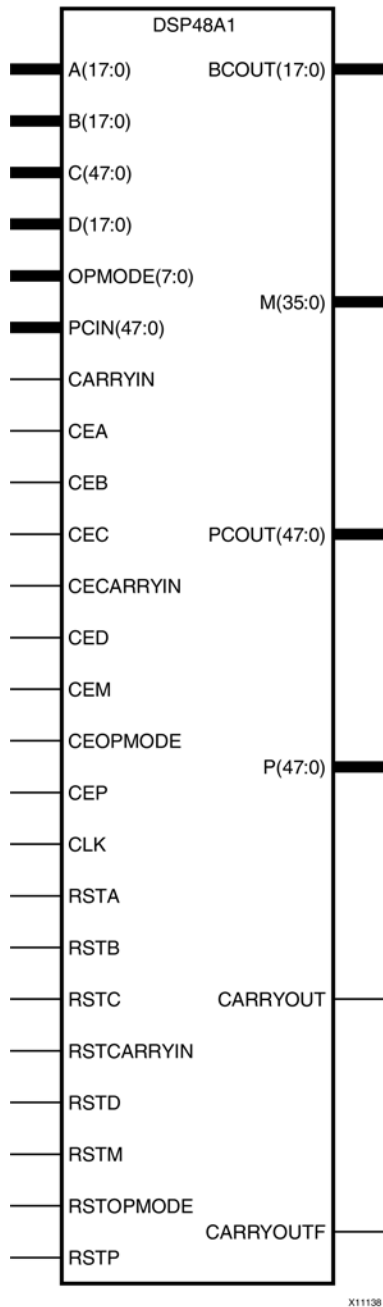
属性	タイプ	値	デフォルト	説明
SIM_DNA_VALUE	16 進数	57'h00000000 0000000 ~ 57'h1fffffffff	57'h00000000 0000000	シミュレーションに使用する DNA 値を指定 (実際の値は使用デバイス特有)

詳細情報

- ・ [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)
- ・ [Spartan-6 FPGA の資料 \(ユーザー ガイドおよびデータシート\)](#)

DSP48A1

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



概要

このデザイン エLEMENTは、柔軟性が高い多用途のハード IP ブロックで、多くの DSP アルゴリズムで見られる小型で高速な演算処理を作成できます。このブロックでは、コンフィギュレーション可能な 18 ビットの加減算器に、18 X 18 符号付き乗算器、48 ビットの加減/アキュムレータが順に付けられています。このブロックには、コンフィギュレーション可能なパイプライン レジスタが数個含まれているため、追加されるレイテンシをトレードオフにする高速クロックを実現できます。OpMode ピンでは、ブロック操作を 1 クロック サイクルから次サイクルに変更でき、デザインに含まれる複数の演算ファンクションに 1 つのブロックを使用できます。さらに、複数の DSP48A1 ブロックをカスケード接続して大型の乗算および加算ファンクションを作成できます。

ポートの説明

ポート名	方向	幅	機能
データ ポート			
A	入力	18	OPMODE[1:0] の値に従い乗算器または後置加減算器に送られる 18 ビット データ入力
B	入力	18	OPMODE[3:0] の値に従い乗算器、前置加減算器、または後置加減算器に送られる 18 ビット データ入力
C	入力	48	後置加減算器への 48 ビット入力
D	入力	18	前置加減算器への 18 ビット入力
CARRYIN	入力	1	後置加減算器への外部キャリー入力。別の DSP48A1 ブロックの CARRYOUT ピンにのみ接続します。
P	出力	48	プライマリ データ出力
CARRYOUTF	出力	1	ファブリックに配線可能な後置加減算器の外部キャリー出力信号。
CARRYOUT	出力	1	後置加減算器の外部キャリー出力信号。別の DSP48A1 ブロックの CARRYIN ピンにのみ接続します。
制御入力			
CLK	入力	1	DSP48A1 クロック
OPMODE	入力	8	DSP48A1 の演算処理を選択する制御入力
リセット/クロック イネーブル入力			
RSTA	入力	1	A ポートレジスタ (A0REG=1 または A1REG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTB	入力	1	B ポートレジスタ (B0REG=1 または B1REG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTD	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTM	入力	1	乗算器レジスタ (MREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。

ポート名	方向	幅	機能
RSTP	入力	1	P 出力レジスタ (PREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTCARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG =1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTOPMODE	入力	1	OPMODE レジスタ (OPMODEREG=1) のアクティブ High のリセットで、使用しない場合は論理値を 0 にします。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
CEA	入力	1	A ポートレジスタ (A0REG=1 または A1REG=1) のアクティブ High のクロック イネーブルで、使用しない場合と A0REG=1 または A1REG=1 の場合は論理値を 1 に、A0REG=0 または A1REG=0 の場合は論理値を 0 にします。
CEB	入力	1	B ポートレジスタ (B0REG=1 または B1REG=1) のアクティブ High のクロック イネーブルで、使用しない場合と B0REG=1 または B1REG=1 の場合は論理値を 1 に、B0REG=0 または B1REG=0 の場合は論理値を 0 にします。
CEC	入力	1	C ポートレジスタ (CREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と CREG=1 の場合は論理値を 1 に、CREG=0 の場合は論理値を 0 にします。
CED	入力	1	D ポートレジスタ (DREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と DREG=1 の場合は論理値を 1 に、DREG=0 の場合は論理値を 0 にします。
CEM	入力	1	乗算レジスタ (MREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と MREG=1 の場合は論理値を 1 に、MREG=0 の場合は論理値を 0 にします。
CEP	入力	1	出力ポートレジスタ (PREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と PREG=1 の場合は論理値を 1 に、PREG=0 の場合は論理値を 0 にします。
CECARRYIN	入力	1	キャリー入力レジスタ (CARRYINREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と CARRYINREG=1 の場合は論理値を 1 に、CARRYINREG=0 の場合は論理値を 0 にします。
CEOPMODE	入力	1	OPMODE 入力レジスタ (OPMODEREG=1) のアクティブ High のクロック イネーブルで、使用しない場合と OPMODEREG=1 の場合は論理値を 1 に、OPMODEREG=0 の場合は論理値を 0 にします。
カスケード ポート			
PCIN	入力	48	ポート P のカスケード入力で、カスケード接続されている上位の DSP48A1 の PCOUT に接続します。使用しない場合は、ポートをすべてゼロにします。
PCOUT	出力	48	ポート P のカスケード出力で、カスケード接続されている下位の DSP48A1 の PCIN に接続します。使用しない場合は未接続にします。
BCOUT	出力	18	ポート B のカスケード出力で、カスケード接続されている下位の DSP48A1 の B ポートに接続します。使用しない場合は未接続にします。

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
A0REG	整数	0、1	0	1 段目の A 入力にレジスタを付けるかを指定します。
A1REG	整数	1、0	1	2 段目の A 入力にレジスタを付けるかを指定します。
B0REG	整数	0、1	0	1 段目の B 入力にレジスタを付けるかを指定します。
B1REG	整数	1、0	1	2 段目の B 入力にレジスタを付けるかを指定します。
CARRYINREG	整数	1、0	1	キャリー出力パイプライン レジスタの使用の有無を指定します。使用する場合は 1 に設定します。レジスタが付けられた出力には CARRYOUT および CARRYOUTF があります。
CARRYINSEL	文字列	CARRYIN、OPMODE5	OPMODE5	後置加減算器のキャリー入力信号を別の DSP48A1 の CARRYOUT ピンに接続されている CARRYIN ピンから送るか、OPMODE[5] 入力を使用して FPGA から直接制御するか指定します。
CARRYOUTREG	整数	1、0	1	出力パイプライン レジスタ。イネーブル = 1/ディスエーブル = 0
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 前置加算器入力パイプラインレジスタの使用の有無を指定します。使用する場合は 1 に設定します。
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 入力パイプライン レジスタの使用の有無を指定します。使用する場合は 1 に設定します。
PREG	整数	1、0	1	C 入力にレジスタを付けるかを指定します。
RSTTYPE	文字列	SYNC、ASYNC	SYNC	リセットすべてを同期リセットにするか非同期リセットにするかを指定します。タイミングの向上と回路の安定性の点から、非同期リセットが必要でない限り常に SYNC に設定してください。

VHDL 記述 (インスタンスエーション)

```

-- DSP48A1: Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

DSP48A1_inst : DSP48A1
generic map (
  AOREG => 0,           -- Enable=1/disable=0 first stage A input pipeline register
  A1REG => 1,           -- Enable=1/disable=0 second stage A input pipeline register
  BOREG => 0,           -- Enable=1/disable=0 first stage B input pipeline register
  B1REG => 1,           -- Enable=1/disable=0 second stage B input pipeline register
  CARRYINREG => 1,      -- Enable=1/disable=0 CARRYIN input pipeline register
  CARRYINSEL => "OPMODE5", -- Specify carry-in source, "CARRYIN" or "OPMODE5"
  CARRYOUTREG => 1,     -- Enable=1/disable=0 CARRYOUT output pipeline register
  CREG => 1,            -- Enable=1/disable=0 C input pipeline register
  DREG => 1,            -- Enable=1/disable=0 D pre-adder input pipeline register
  MREG => 1,            -- Enable=1/disable=0 M pipeline register
  OPMODEREG => 1,       -- Enable=1/disable=0 OPMODE input pipeline registers
  PREG => 1,            -- Enable=1/disable=0 P output pipeline register
  RSTTYPE => "SYNC"     -- Specify reset type, "SYNC" or "ASYN"
)
port map (
  -- Cascade Ports: 18-bit (each) Cascade Ports
  BCOUT => BCOUT,       -- 18-bit B port cascade output
  PCOUT => PCOUT,       -- 48-bit cascade output
  -- Data Ports: 1-bit (each) Data Ports
  CARRYOUT => CARRYOUT, -- 1-bit carry output
  CARRYOUTF => CARRYOUTF, -- 1-bit fabric carry output
  M => M,               -- 36-bit fabric multiplier data output
  P => P,               -- 48-bit output
  -- Cascade Ports: 48-bit (each) Cascade Ports
  PCIN => PCIN,         -- 48-bit P cascade input
  -- Control Inputs: 1-bit (each) Control Inputs
  CLK => CLK,           -- 1-bit Clock input
  OPMODE => OPMODE,     -- 8-bit operation mode input
  -- Data Ports: 18-bit (each) Data Ports
  A => A,               -- 18-bit A data input
  B => B,               -- 18-bit B data input (can be connected to fabric or BCOUT of adjacent DSP48A1)
  C => C,               -- 48-bit C data input
  CARRYIN => CARRYIN,   -- 1-bit carry input signal
  D => D,               -- 18-bit B pre-adder data input
  -- Reset/Clock Enable Inputs: 1-bit (each) Reset/Clock Enable Inputs
  CEA => CEA,           -- 1-bit active high clock enable input for A input registers
  CEB => CEB,           -- 1-bit active high clock enable input for B input registers
  CEC => CEC,           -- 1-bit active high clock enable input for C input registers
  CECARRYIN => CECARRYIN, -- 1-bit active high clock enable input for CARRYIN registers
  CED => CED,           -- 1-bit active high clock enable input for D input registers
  CEM => CEM,           -- 1-bit active high clock enable input for multiplier registers
  CEOPMODE => CEOPMODE, -- 1-bit active high clock enable input for OPMODE registers
  CEP => CEP,           -- 1-bit active high clock enable input for P output registers
  RSTA => RSTA,         -- 1-bit reset input for A input pipeline registers
  RSTB => RSTB,         -- 1-bit reset input for B input pipeline registers
  RSTC => RSTC,         -- 1-bit reset input for C input pipeline registers
  RSTCARRYIN => RSTCARRYIN, -- 1-bit reset input for CARRYIN input pipeline registers
  RSTD => RSTD,         -- 1-bit reset input for D input pipeline registers
  RSTM => RSTM,         -- 1-bit reset input for M pipeline registers
  RSTOPMODE => RSTOPMODE, -- 1-bit reset input for OPMODE input pipeline registers
  RSTP => RSTP         -- 1-bit reset input for P pipeline registers
);

-- End of DSP48A1_inst instantiation

```

Verilog 記述 (インスタンス化)

```
// DSP48A1: Multi-Functional, Cascadable, 48-bit Output, Arithmetic Block
//          Spartan-6
// Xilinx HDL Language Template, version 11.1

DSP48A1 #(
    .AOREG(0),           // Enable=1/disable=0 first stage A input pipeline register
    .AIREG(1),           // Enable=1/disable=0 second stage A input pipeline register
    .BOREG(0),           // Enable=1/disable=0 first stage B input pipeline register
    .BIREG(1),           // Enable=1/disable=0 second stage B input pipeline register
    .CARRYINREG(1),      // Enable=1/disable=0 CARRYIN input pipeline register
    .CARRYINSEL("OPMODE5"), // Specify carry-in source, "CARRYIN" or "OPMODE5"
    .CARRYOUTREG(1),     // Enable=1/disable=0 CARRYOUT output pipeline register
    .CREG(1),            // Enable=1/disable=0 C input pipeline register
    .DREG(1),            // Enable=1/disable=0 D pre-adder input pipeline register
    .MREG(1),            // Enable=1/disable=0 M pipeline register
    .OPMODEREG(1),       // Enable=1/disable=0 OPMODE input pipeline registers
    .PREG(1),            // Enable=1/disable=0 P output pipeline register
    .RSTTYPE("SYNC")     // Specify reset type, "SYNC" or "ASYN"
)
DSP48A1_inst (
    // Cascade Ports: 18-bit (each) Cascade Ports
    .BCOUT(BCOUT),        // 18-bit B port cascade output
    .PCOUT(PCOUT),        // 48-bit cascade output
    // Data Ports: 1-bit (each) Data Ports
    .CARRYOUT(CARRYOUT),  // 1-bit carry output
    .CARRYOUTF(CARRYOUTF), // 1-bit fabric carry output
    .M(M),                // 36-bit fabric multiplier data output
    .P(P),                // 48-bit output
    // Cascade Ports: 48-bit (each) Cascade Ports
    .PCIN(PCIN),          // 48-bit P cascade input
    // Control Inputs: 1-bit (each) Control Inputs
    .CLK(CLK),            // 1-bit Clock input
    .OPMODE(OPMODE),      // 8-bit operation mode input
    // Data Ports: 18-bit (each) Data Ports
    .A(A),                // 18-bit A data input
    .B(B),                // 18-bit B data input (can be connected to fabric or BCOUT of adjacent DSP48A1)
    .C(C),                // 48-bit C data input
    .CARRYIN(CARRYIN),    // 1-bit carry input signal
    .D(D),                // 18-bit B pre-adder data input
    // Reset/Clock Enable Inputs: 1-bit (each) Reset/Clock Enable Inputs
    .CEA(CEA),            // 1-bit active high clock enable input for A input registers
    .CEB(CEB),            // 1-bit active high clock enable input for B input registers
    .CEC(CEC),            // 1-bit active high clock enable input for C input registers
    .CECARRYIN(CECARRYIN), // 1-bit active high clock enable input for CARRYIN registers
    .CED(CED),            // 1-bit active high clock enable input for D input registers
    .CEM(CEM),            // 1-bit active high clock enable input for multiplier registers
    .CEOPMODE(CEOPMODE),  // 1-bit active high clock enable input for OPMODE registers
    .CEP(CEP),            // 1-bit active high clock enable input for P output registers
    .RSTA(RSTA),          // 1-bit reset input for A input pipeline registers
    .RSTB(RSTB),          // 1-bit reset input for B input pipeline registers
    .RSTC(RSTC),          // 1-bit reset input for C input pipeline registers
    .RSTCARRYIN(RSTCARRYIN), // 1-bit reset input for CARRYIN input pipeline registers
    .RSTD(RSTD),          // 1-bit reset input for D input pipeline registers
    .RSTM(RSTM),          // 1-bit reset input for M pipeline registers
    .RSTOPMODE(RSTOPMODE), // 1-bit reset input for OPMODE input pipeline registers
    .RSTP(RSTP),          // 1-bit reset input for P pipeline registers
);

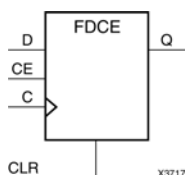
// End of DSP48A1_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)
- ・ [Spartan-6 FPGA DSP48A1 スライス ユーザー ガイド](#)

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 (インスタンスレーション)

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

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

-- FDCE: 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 (インスタンスレーション)

```
// 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
);

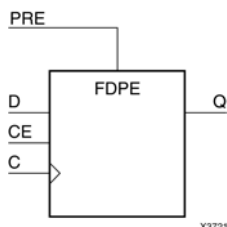
// End of FDCE_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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 に切り替わるときに D 入力の値がフリップフロップにロードされます。CE が Low の場合、クロック遷移は無視されます。

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

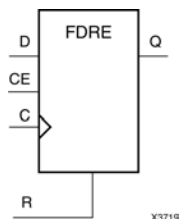
属性	タイプ	値	デフォルト	説明
INIT	2 進数	1	1	コンフィギュレーション後の Q 出力の初期値を指定。 Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDPE の場合、INIT 値は 1 である必要があります。0 に設定する場合は、この動作を表す非同期回路を作成する必要がありますが、ザイリンクスでは推奨されていません。

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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 に切り替わるときに出力 (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 に切り替わるときに D 入力の値がフリップフロップにロードされます。

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

論理表

入力				出力
R	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

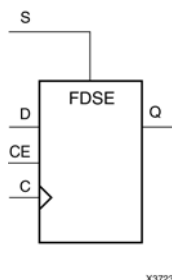
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0	0	コンフィギュレーション後の Q 出力の初期値を指定 Spartan®-6 デバイスでは、INIT 値とセットまたはリセットの極性を常に一致させる必要があります。FDRE の場合、INIT 値を 0 に設定する必要があります。1 に設定すると余分なロジックが挿入されます。

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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 に切り替わるたびに Q 出力が高にセットされます。S が Low、CE が High の場合、クロック (C) が Low から High に切り替わるたびに D 入力の値がフリップフロップにロードされます。

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	<p>コンフィギュレーション後の Q 出力の初期値を指定。</p> <p>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。</p>

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

GTPA1_DUAL

プリミティブ : Dual Gigabit Transceiver

GTPA1 DUAL

The diagram shows a vertical list of 115 pins for the GTPA1 DUAL component. Each pin is numbered and has a corresponding function label. The functions are categorized by I/O direction: Input (I), Output (O), and Bidirectional (IO). The pins are arranged in a single column, with the component name 'GTPA1 DUAL' at the top and the reference number 'X11154' at the bottom.

X11154

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 Spartan®-6 FPGA RocketIO™ GTP トランシーバが表現されます。このELEMENTの詳細は、Spartan-6 FPGA RocketIO GTP トランシーバ ユーザー ガイドを参照してください。Spartan-6 FPGA RocketIO GTX Transceiver Wizard は、GTPA1_DUAL 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

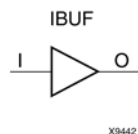
このELEMENTをインスタンスエートするには、Spartan-6 FPGA RocketIO GTX Transceiver Wizard またはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンスエートしないでください。

詳細情報

- ・ [Spartan-6 FPGA RocketIO GTP トランシーバ ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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/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;

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

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

-- End of IBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

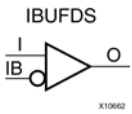
// End of IBUF_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

IBUFDS

プリミティブ : Differential Signaling Input Buffer



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

デザイン階層を保つために、すべての I/O コンポーネントを必ずデザインの最上位に配置してください。I ポートを直接デザインの最上位のマスタとなる入力ポートに、IB ポートを最上位のスレーブとなる入力ポートに、O ポートをこの入力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 規格をエレメントに割り当て

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of IBUFDS_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

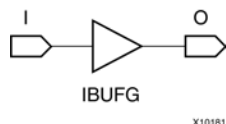
// End of IBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

IBUFG

プリミティブ : Dedicated Input Clock Buffer



概要

IBUFG は、FPGA への入力クロックをグローバル クロック配線リソースに接続するために使用する専用入力です。DCM、PLL、および BUFG への専用接続となり、デバイスのクロック遅延とジッタが最小限に抑えられます。IBUFG の入力は、グローバル クロック (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)
);

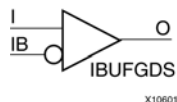
// End of IBUFG_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

IBUFGDS

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



概要

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

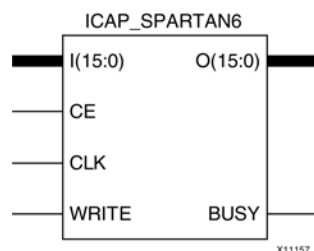
// End of IBUFGDS_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

ICAP_SPARTAN6

プリミティブ : Internal Configuration Access Port



概要

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

ポートの説明

ポート名	タイプ	幅	機能
BUSY	出力	1	アクティブ High の BUSY ステータス信号。読み出し操作でのみ使用され、書き込み中は Low に保持されます。
CE	入力	1	アクティブ Low のクロック イネーブル入力
CLK	入力	1	クロック入力
I[15:0]	入力	16	コンフィギュレーション データ入力バス
O[15:0]	出力	16	コンフィギュレーション データ出力バス
WRITE	入力	1	アクティブ Low の書き込み入力

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 (インスタンスレーション)

```
-- ICAP_SPARTAN6: Internal Configuration Access Port
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

ICAP_SPARTAN6_inst : ICAP_SPARTAN6
generic map (
)
port map (
    BUSY => BUSY,    -- 1-bit Busy output
    O => O,           -- 16-bit Configuration data output bus
    CE => CE,         -- 1-bit Active low clock enable input
    CLK => CLK,       -- 1-bit Clock input
    I => I,           -- 16-bit Configuration data input bus
    WRITE => WRITE    -- 1-bit Active low write input
);

-- End of ICAP_SPARTAN6_inst instantiation
```

Verilog 記述 (インスタンスレーション)

```
// ICAP_SPARTAN6: Internal Configuration Access Port
//                      Spartan-6
// Xilinx HDL Language Template, version 11.1

ICAP_SPARTAN6 ICAP_SPARTAN6_inst (
    .BUSY(BUSY),      // 1-bit Busy output
    .O(O),            // 16-bit Configuration data output bus
    .CE(CE),          // 1-bit Active low clock enable input
    .CLK(CLK),        // 1-bit Clock input
    .I(I),            // 16-bit Configuration data input bus
    .WRITE(WRITE)     // 1-bit Active low write input
);

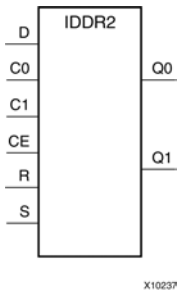
// End of ICAP_SPARTAN6_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

IDDR2

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



概要

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

論理表

入力						出力	
S	R	CE	D	C0	C1	Q0	Q1
1	X	X	X	X	X	INIT_Q0	INIT_Q1
0	1	X	X	X	X	not INIT_Q0	not INIT_Q1
0	0	0	X	X	X	変化なし	変化なし
0	0	1	D	↑	X	D	変化なし
0	0	1	D	X	↑	変化なし	D

セット/リセットは SRTYPE 値で同期に設定可能

デザインの入力方法

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

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

使用可能な属性

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

VHDL 記述 (インスタンスエーション)

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

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

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

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

-- End of IDDR2_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

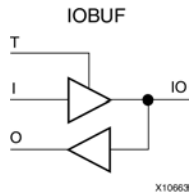
// End of IDDR2_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

IOBUF

プリミティブ : Bi-Directional Buffer



概要

このデザイン エ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
);

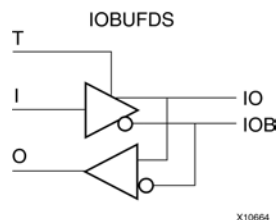
// End of IOBUF_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

IOBUFDS

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



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスエーション)

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
);

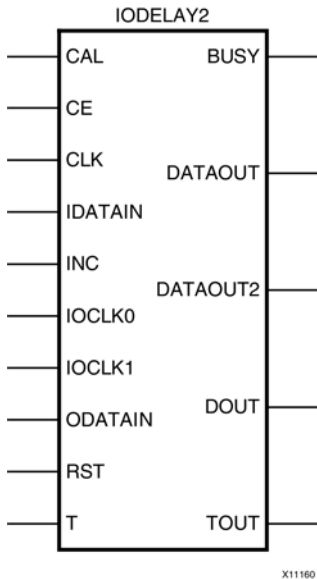
// End of IOBUFDS_inst instantiation
```

詳細情報

- [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

IODELAY2

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、Spartan®-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、チップへの入力データまたはチップからの出力データのデータ アライメントに便利で、出力遅延パスは、固定遅延でのみ使用できます。IODELAY は、FPGA の内部パスに一定の遅延または可変遅延を追加する場合にも使用できます。ただし、このように使用する場合は、入力パス遅延または出力パス遅延に関連する I/O に使用できなくなります。

ポートの説明

ポート名	タイプ	幅	機能
BUSY	出力	1	キャリブレーションが完了したこと、または同期タップ遅延の更新が完了したことを示します。
CAL	入力	1	IODELAY キャリブレーション シーケンスに入ります。このシーケンスは 8 ~ 16 クロック サイクル続きます。完了すると BUSY が Low になります。
CE	入力	1	アクティブ High になるとインクリメント/デクリメントを有効にします。
CLK	入力	1	グローバル クロック ネットワーク入力。FPGA ロジック インターコネクトのクロックです。
DATAOUT	出力	1	ILOGIC2 または ISERDES2 サイトの D ピンへの遅延データ信号。
DATAOUT2	出力	1	PCI™ アプリケーションで使用するセカンダリ遅延。
DOUT	出力	1	出力遅延として使用される場合の IOB への遅延データ信号。
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)。

ポート名	タイプ	幅	機能
INC	入力	1	インクリメント/デクリメント信号で、タップ設定の上下を 1 つずつ調節するのに使用します。
IOCLK0	入力	1	I/O クロック ネットワークからの入力。倍クロック回路が使用されていない場合のプライマリ クロック入力です (DATA_RATE を参照)。反転可能。
IOCLK1	入力	1	I/O クロック ネットワークの入力。倍クロック回路が使用されている場合のみに使用されるセカンダリ クロック入力です (DATA_RATE を参照)。反転可能。
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)。
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。
TOUT	出力	1	出力遅延として使用される場合の IOB への遅延トライステート信号。

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
COUNTER_WRAP AROUND	文字列	WRAPAROUND、STAY_AT_LIMIT	WRAPAROUND	最大または最小タップ カウントを超えたときの動作を選択します。タップ設定がインクリメントなのかデクリメントなのかによって変わります。タップ カウントが常に正しい動作範囲にあることを確認します。
DATA_RATE	文字列	SDR、DDR	SDR	シングル データ レートまたはダブル データ レートを指定します。

属性	タイプ	値	デフォルト	説明
DELAY_SRC	文字列	IO、 IDATAIN、 ODATAIN	IO	<p>IODELAY2 入力がどこから供給されるのかを示します。</p> <ul style="list-style-type: none"> ODATAIN は、遅延ソースが OSERDES2 または OLOGIC2 の ODATAIN ピンであることを示します。 IDATAIN は、遅延ソースが入力ピンであることを示します。 IO は、T (トライステート) 入力の極性に従い、信号ソースが IDATAIN と ODATAIN との間で切り替わることを示します。
IDELAY_MODE	文字列	NORMAL、 PCI	NORMAL	この属性を指定または変更しないでください。
IDELAY_TYPE	文字列	DEFAULT、 DIFF_PHASE_ DETECTOR、 FIXED、VARIABLE_FROM_ HALF_MAX、 VARIABLE_FROM_ ZERO	DEFAULT	<p>遅延のタイプを選択します。</p> <ul style="list-style-type: none"> FIXED は、固定入力遅延を使用し、このブロックにクロックは適用されなくなります。 DEFAULT は、ゼロ ホールド タイムを使用します。 VARIABLE は、インクリメント/デクリメント遅延モードを使用し、キャリブレーションを有効にします。 VARIABLE_FROM_ZERO および VARIABLE_FROM_HALF_MAX は、RST ピンがアサートされたときのリセットの動作タイプを使用します。 DIFF_PHASE_DETECTOR は、オプションの位相検出器を使用するためにマスタおよびスレーブの IODELAY2 と ISERDES2 がカスケードされるモードを使用します。
IDELAY_VALUE	整数	0 ~ 255	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します。

属性	タイプ	値	デフォルト	説明
IDELAY2_VALUE	整数	0 ~ 255	0	セカンダリ入力遅延モードの遅延タップ値を定義します。IDELAY_MODE が PCI に設定されている場合にのみアクティブです。
ODELAY_VALUE	整数	0 ~ 255	0	出力パス遅延のタップ数を指定します。
SERDES_MODE	文字列	NONE、MASTER、SLAVE	NONE	IDELAY2 が ISERDES2 と併用される場合、ISERDES2 がスタンドアロンであるか、またはカスケードされたマスタまたはスレーブであることを定義します。
SIM_TAPDELAY_VALUE	整数	10 ~ 90	75	シミュレーションのみの属性で、標準タップ遅延をシミュレーション用に別の設定に変更することができます。

VHDL 記述 (インスタンスエーション)

```
-- IDELAY2: Input and Output Fixed or Variable Delay Element
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

IDELAY2_inst : IDELAY2
generic map (
    COUNTER_WRAPAROUND => "WRAPAROUND", -- Sets behavior when tap count exceeds max or min, depending on
                                           -- whether tap setting is being incremented or decremented.

    DATA_RATE => "SDR",
    DELAY_SRC => "IO",
    -- ODATAIN indicates delay source is the ODATAIN pin from the OSERDES
    -- or OLOGIC. IDATAIN indicates the delay source is from the IDATAIN
    -- pin; one of the dedicated IOB (P/N) Pads. IO means that the signal
    -- source switches between IDATAIN and ODATAIN depending on the sense
    -- of the T (tristate) input.

    IDELAY2_VALUE => 0,
    -- Delay tap value for IDELAY Mode. Only used when IDELAY_MODE is set
    -- to PCI.

    IDELAY_MODE => "NORMAL",
    -- Delay Mode setting - PCI is for handling PCI/Extended Mode.
    -- Affects Input delays only.

    IDELAY_TYPE => "DEFAULT",
    -- Delay Type. VARIABLE refers to the customer calibrated delay mode.
    -- DEFAULT will utilize physical chip settings for best approximation
    -- of zero hold time programming. VARIABLE_FROM_ZERO and
    -- VARIABLE_FROM_HALF_MAX refer to the reset behavior.

    -- DIFF_PHASE_DETECTOR is a special mode where the master and slave
    -- IDELAY2s are cascaded for video application support.

    IDELAY_VALUE => 0,
    ODELAY_VALUE => 0,
    SERDES_MODE => "NONE",
    SIM_TAPDELAY_VALUE => 75
)
port map (
    BUSY => BUSY,
    DATAOUT => DATAOUT, -- 1-bit Delayed data signal to DDLY of ILOGIC2 or ISERDES2 sites.
    DATAOUT2 => DATAOUT2, -- 1-bit Delayed data signal to DDLY2 of ILOGIC2 or ISERDES2 sites.
    DOUT => DOUT,
    TOUT => TOUT,
    CAL => CAL,
    -- 1-bit Delayed Tristate signal to IOB.
    -- 1-bit Enter the IDELAY calibration sequence. This will last between 8 and 16
    -- GCLK cycles. Drives RDY HIGH when complete. Shared with DRP pin ADD(add).

    CE => CE,
    CLK => CLK,
    IDATAIN => IDATAIN,
    INC => INC,
    -- 1-bit Enable the Increment/Decrement signal.
    -- 1-bit CLK0 from INT connects to "Master" and CLK1 from INT connects to "Slave"
    -- 1-bit Data signal from IOB.
    -- 1-bit Increment / Decrement signal. Used for setting tap or delay length.
    -- 1-bit (each) Optionally Invertible IO clock network input.
    IOCLK0 => IOCLK0,
    IOCLK1 => IOCLK1,
    ODATAIN => ODATAIN,
    RST => RST,
    -- 1-bit Data input signal from OLOGIC or OSERDES.
    -- 1-bit Reset the IDELAY2 to either zero or 1/2 of total period. RST_VALUE

```

```

-- attribute controls this choice.

T => T -- 1-bit Tristate input signal from OLOGIC or OSERDES.
);

-- End of IODELAY2_inst instantiation

```

Verilog 記述 (インスタンス化)

```

// IODELAY2: Input and Output Fixed or Variable Delay Element
//      Spartan-6
// Xilinx HDL Language Template, version 11.1

IODELAY2 #(
    .COUNTER_WRAPAROUND("WRAPAROUND"), // Sets behavior when tap count exceeds max or min, depending on
                                         // whether tap setting is being incremented or decremented.

    .DATA_RATE("SDR"),
    .DELAY_SRC("IO"), // ODATAIN indicates delay source is the ODATAIN pin from the OSERDES
                     // or OLOGIC. IDATAIN indicates the delay source is from the IDATAIN
                     // pin; one of the dedicated IOB (P/N) Pads. IO means that the signal
                     // source switches between IDATAIN and ODATAIN depending on the sense
                     // of the T (tristate) input.

    .IDELAY2_VALUE(0), // Delay tap value for IDELAY Mode. Only used when IDELAY_MODE is set
                     // to PCI.

    .IDELAY_MODE("NORMAL"), // Delay Mode setting - PCI is for handling PCI/Extended Mode. Affects
                     // Input delays only.

    .IDELAY_TYPE("DEFAULT"), // Delay Type. VARIABLE refers to the customer calibrated delay mode.
                     // DEFAULT will utilize physical chip settings for best approximation
                     // of zero hold time programming. VARIABLE_FROM_ZERO and
                     // VARIABLE_FROM_HALF_MAX refer to the reset behavior.
                     // DIFF_PHASE_DETECTOR is a special mode where the master and slave
                     // IODELAY2s are cascaded for video application support.

    .IDELAY_VALUE(0), // Delay tap value for IDELAY Mode.
    .ODELAY_VALUE(0), // Delay tap value for ODELAY Mode.
    .SERDES_MODE("NONE"),
    .SIM_TAPDELAY_VALUE(75)
)
IODELAY2_inst (
    .BUSY(BUSY),
    .DATAOUT(DATAOUT), // 1-bit Delayed data signal to DDLY of ILOGIC2 or ISERDES2 sites.
    .DATAOUT2(DATAOUT2), // 1-bit Delayed data signal to DDLY2 of ILOGIC2 or ISERDES2 sites.
    .DOUT(DOUT), // 1-bit Delayed data signal to IOB.
    .TOUT(TOUT), // 1-bit Delayed Tristate signal to IOB.
    .CAL(CAL), // 1-bit Enter the IODELAY calibration sequence. This will last between 8 and 16
               // GCLK cycles. Drives RDY HIGH when complete. Shared with DRP pin ADD(add).

    .CE(CE), // 1-bit Enable the Increment/Decrement signal.
    .CLK(CLK), // 1-bit CLK0 from INT connects to "Master" and CLK1 from INT connects to "Slave"
    .IDATAIN(IDATAIN), // 1-bit Data signal from IOB.
    .INC(INC), // 1-bit Increment / Decrement signal. Used for setting tap or delay length.
    // IOCLK0 - IOCLK1: 1-bit (each) Optionally Invertible IO clock network input.
    .IOCLK0(IOCLK0),
    .IOCLK1(IOCLK1),
    .ODATAIN(ODATAIN), // 1-bit Data input signal from OLOGIC or OSERDES.
    .RST(RST), // 1-bit Reset the IODELAY2 to either zero or 1/2 of total period. RST_VALUE
               // attribute controls this choice.

    .T(T) // 1-bit Tristate input signal from OLOGIC or OSERDES.
);

// End of IODELAY2_inst instantiation

```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

IODRP2

プリミティブ : I/O Control Port

概要

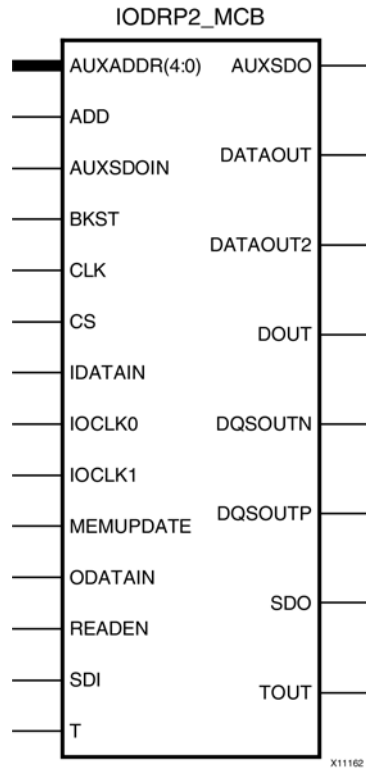
ザイリンクスではこのELEMENTの使用をサポートしていません。

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

IODRP2_MCB

プリミティブ : I/O Control Port for the Memory Controller Block



概要

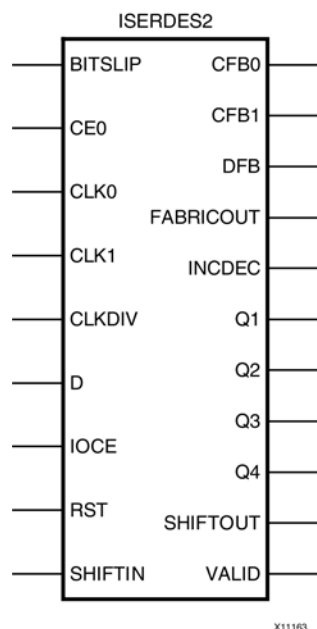
このデザイン エLEMENTは、外部メモリ インターフェイスをインプリメントするために MIG (Memory Interface Generator) コアで MCB ブロックと組み合わせて使用されます。MIG の外部での使用はサポートされていません。

詳細情報

- ・ [ザイリンクス メモリ インターフェイス ジェネレータ \(MIG\) ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

ISERDES2

プリミティブ : Input SERial/DESerializer.



概要

これは、入力のシリアル/パラレル データ コンバータで、高速のソース同期シリアル データを取り込みます。このコンポーネントには、クロック管理およびデータ アライメントを行うロジックが含まれており、シングル データ レート (SDR) またはダブル データ レート (DDR) のデータを、単一のインスタンス (MASTER) では 2 ~ 6 ビットのデータ幅に、カスケード接続された 2 つの ISERDES2 (MASTER/SLAVE) では、7 ~ 10 ビットのデータ幅にできます。このコンポーネントは、メモリ、さまざまなタイプのデータ インターフェイス アプリケーションに使用できます。また、IODELAY と共に使用して、入力シリアル データのアライメントを実行できます。DDR モードでは、データ キャプチャに 1 クロックまたは 2 クロック かかります。2 クロック モードを使用すると、パフォーマンスが向上する場合がありますが、必要なクロック リソース数および消費電力が増加し、配置配線での制限も発生する可能性があります。高速の I/O パフォーマンスが不要な場合は、1 クロック モードを使用してください。

ポートの説明

ポート名	タイプ	幅	機能
BITSLLIP	入力	1	ビットスリップを有効にします。カスケードの有無に関らず DATA_WIDTH と共に使用できます。スリップするビット数は、DATA_WIDTH で選択されている値によって決まります。
CE0	入力	1	最終 (グローバル クロック ドリブン) レジスタのクロック イネーブル入力
CFB0	出力	1	PLL/DCM で生成されたクロックを BUFIO2FB を介して PLL/DCM にフィードバックします。
CFB1	出力	1	PLL/DCM で生成されたクロックを BUFIO2FB を介して PLL/DCM にフィードバックします (セカンダリ)。
CLKDIV	入力	1	グローバル クロック ネットワーク入力。ハードウェアのクロックです。

ポート名	タイプ	幅	機能
CLK0	入力	1	I/O クロック ネットワークの入力。オプションで反転可能です。倍クロック回路が使用されていない場合に使用するプライマリクロック入力です (DATA_RATE を参照)。
CLK1	入力	1	I/O クロック ネットワークの入力。オプションで反転可能です。倍クロック回路が使用されている場合にのみ使用されるセカンダリクロック入力です (DATA_RATE を参照)。
D	入力	1	データ入力。IODELAY2 ブロックによる遅延の後のデータ入力です。
DFB	出力	1	IODELAY2 エLEMENTで遅延した入力クロックを BUFIO2 を介して DCM、PLL、または BUFG に転送します。
FABRICOUT	出力	1	FPGA ロジックで使用される非同期データ。
INCDEC	出力	1	マスタ モードでの位相検出器の出力です (スレーブ モードではダミー)。受信データのサンプリングのタイミングが早かったか遅かったかを FPGA ロジックに通知します。
IOCE	入力	1	BUFIO CE から派生するデータ ストローブ信号。選択されている SerDes モードでの I/O およびグローバル クロックに対し正しいタイミングでストローブ データ キャプチャが行われます。
Q1 ~ Q4	出力	1	ハードウェアへのレジスタ付き出力信号。
RST	入力	1	非同期リセットのみ。
SHIFTIN	入力	1	マスタ/スレーブ I/O のカスケード入力信号。マスタおよびスレーブのサイトが、4 よりも大きい値の DATA_WIDTH と共に使用される場合に使用します。ブロックがマスタの場合、位相検出器モードで使用されるデータ入力を送信します。スレーブの場合は、パラレル データとなるシリアル データ入力を送信します。
SHIFTOUT	出力	1	マスタ/スレーブ I/O のカスケード出力信号。サンプルされたデータをスレーブから送信するのに使用します。マスタ モードの場合、入力シフトレジスタの 4 段目からシリアル データをスレーブに送信します。
VALID	出力	1	マスタ モードでの位相検出器の出力です (スレーブ モードではダミー)。入力データにエッジがない場合 (位相検出器で利用できる情報がない場合)、VALID 信号は Low になり、FPGA ロジックで INCDEC 信号が無視されることを示します。

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
BITSLIP_ENABLE	ブール代数	FALSE、TRUE	FALSE	BITSLIP 入力ピンで制御するビットスリップ ファンクションのオン/オフを指定します。ス リップするビット数は、DATA_WIDTH で選 択されている値によって決まります。オフの 場合は、Bitslip CE が常に、IOCE クロック イネーブルの前の I/O クロック のデフォ ルト値になります。
DATA_RATE	文字列	SDR、DDR	SDR	データレート設定。DDR クロックは別の複 数の I/O クロックまたは 1 つの I/O クロ ックによって供給されます。2 つのクロックが 供給される場合、その位相差は約 180 度で ある必要があります。
DATA_WIDTH	整数	1、2、3、4、5、6、7、 8	1	データ幅。シリアルからパラレルへのコン バータのパラレル データ出力幅を定義しま す。2 つの ISERDES2 ブロックがカスケードさ れる場合にのみ、4 より大きい値が有効になり ます。この場合、同じ値がマスタブロックとス レーブブロックに適用される必要があります。
INTERFACE_TYPE	文字列	NETWORKING、 NETWORKING_ PIPELINED、 RETIMED	NETWORKING	操作モードを選択し、どのセットのパラレル データが FPGA ロジックに使用可能である かを決定します。
SERDES_MODE	文字列	NONE、 MASTER、 SLAVE	NONE	2 つの ISERDES2 ブロックがカスケードされ ている場合、ISERDES が単独で使用されて いるか、マスタまたはスレーブとして使用され ているかを示します。

VHDL 記述 (インスタンスエーション)

```
-- ISERDES2: Input SERIAL/DESerializer
--          Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

ISERDES2_inst : ISERDES2
generic map (
    BITSLIP_ENABLE => FALSE,      -- Enable the Bitslip functionality. Only available in NETWORKING mode.
    DATA_RATE      => "SDR",      -- The number of bits slipped is a function of the DATA_WIDTH selected.
    DATA_WIDTH     => 1,           -- Single Data Rate or Double Data Rate operation. The DDR clock can be
    INTERFACE_TYPE  => "NETWORKING", -- supplied by separate IO clocks or by a single IO clock. If two clocks
    SERDES_MODE     => "MASTER"   -- are supplied they must be approximately 180 degrees out of phase. A
                                     -- MAX SKEW timing check should be specified.
                                     -- Parallel data width selection.
                                     -- Memory or Networking interface type.
                                     -- Specify whether the ISERDES2 is operating in master or slave modes
                                     -- when cascaded width expansion.
)
port map (
    CFB0 => CFB0,      -- 1-bit For VIDEO use cases. Connects to BUFIO2FB.
    CFB1 => CFB1,
    DFB => DFB,        -- 1-bit For VIDEO use cases. Connects to BUFIO2.
    FABRICOUT => FABRICOUT, -- 1-bit Allows signals from PAD IOBs to go to fabric unregistered and optionally
                                     -- delayed.

    INCDEC => INCDEC,   -- 1-bit Output of Phase Detector (Dummy in slave)
    -- Q1 - Q4: 1-bit (each) Registered output to fabric.
    Q1 => Q1,
    Q2 => Q2,
    Q3 => Q3,
    Q4 => Q4,
    SHIFTOUT => SHIFTOUT, -- 1-bit Cascade out signal for Master/Slave IO. In Phase Detector mode used to
                                     -- send slave sampled data.
```

```
VALID => VALID,          -- 1-bit Output of Phase Detector (Dummy in Slave). If the input data contains no
                          -- edges (no info for the phase detector to work with) the VALID signal will go
                          -- LOW to indicate that the fabric should ignore the INCDEC signal.

BITSLIP => BITSLIP,       -- 1-bit Invoke Bitslip. This can be used with any DATA_WIDTH, cascaded or not.
                          -- The amount of bitslip is fixed by the DATA_WIDTH selection.

CE0 => CE0,               -- 1-bit Clock enable input
CLK0 => CLK0,             -- 1-bit IO Clock network input. Optionally Invertible. This is the primary clock
                          -- input used when the clock doubler circuit is not engaged (see DATA_RATE
                          -- attribute).

CLK1 => CLK1,             -- 1-bit Optionally invertible IO Clock network input. Timing note: CLK1 should
                          -- be 180 degrees out of phase with CLK0.

CLKDIV => CLKDIV,         -- 1-bit Global clock network input. This is the clock for the fabric domain.
D => D,                   -- 1-bit Input signal from IOB.
IOCE => IOCE,             -- 1-bit Data strobe signal derived from BUFIO CE. Strokes data capture for
                          -- NETWORKING and NETWORKING_PIPELINES alignment modes.

RST => RST,               -- 1-bit Asynchronous reset only.
SHIFTIN => SHIFTIN        -- 1-bit Cascade in signal for Master/Slave IO. Master and Slave sites can be
                          -- used together for DATA_WIDTHs greater than 4. In Phase Detector mode used to
                          -- send slave sampled data.

);

-- End of ISERDES2_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ISERDES2: Input SERIAL/DESerializer
//          Spartan-6
// Xilinx HDL Language Template, version 11.1

ISERDES2 #(
    .BITSLIP_ENABLE("FALSE"),      // Enable the Bitflip functionality. Only available in NETWORKING mode.
                                    // The number of bits slipped is a function of the DATA_WIDTH selected.
    .DATA_RATE("SDR"),            // Single Data Rate or Double Data Rate operation. The DDR clock can be
                                    // supplied by separate IO clocks or by a single IO clock. If two clocks
                                    // are supplied they must be approximately 180 degrees out of phase. A MAX
                                    // SKEW timing check should be specified.
    .DATA_WIDTH(1),               // Parallel data width selection.
    .INTERFACE_TYPE("NETWORKING"), // Memory or Networking interface type.
    .SERDES_MODE("MASTER")        // Specify whether the ISERDES2 is operating in master or slave modes when
                                    // cascaded width expansion.
)
ISERDES2_inst (
    .CFB0(CFB0),                  // 1-bit For VIDEO use cases. Connects to BUFIO2FB.
    .CFB1(CFB1),
    .DFB(DFB),                    // 1-bit For VIDEO use cases. Connects to BUFIO2.
    .FABRICOUT(FABRICOUT),        // 1-bit Allows signals from PAD IOBs to go to fabric unregistered and optionally
                                    // delayed.

    .INCDEC(INCDEC),              // 1-bit Output of Phase Detector (Dummy in slave)
    // Q1 - Q4: 1-bit (each) Registered output to fabric.
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .SHIFTOUT(SHIFTOUT),          // 1-bit Cascade out signal for Master/Slave IO. In Phase Detector mode used to
                                    // send slave sampled data.

    .VALID(VALID),               // 1-bit Output of Phase Detector (Dummy in Slave). If the input data contains no
                                    // edges (no info for the phase detector to work with) the VALID signal will go
                                    // LOW to indicate that the fabric should ignore the INCDEC signal.

    .BITSLIP(BITSLIP),           // 1-bit Invoke Bitflip. This can be used with any DATA_WIDTH, cascaded or not.
                                    // The amount of bitflip is fixed by the DATA_WIDTH selection.

    .CE0(CE0),                   // 1-bit Clock enable input
    .CLK0(CLK0),                 // 1-bit IO Clock network input. Optionally Invertible. This is the primary clock
                                    // input used when the clock doubler circuit is not engaged (see DATA_RATE
                                    // attribute).

    .CLK1(CLK1),                 // 1-bit Optionally invertible IO Clock network input. Timing note: CLK1 should be
                                    // 180 degrees out of phase with CLK0.

    .CLKDIV(CLKDIV),             // 1-bit Global clock network input. This is the clock for the fabric domain.
    .D(D),                       // 1-bit Input signal from IOB.
    .IOCE(IOCE),                 // 1-bit Data strobe signal derived from BUFIO CE. Strobos data capture for
                                    // NETWORKING and NETWORKING_PIPELINES alignment modes.

    .RST(RST),                   // 1-bit Asynchronous reset only.
    .SHIFTIN(SHIFTIN)            // 1-bit Cascade in signal for Master/Slave IO. Master and Slave sites can be used
                                    // together for DATA_WIDTHs greater than 4. In Phase Detector mode used to send
                                    // slave sampled data.
);

// End of ISERDES2_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

KEEPER

プリミティブ : KEEPER Symbol



概要

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

ポートの説明

属性	方向	幅	機能
O	出力	1 ビット	キーパ出力

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of KEEPER_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

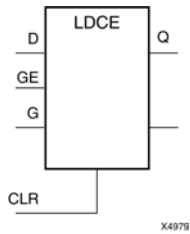
// End of KEEPER_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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

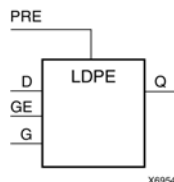
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	0	コンフィギュレーション後の Q 出力の初期値を指定

詳細情報

[Spartan-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

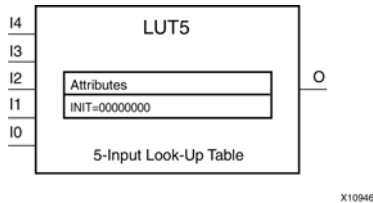
属性	タイプ	値	デフォルト	説明
INIT	2 進数	0、1	1	電源投入時または Q ポートに対する GSR のアサート時の初期値を指定

詳細情報

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

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

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

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

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

論理表

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

ポートの説明

ポート名	方向	幅	機能
O	出力	1	5 入力 LUT 出力
I0、I1、I2、I3、I4	入力	1	LUT 入力

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	32 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of LUT5_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

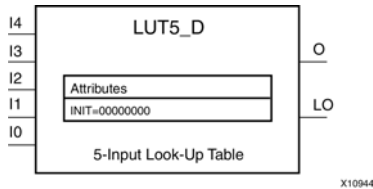
// End of LUT5_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT5_D

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



概要

このデザイン エレメントは、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
);

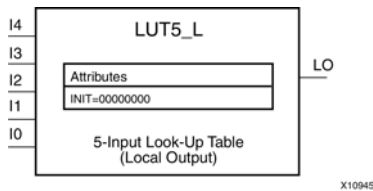
// End of LUT5_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT5_L

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



概要

このデザイン エLEMENTは、入力 5 個、出力 1 個のルックアップ テーブル (LUT) で、非同期 32 ビット ROM (5 ビットのアドレス指定) として動作するか、5 入力のロジック ファンクションをインプリメントできます。LUT は基本的なロジック構築ブロックで、デザインに含まれる多くのロジック ファンクションをインプリメントするときに使用します。LUT5 が 1 つの場合はスライス内に含まれる LUT6 に、2 つの場合は 1 つの LUT6 に多少の制限はありますがパックできます。LUT5、LUT5_L、および LUT5_D の機能は同じですが、LUT5_L および LUT5_D では、LUT5 の出力信号を内部スライスに接続したり、LO 出力を使用して CLB に接続することができます。LUT5_L では LUT5 からの接続のみが 1 つのスライスまたは CLB に含まれるように指定できるのに対し、LUT5_D では LUT5 の出力を内部スライス/CLB ロジックと外部ロジックの両方に接続するように指定できます。LUT5 では出力の接続は特定されないため、内部スライスまたは CLB 信号の接続を暗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(0), // LUT general output
    .I0(I0), // LUT input
    .I1(I1), // LUT input
    .I2(I2), // LUT input
    .I3(I3), // LUT input
    .I4(I4) // LUT input
);

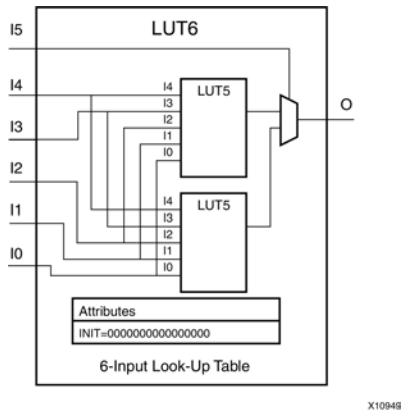
// End of LUT5_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT6

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



概要

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

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

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

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

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

論理表

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

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

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of LUT6_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

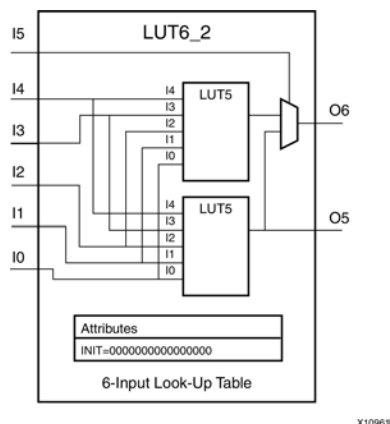
// End of LUT6_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

LUT6_2

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



概要

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

LUT のロジック ファンクションを指定するために 64 ビットの 16 進数で INIT 属性を設定する必要があります。INIT 値は、関連する入力適用されるときに、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	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)
);

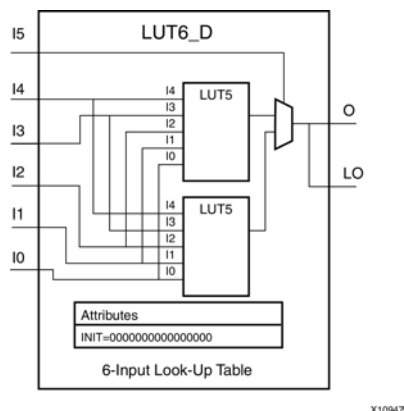
// End of LUT6_2_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

LUT6_D

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



概要

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

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

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

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

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

論理表

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

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

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of LUT6_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

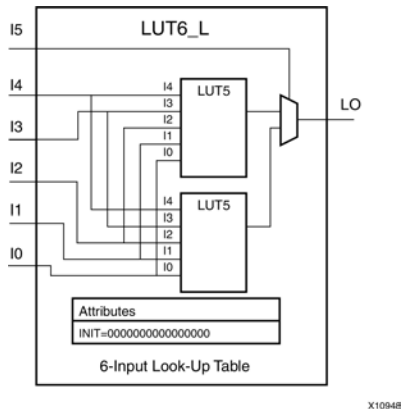
// End of LUT6_inst instantiation
```


詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

LUT6_L

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



概要

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

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

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

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

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

論理表

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

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

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

ポートの説明

ポート名	方向	幅	機能
LO	出力	1	6/5 入力 LUT 出力または内部 CLB 接続
I0、I1、I2、I3、I4、I5	入力	1	LUT 入力

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	64 ビット値	すべてゼロ	ルックアップ テーブルの論理値を指定

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of LUT6_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

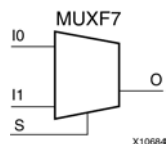
// End of LUT6_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

MUXF7

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



概要

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

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

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF7_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

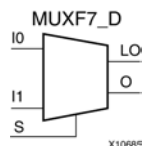
// End of MUXF7_inst instantiation
```

詳細情報

- [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

MUXF7_D

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



概要

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

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

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF7_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

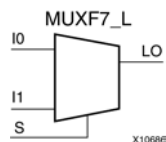
// End of MUXF7_D_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

MUXF7_L

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



概要

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

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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
);

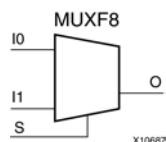
// End of MUXF7_L_inst instantiation
```

詳細情報

- [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

MUXF8

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



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF8_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

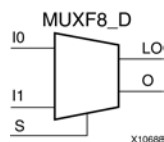
// End of MUXF8_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

MUXF8_D

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



概要

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

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

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF8_D_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

MUXF8_D MUXF8_D_inst (
    .LO(LO), // Ouput of MUX to local routing
    .O(O),   // Output of MUX to general routing
    .IO(IO), // Input (tie to MUXF7 LO out)
    .I1(I1), // Input (tie to MUXF7 LO out)
    .S(S)    // Input select to MUX
);

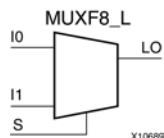
// End of MUXF8_D_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

MUXF8_L

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



概要

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

LO 出力は、同じ CLB スライス内にある別の入力との接続に使用します。

論理表

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of MUXF8_L_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

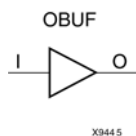
// End of MUXF8_L_inst instantiation
```

詳細情報

- [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

OBUF

プリミティブ : Output Buffer



概要

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

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
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;

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

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

-- End of OBUF_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

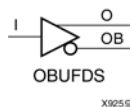
// End of OBUF_inst instantiation
```

詳細情報

- [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

OBUFDS

プリミティブ : Differential Signaling Output Buffer



概要

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

論理表

入力	出力	
I	O	OB
0	0	1
1	1	0

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

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

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

-- End of OBUFDS_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

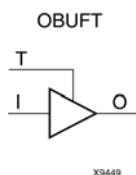
// End of OBUFDS_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

OBUFT

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



概要

このデザイン エLEMENTは、入力 (I)、出力 (O)、アクティブ Low 出力イネーブル (T) を持つ単一のトライステート出力バッファです。このELEMENTでは、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。

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

論理表

入力		出力
T	I	O
1	X	Z
0	I	f

ポートの説明

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

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
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;

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

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

-- End of OBUFT_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

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

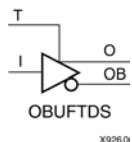
// End of OBUFT_inst instantiation
```

詳細情報

- [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

OBUFTDS

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



概要

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

論理表

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

ポートの説明

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

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

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

OBUFTDS_inst : OBUFTDS
generic map (
    IOSTANDARD => "DEFAULT")
port map (
    O => O,      -- Diff_p output (connect directly to top-level port)
    OB => OB,    -- Diff_n output (connect directly to top-level port)
    I => I,      -- Buffer input
    T => T       -- 3-state enable input
);

-- End of OBUFTDS_inst instantiation
```

Verilog 記述 (インスタンスレーション)

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

OBUFTDS #(
    .IOSTANDARD("DEFAULT") // Specify the output I/O standard
) OBUFTDS_inst (
    .O(O),      // Diff_p output (connect directly to top-level port)
    .OB(OB),    // Diff_n output (connect directly to top-level port)
    .I(I),      // Buffer input
    .T(T)       // 3-state enable input
);

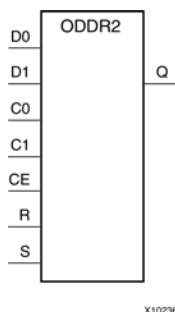
// End of OBUFTDS_inst instantiation
```

詳細情報

- [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

ODDR2

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



概要

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

論理表

入力							出力
S	R	CE	D0	D1	C0	C1	O
1	X	X	X	X	X	X	1
0	1	X	X	X	X	X	not INIT
0	0	0	X	X	X	X	変化なし
0	0	1	D0	X	↑	X	D0
0	0	1	X	D1	X	↑	D1

セット/リセットは SRTYPE 値で同期に設定可能

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

-- ODDR2: Output Double Data Rate Output Register with Set, Reset
--      and Clock Enable.
--      Spartan-3E/3A/6
-- Xilinx HDL Libraries Guide, version 11.2

ODDR2_inst : ODDR2
generic map(
  DDR_ALIGNMENT => "NONE", -- Sets output alignment to "NONE", "C0", "C1"
  INIT => '0', -- Sets initial state of the Q output to '0' or '1'
  SRTYPE => "SYNC") -- Specifies "SYNC" or "ASYNC" set/reset
port map (
  Q => Q, -- 1-bit output data
  C0 => C0, -- 1-bit clock input
  C1 => C1, -- 1-bit clock input
  CE => CE, -- 1-bit clock enable input
  D0 => D0, -- 1-bit data input (associated with C0)
  D1 => D1, -- 1-bit data input (associated with C1)
  R => R, -- 1-bit reset input
  S => S -- 1-bit set input
);

-- End of ODDR2_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// ODDR2: Output Double Data Rate Output Register with Set, Reset
//      and Clock Enable.
//      Spartan-3E/3A/6
// Xilinx HDL Libraries Guide, version 11.2

ODDR2 #(
  .DDR_ALIGNMENT("NONE"), // Sets output alignment to "NONE", "C0" or "C1"
  .INIT(1'b0), // Sets initial state of the Q output to 1'b0 or 1'b1
  .SRTYPE("SYNC") // Specifies "SYNC" or "ASYNC" set/reset
) ODDR2_inst (
  .Q(Q), // 1-bit DDR output data
  .C0(C0), // 1-bit clock input
  .C1(C1), // 1-bit clock input
  .CE(CE), // 1-bit clock enable input
  .D0(D0), // 1-bit data input (associated with C0)
  .D1(D1), // 1-bit data input (associated with C1)
  .R(R), // 1-bit reset input
  .S(S) // 1-bit set input
);
```

```
// End of ODDR2_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

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 メモ : 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

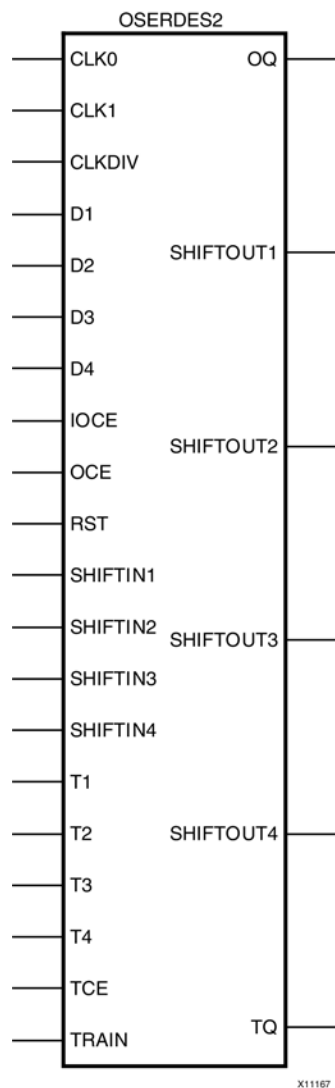
インスタンス化	推奨
推論	不可
CORE Generator™ およびウィザード	不可
マクロのサポート	不可

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

OSERDES2

プリミティブ : Dedicated IOB Output Serializer



概要

このプリミティブを使用すると、同期インターフェイスを簡単にインプリメントできます。このモジュールを使用すると、FPGA のロジックリソースを節約でき、タイミングが複雑になるのを防ぎます。また、さまざまアプリケーションに対応した複数のクロック入力があり、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
CLKDIV	入力	1	グローバル クロック ネットワーク入力。ハードウェアのクロックです。
CLK0	入力	1	オプションで反転可能な I/O クロック ネットワーク入力。倍クロック回路が使用されていない場合に使用するプライマリ クロック入力です。詳細は DATA_RATE を参照。
CLK1	入力	1	I/O クロック ネットワーク入力。オプションで反転可能です。
D1 ~ D4	入力	1	CLK0 と CLK1 の位相差は 180 度である必要があります。
IOCE	入力	1	BUFIO CE から派生する伝送出力イネーブル信号。入力がサンプルされるタイミングを決定するストロボです。
OCE	入力	1	データ入力のクロック イネーブル。
OQ	出力	1	パッドまたは IODELAY2 へのデータ パス出力。
RST	入力	1	共有データ/トライステートリセットピン。非同期のみ。
SHIFTIN1 ~ SHIFTIN2	入力	1	カスケード データ入力信号 (マスタでダミー)。4 を超える DATA_WIDTH に使用。
SHIFTIN3 ~ SHIFTIN4	入力	1	差動データ入力信号 (スレーブでダミー)。
SHIFTOUT1 ~ SHIFTOUT2	出力	1	カスケード データ出力信号 (スレーブでダミー)。4 を超える DATA_WIDTH に使用。
SHIFTOUT3 ~ SHIFTOUT4	出力	1	差動データ入力信号 (マスタでダミー)。
TCE	入力	1	トライステート入力のクロック イネーブル。
TQ	出力	1	パッドまたは IODELAY2 へのトライステート パス出力。
TRAIN	入力	1	トレーニング パターンの使用を有効にします。信号受信回路の補正に使用する固定出力パターンを指定するためにします。このピンを使用し、デバイスで出力が固定パターンであるか、ピンからの入力データであるかを制御することができます。
T1 ~ T4	入力	1	パラレル トライステート入力。OSERDES2 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。

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	データレート設定。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_OT	文字列	DDR、BUF、SDR	DDR	トライステート データレート設定。トライステートを CLK の各エッジで変化させるか、各立ち上がりエッジで変化させるか、バッファのコンフィギュレーションで変化させるかを指定します。
DATA_WIDTH	整数	2、1、3、4、5、6、7、8	2	デバイスから IOB にシリアル化されるビット数を設定。
OUTPUT_MODE	文字列	SINGLE_ENDED、DIFFERENTIAL	SINGLE_ENDED	出力モード。
SERDES_MODE	文字列	MASTER、SLAVE	MASTER	カスケードされている場合 SERDES がマスタまたはスレーブのどちらで使用されているかを示します。
TRAIN_PATTERN	Integer	0、1、2、3、4、5、6、7、8、9、10、11、12、13、14、15	0	トレーニング パターン。TRAIN ピンを参照。

VHDL 記述 (インスタンスエーション)

```

-- OSERDES2: Dedicated IOB Output Serializer
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

OSERDES2_inst : OSERDES2
generic map (
  BYPASS_GCLK_FF => FALSE,
  DATA_RATE_OQ => "DDR",           -- Data rate setting. Defines whether the data changes at every clock edge
                                     -- or every positive clock edge with respect to CLK.
  DATA_RATE_OT => "DDR",           -- Tristate Data rate setting. Defines whether the 3-state changes at
                                     -- every clock edge, every positive clock edge, or buffer configuration
                                     -- with respect to CLK.
  DATA_WIDTH => 2,                 -- Sets how many bits from the fabric to serialize to the IOB.
  OUTPUT_MODE => "SINGLE_ENDED",     -- Output Mode.
  SERDES_MODE => "MASTER",          -- Indicates whether SERDES is being used as a Master or Slave when
                                     -- cascaded.
  TRAIN_PATTERN => 0                -- Training pattern. See comments with TRAIN pin.
)
port map (
  OQ => OQ,                         -- 1-bit Data path output to pad or IODELAY2.
  -- SHIFTOUT1 - SHIFTOUT2: 1-bit (each) Cascade data output signal (dummy in Slave). Used for DATA_WIDTHs
  -- greater than 4.
  SHIFTOUT1 => SHIFTOUT1,
  SHIFTOUT2 => SHIFTOUT2,
  -- SHIFTOUT3 - SHIFTOUT4: 1-bit (each) Differential data output signal (dummy in Master).
  SHIFTOUT3 => SHIFTOUT3,
  SHIFTOUT4 => SHIFTOUT4,
  TQ => TQ,                         -- 1-bit Tristate path output to pad or IODELAY2.
  CLK0 => CLK0,                     -- 1-bit Optionally Invertible IO Clock network input. This is the primary clock
                                     -- input used when the clock doubler circuit is not engaged (see DATA_RATE
                                     -- attribute).
  CLK1 => CLK1,                     -- 1-bit IO Clock network input. Optionally Invertible. Timing note: CLK1 should
                                     -- be 180 degrees out of phase with CLK0.
  CLKDIV => CLKDIV,                 -- 1-bit Global clock network input. This is the clock for the fabric domain.
  -- D1 - D4: 1-bit (each) Data input
  D1 => D1,
  D2 => D2,
  D3 => D3,
  D4 => D4,
  IOCE => IOCE,                     -- 1-bit "Transfer Out" enable signal derived from BUFIO CE. This is the strobe

```

```
-- that determines when the input data is sampled.

OCE => OCE,           -- 1-bit Clock enable for data inputs.
RST => RST,           -- 1-bit Shared Data/Tristate Reset pin. Asynchronous only.
-- SHIFTIN1 - SHIFTIN2: 1-bit (each) Cascade data input signal (dummy in Master). Used for DATA_WIDTHS
-- greater than 4.
SHIFTIN1 => SHIFTIN1,
SHIFTIN2 => SHIFTIN2,
-- SHIFTIN3 - SHIFTIN4: 1-bit (each) Differential data input Signal (dummy in Slave).
SHIFTIN3 => SHIFTIN3,
SHIFTIN4 => SHIFTIN4,
-- T1 - T4: 1-bit (each) Parallel 3-State Inputs - Ports T1 to T4 are the location in which all parallel
-- 3-state signals enters the OSERDES2 module. This port is connected to the FPGA fabric, and can be
-- configured from 1 to 4 bits. This feature is not supported in the extended width mode.
T1 => T1,
T2 => T2,
T3 => T3,
T4 => T4,
TCE => TCE,           -- 1-bit Clock enable for tristate inputs.
TRAIN => TRAIN        -- 1-bit Enable use of the training pattern. The "train" function is a means of
-- specifying a fixed output pattern that is used to calibrate the receiver of
-- the signal. This pin allows the fabric to control whether the output is that
-- fixed pattern or the input data from the pins.

);

-- End of OSERDES2_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// OSERDES2: Dedicated IOB Output Serializer
//          Spartan-6
// Xilinx HDL Language Template, version 11.1

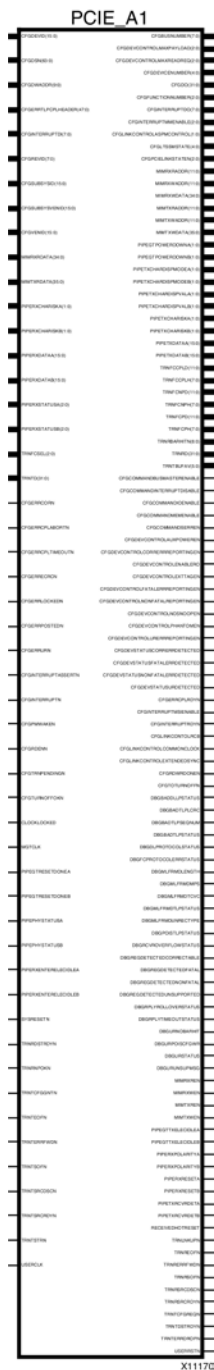
OSERDES2 #(
    .BYPASS_GCLK_FF("FALSE"),
    .DATA_RATE_OQ("DDR"),           // Data rate setting. Defines whether the data changes at every clock edge
                                     // or every positive clock edge with respect to CLK.
    .DATA_RATE_OT("DDR"),           // Tristate Data rate setting. Defines whether the 3-state changes at every
                                     // clock edge, every positive clock edge, or buffer configuration with
                                     // respect to CLK.
    .DATA_WIDTH(2),                 // Sets how many bits from the fabric to serialize to the IOB.
    .OUTPUT_MODE("SINGLE_ENDED"),    // Output Mode.
    .SERDES_MODE("MASTER"),         // Indicates whether SERDES is being used as a Master or Slave when
                                     // cascaded.
    .TRAIN_PATTERN(0)               // Training pattern. See comments with TRAIN pin.
)
OSERDES2_inst (
    .OQ(OQ),                        // 1-bit Data path output to pad or IODELAY2.
    // SHIFTOUT1 - SHIFTOUT2: 1-bit (each) Cascade data output signal (dummy in Slave). Used for DATA_WIDTHS
    // greater than 4.
    .SHIFTOUT1(SHIFTOUT1),
    .SHIFTOUT2(SHIFTOUT2),
    // SHIFTOUT3 - SHIFTOUT4: 1-bit (each) Differential data output signal (dummy in Master).
    .SHIFTOUT3(SHIFTOUT3),
    .SHIFTOUT4(SHIFTOUT4),
    .TQ(TQ),                        // 1-bit Tristate path output to pad or IODELAY2.
    .CLK0(CLK0),                    // 1-bit Optionally Invertible IO Clock network input. This is the primary clock
    // input used when the clock doubler circuit is not engaged (see DATA_RATE
    // attribute).
    .CLK1(CLK1),                    // 1-bit IO Clock network input. Optionally Invertible. Timing note: CLK1 should
    // be 180 degrees out of phase with CLK0.
    .CLKDIV(CLKDIV),                // 1-bit Global clock network input. This is the clock for the fabric domain.
    // D1 - D4: 1-bit (each) Data input
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .IOCE(IOCE),                    // 1-bit "Transfer Out" enable signal derived from BUFIO CE. This is the strobe
    // that determines when the input data is sampled.
    .OCE(OCE),                      // 1-bit Clock enable for data inputs.
    .RST(RST),                      // 1-bit Shared Data/Tristate Reset pin. Asynchronous only.
    // SHIFTIN1 - SHIFTIN2: 1-bit (each) Cascade data input signal (dummy in Master). Used for DATA_WIDTHS
    // greater than 4.
    .SHIFTIN1(SHIFTIN1),
    .SHIFTIN2(SHIFTIN2),
    // SHIFTIN3 - SHIFTIN4: 1-bit (each) Differential data input Signal (dummy in Slave).
    .SHIFTIN3(SHIFTIN3),
    .SHIFTIN4(SHIFTIN4),
    // T1 - T4: 1-bit (each) Parallel 3-State Inputs - Ports T1 to T4 are the location in which all parallel
    // 3-state signals enters the OSERDES2 module. This port is connected to the FPGA fabric, and can be
    // configured from 1 to 4 bits. This feature is not supported in the extended width mode.
    .T1(T1),
    .T2(T2),
    .T3(T3),
    .T4(T4),
    .TCE(TCE),                      // 1-bit Clock enable for tristate inputs.
    .TRAIN(TRAIN)                   // 1-bit Enable use of the training pattern. The "train" function is a means of
    // specifying a fixed output pattern that is used to calibrate the receiver of the
    // signal. This pin allows the fabric to control whether the output is that fixed
    // pattern or the input data from the pins.
);

// End of OSERDES2_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

プリミティブ : PCI Express



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。PCI EXPRESS® デザインを PCIE_A1 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_A1 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

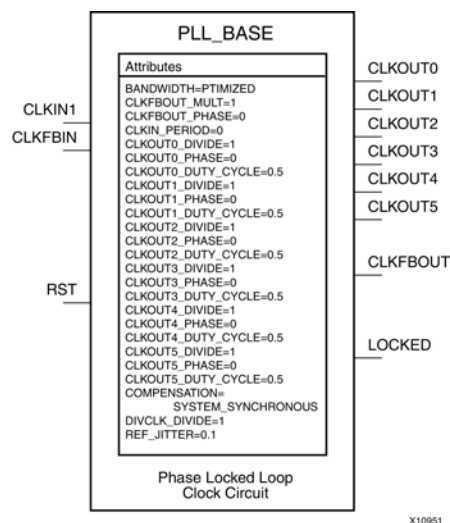
このELEMENTをインスタンス化するには、PCI EXPRESS® コアまたはこのELEMENTを含む関連コアを使用します。このELEMENTは直接インスタンス化しないでください。

詳細情報

- ・ [Spartan-6 FPGA RocketIO GTP トランシーバ ユーザー ガイド](#)
- ・ [PCI EXPRESS® 用 LogiCORE™ IP Spartan-6 FPGA 統合エンドポイント ブロック v1.1 ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

PLL_BASE

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



概要

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

ポートの説明

ポート名	方向	幅	機能
クロック出力/入力			
CLKOUT0 ~ 5	出力	1	位相が制御される 6 個の出力クロックの 1 つ
CLKFBOUT	出力	1	クロック ネットワークの遅延調整方法を指定するために使用する専用 PLL フィードバック出力。この出力の接続の有無は、調整方法によって異なります。
CLKIN	入力	1	PLL のクロック ソース入力。FPGA の専用クロック ピン、DCM 出力クロック ピン、または BUFG 出力ピンによって駆動されます。
CLKFBIN	入力	1	クロック フィードバック入力。CLKFBOUT ポートからのみ接続できます。
ステータス出力/制御入力			
LOCKED	出力	1	位相アライメントが完了し、操作が開始可能であることを示す同期出力
RST	入力	1	非同期リセット

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of PLL_BASE_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

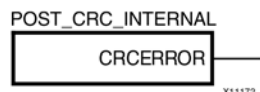
// End of PLL_BASE_inst instantiation
```

詳細情報

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

POST_CRC_INTERNAL

プリミティブ : Post-configuration CRC error detection



概要

このプリミティブを使用しハードウェアから CRC エラーを報告します。この新しいプリミティブは POST_CRC を拡張するために追加されています。また、CRC_EXTSTAT_DISABLE がアクティベートされているときの POST CRC ステータスへの唯一アクセスでもあります。

ポートの説明

ポート名	タイプ	幅	機能
CRCERROR	出力	1	コンフィギュレーション後の CRC エラー

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)

```
-- POST_CRC_INTERNAL: Post-configuration CRC error detection
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

POST_CRC_INTERNAL_inst : POST_CRC_INTERNAL
generic map (
)
port map (
  CRCERROR => CRCERROR  -- 1-bit Post-configuration CRC error
);

-- End of POST_CRC_INTERNAL_inst instantiation
```

Verilog 記述 (インスタンス化)

```
// POST_CRC_INTERNAL: Post-configuration CRC error detection
//                      Spartan-6
// Xilinx HDL Language Template, version 11.1

POST_CRC_INTERNAL POST_CRC_INTERNAL_inst (
  .CRCERROR(CRCERROR) // 1-bit Post-configuration CRC error
);

// End of POST_CRC_INTERNAL_inst instantiation
```

詳細情報

- [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

PULLDOWN

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

PULLDOWN



概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of PULLDOWN_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

// End of PULLDOWN_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

PULLUP

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



概要

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

ポートの説明

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

デザインの入力方法

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

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of PULLUP_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

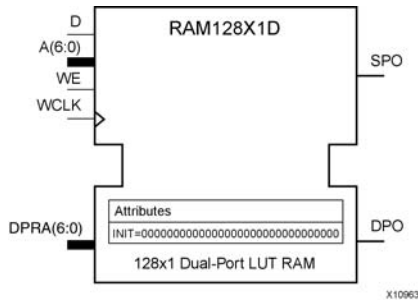
// End of PULLUP_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA SelectIO リソース ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM128X1D

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



概要

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

ポートの説明

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

インスタンスエートする場合は、このコンポーネントを次のように接続します。

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

指定しない場合は、初期値はすべてゼロになります。

デザインの入力方法

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	128 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンス化)

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

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

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

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

-- End of RAM128X1D_inst instantiation
```

Verilog 記述 (インスタンス化)

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

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

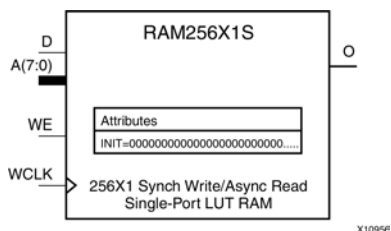
// End of RAM128X1D_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM256X1S

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



概要

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

ポートの説明

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

デザインの入力方法

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

インスタンス化する場合、このコンポーネントを次のように接続します。

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

指定しない場合は、初期値はすべてゼロになります。

使用可能な属性

属性	タイプ	値	デフォルト	説明
INIT	16 進数	256 ビット値	すべてゼロ	RAM の初期値を指定

VHDL 記述 (インスタンスエーション)

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

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

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

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

-- End of RAM256X1S_inst instantiation
```

Verilog 記述 (インスタンスエーション)

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

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

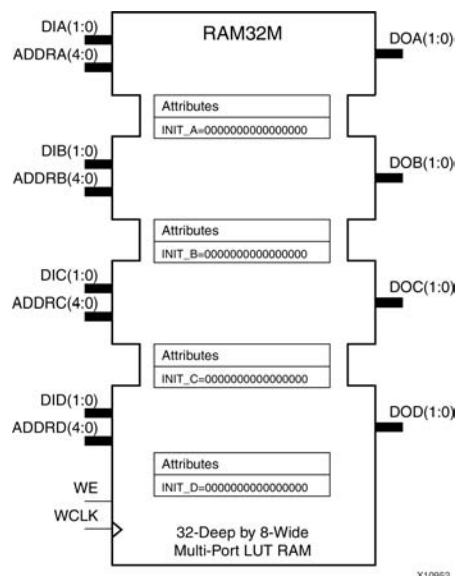
// End of RAM256X1S_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM32M

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



概要

このデザイン エLEMENTは、32 ワード X 8 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™) を使用してインプリメントされるため、デバイスのブロック RAM リソースを使用しません。RAM32M は、1 つのスライスにインプリメントされます。含まれるのは、8 ビット書き込み、2 ビット読み出しのポート 1 つと、同じメモリからの 2 ビットの読み出しポートが 3 つです。これにより、バイト幅の書き込みと独立した 2 ビットの読み出しが可能な RAM ができます。DIA、DIB、DIC、および DID 入力すべてが同じデータ入力に接続されると、この RAM は読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 32x2 クワッドポートメモリになります。DID がグランドに接続される場合、DOD は使用されません。ADDRA、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 ADDR B 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 ADDR D 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	ADDR D 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	ADDR D で指定された書き込みデータ入力 (読み出し出力は ADDR B で指定)
DIC	入力	2	ADDR D 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 ADDR D 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
);

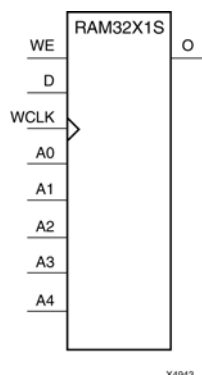
// End of RAM32M_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM32X1S

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



概要

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

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

論理表

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスエーション)

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

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

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

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

-- End of RAM32X1S_inst instantiation
```

Verilog 記述 (インスタンスエーション)

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

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

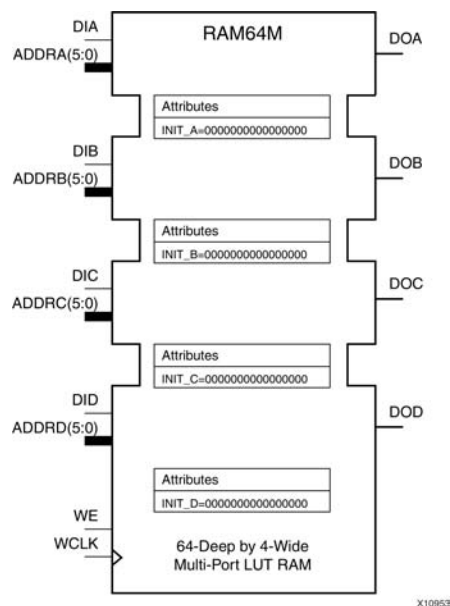
// End of RAM32X1S_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

RAM64M

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



概要

このデザイン エLEMENTは、64 ワード X 4 ビットのマルチポート RAM で、同期書き込みと非同期読み出し機能を備えています。この RAM は、デバイスの LUT (SelectRAM™ と呼ばれる) を使用してインプリメントされるため、ブロック RAM リソースを使用しません。RAM64M は、1 つのスライスにインプリメントされます。含まれるのは、4 ビット書き込み、1 ビット読み出しのポート 1 つと、同じメモリからの 1 ビットの読み出しポートが 3 つです。DIA、DIB、DIC、および DID 入力すべてが同じデータ入力に接続されると、この RAM は読み出し/書き込みポート 1 つ、独立した読み出しポート 3 つの 64x1 クワッド ポート メモリにできます。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 ADDRb で指定された読み出しポートのデータ出力
DOC	出力	1	アドレス バス ADDRC で指定された読み出しポートのデータ出力
DOD	出力	1	アドレス バス ADDRd で指定された読み出し/書き込みポートのデータ出力
DIA	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRA で指定)
DIB	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRb で指定)
DIC	入力	1	ADDRd で指定された書き込みデータ入力 (読み出し出力は ADDRC で指定)
DID	入力	1	アドレス バス ADDRd で指定された書き込みデータ入力
ADDRA	入力	6	読み出しアドレス バス A
ADDRb	入力	6	読み出しアドレス バス B
ADDRC	入力	6	読み出しアドレス バス C
ADDRd	入力	6	4 ビットのデータ書き込みポート、1 ビットのデータ読み出しポートのアドレス バス D
WE	入力	1	ライト イネーブル
WCLK	入力	1	ライト クロック (読み出しは非同期)

デザインの入力方法

インスタンス化	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 ADDRd は読み出し/書き込みアドレスに、5 ビット バス ADDRA、ADDRb、ADDRC は読み出しアドレスに接続する必要があります。オプションで INIT_A、INIT_B、INIT_C、INIT_D 属性を使用すると、各ポートの初期メモリ内容を 64 ビット (16 進数) で指定できます。RAM の INIT 値は、ADDRy[z] = INIT_y[z] で計算されます。

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

使用可能な属性

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

VHDL 記述 (インスタンスエーション)

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

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

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

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

Verilog 記述 (インスタンスレーション)

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

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

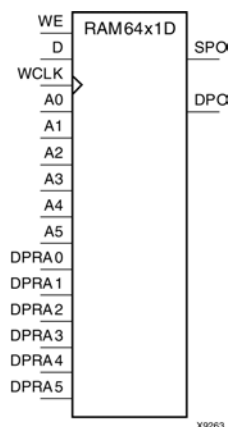
// End of RAM64M_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギャブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM64X1D

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



概要

このデザイン エLEMENTは 64 ワード X 1 ビットのデュアル ポート SRAM で、同期書き込み機能を備えています。デバイスには、読み出しアドレス (DPRA5 ~ DPRA0) と書き込みアドレス (A5 ~ A0) の独立した 2 種類のアドレス ポートがあります。この 2 種類のアドレス ポートは完全に非同期です。読み出しアドレスによって出力ピン (DPO) に出力される値が指定され、書き込みアドレスによって書き込みを行う位置が指定されます。ライト イネーブル (WE) が Low の場合、ライト クロック (WCLK) の遷移は無視され、RAM に格納されている値は変化しません。

WE が High になると、WCLK が Low から High に切り替わるときに、データ入力 (D) の値が 6 ビットの書き込みアドレス (A0 ~ A5) で選択されたワードにロードされます。書き込みを正しく行うには、WCLK が Low から High に切り替わる前に、書き込みアドレスとデータ入力の値を安定させる必要があります。WCLK はデフォルトではアクティブ High ですが、インバータを使用してアクティブ Low にすることもできます。WCLK の入力ネットに配置されたインバータは、RAM ブロック内に組み込まれます。

SPO 出力には、A5 ~ A0 で指定されたメモリ セルの値が出力されます。DPO 出力には、DPRA5 ~ DPRA0 で指定されたメモリ セルの値が出力されます。

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

論理表

入力			出力	
WE (モード)	WCLK	D	SPO	DPO
0 (読み出し)	X	X	data_a	data_d
1 (読み出し)	0	X	data_a	data_d
1 (読み出し)	1	X	data_a	data_d
1 (書き込み)	↑	D	D	data_d
1 (読み出し)	↓	X	data_a	data_d
data_a = A5 ~ A0 で指定されたワード				
data_d = DPRA5 ~ DPRA0 で指定されたワード				

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンス化)

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

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

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

RAM64X1D_inst : RAM64X1D
generic map (
    INIT => X"0000000000000000")
port map (
    DPO => DPO,      -- Read-only 1-bit data output
    SPO => SPO,      -- R/W 1-bit data output
    A0 => A0,        -- R/W address[0] input bit
    A1 => A1,        -- R/W address[1] input bit
    A2 => A2,        -- R/W address[2] input bit
    A3 => A3,        -- R/W address[3] input bit
    A4 => A4,        -- R/W address[4] input bit
    A5 => A5,        -- R/W address[5] input bit
    D => D,          -- Write 1-bit data input
    DPRA0 => DPRA0,  -- address[0] input bit
    DPRA1 => DPRA1,  -- Read-only address[1] input bit
    DPRA2 => DPRA2,  -- Read-only address[2] input bit
    DPRA3 => DPRA3,  -- Read-only address[3] input bit
    DPRA4 => DPRA4,  -- Read-only address[4] input bit
    DPRA5 => DPRA5,  -- Read-only address[5] input bit
    WCLK => WCLK,    -- Write clock input
    WE => WE         -- Write enable input
);

-- End of RAM64X1D_inst instantiation
```


Verilog 記述 (インスタンス化)

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

RAM64X1D #(
    .INIT(64'h0000000000000000) // Initial contents of RAM
) RAM64X1D_inst (
    .DPO(DPO),           // Read-only 1-bit data output
    .SPO(SPO),           // R/W 1-bit data output
    .A0(A0),             // R/W address[0] input bit
    .A1(A1),             // R/W address[1] input bit
    .A2(A2),             // R/W address[2] input bit
    .A3(A3),             // R/W address[3] input bit
    .A4(A4),             // R/W address[4] input bit
    .A5(A5),             // R/W address[5] input bit
    .D(D),               // Write 1-bit data input
    .DPRA0(DPRA0),       // Read-only address[0] input bit
    .DPRA1(DPRA1),       // Read-only address[1] input bit
    .DPRA2(DPRA2),       // Read-only address[2] input bit
    .DPRA3(DPRA3),       // Read-only address[3] input bit
    .DPRA4(DPRA4),       // Read-only address[4] input bit
    .DPRA5(DPRA5),       // Read-only address[5] input bit
    .WCLK(WCLK),         // Write clock input
    .WE(WE)              // Write enable input
);

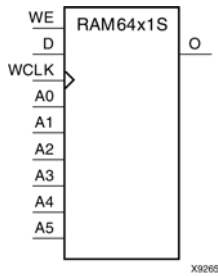
// End of RAM64X1D_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAM64X1S

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



概要

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

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

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

論理表

モード選択を次の論理表に示します。

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

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンスレーション)

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

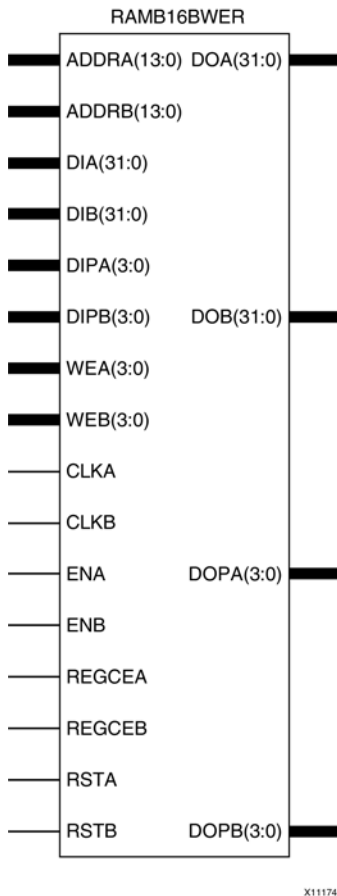
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

RAMB16BWER

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



概要

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

ポートの説明

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

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

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

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

デザインの入力方法

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

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

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

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A	整数	0、1、2、4、9、18、36	0	ポート A のデータ幅を指定
DATA_WIDTH_B	整数	0、1、2、4、9、18、36	0	ポート B のデータ幅を指定
DOA_REG	整数	0、1	0	RAM の出力レジスタを使用するか、バイパス接続するか指定します。
DOB_REG	整数	0、1	0	RAM の出力レジスタを使用するか、バイパス接続するか指定します。
EN_RSTRAM_A	文字列	TRUE、FALSE	TRUE	出力ラッチでリセットを有効にするかどうかを指定します。
EN_RSTRAM_B	文字列	TRUE、FALSE	TRUE	出力ラッチでリセットを有効にするかどうかを指定します。
INIT_A	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	コンフィギュレーション後のポート A の出力の初期値を指定します。
INIT_B	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。
INIT_FILE	文字列	0 ビット文字列	NONE	ブロック RAM の初期値を指定するためのファイル名を指定します。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータメモリ アレイの初期内容を指定します。
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティメモリ アレイの初期内容を指定します。
RST_PRIORITY_A	文字列	CE、SR	CE	RSTA ピンか ENA ピン (ラッチ モード) を指定します。または REGCEA ピン (出力レジスタ モード) が優先されます。
RST_PRIORITY_B	文字列	CE、SR	CE	RSTB ピンか ENB ピン (ラッチ モード) を指定します。または REGCEB ピン (出力レジスタ モード) が優先されます。
RSTTYPE	文字列	SYNC、ASYNC	SYNC	リセット、同期、または非同期を指定します。
SIM_COLLISION_CHECK	文字列	ALL、 GENERATE_X_ONLY、 WARNING_ONLY、 NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。

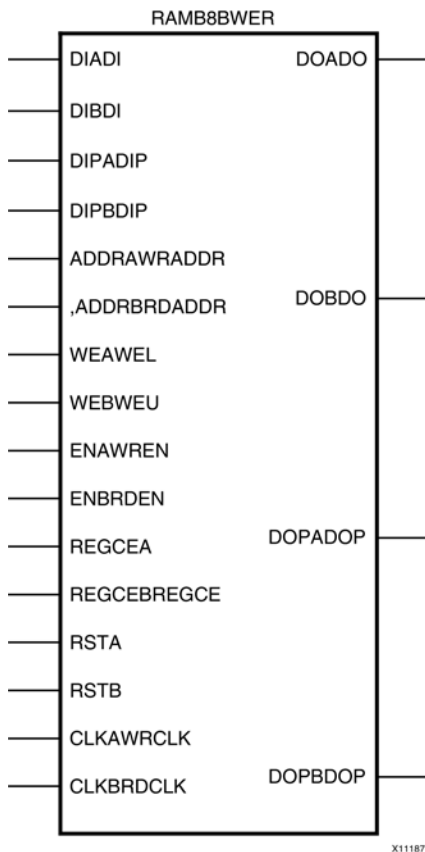
属性	タイプ	値	デフォルト	説明
				メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。
SRVAL_A	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	リセット信号 (RSTA) がアサートされたときのポート A の出力値を指定
SRVAL_B	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	リセット信号 (RSTB) がアサートされたときのポート B の出力値を指定します。
WRITE_MODE_A	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	ポート A の出力ビヘイビアを指定します。 <ul style="list-style-type: none"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。
WRITE_MODE_B	文字列	WRITE_FIRST、 READ_FIRST、 NO_CHANGE	WRITE_FIRST	ポート B の出力ビヘイビアを指定します。 <ul style="list-style-type: none"> WRITE_FIRST に設定すると、書き込まれた値が出力ポートに出力されます。 READ_FIRST に設定すると、そのメモリロケーションに直前に格納されていた値が出力ポートに出力されます。 NO_CHANGE に設定すると、出力ポートから直前に出力された値が保持されます。

詳細情報

- ・ [Spartan-6 FPGA ブロック RAM ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

RAMB8BWER

プリミティブ : 8K-bit Data and 1K-bit Parity Configurable Synchronous Dual Port Block RAM with Optional Output Registers



概要

複数のブロックRAMメモリが含まれていますが、これは汎用 9kb データ + 1kb パリティ RAM/ROM メモリとしてコンフィギュレーションできます。これらのブロック RAM には、大量のオンチップ データを高速かつ柔軟に格納できます。このコンポーネントは、1 ビット X 8K ワード ~ 36 ビット X 512 ワードのシングル ポート RAM またはデュアル ポート RAM としてコンフィギュレーションできます。完全なデュアル ポート モードでは、1 ビット X 8K ワードから 18 ビット X 1K ワードがサポートされています。コンポーネントに供給されるクロックに完全に同期して、読み出しと書き込みが同時に実行されます。ただし、ポート A とポート B は互いに独立しており、同じメモリアレイにアクセスする間は非同期になります。データ幅の広いモードでコンフィギュレーションすれば、バイト イネーブル書き込み操作が可能です。この RAM には、コンフィギュレーション可能な出力レジスタもあり、読み出し操作中 1 クロック サイクルのレイテンシが発生するときに RAM の clock-to-out タイムを向上させることができます。

ポートの説明

ポート名	方向	幅	機能
DOADO	出力	16	ポート A のデータ出力バス
DOBDO	出力	16	ポート B のデータ出力バス

ポート名	方向	幅	機能
DOPADOP	出力	2	完全なデュアルポートモードでは、これは A ポートパリティバス出力です。シンプルデュアルポートモードでは、下位ビットのパリティバス出力です。DATA_WIDTH が 36 の場合、DOPBDOP は上位パリティビットです。
DOPBDOP	出力	2	完全なデュアルポートモードでは、これは B ポートパリティバス出力です。シンプルデュアルポートモードでは、DATA_WIDTH が 36 の場合、これは上位パリティビットのパリティバス出力です。DOPADOP は下位パリティビットです。
DIADI	入力	16	完全なデュアルポートモードでは A ポートデータバスで、シンプルデュアルポートモードでは下位ビットのデータバスです。DATA_WIDTH が 36 の場合、DIBDI は上位パリティビット [16:31] です。
DIBDI	入力	16	完全なデュアルポートモードでは B ポートデータバスで、シンプルデュアルポートモードでは、DATA_WIDTH が 36 の場合、上位ビット [16:31] のデータバスです。DIADI は下位アドレスビットです。
DIPADIP	入力	2	完全なデュアルポートモードでは、これは A ポートパリティバス入力です。シンプルデュアルポートモードでは、下位ビットのパリティバス入力です。DATA_WIDTH が 36 の場合、DIPBDIP は上位パリティビットです。
DIPBDIP	入力	2	完全なデュアルポートモードでは、これは B ポートパリティバス入力です。シンプルデュアルポートモードでは、DATA_WIDTH が 36 の場合、これは上位パリティビットのパリティバス出力です。DIPADIP は下位パリティビットです。
ADDRAWRADDR	入力	13	完全なデュアルポートモードでは A ポートアドレスバスで、シンプルデュアルポートモードでは書き込みポートのアドレスバスです。
ADDRBRDADDR	入力	13	完全なデュアルポートモードでは B ポートアドレスバスで、シンプルデュアルポートモードでは読み出しポートのアドレスバスです。
WEAWEL	入力	2	完全なデュアルポートモードでは、これは A ポートライトイネーブルです。シンプルデュアルポートモードでは、下位ビットのライトイネーブルです。
WEBWEU	入力	2	完全なデュアルポートモードでは、これは B ポートライトイネーブルです。シンプルデュアルポートモードでは、DATA_WIDTH が 36 の場合、上位ビットのライトイネーブルです。
ENAWREN	入力	1	完全なデュアルポートモードでは、これは A ポートイネーブルです。シンプルデュアルポートモードではライトイネーブルです。
ENBRDEN	入力	1	完全なデュアルポートモードでは、これは B ポートイネーブルです。シンプルデュアルポートモードではリードイネーブルです。
REGCEA	入力	1	完全なデュアルポートモードでは、これは A ポートクロックイネーブルです。シンプルデュアルポートモードでは、未使用でロジック 0 に接続されている必要があります。
REGCEBREGCE	入力	1	完全なデュアルポートモードでは、これは B ポートクロックイネーブルです。シンプルデュアルポートモードではリードクロックイネーブルです。
RSTA	入力	1	ポート A の出力レジスタのセット/リセット。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。
RSTB	入力	1	ポート B の出力レジスタのセット/リセット。このリセットは、RSTTYPE 属性の値に従い同期または非同期にコンフィギュレーションできます。

ポート名	方向	幅	機能
CLKAWRCLK	入力	1	完全なデュアル ポート モードでは、これは A ポートクロック入力です。シンプル デュアル ポート モードではライト モードクロック入力です。
CLKBRDCLK	入力	1	完全なデュアル ポート モードでは、これは B ポートクロック入力です。シンプル デュアル ポート モードではリード クロック入力です。

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	不可

使用可能な属性

属性	タイプ	値	デフォルト	説明
DATA_WIDTH_A	整数	0、1、2、4、9、18、36	0	ポート A のコンフィギュレーション可能なデータ幅を指定。データ幅 36 は RAM_MODE=SDP でのみ使用可能です。
DATA_WIDTH_B	整数	0、1、2、4、9、18、36	0	ポート B のコンフィギュレーション可能なデータ幅を指定。データ幅 36 は RAM_MODE=SDP でのみ使用可能です。
DOA_REG	整数	0、1	0	RAM の出力レジスタを使用するか、バイパス接続するか指定します。
DOB_REG	整数	0、1	0	RAM の出力レジスタを使用するか、バイパス接続するか指定します。
EN_RSTRAM_A	文字列	TRUE、FALSE	TRUE	出力ラッチでリセットを有効にするかどうかを指定します。
EN_RSTRAM_B	文字列	TRUE、FALSE	TRUE	出力ラッチでリセットを有効にするかどうかを指定します。
INIT_A	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	コンフィギュレーション後のポート A の出力の初期値を指定します。シンプル デュアル ポート モードでは、DATA_WIDTH が 36 のとき INIT_B は上位ビット [31:16] で、INIT_A は下位ビットになります。
INIT_B	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	コンフィギュレーション後のポート B の出力の初期値を指定します。シンプル デュアル ポート モードでは、DATA_WIDTH が 36 のとき INIT_B は上位ビット [31:16] で、INIT_A は下位ビットになります。
INIT_FILE	文字列	0 ビット文字列	NONE	ブロック RAM の初期値を指定するためのファイル名を指定します。
INIT_00 ~ INIT_3F	16 進数	256 ビット値	すべてゼロ	16kb のデータメモリ アレイの初期内容を指定します。
INITP_00 ~ INITP_07	16 進数	256 ビット値	すべてゼロ	2kb のパリティメモリ アレイの初期内容を指定します。

属性	タイプ	値	デフォルト	説明
RAM_MODE	文字列	TDP、SDP	TDP	ブロック RAM のポートコンフィギュレーションを定義します。完全なデュアルポート (TDP) では、各ポートを読み出し/書き込みの両方に使用することができます。シンプルデュアルポート (SDP) では、ポート A が書き込みポート、ポート B が読み出しポートに割り当てられます。
RST_PRIORITY_A	文字列	CE、SR	CE	RSTA ピンか ENA ピン (ラッチ モード) を指定します。または REGCEA ピン (出力レジスタ モード) が優先されます。
RST_PRIORITY_B	文字列	CE、SR	CE	RSTB ピンか ENB ピン (ラッチ モード) を指定します。または REGCEB ピン (出力レジスタ モード) が優先されます。
RSTTYPE	文字列	SYNC、ASYN	SYNC	リセット、同期、または非同期を指定します。
SIM_COLLISION_CHECK	文字列	ALL、 GENERATE_X_ONLY、 WARNING_ONLY、 NONE	ALL	<p>メモリの競合が発生した場合にシミュレーションの動作を変更できます。</p> <ul style="list-style-type: none"> ALL に設定すると、警告メッセージが出力され、関連する出力およびメモリの値が不定 (X) になります。 WARNING_ONLY に設定すると、警告メッセージのみが出力され、関連する出力およびメモリの値はそのまま保持されます。 GENERATE_X_ONLY に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値が不定 (X) になります。 NONE に設定すると、警告メッセージは出力されず、関連する出力およびメモリの値はそのまま保持されます。 <p>メモ: ALL 以外の値に設定すると、シミュレーション中にデザインの問題を認識できなくなるため、この値を変更する場合は注意が必要です。</p>
SRVAL_A	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	リセット信号 (RSTA) がアサートされたときのポート A の出力値を指定。シンプルデュアルポートモードでは、DATA_WIDTH が 36 のとき SRVAL_B は上位ビット [31:16] で、SRVAL_A は下位ビットになります。
SRVAL_B	16 進数	36'h000000000 ~ 36'h68719476735	すべてゼロ	リセット信号 (RSTB) がアサートされたときのポート B の出力値を指定します。シンプルデュアルポートモードでは、DATA_WIDTH が 36 のとき SRVAL_B は上位ビット [31:16] で、SRVAL_A は下位ビットになります。

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

VHDL 記述 (インスタンスエーション)

```
-- RAMB8BWER: 8K-bit Data and 1K-bit Parity Configurable Synchronous Block RAM
--           Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

RAMB8BWER_inst : RAMB8BWER
generic map (
    DATA_WIDTH_A => 0,
    DATA_WIDTH_B => 0,
    DOA_REG => 0,
    DOB_REG => 0,
    EN_RSTRAM_A => "TRUE",
    EN_RSTRAM_B => "TRUE",

    -- INITP_00 to INITP_03: Allows specification of the initial contents of the 1KB parity data memory
    -- array.
    INITP_00 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INITP_01 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INITP_02 => X"0000000000000000000000000000000000000000000000000000000000000000",
    INITP_03 => X"0000000000000000000000000000000000000000000000000000000000000000",
    -- INIT_00 to INIT_1F: Allows specification of the initial contents of the 8KB 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",
    -- 0, 1, 2, 4, 9, 18, or
    -- 36
    -- 0, 1, 2, 4, 9, 18, or
    -- 36
    -- Optional output
    -- register on A port (0
    -- or 1)
    -- Optional output
    -- register on B port (0
    -- or 1)
    -- Enable/disable A port
    -- RST
    -- Enable/disable B port
    -- RST
    -- RST
```

253

```
-- End of RAMB8BWER_inst instantiation
```

Verilog 記述 (インスタンスエーション)

```
// RAMB8BWER: 8K-bit Data and 1K-bit Parity Configurable Synchronous Block RAM
// Spartan-6
// Xilinx HDL Language Template, version 11.1

RAMB8BWER #(
    .DATA_WIDTH_A(0), // 0, 1, 2, 4, 9, 18,
                        // or 36
    .DATA_WIDTH_B(0), // 0, 1, 2, 4, 9, 18,
                        // or 36
    .DOA_REG(0), // Optional output
                  // register on A port
                  // (0 or 1)
    .DOB_REG(0), // Optional output
                  // register on B port
                  // (0 or 1)
    .EN_RSTRAM_A("TRUE"), // Enable/disable A
                           // port RST
    .EN_RSTRAM_B("TRUE"), // Enable/disable B
                           // port RST

    // INITP_00 to INITP_03: Allows specification of the initial contents of the 1KB parity data memory
    // array.
    .INITP_00(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_01(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_02(256'h0000000000000000000000000000000000000000000000000000000000000000),
    .INITP_03(256'h0000000000000000000000000000000000000000000000000000000000000000),
    // INIT_00 to INIT_1F: Allows specification of the initial contents of the 8KB data memory array.
    .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),
    .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_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
    .RSTTYPE("SYNC"), // SYNC or ASYNC
);
```

```

.RST_PRIORITY_A("CE"),
.RST_PRIORITY_B("CE"),
.SIM_COLLISION_CHECK("ALL"),

.SRVAL_A(18'h00000),
.SRVAL_B(18'h00000),
.WRITE_MODE_A("WRITE_FIRST"),
.WRITE_MODE_B("WRITE_FIRST")

)
RAMB8BWER_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
.ADDRAWRADDR(ADDRWRADDR), // 13-bit A port address/Write address input
.ADDRBRDADDR(ADDRBRDADDR), // 13-bit B port address/Read address input
.CLKAWRCLK(CLKAWRCLK),    // 1-bit A port clock/Write clock input
.CLKBRDCLK(CLKBRDCLK),    // 1-bit B port clock/Read 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
.ENAWREN(ENAWREN),        // 1-bit A port enable/Write enable input
.ENBRDEN(ENBRDEN),        // 1-bit B port enable/Read enable input
.REGCEA(REGCEA),          // 1-bit A port register enable input
.REGCEBREGCE(REGCEBREGCE), // 1-bit B port register enable/Register enable input
.RSTA(RSTA),              // 1-bit A port set/reset input
.RSTBRST(RSTBRST),        // 1-bit B port set/reset input
.WEAWEL(WEAWEL),          // 2-bit A port write enable input
.WEBWEU(WEBWEU)           // 2-bit B port write enable input
);

// End of RAMB8BWER_inst instantiation

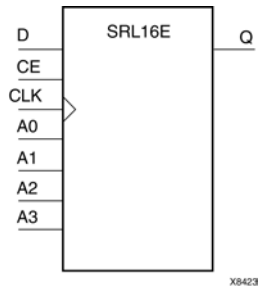
```

詳細情報

- [Spartan-6 FPGA ブロック RAM ユーザー ガイド](#)
- [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

SRL16E

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



概要

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

シフトレジスタの長さは、固定することも、変動させることもできます。

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

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

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

論理表

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

ポートの説明

ポート名	方向	幅	機能
Q	出力	1	シフトレジスタ データ出力
D	入力	1	シフトレジスタ データ入力
CLK	入力	1	クロック
CE	入力	1	アクティブ High のクロック イネーブル
A	入力	4	SRL のワード数のダイナミック選択 ・ A=0000 ==> 1 ビット シフト長 ・ A=1111 ==> 16 ビット シフト長

デザインの入力方法

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

使用可能な属性

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

VHDL 記述 (インスタンシエーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

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

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

-- End of SRL16E_inst instantiation

```

Verilog 記述 (インスタンスレーション)

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

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

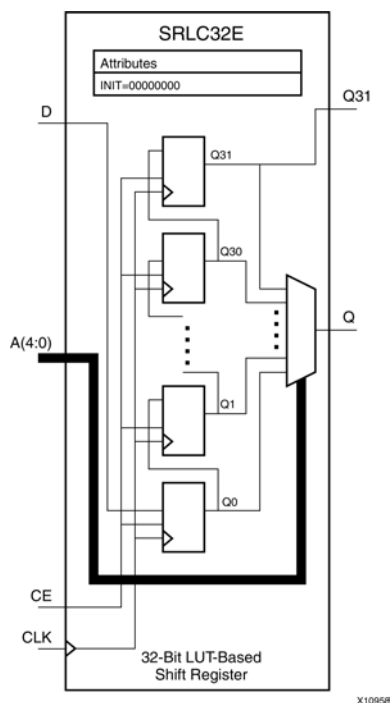
// End of SRL16E_inst instantiation
```

詳細情報

[Spartan-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 (インスタンス化)

```
// 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
);

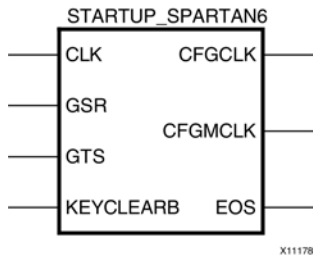
// End of SRLC32E_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュラブル ロジック ブロック ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート : DC 特性およびスイッチ特性](#)

STARTUP_SPARTAN6

プリミティブ : Spartan®-6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface



概要

このデザイン エLEMENTは、デバイス ピンおよびグローバル セット/リセット (GSR) 信号へのロジックにインターフェイスするため、またはグローバル トライステート (GTS) 専用配線のために使用します。また、デバイスのコンフィギュレーションの終了時に、スタートアップ シーケンスに別のクロックを指定することもできます。

ポートの説明

ポート名	タイプ	幅	機能
CFGCLK	出力	1	コンフィギュレーション ロジックのメイン 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。固定周波数 : 平常状態で 50MHz
CLK	入力	1	コンフィギュレーション スタートアップ シーケンス クロック (StartClk) 配線への入力接続。
EOS	出力	1	スタートアップの終わり
GSR	入力	1	GSR 配線への入力接続
GTS	入力	1	GTS 配線への入力接続
KEYCLEARB	入力	1	セット時のクリア BBR キー。グリッチを防ぐため KEYCLEAR 機能を有効にするには、この信号を 200ns (4 クロック サイクル) 間 Low に保持します。

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

専用の GSR 回路はソース ピンまたはロジックを GSR ピンに接続すれば使用できますが、このコンポーネントの GSR 回路の使用には、特別の注意が必要です。GSR ネットのスキューは確定できないので、セット/リセット信号の一般配線を使用して、配線遅延とスキューをデザインのタイミング解析の一部として計算できるようにするか、クロック サイクルのリリース時にスキューが回路の動作を邪魔しないような回避策をとってください。

同様に、専用のグローバル トライステートが使用される場合は、適切なソース ピンまたはロジックをこのプリミティブの GTS 入力ピンに接続します。コンフィギュレーションのスタートアップ シーケンスのクロックを指定するには、デザインからのクロックをこのデザイン エLEMENTの CLK ピンに接続します。

VHDL 記述 (インスタンスレーション)

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

```

Library UNISIM;
use UNISIM.vcomponents.all;

-- STARTUP_SPARTAN6: sp6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface
--                      Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

STARTUP_SPARTAN6_inst : STARTUP_SPARTAN6
generic map (
)
port map (
    CFGCLK => CFGCLK,          -- 1-bit Configuration logic main clock output.
    CFGMCLK => CFGMCLK,        -- 1-bit Configuration internal oscillator clock output. Fixed frequency: 50MHz
                                -- at typical condition.

    EOS => EOS,                -- 1-bit Indicates end of startup.
    CLK => CLK,                -- 1-bit Input connection to the configuration startup sequence clock (GSR)
                                -- routing.

    GSR => GSR,                -- 1-bit Input connection to the global set / reset (GSR) routing.
    GTS => GTS,                -- 1-bit Input connection to the global 3-state (GTS) routing.
    KEYCLEARB => KEYCLEARB     -- 1-bit Clear BBR key when it is set. Note that this signal needs to stay low
                                -- for 200ns (4 clock cycles) to enable KEYCLEAR function (to prevent glitches).
);

-- End of STARTUP_SPARTAN6_inst instantiation

```

Verilog 記述 (インスタンスレーション)

```

// STARTUP_SPARTAN6: sp6 Global Set/Reset, Global 3-State and Configuration Start-Up Clock Interface
//                      Spartan-6
// Xilinx HDL Language Template, version 11.1

STARTUP_SPARTAN6 STARTUP_SPARTAN6_inst (
    .CFGCLK(CFGCLK),          // 1-bit Configuration logic main clock output.
    .CFGMCLK(CFGMCLK),        // 1-bit Configuration internal oscillator clock output. Fixed frequency: 50MHz at
                                // typical condition.

    .EOS(EOS),                // 1-bit Indicates end of startup.
    .CLK(CLK),                // 1-bit Input connection to the configuration startup sequence clock (GSR)
                                // routing.

    .GSR(GSR),                // 1-bit Input connection to the global set / reset (GSR) routing.
    .GTS(GTS),                // 1-bit Input connection to the global 3-state (GTS) routing.
    .KEYCLEARB(KEYCLEARB)     // 1-bit Clear BBR key when it is set. Note that this signal needs to stay low for
                                // 200ns (4 clock cycles) to enable KEYCLEAR function (to prevent glitches).
);

// End of STARTUP_SPARTAN6_inst instantiation

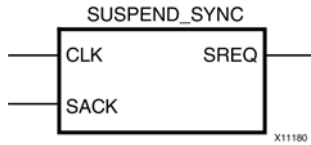
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)

SUSPEND_SYNC

プリミティブ : Suspend Mode Access



概要

このデザイン エLEMENTは、一時停止モードを使用しているアプリケーションのデザインを同期化する機能を拡張します。同期化が必要なクロックドメインが数個あっても、一時停止モードを開始するトリガの同期化を行うのに 3 ピン インターフェイスが使用されます。SREQ 出力は、一時停止モードを開始するためデバイスに要求を送信し、SACK は、デバイスで開始準備が整っていることを確認します。SACK ピンは CLK ピンに同期します。

ポートの説明

ポート名	タイプ	幅	機能
CLK	入力	1	ユーザー クロック
SACK	入力	1	一時停止確認応答で CLK に同期します。
SREQ	出力	1	SUSPEND ピンからのリクエストを一時停止します。

デザインの入力方法

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

VHDL 記述 (インスタンス化)

```
-- SUSPEND_SYNC: Suspend Mode Access
--               Spartan-6
-- Xilinx HDL Libraries Guide, version 11.2

SUSPEND_SYNC_inst : SUSPEND_SYNC
generic map (
)
port map (
  SREQ => SREQ, -- 1-bit Suspend request from either SUSPEND pin or SSPI SUSPEND command.
  CLK  => CLK,  -- 1-bit User clock
  SACK => SACK  -- 1-bit SUSPEND acknowledgement; synchronous to CLK
);

-- End of SUSPEND_SYNC_inst instantiation
```


Verilog 記述 (インスタンス化)

```
// SUSPEND_SYNC: Suspend Mode Access
//                               Spartan-6
// Xilinx HDL Language Template, version 11.1

SUSPEND_SYNC SUSPEND_SYNC_inst (
    .SREQ(SREQ), // 1-bit Suspend request from either SUSPEND pin or SSPI SUSPEND command.
    .CLK(CLK),   // 1-bit User clock
    .SACK(SACK)  // 1-bit SUSPEND acknowledgement; synchronous to CLK
);

// End of SUSPEND_SYNC_inst instantiation
```

詳細情報

- ・ [Spartan-6 FPGA コンフィギュレーション ユーザー ガイド](#)
- ・ [Spartan-6 FPGA データシート: DC 特性およびスイッチ特性](#)