

Arora V プログラマブル IO(GPIO) ユーザーガイド

UG304-1.0J, 2023/04/20

著作権について(2023)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

GOWIN高云、Gowin、及び GOWINSEMI は、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

免責事項

当社は、GOWINSEMI Terms and Conditions of Sale (GOWINSEMI取引条件) に規定されている内容を除き、(明示的か又は黙示的に拘わらず) いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

バージョン履歴

日付	バージョン	説明
2023/04/20	1.0J	初版。

目次

目次	i
図一覧	iv
表一覧	vi
1 本マニュアルについて	1
1.1 マニュアル内容	1
1.2 関連ドキュメント	1
1.3 用語、略語	1
1.4 テクニカル・サポートとフィードバック	2
2 GPIO の概要	3
3 入出力バッファ	5
3.1 GPIO 規格	5
3.2 GPIO のバンキング・スキーム	5
3.3 電源供給の要件	5
3.3.1 LVCMOS バッファの構成	6
3.3.2 差動バッファの構成	6
3.4 エミュレート差動回路終端方式	7
3.4.1 エミュレート LVDS	7
3.4.2 エミュレート LVPECL	7
3.4.3 エミュレート RSDS	7
3.4.4 エミュレート BLVDS	7
3.5 GPIO の構成	8
3.5.1 位置	8
3.5.2 レベル規格	8
3.5.3 ドライブ強度	8
3.5.4 プルアップ/プルダウン	8
3.5.5 リファレンス電圧	9
3.5.6 ヒステリシス	9
3.5.7 オープンドレイン	9

3.5.8 シングルエンド終端抵抗.....	9
3.5.9 差動終端抵抗.....	9
3.5.10 PCI Clamp	9
3.5.11 プルアップ/プルダウンの強度.....	9
3.6 GPIO プリミティブ	10
3.6.1 IBUF	10
3.6.2 OBUF	11
3.6.3 TBUF	12
3.6.4 IOBUF	13
3.6.5 LVDS Input Buffer.....	14
3.6.6 LVDS Output Buffer	15
3.6.7 LVDS Tristate Buffer.....	17
3.6.8 LVDS Inout Buffer.....	19
3.6.9 MIPI_IBUF.....	21
3.6.10 MIPI_OBUF_A.....	23
3.6.11 I3C_IOBUF	25
3.6.12 IBUF_R.....	26
3.6.13 IOBUF_R.....	27
3.6.14 ELVDS_IOBUF_R	29
3.6.15 TLVDS_IBUF_ADC	30
4 入出力ロジック	33
4.1 SDR モード	34
4.1.1 DFFSE.....	34
4.1.2 DFFRE	36
4.1.3 DFFPE.....	37
4.1.4 DFFCE	39
4.1.5 DLCE.....	41
4.1.6 DLPE.....	43
4.2 DDR モードの入出力ロジック	45
4.2.1 IDDR.....	45
4.2.2 IDDR_C	47
4.2.3 IDES4	49
4.2.4 IDES8	51
4.2.5 IDES10	55
4.2.6 IVIDEO	57
4.2.7 IDES16.....	60
4.2.8 IDDR_MEM	63
4.2.9 IDES4_MEM.....	66
4.2.10 IDES8_MEM.....	69

4.2.11 IDER14	72
4.2.12 IDER32	75
4.2.13 OSIDER32	80
4.3 DDR モードの出力ロジック	84
4.3.1 ODDR	84
4.3.2 ODDRC	88
4.3.3 OSER4	90
4.3.4 OSER8	94
4.3.5 OSER10	98
4.3.6 OVIDEO	100
4.3.7 OSER16	103
4.3.8 ODDR_MEM	106
4.3.9 OSER4_MEM	109
4.3.10 OSER8_MEM	113
4.4 遅延モジュール	118
4.4.1 IODELAY	118

図一覧

図 2-1 入出力ブロックの構造	4
図 3-1 LVDS25E の外部終端	7
図 3-2 LVPECL の外部終端	7
図 3-3 RSDS の外部終端	7
図 3-4 BLVDS の外部終端	8
図 3-5 IBUF のポート図	10
図 3-6 OBUF のポート図	11
図 3-7 TBUF のポート図	12
図 3-8 IOBUF のポート図	13
図 3-9 TLVDS_IBUF のポート図	14
図 3-10 TLVDS_OBUF/ELVDS_OBUF のポート図	15
図 3-11 TLVDS_TBUF/ELVDS_TBUF のポート図	17
図 3-12 TLVDS_IOBUF/ELVDS_IOBUF のポート図	19
図 3-13 MIPI_IBUF のポート図	21
図 3-14 MIPI_OBUF_A のポート図	24
図 3-15 I3C_IOBUF のポート図	25
図 3-16 IBUF_R のポート図	26
図 3-17 IOBUF_R のポート図	28
図 3-18 ELVDS_IOBUF_R のポート図	29
図 3-19 TLVDS_IBUF_ADC のポート図	31
図 4-1 入出力ロジックの説明図 –出力	33
図 4-2 入出力ロジックの説明図 –入力	33
図 4-3 DFFSE のポート図	34
図 4-4 DFFRE のポート図	36
図 4-5 DFFPE のポート図	38
図 4-6 DFFCE のポート図	39
図 4-7 DLCE のポート図	41
図 4-8 DLPE のポート図	43
図 4-9 IDDR のブロック図	45
図 4-10 IDDR のタイミング図	46
図 4-11 IDDR のポート図	46
図 4-12 IDDR のポート図	47
図 4-13 CALIB のタイミングの例	49

図 4-14 IDER4 のポート図	49
図 4-15 IDER8 のポート図	52
図 4-16 IDER10 のポート図	55
図 4-17 IVIDEO のポート図	58
図 4-18 IDER16 のポート図	60
図 4-19 IDDR_MEM のポート図	64
図 4-20 IDER4_MEM のポート図	66
図 4-21 IDER8_MEM のポート図	69
図 4-22 IDER14 のポート図	72
図 4-23 IDER32 のポート図	76
図 4-24 OSIDER32 のポート図	81
図 4-25 ODDR のブロック図	84
図 4-26 ODDR のタイミング図	85
図 4-27 ODDR のポート図	85
図 4-28 ODDRC のブロック図	88
図 4-29 ODDRC のポート図	88
図 4-30 OSER4 のブロック図	91
図 4-31 OSER4 のポート図	91
図 4-32 OSER8 のブロック図	94
図 4-33 OSER8 のポート図	95
図 4-34 OSER10 のポート図	98
図 4-35 OVIDEO のポート図	101
図 4-36 OSER16 のポート図	103
図 4-37 ODDR_MEM のブロック図	106
図 4-38 ODDR_MEM のポート図	106
図 4-39 OSER4_MEM のブロック図	109
図 4-40 OSER4_MEM のポート図	110
図 4-41 OSER8_MEM のブロック図	114
図 4-42 OSER8_MEM のポート図	114
図 4-43 IODELAY のポート図	119

表一覧

表 1-1 用語、略語.....	1
表 3-1 IBUF のポートの説明	10
表 3-2 OBUF のポートの説明	11
表 3-3 TBUF のポートの説明	12
表 3-4 IOBUF のポートの説明	13
表 3-5 TLVDS_IBUF のポートの説明.....	14
表 3-6 TLVDS_OBUF/ELVDS_OBUF のポートの説明	16
表 3-7 TLVDS_TBUF/ELVDS_TBUF のポートの説明	17
表 3-8 TLVDS_IOBUF/ELVDS_IOBUF のポートの説明	20
表 3-9 MIPI_IBUF のポートの説明.....	21
表 3-10 MIPI_OBUF_A 対応デバイス	23
表 3-11 MIPI_OBUF_A のポートの説明.....	24
表 3-12 I3C_IOBUF のポート図.....	25
表 3-13 IBUF_R のポートの説明	27
表 3-14 IOBUF_R のポートの説明.....	28
表 3-15 ELVDS_IOBUF_R のポートの説明	29
表 3-16 TLVDS_IBUF_ADC 対応デバイス.....	30
表 3-17 TLVDS_IBUF_ADC のポートの説明	31
表 4-1 DFFSE のポートの説明	34
表 4-2 DFFSE のパラメータの説明	35
表 4-3 DFFRE のポートの説明	36
表 4-4 DFFRE のパラメータの説明	36
表 4-5 DFFPE のポートの説明	38
表 4-6 DFFPE のパラメータの説明	38
表 4-7 DFFCE のポートの説明	40
表 4-8 DFFCE のパラメータの説明	40
表 4-9 DLCE のポートの説明.....	41
表 4-10 DLCE のパラメータの説明	42
表 4-11 DLPE のポートの説明	44
表 4-12 DLPE のパラメータの説明.....	44
表 4-13 IDDR のポートの説明	46
表 4-14 IDDRC のポートの説明.....	47
表 4-15 IDES4 のポートの説明.....	50

表 4-16 IDES8 のポートの説明.....	52
表 4-17 IDES10 のポートの説明.....	55
表 4-18 IVIDEO のポートの説明.....	58
表 4-19 IDES16 のポートの説明.....	60
表 4-20 IDDR_MEM のポートの説明.....	64
表 4-21 IDES4_MEM のポートの説明.....	66
表 4-22 IDES8_MEM のポートの説明.....	69
表 4-23 IDES14 のポートの説明.....	73
表 4-24 IDES32 のポートの説明.....	76
表 4-25 OSIDES32 のポートの説明.....	81
表 4-26 OSIDES32 のパラメータの説明.....	82
表 4-27 ODDR のポートの説明.....	85
表 4-28 ODDR のパラメータの説明.....	85
表 4-29 ODDRC のポートの説明.....	89
表 4-30 ODDRC のパラメータの説明.....	89
表 4-31 OSER4 のポートの説明.....	91
表 4-32 OSER4 のパラメータの説明.....	92
表 4-33 OSER8 のポートの説明.....	95
表 4-34 OSER8 のパラメータの説明.....	95
表 4-35 OSER10 のポートの説明.....	98
表 4-36 OVIDEO のポートの説明.....	101
表 4-37 OSER16 のポートの説明.....	103
表 4-38 ODDR_MEM のポートの説明.....	107
表 4-39 ODDR_MEM のパラメータの説明.....	107
表 4-40 OSER4_MEM のポートの説明.....	110
表 4-41 OSER4_MEM のパラメータの説明.....	111
表 4-42 OSER8_MEM のポートの説明.....	114
表 4-43 OSER8_MEM のパラメータの説明.....	115
表 4-44 IODELAY のポートの説明.....	119
表 4-45 IODELAY のパラメータの説明.....	119

1 本マニュアルについて

1.1 マニュアル内容

このマニュアルは、Gowin セミコンダクターの Arora V FPGA 製品でサポートされる入出力バッファのレベル規格、バンキング・スキーム、および入出力ロジックの機能について説明します。また、ユーザーが GPIO を使いこなせるよう GPIO の構造と Gowin ソフトウェアの使用法も解説されています。

1.2 関連ドキュメント

GOWIN セミコンダクターの公式 Web サイト www.gowinsemi.com/ja から、以下の関連ドキュメントがダウンロード、参考できます：

- GW5AT シリーズ FPGA 製品データシート([DS981](#))
- GW5A シリーズ FPGA 製品データシート([DS1103](#))
- GW5AST シリーズ FPGA 製品データシート([DS1104](#))
- Gowin ソフトウェア ユーザーガイド([SUG100](#))

1.3 用語、略語

表 1-1 に、本マニュアルで使用される用語、略語、及びその意味を示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
Bus Keeper	Bus Keeper	バスキーパ(バスホールド・ラッチ)
CFU	Configurable Function Unit	コンフィギャラブル機能ユニット
CRU	Configurable Routing Unit	コンフィギャラブル配線ユニット
DDR	Double Data Rate	ダブル・データ・レート
DES	Deserializer	デシリアライザ

用語、略語	正式名称	意味
ELDO	Emulated LVDS Output	エミュレート LVDS 出力 (電圧出力)
GPIO	Gowin Programmable Input/Output	Gowin プログラマブル汎用 IO
IOB	Input/Output Block	入出力ブロック
IO Buffer	Input/Output Buffer	入出力バッファ
IO Logic	Input/Output Logic	入出力ロジック
Open Drain	Open Drain	オープンドレイン
SDR	Single Data Rate	シングル・データ・レート
SER	Serializer	シリアライザ
TLDO	True LVDS Output	True LVDS 出力(電流出力)

1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

Web サイト : www.gowinsemi.com/ja

E-mail : support@gowinsemi.com

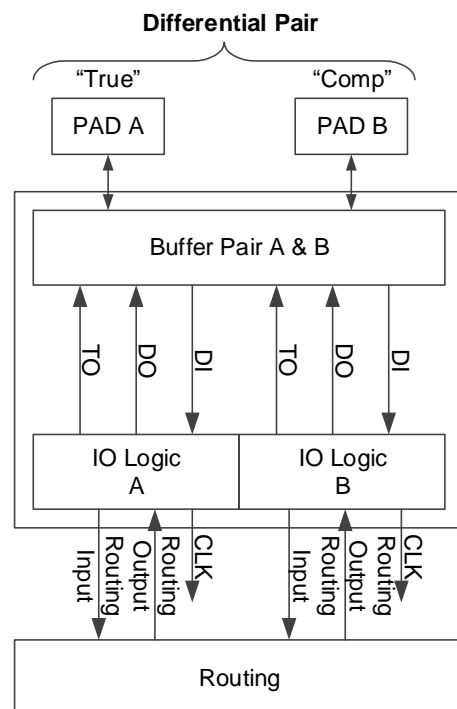
2 GPIO の概要

Arora V FPGA 製品の GPIO には、シングルエンドレベル規格から差動レベル規格まで、さまざまな外部バス、メモリデバイス、ビデオアプリケーションなどとの接続を容易にするための業界のさまざまなレベル規格をサポートする柔軟性があります。

Arora V FPGA 製品の GPIO の基本要素は、入出力バッファ(IO Buffer)、入出力ロジック(IO Logic)、および対応するコンフィギュラブル配線ユニットなどで構成される出力ブロック(IOB)です。そのうちコンフィギュラブル配線ユニットは、コンフィギュラブル機能ユニット(CFU)内のコンフィギュラブル配線ユニット(CRU)と同様です。

図 2-1 示すように、各入出力ブロックには、A および B とマークされる 2 つの入出力ピンがあります。それらは 1 つの差動信号ペアを構成するか、シングルエンド信号として個別に使用することができます。入出力バッファは、主にさまざまなシングルエンドレベル規格および差動レベル規格のサポートに使用されます。入出力ロジックは、シリアル-パラレル変換、パラレル-シリアル変換、遅延制御、およびバイトアライメントなどの機能を統合し、主に高速データ伝送に使用されます。コンフィギュラブル配線ユニットは、入出力ブロックと他のオンチップリソースとの間の相互接続に使用されます。

図 2-1 入出力ブロックの構造



Arora V FPGA 製品の入出力ブロックの特徴：

- Bank 毎に供給される V_{CCIO}
- LVCMOS、PCI、LVTTL、SSTL、HSTL、LVDS、Mini_LVDS、RSDS、PPDS、BLVDS 等複数のレベル規格をサポート
- MIPI 規格および MIPI I3C OpenDrain/PushPull 変換をサポート
- 入力信号のヒステリシス・オプションを提供
- 出力信号のドライブ強度オプションを提供
- 出力信号のスルーレートオプションを提供
- 各ピンに独立したバスホールド、プルアップ/プルダウン抵抗、及びオープンドレイン出力オプションを提供
- ホットプラグをサポート
- 入出力ロジックは、シングルデータレート(SDR)、ダブル・データ・レート(DDR)など、複数のモードをサポート

3 入出力バッファ

3.1 GPIO 規格

Arora V FPGA 製品は、シングルエンド規格と差動規格をサポートしています。シングルエンド規格では、内部のピン電圧をリファレンス電圧として使用するか、任意のピンを外部リファレンス電圧入力ピンとして使用することができます。Arora V FPGA 製品のすべてのバンクは差動入力をサポートしています。エミュレート LVDS 差動出力は外部終端抵抗および差動 LVCMOS バッファ出力により実装されます。さらに、True LVDS 差動出力および差動入力終端をサポートする特定のバンクがあります。詳細については、3.2GPIO のバンキング・スキームを参照してください。

Arora V FPGA 製品でサポートされている GPIO 規格については、対応するデータシートの“I/O 規格”セクションを参照してください。

3.2 GPIO のバンキング・スキーム

GPIO の汎用属性：

- すべての Bank はエミュレート LVDS 差動出力をサポートしますが、外部抵抗ネットワークが必要です。
- すべての Bank は、プルアップ、プルダウン、およびバスホールド設定をサポートしています。
- 各 Bank は 1 つのピン電圧をサポートします。
- 各 Bank は、外部ピンまたは内部リファレンス電圧発生器からの 1 つのリファレンス電圧信号をサポートしています。

3.3 電源供給の要件

コア電圧(V_{CC})とピン電圧(V_{CCIO})が特定のしきい値に達すると、内部パワーオンリセット信号(PoR)がアサートされ、Arora V FPGA 製品のコアロジックがアクティブになります。コンフィギュレーション前およびコンフィギュレーション中は、デバイスのすべての GPIO の状態はデフォルトで弱いプルアップです。コンフィギュレーション完了後、I/O 状態はデフォルトで None となり、EDA で設定することができます。コンフィギュ

レーション関連 I/O の状態は、コンフィギュレーションモードによって異なります。Arora V FPGA 製品には、コア電圧とピン電圧のパワーアップ/パワーダウン・シーケンスの要件はありません。

各 Bank は 1 つのリファレンス電圧(V_{REF})入力をサポートします。Bank 内の任意のピンをリファレンス電圧入力ピンとして構成することができます。SSTL、HSTL などの入力規格をサポートするために、各 Bank には 1 つの独立したリファレンス電圧(V_{REF})が提供されます。ユーザーは、IOB 内部の V_{REF} ソース(GW5AT-138 の場合は 0.6V/0.675V/0.75V/0.9V、GW5A-25 の場合は 0.6V/0.75V/0.9V/1.25V/1.5V)、および V_{CCIO} に基づく比例電圧(33%、42%、50%、58%))を使用するか、外部の V_{REF} 入力(Bank 内の何れか 1 つの I/O ピンを外部 V_{REF} 入力として使用)を使用することができます。

Arora V FPGA 製品の GPIO バッファには、A および B とマークされる 2 つの入出力ピンがあります。ピン A は差動信号の T(True、正側)側に対応し、ピン B は差動信号の C(Comp、負側)側に対応します。

3.3.1 LVCMOS バッファの構成

すべての GPIO には、アプリケーションに応じて複数のモードに構成できる LVCMOS バッファが含まれています。各 LVCMOS バッファは、弱いプルアップ、弱いプルダウン、およびバスホールドに構成できます。弱いプルアップおよび弱いプルダウンは、ワイヤード AND、ワイヤード OR 等の論理制御に幅広く適用できます。バスホールドは最小の電力消費で信号の前の状態をラッチし、バスホールドをオフにすると入力リーク電流が減少します。

すべての LVCMOS バッファは、プログラマブルなドライブ強度を持っています。各規格のドライブ強度オプションについては、対応するデータシートの“I/O 規格”セクションを参照してください。Arora V FPGA 製品のプログラマブルなドライブ強度は各設定の最低限のドライブ強度を保証するだけです。

ヒステリシスの設定は、主にノイズの多い場合のレベルの急激変動を防ぐために使用され、すべての LVCMOS バッファはヒステリシスの設定をサポートします。

差動ペアが 2 本のシングルエンドピンとして構成されている場合、ピン間の相対遅延が最小になり、信号の一貫性が最もよくなります。

3.3.2 差動バッファの構成

GPIO バッファが差動モードに構成された場合、入力ヒステリシスとバスホールド特性は無効になります。

Arora V デバイスのすべての Bank は、オンチップのプログラマブル 100 Ω 入力差動終端抵抗をサポートします。

すべてのシングルエンド GPIO バッファペアは、LVPECL33E、MLVDS25E、BLVDS25E などのエミュレート LVDS 差動出力規格に構成

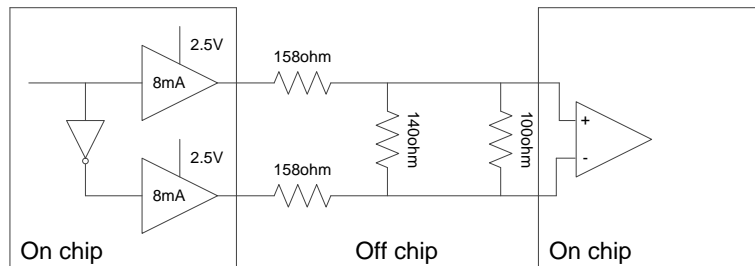
できます。この場合、外部終端を追加する必要があります。

3.4 エミュレート差動回路終端方式

3.4.1 エミュレート LVDS

Arora V FPGA 製品では、コンプリメンタリな LVCMOS 出力と外部終端を利用して、エミュレート LVDS 出力規格を構築することができます。その外部終端方式は下図に示すとおりです。

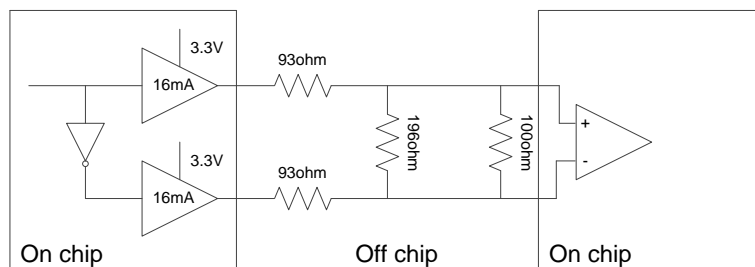
図 3-1 LVDS25E の外部終端



3.4.2 エミュレート LVPECL

Arora V FPGA 製品では、コンプリメンタリな LVCMOS 出力と外部終端を利用して、エミュレート LVPECL 出力規格を構築することができます。その外部終端方式は下図に示すとおりです。

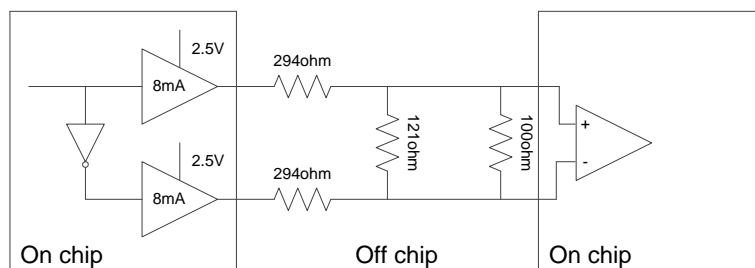
図 3-2 LVPECL の外部終端



3.4.3 エミュレート RSDS

Arora V FPGA 製品では、コンプリメンタリな LVCMOS 出力と外部終端を利用して、エミュレート RSDS 出力規格を構築することができます。その外部終端方式は下図に示すとおりです。

図 3-3 RSDS の外部終端

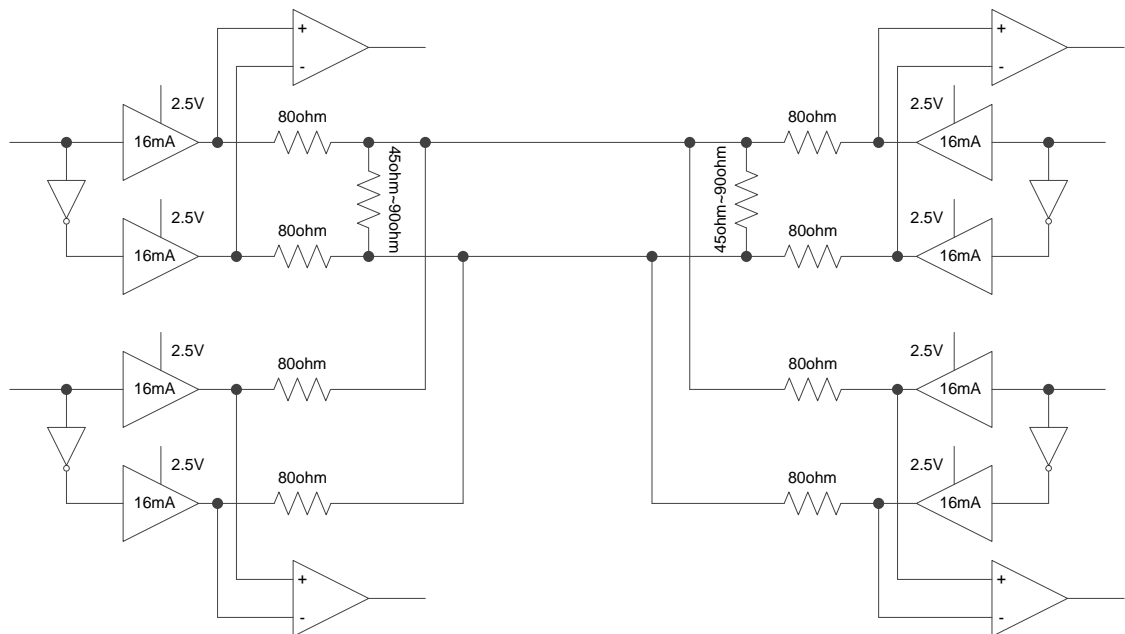


3.4.4 エミュレート BLVDS

Arora V FPGA 製品では、LVCMOS 出力と外部終端を利用して、互換

性のある BLVDS 出力規格を構築することができます。その外部終端方式は下図に示すとおりです。

図 3-4 BLVDS の外部終端



3.5 GPIO の構成

Gowin ソフトウェアの Floorplanner を使用して GPIO の位置と属性を設定するか、CST ファイルをカスタマイズしてそれを実現できます。以下は、CST ファイルによりサポートされている物理制約について説明します。

3.5.1 位置

GPIO の物理位置をロックします。

```
IO_LOC "xxx" H4 exclusive;
```

3.5.2 レベル規格

GPIO の規格を設定します。

```
IO_PORT "xxx" IO_TYPE=LVCMOS18D;
```

3.5.3 ドライブ強度

出力ピンまたは双方向ピンのドライブ強度を設定します。

```
IO_PORT "xxx" DRIVE=12;
```

3.5.4 プルアップ/プルダウン

プルアップ/プルダウンを設定します。UP : プルアップ。DOWN : プルダウン。KEEPER : バスホールド。NONE : ハイインピーダンス。

```
IO_PORT "xxx" PULL_MODE=DOWN;
```

3.5.5 リファレンス電圧

GPIO のリファレンス電圧を設定します。リファレンス電圧は外部ピンまたは内部リファレンス電圧発生器から提供されます。

```
IO_PORT "xxx" VREF=VREF1_LOAD;
```

3.5.6 ヒステリシス

入力ピンまたは双方向ピンのためにヒステリシスを設定します。小さい順 : NONE->H2L->L2H->HIGH。

```
IO_PORT "xxx" HYSTERESIS=L2H;
```

3.5.7 オープンドレイン

出力ピンまたは双方向ピンのためにオープンドレインを開閉します。ON/OFF オプションが提供されます。

```
IO_PORT "xxx" OPEN_DRAIN=ON;
```

3.5.8 シングルエンド終端抵抗

シングルエンド信号のために終端抵抗を設定します。ON/OFF オプションが提供されます。

```
IO_PORT "xxx" SINGLE_RESISTOR=ON;
```

3.5.9 差動終端抵抗

差動信号のために終端抵抗を設定します。ON/OFF オプションが提供されます。

```
IO_PORT "xxx" Diff_RESISTOR=ON;
```

3.5.10 PCI Clamp

PCI クランプダイオードのオン/オフを設定します。クランプダイオードをオンにすることにより、入出力ピンに発生するオーバーシュートを制限することができます。ON/OFF オプションが提供されます。

```
IO_PORT "xxx" PCI_CLAMP=ON;
```

3.5.11 プルアップ/プルダウンの強度

プルアップ/プルダウンの強度を設定することによりピンのプルアップ/プルダウン能力を変更します。WEAK、MEDIUM、および STRONG の 3 つのオプションが提供されます。

```
IO_PORT "xxx" PULL_STRENGTH=MEDIUM;
```

3.6 GPIO プリミティブ

IO Buffer は、機能によって通常のバッファ、エミュレート LVDS(ELVDS)バッファ、および True LVDS(TLVDS)バッファに分類できます。

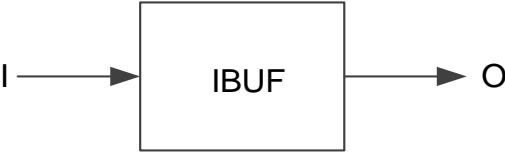
3.6.1 IBUF

プリミティブの紹介

IBUF(Input Buffer)は入力バッファです。

ポート図

図 3-5 IBUF のポート図



ポートの説明

表 3-1 IBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
IBUF uut(  
    .O(O),  
    .I(I)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IBUF  
    PORT (  
        O:OUT std_logic;  
        I:IN std_logic  
    );  
END COMPONENT;  
uut:IBUF  
    PORT MAP(  

```

```
O=>O,  
I=>I  
);
```

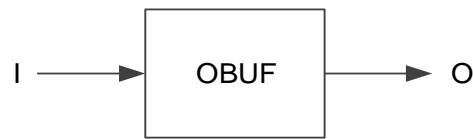
3.6.2 OBUF

プリミティブの紹介

OBUF(Output Buffer)は出力バッファです。

ポート図

図 3-6 OBUF のポート図



ポートの説明

表 3-2 OBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
OBUF uut(  
    .O(O),  
    .I(I)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT OBUF  
    PORT (  
        O:OUT std_logic;  
        I:IN std_logic  
    );  
END COMPONENT;  
uut:OBUF  
    PORT MAP(  
        O=>O,  
        I=>I
```

);

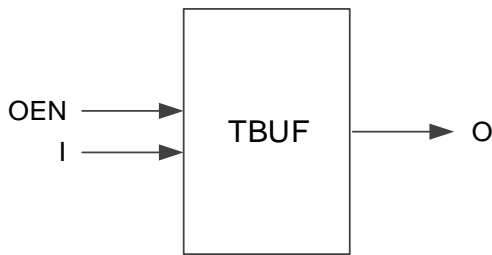
3.6.3 TBUF

プリミティブの紹介

TBUF(Output Buffer with Tristate Control)は、アクティブ Low のトライステートバッファです。

ポート図

図 3-7 TBUF のポート図



ポートの説明

表 3-3 TBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OEN	入力	トライステート出力イネーブル
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
TBUF uut(  
    .O(O),  
    .I(I),  
    .OEN(OEN)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT TBUF  
PORT (  
    O:OUT std_logic;  
    I:IN std_logic;  
    OEN:IN std_logic
```

```
);
END COMPONENT;
uut:TBUF
PORT MAP(
    O=>O,
    I=>I,
    OEN=> OEN
);
```

3.6.4 IOBUF

プリミティブの紹介

IOBUF(Bi-Directional Buffer)は双方向バッファです。OEN が High の場合は入力バッファとして使用され、Low の場合は出力バッファとして使用されます。

ポート図

図 3-8 IOBUF のポート図



ポートの説明

表 3-4 IOBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OEN	入力	トリステート出力イネーブル
IO	双方向	入出力信号
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
IOBUF uut(
    .O(O),
    .IO(IO),
    .I(I),
    .OEN(OEN)
```

```
);
VHDL でのインスタンス化 :
COMPONENT IOBUF
  PORT (
    O:OUT std_logic;
    IO:INOUT std_logic;
    I:IN std_logic;
    OEN:IN std_logic
  );
END COMPONENT;
uut:IOBUF
  PORT MAP(
    O=>O,
    IO=>IO,
    I=>I,
    OEN=> OEN
  );
```

3.6.5 LVDS Input Buffer

プリミティブの紹介

LVDS 差動入力 : TLVDS_IBUF。
TLVDS_IBUF(True LVDS Input Buffer)は、真の差動入力バッファです。

ポート図

図 3-9 TLVDS_IBUF のポート図



ポートの説明

表 3-5 TLVDS_IBUF のポートの説明

ポート	I/O	説明
I	入力	差動入力 A
IB	入力	差動入力 B
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
TLVDS_IBUF uut(
    .O(O),
    .I(I),
    .IB(IB)
);
```

VHDL でのインスタンス化 :

```
COMPONENT TLVDS_IBUF
PORT (
    O:OUT std_logic;
    I:IN std_logic;
    IB:IN std_logic
);
END COMPONENT;
uut:TLVDS_IBUF
PORT MAP(
    O=>O,
    I=>I,
    IB=> IB
);
```

3.6.6 LVDS Output Buffer

プリミティブの紹介

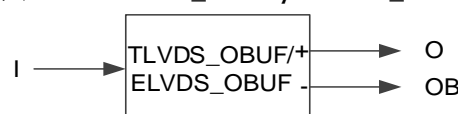
LVDS 差動出力には、TLVDS_OBUF と ELVDS_OBUF の 2 種類があります。

TLVDS_OBUF(True LVDS Output Buffer)は、真の差動出力バッファです。

ELVDS_OBUF(Emulated LVDS Output Buffer)は、エミュレート差動出力バッファです。

ポート図

図 3-10 TLVDS_OBUF/ELVDS_OBUF のポート図



ポートの説明

表 3-6 TLVDS_OBUF/ELVDS_OBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OB	出力	差動出力 B
O	出力	差動出力 A

プリミティブのインスタンス化

例 1

Verilog でのインスタンス化 :

```
TLVDS_OBUF uut(
    .O(O),
    .OB(OB),
    .I(I)
);
```

VHDL でのインスタンス化 :

```
COMPONENT TLVDS_OBUF
PORT (
    O:OUT std_logic;
    OB:OUT std_logic;
    I:IN std_logic
);
END COMPONENT;
uut:TLVDS_OBUF
PORT MAP(
    O=>O,
    OB=>OB,
    I=> I
);
```

例 2

Verilog でのインスタンス化 :

```
ELVDS_OBUF uut(
    .O(O),
    .OB(OB),
    .I(I)
```

```
);
VHDL でのインスタンス化 :
COMPONENT ELVDS_OBUF
  PORT (
    O:OUT std_logic;
    OB:OUT std_logic;
    I:IN std_logic
  );
END COMPONENT;
uut:ELVDS_OBUF
  PORT MAP(
    O=>O,
    OB=>OB,
    I=> I
  );
```

3.6.7 LVDS Tristate Buffer

プリミティブの紹介

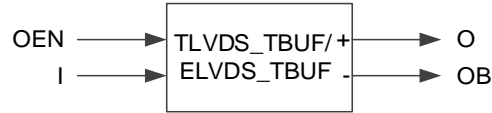
LVDS トライステート差動出力には、TLVDS_TBUF と ELVDS_TBUF の 2 種類があります。

TLVDS_TBUF(True LVDS Tristate Buffer)は、真の差動トライステートバッファで、アクティブ Low です。

ELVDS_TBUF(Emulated LVDS Tristate Buffer)は、エミュレート差動トライステートバッファで、アクティブ Low です。

ポート図

図 3-11 TLVDS_TBUF/ELVDS_TBUF のポート図



ポートの説明

表 3-7 TLVDS_TBUF/ELVDS_TBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OEN	入力	トライステート出力イネーブル
OB	出力	差動出力 B

ポート	I/O	説明
O	出力	差動出力 A

プリミティブのインスタンス化

例 1

Verilog でのインスタンス化 :

```
TLVDS_TBUF uut(
    .O(O),
    .OB(OB),
    .I(I),
    .OEN(OEN)
);
```

VHDL でのインスタンス化 :

```
COMPONENT TLVDS_TBUF
    PORT (
        O:OUT std_logic;
        OB:OUT std_logic;
        I:IN std_logic;
        OEN:IN std_logic
    );
END COMPONENT;
uut:TLVDS_TBUF
    PORT MAP(
        O=>O,
        OB=>OB,
        I=> I,
        OEN=>OEN
    );
```

例 2

Verilog でのインスタンス化 :

```
ELVDS_TBUF uut(
    .O(O),
    .OB(OB),
    .I(I),
```

```

.OEN(OEN)
);
VHDL でのインスタンス化 :
COMPONENT ELVDS_TBUF
  PORT (
    O:OUT std_logic;
    OB:OUT std_logic;
    I:IN std_logic;
    OEN:IN std_logic
  );
END COMPONENT;
uut:ELVDS_TBUF
  PORT MAP(
    O=>O,
    OB=>OB,
    I=>I,
    OEN=>OEN
  );

```

3.6.8 LVDS Inout Buffer

プリミティブの紹介

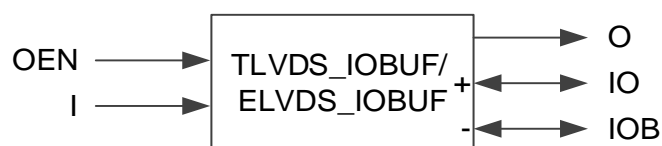
LVDS 差動入出力には、TLVDS_IOBUF と ELVDS_IOBUF の 2 種類あります。

TLVDS_IOBUF(True LVDS Bi-Directional Buffer)は、真の双方向バッファです。OEN が High の場合は、真の差動入力バッファとして使用され、OEN が Low の場合は、真の差動出力バッファとして使用されます。

ELVDS_IOBUF(Emulated LVDS Bi-Directional Buffer)は、エミュレート差動双方向バッファです。OEN が High の場合はエミュレート差動入力バッファとして使用され、OEN が Low の場合はエミュレート差動出力バッファとして使用されます。

ポート図

図 3-12 TLVDS_IOBUF/ELVDS_IOBUF のポート図



ポートの説明

表 3-8 TLVDS_IOBUF/ELVDS_IOBUF のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OEN	入力	トライステート出力イネーブル
O	出力	データ出力信号
IOB	双方向	差動入出力 B
IO	双方向	差動入出力 A

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
ELVDS_IOBUF uut(
    .O(O),
    .IO(IO),
    .IOB(IOB),
    .I(I),
    .OEN(OEN)
);
```

VHDL でのインスタンス化 :

```
COMPONENT ELVDS_IOBUF
    PORT (
        O:OUT std_logic;
        IO:INOUT std_logic;
        IOB:INOUT std_logic;
        I:IN std_logic;
        OEN:IN std_logic
    );
END COMPONENT;
uut:ELVDS_IOBUF
    PORT MAP(
        O=>O,
        IO=>IO,
        IOB=>IOB,
        I=> I,
```

```
OEN=>OEN
);
```

3.6.9 MIPI_IBUF

プリミティブの紹介

MIPI_IBUF(MIPI Input Buffer)は、抵抗の動的構成をサポートする HS 入力モードと、LP 双方向モードとの 2 つの動作モードがあります。

機能の説明

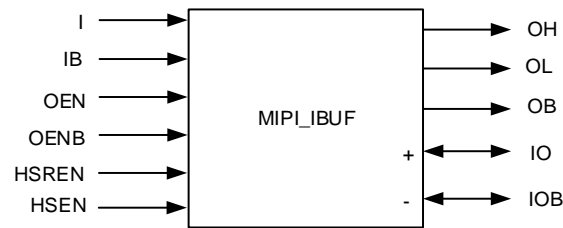
MIPI_IBUF は、LP モードと HS モードをサポートします。IO および IOB は、pad に接続されます。

LP モード：双方向をサポートします。OEN が Low の場合、I は入力で、IO は出力です。OEN が High の場合、IO は入力で、OL は出力です。OENB が Low の場合、IB は入力で、IOB は出力です。OENB が High の場合、IOB は入力で、OB は出力です。

HS モード：IO と IOB は差動入力で、OH は出力です。この場合、HSREN は終端抵抗の制御に使用され、HSEN は HS モードのイネーブルに使用されます。

ポート図

図 3-13 MIPI_IBUF のポート図



ポートの説明

表 3-9 MIPI_IBUF のポートの説明

ポート	I/O	説明
I	入力	LP モードでは、OEN が Low の場合、I は入力です。
IB	入力	LP モードでは、OENB が Low の場合、IB は入力です。
HSREN	入力	HS モードにおける終端抵抗制御信号
HSEN	入力	HS モードのイネーブル信号
OEN	入力	LP モードにおけるトライステート入出力の制御信号
OENB	入力	LP モードにおけるトライステート入出力の制御信号
OH	出力	HS モードにおけるデータ出力信号
OL	出力	LP モードでは、OEN が High の場合、OL は出力です。

ポート	I/O	説明
OB	出力	LP モードでは、OENB が High の場合、OB は出力です。
IO	双方向	<ul style="list-style-type: none"> ● LP モードでは、OEN が Low の場合、IO は出力で、OEN が High の場合、IO は入力です。 ● HS モードでは、IO は入力です。
IOB	双方向	<ul style="list-style-type: none"> ● LP モードでは、OENB が Low の場合、IOB は出力で、OENB が High の場合、IOB は入力です。 ● HS モードでは、IOB は入力です。

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

MIPI_IBUF uut(
    .OH(OH),
    .OL(OL),
    .OB(OB),
    .IO(IO),
    .IOB(IOB),
    .I(I),
    .IB(IB),
    .OEN(OEN),
    .OENB(OENB),
    .HSEN(HSEN),
    .HSREN(HSREN)
);

```

VHDL でのインスタンス化 :

```

COMPONENT MIPI_IBUF
PORT (
    OH:OUT std_logic;
    OL: OUT std_logic;
    OB:OUT std_logic;
    IO:INOUT std_logic;
    IOB:INOUT std_logic;
    I:IN std_logic;
    IB:IN std_logic;
    OEN:IN std_logic;

```



```

        OENB:IN std_logic;
        HSEN:IN std_logic;
        HSREN:IN std_logic

    );
END COMPONENT;
uut: MIPI_IBUF
    PORT MAP(
        OH=>OH,
        OL=>OL,
        OB=>OB,
        IO=>IO,
        IOB=>IOB,
        I=>I,
        IB=>IB,
        OEN=>OEN,
        OENB=>OENB,
        HSEN=>HSEN,
        HSREN=>HSREN

    );
```

3.6.10 MIPI_OBUF_A

プリミティブの紹介

MIPI_OBUF_A には HS と LP の 2 種類の動作モードがあります。

MIPI_OBUF_A(MIPI Output Buffer with IL Signal)は、MIPI 出力バッファです。MODESEL が High の場合は(HS)MIPI 高速出力バッファとして使用され、MODESEL が Low の場合は(LP)MIPI 低消費電力出力バッファとして使用されます。

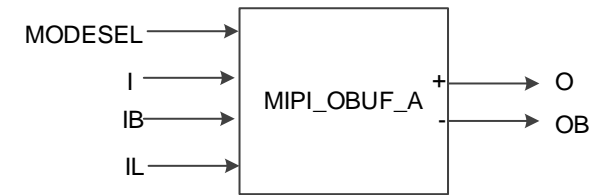
サポートされるデバイス

表 3-10 MIPI_OBUF_A 対応デバイス

ファミリー	シリーズ	デバイス
Arora	GW5A	GW5A-25

ポート図

図 3-14 MIPI_OBUF_A のポート図



ポートの説明

表 3-11 MIPI_OBUF_A のポートの説明

ポート	I/O	説明
I	入力	HS モードにおけるデータ入力 A
IB	入力	LP モードにおけるデータ入力 B
IL	入力	LP モードにおけるデータ入力 A
MODESEL	入力	モード選択(HS または LP)
O	出力	データ出力 A。(HS モードでは差動出力 A、LP モードではシングルエンド出力 A)
OB	出力	データ出力 B。(HS モードでは差動出力 B、LP モードではシングルエンド出力 B)

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
MIPI_OBUF_A uut(  
    .O(O),  
    .OB(OB),  
    .I(I),  
    .IB(IB),  
    .IL(IL),  
    .MODESEL(MODESEL)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT MIPI_OBUF_A  
PORT (  
    O:OUT std_logic;  
    OB:OUT std_logic;  
    I:IN std_logic;  
    IB:IN std_logic;
```

```

        IL: IN std_logic;
        MODESEL:IN std_logic

    );
END COMPONENT;
uut: MIPI_OBUF_A
    PORT MAP(
        O=>O,
        OB=>OB,
        I=>I,
        IB=>IB,
        IL=>IL,
        MDOESEL=>MODESEL
    );
```

3.6.11 I3C_IOBUF

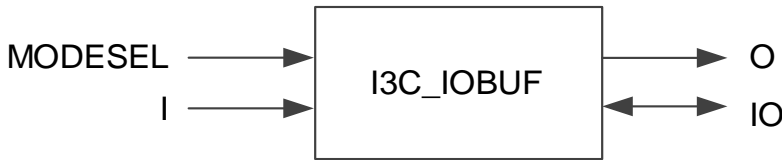
プリミティブの紹介

I3C_IOBUF には、Normal と I3C の 2 つの動作モードがあります。

I3C_IOBUF(I3C Bi-Directional Buffer)は、I3C 双方向バッファです。MODESEL が High の場合は I3C 双方向バッファとして使用され、MODESEL が Low の場合は通常の双方向バッファとして使用されます。

ポート図

図 3-15 I3C_IOBUF のポート図



ポートの説明

表 3-12 I3C_IOBUF のポート図

ポート	I/O	説明
I	入力	データ入力信号
IO	双方向	入出力信号
MODESEL	入力	モード(Normal または I3C)選択信号
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
I3C_IOBUF uut(
    .O(O),
    .IO(IO),
    .I(I),
    .MODESEL(MODESEL)
);
```

VHDL でのインスタンス化 :

```
COMPONENT I3C_IOBUF
    PORT (
        O:OUT std_logic;
        IO:INOUT std_logic;
        I:IN std_logic;
        MODESEL:IN std_logic
    );
END COMPONENT;

uut: I3C_IOBUF
    PORT MAP(
        O=>O,
        IO=>IO,
        I=>I,
        MDOESEL=>MODESEL
    );
```

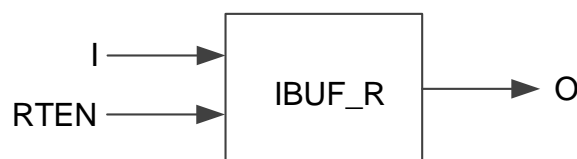
3.6.12 IBUF_R

プリミティブの紹介

IBUF_R(Input Buffer with dynamic ODT)は、動的 ODT 付きの入力バッファです。

ポート図

図 3-16 IBUF_R のポート図



ポートの説明

表 3-13 IBUF_R のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
RTEN	入力	ODT の動的イネーブル
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
IBUF_R uut(
    .O(O),
    .I(I),
    .RTEN(RTEN)
);
```

VHDL でのインスタンス化 :

```
COMPONENT IBUF_R
PORT (
    O:OUT std_logic;
    I:IN std_logic;
    RTEN:IN std_logic
);
END COMPONENT;

uut:IBUF_R
PORT MAP(
    O=>O,
    I=>I,
    RTEN=>RTEN
);
```

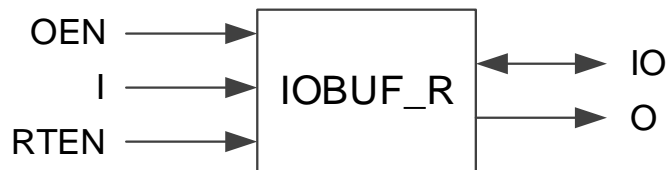
3.6.13 IOBUF_R

プリミティブの紹介

IOBUF_R(Bi-Directional Buffer with dynamic ODT)は、動的 ODT 付きの双方向バッファです。

ポート図

図 3-17 IOBUF_R のポート図



ポートの説明

表 3-14 IOBUF_R のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OEN	入力	トライステート出力イネーブル
RTEN	入力	ODT の動的イネーブル
O	出力	データ出力信号
IO	双方向	入出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

IOBUF_R uut(
    .O(O),
    .IO(IO),
    .I(I),
    .OEN(OEN),
    .RTEN(RTEN)
);

```

VHDL でのインスタンス化 :

```

COMPONENT IOBUF_R
PORT (
    O:OUT std_logic;
    IO:INOUT std_logic;
    I:IN std_logic;
    OEN:IN std_logic;
    RTEN:IN std_logic
);
END COMPONENT;

```

```
uut:IOBUF_R
  PORT MAP(
    O=>O,
    IO=>IO,
    I=>I,
    OEN=> OEN,
    RTEN=>RTEN
  );
```

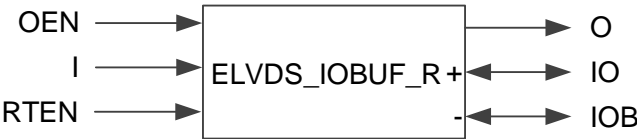
3.6.14 ELVDS_IOBUF_R

プリミティブの紹介

ELVDS_IOBUF_R(Emulated LVDS Bi-Directional Buffer with dynamic ODT)は、動的 ODT 付きのエミュレート差動双方向バッファです。

ポート図

図 3-18 ELVDS_IOBUF_R のポート図



ポートの説明

表 3-15 ELVDS_IOBUF_R のポートの説明

ポート	I/O	説明
I	入力	データ入力信号
OEN	入力	トライステート出力イネーブル
RTEN	入力	ODT の動的イネーブル
IO	双方向	差動入出力 A
IOB	双方向	差動入出力 B
O	出力	データ出力信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
ELVDS_IOBUF_R uut(
  .O(O),
```

```
.IO(IO),
.IOB(IOB),
.I(I),
.OEN(OEN),
.RTEN(RTEN)
);
VHDL でのインスタンス化 :
COMPONENT ELVDS_IOBUF_R
PORT (
    O:OUT std_logic;
    IO:INOUT std_logic;
    IOB:INOUT std_logic;
    I:IN std_logic;
    OEN:IN std_logic;
    RTEN:IN std_logic;
);
END COMPONENT;
uut:ELVDS_IOBUF_R
PORT MAP(
    O=>O,
    IO=>IO,
    IOB=>IOB,
    I=> I,
    OEN=>OEN,
    RTEN=>RTEN
);
```

3.6.15 TLVDS_IBUF_ADC

プリミティブの紹介

TLVDS_IBUF_ADC は、真の差動入力バッファであり、ADC モジュールと併用することで ADC の動的電圧源選択を実装できます。

サポートされるデバイス

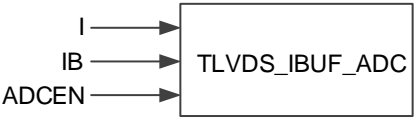
表 3-16 TLVDS_IBUF_ADC 対応デバイス

ファミリー	シリーズ	デバイス
Arora	GW5AT	B バージョンの GW5AT-138

	GW5AST	B バージョンの GW5AST-138
	GW5A	GW5A-25

ポート図

図 3-19 TLVDS_IBUF_ADC のポート図



ポートの説明

表 3-17 TLVDS_IBUF_ADC のポートの説明

ポート	I/O	説明
I	入力	差動入力 A
IB	入力	差動入力 B
ADCEN	入力	ADC の動的イネーブル信号

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
TLVDS_IBUF_ADC uut(  
    .ADCEN(ADCEN),  
    .IB(IB),  
    .I(I)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT TLVDS_IBUF_ADC  
PORT (  
    ADCEN:IN std_logic;  
    IB:IN std_logic;  
    I:IN std_logic  
);  
END COMPONENT;  
uut: TLVDS_IBUF_ADC  
PORT MAP(  
    ADCEN=>ADCEN,  
    IB=>IB,
```

```
    |=>|  
);
```

4 入出力ロジック

Arora V FPGA 製品の入出力ロジック(IOLOGIC)は、SDR、DDR などの動作モードをサポートします。各動作モードでは、信号(または差動信号ペア)は、出力信号、入力信号、双方向信号、及びトリステート出力信号(トリステート制御付きの出力信号)に設定できます。

図 4-1 は Arora V FPGA 製品の入出力ロジックの出力を示します。

図 4-1 入出力ロジックの説明図 -出力

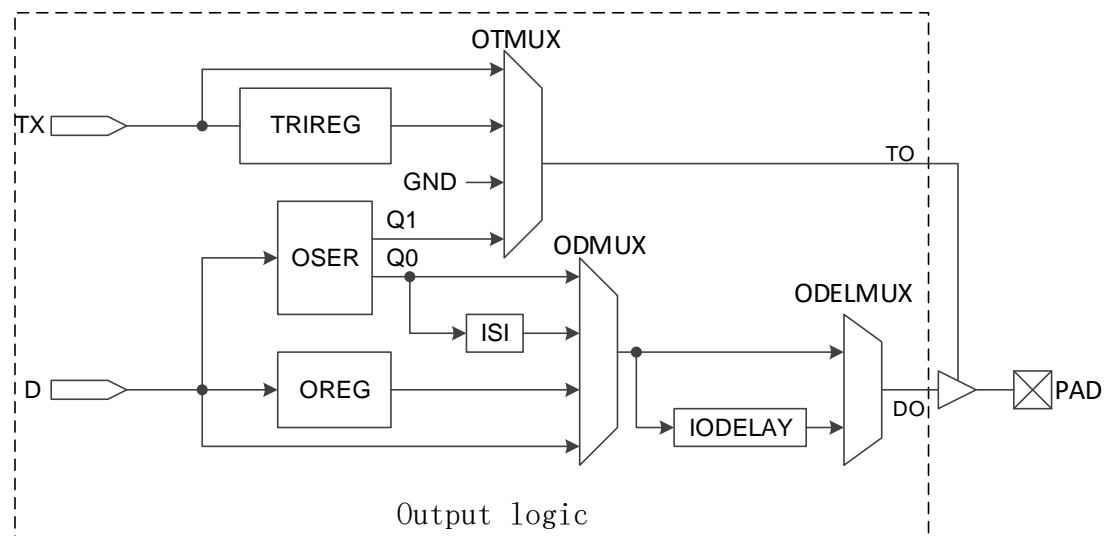
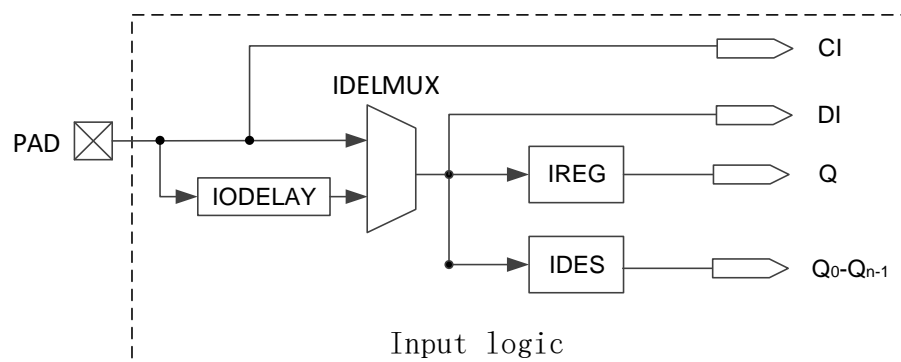


図 4-2 は Arora V FPGA 製品の入出力ロジックの入力を示します。

図 4-2 入出力ロジックの説明図 -入力



注記：
CI は GCLK 入力信号であり、ファブリックに接続できません。DI はファブリックに直接入力されます。

4.1 SDR モード

入出力ロジックは SDR モードをサポートし、入力レジスタ(IREG)、出力レジスタ(OREG)、およびトライステート・レジスタ(TRIREG)を提供します。その機能は CFU の FF/LATCH と同様です。FF/LATCH の入力 D が Buffer/IODELAY によって駆動され、かつ Buffer/IODELAY が他の IOLOGIC を駆動しない場合、または FF/LATCH の出力 Q が Buffer/IODELAY のみを駆動し、かつ Buffer が MIPI バッファでない場合、IOLOGIC として使用できます。

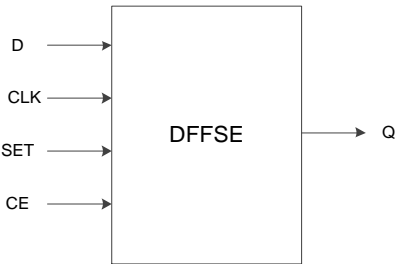
4.1.1 DFFSE

プリミティブの紹介

DFFSE(D Flip-Flop with Clock Enable and Synchronous Set)は立ち上がりエッジでトリガする D フリップフロップで、同期セットとクロックイネーブル機能を備えています。

ポート図

図 4-3 DFFSE のポート図



ポートの説明

表 4-1 DFFSE のポートの説明

ポート名	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
SET	入力	同期セット信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 4-2 DFFSE のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
INIT	1' b1	1' b1	DFFSE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DFFSE instName (
    .D(D),
    .CLK(CLK),
    .SET(SET),
    .CE(CE),
    .Q(Q)
);
defparam instName.INIT=1' b1;
```

VHDL でのインスタンス化 :

```
COMPONENT DFFSE
    GENERIC (INIT:bit:= ' 1' );
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        SET:IN std_logic;
        CE:IN std_logic
    );
END COMPONENT;

uut:DFFSE
    GENERIC MAP(INIT=>' 1' )
    PORT MAP (
        Q=>Q,
        D=>D,
        CLK=>CLK,
        SET=>SET,
        CE=>CE
```

);

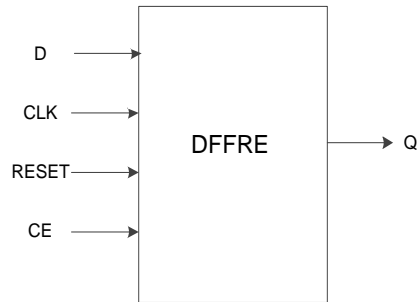
4.1.2 DFFRE

プリミティブの紹介

DFFRE(D Flip-Flop with Clock Enable and Synchronous Reset)は立ち上がりエッジでトリガする D フリップフロップで、同期リセットとクロックイネーブル機能を備えています。

ポート図

図 4-4 DFFRE のポート図



ポートの説明

表 4-3 DFFRE のポートの説明

ポート名	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
RESET	入力	同期リセット信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 4-4 DFFRE のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
INIT	1' b0	1' b0	DFFRE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DFFRE instName (  
    .D(D),  
    .CLK(CLK),
```

```

        .RESET(RESET),
        .CE(CE),
        .Q(Q)
    );
defparam instName.INIT=1' b0;
VHDL でのインスタンス化 :
COMPONENT DFFRE
    GENERIC (INIT:bit:= ' 0' );
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        RESET:IN std_logic;
        CE:IN std_logic
    );
END COMPONENT;
uut:DFFRE
    GENERIC MAP(INIT=>' 0' )
    PORT MAP (
        Q=>Q,
        D=>D,
        CLK=>CLK,
        RESET=>RESET,
        CE=>CE
    );

```

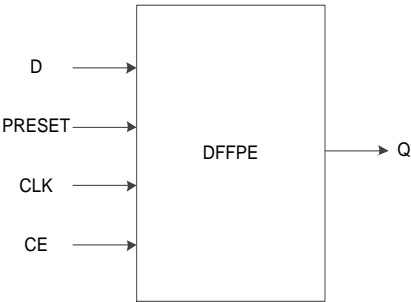
4.1.3 DFFPE

プリミティブの紹介

DFFPE(D Flip-Flop with Clock Enable and Asynchronous Preset)は立ち上がりエッジでトリガする D フリップフロップで、非同期セットとクロックイネーブル機能を備えています。

ポート図

図 4-5 DFFPE のポート図



ポートの説明

表 4-5 DFFPE のポートの説明

ポート名	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
PRESET	入力	非同期セット信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 4-6 DFFPE のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
INIT	1’ b1	1’ b1	DFFPE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DFFPE instName (
    .D(D),
    .CLK(CLK),
    .PRESET(PRESET),
    .CE(CE),
    .Q(Q)
);

defparam instName.INIT=1’ b1;
```

VHDL でのインスタンス化 :


```

COMPONENT DFFPE
  GENERIC (INIT:bit:= ' 1' );
  PORT(
    Q:OUT std_logic;
    D:IN std_logic;
    CLK:IN std_logic;
    PRESET:IN std_logic;
    CE:IN std_logic
  );
END COMPONENT;

uut:DFFPE
  GENERIC MAP(INIT=>' 1' )
  PORT MAP (
    Q=>Q,
    D=>D,
    CLK=>CLK,
    PRESET=>PRESET,
    CE=>CE
  );

```

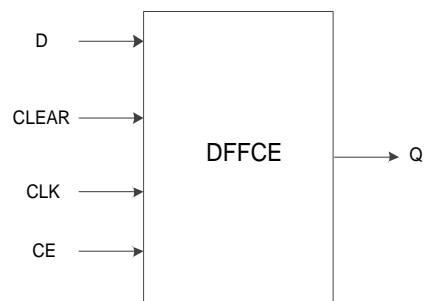
4.1.4 DFFCE

プリミティブの紹介

DFFCE(D Flip-Flop with Clock Enable and Asynchronous Clear)は立ち上がりエッジでトリガする D フリップフロップで、非同期リセットとクロックイネーブル機能を備えています。

ポート図

図 4-6 DFFCE のポート図



ポートの説明

表 4-7 DFFCE のポートの説明

ポート名	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
CLEAR	入力	非同期クリア信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 4-8 DFFCE のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
INIT	1' b0	1' b0	DFFCE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
DFFCE instName (
    .D(D),
    .CLK(CLK),
    .CLEAR(CLEAR),
    .CE(CE),
    .Q(Q)
);
defparam instName.INIT=1' b0;
```

VHDL でのインスタンス化：

```
COMPONENT DFFCE
    GENERIC (INIT:bit:= ' 0' );
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        CLEAR:IN std_logic;
        CE:IN std_logic
    );
```

```
END COMPONENT;  
uut:DFFCE  
    GENERIC MAP(INIT=>' 0' )  
    PORT MAP (  
        Q=>Q,  
        D=>D,  
        CLK=>CLK,  
        CLEAR=>CLEAR,  
        CE=>CE  
    );
```

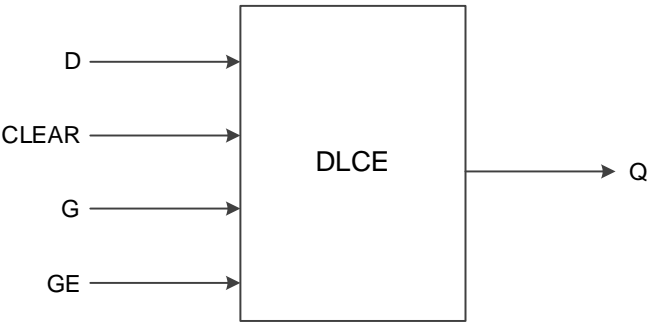
4.1.5 DLCE

プリミティブの紹介

DLCE(Data Latch with Asynchronous Clear and Latch Enable)はイネーブル制御とリセット機能を備えたラッチで、制御信号 **G** はアクティブ High です。

ポート図

図 4-7 DLCE のポート図



ポートの説明

表 4-9 DLCE のポートの説明

ポート名	I/O	説明
D	入力	データ入力信号
CLEAR	入力	非同期クリア信号、アクティブ High
G	入力	データ制御信号、アクティブ High
GE	入力	データ制御イネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 4-10 DLCE のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
INIT	1' b0	1' b0	DLCE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DLCE instName (
    .D(D),
    .CLEAR(CLEAR),
    .G(G),
    .GE(GE),
    .Q(Q)
);
defparam instName.INIT=1' b0;
```

VHDL でのインスタンス化 :

```
COMPONENT DLCE
    GENERIC (INIT:bit:= ' 0' );
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        GE:IN std_logic;
        CLEAR:IN std_logic
    );
END COMPONENT;

uut:DLCE
    GENERIC MAP(INIT=>' 0' )
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G,
        GE=>GE,
        CLEAR=>CLEAR
```

);

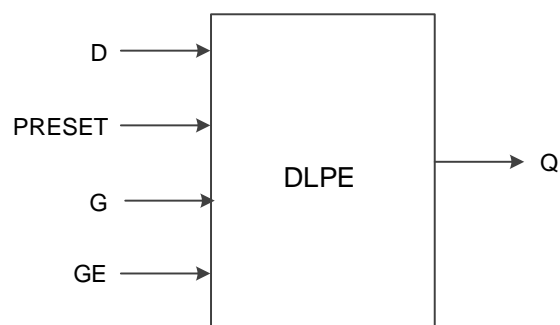
4.1.6 DLPE

プリミティブの紹介

DLPE(Data Latch with Asynchronous Preset and Latch Enable)はイネーブル制御とセット機能を備えたラッチで、制御信号 **G** はアクティブ High です。

ポート図

図 4-8 DLPE のポート図



ポートの説明

表 4-11 DLPE のポートの説明

ポート名	I/O	説明
D	入力	データ入力信号
PRESET	入力	非同期セット信号、アクティブ High
G	入力	データ制御信号、アクティブ High
GE	入力	データ制御イネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 4-12 DLPE のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
INIT	1' b1	1' b1	DLPE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
DLPE instName (
    .D(D),
    .PRESET(PRESET),
    .G(G),
    .GE(GE),
    .Q(Q)
);
defparam instName.INIT=1' b1;
```

VHDL でのインスタンス化：

```
COMPONENT DLPE
    GENERIC (INIT:bit:= ' 1' );
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        GE:IN std_logic;
        PRESET:IN std_logic
    );
```

```

END COMPONENT;
uut:DLPE
    GENERIC MAP(INIT=>' 1' )
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G,
        GE=>GE,
        PRESET =>PRESET
    );

```

4.2 DDR モードの入出力ロジック

4.2.1 IDDR

プリミティブの紹介

IDDR(Dual Data Rate Input)は、ダブル・データ・レートの実現します。

機能の説明

IDDR モードでは、出力データは同じクロックエッジで FPGA ロジックに提供されます。IDDR モードのブロック図は、タイミング図は図 4-10 に示すとおりです。

図 4-9 IDDR のブロック図

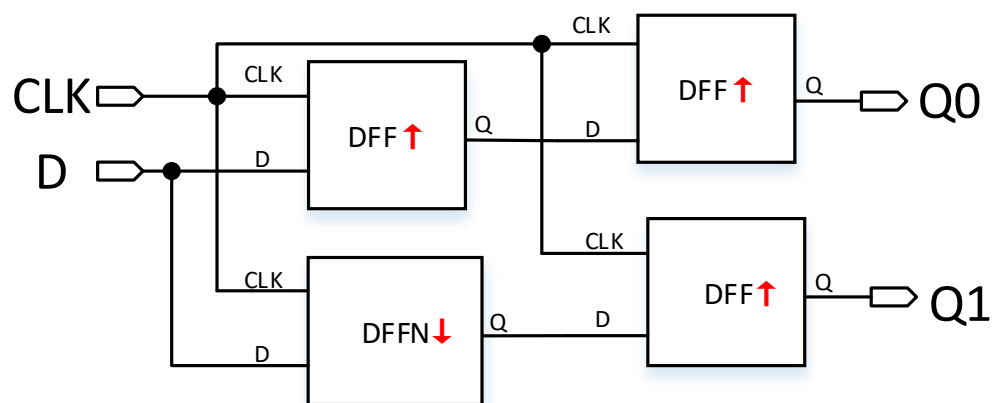
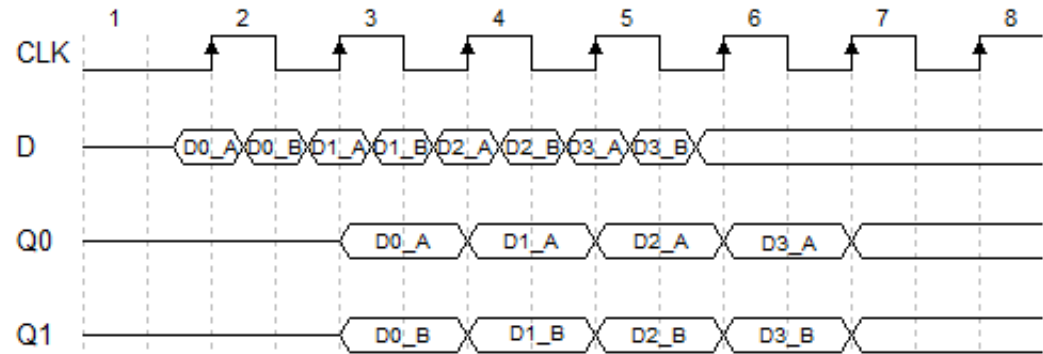


図 4-10 IDDR のタイミング図



ポート図

図 4-11 IDDR のポート図



ポートの説明

表 4-13 IDDR のポートの説明

ポート名	I/O	説明
D	入力	IDDR データ入力信号
CLK	入力	クロック入力信号
Q0, Q1	出力	IDDR データ出力信号

接続ルール

IDDR のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```
IDDR uut(
    .Q0(Q0),
    .Q1(Q1),
    .D(D),
    .CLK(CLK)
);
```

VHDL でのインスタンス化 :


```
COMPONENT IDDR
PORT(
    Q0:OUT std_logic;
    Q1:OUT std_logic;
    D:IN std_logic;
    CLK:IN std_logic
);
END COMPONENT;
uut:IDDR
PORT MAP (
    Q0=>Q0,
    Q1=>Q1,
    D=>D,
    CLK=>CLK
);
```

4.2.2 IDDRC

プリミティブの紹介

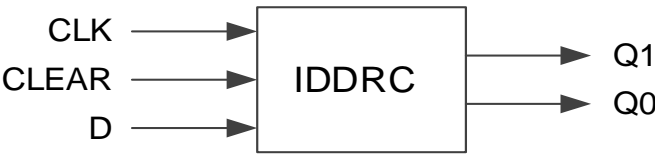
IDDRC(Dual Data Rate Input with Asynchronous Clear)は、IDDR に比べて、非同期リセット機能をさらに備えています。

機能の説明

IDDRC モードでは、出力データは同じクロックエッジで FPGA ロジックに提供されます。

ポート図

図 4-12 IDDRC のポート図



ポートの説明

表 4-14 IDDRC のポートの説明

ポート名	I/O	説明
D	入力	IDDRC データ入力信号
CLK	入力	クロック入力信号
CLEAR	入力	非同期クリア入力、アクティブ High
Q0, Q1	出力	IDDRC データ出力信号

接続ルール

IDDRC のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```
IDDRC uut(  
    .Q0(Q0),  
    .Q1(Q1),  
    .D(D),  
    .CLK(CLK),  
    .CLEAR(CLEAR)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IDDRC  
    PORT(  
        Q0:OUT std_logic;  
        Q1:OUT std_logic;  
        D:IN std_logic;  
        CLEAR:IN std_logic;  
        CLK:IN std_logic  
    );  
END COMPONENT;  
uut:IDDRC  
    PORT MAP (  
        Q0=>Q0,  
        Q1=>Q1,  
        D=>D,  
        CLEAR=>CLEAR,  
        CLK=>CLK  
    );
```

4.2.3 IDES4

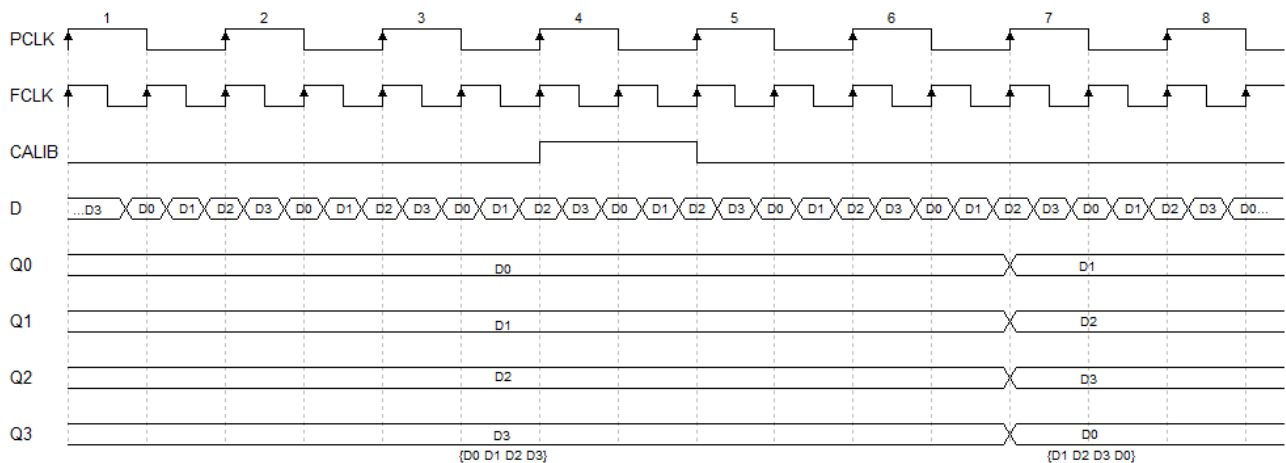
プリミティブの紹介

IDES4(1 to 4 Deserializer)は、1 ビットのシリアル入力、4 ビットの平行出力のデシリアライザです。

機能の説明

IDES4 モードでは、データは 1 : 4 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。**CALIB** によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、4 回シフトされると、出力データはシフト前のデータと同じになります。**CALIB** のタイミングの例を次に示します。

図 4-13 CALIB のタイミングの例



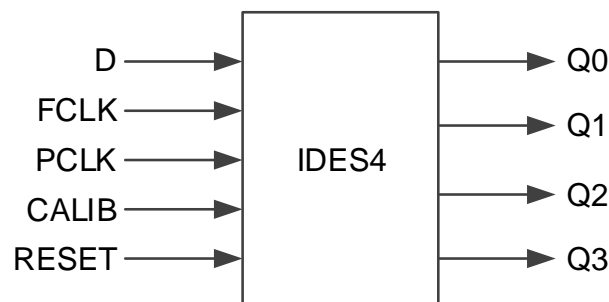
注記：

この例における **CALIB** 信号のパルス幅とタイミングは参照用であり、必要に応じて調整できます。しかし、そのパルス幅は T_{PCLK} 以上である必要があります。

PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/2 f_{FCLK}$ 。

ポート図

図 4-14 IDES4 のポート図



ポートの説明

表 4-15 IDES4 のポートの説明

ポート名	I/O	説明
D	入力	IDES4 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q3~Q0	出力	IDES4 データ出力信号

接続ルール

IDES4 のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
IDES4 uut(
    .Q0(Q0),
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .D(D),
    .FCLK(FCLK),
    .PCLK(PCLK),
    .CALIB(CALIB),
    .RESET(RESET)
);
```

VHDL でのインスタンス化：

```
COMPONENT IDES4
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        Q2:OUT std_logic;
        Q3:OUT std_logic;
```

```

        D:IN std_logic;
        FCLK:IN std_logic;
        PCLK:IN std_logic;
        CALIB:IN std_logic;
        RESET:IN std_logic

    );
END COMPONENT;
uut:IDES4
    PORT MAP (
        Q0=>Q0,
        Q1=>Q1,
        Q2=>Q2,
        Q3=>Q3,
        D=>D,
        FCLK=>FCLK,
        PCLK=>PCLK,
        CALIB=>CALIB,
        RESET=>RESET
    );

```

4.2.4 IDES8

プリミティブの紹介

IDES8(1 to 8 Deserializer)は、1 ビットのシリアル入力、8 ビットのパラレル出力のデシリアライザです。

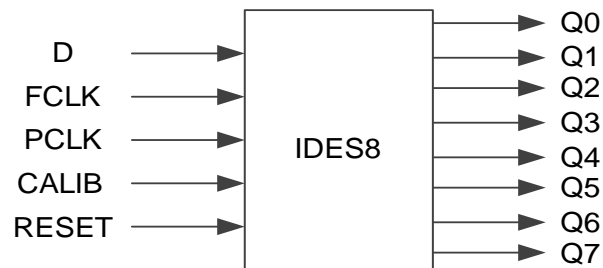
機能の説明

IDES8 モードでは、データは 1 : 8 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力シーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、8 回シフトされると、出力データはシフト前のデータと同じになります。

PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/4 f_{FCLK}$ 。

ポート図

図 4-15 IDES8 のポート図



ポートの説明

表 4-16 IDES8 のポートの説明

ポート名	I/O	説明
D	入力	IDES8 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタクロック入力信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q7~Q0	出力	IDES8 データ出力信号

接続ルール

IDES8 のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```

IDES8 uut(
    .Q0(Q0),
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
    .Q7(Q7),
    .D(D),
    .FCLK(FCLK),

```

```
.PCLK(PCLK),  
.CALIB(CALIB),  
.RESET(RESET)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IDES8  
  PORT(  
    Q0:OUT std_logic;  
    Q1:OUT std_logic;  
    Q2:OUT std_logic;  
    Q3:OUT std_logic;  
    Q4:OUT std_logic;  
    Q5:OUT std_logic;  
    Q6:OUT std_logic;  
    Q7:OUT std_logic;  
    D:IN std_logic;  
    FCLK:IN std_logic;  
    PCLK:IN std_logic;  
    CALIB:IN std_logic;  
    RESET:IN std_logic  
  );  
END COMPONENT;  
uut:IDES8  
  PORT MAP (  
    Q0=>Q0,  
    Q1=>Q1,  
    Q2=>Q2,  
    Q3=>Q3,  
    Q4=>Q4,  
    Q5=>Q5,  
    Q6=>Q6,  
    Q7=>Q7,  
    D=>D,  
    FCLK=>FCLK,  
    PCLK=>PCLK,
```

```
CALIB=>CALIB,  
RESET=>RESET  
);
```


4.2.5 IDES10

プリミティブの紹介

IDES10(1 to 10 Deserializer)は、1 ビットのシリアル入力、10 ビットの平行出力のデシリアライザです。

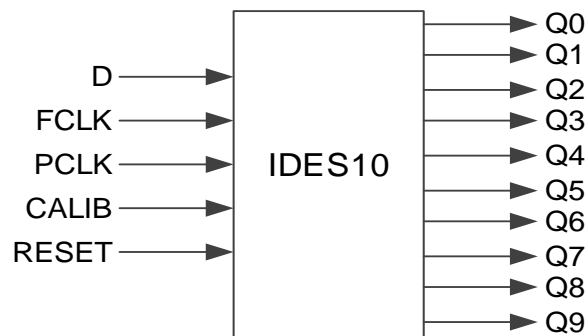
機能の説明

IDES10 モードでは、データは 1 : 10 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、10 回シフトされると、出力データはシフト前のデータと同じになります。

PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/5 f_{FCLK}$ 。

ポート図

図 4-16 IDES10 のポート図



ポートの説明

表 4-17 IDES10 のポートの説明

ポート名	I/O	説明
D	入力	IDES10 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q9~Q0	出力	IDES10 データ出力信号

接続ルール

IDES10 のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```
IDES10 uut(  
    .Q0(Q0),  
    .Q1(Q1),  
    .Q2(Q2),  
    .Q3(Q3),  
    .Q4(Q4),  
    .Q5(Q5),  
    .Q6(Q6),  
    .Q7(Q7),  
    .Q8(Q8),  
    .Q9(Q9),  
    .D(D),  
    .FCLK(FCLK),  
    .PCLK(PCLK),  
    .CALIB(CALIB),  
    .RESET(RESET)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IDES10  
PORT(  
    Q0:OUT std_logic;  
    Q1:OUT std_logic;  
    Q2:OUT std_logic;  
    Q3:OUT std_logic;  
    Q4:OUT std_logic;  
    Q5:OUT std_logic;  
    Q6:OUT std_logic;  
    Q7:OUT std_logic;  
    Q8:OUT std_logic;  
    Q9:OUT std_logic;  
    D:IN std_logic;  
    FCLK:IN std_logic;
```

```

        PCLK:IN std_logic;
        CALIB:IN std_logic;
        RESET:IN std_logic

    );
END COMPONENT;
 uut:IDES10
  PORT MAP (
    Q0=>Q0,
    Q1=>Q1,
    Q2=>Q2,
    Q3=>Q3,
    Q4=>Q4,
    Q5=>Q5,
    Q6=>Q6,
    Q7=>Q7,
    Q8=>Q8,
    Q9=>Q9,
    D=>D,
    FCLK=>FCLK,
    PCLK=>PCLK,
    CALIB=>CALIB,
    RESET=>RESET
  );

```

4.2.6 IVIDEO

プリミティブの紹介

IVIDEO(1 to 7 Deserializer)は、1 ビットのシリアル入力、7 ビットのパラレル出力のデシリアライザです。

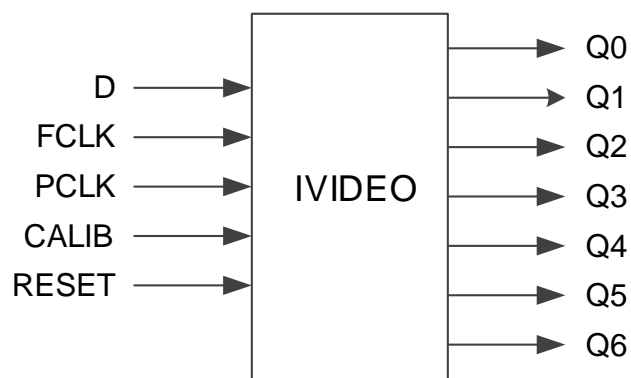
機能の説明

IVIDEO モードでは、データは 1 : 7 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 2 ビットシフトされ、7 回シフトされると、出力データはシフト前のデータと同じになります。

PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/3.5 f_{FCLK}$ 。

ポート図

図 4-17 IVIDEO のポート図



ポートの説明

表 4-18 IVIDEO のポートの説明

ポート名	I/O	説明
D	入力	IVIDEO データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q6~Q0	出力	IVIDEO データ出力

接続ルール

IVIDEO のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
IVIDEO uut(
    .Q0(Q0),
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
```

```
.D(D),  
.FCLK(FCLK),  
.PCLK(PCLK),  
.CALIB(CALIB),  
.RESET(RESET)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IVIDEO  
  PORT(  
    Q0:OUT std_logic;  
    Q1:OUT std_logic;  
    Q2:OUT std_logic;  
    Q3:OUT std_logic;  
    Q4:OUT std_logic;  
    Q5:OUT std_logic;  
    Q6:OUT std_logic;  
    D:IN std_logic;  
    FCLK:IN std_logic;  
    PCLK:IN std_logic;  
    CALIB:IN std_logic;  
    RESET:IN std_logic  
  );  
END COMPONENT;  
 uut:IVIDEO  
  PORT MAP (  
    Q0=>Q0,  
    Q1=>Q1,  
    Q2=>Q2,  
    Q3=>Q3,  
    Q4=>Q4,  
    Q5=>Q5,  
    Q6=>Q6,  
    D=>D,  
    FCLK=>FCLK,  
    PCLK=>PCLK,
```

```

    CALIB=>CALIB,
    RESET=>RESET
);

```

4.2.7 IDES16

プリミティブの紹介

IDES16(1 to 16 Deserializer)は、1 ビットのシリアル入力、16 ビットの平行出力のデシリアライザです。

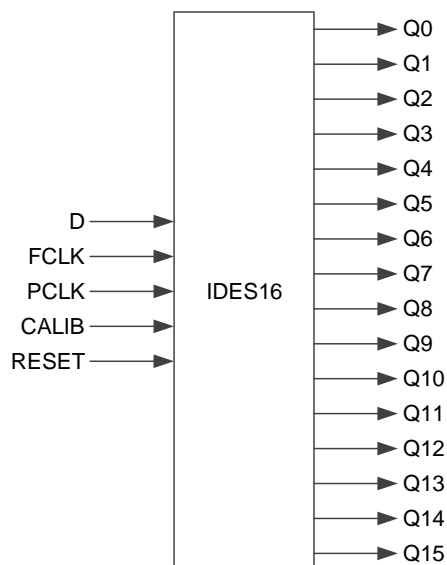
機能の説明

IDES16 モードでは、データは 1 : 16 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、16 回シフトされると、出力データはシフト前のデータと同じになります。

PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/8 f_{FCLK}$ 。

ポート図

図 4-18 IDES16 のポート図



ポートの説明

表 4-19 IDES16 のポートの説明

ポート名	I/O	説明
D	入力	IDES16 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号

ポート名	I/O	説明
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q15~Q0	出力	IDES16 データ出力信号

接続ルール

IDES16 のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
IDES16 uut(
    .Q0(Q0),
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
    .Q7(Q7),
    .Q8(Q8),
    .Q9(Q9),
    .Q10(Q10),
    .Q11(Q11),
    .Q12(Q12),
    .Q13(Q13),
    .Q14(Q14),
    .Q15(Q15),
    .D(D),
    .FCLK(FCLK),
    .PCLK(PCLK),
    .CALIB(CALIB),
    .RESET(RESET)
);
```

VHDL でのインスタンス化 :**COMPONENT IDES16**PORT(
Q0:OUT std_logic;
Q1:OUT std_logic;
Q2:OUT std_logic;
Q3:OUT std_logic;
Q4:OUT std_logic;
Q5:OUT std_logic;
Q6:OUT std_logic;
Q7:OUT std_logic;
Q8:OUT std_logic;
Q9:OUT std_logic;
Q10:OUT std_logic;
Q11:OUT std_logic;
Q12:OUT std_logic;
Q13:OUT std_logic;
Q14:OUT std_logic;
Q15:OUT std_logic;
D:IN std_logic;
FCLK:IN std_logic;
PCLK:IN std_logic;
CALIB:IN std_logic;
RESET:IN std_logic
);**END COMPONENT;**

uut:IDES16

PORT MAP (
Q0=>Q0,
Q1=>Q1,
Q2=>Q2,
Q3=>Q3,
Q4=>Q4,
Q5=>Q5,
Q6=>Q6,


```

Q7=>Q7,
Q8=>Q8,
Q9=>Q9,
Q10=>Q10,
Q11=>Q11,
Q12=>Q12,
Q13=>Q13,
Q14=>Q14,
Q15=>Q15,
D=>D,
FCLK=>FCLK,
PCLK=>PCLK,
CALIB=>CALIB,
RESET=>RESET
);

```

4.2.8 IDDR_MEM

プリミティブの紹介

IDDR_MEM(Dual Data Rate Input with Memory)は、メモリ付きのダブル・データ・レートの実現します。

機能の説明

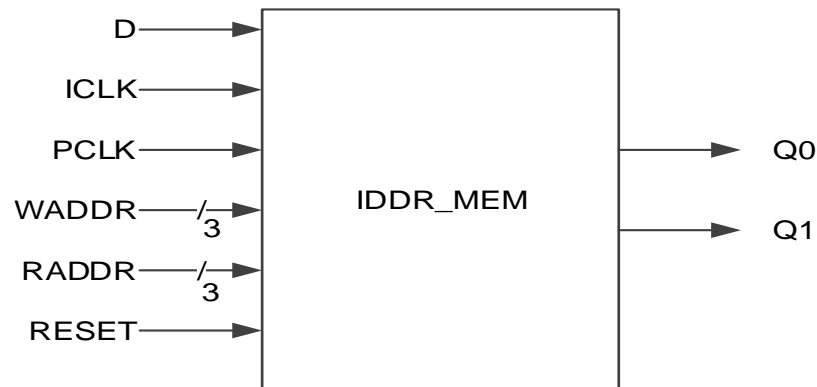
IDDR_MEM 出力データは同じクロックエッジで FPGA ロジックに提供されます。IDDR_MEM には DQS が必要です。ICLK は DQS の出力信号 DQSR90 に接続され、データは ICLK のクロックエッジに従って IDDR_MEM に入力されます。WADDR[2:0]は DQS の出力信号 WPOINT に接続されます。RADDR[2:0]は DQS の出力信号 RPOINT に接続されます。

PCLK と ICLK の周波数関係は： $f_{PCLK} = f_{ICLK}$ 。

PCLK と ICLK の間には一定の位相関係があり、その位相関係は DQS の DLLSTEP 値により決定できます。

ポート図

図 4-19 IDDR_MEM のポート図



ポートの説明

表 4-20 IDDR_MEM のポートの説明

ポート名	I/O	説明
D	入力	IDDR_MEM データ入力信号
ICLK	入力	DQS モジュールの DQSR90 からのクロック入力信号
PCLK	入力	マスタークロック入力信号
WADDR[2:0]	入力	DQS モジュールの WPOINT からの書き込みアドレス信号
RADDR[2:0]	入力	DQS モジュールの RPOINT からの読み出しアドレス信号
RESET	入力	非同期リセット入力、アクティブ High
Q1~Q0	出力	IDDR_MEM データ出力信号

接続ルール

- IDDR_MEM のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。
- ICLK は、DQS モジュールの DQSR90 に接続する必要があります。
- WADDR[2:0]は、DQS モジュールの WPOINT に接続する必要があります。
- RADDR[2:0]は、DQS モジュールの RPOINT に接続する必要があります。

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
IDDR_MEM iddr_mem_inst(
```

```
.Q0(q0),  
.Q1(q1),  
.D(d),  
.ICLK (iclk),  
.PCLK(pclk),  
.WADDR(waddr[2:0]),  
.RADDR(raddr[2:0]),  
.RESET(reset)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IDDR_MEM  
  PORT(  
    Q0:OUT std_logic;  
    Q1:OUT std_logic;  
    D:IN std_logic;  
    ICLK:IN std_logic;  
    PCLK:IN std_logic;  
    WADDR:IN std_logic_vector(2 downto 0);  
    RADDR:IN std_logic_vector(2 downto 0);  
    RESET:IN std_logic  
  );  
END COMPONENT;  
uut:IDDR_MEM  
  PORT MAP (  
    Q0=>q0,  
    Q1=>q1,  
    D=>d,  
    ICLK=>iclk,  
    PCLK=>pclk,  
    WADDR=>waddr,  
    RADDR=>raddr,  
    RESET=>reset  
  );
```

4.2.9 IDES4_MEM

プリミティブの紹介

IDES4_MEM(1 to 4 Deserializer with Memory) は、メモリ付きの 1:4 デシリアライザです。

機能の説明

IDES4_MEM モードでは、データは 1:4 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、4 回シフトされると、出力データはシフト前のデータと同じになります。

IDES4_MEM には DQS が必要です。ICLK は DQS の出力信号 DQSR90 に接続され、データは ICLK のクロックエッジに従って IDES4_MEM に入力されます。WADDR[2:0] は DQS の出力信号 WPOINT に接続されます。RADDR[2:0] は DQS の出力信号 RPOINT に接続されます。

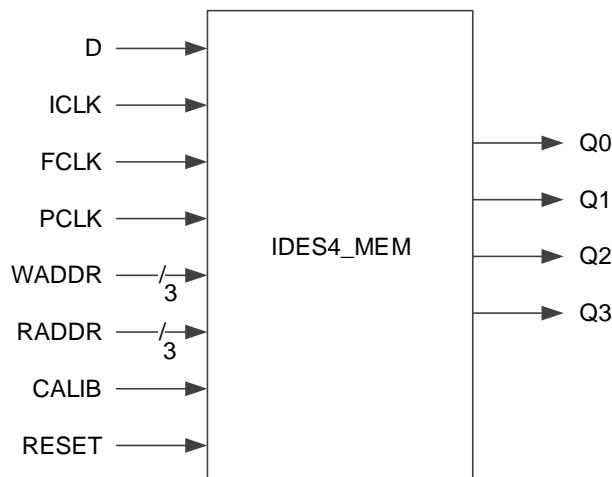
PCLK、FCLK、および ICLK の周波数関係は：

$$f_{PCLK} = 1/2 f_{FCLK} = 1/2 f_{ICLK}。$$

FCLK と ICLK の間には一定の位相関係があり、その位相関係は DQS の DLLSTEP 値により決定できます。

ポート図

図 4-20 IDES4_MEM のポート図



ポートの説明

表 4-21 IDES4_MEM のポートの説明

ポート名	I/O	説明
D	入力	IDES4_MEM データ入力信号

ポート名	I/O	説明
ICLK	入力	DQS モジュールの DQSR90 からのクロック入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
WADDR[2:0]	入力	DQS モジュールの WPOINT からの書き込みアドレス信号
RADDR[2:0]	入力	DQS モジュールの RPOINT からの読み出しアドレス信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q3~Q0	出力	IDES4_MEM データ出力信号

接続ルール

- IDES4_MEM のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。
- ICLK は、DQS モジュールの DQSR90 に接続する必要があります。
- WADDR[2:0]は、DQS モジュールの WPOINT に接続する必要があります。
- RADDR[2:0]は、DQS モジュールの RPOINT に接続する必要があります。

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
IDES4_MEM ides4_mem_inst(
    .Q0(q0),
    .Q1(q1),
    .Q2(q2),
    .Q3(q3),
    .D(d),
    .ICLK(iclk),
    .FCLK(fclk),
    .PCLK(pclk),
    .WADDR(waddr[2:0]),
    .RADDR(raddr[2:0]),
    .CALIB(calib),
```

```
        .RESET(reset)
    );
VHDL でのインスタンス化 :
COMPONENT IDES4_MEM
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        Q2:OUT std_logic;
        Q3:OUT std_logic;
        D:IN std_logic;
        ICLK:IN std_logic;
        FCLK:IN std_logic;
        PCLK:IN std_logic;
        WADDR:IN std_logic_vector(2 downto 0);
        RADDR:IN std_logic_vector(2 downto 0);
        CALIB:IN std_logic;
        RESET:IN std_logic
    );
END COMPONENT;
uut:IDES4_MEM
    PORT MAP (
        Q0=>q0,
        Q1=>q1,
        Q2=>q2,
        Q3=>q3,
        D=>d,
        ICLK=>iclk,
        FCLK=>fclk,
        PCLK=>pclk,
        WADDR=>waddr,
        RADDR=>raddr,
        CALIB=>calib,
        RESET=>reset
    );
```

4.2.10 IDES8_MEM

プリミティブの紹介

IDES8_MEM(1 to 8 Deserializer with Memory) は、メモリ付きの 1:8 デシリアライザです。

機能の説明

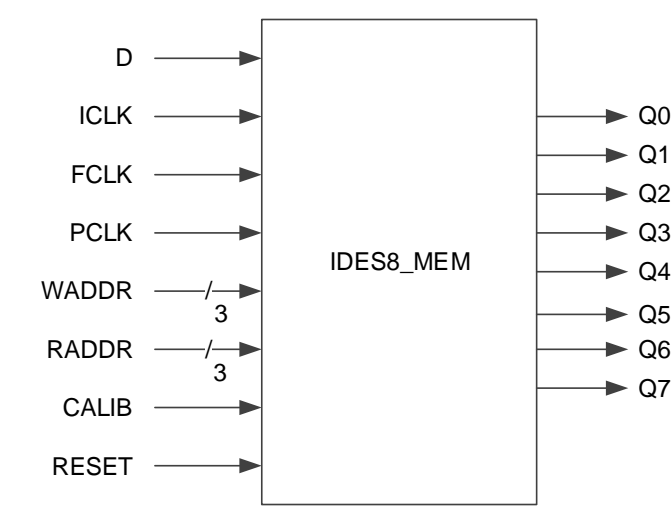
IDES8_MEM モードでは、データは 1:8 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力シーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、8 回シフトされると、出力データはシフト前のデータと同じになります。IDES8_MEM には DQS が必要です。ICLK は DQS の出力信号 DQSR90 に接続され、データは ICLK のクロックエッジに従って IDES8_MEM に入力されます。WADDR[2:0]は DQS の出力信号 WPOINT に接続されます。RADDR[2:0]は DQS の出力信号 RPOINT に接続されます。

PCLK、FCLK、および ICLK の周波数関係は：
 $f_{PCLK} = 1/4 f_{FCLK} = 1/4 f_{ICLK}$ 。

FCLK と ICLK の間には一定の位相関係があり、その位相関係は DQS の DLLSTEP 値により決定できます。

ポート図

図 4-21 IDES8_MEM のポート図



ポートの説明

表 4-22 IDES8_MEM のポートの説明

ポート名	I/O	説明
D	入力	IDES8_MEM データ入力信号
ICLK	入力	DQS モジュールの DQSR90 からのクロック入力信号

ポート名	I/O	説明
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
WADDR[2:0]	入力	DQS モジュールの WPOINT からの書き込みアドレス信号
RADDR[2:0]	入力	DQS モジュールの RPOINT からの読み出しアドレス信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q7~Q0	出力	IDES8_MEM データ出力信号

接続ルール

- IDES8_MEM のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。
- ICLK は、DQS モジュールの DQSR90 に接続する必要があります。
- WADDR[2:0]は、DQS モジュールの WPOINT に接続する必要があります。
- RADDR[2:0]は、DQS モジュールの RPOINT に接続する必要があります。

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
IDES8_MEM ides8_mem_inst(
    .Q0(q0),
    .Q1(q1),
    .Q2(q2),
    .Q3(q3),
    .Q4(q4),
    .Q5(q5),
    .Q6(q6),
    .Q7(q7),
    .D(d),
    .ICLK(iclk),
    .FCLK(fclk),
    .PCLK(pclk),
    .WADDR(waddr[2:0]),
```



```
.RADDR(raddr[2:0]),  
.CALIB(calib),  
.RESET(reset)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT IDES8_MEM  
  PORT(  
    Q0:OUT std_logic;  
    Q1:OUT std_logic;  
    Q2:OUT std_logic;  
    Q3:OUT std_logic;  
    Q4:OUT std_logic;  
    Q5:OUT std_logic;  
    Q6:OUT std_logic;  
    Q7:OUT std_logic;  
    D:IN std_logic;  
    ICLK:IN std_logic;  
    FCLK:IN std_logic;  
    PCLK:IN std_logic;  
    WADDR:IN std_logic_vector(2 downto 0);  
    RADDR:IN std_logic_vector(2 downto 0);  
    CALIB:IN std_logic;  
    RESET:IN std_logic  
  );  
END COMPONENT;  
uut:IDES8_MEM  
  PORT MAP (  
    Q0=>q0,  
    Q1=>q1,  
    Q2=>q2,  
    Q3=>q3,  
    Q4=>q4,  
    Q5=>q5,  
    Q6=>q6,  
    Q7=>q7,
```

```

D=>d,
ICLK=>iclk,
FCLK=>fclk,
PCLK=>pclk,
WADDR=>waddr,
RADDR=>raddr,
CALIB=>calib,
RESET=>reset
);

```

4.2.11 IDES14

プリミティブの紹介

IDES14(1 to 14 Deserializer)は、1 ビットのシリアル入力、14 ビットの平行出力のデシリアライザです。

機能の説明

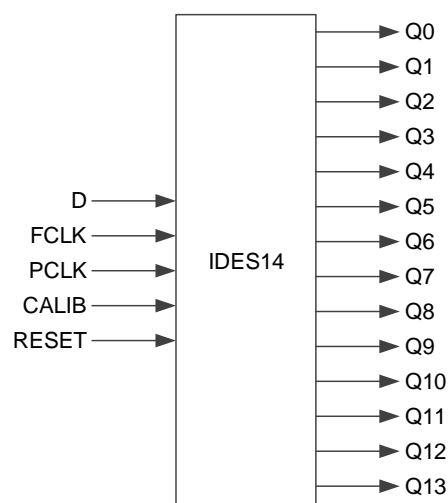
IDES14 モードでは、データは 1 : 14 デシリアライズされ、出力データは同じクロックエッジで FPGA ロジックに提供されます。CALIB によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、14 回シフトされると、出力データはシフト前のデータと同じになります。

PCLK は通常、FCLK の分周によって得られます：

$$f_{PCLK} = 1/7 f_{FCLK}。$$

ポート図

図 4-22 IDES14 のポート図



ポートの説明

表 4-23 IDES14 のポートの説明

ポート名	I/O	説明
D	入力	IDES14 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q13~Q0	出力	IDES14 データ出力信号

接続ルール

IDES14 のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
IDES14 uut(
    .Q0(Q0),
    .Q1(Q1),
    .Q2(Q2),
    .Q3(Q3),
    .Q4(Q4),
    .Q5(Q5),
    .Q6(Q6),
    .Q7(Q7),
    .Q8(Q8),
    .Q9(Q9),
    .Q10(Q10),
    .Q11(Q11),
    .Q12(Q12),
    .Q13(Q13),
    .D(D),
    .FCLK(FCLK),
    .PCLK(PCLK),
    .CALIB(CALIB),
```

```
.RESET(RESET)
);
VHDL でのインスタンス化 :
COMPONENT IDES14
  PORT(
    Q0:OUT std_logic;
    Q1:OUT std_logic;
    Q2:OUT std_logic;
    Q3:OUT std_logic;
    Q4:OUT std_logic;
    Q5:OUT std_logic;
    Q6:OUT std_logic;
    Q7:OUT std_logic;
    Q8:OUT std_logic;
    Q9:OUT std_logic;
    Q10:OUT std_logic;
    Q11:OUT std_logic;
    Q12:OUT std_logic;
    Q13:OUT std_logic;
    D:IN std_logic;
    FCLK:IN std_logic;
    PCLK:IN std_logic;
    CALIB:IN std_logic;
    RESET:IN std_logic
  );
END COMPONENT;
uut:IDES14
  PORT MAP (
    Q0=>Q0,
    Q1=>Q1,
    Q2=>Q2,
    Q3=>Q3,
    Q4=>Q4,
    Q5=>Q5,
    Q6=>Q6,
```

```

Q7=>Q7,
Q8=>Q8,
Q9=>Q9,
Q10=>Q10,
Q11=>Q11,
Q12=>Q12,
Q13=>Q13,
D=>D,
FCLK=>FCLK,
PCLK=>PCLK,
CALIB=>CALIB,
RESET=>RESET
);

```

4.2.12 IDES32

プリミティブの紹介

IDES32(1 to 32 Deserializer)は、1 ビットのシリアル入力、32 ビットの平行出力のデシリアライザです。

機能の説明

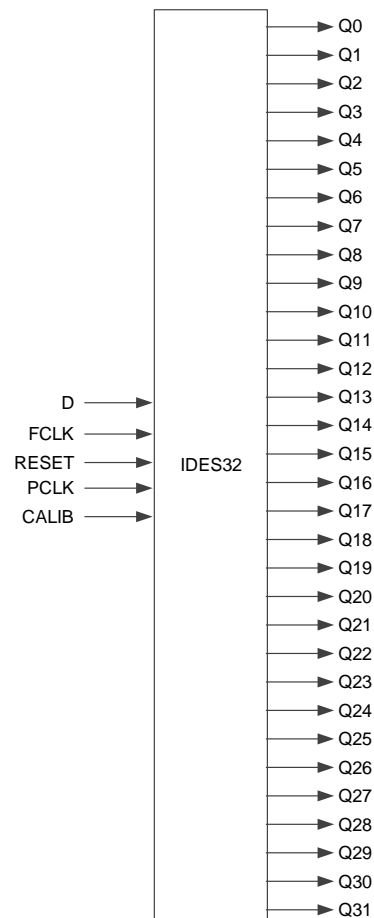
IDES32 モードでは、データは 1 : 32 デシリアライズされ、出力データは同じクロックエッジで **FPGA** ロジックに提供されます。**CALIB** によって出力データのシーケンスを調整することがサポートされます。データはパルスごとに 1 ビットシフトされ、32 回シフトされると、出力データはシフト前のデータと同じになります。

PCLK は通常、FCLK の分周によって得られます：

$$f_{PCLK} = 1/16 f_{FCLK}。$$

ポート図

図 4-23 IDES32 のポート図



ポートの説明

表 4-24 IDES32 のポートの説明

ポート名	I/O	説明
D	入力	IDES32 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
CALIB	入力	出力データのシーケンスの調整用、アクティブ High
RESET	入力	非同期リセット入力、アクティブ High
Q31~Q0	出力	IDES32 データ出力信号

接続ルール

IDES32 のデータ入力 D は、IBUF から直接取得するか、IODELAY モジュールを介してその出力 DO から取得することができます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
IDES32 uut(  
    .Q0(Q0),  
    .Q1(Q1),  
    .Q2(Q2),  
    .Q3(Q3),  
    .Q4(Q4),  
    .Q5(Q5),  
    .Q6(Q6),  
    .Q7(Q7),  
    .Q8(Q8),  
    .Q9(Q9),  
    .Q10(Q10),  
    .Q11(Q11),  
    .Q12(Q12),  
    .Q13(Q13),  
    .Q14(Q14),  
    .Q15(Q15),  
    .Q16(Q16),  
    .Q17(Q17),  
    .Q18(Q18),  
    .Q19(Q19),  
    .Q20(Q20),  
    .Q21(Q21),  
    .Q22(Q22),  
    .Q23(Q23),  
    .Q24(Q24),  
    .Q25(Q25),  
    .Q26(Q26),  
    .Q27(Q27),  
    .Q28(Q28),  
    .Q29(Q29),  
    .Q30(Q30),
```

```
.Q31(Q31),  
.D(D),  
.FCLK(FCLK),  
.PCLK(PCLK),  
.CALIB(CALIB),  
.RESET(RESET)  
);
```

VHDL でのインスタンス化 :

COMPONENT IDES32

```
PORT(  
    Q0:OUT std_logic;  
    Q1:OUT std_logic;  
    Q2:OUT std_logic;  
    Q3:OUT std_logic;  
    Q4:OUT std_logic;  
    Q5:OUT std_logic;  
    Q6:OUT std_logic;  
    Q7:OUT std_logic;  
    Q8:OUT std_logic;  
    Q9:OUT std_logic;  
    Q10:OUT std_logic;  
    Q11:OUT std_logic;  
    Q12:OUT std_logic;  
    Q13:OUT std_logic;  
    Q14:OUT std_logic;  
    Q15:OUT std_logic;  
    Q16:OUT std_logic;  
    Q17:OUT std_logic;  
    Q18:OUT std_logic;  
    Q19:OUT std_logic;  
    Q20:OUT std_logic;  
    Q21:OUT std_logic;  
    Q22:OUT std_logic;  
    Q23:OUT std_logic;  
    Q24:OUT std_logic;
```



```
    Q25:OUT std_logic;
    Q26:OUT std_logic;
    Q27:OUT std_logic;
    Q28:OUT std_logic;
    Q29:OUT std_logic;
    Q30:OUT std_logic;
    Q31:OUT std_logic;
    D:IN std_logic;
    FCLK:IN std_logic;
    PCLK:IN std_logic;
    CALIB:IN std_logic;
    RESET:IN std_logic
);
END COMPONENT;
uut:IDES32
  PORT MAP (
    Q0=>Q0,
    Q1=>Q1,
    Q2=>Q2,
    Q3=>Q3,
    Q4=>Q4,
    Q5=>Q5,
    Q6=>Q6,
    Q7=>Q7,
    Q8=>Q8,
    Q9=>Q9,
    Q10=>Q10,
    Q11=>Q11,
    Q12=>Q12,
    Q13=>Q13,
    Q14=>Q14,
    Q15=>Q15,
    Q16=>Q16,
    Q17=>Q17,
    Q18=>Q18,
```

```

Q19=>Q19,
Q20=>Q20,
Q21=>Q21,
Q22=>Q22,
Q23=>Q23,
Q24=>Q24,
Q25=>Q25,
Q26=>Q26,
Q27=>Q27,
Q28=>Q28,
Q29=>Q29,
Q30=>Q30,
Q31=>Q31,
D=>D,
FCLK=>FCLK,
PCLK=>PCLK,
CALIB=>CALIB,
RESET=>RESET
);

```

4.2.13 OSIDES32

プリミティブの紹介

OSIDES32 は、1 ビットのシリアル入力、32 ビットの平行出力のオーバーサンプリングシリのデシリアライザです。

機能の説明

1:32 のオーバーサンプリングのデシリアライズは 2 つの IOL で実現できます。また、非同期リセット機能がサポートされています。ただし、CALIB による出力データのシーケンスの調整はサポートされていません。

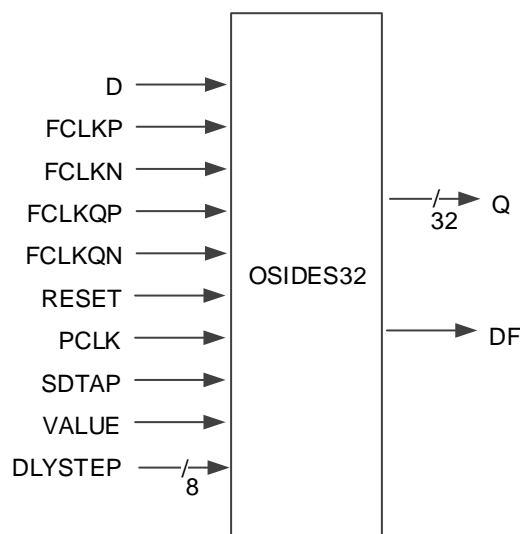
オーバーサンプリングのデシリアライズと通常のデシリアライズの主な違いは次のとおりです：オーバーサンプリングの場合、1 fclk サイクルでサンプリングが 4 回行われ、通常のデシリアライズの場合、1 fclk サイクルでサンプリングが fclk の立ち上がりエッジと立ち下がりエッジで 2 回だけ行われます。

PCLK と、FCLKP, FCLKN, FCLKQP, FCLKQN との周波数関係は次のとおりです。

$$f_{PCLK} = 1/4f_{FCLKP} = 1/4f_{FCLKN} = 1/4f_{FCLKQP} = 1/4f_{FCLKQN}$$

ポート図

図 4-24 OSIDES32 のポート図



ポートの説明

表 4-25 OSIDES32 のポートの説明

ポート名	I/O	説明
D	入力	OSIDES32 データ入力信号
FCLKP	入力	高速クロック入力信号
FCLKN	入力	高速クロック入力信号
FCLKQP	入力	高速クロック入力信号
FCLKQN	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
SDTAP	入力	静的/動的遅延の選択 0 : 静的遅延を使用します 1 : 遅延を動的に調整します
VALUE	入力	VALUE の立ち下がりエッジで遅延が動的に調整されます。パルスごとに 1 遅延ステップ移動します
DLYSTEP[7:0]	入力	動的遅延値
Q[31:0]	出力	OSIDES32 データ出力信号
DF	出力	IODELAY 出力フラグ

パラメータの説明

表 4-26 OSIDES32 のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
C_STATIC_DLY	0~255	0	静的遅延ステップ数の制御
DYN_DLY_EN	“FALSE” / TRUE	“FALSE”	動的モードのイネーブルの制御
ADAPT_EN	“FALSE” / TRUE	“FALSE”	アダプティブモードのイネーブルの制御

接続ルール

OSIDES32 のデータ入力 D は、IBUF から直接取得されます。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
OSIDES32 uut(
    .Q(Q),
    .D(D),
    .FCLKP(FCLKP),
    .FCLKN(FCLKN),
    .FCLKQP(FCLKQP),
    .FCLKQN(FCLKQN),
    .PCLK(PCLK),
    .SDTAP(SDTAP),
    .VALUE(VALUE),
    .RESET(RESET),
    .DLYSTEP(DLYSTEP),
    .DF(DF)
);
defparam uut.C_STATIC_DLY=0;
defparam uut.DYN_DLY_EN=" FALSE" ;
defparam uut.ADAPT_EN=" FALSE" ;
```

VHDL でのインスタンス化：

```
COMPONENT OSIDES32(
    C_STATIC_DLY:integer:=0;
    DYN_DLY_EN:string:=" FALSE" ;
```

```

        ADAPT_EN:string:= " FALSE"
    );
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        FCLKP:IN std_logic;
        FCLKN:IN std_logic;
        FCLKQP:IN std_logic;
        FCLKQN:IN std_logic;
        PCLK:IN std_logic;
        SDTAP:IN std_logic;
        VALUE:IN std_logic;
        RESET:IN std_logic;
        DLYSTEP: IN std_logic;
        DF:OUT std_logic
    );
END COMPONENT;
 uut:OSIDES32
    GENERIC MAP (C_STATIC_DLY=>0,
                  DYN_DLY_EN=>" FALSE" ,
                  ADAPT_EN=>" FALSE"
    )
    PORT MAP (
        Q=>Q,
        D=>D,
        FCLKP=>FCLKP,
        FCLKN=>FCLKN,
        FCLKQP=>FCLKQP,
        FCLKQN=>FCLKQN,
        PCLK=>PCLK,
        SDTAP=>SDTAP,
        VALUE=>VALUE,
        RESET=>RESET,
        DLYSTEP=>DLYSTEP,
        DF=>DF
    );

```

);

4.3 DDR モードの出力ロジック

4.3.1 ODDR

プリミティブの紹介

ODDR(Dual Data Rate Output)は、ダブル・データ・レートの実現します。

機能の説明

ODDR モードは、FPGA からダブル・データ・レートの信号を転送するために使用されます。Q0 はダブル・データ・レートのデータ出力で、Q1 は Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。ODDR モードのブロック図は、タイミング図は図 4-26 に示すとおりです。

図 4-25 ODDR のブロック図

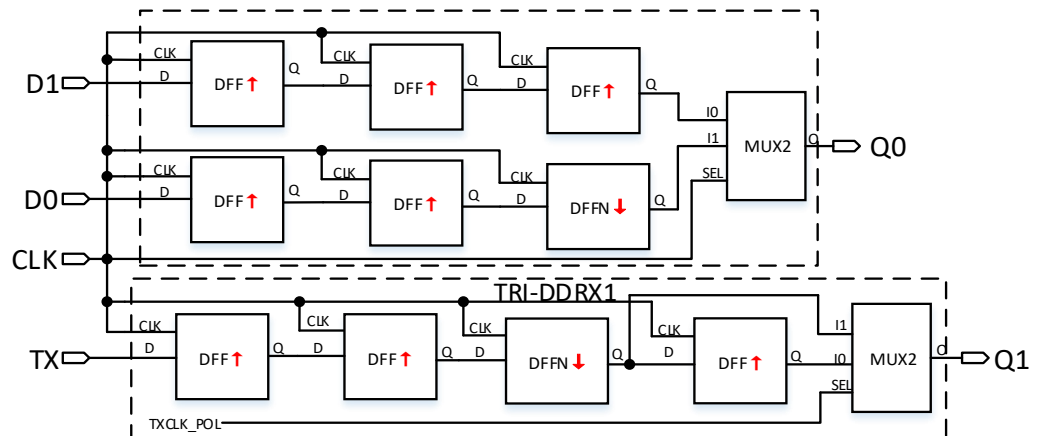
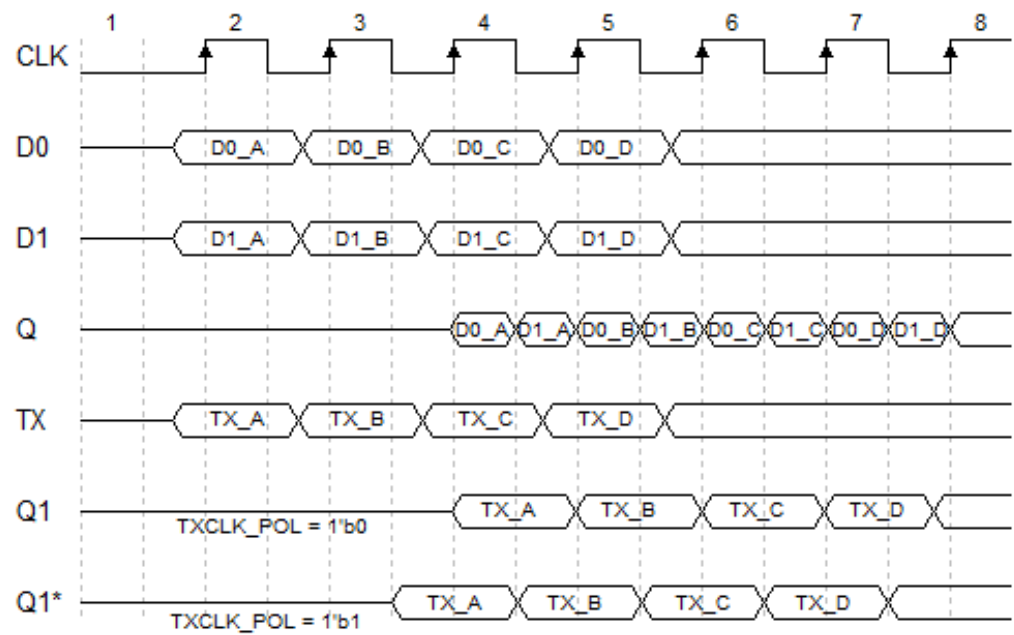


図 4-26 ODDR のタイミング図



ポート図

図 4-27 ODDR のポート図



ポートの説明

表 4-27 ODDR のポートの説明

ポート名	I/O	説明
D0, D1	入力	ODDR データ入力信号
TX	入力	TRI-DDRX1 を通じて Q1 を生成
CLK	入力	クロック入力信号
Q0	出力	ODDR データ出力信号
Q1	出力	ODDR トライステート・イネーブル信号。Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにします。

パラメータの説明

表 4-28 ODDR のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の制御

パラメータ名	値の範囲	デフォルト値	説明
			<ul style="list-style-type: none"> ● 1'b0 : Q1 は立ち上がりエッジで出力されます ● 1'b1 : Q1 は立ち下がりエッジで出力されます

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。
- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```

ODDR uut(
    .Q0(Q0),
    .Q1(Q1),
    .D0(D0),
    .D1(D1),
    .TX(TX),
    .CLK(CLK)
);
defparam uut.TXCLK_POL=1'b0;

```

VHDL でのインスタンス化 :

```

COMPONENT ODDR
    GENERIC (
        TXCLK_POL:bit:= '0'
    );
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        D0:IN std_logic;
        D1:IN std_logic;
        TX:IN std_logic;
        CLK:IN std_logic
    );

```



```
END COMPONENT;  
uut:ODDR  
    GENERIC MAP (  
        TXCLK_POL=>'0'  
    )  
    PORT MAP (  
        Q0=>Q0,  
        Q1=>Q1,  
        D0=>D0,  
        D1=>D1,  
        TX=>TX,  
        CLK=>CLK  
    );
```

4.3.2 ODDRC

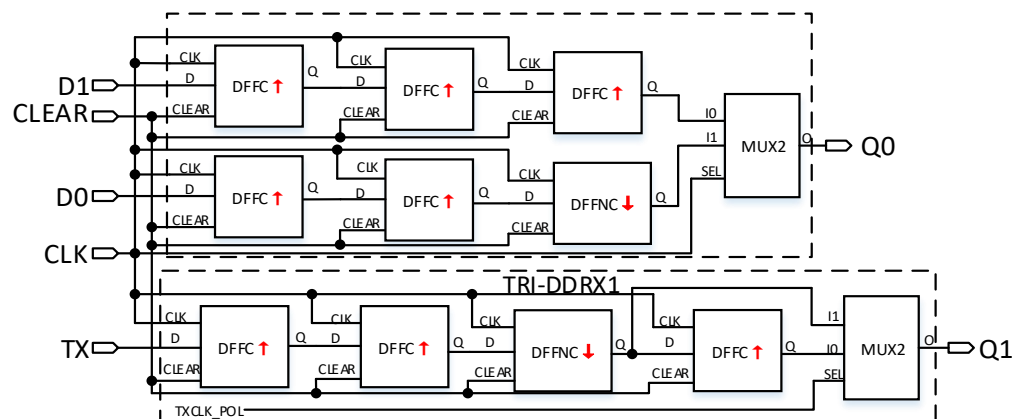
プリミティブの紹介

ODDRC(Dual Data Rate Output with Asynchronous Clear)は ODDR に比べて、非同同期リセット機能をさらに備えています。

機能の説明

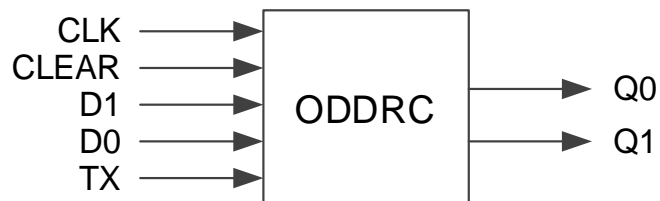
ODDRC モードは、FPGA からダブル・データ・レートの信号を転送するために使用されます。Q0 はダブル・データ・レートのデータ出力で、Q1 は Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。そのブロック図を次に示します。

図 4-28 ODDRC のブロック図



ポート図

図 4-29 ODDRC のポート図



ポートの説明

表 4-29 ODDRC のポートの説明

ポート名	I/O	説明
D0, D1	入力	ODDRC データ入力信号
TX	入力	TRI-DDRX1 を通じて Q1 を生成
CLK	入力	クロック入力信号
CLEAR	入力	非同期クリア入力、アクティブ High
Q0	出力	ODDRC データ出力
Q1	出力	ODDRC トライステート・イネーブル信号。 Q0 に接続される IOBUF/TBUF の OEN 信号 に接続するか、フローティングのままにしま す。

パラメータの説明

表 4-30 ODDRC のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の 制御 <ul style="list-style-type: none"> ● 1'b0 : Q1 は立ち上がり エッジで出力されます ● 1'b1 : Q1 は立ち下がり エッジで出力されます

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。
- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
ODDRC uut(
    .Q0(Q0),
    .Q1(Q1),
    .D0(D0),
    .D1(D1),
    .TX(TX),
    .CLK(CLK),
    .CLEAR(CLEAR)
```

```

);
defparam uut.TXCLK_POL=1'b0;
VHDL でのインスタンス化 :
COMPONENT ODDRC
    GENERIC (
        TXCLK_POL : bit := '0'
    );
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        D0:IN std_logic;
        D1:IN std_logic;
        TX:IN std_logic;
        CLK:IN std_logic;
        CLEAR:IN std_logic
    );
END COMPONENT;
uut:ODDRC
    GENERIC MAP (
        TXCLK_POL=>'0'
    )
    PORT MAP (
        Q0=>Q0,
        Q1=>Q1,
        D0=>D0,
        D1=>D1,
        TX=>TX,
        CLK=>CLK,
        CLEAR=>CLEAR
    );

```

4.3.3 OSER4

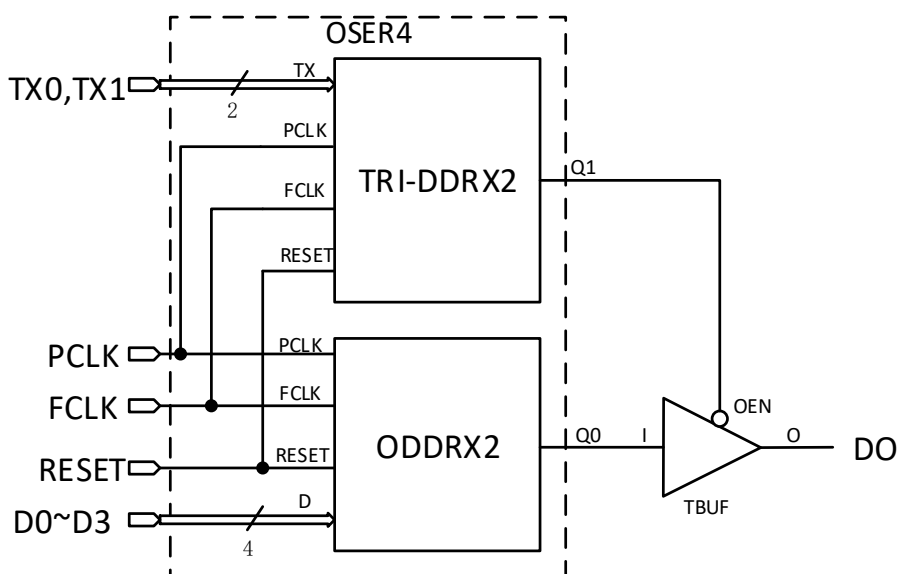
プリミティブの紹介

OSER4(4 to 1 Serializer)は、4 ビットの平行入力、1 ビットのシリアル出力のシリアライザです。

機能の説明

OSER4 モードでは、データは 4 : 1 シリアライズされます。Q0 は、OSER4 データのシリアル出力で、Q1 は Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。TX0/TX1 は、IOBUF/TBUF の OEN 入力制御信号であり、データ D0~D3 と同期に DDR 経由で出力できます。TX0/TX1 は、TRI-DDRX2 を介して、IOBUF/TBUF の OEN 信号に接続される Q1 として出力されます。D0~D3 は、ODDRX2 を介して、IOBUF/TBUF に接続されるデータ入力 I として出力されます(出力順序は D0、D1、D2、D3)。そのブロック図を次に示します。

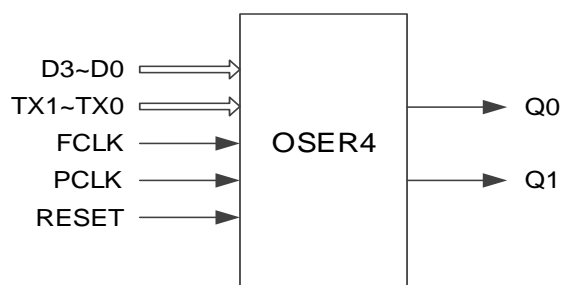
図 4-30 OSER4 のブロック図



PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/2 f_{FCLK}$ 。

ポート図

図 4-31 OSER4 のポート図



ポートの説明

表 4-31 OSER4 のポートの説明

ポート名	I/O	説明
D3~D0	入力	OSER4 データ入力信号

ポート名	I/O	説明
TX1~TX0	入力	TRI-DDRX2 を通じて Q1 を生成
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q0	出力	OSER4 データ出力信号
Q1	出力	OSER4 トライステート・イネーブル信号。 Q0 に接続される IOBUF/TBUF の OEN 信号 に接続するか、フローティングのままに することができます。

パラメータの説明

表 4-32 OSER4 のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の制御 <ul style="list-style-type: none"> ● 1'b0 : データは立ち上がりエッジで出力されます ● 1'b1 : データは立ち下がりエッジで出力されます。
HWL	"false", "true"	"false"	OSER4 データ d_up0/1 のタイミング関係の制御 <ul style="list-style-type: none"> ● "false" : d_up1 は d_up0 より 1 サイクル先です。 ● "true" : d_up1 と d_up0 のタイミングは同じです。

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。
- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```
OSER4 uut(
    .Q0(Q0),
    .Q1(Q1),
    .D0(D0),
```

```

.D1(D1),
.D2(D2),
.D3(D3),
.TX0(TX0),
.TX1(TX1),
.PCLK(PCLK),
.FCLK(FCLK),
.RESET(RESET)
);
defparam uut.HWL ="false";
defparam uut.TXCLK_POL =1'b0;

```

VHDL でのインスタンス化 :

```

COMPONENT OSER4
    GENERIC (
        HWL:string:="false";
        TXCLK_POL:bit:='0'
    );
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        D0:IN std_logic;
        D1:IN std_logic;
        D2:IN std_logic;
        D3:IN std_logic;
        TX0:IN std_logic;
        TX1:IN std_logic;
        FCLK:IN std_logic;
        PCLK:IN std_logic;
        RESET:IN std_logic
    );
END COMPONENT;
uut:OSER4
    GENERIC MAP (
        HWL=>"false",
        TXCLK_POL=>'0'
    );

```

```

)
PORT MAP (
  Q0=>Q0,
  Q1=>Q1,
  D0=>D0,
  D1=>D1,
  D2=>D2,
  D3=>D3,
  TX0=>TX0,
  TX1=>TX1,
  FCLK=>FCLK,
  PCLK=>PCLK,
  RESET=>RESET
);

```

4.3.4 OSER8

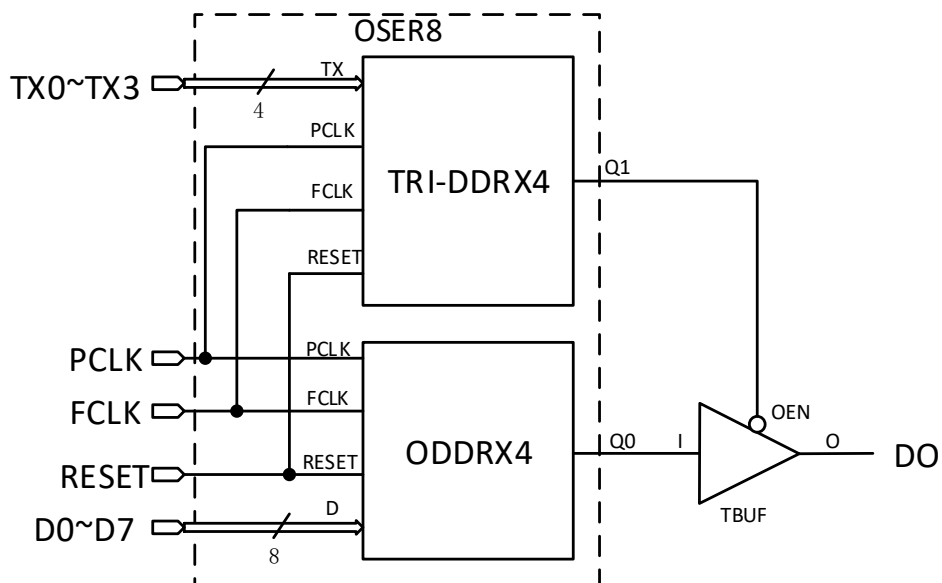
プリミティブの紹介

OSER8(8 to 1 Serializer)は、8 ビットの平行入力、1 ビットのシリアル出力のシリアライザです。

機能の説明

OSER8 モードでは、データは 8:1 シリアライズされます。Q0 は OSER8 データのシリアル出力で、Q1 は Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。そのブロック図を次に示します。

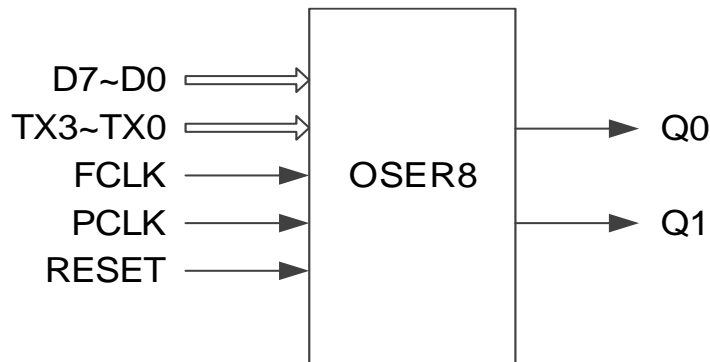
図 4-32 OSER8 のブロック図



PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/4 f_{FCLK}$ 。

ポート図

図 4-33 OSER8 のポート図



ポートの説明

表 4-33 OSER8 のポートの説明

ポート名	I/O	説明
D7~D0	入力	OSER8 データ入力信号
TX3~TX0	入力	TRI-DDRX4 を通じて Q1 を生成
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q0	出力	OSER8 データ出力信号
Q1	出力	OSER8 トライステート・イネーブル信号。Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにすることができます。

パラメータの説明

表 4-34 OSER8 のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の制御 <ul style="list-style-type: none"> ● 1'b0 : データは立ち上がりエッジで出力されます ● 1'b1 : データは立ち下がりエッジで出力されます。
HWL	"false", "true"	"false"	OSER8 データ d_up0/1 のタイミング関係の制御 <ul style="list-style-type: none"> ● "false" : d_up1 は d_up0 より 1 サイクル先です。

パラメータ名	値の範囲	デフォルト値	説明
			<ul style="list-style-type: none"> ● "true" : d_up1 と d_up0 のタイミングは同じです。

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。
- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```
OSER8 uut(
    .Q0(Q0),
    .Q1(Q1),
    .D0(D0),
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .D5(D5),
    .D6(D6),
    .D7(D7),
    .TX0(TX0),
    .TX1(TX1),
    .TX2(TX2),
    .TX3(TX3),
    .PCLK(PCLK),
    .FCLK(FCLK),
    .RESET(RESET)
);
defparam uut.HWL = "false";
defparam uut.TXCLK_POL = 1'b0;
```

VHDL でのインスタンス化 :

```
COMPONENT OSER8
  GENERIC (
    HWL:string:="false";
    TXCLK_POL:bit:='0'
  );
  PORT(
    Q0:OUT std_logic;
    Q1:OUT std_logic;
    D0:IN std_logic;
    D1:IN std_logic;
    D2:IN std_logic;
    D3:IN std_logic;
    D4:IN std_logic;
    D5:IN std_logic;
    D6:IN std_logic;
    D7:IN std_logic;
    TX0:IN std_logic;
    TX1:IN std_logic;
    TX2:IN std_logic;
    TX3:IN std_logic;
    FCLK:IN std_logic;
    PCLK:IN std_logic;
    RESET:IN std_logic
  );
END COMPONENT;

 uut:OSER8
  GENERIC MAP (
    HWL=>"false",
    TXCLK_POL=>'0'
  )
  PORT MAP (
    Q0=>Q0,
    Q1=>Q1,
    D0=>D0,
    D1=>D1,
```

```
D2=>D2,  
D3=>D3,  
D4=>D4,  
D5=>D5,  
D6=>D6,  
D7=>D7,  
TX0=>TX0,  
TX1=>TX1,  
TX2=>TX2,  
TX3=>TX3,  
FCLK=>FCLK,  
PCLK=>PCLK,  
RESET=>RESET  
);
```

4.3.5 OSER10

プリミティブの紹介

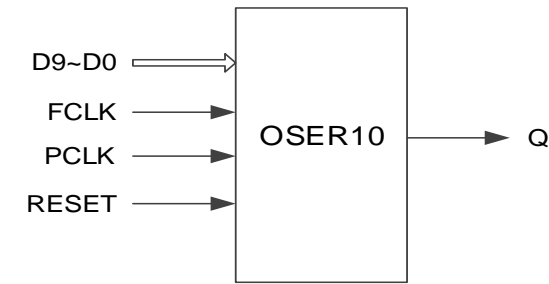
OSER10(10 to 1 Serializer)は、10 ビットのパラレル入力、1 ビットのシリアル出力のシリアライザです。

機能の説明

OSER10 モードでは、データは 10:1 シリアライズされます。PCLK は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/5 f_{FCLK}$ 。

ポート図

図 4-34 OSER10 のポート図



ポートの説明

表 4-35 OSER10 のポートの説明

ポート名	I/O	説明
D9~D0	入力	OSER10 データ入力信号
FCLK	入力	高速クロック入力信号

ポート名	I/O	説明
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q	出力	OSER10 データ出力信号

接続ルール

Q は、OBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化 :

```
OSER10 uut(
    .Q(Q),
    .D0(D0),
    .D1(D1),
    .D2(D2),
    .D3(D3),
    .D4(D4),
    .D5(D5),
    .D6(D6),
    .D7(D7),
    .D8(D8),
    .D9(D9),
    .PCLK(PCLK),
    .FCLK(FCLK),
    .RESET(RESET)
);
```

VHDL でのインスタンス化 :

```
COMPONENT OSER10
PORT(
    Q:OUT std_logic;
    D0:IN std_logic;
    D1:IN std_logic;
    D2:IN std_logic;
    D3:IN std_logic;
```

```

        D4:IN std_logic;
        D5:IN std_logic;
        D6:IN std_logic;
        D7:IN std_logic;
        D8:IN std_logic;
        D9:IN std_logic;
        FCLK:IN std_logic;
        PCLK:IN std_logic;
        RESET:IN std_logic

    );
END COMPONENT;
uut:OSER10
    PORT MAP (
        Q=>Q,
        D0=>D0,
        D1=>D1,
        D2=>D2,
        D3=>D3,
        D4=>D4,
        D5=>D5,
        D6=>D6,
        D7=>D7,
        D8=>D8,
        D9=>D9,
        FCLK=>FCLK,
        PCLK=>PCLK,
        RESET=>RESET
    );

```

4.3.6 OVIDEO

プリミティブの紹介

OVIDEO(7 to 1 Serializer)は、7 ビットのパラレル入力、1 ビットのシリアル出力のシリアライザです。

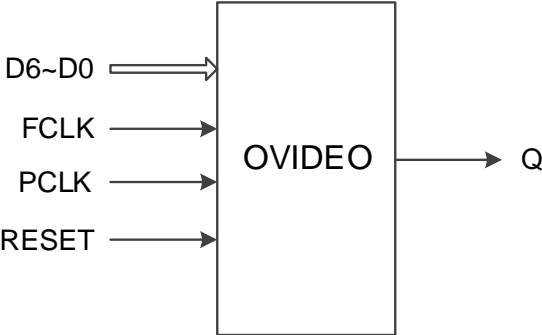
機能の説明

OVIDEO モードでは、データは 7:1 シリアライズされます。PCLK は

通常、FCLK の分周によって得られます： $f_{PCLK} = 1/3.5 f_{FCLK}$ 。

ポート図

図 4-35 OVIDEO のポート図



ポートの説明

表 4-36 OVIDEO のポートの説明

ポート名	I/O	説明
D6~D0	入力	OVIDEO データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q	出力	OVIDEO データ出力

接続ルール

Q は、OBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
OVIDEO uut(  
    .Q(Q),  
    .D0(D0),  
    .D1(D1),  
    .D2(D2),  
    .D3(D3),  
    .D4(D4),  
    .D5(D5),
```

```
.D6(D6),  
.PCLK(PCLK),  
.FCLK(FCLK),  
.RESET(RESET)  
);  
VHDL でのインスタンス化 :  
COMPONENT OVIDEO  
  PORT(  
    Q:OUT std_logic;  
    D0:IN std_logic;  
    D1:IN std_logic;  
    D2:IN std_logic;  
    D3:IN std_logic;  
    D4:IN std_logic;  
    D5:IN std_logic;  
    D6:IN std_logic;  
    FCLK:IN std_logic;  
    PCLK:IN std_logic;  
    RESET:IN std_logic  
  );  
END COMPONENT;  
 uut:OVIDEO  
  PORT MAP (  
    Q=>Q,  
    D0=>D0,  
    D1=>D1,  
    D2=>D2,  
    D3=>D3,  
    D4=>D4,  
    D5=>D5,  
    D6=>D6,  
    FCLK=>FCLK,  
    PCLK=>PCLK,  
    RESET=>RESET  
  );
```


4.3.7 OSER16

プリミティブの紹介

OSER16(16 to 1 Serializer)は、16 ビットの平行入力、1 ビットのシリアル出力のシリアライザです。

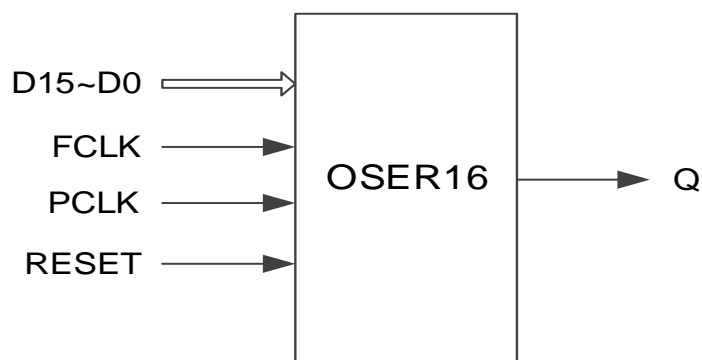
機能の説明

OSER16 モードでは、データは 16:1 シリアライズされます。PCLK

は通常、FCLK の分周によって得られます： $f_{PCLK} = 1/8 f_{FCLK}$ 。

ポート図

図 4-36 OSER16 のポート図



ポートの説明

表 4-37 OSER16 のポートの説明

ポート名	I/O	説明
D15~D0	入力	OSER16 データ入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q	出力	OSER16 データ出力信号

接続ルール

Q は、OBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。

プリミティブのインスタンス化

プリミティブを直接インスタンス化することができます。

Verilog でのインスタンス化：

```
OSER16 uut(
    .Q(Q),
```

```
.D0(D0),  
.D1(D1),  
.D2(D2),  
.D3(D3),  
.D4(D4),  
.D5(D5),  
.D6(D6),  
.D7(D7),  
.D8(D8),  
.D9(D9),  
.D10(D10),  
.D11(D11),  
.D12(D12),  
.D13(D13),  
.D14(D14),  
.D15(D15),  
.PCLK(PCLK),  
.FCLK(FCLK),  
.RESET(RESET)  
);
```

VHDL でのインスタンス化 :

```
COMPONENT OSER16  
PORT(  
    Q:OUT std_logic;  
    D0:IN std_logic;  
    D1:IN std_logic;  
    D2:IN std_logic;  
    D3:IN std_logic;  
    D4:IN std_logic;  
    D5:IN std_logic;  
    D6:IN std_logic;  
    D7:IN std_logic;  
    D8:IN std_logic;  
    D9:IN std_logic;  
    D10:IN std_logic;
```

```
D11:IN std_logic;
D12:IN std_logic;
D13:IN std_logic;
D14:IN std_logic;
D15:IN std_logic;
FCLK:IN std_logic;
PCLK:IN std_logic;
RESET:IN std_logic
);
END COMPONENT;
uut:OSER16
PORT MAP (
    Q=>Q,
    D0=>D0,
    D1=>D1,
    D2=>D2,
    D3=>D3,
    D4=>D4,
    D5=>D5,
    D6=>D6,
    D7=>D7,
    D8=>D8,
    D9=>D9,
    D10=>D10,
    D11=>D11,
    D12=>D12,
    D13=>D13,
    D14=>D14,
    D15=>D15,
    FCLK=>FCLK,
    PCLK=>PCLK,
    RESET=>RESET
);
```

4.3.8 ODDR_MEM

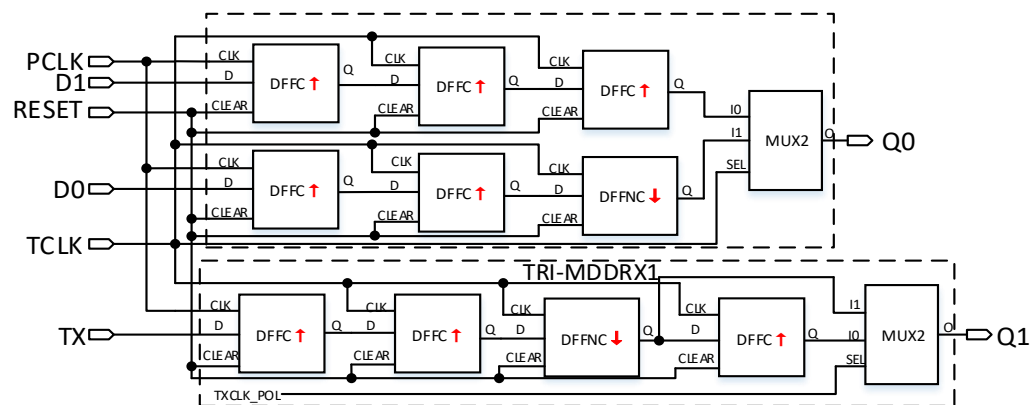
プリミティブの紹介

ODDR_MEM(Dual Data Rate Output with Memory)は、メモリ付きのダブル・データ・レートの実現します。

機能の説明

ODDR_MEM モードは、FPGA からダブル・データ・レートの信号を転送するために使用されます。ODDR とは異なり、ODDR_MEM には DQS が必要です。TCLK は DQS の出力信号 DQSW0 または DQSW270 に接続され、データは TCLK のクロックエッジに基づいて ODDR_MEM から出力されます。ODDR_MEM の Q0 は、ダブル・データ・レートのデータ出力であり、Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。そのブロック図を示します。

図 4-37 ODDR_MEM のブロック図

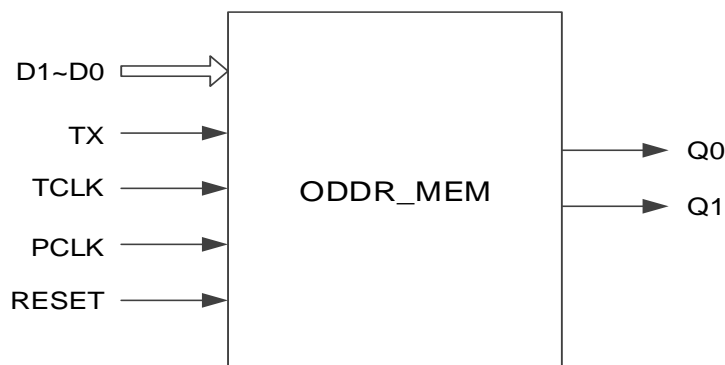


PCLK と TCLK の周波数関係は： $f_{PCLK} = f_{TCLK}$ 。

PCLK と TCLK の間には一定の位相関係があり、その位相関係は DQS の DLLSTEP 値および WSTEP 値により決定できます。

ポート図

図 4-38 ODDR_MEM のポート図



ポートの説明

表 4-38 ODDR_MEM のポートの説明

ポート名	I/O	説明
D1~D0	入力	ODDR_MEM データ入力信号
TX	入力	TRI-MDDR1X1 を通じて Q1 を生成
TCLK	入力	DQS モジュールの DQSW0 または DQSW270 からのクロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q0	出力	ODDR_MEM データ出力
Q1	出力	ODDR_MEM トライステート・イネーブル信号。 Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにします。

パラメータの説明

表 4-39 ODDR_MEM のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の制御 <ul style="list-style-type: none"> 1'b0 : データは立ち上がりエッジで出力されます。 1'b1 : データは立ち下がりエッジで出力されます。
TCLK_SOURCE	"DQSW", "DQSW270"	"DQSW"	TCLK のソースの選択 <ul style="list-style-type: none"> "DQSW" : DQS モジュールの DQSW0 から。 "DQSW270" : DQS モジュールの DQSW270 から。

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。
- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。
- TCLK は、DQS モジュールの DQSW0 または DQSW270 に接続し、対応するパラメータを構成する必要があります。

プリミティブのインスタンス化

Verilog でのインスタンス化：

```

ODDR_MEM oddr_mem_inst(
    .Q0(q0),
    .Q1(q1),
    .D0(d0),
    .D1(d1),
    .TX(tx),
    .TCLK(tclk),
    .PCLK(pclk),
    .RESET(reset)
);
defparam uut.TCLK_SOURCE ="DQSW";
defparam uut.TXCLK_POL=1'b0;

```

VHDL でのインスタンス化 :

```

COMPONENT ODDR_MEM
    GENERIC (
        TXCLK_POL:bit:='0';
        TCLK_SOURCE:string:="DQSW"
    );
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        D0:IN std_logic;
        D1:IN std_logic;
        TX:IN std_logic;
        TCLK:IN std_logic;
        PCLK:IN std_logic;
        RESET:IN std_logic
    );
END COMPONENT;
uut:ODDR_MEM
    GENERIC MAP (
        TXCLK_POL=>'0',
        TCLK_SOURCE=>"DQSW"
    )
    PORT MAP (

```

```

Q0=>q0,
Q1=>q1,
D0=>d0,
D1=>d1,
TX=>tx,
TCLK=>tclk,
PCLK=>pclk,
RESET=>reset
);

```

4.3.9 OSER4_MEM

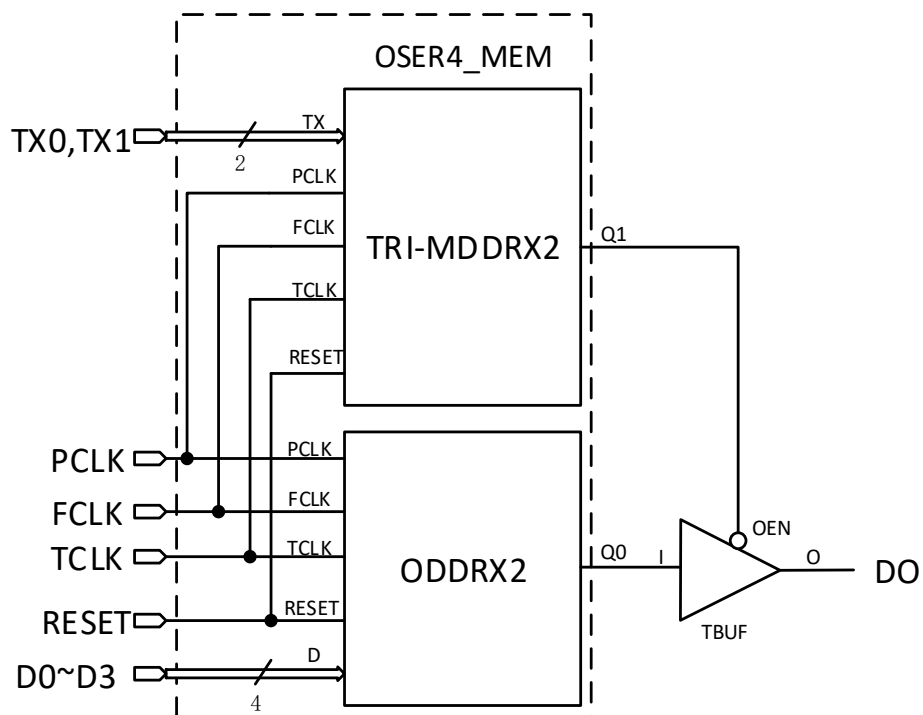
プリミティブの紹介

OSER4_MEM(4 to 1 Serializer with Memory)は、メモリ付きの 4:1 シリアライザです。

機能の説明

OSER4_MEM モードでは、データは 4 : 1 シリアライズされます。OSER4 とは異なり、OSER4_MEM には DQS が必要です。TCLK は DQS の出力信号 DQSW0 または DQSW270 に接続され、データは TCLK のクロックエッジに基づいて OSER4_MEM から出力されます。OSER4_MEM の Q0 は、データのシリアル出力で、Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。そのブロック図を次に示します。

図 4-39 OSER4_MEM のブロック図



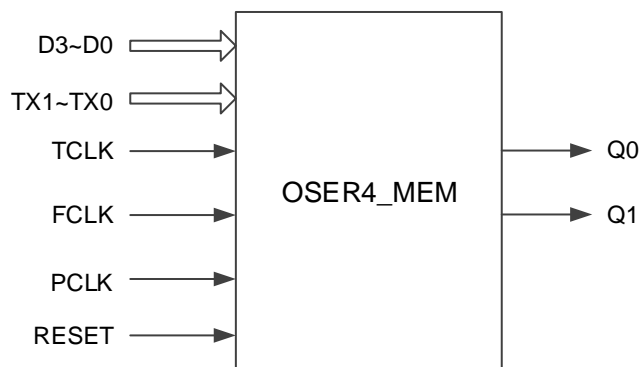
PCLK、FCLK、および TCLK の周波数関係は：

$$f_{PCLK} = 1/2 f_{FCLK} = 1/2 f_{TCLK}。$$

FCLK と TCLK の間には一定の位相関係があり、その位相関係は DQS の DLLSTEP 値および WSTEP 値により決定できます。

ポート図

図 4-40 OSER4_MEM のポート図



ポートの説明

表 4-40 OSER4_MEM のポートの説明

ポート名	I/O	説明
D3~D0	入力	OSER4_MEM データ入力信号
TX1~TX0	入力	TRI-MDDR2 を通じて Q1 を生成
TCLK	入力	DQS モジュールの DQSW0 または DQSW270 からのクロック入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q0	出力	OSER4_MEM データ出力信号
Q1	出力	OSER4_MEM トライステート・イネーブル信号。Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにすることができます。

パラメータの説明

表 4-41 OSER4_MEM のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の制御 <ul style="list-style-type: none"> ● 1'b0 : データは立ち上がりエッジで出力されます。 ● 1'b1 : データは立ち下がりエッジで出力されます。
TCLK_SOURCE	"DQSW","DQSW270"	" DQSW "	TCLK のソースの選択 <ul style="list-style-type: none"> ● "DQSW" : DQS モジュールの DQSW0 から。 ● "DQSW270" : DQS モジュールの DQSW270 から。
HWL	"false", "true"	"false"	OSER4_MEM データ d_up0/1 のタイミング関係の制御 <ul style="list-style-type: none"> ● "false" : d_up1 は d_up0 より 1 サイクル先です。 ● "true" : d_up1 と d_up0 のタイミングは同じです。

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュールを介してその入力ポート DI に接続します。
- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。
- TCLK は、DQS モジュールの DQSW0 または DQSW270 に接続し、対応するパラメータを構成する必要があります。

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
OSER4_MEM oser4_mem_inst(
    .Q0(q0),
    .Q1(q1),
    .D0(d0),
```

```

        .D1(d1),
        .D2(d2),
        .D3(d3),
        .TX0(tx0),
        .TX1(tx1),
        .TCLK(tclk),
        .FCLK(fclk),
        .PCLK(pclk),
        .RESET(reset)
    );
    defparam uut.HWL ="false";
    defparam uut.TCLK_SOURCE ="DQSW";
    defparam uut.TXCLK_POL=1'b0;
VHDL でのインスタンス化 :
    COMPONENT OSER4_MEM
        GENERIC (
            HWL:string:="false";
            TXCLK_POL:bit:='0';
            TCLK_SOURCE:string:="DQSW"
        );
    PORT(
        Q0:OUT std_logic;
        Q1:OUT std_logic;
        D0:IN std_logic;
        D1:IN std_logic;
        D2:IN std_logic;
        D3:IN std_logic;
        TX0:IN std_logic;
        TX1:IN std_logic;
        TCLK:IN std_logic;
        FCLK:IN std_logic;
        PCLK:IN std_logic;
        RESET:IN std_logic
    );
    END COMPONENT;

```

```

uut:OSER4_MEM
    GENERIC MAP (
        HWL=>"false",
        TXCLK_POL=>'0',
        TCLK_SOURCE=>"DQSW"
    )
    PORT MAP (
        Q0=>q0,
        Q1=>q1,
        D0=>d0,
        D1=>d1,
        D2=>d2,
        D3=>d3,
        TX0=>tx0,
        TX1=>tx1,
        TCLK=>tclk,
        FCLK=>fclk,
        PCLK=>pclk,
        RESET=>reset
    );

```

4.3.10 OSER8_MEM

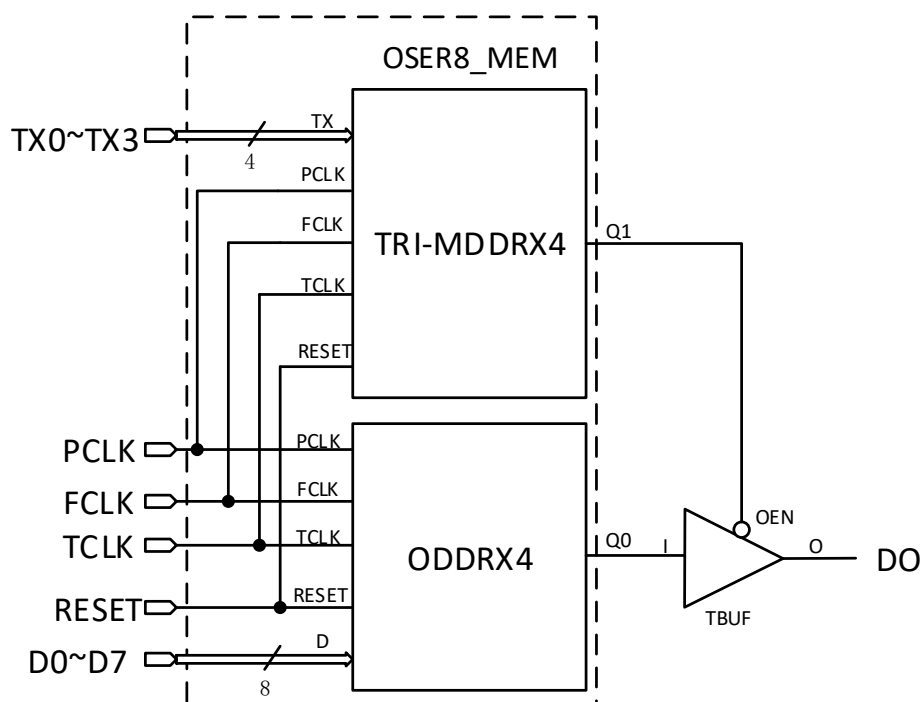
プリミティブの紹介

OSER8_MEM(8 to 1 Serializer with Memory)は、メモリ付きの 8:1 シリアルライザです。

機能の説明

OSER8_MEM モードでは、データは 8:1 シリアルライズされます。OSER8 とは異なり、OSER8_MEM には DQS が必要です。TCLK は DQS の出力信号 DQSW0 または DQSW270 に接続され、データは TCLK のクロックエッジに基づいて OSER8_MEM から出力されます。OSER8_MEM の Q0 は、データのシリアル出力で、Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に使用されます。そのブロック図を次に示します。

図 4-41 OSER8_MEM のブロック図



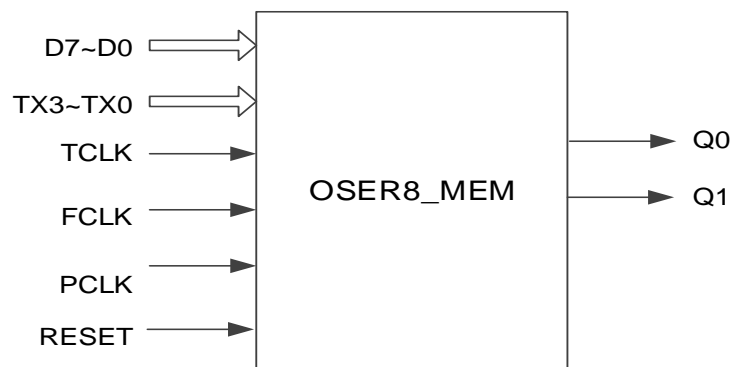
PCLK、FCLK、および TCLK の周波数関係は：

$$f_{PCLK} = 1/4 f_{FCLK} = 1/4 f_{TCLK}。$$

FCLK と TCLK の間には一定の位相関係があり、その位相関係は DQS の DLLSTEP 値および WSTEP 値により決定できます。

ポート図

図 4-42 OSER8_MEM のポート図



ポートの説明

表 4-42 OSER8_MEM のポートの説明

ポート名	I/O	説明
D7~D0	入力	OSER8_MEM データ入力信号
TX3~TX0	入力	TRI-MDDR4 を通じて Q1 を生成

ポート名	I/O	説明
TCLK	入力	DQS モジュールの DQSW0 または DQSW270 からのクロック入力信号
FCLK	入力	高速クロック入力信号
PCLK	入力	マスタークロック入力信号
RESET	入力	非同期リセット入力、アクティブ High
Q0	出力	OSER8_MEM データ出力信号
Q1	出力	OSER8_MEM トライステート・イネーブル信号。Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにすることができます。

パラメータの説明

表 4-43 OSER8_MEM のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
TXCLK_POL	1'b0, 1'b1	1'b0	Q1 の出力クロックの極性の制御 <ul style="list-style-type: none"> ● 1'b0 : データは立ち上がりエッジで出力されます ● 1'b1 : データは立ち下がりエッジで出力されます。
TCLK_SOURCE	"DQSW","DQSW270"	" DQSW "	TCLK のソースの選択 <ul style="list-style-type: none"> ● "DQSW" : DQS モジュールの DQSW0 から。 ● DQSW270" : DQS モジュールの DQSW270 から。
HWL	"false", "true"	"false"	OSER8_MEM データ d_up0/1 のタイミング関係の制御 <ul style="list-style-type: none"> ● "false" : d_up1 は d_up0 より 1 サイクル先です。 ● "true" : d_up1 と d_up0 のタイミングは同じです。

接続ルール

- Q0 は、OBUF/IOBUF/TBUF に直接接続するか、IODELAY モジュール

を介してその入力ポート DI に接続します。

- Q1 は、Q0 に接続される IOBUF/TBUF の OEN 信号に接続するか、フローティングのままにする必要があります。
- TCLK は、DQS モジュールの DQSW0 または DQSW270 に接続し、対応するパラメータを構成する必要があります。

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
OSER8_MEM oser8_mem_inst(
    .Q0(q0),
    .Q1(q1),
    .D0(d0),
    .D1(d1),
    .D2(d2),
    .D3(d3),
    .D4 (d4),
    .D5 (d5),
    .D6 (d6),
    .D7 (d7),
    .TX0 (tx0),
    .TX1 (tx1),
    .TX2 (tx2),
    .TX3 (tx3),
    .TCLK (tclk),
    .FCLK (fclk),
    .PCLK (pclk),
    .RESET(reset)
);
defparam uut.HWL ="false";
defparam uut.TCLK_SOURCE ="DQSW";
defparam uut.TXCLK_POL=1'b0;
```

VHDL でのインスタンス化 :

```
COMPONENT OSER8_MEM
    GENERIC (
        HWL:string:="false";
        TXCLK_POL:bit:='0';
```

```
        TCLK_SOURCE:string:="DQSW"
    );
PORT(
    Q0:OUT std_logic;
    Q1:OUT std_logic;
    D0:IN std_logic;
    D1:IN std_logic;
    D2:IN std_logic;
    D3:IN std_logic;
    D4:IN std_logic;
    D5:IN std_logic;
    D6:IN std_logic;
    D7:IN std_logic;
    TX0:IN std_logic;
    TX1:IN std_logic;
    TX2:IN std_logic;
    TX3:IN std_logic;
    TCLK:IN std_logic;
    FCLK:IN std_logic;
    PCLK:IN std_logic;
    RESET:IN std_logic
);
END COMPONENT;
 uut:OSER8_MEM
    GENERIC MAP (
        HWL=>"false",
        TXCLK_POL=>'0',
        TCLK_SOURCE=>"DQSW"
    )
PORT MAP (
    Q0=>q0,
    Q1=>q1,
    D0=>d0,
    D1=>d1,
    D2=>d2,
```

```
D3=>d3,  
D4=>d4,  
D5=>d5,  
D6=>d6,  
D7=>d7,  
TX0=>tx0,  
TX1=>tx1,  
TX2=>tx2,  
TX3=>tx3,  
TCLK=>tclk,  
FCLK=>fclk,  
PCLK=>pclk,  
RESET=>reset  
);
```

4.4 遅延モジュール

4.4.1 IODELAY

プリミティブの紹介

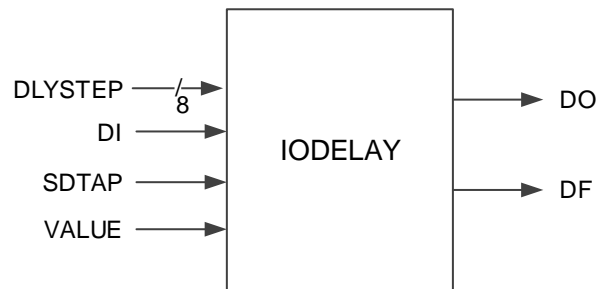
IODELAY(Input/Output delay)は、IO ブロックに含まれるプログラマブルな遅延モジュールです。

機能の説明

Arora V FPGA 製品は、合計 256 (0～255)ステップの遅延(1 ステップは約 12.5ps)を提供できる IODELAY モジュールをサポートします。IODELAY は、IO ロジックと一般ロジックの入出力遅延に使用できます。IO の IODELAY は、それぞれ入力ロジックと出力ロジックに作用する IDELAY と ODELAY に分けられます。遅延モードには、静的モード、動的モード、およびアダプティブモードがあります。

ポート図

図 4-43 IODELAY のポート図



ポートの説明

表 4-44 IODELAY のポートの説明

ポート名	I/O	説明
DI	入力	データ入力信号
SDTAP	入力	静的/動的遅延の選択 <ul style="list-style-type: none"> ● 0 : 静的遅延を使用します ● 1 : 遅延を動的に調整します
VALUE	入力	VALUE の立ち下がりエッジで遅延が動的に調整されます。パルスごとに 1 遅延ステップ移動します
DLYSTEP[7:0]	入力	動的遅延値
DO	出力	データ出力信号
DF	出力	出力フラグ

パラメータの説明

表 4-45 IODELAY のパラメータの説明

パラメータ名	値の範囲	デフォルト値	説明
C_STATIC_DLY	0~255	0	静的遅延ステップ数の制御
DYN_DLY_EN	“FALSE” / “TRUE”	“FALSE”	動的モードのイネーブルの制御
ADAPT_EN	“FALSE” / “TRUE”	“FALSE”	アダプティブモードのイネーブルの制御

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
IODELAY iodelay_inst(
```

```

        .DO(dout),
        .DF(df),
        .DI(di),
        .SDTAP(sdtap),
        .VALUE(value),
        .DLYSTEP(dlystep)
    );
    defparam iodelay_inst.C_STATIC_DLY=0;
    defparam iodelay_inst.DYN_DLY_EN=" FALSE" ;
    defparam iodelay_inst.ADAPT_EN=" FALSE" ;

```

VHDL でのインスタンス化 :

```

COMPONENT IODELAY
    GENERIC (C_STATIC_DLY:integer:=0;
             DYN_DLY_EN:string:= " FALSE" ;
             ADAPT_EN:string:= " FALSE"

    );
    PORT(
        DO:OUT std_logic;
        DF:OUT std_logic;
        DI:IN std_logic;
        SDTAP:IN std_logic;
        VALUE:IN std_logic;
        DLYSTEP:IN std_logic
    );
END COMPONENT;

 uut:IODELAY
    GENERIC MAP (C_STATIC_DLY=>0,
                 DYN_DLY_EN=>" FALSE" ,
                 ADAPT_EN=>" FALSE"

    )
    PORT MAP (
        DO=>dout,
        DF=>df,
        DI=>di,

```

```
SDTAP=>sdtap,  
VALUE=>value,  
DLYSTEP=>dlystep  
);
```

