

# アルテラ GPIO IP コアの利用ガイド

インテル® Quartus® Prime 開発環境の更新情報: **17.0**

## 目次

---

<b>アルテラ GPIO IP コアของผู้ใช้指南.....</b>	<b>3</b>
アルテラ GPIO IP コアの機能.....	3
アルテラ GPIO IP コアのデータパス.....	4
入力パス.....	5
出力および出力イネーブルパス.....	6
アルテラ GPIO のインターフェイス信号.....	8
共有信号.....	10
データ・インターフェイスのためのデータのビット順.....	10
入力バスと出力バスの上位ビットおよび下位ビット.....	11
データ・インターフェイス信号と対応するクロック.....	11
リソース使用率とデザインのパフォーマンスの検証.....	12
アルテラ GPIO のパラメーター設定.....	12
レジスターパッキング.....	14
アルテラ GPIO のタイミング.....	15
タイミング・コンポーネント.....	15
遅延エレメント.....	17
タイミング解析.....	17
タイミング・クロージャーのガイドライン.....	20
アルテラ GPIO IP コアのデザイン例.....	20
アルテラ GPIO 合成可能な Quartus Prime デザイン例.....	21
アルテラ GPIO IP コア・シミュレーションのデザイン例.....	21
Arria V、Cyclone V、および Stratix V デバイスの IP マイグレーション・フロー.....	22
ALTDDIO_IN、ALTDDIO_OUT、ALTDDIO_BIDIR、および ALTIOBUF IP コアの移行.....	22
ガイドライン: 移行された IP における datain_h と datain_l ポートのスワップ.....	23
アルテラ GPIO IP コアของผู้ใช้指南のアーカイブ.....	23
改訂履歴.....	23



## アルテラ GPIO IP コアの利用ガイド

アルテラ GPIO IP コアは GPIO (汎用 I/O) の機能とコンポーネントをサポートしています。トランシーバー、メモリー・インターフェイス、または LVDS に特化していない一般的なアプリケーションでは、GPIO を使用できます。

アルテラ GPIO IP コアは、インテル® Arria® 10 デバイスでのみ使用できます。Stratix® V、Arria V、または Cyclone® IV デバイスからデザインを移行する場合は、ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、または ALTIOBUF IP コアを移行する必要があります。

### 関連情報

- 22 ページの [Arria V、Cyclone V、および Stratix V デバイスの IP マイグレーション・フロー](#)
- [Introduction to Intel FPGA IP Cores](#)  
パラメーター化、アップグレード、IP コアのシミュレーションを含むすべてのインテル FPGA IP コアに関する基本的な情報を提供します。
- [Creating Version-Independent IP and Qsys Simulation Scripts](#)  
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)  
プロジェクトと IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。
- 23 ページの [アルテラ GPIO IP コアの利用ガイドのアーカイブ](#)  
前バージョンのアルテラ GPIO IP コア向けのユーザーガイドのリストを提供します。
- [Double Data Rate I/O \(ALTDDIO\\_IN, ALTDDIO\\_OUT, and ALTDDIO\\_BIDIR\) IP Cores User Guide](#)
- [I/O Buffer \(ALTIOBUF\) IP Core User Guide](#)

## アルテラ GPIO IP コアの機能

アルテラ GPIO IP コアには、デバイス I/O ブロックをサポートする機能が含まれています。Quartus Prime パラメーター・エディターを使用して、アルテラ GPIO IP コアをコンフィグレーションすることができます。

アルテラ GPIO IP コアには、以下のコンポーネントがあります。

- DDIO (ダブル・データレート入力/出力) : 通信チャネルのデータレートを 2 倍または半分にするデジタル・コンポーネントです。
- 遅延チェーン: 特定の遅延を実行し、I/O タイミング・クロージャを支援するために、遅延チェーンをコンフィグレーションします。
- I/O バッファ: パッドを FPGA に接続します。

## アルテラ GPIO IP コアのデータパス

図 -1: シングルエンド GPIO の上位レベルの図

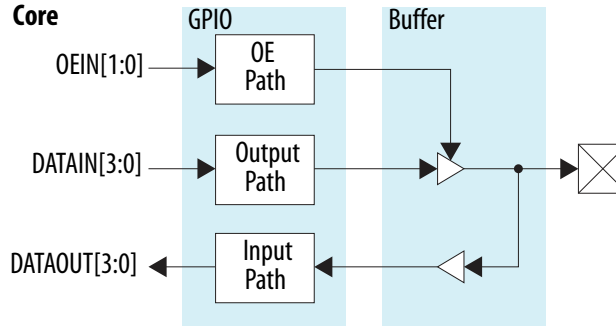


表 1. アルテラ GPIO データパスのモード

データパス	Register Mode			
	Bypass	Simple Register	DDR I/O	
			Full-Rate	Half-Rate
入力	データはすべての DDIO (ダブルデータレート I/O) をバイパスして、遅延エレメントからコアへ送信されます。	フルレート DDIO は、ハーフレート DDIO をバイパスし、シンプルレジスターとして動作します。フィッターは、エリアとタイミングのトレードオフに応じて、I/O にレジスターをバックするまたはコアにレジスターを実装することを選択します。	フルレート DDIO は、ハーフレート DDIO をバイパスし、通常の DDIO として動作します。	フルレート DDIO は通常の DDIO として動作します。ハーフレート DDIO は、フルレートのデータをハーフレートに変換します。
出力	データはすべての DDIO をバイパスして、コアから遅延エレメントに直接送信されます。	フルレート DDIO は、ハーフレート DDIO をバイパスし、シンプルレジスターとして動作します。フィッターは、エリアとタイミングのトレードオフに応じて、I/O にレジスターをバックするまたはコアにレジスターを実装することを選択します。	フルレート DDIO は、ハーフレート DDIO をバイパスし、通常の DDIO として動作します。	フルレート DDIO は通常の DDIO として動作します。ハーフレート DDIO は、フルレートのデータをハーフレートに変換します。
双方向	出力バッファは、出力ピンと入力バッファの両方を駆動します。	フルレート DDIO はシンプルレジスターとして動作します。出力バッファは出力ピンと入力バッファの両方を駆動します。	フルレート DDIO は通常の DDIO として動作します。出力バッファは出力ピンと入力バッファの両方を駆動します。入力バッファは 3 つのフリップフロップのセットを駆動します。	フルレート DDIO は通常の DDIO として動作します。ハーフレート DDIO は、フルレートのデータをハーフレートに変換します。出力バッファは出力ピンと入力バッファの両方を駆動します。入力バッファは 3 つのフリップフロップのセットを駆動します。

非同期クリアとプリセット信号を使用する場合、すべての DDIO は同じ信号を共有します。

ハーフレートおよびフルレート DDIO は、別々のクロックに接続します。ハーフレートおよびフルレート DDIO を使用すると、フルレートのクロックは 2 倍のハーフレート周波数で動作する必要があります。タイミング要件を満たすために、異なる位相関係を使用できます。

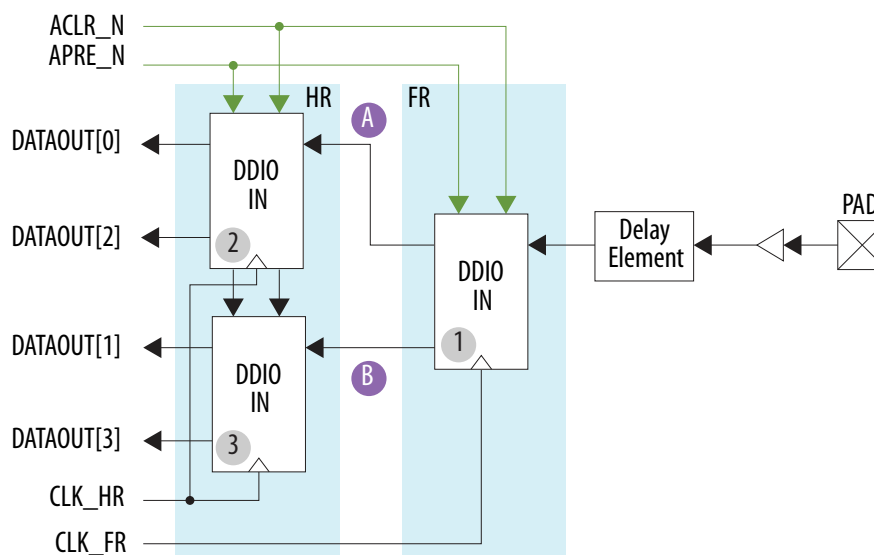
## 関連情報

11 ページの [入力バスと出力バスの上位ビットおよび下位ビット](#)

## 入力バス

パッドは入力バッファにデータを送信し、入力バッファは遅延エレメントにデータを送ります。データが遅延エレメントの出力に送られた後、プログラマブル・バイパス・マルチプレクサーは、使用する機能とパスを選択します。各 入力バスには、フルレートとハーフレートの 2 つの DDIO ステージが含まれています。

図 -2: シングルエンド GPIO 入力バスの簡略図

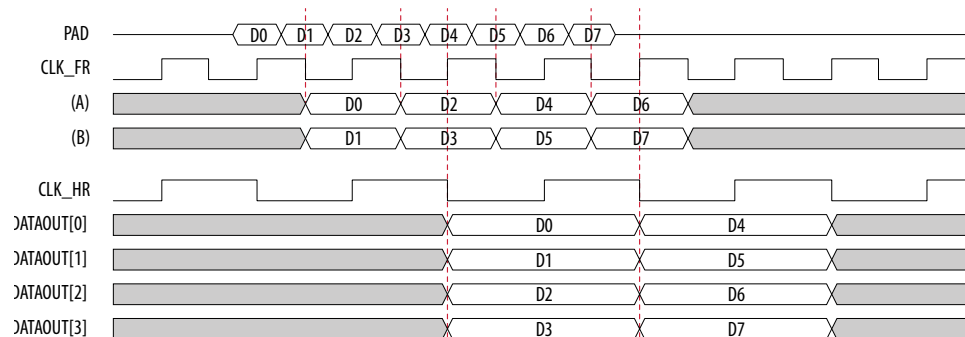


1. パッドはデータを受信します。
2. DDIO IN (1) は CLK\_FR の立ち上がりエッジと立ち下がりエッジでデータをキャプチャーし、以下の波形図の信号 (A) と (B) のデータをシングル・データレートで送信します。
3. DDIO IN (2) と DDIO IN (3) はデータレートを半分にします。
4. DATAOUT [3:0] は、データをハーフレート・バスとして提示します。

図 -3: ハーフレート変換を使用した DDIO モードの入力パス波形

以下の図では、データは、ダブル・データレートのフルレートクロックからシングル・データレートのハーフレート・クロックに移行します。データレートは 4 で分周され、バスのサイズは同じ比率で増加します。アルテラ GPIO IP コアを介した全体のスレーブットは変わりません。

異なる信号間の実際のタイミング関係は、フルレートおよびハーフレートのクロック向けに選択した特定のデザイン、遅延、および位相に応じて異なる場合があります。



**注意:** アルテラ GPIO IP コアは、入力パスのダイナミック・キャリブレーションをサポートしていません。入力パスのダイナミック・キャリブレーションを必要とするアプリケーションの場合、関連情報を参照してください。

#### 関連情報

[Altera PHYLite for Parallel Interfaces IP Core User Guide](#)

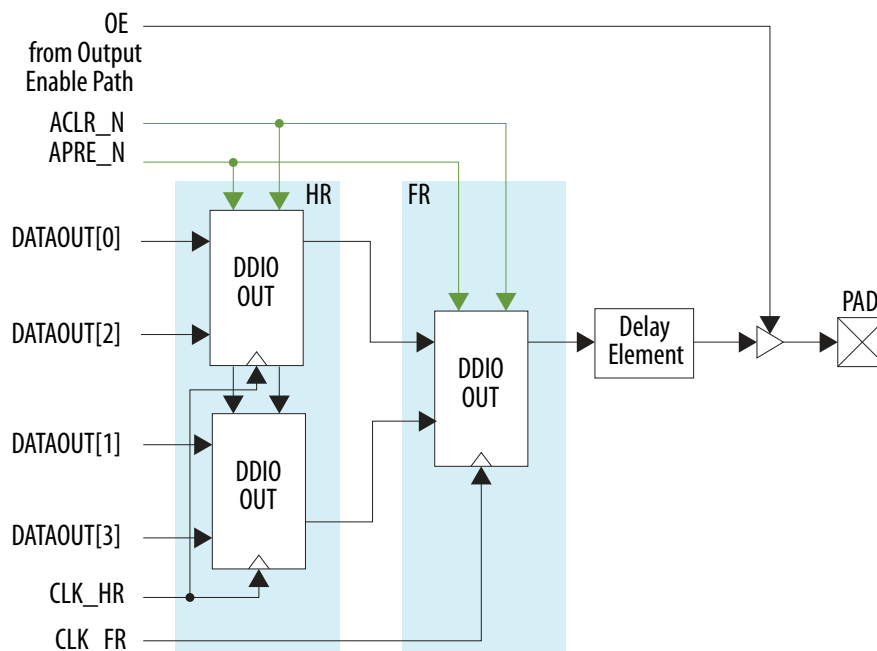
入力パスのダイナミック・キャリブレーションを必要とするアプリケーション向けに詳細情報を提供します。

### 出力および出力イネーブルパス

出力遅延エレメントは、出力バッファを介してパッドにデータを送信します。

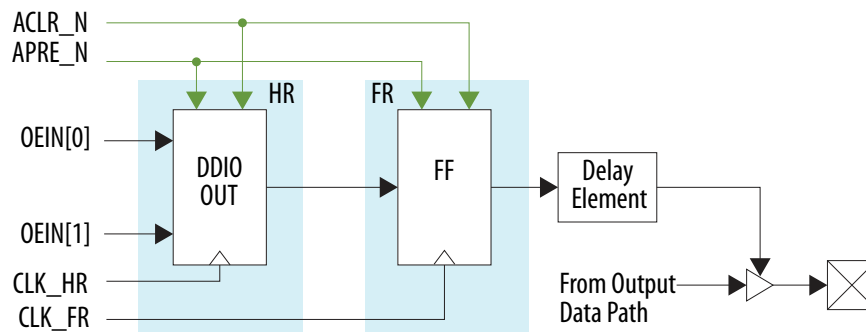
各出力パスには、フルレートとハーフレートの 2 つの DDIO ステージが含まれています。

図 -4: シングルエンド GPIO 出力パスの簡略図



出力パス波形は入力パス波形とは対称的です。

図 -5: 出カイナーブルパスの簡略図



出力パスと出カイナーブル (OE) パスの違いは、OE パスにフルレートの DDIO が含まれていないことです。OE パスでパックドレジスターの実装をサポートするために、シンプルレジスターはフルレート DDIO として動作します。同じ理由で、1 つのハーフレート DDIO しか存在しません。

OE パスは、次の 3 つの基本モードで動作します。

- バイパス: コアはすべての DDIO をバイパスして、データを遅延エレメントに直接送信します。
- パックドレジスター: ハーフレート DDIO をバイパスします。
- ハーフレートでの SDR 出力: ハーフレート DDIO は、データをフルレートからハーフレートに変換します。

**注意:** アルテラ GPIO IP コアは、出力パスのダイナミック・キャリブレーションをサポートしていません。出力パスのダイナミック・キャリブレーションを必要とするアプリケーションの場合、関連情報を参照してください。

#### 関連情報

- [Altera PHYLite for Parallel Interfaces IP Core User Guide](#)  
出力パスのダイナミック・キャリブレーションを必要とするアプリケーション向けに詳細情報を提供します。
- 5 ページの [入力パス](#)

## アルテラ GPIO のインターフェイス信号

指定したパラメーター設定に応じて、アルテラ GPIO IP コアにさまざまなインターフェイス信号を使用することができます。

図 -6: アルテラ GPIO IP コアのインターフェイス



図 -7: アルテラ GPIO のインターフェイス信号

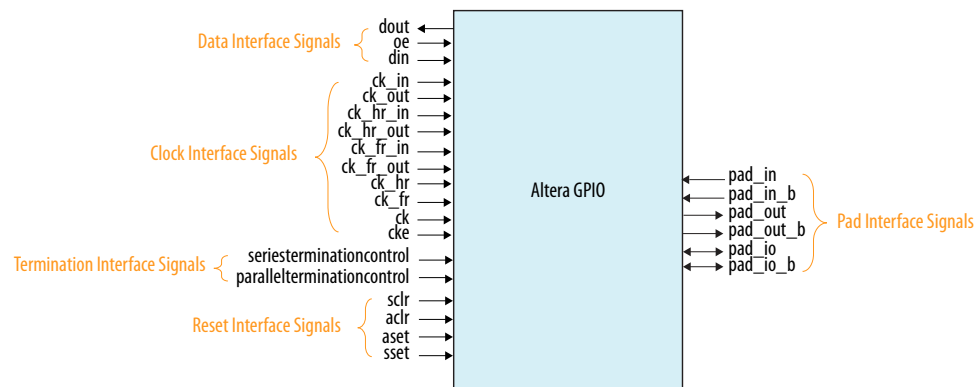


表 2. パッド・インターフェイス信号

パッド・インターフェイスは、アルテラ GPIO IP コアからパッドまでの物理的な接続です。このインターフェイスは、IP コアのコンフィグレーションに応じて、入力、出力、または双方向インターフェイスにすることができます。以下の表では、*SIZE* は IP コア・パラメーター・エディターで指定されたデータ幅です。

信号名	入力/出力	説明
pad_in[ <i>SIZE</i> -1:0]	入力	パッドからの入力信号です。
pad_in_b[ <i>SIZE</i> -1:0]	入力	パッドからの差動入力信号の負ノードです。このポートは、 <b>Use differential buffer</b> オプションをオンにした場合に使用できます。
pad_out[ <i>SIZE</i> -1:0]	出力	パッドへの出力信号です。
continued...		





信号名	入力/出力	説明
pad_out_b[SIZE-1:0]	出力	パッドへの差動出力信号の負ノードです。このポートは、 <b>Use differential buffer</b> オプションをオンにした場合に使用できます。
pad_io[SIZE-1:0]	双方向	パッドとの双方向信号接続です。
pad_io_b[SIZE-1:0]	双方向	パッドとの差動双方向信号接続の負ノードです。このポートは、 <b>Use differential buffer</b> オプションをオンにした場合に使用できます。

表 3. データ・インターフェイス信号

データ・インターフェイスは、アルテラ GPIO IP コアから FPGA コアまでの入力または出力インターフェイスです。以下の表では、SIZE は IP コア・パラメーター・エディターで指定されたデータ幅です。

信号名	入力/出力	説明
din[DATA_SIZE-1:0]	入力	出力または双方向モードで FPGA コアからのデータ入力です。DATA_SIZE は以下の Register Mode によって異なります。 <ul style="list-style-type: none"> <li>Bypass または Simple Register: DATA_SIZE = SIZE</li> <li>Half Rate logic なし DDIO: DATA_SIZE = 2 × SIZE</li> <li>Half Rate logic を使用した DDIO: DATA_SIZE = 4 × SIZE</li> </ul>
dout[DATA_SIZE-1:0]	出力	入力または双方向モードで FPGA コアに出力されるデータです。DATA_SIZE は以下の Register Mode によって異なります。 <ul style="list-style-type: none"> <li>Bypass または Simple Register: DATA_SIZE = SIZE</li> <li>Half-rate logic なし DDIO: DATA_SIZE = 2 × SIZE</li> <li>Half Rate logic を使用した DDIO: DATA_SIZE = 4 × SIZE</li> </ul>
oe[OE_SIZE-1:0]	入力	<b>Enable output enable port</b> をオンにした出力モードまたは双方向モードで FPGA コアからの OE 入力です。OE はアクティブ High です。データを送信するときは、この信号を 1 に設定します。データを受信するときは、この信号を 0 に設定します。OE_SIZE は以下の Register Mode によって異なります。 <ul style="list-style-type: none"> <li>Bypass または Simple Register: DATA_SIZE = SIZE</li> <li>Half Rate logic なし DDIO: DATA_SIZE = SIZE</li> <li>Half Rate logic を使用した DDIO: DATA_SIZE = 2 × SIZE</li> </ul>

表 4. クロック・インターフェイス信号

クロック・インターフェイスは入力クロック・インターフェイスです。クロック・インターフェイスはコンフィグレーションに応じて異なる信号で構成されます。アルテラ GPIO IP コアは 0、1、2、または 4 つのクロック入力を有することができます。クロックポートはそれぞれのコンフィグレーションで異なって表示され、クロック信号によって実行される実際の機能を反映します。

信号名	入力/出力	説明
ck	入力	入力バスと出力バスでは、 <b>Half Rate logic</b> パラメーターをオフにすると、このクロックがバックドレジスターまたは DDIO に供給されます。 双方向モードでは、 <b>Separate input/output Clocks</b> パラメーターをオフにすると、このクロックは入力バスと出力バスの一意のクロックです。
ck_fr	入力	入力バスと出力バスでは、 <b>Half Rate logic</b> パラメーターをオンにすると、これらのクロックがフルレートとハーフレートの DDIO に供給されます。 双方向モードでは、 <b>Separate input/output Clocks</b> パラメーターをオフにすると、入力バスと出力バスでこれらのクロックが使用されます。
ck_hr		
ck_in	入力	双方向モードでは、以下の両方の設定を指定すると、これらのクロックは入力バスと出力バスでバックドレジスターまたは DDIO に供給されます。 <ul style="list-style-type: none"> <li><b>Half Rate logic</b> パラメーターをオフにする</li> <li><b>Separate input/output Clocks</b> パラメーターをオンにする</li> </ul>
ck_out		

continued...

信号名	入力/出力	説明
ck_fr_in	入力	双方向モードでは、以下の両方の設定を指定すると、これらのクロックは入力バスと出力バスでフルレートおよびハーフレートの DDIO に供給されます。 <ul style="list-style-type: none"> <li>• <b>Half Rate logic</b> パラメーターをオンにする</li> <li>• <b>Separate input/output Clocks</b> パラメーターをオンにする</li> </ul> たとえば、ck_fr_out は出力バスでフルレート DDIO に供給されます。
ck_fr_out		
ck_hr_in		
ck_hr_out		
cke	入力	クロックイネーブルです。

表 5. 終端インターフェイス信号

終端インターフェイスは、アルテラ GPIO IP コアを I/O バッファに接続します。

信号名	入力/出力	説明
seriesterminationcontrol	入力	終端制御ブロック (OCT) からバッファへの入力です。これはバッファ直列インピーダンス値を設定します。
parallelerminationcontrol	入力	終端制御ブロック (OCT) からバッファへの入力です。これはバッファ並列インピーダンス値を設定します。

表 6. リセット・インターフェイス信号

リセット・インターフェイスは、アルテラ GPIO IP コアを DDIO に接続します。

信号名	入力/出力	説明
sclr	入力	同期クリア
acclr	入力	非同期クリア
aset	入力	非同期セット
sset	入力	同期セット

## 関連情報

11 ページの [入力バスと出力バスの上位ビットおよび下位ビット](#)

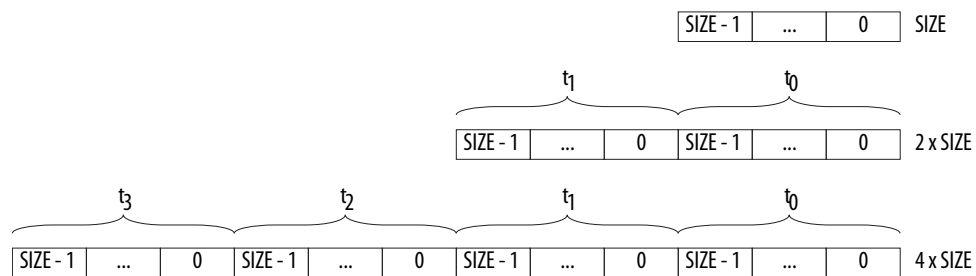
## 共有信号

- 入力、出力、および OE パスは同じクリアとプリセット信号を共有しています。
- 出力および OE パスは同じクロック信号を共有しています。

## データ・インターフェイスのためのデータのビット順

図 -8: データのビット順序の規則

以下の図は、din、dout、および oe データ信号のビット順序の規則を示しています。





- データ・バス・サイズの値が SIZE の場合、LSB は一番右の位置にあります。
- データ・バス・サイズの値が  $2 \times \text{SIZE}$  の場合、バスは SIZE の 2 ワードで構成されます。
- データ・バス・サイズの値が  $4 \times \text{SIZE}$  の場合、バスは SIZE の 4 ワードで構成されます。
- LSB は各ワードの右端の位置にあります。
- 右端のワードは、出力バス用の最初の送信ワードと入力バス用の最初の受信ワードを指定します。

#### 関連情報

5 ページの [入力バス](#)

## 入力バスと出力バスの上位ビットおよび下位ビット

入力信号または出力信号の上位ビットおよび下位ビットは、din と dout 入力と出力バスに含まれます。

#### 入力バス

din バスでは、datain\_h と datain\_l が上位ビットと下位ビットであり、各幅が datain\_width の場合、

- `datain_h = din[(2 × datain_width - 1):datain_width]`
- `datain_l = din[(datain_width - 1):0]`

たとえば、`din[7:0] = 8'b11001010` の場合、

- `datain_h = 4'b1100`
- `datain_l = 4'b1010`

#### 出力バス

dout バスでは、dataout\_h と dataout\_l が上位ビットと下位ビットであり、各幅が dataout\_width の場合、

- `dataout_h = dout[(2 × dataout_width - 1):dataout_width]`
- `dataout_l = dout[(dataout_width - 1):0]`

たとえば、`dout[7:0] = 8'b11001010` の場合、

- `dataout_h = 4'b1100`
- `dataout_l = 4'b1010`

## データ・インターフェイス信号と対応するクロック

表 7. データ・インターフェイス信号と対応するクロック

信号名	パラメーター・コンフィグレーション			クロック
	Register Mode	Half Rate	Separate Clocks	
din	<ul style="list-style-type: none"> <li>Simple Register</li> <li>DDIO</li> </ul>	Off	Off	ck
	DDIO	On	Off	ck_hr

*continued...*

信号名	パラメーター・コンフィグレーション			クロック
	Register Mode	Half Rate	Separate Clocks	
	<ul style="list-style-type: none"> <li>Simple Register</li> <li>DDIO</li> </ul>	Off	On	ck_in
	DDIO	On	On	ck_hr_in
<ul style="list-style-type: none"> <li>dout</li> <li>oe</li> </ul>	<ul style="list-style-type: none"> <li>Simple Register</li> <li>DDIO</li> </ul>	Off	Off	ck
	DDIO	On	Off	ck_hr
	<ul style="list-style-type: none"> <li>Simple Register</li> <li>DDIO</li> </ul>	Off	On	ck_out
	DDIO	On	On	ck_hr_out
<ul style="list-style-type: none"> <li>sclr</li> <li>sset</li> <li>すべてのパッド信号</li> </ul>	<ul style="list-style-type: none"> <li>Simple Register</li> <li>DDIO</li> </ul>	Off	Off	ck
	DDIO	On	Off	ck_fr
	<ul style="list-style-type: none"> <li>Simple Register</li> <li>DDIO</li> </ul>	Off	On	<ul style="list-style-type: none"> <li>入力バス: ck_in</li> <li>出力バス: ck_out</li> </ul>
	DDIO	On	On	<ul style="list-style-type: none"> <li>入力バス: ck_fr_in</li> <li>出力バス: ck_fr_out</li> </ul>

## リソース使用率とデザインのパフォーマンスの検証

Quartus Prime のコンパイルレポートを参照して、デザインのリソース使用量とパフォーマンスについての詳細を参照できます。

- メニューで **Processing > Start Compilation** をクリックしてフルコンパイルを実行します。
- デザインをコンパイルした後、**Processing > Compilation Report** をクリックします。
- Table of Contents** を使用して、**Fitter > Resource Section** に移動します。
  - リソース使用状況の情報を表示するには、**Resource Usage Summary** を選択します。
  - リソース使用率情報を表示するには、**Resource Utilization by Entity** を選択します。

## アルテラ GPIO のパラメーター設定

アルテラ GPIO IP コアのパラメーター設定は、Quartus Prime ソフトウェアで設定することができます。**General**、**Buffer**、および **Registers** のオプションの 3 つのグループがあります。



表 8. アルテラ GPIO の General パラメーター

パラメーター	条件	許容値	説明
Data Direction	—	<ul style="list-style-type: none"> <li>Input</li> <li>Output</li> <li>Bidir</li> </ul>	GPIO のデータ方向を指定します。
Data width	—	1 ~ 128	データ幅を指定します。
Use legacy top-level port names	—	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	<p>Stratix V、Arria V、および Cyclone V デバイスと同じポート名を使用します。</p> <p>たとえば、dout は dataout_h と dataout_l になり、din は datain_h と datain_l になります。</p> <p>注 これらのポートの動作は、Stratix V、Arria V、および Cyclone V デバイスとは異なります。移行のガイドラインについては、関連情報を参照してください。</p>

表 9. アルテラ GPIO の Buffer パラメーター

パラメーター	条件	許容値	説明
Use differential buffer	—	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	オンにした場合、差動 I/O バッファを有効にします。
Use pseudo differential Buffer	<ul style="list-style-type: none"> <li>Data Direction = Output</li> <li>Use differential buffer = On</li> </ul>	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	<p>出力モードでオンにすると、擬似差動出力バッファが有効にされます。</p> <p><b>Use differential buffer</b> をオンにすると、このオプションは双方向モードで自動的にオンになります。</p>
Use bus-hold circuitry	<ul style="list-style-type: none"> <li>Data Direction = Input または Bidir</li> <li>Use differential buffer = Off</li> </ul>	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	オンにすると、バスホールド回路が微弱な電流で I/O ピンの信号を最後に駆動された状態に保持し、これにより、出力バッファーステートはハイ・インピーダンスではなく 1 または 0 になります。
Use open drain output	<ul style="list-style-type: none"> <li>Data Direction = Output または Bidir</li> <li>Use differential buffer = Off</li> </ul>	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	オンにすると、オープンドレイン出力によってデバイスが割り込み信号や書き込みイネーブル信号といったシステムレベルのコントロール信号を提供できるようになり、これらの信号は、システム内の複数のデバイスがアサートすることができます。
Enable output enable port	Data Direction = Output	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	オンにすると、OE ポートへのユーザー入力を有効にします。このオプションは、双方向モードでは自動的にオンになっています。
Enable serietermination / paralleltermination ports	—	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>	オンにすると、出力バッファの serieterminationcontrol と parallelterminationcontrol ポートを有効にします。

表 10. アルテラ GPIO の Registers パラメーター

パラメーター	条件	許容値	説明
Register mode	—	<ul style="list-style-type: none"> <li>None</li> <li>Simple register</li> <li>DDIO</li> </ul>	アルテラ GPIO IP コアのレジスターモードを指定します。

continued...

パラメーター	条件	許容値	説明
			<ul style="list-style-type: none"> <li>• <b>None</b>: バッファ間の単純な配線接続を指定します。</li> <li>• <b>Simple register</b>: DDIO がシングル・データレート・モード (SDR) でシンプルレジスターとして使用されることを指定します。フィッターはこのレジスターを I/O にパッキングすることがあります。</li> <li>• <b>DDIO</b>: IP コアが DDIO を使用することを指定します。</li> </ul>
Enable synchronous clear / preset port	<ul style="list-style-type: none"> <li>• Register mode = DDIO</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Clear</li> <li>• Preset</li> </ul>	同期リセットポートの実装方法を指定します。 <ul style="list-style-type: none"> <li>• <b>None</b>: 同期リセットポートを無効にします。</li> <li>• <b>Clear</b>: 同期クリア用に SCLR ポートを有効にします。</li> <li>• <b>Preset</b>: 同期プリセット用に SSET ポートを有効にします。</li> </ul>
Enable asynchronous clear / preset port	<ul style="list-style-type: none"> <li>• Register mode = DDIO</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> <li>• Clear</li> <li>• Preset</li> </ul>	非同期リセットポートの実装方法を指定します。 <ul style="list-style-type: none"> <li>• <b>None</b>: 非同期リセットポートを無効にします。</li> <li>• <b>Clear</b>: 非同期クリア用に ACLR ポートを有効にします。</li> <li>• <b>Preset</b>: 非同期プリセット用に ASET ポートを有効にします。</li> </ul>
Enable clock enable ports	Register mode = DDIO	<ul style="list-style-type: none"> <li>• On</li> <li>• Off</li> </ul>	<ul style="list-style-type: none"> <li>• <b>On</b>: クロック・イネーブル・ポート (CKE) を公開し、データがクロックインまたはクロックアウトする際の制御を可能にします。この信号は、データがユーザーの制御なしで通過することを防ぎます。</li> <li>• <b>Off</b>: クロック・イネーブル・ポートが公開されず、データは常にレジスターを自動的に通過します。</li> </ul>
Half Rate logic	Register mode = DDIO	<ul style="list-style-type: none"> <li>• On</li> <li>• Off</li> </ul>	オンにすると、ハーフレートの DDIO が有効になります。
Separate input / output Clocks	<ul style="list-style-type: none"> <li>• Data Direction = Bidir</li> <li>• Register mode = Simple register または DDIO</li> </ul>	<ul style="list-style-type: none"> <li>• On</li> <li>• Off</li> </ul>	オンにすると、双方向モードの入力バスと出力バスに対して別々のクロック (CK_IN と CK_OUT) を有効にします。

#### 関連情報

- 11 ページの [入力バスと出力バスの上位ビットおよび下位ビット](#)
- 23 ページの [ガイドライン: 移行された IP における datain\\_h と datain\\_l ポートのスワップ](#)

## レジスターパッキング

アルテラ GPIO IP コアを使用すると、エリアとリソース使用率を節約するためにペリフェラルにレジスターをパックすることができます。

入力バスおよび出力バスにフルレート DDIO をフリップフロップとして設定することができます。これを行うには、次の表にリストされている .qsf アサインメントを追加します。



表 11. レジスタパッキング QSF アサインメント

バス	QSF アサインメント
入力レジスタパッキング	set_instance_assignment -name FAST_INPUT_REGISTER ON -to <path to register>
出力レジスタパッキング	set_instance_assignment -name FAST_OUTPUT_REGISTER ON -to <path to register>
出カインエーブル・レジスタパッキング	set_instance_assignment -name FAST_OUTPUT_ENABLE_REGISTER ON -to <path to register>

注意: これらのアサインメントは、レジスタパッキングを保証するのではなく、法的な配置を見つけるために、フィッターを有効にします。そうでない場合、フィッターは、フリップフロップをコアに保持します。

## アルテラ GPIO のタイミング

アルテラ GPIO IP コアのパフォーマンスは、I/O 制約とクロックフェーズに依存します。アルテラ GPIO コンフィグレーションのタイミングを検証するには、インテルは TimeQuest Timing Analyzer を使用することを推奨します。

### 関連情報

[The Quartus Prime TimeQuest Timing Analyzer](#)

## タイミング・コンポーネント

アルテラ GPIO IP コアのタイミング・コンポーネントは、以下の 3 つのパスで構成されています。

- I/O インターフェイス・パス: FPGA から外部の受信デバイスへ、また外部の送信デバイスから FPGA へのパスです。
- データとクロックのコア・インターフェイス・パス: I/O からコアへ、またコアから I/O へのパスです。
- 転送パス: ハーフレート DDIO からフルレート DDIO へ、またフルレート DDIO からハーフレート DDIO へのパスです。

注意: TimeQuest Timing Analyzer は、DDIO\_IN ブロックと DDIO\_OUT ブロック内のパスをブラックボックスとして扱います。

図 -9: 入力パスのタイミング・コンポーネント

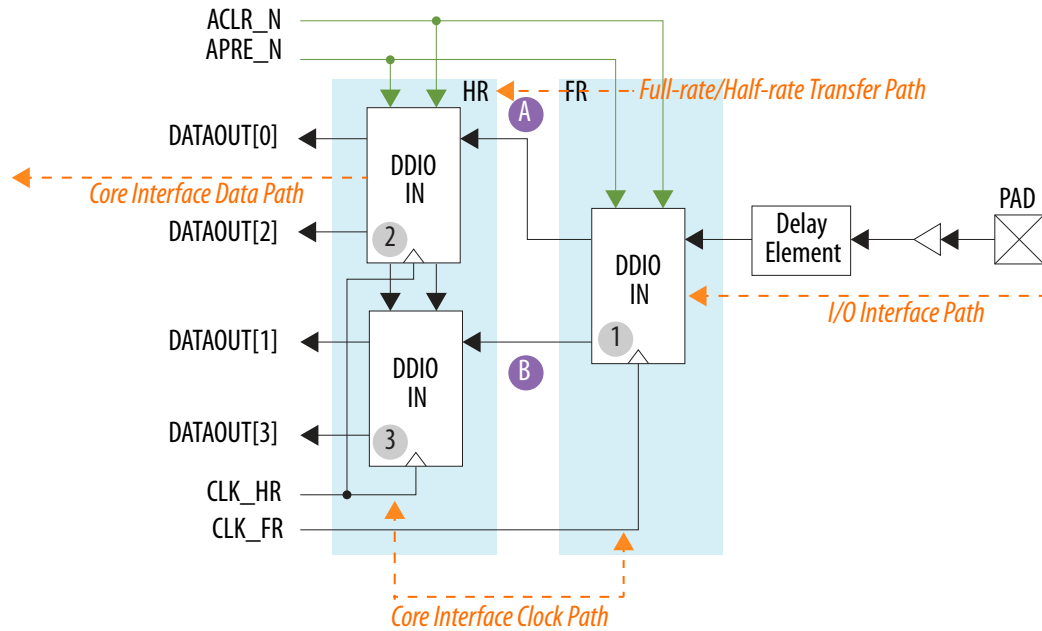


図 -10: 出力パスのタイミング・コンポーネント

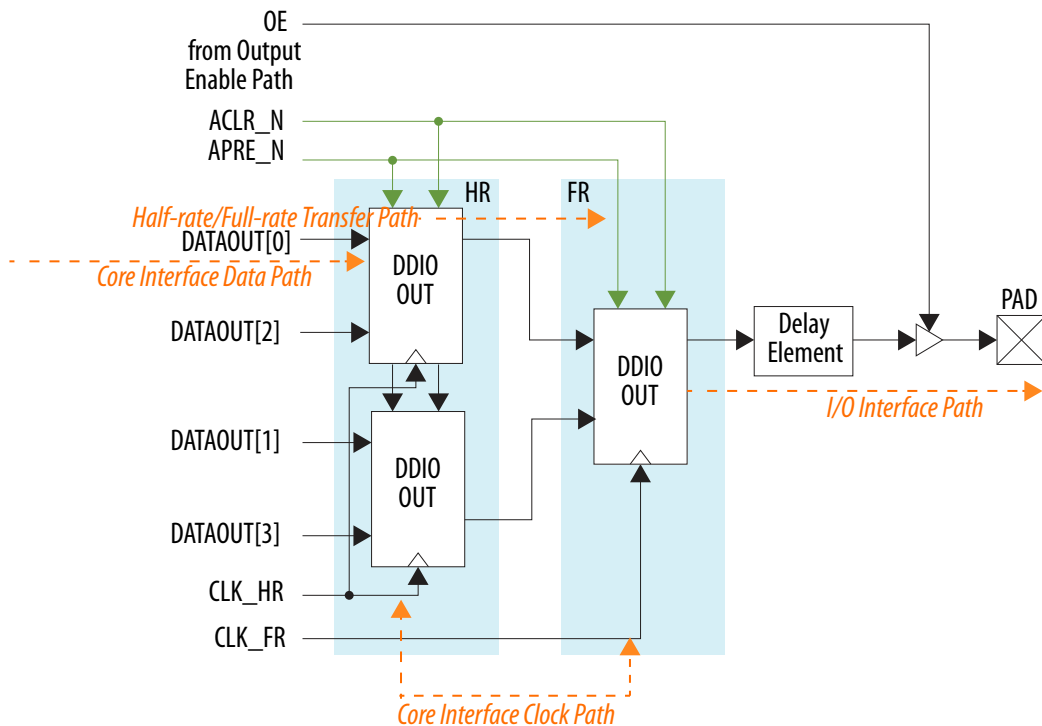
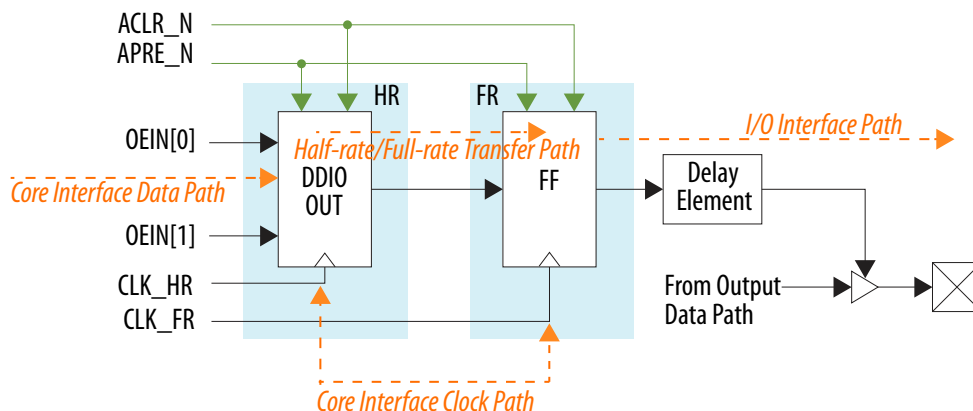




図 -11: 出カインーブルパスのタイミング・コンポーネント



## 遅延エレメント

Quartus Prime ソフトウェアは、I/O タイミング解析のスラックを最大にするために遅延エレメントを自動的に設定しません。タイミングを閉じる、またはスラックを最大にするには、遅延エレメントを Quartus Prime 設定ファイル (.qsf) でマニュアルで設定する必要があります。

表 12. 遅延エレメントの .qsf アサインメント

遅延エレメントにアクセスするには、以下のアサインメントを .qsf で指定します。

遅延エレメント	.qsf アサインメント
入力遅延エレメント	<code>set_intance_assignment -to &lt;PIN&gt; -name INPUT_DELAY_CHAIN &lt;0..63&gt;</code>
出力遅延エレメント	<code>set_intance_assignment -to &lt;PIN&gt; -name OUTPUT_DELAY_CHAIN &lt;0..15&gt;</code>
出カインーブル遅延エレメント	<code>set_intance_assignment -to &lt;PIN&gt; -name OE_DELAY_CHAIN &lt;0..15&gt;</code>

## タイミング解析

Quartus Prime ソフトウェアは、アルテラ GPIO IP コアの SDC タイミング制約を自動的に生成しません。タイミング制約をマニュアルで入力する必要があります。

タイミング・ガイドラインと例を参照して、TimeQuest Timing Analyzer が I/O タイミングを正しく解析するようにしてください。

- 適切に I/O インターフェイス・パスのタイミング解析を実行するには、.sdc ファイルでシステム・クロックピンに対するデータピンのシステムレベルの制約を指定します。
- 適切にコア・インターフェイス・パスのタイミング解析を実行するには、以下のクロック設定を .sdc ファイルで定義します。
  - コアレジスターへのクロック
  - Simple Register モードと DDIO モード用の I/O レジスターへのクロック

## 関連情報

### AN 433: Constraining and Analyzing Source-Synchronous Interfaces

このアプリケーション・ノートでは、ソース同期インタフェースを制約および解析するための手法について説明します。

## シングル・データレートの入力レジスター

図 -12: シングル・データレートの入力レジスター

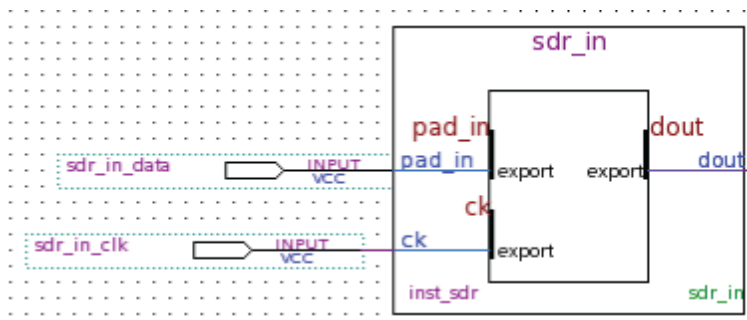


表 13. シングル・データレート入力レジスター.sdc コマンドの例

コマンド	コマンドの例	説明
create_clock	create_clock -name sdr_in_clk -period "100 MHz" sdr_in_clk	入力クロックのクロック設定を作成します。
set_input_delay	set_input_delay -clock sdr_in_clk 0.15 sdr_in_data	TimeQuest Timing Analyzer に、0.15 ns の入力遅延で入力 I/O のタイミングを解析するように指示します。

## フルレートまたはハーフレート DDIO 入力レジスター

フルレートとハーフレート DDIO 入力レジスターの入力側は同じです。仮想クロックを使用して FPGA へのオフチップ・トランスミッタをモデル化することにより、システムを適切に制約することができます。

図 -13: フルレートまたはハーフレート DDIO 入力レジスター

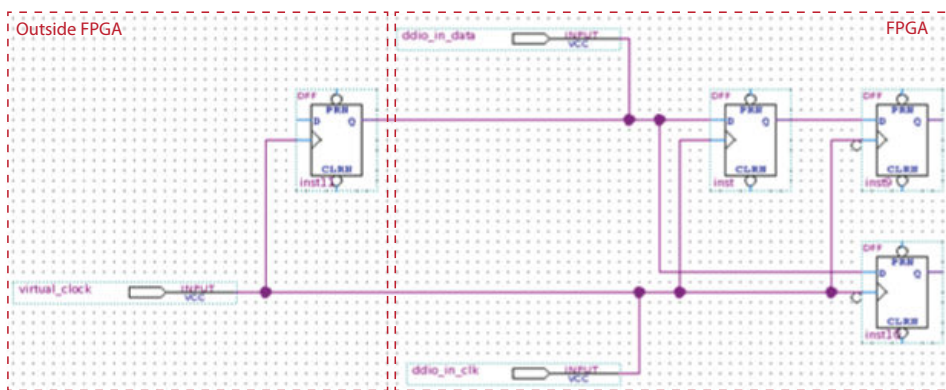


表 14. フルレートまたはハーフレート DDIO 入力レジスター .sdc コマンドの例

コマンド	コマンドの例	説明
create_clock	create_clock -name virtual_clock -period "200 MHz" create_clock -name ddio_in_clk -period "200 MHz" ddio_in_clk	仮想クロックと DDIO クロックのクロック設定を作成します。
set_input_delay	set_input_delay -clock virtual_clock 0.25 ddio_in_data set_input_delay -add_delay -clock_fall -clock virtual_clock 0.25 ddio_in_data	TimeQuest Timing Analyzer に、転送の正のクロックエッジと負のクロックエッジを解析するように指示します。2 番目の set_input_delay コマンドの -add_delay に注意してください。
set_false_path	set_false_path -fall_from virtual_clock -rise_to ddio_in_clk set_false_path -rise_from virtual_clock -fall_to ddio_in_clk	TimeQuest Timing Analyzer に、負のエッジでトリガーされるレジスターへの正のクロックエッジ、および正のエッジでトリガーされるレジスターへの負のクロックエッジを無視するように指示します。 注 CLK_HR 周波数は、CLK_FR 周波数の半分でなければなりません。I/O PLL がクロックを駆動する場合は、derive_pll_clocks .sdc コマンドを使用することが可能です。

## シングル・データレートの出力レジスター

図 -14: シングル・データレートの出力レジスター

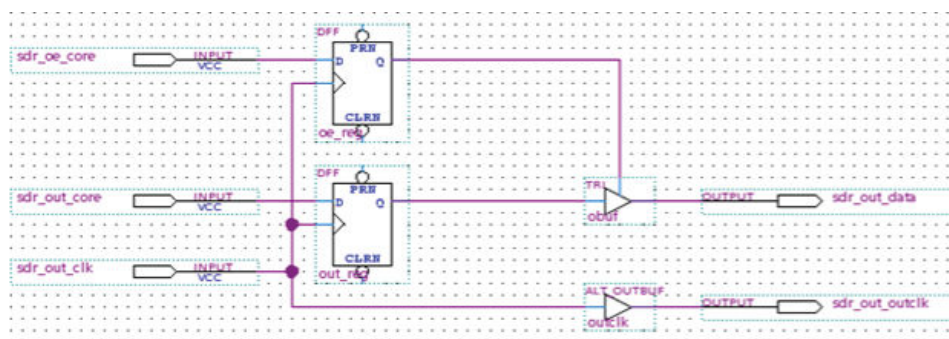


表 15. シングル・データレート出力レジスター .sdc コマンドの例

コマンド	コマンドの例	説明
create_clock と create_generated_clock	create_clock -name sdr_out_clk -period "100 MHz" sdr_out_clk create_generated_clock -source sdr_out_clk -name sdr_out_outclk sdr_out_outclk	送信されるソースクロックと出力クロックを生成します。
set_output_delay	set_output_delay -clock sdr_out_clk 0.45 sdr_out_data	TimeQuest Timing Analyzer に、送信される出力クロックに対して送信される出力データを解析するように指示します。

## フルレートまたはハーフレート DDIO 出力レジスター

フルレートとハーフレート DDIO 出力レジスターの出力側は同じです。

表 16. DDIO 出力レジスター .sdc コマンドの例

コマンド	コマンドの例	説明
create_clock と create_generated_ clock	create_clock -name ddio_out_fr_clk -period "200 MHz" ddio_out_fr_clk create_generated_clock -source ddio_out_fr_clk -name ddio_out_fr_outclk ddio_out_fr_outclk	DDIO へのクロックと送信するクロックを生成します。
set_output_delay	set_output_delay -clock ddio_out_fr_outclk 0.55 ddio_out_fr_data set_output_delay -add_delay -clock_fall -clock ddio_out_fr_outclk 0.55 ddio_out_fr_data	TimeQuest Timing Analyzer に、出力クロックに対し て正のデータと負のデータを解析するように指示します。
set_false_path	set_false_path -rise_from ddio_out_fr_clk -fall_to ddio_out_fr_outclk set_false_path -fall_from ddio_out_fr_clk -rise_to ddio_out_fr_outclk	TimeQuest Timing Analyzer に、出力クロックの立ち 下がりエッジに対するソースクロックの立ち上がりエッ ジ、および出力クロックの立ち上がりエッジに対するソー スクロックの立ち下がりエッジを無視するように指示しま す。

## タイミング・クロージャのガイドライン

アルテラ GPIO 入力レジスターでは、入力遅延チェーンを設定しないと、入力 I/O 転送はホールド時間を満たさなくなる可能性があります。この障害は、クロック遅延がデータ遅延よりも大きいために発生します。

ホールド時間を満たすには、入力遅延チェーンを使用して入力データパスに遅延を追加します。一般に、入力遅延チェーンは、-1 のスピードグレードでステップごとに約 60 ps です。タイミングを満たすためにおおよその入力遅延チェーン設定を取得するには、負のホールドスラックを 60 ps で割ります。

ただし、I/O PLL が GPIO 入力レジスター (Simple Register または DDIO モード) のクロックを駆動する場合は、補償モードをソース同期モードに設定できます。フィッターは、入力 I/O タイミング解析のより良いセットアップとホールドスラックのために I/O PLL を構成しようとします。

アルテラ GPIO 出力レジスターと出力イネーブルレジスターでは、出力遅延チェーンと出力イネーブル遅延チェーンを使用して出力データおよびクロックに遅延を追加できます。

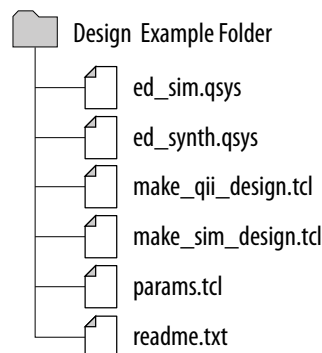
- セットアップ時間違反が発生した場合は、出力クロック遅延チェーンの設定を増加することができます。
- ホールド時間違反が発生した場合は、出力データ遅延チェーンの設定を増加することができます。

## アルテラ GPIO IP コアのデザイン例

アルテラ GPIO IP コアは、パラメーター・エディターで IP コンフィグレーションに一致するデザイン例を生成できます。これらのデザイン例は、IP コアのインスタンス化、およびシミュレーションで予期される動作のためのリファレンスとして使用できます。

デザイン例は、アルテラ GPIO IP コア・パラメーター・エディターから生成できます。必要なパラメーターを設定した後、**Generate Example Design** をクリックします。IP コアは、指定したディレクトリにデザイン例のソースファイルを生成します。

図 -15: 生成されたデザイン例のディレクトリーにおけるソースファイル



注意: .qsys ファイルは、デザイン例の生成中にのみ内部的に使用されます。これらの .qsys ファイルは編集できません。

## アルテラ GPIO 合成可能な Quartus Prime デザイン例

合成可能なデザイン例は、Quartus Prime プロジェクトに含めることができるコンパイル可能な Qsys システムです。

### デザイン例の生成と使用

合成可能な Quartus Prime デザイン例をソースファイルから生成するには、デザイン例のディレクトリーで以下のコマンドを実行します。

```
quartus_sh -t make_qii_design.tcl
```

使用するデバイスを正確に指定するには、次のコマンドを実行します。

```
quartus_sh -t make_qii_design.tcl [device_name]
```

TCL スクリプトは、ed\_synth.qpf プロジェクト・ファイルを含む qii ディレクトリーを作成します。Quartus Prime ソフトウェアでこのプロジェクトを開いてコンパイルすることができます。

## アルテラ GPIO IP コア・シミュレーションのデザイン例

シミュレーション・デザイン例では、アルテラ GPIO IP コア・パラメーター設定を使用して、シミュレーション・ドライバーに接続された IP インスタンスを構築します。ドライバーはランダムなトラフィックを生成し、出力データの正当性を内部的にチェックします。

デザイン例を使用して、使用するシミュレーターに応じて、単一のコマンドを使用してシミュレーションを実行できます。このシミュレーションは、アルテラ GPIO IP コアの使用法を示しています。

### デザイン例の生成と使用

Verilog シミュレーター向けにソースファイルからシミュレーションのデザイン例を生成するには、デザイン例のディレクトリーで以下のコマンドを実行します。

```
quartus_sh -t make_sim_design.tcl
```

VHDL シミュレーター向けにソースファイルからシミュレーションのデザイン例を生成するには、デザイン例のディレクトリで以下のコマンドを実行します。

```
quartus_sh -t make_sim_design.tcl VHDL
```

TCL スクリプトは、サポートされているシミュレーション・ツールごとにサブディレクトリを含む sim ディレクトリを作成します。各シミュレーション・ツールのスクリプトは、対応するディレクトリにあります。

## Arria V、Cyclone V、および Stratix V デバイスの IP マイグレーション・フロー

IP マイグレーション・フローを使用すると、Arria V、Cyclone V、および Stratix V デバイスの ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、および ALTIOBUF IP コアを Arria 10 デバイスのアルテラ GPIO IP コアに移行できます。

この IP マイグレーション・フローは、アルテラ GPIO IP コアを ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、および ALTIOBUF IP コアの設定と一致するようにコンフィグレーションし、IP コアの再生成を可能にします。

**注意:** 一部の IP コアは、特定のモードでのみ IP マイグレーション・フローをサポートします。IP コアがサポートしないモードにある場合は、アルテラ GPIO IP コア用の IP Parameter Editor を実行し、IP コアをマニュアルでコンフィグレーションする必要があります。

## ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、および ALTIOBUF IP コアの移行

ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、および ALTIOBUF IP コアをアルテラ GPIO IP コアに移行するには、次の手順を実行してください。

1. IP Parameter Editor で ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、または ALTIOBUF IP コアを開きます。
2. **Currently selected device family** リストから **Arria 10** を選択します。
3. **Finish** をクリックして、アルテラ GPIO IP Parameter Editor を開きます。  
IP Parameter Editor は、アルテラ GPIO IP コア設定を ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、または ALTIOBUF IP コア設定と同様にコンフィグレーションします。
4. 2 つの IP コア間で互換性のない設定がある場合は、**new supported settings** を選択します。
5. **Finish** をクリックして IP コアを再生成します。
6. RTL での ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、または ALTIOBUF IP コア・インスタンス化を、アルテラ GPIO IP コアで置き換えます。

**注意:** アルテラ GPIO IP コアのポート名は、ALTDDIO\_IN、ALTDDIO\_OUT、ALTDDIO\_BIDIR、または ALTIOBUF IP コアのポート名と一致しない場合があります。したがって、インスタンス化で IP コア名を変更するだけでは不十分な場合があります。

### 関連情報

11 ページの [入力バスと出力バスの上位ビットおよび下位ビット](#)



## ガイドライン: 移行された IP における datain\_h と datain\_l ポートのスワップ

GPIO IP をアルテラ GPIO IP コアに移行する際は、アルテラ GPIO IP コア・パラメーター・エディターで **Use legacy top-level port names** オプションをオンにすることができます。ただし、アルテラ GPIO IP コアでのこれらのポートの動作は、Stratix V、Arria V、および Cyclone V デバイスで使用する IP コアでの動作とは異なります。

アルテラ GPIO IP コアは、以下のポートを次のクロックエッジでの出力レジスターに駆動します。

- datain\_h:outclock の立ち下がりエッジで
- datain\_l:outclock の立ち上がりエッジで

Stratix V、Arria V、および Cyclone V デバイスから GPIO IP を移行した場合は、アルテラ GPIO IP コアで生成された IP をインスタンス化するときに datain\_h および datain\_l ポートをスワップしてください。

### 関連情報

11 ページの [入力バスと出力バスの上位ビットおよび下位ビット](#)

## アルテラ GPIO IP コアของผู้ใช้ 가이드のアーカイブ

IP コアのバージョンが記載されていない場合には、以前の IP コアバージョン向けのユーザーガイドが当てはまります。

IP コアバージョン	ユーザーガイド
16.1	<a href="#">Altera GPIO IP Core User Guide</a>
16.0	<a href="#">Altera GPIO IP Core User Guide</a>
14.1	<a href="#">Altera GPIO Megafunction User Guide</a>
13.1	<a href="#">Altera GPIO Megafunction User Guide</a>

## 改訂履歴

日付	バージョン	変更内容
2017 年 5 月	2017.05.08	<ul style="list-style-type: none"><li>• <b>Use bus-hold circuitry</b> パラメーター・オプションの条件を指定してアルテラ GPIO の Buffer パラメーターの表を更新</li><li>• ブランド名を「インテル」に変更</li></ul>
2016 年 10 月	2016.10.31	<ul style="list-style-type: none"><li>• 入力バスの波形を更新</li><li>• din バスと dout バスの上位ビットと下位ビットについて説明するトピックを追加</li></ul>
continued...		



日付	バージョン	変更内容
2016 年 8 月	2016.08.05	<ul style="list-style-type: none"><li>• アルテラ GPIO IP コアにダイナミック OCT サポートに関するノートを追加</li><li>• 正確性と明確性を向上させるためにパラメーター設定についてのトピックを更新</li><li>• デザイン例の生成に関するセクションを更新</li><li>• Stratix V、Arria V、および Cyclone V デバイスからアルテラ GPIO IP コアに移行する際のレガシーポートの動作に関するガイドライン・トピックを追加</li><li>• 明確性と参照の容易さを向上させるためにドキュメントを書き直して再構築</li><li>• 表記を <i>Quartus II</i> から <i>Quartus Prime</i> へ変更</li></ul>
2014 年 8 月	2014.08.18	<ul style="list-style-type: none"><li>• タイミング情報を追加</li><li>• レジスターバックアップの情報を追加</li><li>• <b>Use legacy top-level port names</b> という新しいパラメーターを追加</li><li>• レジスターバックアップの情報を追加</li><li>• メガファンクションという用語を IP コアに置換</li></ul>
2013 年 11 月	2013.11.29	初版