

LatticeECP2M SERDES/PCS ユーザガイド

PCSの概要

LatticeECP2M™ FPGAファミリは業界をリードするアーキテクチャを持ち、高性能FPGAファブリックと、高性能I/O、および大規模組み込みメモリを単一チップに集積しています。また、すべてのLatticeECP2Mデバイスが特徴としているのは、組み込みSERDES（最大16チャンネル）とこれに付随するPhysical Coding Sublayer(PCS)ロジックです。多数の業界標準高速データ転送プロトコルをサポートするためにPCSロジックを構成することができます。

各チャンネルに専用のPCSロジックは送受信SERDESを含み、データレート3.125Gbpsの高速全二重シリアルデータ転送に対応します。PCSロジックはイーサネット(1GbEとSGMII)、PCI Express、CPRI、およびOBSAIを含む一連のポピュラーなデータプロトコルをサポートするために各チャンネルを構成することができます。さらに、ユーザの高速データインターフェイス設計に柔軟に対応できるように、多くの構成でプロトコルベースのロジックを完全か部分的にバイパスすることができます。

またPCSはバイパスモードを提供し、SERDESからFPGAロジックとは8ビット、または10ビットの直接インターフェイスがあります。また、それぞれのSERDESピンは個別にDC結合に設定することで、高速動作と、シリアル・デジタルビデオ(SDI)アプリケーションのような低速動作を、同じSERDESピンで対応することが可能です。

機能

- ・ 最大16チャンネルの高速SERDES
 - チャンネルあたり250Mbpsから3.125Gbps
 - 3.125Gbps動作時にチャンネルあたり100mWの低消費電力
 - フォームファクタの小さいバックプレーン動作の為の受信等化と送信プリエンファシス
 - PCI Express、イーサネット(1GbEとSGMII)、および他の複数の規格をサポート
 - ユーザ指定の汎用(Generic) 8b10bモードをサポート
 - PCI Expressのビーコンをサポート
 - 低速入力用のOOB (out-of-band)信号インターフェイス(ビデオ用途)
- ・ 複数のクロックレートをサポート
 - 各PCSクワッド用に個別の基準クロックが使用でき、単一デバイスで複数のプロトコルレートを容易に扱うことが可能
- ・ フル機能の組み込み物理コーディング・サブレイヤ(PCS)ロジックで業界標準プロトコルをサポート
 - デバイス当たり最大16チャンネルの全二重データ通信をサポート
 - 単一チップで複数プロトコルをサポート
 - 8b10bベースの広く採用されているパケットプロトコルをサポート
 - SERDES OnlyモードはFPGAロジックと8ビット、または10ビットで直接インターフェイスが可能
- ・ ギガビット・イーサネットのサポート
 - IEEE 1000BASE-Xに準拠
 - 8b10bエンコーディング/デコーディング
 - 自動ネゴシエーションをサポートするための受信データストリーム中への 1/2 シンボル挿入
 - コンマキャラクタのワードアライメント
 - クロックトレランス補償 (CTC) 回路

- ・ PCI Expressサポート
 - 単一PCSクワッドでx1またはx4をサポート
 - ワードアライナを集積
 - 8b10bエンコーディング/デコーディング
 - クロックトレランス補償(CTC)回路
 - 電氣的アイドルとレシーバ検出のサポート
 - ビーコン送信とビーコン検出のサポート
- ・ 複数プロトコルに準拠するクロックトレランス補償(CTC)ロジック
 - 基準クロックと受信データレートとの周波数差を補償
 - 長さ1/2/4バイトのユーザ定義スキップパターンの使用が可能
- ・ システムデバッグのためのループバック・モードを集積
 - システムデバッグのために3つのループバック・モードを提供

サポート規格

サポートする規格を表8-1にリストアップします。

表8-1 サポートするSERDES規格 (フルサポート)

規格	レート [Mbps]	REFCLK [MHz]	FPGA CLK [MHz]	符号化	信号タイプ
PCI Express	2500	100	250	8b10b	CML
GbE/SGMII	1250	125	125	8b10b	CML
Generic 8b10b	250 ~ 3125	25.0 ~ 312.5	25.0 ~ 312.5	8b10b / なし	CML
10-bit SERDES Only ¹	250 ~ 3125	25.0 ~ 312.5	25.0 ~ 312.5	なし	CML
8-bit SERDES Only ¹	250 ~ 3125	25.0 ~ 312.5	25.0 ~ 312.5	なし	CML
SD-SDI ²	143, 177, 270, 360	14.3, 17.7, 27.0, 36.0	143, 177, 135, 180	SMPTEスクランブル	CML
HD-SDI	1483.5, 1485	148.35, 148.5	148.35, 148.5	SMPTEスクランブル	CML
CPRI	614.4 1228.8	61.44 122.88	61.44 122.88	8b10b	CML

1. SERDES Onlyモードは8ビットと10ビット共に、リンクアライン/コンマアラインと、8b10bエンコーダ/デコーダ、およびCTCをバイパスします。CDRはバイパスしません。

2. 標準精細度(SD)用のシリアルデジタル・インターフェイス(SDI): 143Mbps、177MbpsはSERDES/PCSブロックをバイパスします。クロックとデータはRXピンからBSCANパスを通してFPGAコアに入ります。CDRはFPGAコアで行われます。送信方向では、これらの低いビットレートにはデシメーションが用いられます。FPGA内でCDRを実行するためには、14.3MHzと17.7MHzの基準クロックが必要です。270Mbpsは最も一般的な周波数です。これは10ビットのデータバスを通ります。

上記SERDESモードでXAUI、SRIO、OBSAI、CPRI、1XFC、2XFC、PICMG3.1、PICMG3.4、PICMG3.5、および3G-SDIの各規格をサポートすることができます。より詳細な情報が必要な場合はラティスセミコンダクターのテクニカルサポート・グループにご連絡ください。

アーキテクチャ概要

PCSロジックは独立する4系統の全二重データチャンネル用のロジックを含み、クワッドごとにアレンジされています。表8-2はLatticeECP2Mファミリで各デバイスにあるSERDES/PCSクワッドを示します。

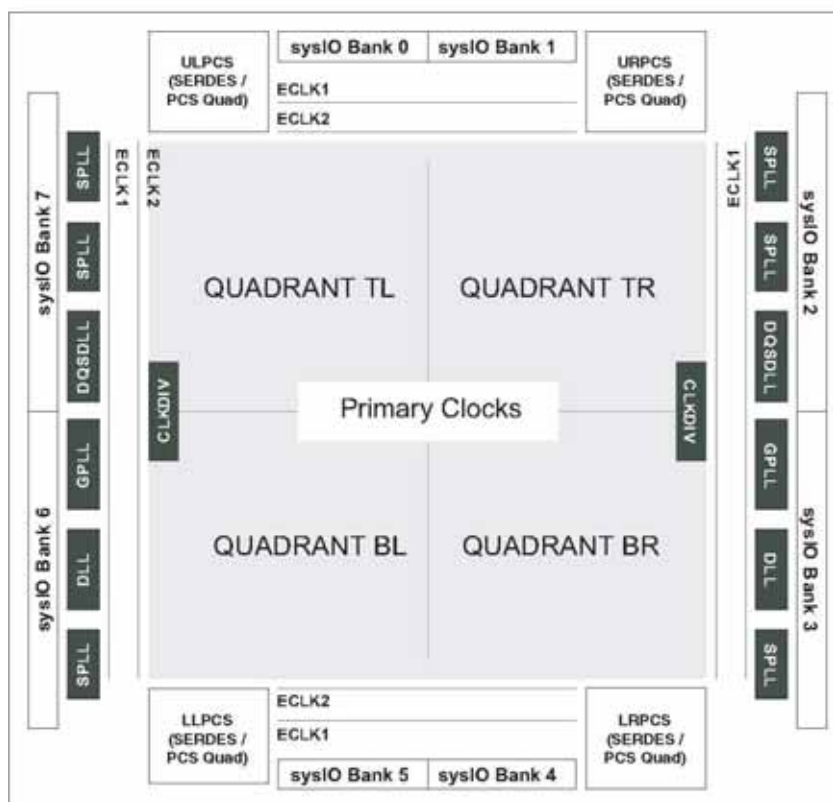
表8-2 LatticeECP2MファミリのデバイスごとのSERDES/PCSクワッド

デバイス	ECP2M20	ECP2M35	ECP2M50	ECP2M70	ECP2M100
Quad URC (右上)	Yes	Yes	Yes	Yes	Yes
Quad LRC (右下)	-	-	Yes	Yes	Yes
Quad ULC (左上)	-	-	-	Yes	Yes
Quad LLC (左下)	-	-	-	Yes	Yes

PCSクワッド

図8-1はデバイス内でPCSクワッドの配置を示すLatticeECP2Mのレイアウトです。(4つのクワッドを含む最も大きいアレイが示されています。他のデバイスではクワッド数は少なくなります。)

図8-1 LatticeECP2M70/100 ブロック図



全てのクワッドは複数のプロトコルベースのモードの1つにプログラムすることができます。各クワッドはそれ自身の基準クロックを必要とし、外部からパッケージピンを介して、或いは内部FPGAロジックから供給することができます。

各クワッドにはそれぞれ基準クロックがあるので、クワッドが異なれば同一チップでも違う規格をサポートすることができます。この機能はLatticeECP2Mファミリを、異なる規格間のブリッジ機能の実装に理想的な

デバイスにします。

PCSクワッドは単に業界標準プロトコル専用なだけではありません。多くのユーザ定義のデータ操作モードのために、各クワッド(そしてクワッド内の各チャンネル)をプログラムすることができます。例えば、ユーザ定義のワードアライメントとクロックトレランス補償を司るモードは、非プロトコル動作としてプログラムすることができます。

PCSクワッドとチャンネル

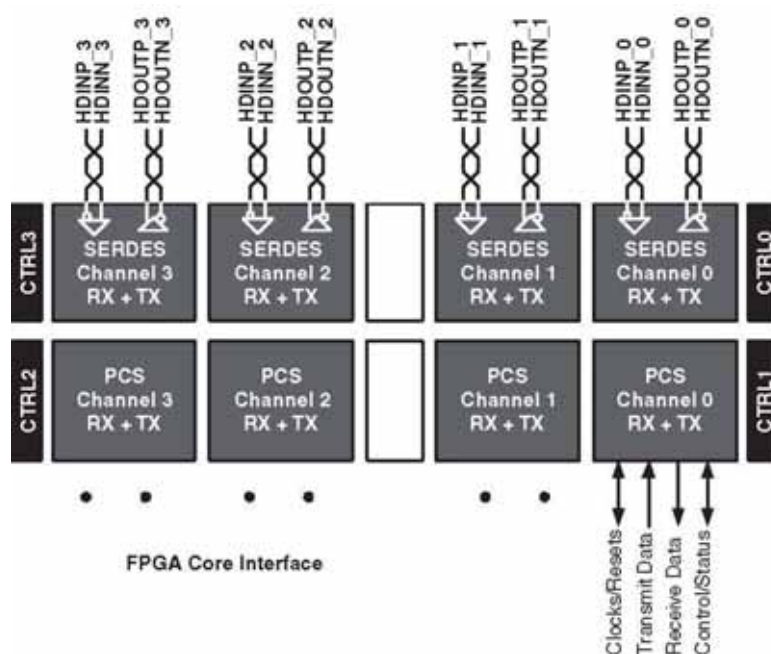
デバイスの各クワッドは最大4チャンネルの全二重データをサポートします。アプリケーションによって、ユーザは単一クワッドで1~4チャンネルのいずれも利用することができます。どのクワッドであれ、ユーザが各チャンネルごとに独立して設定できる多くのオプションがあります。

また図8-1には4つのPCSクワッド、合計16のPCSチャンネルを含むデバイスの例を示します。それぞれのクワッドはLatticeECP2Mアレイ内を4分割した位置に従って命名されています。すなわちURPCS(右上のPCS)、ULPCS(左上のPCS)、LRPCS(右下のPCS)、LLPCS(左下のPCS)です。

チャンネル単位のPCS/FPGAインターフェイス・ポート

選ばれたモードにかかわらず、すべてのPCSクワッドはパッケージピンに共通の外部高速シリアル・インターフェースを持っています。しかしながら、各クワッドとFPGAロジック間のI/Oポートには、それぞれのPCSモードで選択されたプロトコルに適切となる独自のリストがあります。本ドキュメントでは各モード用のクワッド入出力信号の詳細を記述します。単一クワッド内のチャンネルを示す概略図を図8-2に示します。

図8-2 PCSクワッドのブロック図



PCSクワッドの配置

LatticeECP2M-50以上の規模のデバイスは2~4つのPCSクワッドを含んでいます。

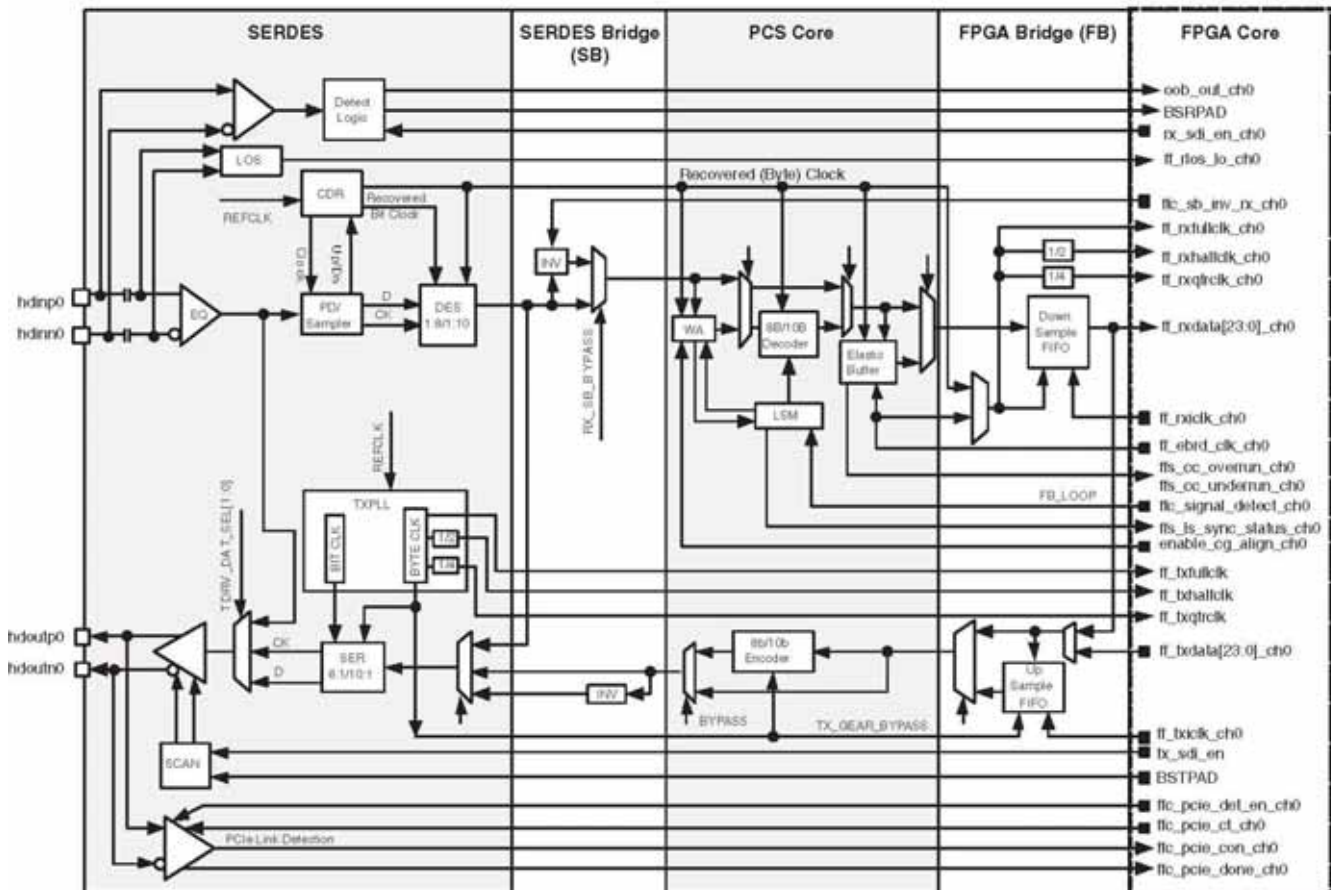
ユーザは、プリファレンス(設計制約)ファイルの中でLOCATEプリファレンスを用いることで、各PCSクワッド配置を所望の位置に指定することができます。LOCATE制約については、以下に例を示します。

LOCATE COMP "PCS_instantiation_1" SITE "URPCS"; (訳注; PCS_instantiation_1はコンポーネント名)

詳細なチャネルブロック図

図8-3は、LatticeECP2M SERDES/PCS単一チャネルの主要な機能の詳細なブロック図を表します。この図にはFPGAでユーザロジックから見えるすべての主要なブロック、および制御とステータス信号の大部分が示されています。また、この図はチャネルSERDES、SERDESブリッジ、PCSコア、およびFPGAブリッジの主要なサブブロックも示しています。

図8-3 LatticeECP2M SERDES/PCSチャネルの詳細ブロック図



以下にFPGAインターフェイス信号の概要説明が続きます。

クロックとリセット

PCSクワッドはFPGAロジック・インターフェイスに対して、(Tx PLLに)ロックした基準クロックおよびチャネルごとの受信再生クロックを供給します。各PCSクワッドはこれらのクロックをプライマリ、及びセカンダリのFPGAクロック配線に提供します。また、各クワッドのPCS/FPGAインターフェイスには、FPGAファブリックから全4チャネルに供給する送受信クロックポートがあります。

各クワッドには、クワッド内のSERDESとPCSロジックの両方が、或いはSERDESのみを強制リセットする信号入力があります。さらに送受信方向共に、PCSロジック専用のリセットが各チャネルごとにあります。

送信データバス

送信データパスの信号はFPGAからPCSブロック内のFPGAブリッジまでです。高速伝送規格のために、(制御/ステータス信号と共に)8ビット幅の内部PCSデータパス用に2:1変換すること(ギアリング)ができます。

PCI Express x1用インターフェイスの最高速度は2:1変換しないモードで250MHzです。ギアリング2:1あり(すなわち、16ビット幅データパス)で、最大速度は156.25MHz(XAUI 4xチャンネルモード用)です。SERDESとPCSは、156.25MHz(2:1ギアリングあり)のインターフェイス速度に対応する3.125Gbpsデータレートまでサポートします。

表8-3 異なるモードごとの送信データパスのビット配分

データビット (ff_tx_d[23:0])	GE	XAUI	PCI Express (8b10b)	8-bit I/F (8b10b)	10-bit I/F ^{2,3}
[11], [23]	Correct_disp	0でドライブ	pcie_ei_en	correct_disp	0でドライブ
[10], [22]	xmit ⁴	0でドライブ	disp_sel	disp_sel ¹	0でドライブ
[9], [21]	0でドライブ	0でドライブ	force_disp	force_disp ¹	txd [9]
[8], [20]	k_cntrl	txc	k_cntrl	k_cntrl	txd [8]
[7:0], [19:12]	txd [7:0]	txd [7:0]	txd [7:0]	txd [7:0]	txd [7:0]

1. G8B10Bモードで動作しているとき、force_disp信号はデータワード[7:0]に対してdisp_sel信号で選択されたコラムのビットにディスパリティを強制します。disp_selが1の場合、10ビットコードは現在のRD+(正のディスパリティ)コラムから選択されます。disp_selが0の場合、10ビットコードは現在のRD-(負のディスパリティ)コラムから選択されます
2. データにギアリングがある(2:1)とき、低位ビット(ff_tx_d[9:0])が最初に送信されるオクテットであり、高位ビット(ff_tx_d[19:10])が2番目に送信されるオクテットです。データがギアリングなしの場合、低位ビット(ff_tx_d[9:0])が有効なビットであり、高位ビットは全てゼロにされるべきです。
3. 8ビットモード、或いは10ビットモードで用いられるインターフェイスと8b10bエンコーダ/デコーダはバイパスされます。例えば、SDIなど。
4. 自動ネゴシエーション・ステートマシンは信号xmitを生成します。これはハードロジック内でGigE Idle SMと相互にインターフェイスする為に用いられます。

受信データパス

受信パスの信号は、PCSブロック内のFPGAブリッジからFPGAまでです。データパスは8ビット幅の内部PCSデータパス用に2:1ギアリングされるかもしれませんが。ソフトウェア・レジスタビットでギアリングを禁止することが可能で、その場合パス幅は半分にされます。データがギアリングされる時、低位ビット(ff_rx_d[9:0])が最初に受信されたオクテットであり、高位ビット(ff_rx_d[19:10])が2番目に受信されたオクテットです。データがギアリングなしの場合、低位ビット(ff_rx_d[9:0])が有効なビットであり、高位ビットは使用されるべきではありません。

表8-4 異なるモードごとの受信データのバスビット配分

データビット (ff_rx_d[23:0])	GE ¹	XAUI	PCI Express (8b10b) ²	8-bit I/F (8b10b) ¹	10-bit I/F
[11], [23]	rx_even ²	NC	rxstatus[2] ³	NC	NC
[10], [22]	cv	cv	rxstatus[1] ³	cv	NC
[9], [21]	disp_err	disp_err	rxstatus[0] ³	disp_err	rx_d [9]
[8], [20]	k_cntrl	txc	k_cntrl	k_cntrl	rx_d [8]
[7:0], [19:12]	rx_d [7:0]	rx_d [7:0]	rx_d [7:0]	rx_d [7:0]	rx_d [7:0]

1. コードバイオレーションがあるとき、PCS 8b10bパケットデコーダはその出力を0xEEとKアサートに置き替えます(K=1かつd=EEは8b10b符号スペースに定義がありません)。
2. rx_evenはGigE自動ネゴシエーションと受信ステートマシン(FPGAロジックで実装される)での使用のためにGigEリンク・ステートマシンによって生成される信号です。

。

制御

各モードにはそれぞれ特有の制御信号の組があり、FPGAロジックから様々なPCS機能の直接制御を可能にします。通常これらの制御入力、それぞれの制御レジスタビットへのライト操作と同等の作用を与えます。ispLEVERデザインツールは、これらのポートをFPGAインターフェイスに取り出すためのオプションをユーザに与えます。

ステータス

各モードにはそれぞれ特有のステータスやアラーム信号の組があり、FPGAロジックからモニタできます。通常これらのステータス出力は、特定のステータス・レジスタビットにそれぞれ対応しています。ispLEVERデザインツールは、これらのポートをPCS FPGAインターフェイスに取り出すためのオプションをユーザに与えます。制御とステータスに関する詳細な情報については "モード特有の制御/ステータス信号" セクションを参照してください。

SCI (SERDESクライアント・インターフェイス)バス

SCIはソフトIPで、SERDES/PCSクワッドブロックがコンフィグレーション・メモリセルではなく、レジスタによって制御されることを可能にします。それは簡単なレジスタ・コンフィグレーション・インターフェイスです。

このテクニカルノートの利用

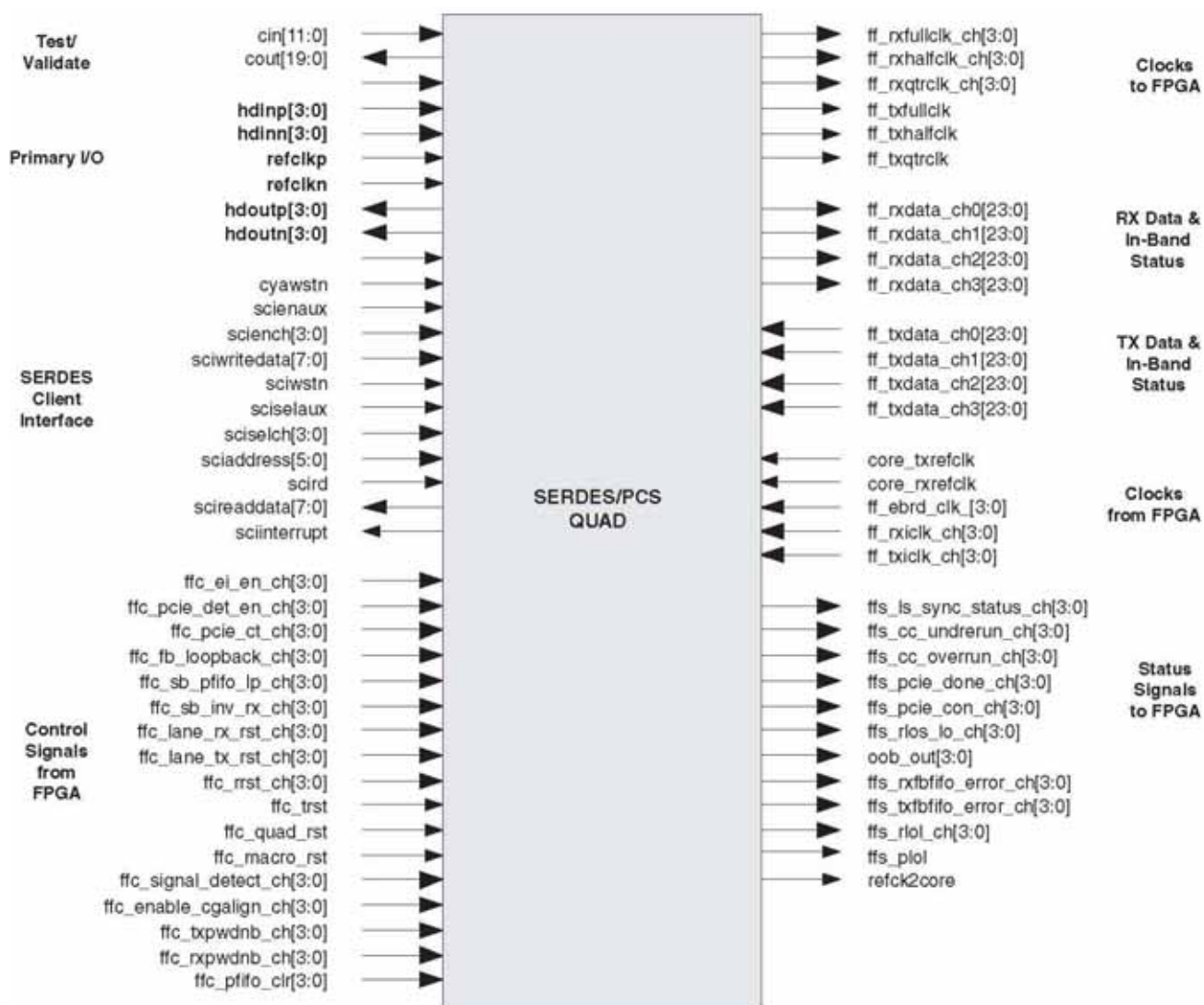
ラティスからのispLEVERデザインツールはPCSの全モードをサポートします。殆んどどのモードが特定の業界標準データ通信プロトコルのアプリケーション専用です。他のモードは、ユーザが自身のカスタムアプリケーション設定を定義する、より汎用のモードです。ispLEVERデザインツールで、ユーザのデザインで使用するモードに各クワッドを定義します。

このドキュメントはispLEVERによってサポートされるSERDESとPCSのすべてのモードの動作について説明します。このドキュメントは組み込みSERDESと、連携するPCSロジックの完全な機能を詳細に記述します。組み込みSERDESの電気的特性とタイミング特性はLatticeECP2/Mファミリデータシートにあります。PCSロジックの動作はPCSセクションにあり、SCIバスを介してアクセスすることができるSERDES/PCSロジックに関連する全ステータス・制御レジスタは、その一覧表がメモリマップ・セクションに提供されています。パッケージピン配置情報はLatticeECP2/Mファミリデータシートのアーキテクチャ・セクションに含まれています。

SERDES/PCS

クワッドは4チャンネルのRxとTx回路、及びTx PLLを含む補助チャンネルを含んでいます。Tx PLLへの基準クロックは、基準クロック差動入力ピンかFPGAコアから供給することができます。クワッドSERDES/PCSマクロは4レーンのデータのためにシリアルパラ変換とパラシリアル変換機能を実行し、マクロ内のPLLはFPGAロジックにシステムクロックを提供します。またクワッドはTxとRx各回路で、フルデータレートとハーフデータレート動作モードの両方を独立してサポートします。ブロックレベルの図が図8-4で示されています。

図8-4 SERDES_PCS ブロックの信号インターフェイス



I/O定義

表8-5にPCSクワッドに対するすべてのデフォルトおよびオプションのI/Oをリストアップします。ユーザは、IPexpress GUIを用いることでPCSクワッドのオプションポートを選ぶことができます。

表8-5 SERDES_PCS I/O記述

信号名	I/O	タイプ	記 述	Default / Optional
プライマリI/O, SERDESクワッド				
hdinp0	I	チャンネル	高速CML非反転入力、チャンネル0	D
hdinn0	I	チャンネル	高速CML反転入力、チャンネル0	D
hdinp1	I	チャンネル	高速CML非反転入力、チャンネル1	D
hdinn1	I	チャンネル	高速CML反転入力、チャンネル1	D
hdinp2	I	チャンネル	高速CML非反転入力、チャンネル2	D
hdinn2	I	チャンネル	高速CML反転入力、チャンネル2	D
hdinp3	I	チャンネル	高速CML非反転入力、チャンネル3	D
hdinn3	I	チャンネル	高速CML反転入力、チャンネル3	D
hdoutp0	O	チャンネル	高速CML非反転出力、チャンネル0	D
hdoutn0	O	チャンネル	高速CML反転出力、チャンネル0	D
hdoutp1	O	チャンネル	高速CML非反転出力、チャンネル1	D
hdoutn1	O	チャンネル	高速CML反転出力、チャンネル1	D
hdoutp2	O	チャンネル	高速CML非反転出力、チャンネル2	D
hdoutn2	O	チャンネル	高速CML反転出力、チャンネル2	D
hdoutp3	O	チャンネル	高速CML非反転出力、チャンネル3	D
hdoutn3	O	チャンネル	高速CML反転出力、チャンネル3	D
refclkp	I	クワッド	基準クロック非反転入力、専用CML入力	D
refclkcn	I	クワッド	基準クロック反転入力、専用CML入力	D
送受信データバス（詳細なバス使用法については表8-3を参照）				
ff_rxdata_ch0[23:0]	I	チャンネル	チャンネル0受信バスのデータ信号	D
ff_rxdata_ch1[23:0]	I	チャンネル	チャンネル1受信バスのデータ信号	D
ff_rxdata_ch2[23:0]	I	チャンネル	チャンネル2受信バスのデータ信号	D
ff_rxdata_ch3[23:0]	I	チャンネル	チャンネル3受信バスのデータ信号	D
ff_txdata_ch0[23:0]	O	チャンネル	チャンネル0送信バスのデータ信号	D
ff_txdata_ch1[23:0]	O	チャンネル	チャンネル1送信バスのデータ信号	D
ff_txdata_ch2[23:0]	O	チャンネル	チャンネル2送信バスのデータ信号	D
ff_txdata_ch3[23:0]	O	チャンネル	チャンネル3送信バスのデータ信号	D
制御信号				
ffc_sb_inv_rx_ch[3:0]	I	チャンネル	受信データの反転制御 1 = データを反転 0 = データを反転しない	O
ffc_enable_cgalalign_ch[3:0] ⁴	I	チャンネル	コンマアライナを制御 1 = コンマアライナをイネーブル 0 = コンマアライナを現在の位置にロック	O
ffc_signal_detect_ch[3:0] ⁴	I	チャンネル	リンクステートマシン (LSM) を制御 1 = LSMをイネーブル 0 = LSMをディセーブル	O
ffc_fb_loopback_ch[3:0]	I	チャンネル	FPGAブリッジ・ループバック 1 = RxからTxへのループバックをイネーブル 0 = 通常動作	O

ffc_sb_pfifo_lp_ch[3:0]	I	チャンネル	SERDESブリッジ・パラレルループバック 1 = RxからTxへのループバックをイネーブル 0 = 通常動作	O
ffc_pfifo_clr_ch[3:0]	I	チャンネル	SERDESブリッジ・パラレルループバックFIFO のクリア 1 = ループバックFIFOをリセット 0 = 通常のループバック動作	D
rx_sdi_en	I	チャンネル	BSCANモードのみで使用	O
tx_sdi_en	I	クワッド		
リセット信号				
ffc_lane_rx_rst_ch[3:0]	I	チャンネル	Highアクティブ、非同期入力。PCS内の個々のRxチャンネルロジックのみをリセット。	D
ffc_lane_rx_tst_ch[3:0]	I	チャンネル	Highアクティブ、非同期入力。PCS内の個々のTxチャンネルロジックのみをリセット。	D
ffc_rrst_ch[3:0]	I	チャンネル	Highアクティブ、非同期入力。SERDES Rxチャンネル内の一部ロジックのみをリセット。	D
ffc_trst	I	クワッド	Highアクティブ、非同期入力。SERDES全Txチャンネル内の一部ロジックをリセット。	D
ffc_quad_rst	I	クワッド	Highアクティブ、非同期入力。補助チャンネルを含むSERDES全チャンネルをリセット。	D
ffc_macro_rst	I	クワッド	Highアクティブ、SERDESクワッドへの非同期入力。PCSロジック以外の補助チャンネルを含むSERDES全チャンネルをリセット。ソフトウェア・レジスタビットfpga_reset_enableとゲートされる。fpga_reset_enableのデフォルトは1。	D
ffc_txpwnb_ch[3:0]	I	チャンネル	Lowアクティブ送信チャンネルパワーダウン。 0 = 送信チャンネルパワーダウン	D
ffc_rxpwnb_ch[3:0]	I	チャンネル	Lowアクティブ受信チャンネルパワーダウン。 0 = 受信チャンネルパワーダウン	D
ステータス信号				
ffs_rlos_lo_ch[3:0]	O	チャンネル	各チャンネル用LOS検出。閾値設定にはレジスタビットrlos_hset[2:0]が用いられる。Low側の閾値はユーザからアクセスできない。 1 = LOS 0 = 信号検出	D
ffs_ls_sync_status_ch[3:0]	O	チャンネル	1 = レーンはコンマに同期 0 = レーンはコンマを不検出。	D
ffs_cc_underrun_ch[3:0] ⁶	O	チャンネル	1 = Rxクロック補償FIFOのアンダーランエラー 0 = FIFOエラーなし	O
ffs_cc_overnrun_ch[3:0] ⁶	O	チャンネル	1 = Rxクロック補償FIFOのオーバーランエラー 0 = FIFOエラーなし	O
ffs_rxfbifo_error_ch[3:0]	O	チャンネル	1 = Rx FPGAブリッジFIFOエラー 0 = FIFOエラーなし	D
ffs_txfbifo_error_ch[3:0]	O	チャンネル	1 = Tx FPGAブリッジFIFOエラー 0 = FIFOエラーなし。	D
ffs_rlol_ch[3:0]	O	チャンネル	1 = Rx CDR LOL 0 = ロックを維持	D
ffs_plol	O	クワッド	1 = Tx PLL LOL 0 = ロックを維持	D

oob_out_ch[3:0] ³	O	チャンネル	FPGA内ビデオSERDES用へのチャンネルのシングルエンド出力	D
refck2core	O	クワッド	FPGAへの基準クロック	O
FPGAへのクロック信号				
ff_rxfullclk_ch[3:0]	O	チャンネル	Rxチャンネル再生クロック。ユーザモードではソースは常にチャンネルの再生クロック。GbEや10GbEなどCTCサポートの規格では、ソースは当該送信チャンネルのシステムクロック。PCSバイパスモードでもTxシステムクロックであり、8b10bモードか8b10bデコーダをディセーブルして(10-bit or 20-bit data path)実行する“raw”モードが必要です。	D
ff_rxhalfclk_ch[3:0]	O	チャンネル	Rxチャンネル再生ハーフクロック。2:1ギアリングでは2分周された出力。	D
ff_rxqtrclk_ch[3:0]	O	チャンネル	Rxチャンネル再生クォータ(1/4)クロック。もう一段の2:1ギアリング用。	O
ff_txfullclk	O	チャンネル	Tx PLLフルレートクロック	D
ff_txhalfclk	O	チャンネル	Tx PLLハーフレートクロック	D
ff_txqtrclk	O	チャンネル	Tx PLLクォータ (1/4) レートクロック	O
FPGAからのクロック信号				
core_rxrefclk	I	クワッド	FPGAからのRx基準クロック。CDR PLL用。	D
core_txrefclk	I	クワッド	FPGAロジックからのTx基準クロック。Tx SERDES PLL用。	D
ff_ebrd_clk_[3:0]	I	チャンネル	FPGAからのRxチャンネルクロック入力。CTC FIFO (エラスティックバッファ) リード用。	D
ff_rxiclk_ch[3:0]	I	チャンネル	FPGAからの各Rxチャンネル用クロック入力。基準クロックや受信基準クロックに同期するFIFOとRx FPGA部のインターフェイス用クロック。	D
ff_txiclk_ch[3:0]	I	チャンネル	FPGAからの各Txチャンネル用クロック入力。基準クロックに同期したFIFOとインターフェイスするため。CTCが用いられる場合は、Rx FPGA部が基準クロックに同期したFIFOにインターフェイスするために用いられる。	D
SERDESクライアント・インターフェイス(SCI)				
scienaux	I	R	1: sciwdataがクワッド制御レジスタにライト 0: メモリデータがクワッド制御レジスタにライト	O
scien_ch[3:0]	I	R	1: sciwdataがチャンネル制御レジスタにライト 0: メモリデータがチャンネル制御レジスタにライト	O
sciselaux	I	R	1: クワッドレジスタを選択	O
scisel_ch[3:0]	I	R	1: チャンネルレジスタを選択	O
sciaddress[5:0]	I	R	アドレスバス入力	O
scireaddata[7:0]	O	R	リードデータ出力	O
sciwrite[7:0]	I	R	ライトデータ入力	

scird	I	R	1: リードデータを選択 0: リードデータは非選択	O
sciwnstn	I	R	ライトストローブ	O
sciinterrupt	O	R	割り込み出力	O
cyawstn	I	R	1: “sciwnstn = 0” の場合、全メモリセルをレジスタにコピー 0: デフォルト	O
SERDESキャラクタライゼーション / テスト用バス				
cin[11:0]	I	R	キャラクタライズ用テストバスロジックのデータ入力	D
cout[19:0]	O	R	キャラクタライズ用テストバスロジックのデータ出力	D

1. コンフィグレーションの間、HDOUTPとHDOUTNは共にVCCOBにプルアップされます。
2. Generic 8b10b PCSモジュールはオプションとして4本のPCI制御/ステータス信号を含んでいます。用いられない場合、制御信号はGNDに接続し、またステータス信号はフローティングのままでも構いません。
3. CDRを用いないで信号を得る唯一の方法はOOB_OUT信号を用いることです。この信号はSDIモード用のみ。
4. 外部リンクステートマシンが選択されると、PCSポートリストにこれらの信号が表れます。図8-28を参照してください。
5. ラティスセミコンダクター社内での用途のみ。
6. これら信号はパルスです。正しくモニタするためにはラッチする必要があります。

SERDES/PCSの機能記述

LatticeECP2Mファミリのデバイスには、組み込みSERDES/PCSロジックのクワッドが1～4あります。各クワッドは、独立する全二重データチャンネルを4系統サポートします。単一チャンネルはデータリンクを1系統サポートすることができ、各クワッドはそのような4つのチャンネルまでサポートすることができます。モード選択はクワッド単位でされることに注意してください。例えば或るクワッドをギガビット・イーサネットモードに選択すると、そのクワッドにおける全4チャンネルがギガビット・イーサネットモード専用になります。

組み込みSERDES CDR PLLとTx PLLは広範囲の業界標準プロトコルをカバーするデータレートをサポートします。

図8-3はSERDES/PCSチャンネル内の主なブロックとサブブロックを説明します。

- SERDES
 - イコライザ
 - CDR(クロックデータ・リカバリ)
 - デシリアライザ
 - プリエンファシス
 - シリアライザ
 - シリアル・ループバック
- SERDESブリッジ(SB)
 - インバータ；受信データを反転します。PCI Expressで必要です。
 - SERDESブリッジ・パラレル・ループバック
- PCSコア
 - ワードアライメント

- 8b10bデコーダ
- 8b10bエンコーダ
- リンクステートマシン
- エラスティック・バッファ(CTC)
- ・FPGAブリッジ(FB)
 - ダウンサンプルFIFO
 - アップサンプルFIFO
 - PCSパラレル・ループバック

SERDES

イコライザ

デジタル伝送のデータレートがGbps以上に進化するにつれて、周波数依存の減衰によって受信信号に厳しいシンボル間干渉をもたらす結果となり得ます。従ってイコライザを用いることは、正しくデータを復元するために必須になります。LatticeECP2Mでは3つの極配置オプションを用意しており、これらは低/中/高周波領域です。

プリエンファシス

プリエンファシスはシステムプロセスに関するものです。ユーザは最大80%のプリエンファシスを選択することができます。

レシーバのCID(同一ディジットの連続)

Rx-CID ; BER 10^{-12} でのレシーバのCID(0/1同一ディジットの連続)

- ・ 3.125Gbpsで7ビット
- ・ 1.25Gbpsで20ビット

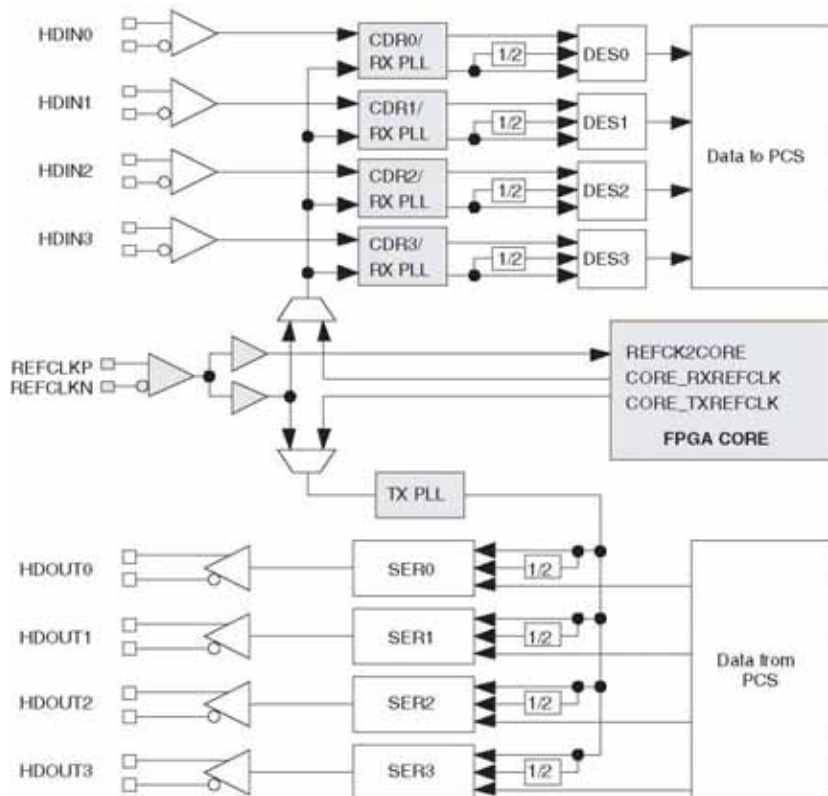
これはDC結合を用いるとき、入力データストリームが0/1の遷移なしで許容されるビット数です。

基準クロックの使用

1本の基準クロック(REFCLK)がLatticeECP2Mファミリではサポートされます。Tx PLLと4個のRx PLLはすべて同じ周波数で、基準クロック周波数の倍数で動作します。各チャネルのTxシリアルライザは、そのままのレート(フルデータレート・モード)か半分のレート(ハーフデータレート・モード)で動作するかを、独立してプログラムすることができます。同様に、各チャネルのRxデシリアルライザもフルデータレート・モード、ハーフデータレート・モードのいずれで動作するかを独立してプログラムすることができます。すべてのTxとRxが同じモードでプログラムされると(通常はフルレート)、クワッドの全4チャネルが同一Tx/Rx周波数で動作することになります。

SERDES内の送信PLLは、ピンからの外部基準クロックか、FPGAコア(core_txrefclk)から提供される基準クロックかのどちらかに、ロックすることができます。SERDES内の受信PLLも、ピンからの外部基準クロックかFPGAコア(core_rxrefclk)から供給される基準クロックのどちらかにロックすることができます。

図8-5 ブロック図、基準クロックの使用



基準クロックのソース

refclkp, refclkn

CMLのSERDESクロック専用入力。送受信に特に別のソースが用意されていない限り、最初のオプションです。ソースはCML、LVDS、LVPECLのいずれかです。インターフェイス例についてはTN1114 (Electrical Recommendations for Lattice SERDES) を参照してください。

core_rxrefclk, core_txrefclk

FPGAロジックからの基準入力。プライマリクロック(PCLK)を使用すべきです。

FPGA PLL

FPGA PLLが基準クロックとして用いられるとき、そのPLLに対する基準クロックは専用クロック入力ポートを用いるべきです。高速データレート用のクロックではFPGA PLL出力のジッタはシステム規格を満たさないかもしれません。ジッタが重要な用途では用いないことを推奨します。

フルデータレートとハーフデータレート

各TxシリアルライザとRxデシリアルライザは、フルデータレートとハーフデータレートに分けることができ、各方向と各チャンネルにおいて2つの異なるデータレートを許容します。

このデュアルレート機能を用いるには、チャンネルベースのプロトコルモードを選択しなければなりません。

例:

1. IPexpress(図8-22)のQuad Tabウィンドウでは、G8B10B Mode、Channel Based Protocol Mode、Channel0(Full Rate)が選択されています。

2. Reference Clock(CM)ウィンドウ(図8-24)で、Full Rate Channelコラムの下に、以下を入力します。

- ・シリアル・ビットクロックレート: 2.5 GHz
- ・基準クロックの乗数: 10X
- ・残り3つのエントリはそのまま

3. Half Rate Channelコラムはハーフレートのために計算された値を表示します:

	フルレートチャンネル	ハーフレートチャンネル
シリアル・ビットクロックレート	2.5GHz	1.25GHz
基準クロック乗数	10X	5X
計算された基準クロックレート	250MHz	250MHz
FPGAインターフェイス・データバス幅	8	8
計算後のFPGA I/Fクロックレート	250MHz	250MHz

フルクロック、ハーフクロック、およびクォータクロックの使用

多くの場合、図8-32で図示されるff_txfullclkはff_rxiclk_chx、ff_txiclk_chx、ff_ebrd_clk_xに用いられます。

Reference Clock MultiplierとFPGA Interface Data Bus Widthが選択されているとき、IPexpress GUIは自動的にFPGAインターフェイスのクロック周波数を計算します。

表8-6はrefclk_multiplierモードと、8ビットまたは16ビットのインターフェイス・データバス幅の取り得るすべての組み合わせに対するクロック使用例について示します。

表8-6 クロック使用例、G8B10B モード、REFCLK=120MHz

基準クロック乗数	10xH	10x	20xH	20x
ビットレート	600 Mbps	1.2 Gbps	1.2 Gbps	2.4 Gbps
8ビット・インターフェイス例				
rxfullclk ¹	60	120	120	240
rxhalfclk	30	60	60	120
txfullclk	120	120	240	240
txhalfclk	60	60	120	120
txqtrclk	30	30	60	60
16ビット・インターフェイス例				
rxfullclk	60	120	120	240
rxhalfclk ¹	30	60	60	120
txfullclk	120	120	240	240
txhalfclk	60	60	120	120
txqtrclk	30 ^{2,3}	30	60	60

1. CTC BYPASSモードでは、これら再生クロックはrxiclkのソースとして用いられます。図8-34/-36参照。
2. 陰影をつけられたセル中のクロックは、FPGAインターフェイス・クロックとして各モードで使用されます。
3. このモードが選択されるとき、コンフィグレーションGUIのOptional PortタブウィンドウでPLL Quarter Clockをチェックしなければなりません(図8-28を参照してください)。

フルデータレート・チャンネルのVCOはビットクロックと同じです。ハーフデータレート・チャンネルではビットクロックはVCOの半分です。

LOS(信号の喪失)

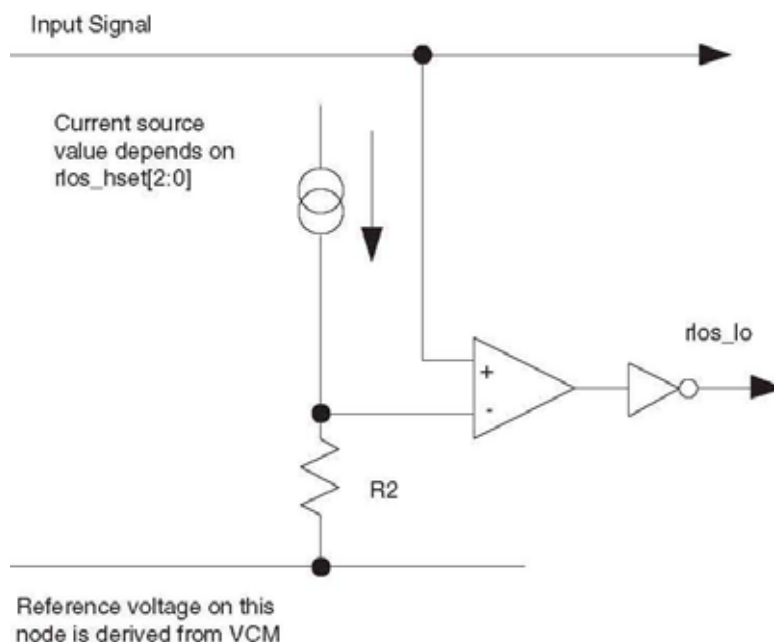
各チャンネルは図8-6で示されるようにプログラマブルなLOS(信号の喪失)検出器を含んでいます。LOS閾値はプログラマブル電流源の値に依存し、rlos_hset[2:0]制御ビットを用いて選ばれます。

LOL(ロックはずれ)

送信PLLと個別チャンネルのCDRには、いずれもカウンタベースのデジタルLOL検出器があります。送信PLLのロックがはずれるとLOLがアサートされ、PLLが再ロックするまでそのままです。

CDRのロックがはずれると、そのチャンネルのLOLがアサートされ、CDRのVCOを基準クロックにロックしようと試みます。これが達成されるとそのチャンネルのLOLはネゲートされ、CDRは入力データにロックするために元に戻ります。CDRはデータにロックし続けるか、または再びロックはずれに戻り再トレーニングへと、その繰り返しとなります。

図8-6 LOS検出器



注：rlos_lsel[2:0]制御ビットと関連するステータスビットはラティス社内のみでの用途です。

Txレーン間スキュー

制御ビットsync_toggleは、アクティブな全Txチャンネルがbit0からパラシリ変換を始めるためのリセット用に加えられました。ほとんどのマルチチャンネル・プロトコル規格では、ある規定値の中にTxレーン間スキューが収まることを保証する要件があります。マルチチャンネル・アライメント (MCA) はLatticeECP2MのハードPCSではサポートされません。

Txシリアルライザへのリセットはsync_toggle制御ビットをトグルするか、PLL LOLの遷移によって生成されます。リセットはすべてのアクティブなTxシリアルライザに適用されます。これらのソース信号が共にレベルである場合、Txシリアルライザは通常動作しています。

PCS機能のセットアップ

LatticeECP2M PCSは様々なアプリケーションでの用途に構成することができます。セットアップはispLEVER IPexpressモジュール生成ツールで選ばれ、ユーザはPCSのモードと機能オプションを選択できます。選択オプションは自動構成(auto-configuration)ファイルの中に保存され、これはispLEVERビットストリーム・ジェネレータによって用いられてビットストリーム内に書き出されます。PCS選択オプションを変えるためには、ユーザはIPexpressを再実行してPCSモジュールを作り直し、新しい自動構成ファイルを作成することを推奨します。ビットストリーム・ジェネレータを実行する前に、マニュアルで自動構成ファイルを編集することによって幾つかのオプションを変えることもできます。

コンフィグレーション後に、オプションのSERDESクライアント・インターフェイス(SCI)バスを通してPCSレジスタに書くことによって、動的にPCSオプションを変えることができます。SCIはソフトIPで、SERDES/PCSクワッドがコンフィグレーション・メモセルの設定と異なるレジスタ値によって制御されることを可能にします。SCIを介してアクセスできる制御/ステータス・レジスタの一覧表を、このドキュメントのメモリマップ章に示してあります。

自動構成 (auto-configuration) ファイル

ispLEVERの自動構成機能を用いることで、それぞれのPCSモードのための初期レジスタ設定を行うことが可能です。モジュール・ジェネレータは、選ばれたモード用にクワッド/チャンネルレジスタ設定の自動構成ファイル (<module名>.txt) を生成します。このファイルはフロントエンド・シミュレーションで参照することができ、またビットストリームに統合されます。すべてのクワッド/チャンネルレジスタが、コンフィグレーションの際に自動構成ファイルで定義された値に設定されます。したがって全レジスタが自動構成ファイルで設定されるのみで良ければ、SCIは不要です。他方、ユーザがデバイス動作中に制御レジスタの値を変えるか、またはステータス・レジスタをモニタする必要がある場合は、デザイン内にSCIを含まなければなりません。

送信データ

PCSクワッド送信データバスはチャンネル単位での8b10bエンコーダとシリアライザから成ります。

8b10bエンコーダ

このモジュールはIEEE 802.3ae-2002 1000BASE-X仕様の中で記述されている8b10bエンコーダを実装しています。エンコーダは仕様で説明されるように8ビットから10ビットコードへの変換を実行し、同時に規定されているディスペリティ規則を維持します。CHx_8B10B (xはチャンネル番号) の属性をBYPASSに設定することによって、8b10bエンコーダをバイパスすることができます。

シリアライザ

8b10b符号化されたデータはパラシリ変換され、組み込みSERDESを介してチップ外部に送信されます。

受信データ

PCSクワッド受信データバスはチャンネルごとに次のサブブロックからなります；デシリアライザ、ワードアライナ、8b10bデコーダ、オプションのリンクステートマシン (LSM)、およびオプションの受信クロックトレランス補償(CTC)FIFO。

デシリアライザ

データは組み込みSERDESを介してオンチップに取り込まれ、シリアルからパラレルになります。

ワードアライメント(バイト境界の検出)

このモジュールはコンマ符号語の検出とアライメント動作を実行します。受信ロジックで入力されるデータストリームに10ビットのワードアライメントを実行するために、コンマキャラクタが用いられています。

CHx_COMMA_ALIGN(xはチャンネル番号)のアトリビュートをBYPASSに設定することによって、ワードアライメントをバイパスすることができます。コンマ記述は802.3.2002 1000BASE-X仕様のセクション36.2.4.9と、セクション48.2.6.3(10GBASE-X仕様の図48-7)にあります。

ワードアライメント・モジュール内で多くのプログラマブル・オプションがサポートされます:

- ・ソフトウェア・イネーブル制御(User Configured ; UCモード)

注: UC_Modelは8-bit SERDES Only、10-bit SERDES Only、SD-SDI、HD-SDIを意図します。

・2つのプログラマブル・ワードアライメント・キャラクタ(通常正と負のデイスパリティのためにそれぞれ1つ)を設定できる機能と、アライメント・コンペア(比較)のためのビット単位でプログラマブルなマスクレジスタ。アライメント・キャラクタとマスクレジスタはクワッド毎に設定されます。

多くのプロトコルにおいて、"XXX0000011" (正のランニング・デイスパリティ・コンマキャラクタでコードグループK28.1、K28.5、およびK28.7に適合するjhgfiiedcbaビット)と "XXX1111100" (負のランニング・デイスパリティ・コンマキャラクタでコードグループK28.1、K28.5、およびK28.7に適合するjhgfiiedcbaビット)にワードアライメント・キャラクタを設定することができます。しかしながら、ユーザは最大10ビット長のどのようなビットパターンも定義することができます。

・最初のアライメント・キャラクタは属性COMMA_Aに割り当てられた10ビットの値によって定義されます。この値はPCSクワッドの全チャンネルに適用されます。

・第二のアライメント・キャラクタは属性COMMA_Bに割り当てられた10ビットの値によって定義されます。この値はPCSクワッドの全チャンネルに適用されます。

・マスクレジスタは、どのワードアライメント・ビットを比較したらよいかを定義します(マスクレジスタで1のビットは、ワードアライメント・キャラクタレジスタで対応するビットをチェックすることを意味します)。属性COMMA_Mに割り当てられた10ビットの値によって定義されたマスクレジスタ。この値はPCSクワッドの全チャンネルに適用されます。

属性CHx_COMMA_ALIGNがAUTOに設定されると、プロトコルベースのリンクステートマシンの1つがワードアライメントを制御します。詳しい動作に関する情報は、後述のプロトコル特有のリンクステートマシン記述を参照してください。

8b10bデコーダ

8b10bデコーダは、IEEE802.3-2002規格で記述されるようにデコーダ動作を実装しています。デコーダはランニング・デイスパリティの検証と共に10ビットから8ビットコードへの変換を実行します。CHx_8B10B(xはチャンネル番号)の属性をBYPASSに設定することによって、8b10bデコーダをバイパスすることができます。

コードバイオレーションが検出されると、ff_rxdata受信データは0xEEに、ff_rx_k_cntrl_chは1に設定されます。

プロトコル特有のリンクステートマシン(LSM)

PCSは様々なクワッドモードで用いられる種々プロトコルのためにLSMを実装しています。

プロトコル特有のLSMが選択されるとき、CH(0-3)_COMMA_ALIGNをAUTOに設定することによって、そのチャンネルのLSMをイネーブルしなければなりません。各モードでイネーブルされる特有LSMの選択は、以下に記述され、また図8-7にまとめられています。

PROTOCOL属性がGIGEであるとき、ギガビット・イーサネット用のLSMが選択されます。リンク同期は、アラインされたコードワード(符号語)が必要数連続して検出された後に達成されます。ギガビット・イーサ

ネットリンク同期ステートマシンは802.3-2002 1000BASE-X仕様の図36-9で示される同期ステート図を実装します。

G8B10Bと10-bit SERDES Onlyプロトコルでは、COMMA_ALIGNがAUTOに設定されるとき、ギガビット・イーサネット・リンクステートマシンが使用されます。

外部リンクステートマシンのオプション

属性CHx_COMMA_ALIGNがDYNAMICに設定されるとき、プロトコル特有のLSMはバイパスされます。その場合、ffc_enable_cgalign_ch(0-3)が少なくとも1クロック周期Lowにされた後、ワードアライナは比較が成功した最初か第二のユーザ定義ワードアライメント・キャラクタのどちらかで、アライメントはロックします。(入って来るデータをユーザ定義のワードアライメント・キャラクタに比較するのをやめて、現在のアライメントを維持します。)その後FPGAインターフェイスのffc_enable_cgalign_ch(0-3)ポートに与えられるLowパルスは、ワードアライナをアンロックします。そして、ワードアライナはユーザ定義のワードアライメント・キャラクタの1つに次に一致した時に再ロックされます。望まれるなら、一定の条件下でのワードアライメントの変化を許容するために、PCSクワッドの外部に実装されたLSMによりffc_enable_cgalign_ch(0-3)を制御することができます。

図8-7はLSMオプションについて図示します。

図8-7 PCS ワードアライナとLSMオプション

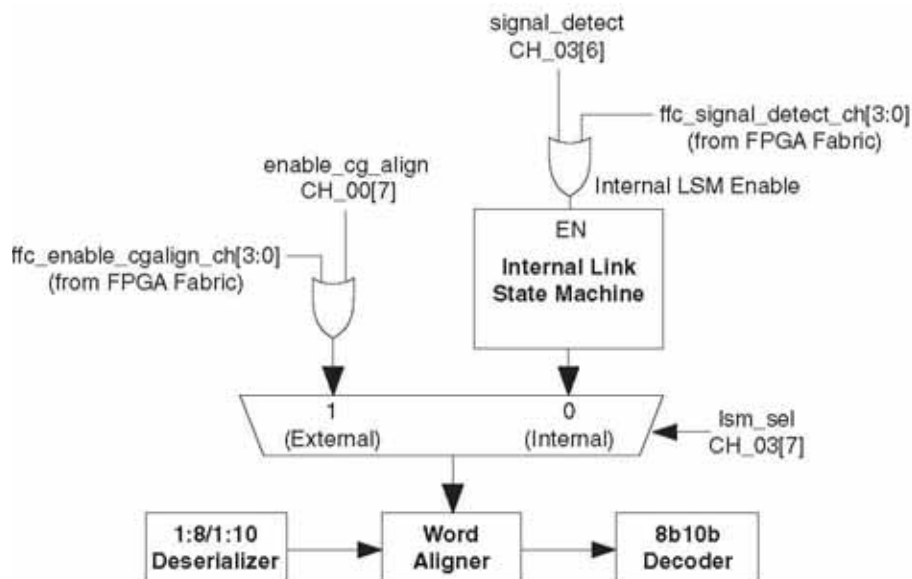


表8-7 リンクステートマシン (LSM) とワードアライナ(WA)の選択

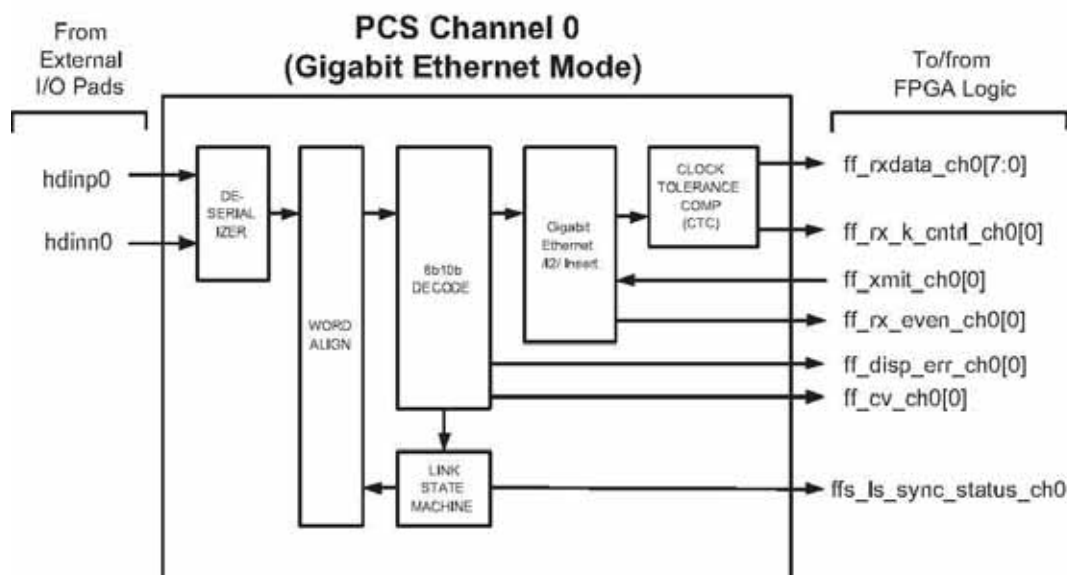
COMMA ALIGNモード	記 述
AUTO	WAがイネーブル、LSMがイネーブル(GbE LSM: デフォルト)
DYNAMIC	WAがイネーブル、LSMはディセーブル。cg_alignとsig_detect信号は0に設定される。潜在的に外部LSMが両信号を制御できる
BYPASS	WAはバイパス、LSMはディセーブル。ユーザは、FPGAコアにワードアライナを設計し、ロジック内に自らのcg_alignとsig_detect信号を与えることもできる。

LSMが選択/イネーブルされたとき、リンク同期に成功すると、その特定チャネルに関するステータス信号ffs_ls_sync_status(0-3)はHighになります。

ギガビット・イーサネットモードのアイドル挿入

Generic 8b10bモードには、さらにワードアライメントのためにLSMを選択するオプションがあります。ギガビット・イーサネットモードに設定されたPCSは、自動ネゴシエーションのために、受信データストリームの中に $1/12$ シンボル挿入を行います。ギガビット・イーサネットの自動ネゴシエーションはソフトロジックで実行されます。この機能は2048クロックサイクル毎に8つの $1/12$ オーダーセット・シーケンスを挿入します。 $1/12$ 挿入は自動ネゴシエーション・ソフトロジックからドライブされるPCSへのff_xmit_ch(0-3)入力によって制御されます。また、PCSから自動ネゴシエーション・ソフトロジックまでの信号ff_rx_even_ch(0-3)[0]を提供します。図8-8はPCSがギガビット・イーサネットモードに設定されたときの、1チャンネルの受信ロジックを示し(この例ではチャンネル0)、これらの制御/ステータス信号を示しています。

図8-8 ギガビット・イーサネットモードのPCS受信パス(チャンネル0の例)



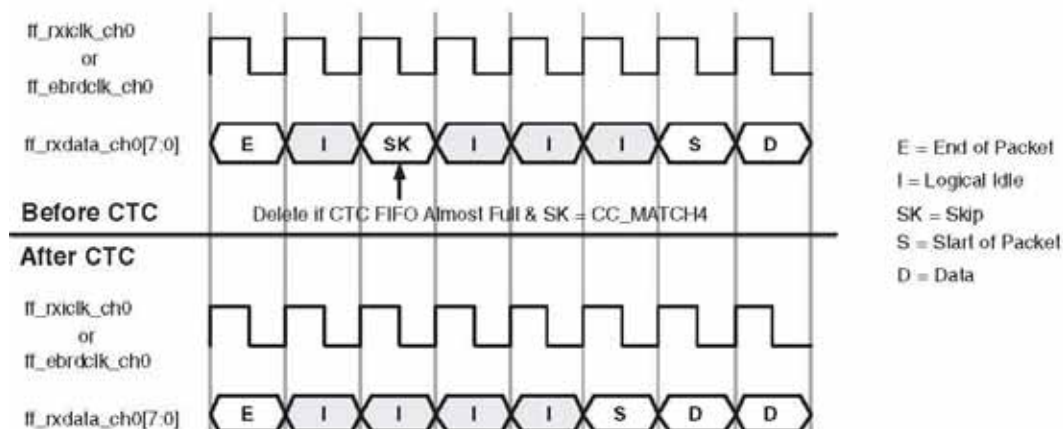
クロックトレランス補償(CTC)

CTCは、受信再生クロックとロックする基準クロックの間のクロックレートの調整を実行します。クロック補償は、パケットデータの損失を引き起こすことなく、事前に定められた位置でバイトを挿入するか、または削除することによって実行されます。16バイトのエラスティックFIFOは、2つのクロックドメイン間のデータ転送に用いられ、LatticeECP2M SERDESのためにクロック偏差の規定ppm上限まで対応します。(LatticeECP2/Mファミリデータシートの "DCおよびスイッチング特性" セクションを参照して下さい。)

属性CHx_CTC_BYPがNORMALに設定されると、そのチャンネルのCTCブロックはイネーブルされ、BYPASSに設定されるとバイパスされます。

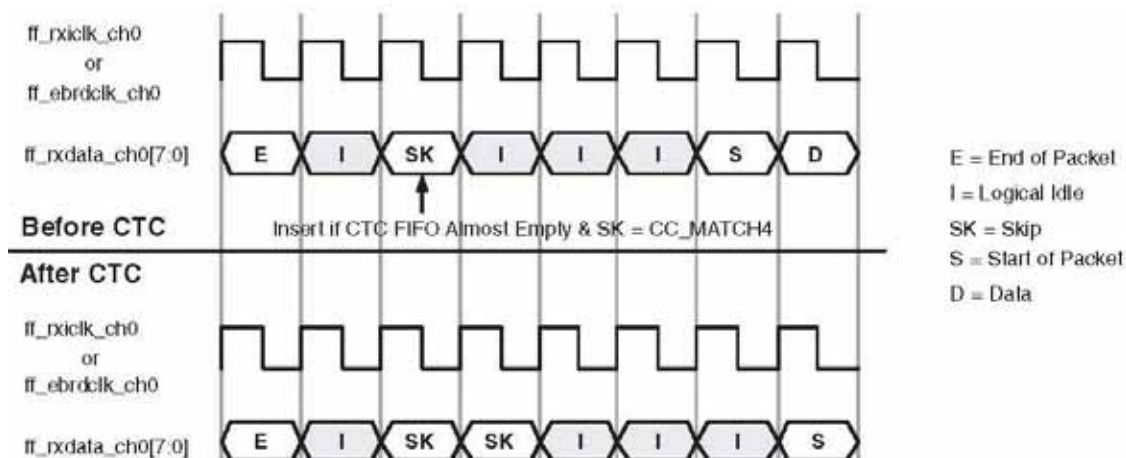
1バイト削除について図8-9で示します。

図8-9 クロックトレランス補償、1バイト削除の例



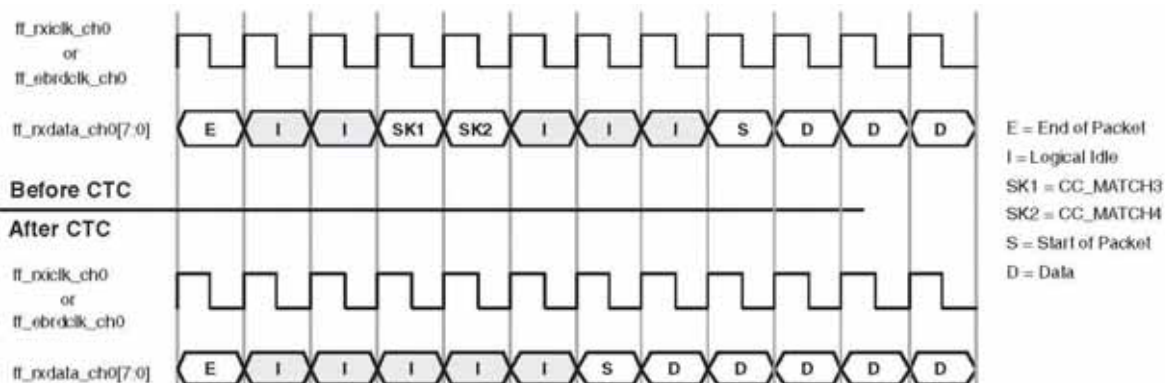
1バイト挿入について図8-10で示します。

図8-10 クロックトレランス補償、1バイト挿入の例



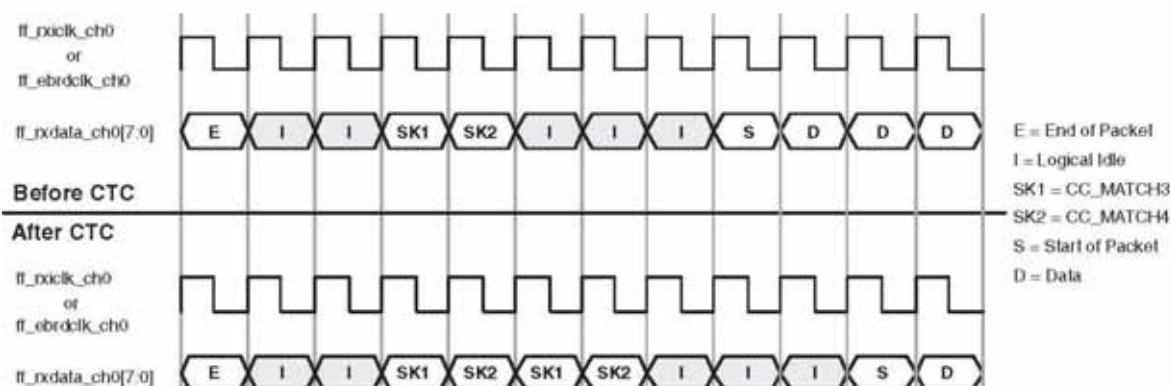
2バイト削除について図8-11で示します。

図8-11 クロックトレランス補償、2バイト削除の例



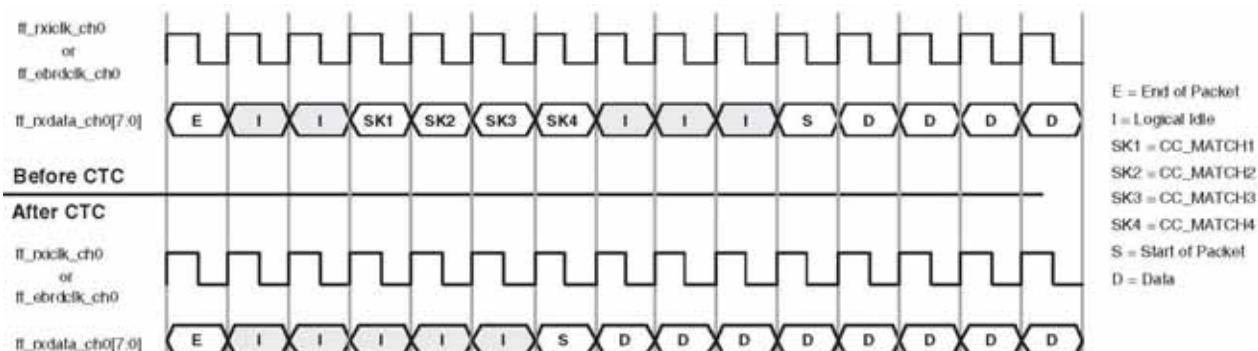
2バイト挿入について図8-12で示します。

図8-12 クロックトレランス補償、2バイト挿入の例



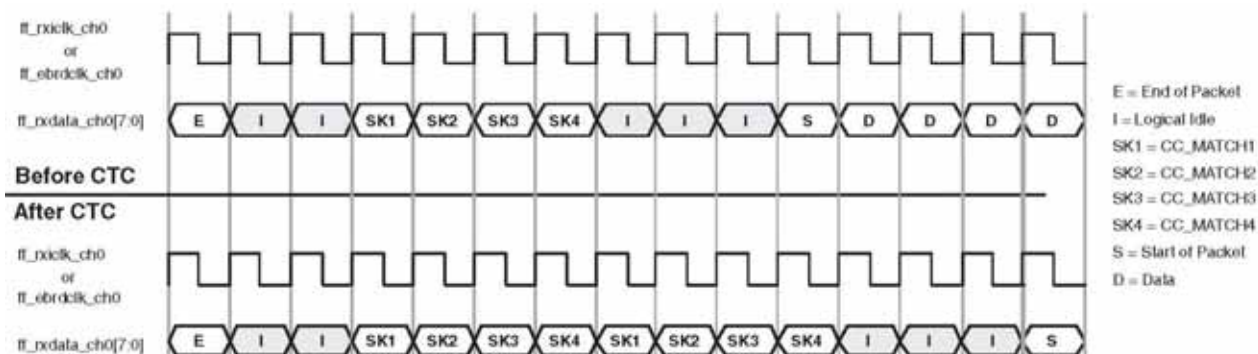
4バイト削除について図8-13で示します。

図8-13 クロックトレランス補償、4バイト削除の例



4バイト挿入について図8-14で示します。

図8-14 クロックトレランス補償、4バイト挿入の例



クロック補償値はクワッドベースで設定されます。CHx_CTC_BYP(xはチャネル番号)の属性をBYPASSに設定することによって、CTCをバイパスすることができます。CHx_CTC_BYPをNORMALに設定することは、CTCがアクティブであることを意味します。ispLEVERモジュール・ジェネレータでは、チャネルをSingleと定義するとCTCをイネーブルする自動構成ファイルが作成されます。チャネルをMCA Group1かMCA Group 2と定義すると、CTCをバイパスする自動構成ファイルが作成されます。CTCが使用されるとき、意

図されたアプリケーションのクロック補償のために、以下の設定を適切に設定しなければなりません。

- ・ CC_MATCHMODE属性を用いて、挿入/削除パターン長を設定します。これは挿入や削除を実行する前に CTCが比較しスキップするバイト数を設定します。CC_MATCHMODEの値は、MATCH_4 (1バイトの挿入/削除)と、MATCH_3_4 (2バイトの挿入/削除)、及びMATCH_1_2_3_4 (4バイトの挿入/削除)です。また、ターゲット・アプリケーションに適切な最小パケット間ギャップを設定しなければなりません。パケット間ギャップは属性CC_MIN_IPGに値を割り当てることによって設定されます。CC_MIN_IPGの許容値は0、1、2、および3です。これらの属性設定に基づいてスキップキャラクタ削除が実行された後の、許容される最小パケット間ギャップは以下の表8-8で説明されています。
- ・ 選ばれたCC_MATCHMODEに対応するように、スキップバイトかオーダーセットが設定されなければなりません。4バイトの挿入/削除(CC_MATCHMODE = MATCH_1_2_3_4)に関しては、属性MATCH_1を最初のバイトに割り当て、属性MATCH_2に第二バイト、属性MATCH_3を第三バイト、そして属性MATCH_4に第四バイトをそれぞれ割り当てなければなりません。割り当てられる値は2進の10ビット値です。例えば、4バイトのスキップ・オーダーセットが /K28.5/D21.4/D21.5/D21.5 である場合、MATCH_1は "0110111100"、MATCH_2 = "0010010101"、そしてMATCH_3 = MATCH_4 = "0010110101" であるべきです。2バイトの挿入/削除 (CC_MATCHMODE = MATCH_3_4) に関しては、属性MATCH_3が最初のバイトに割り当て、属性MATCH_4に第二バイトをそれぞれ割り当てなければなりません。1バイトの挿入/削除 (CC_MATCHMODE = MATCH_4)に関しては、属性MATCH_4が最初のバイトに割り当てられなければなりません。
- ・ クロック補償FIFOにおいて、ターゲット・プロトコル用のHigh/Lowウォーターマーク (水位標) を設定しなければなりません。値は0～15の範囲を取り得、HighウォーターマークをLowウォーターマークより高い値に設定しなければなりません(等しい値に設定するべきではありません)。属性CCHMARKに値を割り当てることによって、Highウォーターマークが設定されます。CCHMARKの許容値は16進値の0からFです。属性CCLMARKに値を割り当てることによって、Lowウォーターマークが設定されますCCLMARKの許容値も16進値の0からFです。
- ・ ispLEVERモジュール・ジェネレータでPCSブロックを生成するとき、Error Status Portsが選択されている場合、チャンネルごとのクロック補償FIFOオーバーランは、PCS/FPGAインターフェイスで ffs_cc_overrun_ch(0-3)という名称のポートでモニタすることができます。
- ・ ispLEVERモジュール・ジェネレータでPCSブロックを生成するときError Status Portsが選択されている場合、チャンネルごとのクロック補償FIFOアンダーランは、PCS/FPGAインターフェイスで ffs_cc_underrun_ch(0-3)という名称のポートでモニタすることができます。

最小パケット間ギャップの計算

表8-8はパケット間ギャップ(CC_MIN_IPG属性で定義される)のユーザ定義の値と、PCSからのスキップ・キャラクタ削除後の保証されるパケット間最小バイト数との関係を示します。表は乗数としてパケット間ギャップを示します。パケット間の最小バイト数と、挿入/削除を表中で示された乗数倍されたバイト数は等しいです。例えば挿入/削除あたりのバイト数が4(CC_MATCHMODEはMATCH_1_2_3_4に設定される)で、かつ最小パケット間ギャップの属性C_MIN_IPGが2の時、最小のパケット間ギャップは4 (CC_MATCHMODE = MATCH_1_2_3_4) 掛ける3 (表8-8のCC_MIN_IPG = 2)、即ち12バイトに等しくなります。最小のパケット間バイト数がCTCを通り抜けるまで、PCSはスキップキャラクタの削除を実行しません。

表8-8 最小パケット間ギャップの乗数

CC_MIN_IPG	挿入/削除の乗数
0	1 X
1	2 X
2	3 X
3	4 X

クロックドメイン

図8-15は、PCS内でCTCブロックを利用するモードでの、単一チャネルの送受信両方向のクロックドメインを示します。

送信側では、FPGAインターフェイスにおけるff_txclk_ch入力からロックした基準クロックまでのクロックドメイン転送は、FPGA送信インターフェイスFIFOで起こります。FPGA送信インターフェイスFIFOは、同じ周波数である2クロック間の位相差を吸収する意図があります。これらのFIFO(1チャネルあたり1つ)は周波数偏差を補うことはできません。

受信側のクロックドメイン転送は、チャネル再生クロックから基準クロック間のCTCブロックで起こります。CTCは、LatticeECP2Mの最大許容値まで、再生受信クロックと基準クロックの周波数差を調節することができます。CTCの後、FPGA受信インターフェイスFIFOで、ロックされた基準クロックとff_rxclk_chの間でクロック・インターフェイスがあります。FPGA受信インターフェイスFIFOは、同一周波数の2クロック間の位相差の調整をすることが目的です。FPGA受信インターフェイスFIFO(1チャネルあたり1つ)は、周波数偏差を補うことはできません。

図8-15 CTCモードのPCSクロックドメイン転送

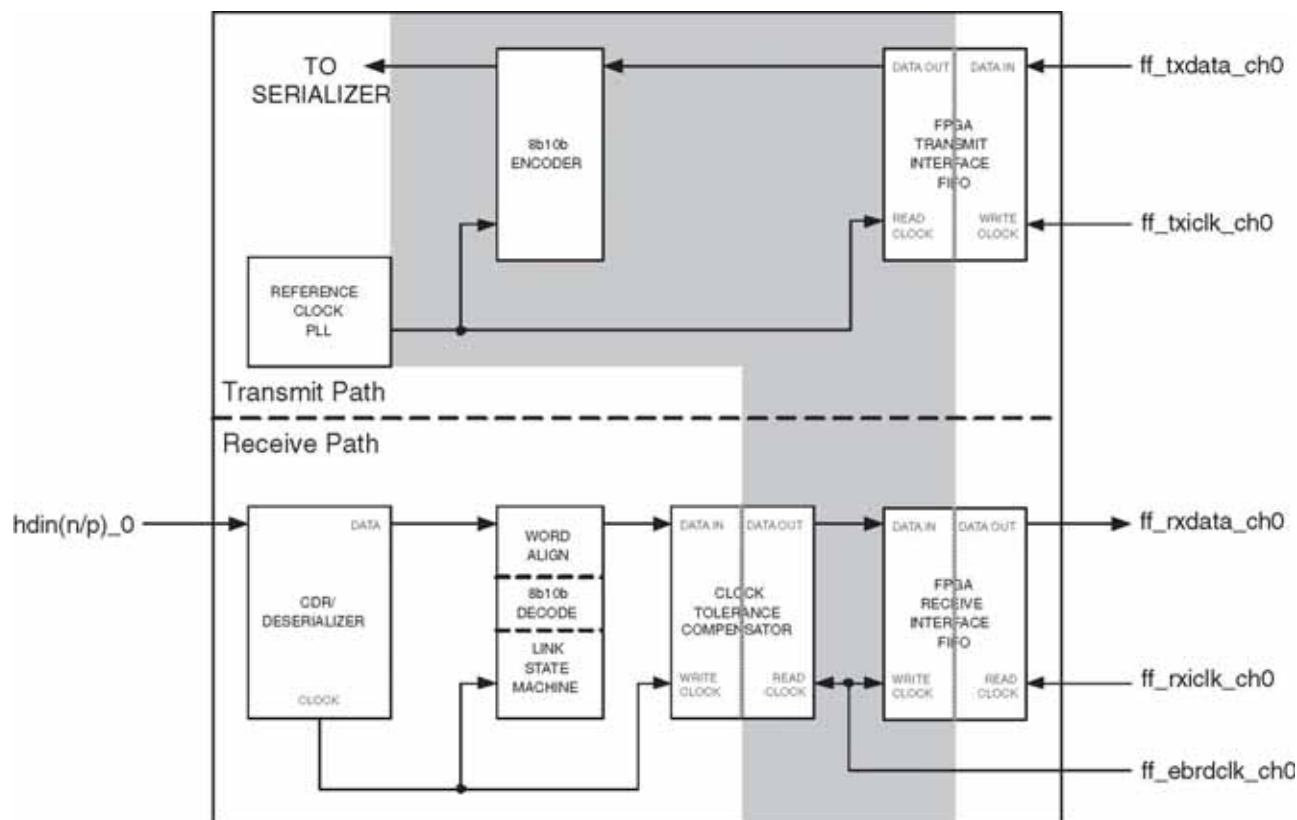
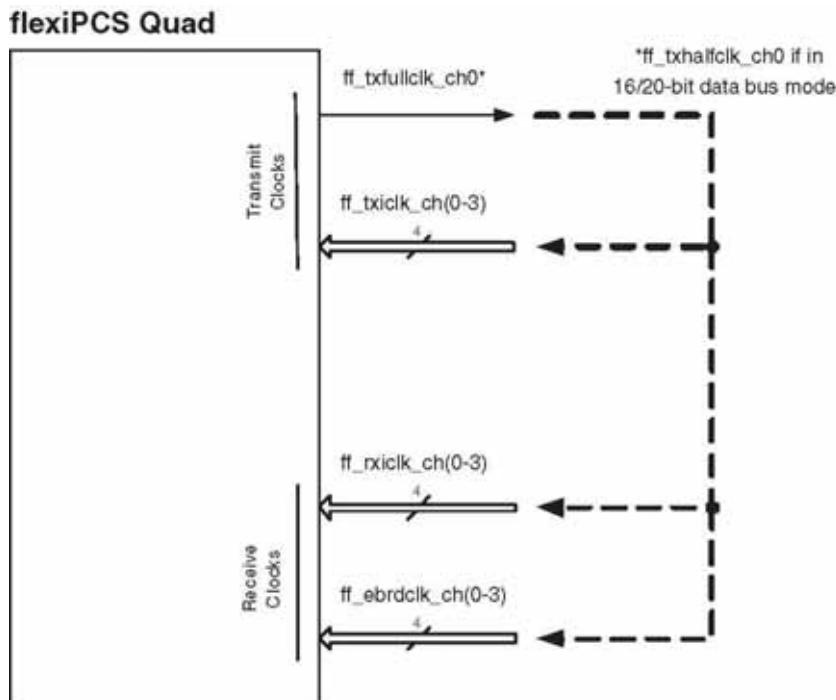


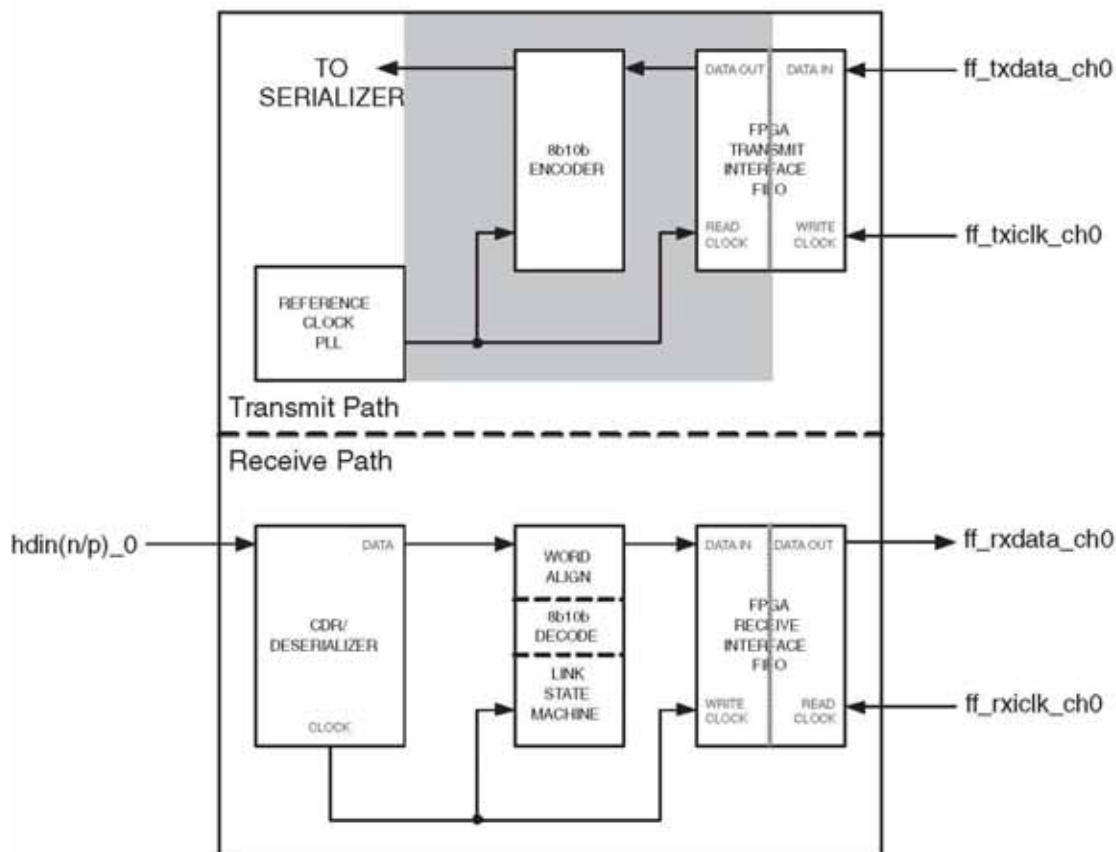
図8-16 CTCを用いるPCSクワッドへの同期入力クロック



同期インターフェイスを保証するためには、送信クロック入力と受信クロック入力の両方が、Generic 8b10b モードに設定されたPCSクワッドの基準クロック出力の1つからドライブされるべきです。図8-16は同期インターフェイスとなるような、取り得る接続について図示します。

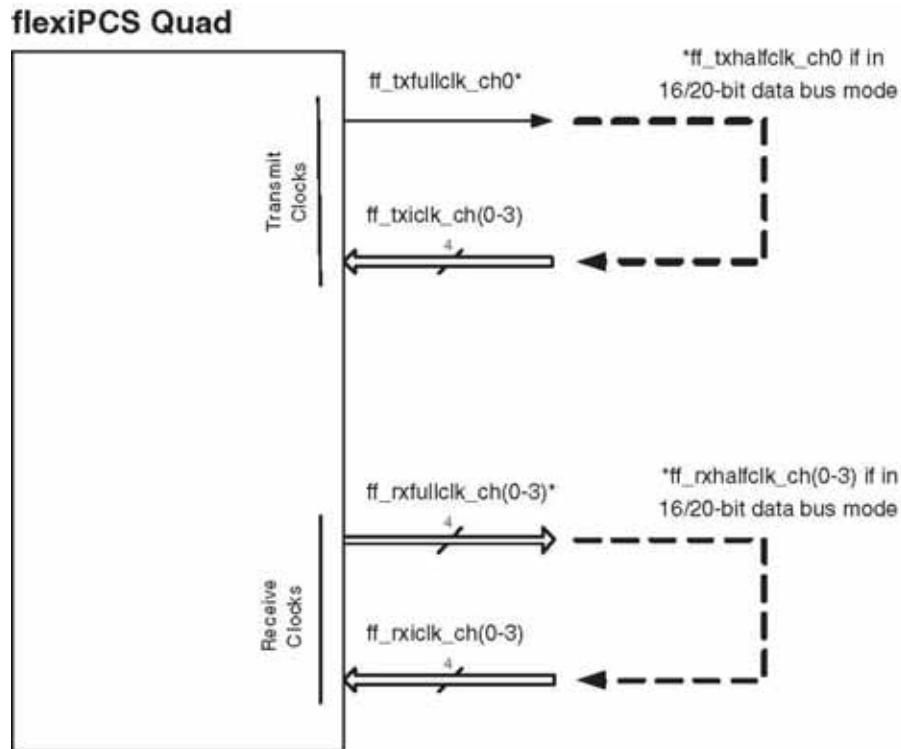
図8-17は、PCS内でCTCブロックを利用しないモードでの、単一チャネルの送受信両方向のクロックドメインを示します。送信側では、FPGAインターフェイスにおけるff_txclk_ch入力からロックした基準クロックまでのクロックドメイン転送は、FPGA送信インターフェイスFIFOで起こります。FPGA送信インターフェイスFIFOは、同一周波数の2クロック間の位相差を調整することが目的です。これらのFIFO(1チャネルあたり1つ)は周波数偏差を補うことはできません。受信側のクロックドメイン転送は、FPGA受信インターフェイスFIFOで受信再生クロックとff_rxclk_chの間で起こります。FPGA受信インターフェイスFIFOは、同一周波数の2クロック間の位相差を調整することが目的です。FPGA受信インターフェイスFIFO(1チャネルあたり1つ)は周波数偏差を補うことはできません。

図8-17 非CTCモードでのPCSクロックドメイン転送



同期インターフェイスを保証するためには、送信クロック入力と受信クロック入力の両方が、Generic 8b10b モードに設定されたPCSクワッドの基準クロック出力の1つからドライブされるべきです。図8-18は同期インターフェイスとなる、取り得る接続について図示します。

図B-18 CTCを用いないIPCSクワッドへの同期入力クロック



スペクトル拡散クロック(SSC)のサポート

LatticeECP2M SERDES/PCSにはスペクトル拡散ジェネレータがありませんが、スペクトル拡散データを受信することができます。

LatticeECP2MでのSSCサポートは同様の機能を必要とするすべてのプロトコルに適用されますが、リンク両端のポートは互いに600ppm以内に収まるレートでデータを伝送しなければなりません。これはビットレート用クロックソースに ± 300 ppmの偏差を許容する仕様です。変調レート30KHzからの33KHzの範囲で、データレートは定格の+0%から-0.5%まで変調ことができます。 ± 300 ppmの許容限界に加えて、データがSSCで変調されるとき、両ポートは同じビットレート・クロックを必要とします。

(PCI Expressでは) ルート・コンプレックスは基準クロックを拡散する責任があります。そしてエンドポイントは、TXを通してスペクトルを戻すために、それと同じクロックを用います。したがって別のRXREFCLKは必要ありません。

この支配的なアプリケーションはアドインカードです。これは、PCI ExpressコネクタのようにコネクタからのREFCLKを用いる必要はありませんが、同じSSCで受信し、送信しなければなりません。

シリアル・デジタルビデオとOOB (帯域外) 低速SERDES動作

レシーバCDRと付随するSERDES/PCSロジックをバイパスすることによって、低速度(250Mbps未満、Out-Of-Band=OOB信号)の入力にSERDES受信バッファを用いることができます。この機能はシリアル・デジタルビデオ用途など、高速データと低速データの両方からDCまでの伝送に、同一の端子で対応する必要があるアプリケーションに有用です。

LatticeECP2M SERDES/PCSは標準精細度シリアル・デジタルインターフェイスSD-SDI(143Mbps、177Mbps、270Mbps、360Mbps)、および高精細度シリアル・デジタルインターフェイスHD-SDI(1.485Gbps

と1.4835Gbps)をサポートします。これらは送受信ピンを共用することが必要です。考えられる実装方法の1つが図8-19で示されます。PCS/FPGAインターフェイスにおけるOOB信号ポートは、250Mbpsより低速の信号を入力するのに用いられます。

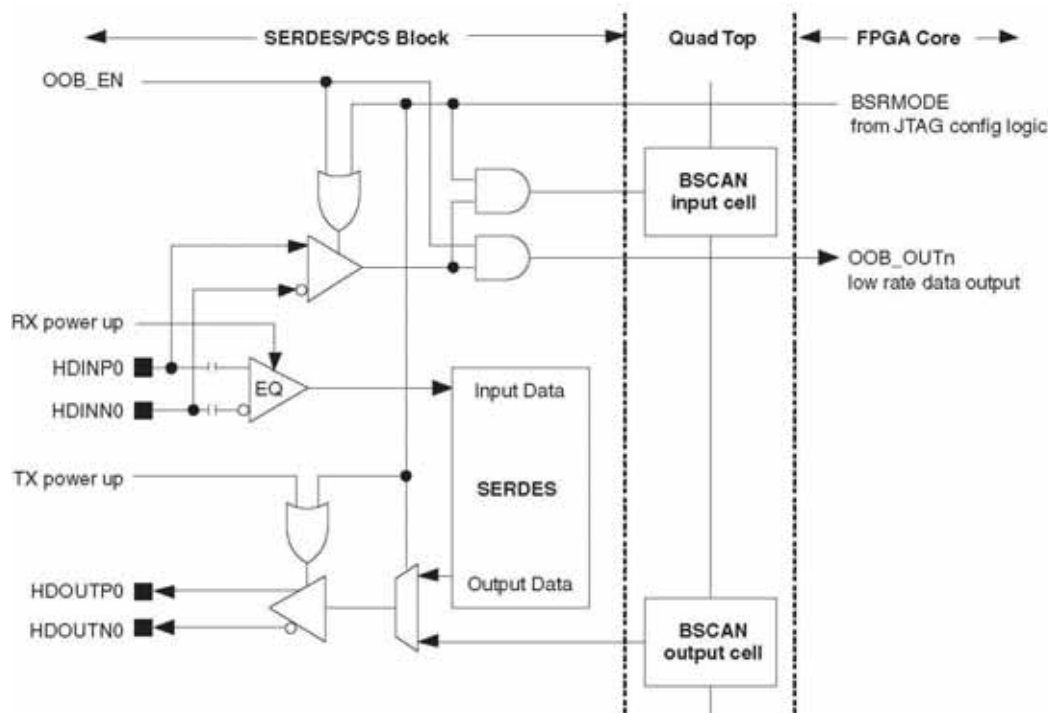
SD-SDIモードが選択されると、このポートが利用できます。OOB_OUT信号は、SERDES入力バッファからFPGAインターフェイスまで直接通され、また基準クロックにはロックしません(図8-3を参照してください)。

低速信号でSERDES入力バッファをドライブするとき、SERDES入力バッファはDCモードに設定されるべきです。これはIPexpress PCSコンフィグレーションGUIのRx I/O Couplingドロップダウン・ボックスにおいてDCを選択することによって、チャンネル単位で行われます(図8-25を参照して下さい)。

以上はシリアル・デジタルビデオに関する議論ですが、FPGAロジックでRx CDR(クロックデータ・リカバリ)を行い、Tx方向にデシメーションを必要とするような、低ビットレート・アプリケーションではいずれも同様な使い方をすることが意図されていることに留意して下さい。

入力BSCAN回路は、高速SERDESと並列に見え、入力データをデバイスの(クワッドでない)ほかの場所に位置する低速度デシリアライザに接続するために用いることができます。BSCANステートマシンに関与せずに入力BSCAN回路(1チャンネルあたり1つ)をオンするためのイネーブル信号が必要です。

図8-19 考えられるシリアル・デジタルビデオのサポート実装例



コンフィグレーションGUI

IPexpress™は、SERDESとPCSブロックを生成し、構成するために用いられます。設計者は、特定のクワッドやチャネルのSERDESプロトコル規格を選択する際に、GUIを用います。IPexpressはこのGUIから入力を得て、自動構成ファイル(.txtファイル)とHDLネットリストを生成します。HDLモデルはシミュレーションと合成フローで用いられます。自動構成ファイルは属性レベルの配置情報を含んでいて、シミュレーションとispLEVER bitgen（ビットファイル生成）プログラムの入力となります。設計に変更やアップデートが生じた場合は、IPexpressに戻って自動構成ファイルを生成し直すことを、強く推奨します。いくつかの例外時に限って、ユーザは自動構成ファイルを変更しても構いません。SERDESプロトコル規格用にSERDES/PCSブロックを生成するためIPexpressを用いるときのツールフローを、図8-20に示します。

プロジェクトが異なるディレクトリに保存されるとき、この自動構成ファイルもプロジェクトファイルと同じディレクトリに手動で移動（又はコピー）する必要があります。

図8-20 SERDES_PCS ispLEVERユーザフロー

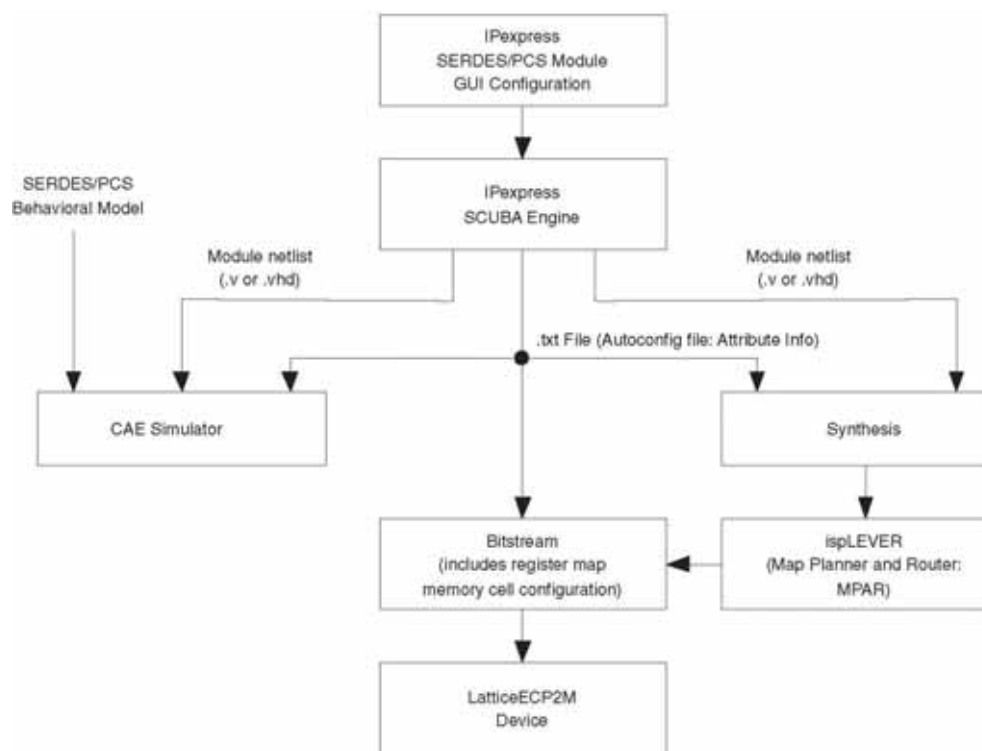
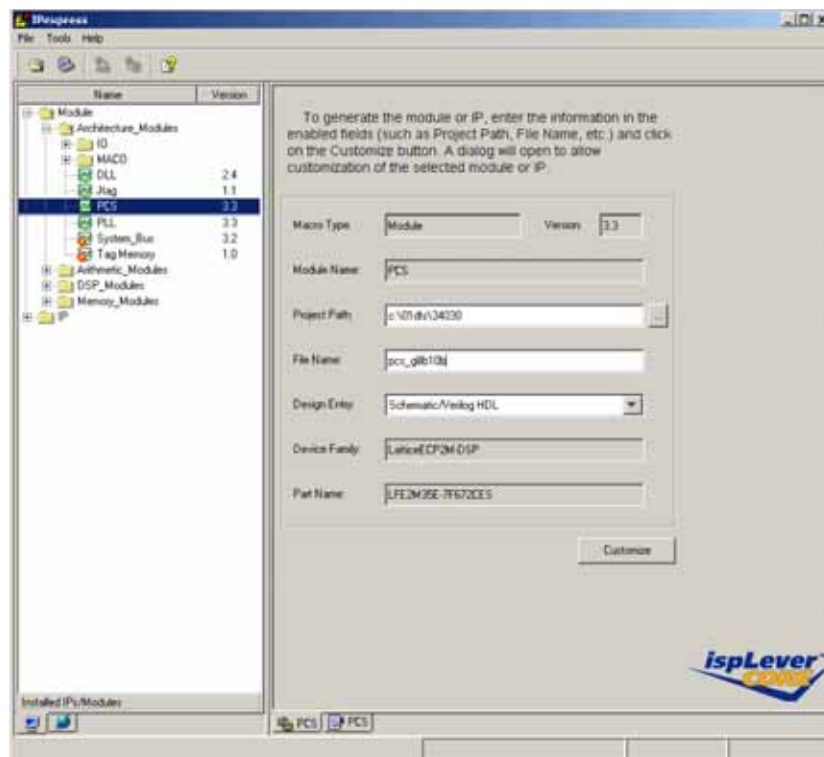


図8-21は、PCSがIPexpress GUIで選択された時のメインウィンドーを示します。

図8-21 IPexpress PCSメインウィンドー



クワッド・セットアップタブ

図8-22は、メインウィンドーでファイル名が入力され、Customizeボタンがチェックされた時のクワッド・セットアップタブを示します。このウィンドウで必要な最初のエントリは、プロトコルモードをQuad Based ModeかChannel Based Modeから選択することです。このスクリーン上の他のエントリは、チャンネルの選択とグループの選択です。図8-23から図8-28で示す5つのタブがあり、デフォルト値設定と共にユーザアクセス可能なすべての属性を示します。

図8-22 コンフィグレーションGUI、クワッド・セットアップタブ

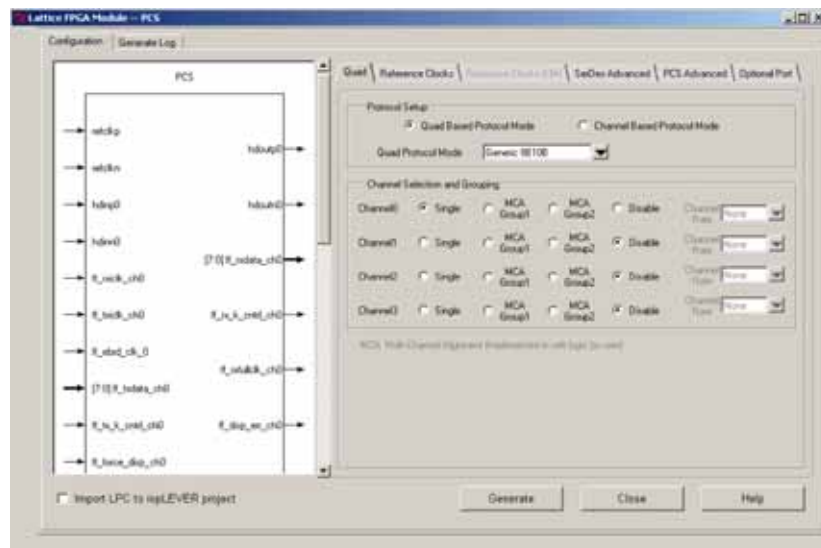


表8-9 SERDES_PCS GUI属性 - クワッドタブ・セットアップ

GUIテキスト	属性名	範 囲	デフォルト値
Protocol Setup		Quad Based Protocol Mode, Channel Based Protocol Mode	Quad Based Protocol Mode
Quad Protocol Mode	PROTOCOL	PCI Express, Gigabit Ethernet, Generic 8b10b, 10-bit SERDES Only, 8-bit SERDES Only, SD-SDI, HD-SDI ²	Generic 8b10b
Single	CH_MODE	チャンネルをイネーブル	Disable
MCA Group1 ³	CH_MODE	Multi-Channel Alignment Group 1	Disable
MCA Group2 ³	CH_MODE	Multi-Channel Alignment Group 2	Disable
Disable	CH_MODE	チャンネルをディセーブル	Disable
Channel Rate ¹	CH_MODE	Full Rate, Half Rate	Full Rate

1. Channel Rate選択はChannel Based Protocol Mode内のみに適用できます。
2. プロトコル属性名:PCI Express=PCIE、ギガビットイーサネット=GIGE、汎用8b10b=G8B10B、8ビットSERDES Only=8BSER、10ビットSERDES Only=10BSER、SD-SDI=SDSDI(HD-SDI=HDSDI)。
3. マルチチャンネル・アライメント (MCA) は送信レーン間スケュー調整のためのものです。レシーバのMCAはハードPCSでは提供していません。MCA Group1とMCA Group2は、ユーザがFPGAコアでMCAを構築できるように、チャンネルをCTCバイパスモードに設定します。マルチプロトコル・アプリケーションで、チャンネルの識別に2つのグループを指定できます。

基準クロック・セットアップタブ

このタブでは、TxとRx基準クロックソースの属性を選択します。クワッドベースの場合、Rx基準クロックソースとしてREFCLKかCORE_RXREFCLKの選択、Tx基準クロックソースとしてREFCLKかCORE_TXREFCLKを選択できます。クワッドベースでは、TxとRx共に全チャンネルが共通の基準クロックソースを用います。また、特定のデータレートに必要なクロックレートと乗数の設定を提供するツールがあります。選択したデータバス幅に対して、クワッドをコアとインターフェイスするために必要なクロックレートを自動計算して表示します。

図8-23 コンフィグレーションGUI、基準クロック・セットアップタブ

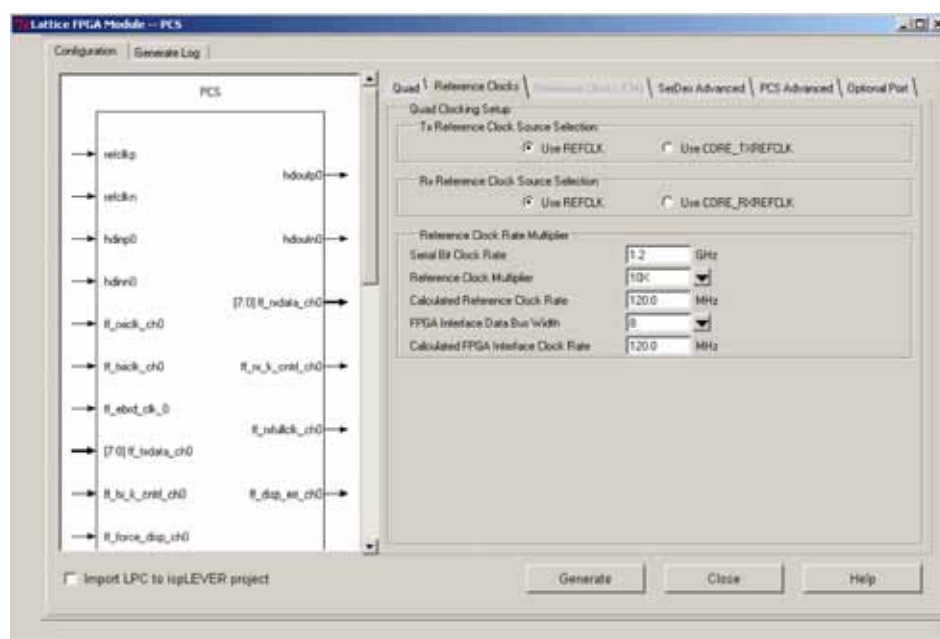


表8-10 SERDES_PCS GUI属性(LatticeECP2M) - 基準クロック・セットアップタブ

GUIテキスト	属性名	ボタン/ボックスタイプ	範囲	デフォルト値	コメント
Tx Reference Clock Source Selection	PLL_SRC	Radio	REFCLK, CORE_TXREFCLK	REFCLK	
Rx Reference Clock Source Selection	CH0_CDR_SRC, CH1_CDR_SRC, CH2_CDR_SRC, CH3_CDR_SRC	Radio	REFCLK, CORE_RXREFCLK	REFCLK	
Serial Bit Clock Rate (GHz)	DATARANGE ¹	Check Box	0.27 ~ 3.125	2.5	LOW: MEDLOW: MED: MEDHIGH: HIGH
Reference Clock Multiplier	CH0_REFCK_MULT, CH1_REFCK_MULT, CH2_REFCK_MULT, CH3_REFCK_MULT	Drop Down	表8-11参照	25X	
Calculated Reference Clock Rate (MHz) ²	CP:REFCLK_RATE	Text Box	---		編集不可
FPGA Interface Data Bus Width	CH0_DATA_WIDTH, CH1_DATA_WIDTH, CH2_DATA_WIDTH, CH3_DATA_WIDTH	Drop Down	表8-11参照	8	
Calculated FPGA Interface Clock Rate (MHz) ²	CP:FPGAINTCLK_RATE	Text Box	---		編集不可

1. DATARANGE ;

量産用デバイス ; Low 500Mbps, 500 Mbps < Medlow 1.0 Gbps, 1.0 Gbps < Med < 2.0 Gbps, 2.0 Gbps
Medhigh < 2.5 Gbps, 2.5 Gbps High 3.2 Gbps

エンジニアリング・サンプル ; Low 540 Mbps, 540 Mbps < Medlow 1.0 Gbps, 1.0 Gbps < Med < 2.0 Gbps, 2.0 Gbps
Medhigh < 2.5 Gbps, 2.5 Gbps High 3.2 Gbps

等価な制御レジスタとビットの詳細については表8-105を参照してください。

2. SERDES Only Modelは8ビットと10ビット共にリンク/コンマ・アラインと、8b10bエンコーダ/デコーダ、およびCTCをバイパスしますが、CDRはバイパスしません。

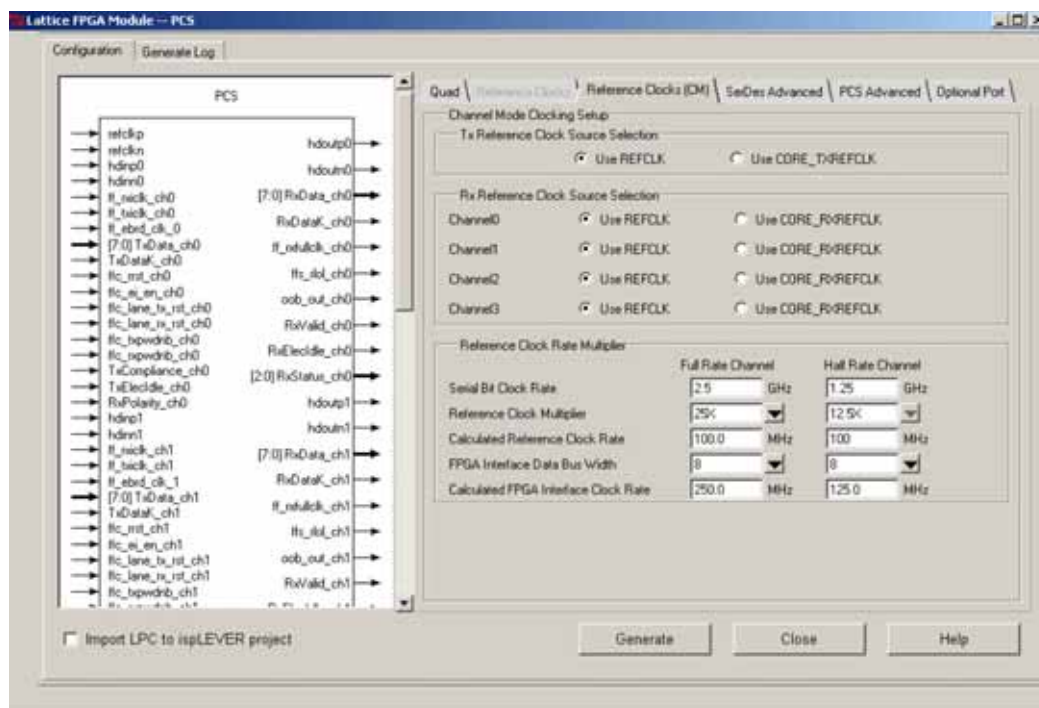
表8-11 プロトコルごとの基準クロック乗数とFPGAインターフェイス・データバス幅

プロトコル	基準クロック乗数	FPGAインターフェイス・データバス幅
PCI Express	20X, 25X	8, 16
GbE	10XH, 10X, 20X	8, 16
G8B10B	10XH, 10X, 20X	8, 16
10-bit SERDES Only	10XH, 10X, 20X	10, 20
8-bit SERDES Only	8HX, 8X, 16X	8, 16

基準クロック・セットアップタブ(チャンネルモード)

このタブでは、選択されたTx基準クロックは全チャンネルに共通ですが、チャンネルモードでのRx基準クロックは、チャンネルごとにREFCLKがCORE_RXREFCLKのどちらかを選択できます。

図8-24 コンフィグレーションGUI - 基準クロック・セットアップタブ(チャンネルモード)



ユーザがQuadタブでChannel Based Protocol Modeを選択して、いずれかのチャンネルをハーフレートモードとして設定すると、このタブはハーフレートモード・クロックデータを表示します。

SERDESアドバンスト・セットアップ

このタブは、全4チャンネル用のSERDESの送受信アドバンスト属性にアクセスするために使用されます。プリエンファシス、終端、差動出力電圧の選択などの送信用属性や、等化、終端、I/Oカップリングなどの受信信用属性が選択されます。また、SERDES送信クロックとPLLの属性も指定できます。

図8-25 コンフィグレーションGUI、SERDESアドバンスト・セットアップタブ

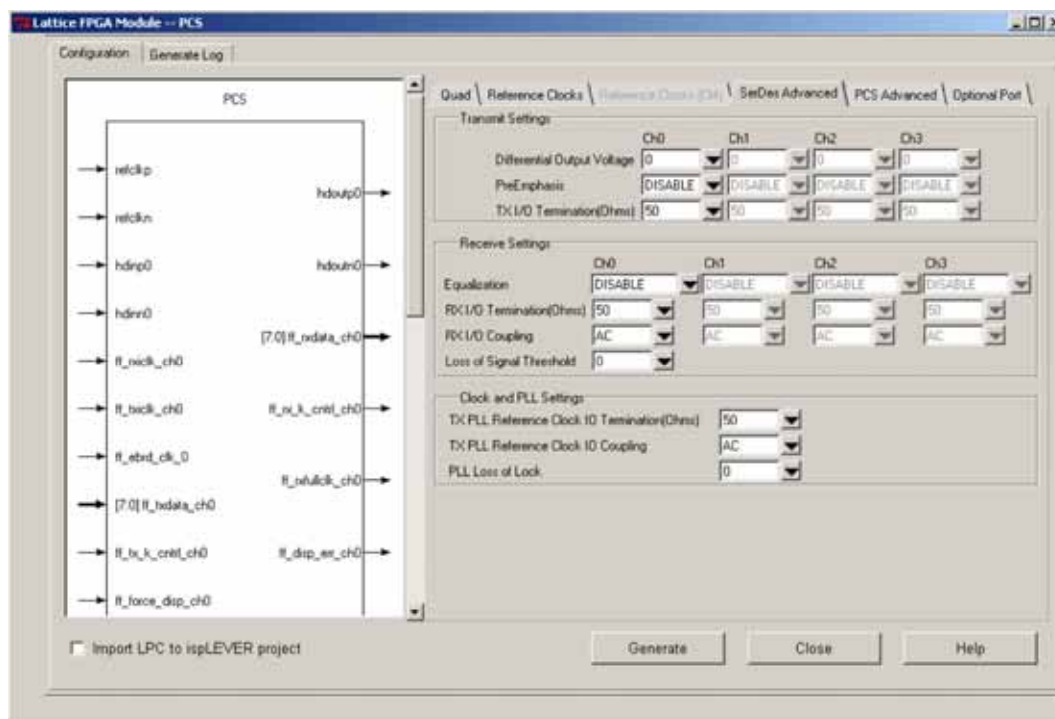


表8-12 SERDES_PCS GUI属性 (LatticeECP2M)、SERDESアドバンスド・セットアップタブ

GUIテキスト		属性名	ボタン/ボックスタイプ	範囲		デフォルト値
				PCI Express, GIGE	G8B10B, 8bSER 10bSER	
Differential Output Voltage (差動出力電圧振幅)	LatticeECP2M-35	CH0_TDRV_AMP CH1_TDRV_AMP CH2_TDRV_AMP CH3_TDRV_AMP	Drop Down	0(1040mV: default), 1(1280mV), 2 (1320mV), 3 (1360mV), 4 (640mV), 5 (760mV), 6 (870mV), 7 (990mV)		0
	その他の全デバイス	0(990mV: default), 1(1250mV), 2 (1300mV), 3 (1350mV), 4 (610mV), 5 (730mV), 6 (820mV), 7 (940mV)				
Pre-Emphasis (プリエンファシス)	LatticeECP2M-35	CH0_TX_PRE CH1_TX_PRE CH2_TX_PRE CH3_TX_PRE	Drop Down	Disable, 0 (0%), 1 (16%), 2 (36%), 3 (40%), 4 (44%), 5 (56%), 6 (80%)		DISABLE
	その他の全デバイス	Disable, 0 (0%), 1 (12%), 2 (26%), 3 (30%), 4 (33%), 5 (40%), 6 (53%)				
Tx I/O Termination (Ohms) ³ (送信I/O終端)		CH0_RTERM_TX CH1_RTERM_TX CH2_RTERM_TX CH3_RTERM_TX	Drop Down	50, 75, 5K		50
Equalization ¹ (等化)		CH0_RX_EQ CH1_RX_EQ CH2_RX_EQ CH3_RX_EQ	Drop Down	Mid_High, Long_High	Mid_Low, Mid_Med Mid_High, Long_Low Long_Med, Long_High	DISABLE
Rx I/O Termination (Ohms) ³ (受信I/O終端)		CH0_RTERM_RX CH1_RTERM_RX CH2_RTERM_RX CH3_RTERM_RX	Drop Down	50, 60, 75, High		50
Rx I/O Coupling (レシーバI/O結合)		CH0_RX_DCC CH1_RX_DCC CH2_RX_DCC CH3_RX_DCC	Drop Down	AC, DC		AC 2
Loss of Signal Threshold (LOS検出閾値)		LOS_THRESHOLD	Drop Down	0 (default), 1 (+10%), 2 (+15%), 3 (+25%), 4 (-10%), 5 (-15%), 6 (-25%), 7 (-30%)		0
Tx PLL Reference Clock I/O Termination (Ohms) ³ (送信PLL基準クロック入力の終端)		PLL_TERM	Drop Down	50, 2K		50
Tx PLL Reference Clock I/O Coupling (送信PLL基準クロック入力の結合)		PLL_DCC	Drop Down	AC, DC		AC
PLL Loss of Lock (PLLロックはずれ検出)		PLL_LOL_SET	Drop Down	Lock	Unlock	0
				0 (+/-600ppmx) 1 (+/-300ppm) 2 (+/-1500ppm) 3 (+/-4000ppm)	0 (+/-1200ppm) 1 (+/-2000ppm) 2 (+/-2200ppm) 3 (+/-6000ppm)	

1. 詳細については付録Dを参照してください。
2. 内部のオンチップAC結合のコンデンサーTyp.値は5pFです。
3. 終端抵抗とその使い方
Rx I/O終端:

50: 現状ではSMTPE以外のプロトコルはすべて50Ω終端抵抗を用います。

60: 柔軟性の目的だけに提供します。

75: SMPTEは75Ω終端抵抗を用います。

High: PCI Expressのレシーバ検出用など。

Tx I/O終端:

50: 現状ではSMTPE以外のプロトコルはすべて50Ω終端抵抗を用います。

75: SMPTEは75Ω終端抵抗を用います。

5K: PCI Expressの電氣的アイドルやレシーバ検出のような場合。

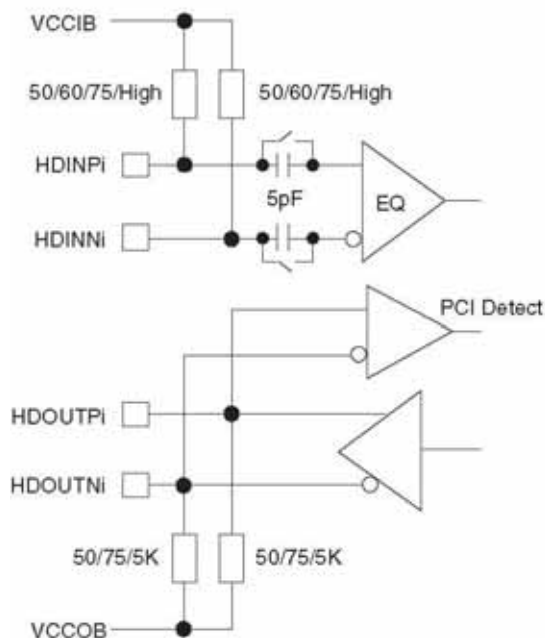
Tx PLL終端:

50: PCB上に50Ω終端抵抗がない場合

2k: PCB上に50Ω終端抵抗がない場合

高速I/O終端トポロジを図8-26に示します。

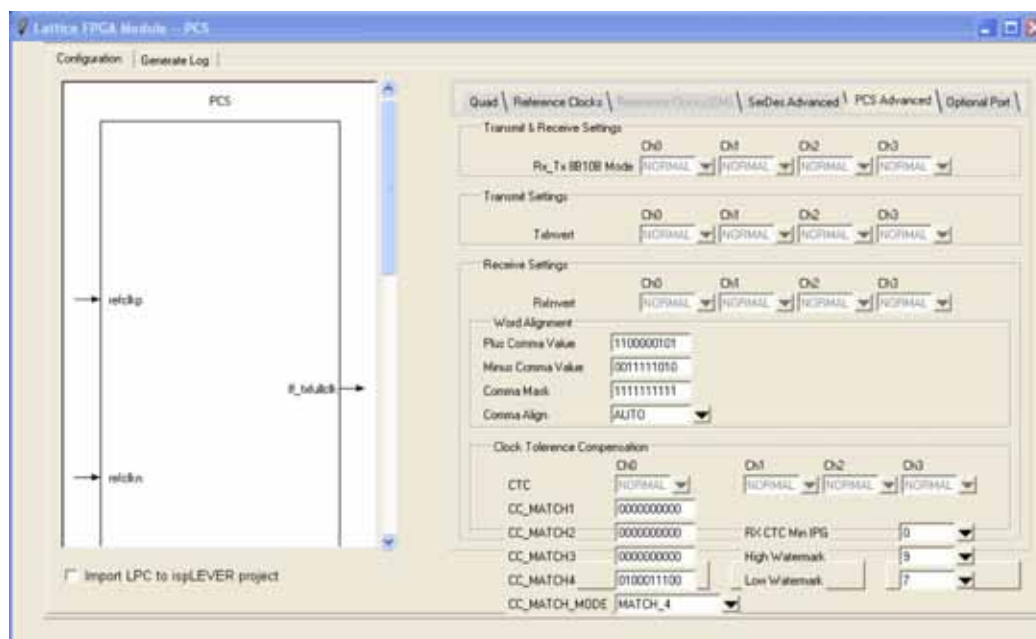
図8-26 高速I/O終端



PCSアドバンスト・セットアップ

このタブは、PCSの送受信全4チャンネルのアドバンスト属性にアクセスするために使用されます。各TxとRxチャンネルの極性や動作モード(例えば、8b10b)を個別に選択することができます。さらに、コンマ値やコンママスク、コンマアラインなどのワードアライメント指定や、CTCブロック用パラメータを設定するためにも用いられます。

図B-27 コンフィグレーションGUI - PCSのアドバンスト・セットアップタブ



表B-13 SERDES/PCS GUI 'Range' - PCSのアドバンスト・セットアップタブ

GUIテキスト	属性名	8b SERDES Only	10b SERDES Only	G8B10 B	PCI Express	GIGE	SD-SDI	HD-SDI	Default
TX Invert	CHx_TX_SB	Normal, Invert							Normal
RX Invert	CHx_RX_SB								Normal
RX_TX 8b10b Mode	CHx_8B10B	Bypass		Normal		Bypass		Normal	
Plus Comma Value	COMMA_A ⁴	N/A	注1			N/A		1100000101	
Minus Comma Value	COMMA_B ⁴							0011111010	
Comma Mask	COMMA_M							1111111111	
Comma Align ²	CHx_COMMA_ALIGN	Bypass	Auto, Dynamic, Bypass	Auto, Dynamic c	Auto	Bypass		Auto	
CTC ³	CHx_CTC_BYP	Bypass	Normal, Bypass		Normal		Bypass		Normal
CC_MATCH1	CC_MATCH1	N/A	注3				N/A	0000000000	
CC_MATCH2	CC_MATCH2							0000000000	
CC_MATCH3	CC_MATCH3							0100011100	
CC_MATCH4	CC_MATCH4							0100011100	
CC_MATCH_MODE	CC_MATCH_MODE		MATCH_3_4		MATCH_4		MATCH_4		
RX CTC Min IFG	RX CTC Min IFG		0, 1, 2, 3						0
High Watermark	CCHMARK		0, 1, 2, ..., 14, 15						9
Low Watermark	CCLMARK								7

1. 詳細な情報はこのドキュメントの“ワードアラインメント”セクションを参照してください。

2. 表8-7を参照してください。
3. 詳細な情報はこのドキュメントの“クロックトレランス補償”セクションを参照してください。
4. 定義上、COMMA_AとCOMM_Bは正と負のランニング・ディスパリティがある1組の8b10bコード化された制御キャラクタです。例えば、一つのデザインでBC(K28.5)とFD(K29.7)をCOMMA_AとCOMM_Bとして用いることはできません。8ビットモードと16ビットモードでのCOMM_AとCOMM_Bの使用法は全く同じです。ユーザはls_syncを得るためには規則に従う必要があります。例えば、1GbEがIDLEとなる(ワードアライメントと同期ステートマシン)にはK28.5+D5.6がD16.2が必要です。

オプションのセットアップ

このタブは、チャンネルあたりのダイナミック・ロジック反転とダイナミックな外部リンクステートマシン機能を選択するのに用いられます。さらにユーザはSCI、エラーレポート、PLLの4分周クロック、およびループバック機能をイネーブルすることができます。

図8-28 コンフィグレーションGUI- オプションのセットアップタブ

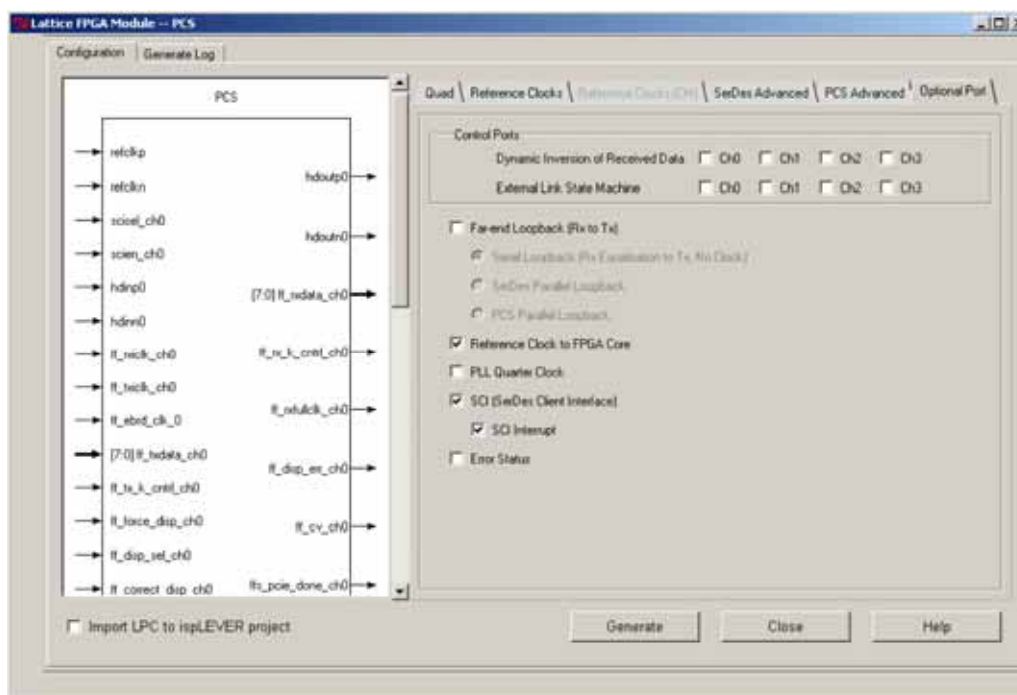


表8-14 Tab5、SERDES_PCS GUI属性(LatticeECP2M) - オプションのセットアップタブ

GUIテキスト	属性名	ボタン/ボックスタイプ	範囲	デフォルト値
Dynamic Inversion of Receive Data		Check Box	TRUE, FALSE	FALSE
External Link State Machine		Check Box	TRUE, FALSE	FALSE
Loopback (Rx to Tx)		Check Box	TRUE, FALSE	FALSE
Loopback Type	OS_SSLB OS_SPLBPORTS OS_PCSLBPORTS	Radio Box	Serial Loopback, SERDES Parallel Loopback, PCS Parallel Loopback	Serial Loopback
Reference Clock to FPGA Core	OS_REFCK2CORE	Check Box	TRUE, FALSE	FALSE
PLL Quarter Clock	OS_PLLQCLKPORTS	Check Box	TRUE, FALSE	FALSE
SCI		Check Box	TRUE, FALSE	FALSE

SCI Interrupt	OS_INT_ALL	Check Box	TRUE, FALSE	FALSE
Error Status		Check Box	TRUE, FALSE	FALSE

コンフィグレーション（自動構成）ファイル記述

IPexpressは属性レベルのマッピング情報を含むこのファイルを生成します。自動構成ファイルはファイルタイプ拡張子として ".txt" を用います。

以下にファイル例を示します。

```
# This file is used by the simulation model as well as the ispLEVER bitstream
# generation process to automatically initialize the PCSC quad to the mode
# selected in the IPexpress. This file is expected to be modified by the
# end user to adjust the PCSC quad to the final design requirements.
DEVICE_NAME "LFE2M35E"
PROTOCOL "G8B10B"
CH0_MODE "SINGLE"
CH1_MODE "DISABLE"
CH2_MODE "DISABLE"
CH3_MODE "DISABLE"
PLL_SRC "REFCLK"
DATA_RANGE "HIGH"
CH0_CDR_SRC "REFCLK"
CH0_DATA_WIDTH "8"
CH0_REFCK_MULT "10X"
#REFCLK_RATE 250.0
#FPGAINTCCLK_RATE 250.0
CH0_TDRV_AMP "0"
CH0_TX_PRE "DISABLE"
CH0_RTERM_TX "50"
CH0_RX_EQ "DISABLE"
CH0_RTERM_RX "50"
CH0_RX_DCC "AC"
LOS_THRESHOLD "0"
PLL_TERM "50"
PLL_DCC "AC"
PLL_LOL_SET "0"
CH0_TX_SB "NORMAL"
CH0_RX_SB "NORMAL"
CH0_8B10B "NORMAL"
COMMA_A "1100000101"
COMMA_B "0011111010"
COMMA_M "1111111111"
CH0_COMMA_ALIGN "AUTO"
CH0_CTC_BYP "BYPASS"
CC_MATCH1 "0000000000"
CC_MATCH2 "0000000000"
CC_MATCH3 "0100011100"
CC_MATCH4 "0100011100"
CC_MATCH_MODE "MATCH_4"
CC_MIN_IPG "0"
CCHMARK "4"
CCLMARK "4"
OS_REFCK2CORE "0"
OS_PLLQCLKPORTS "0"
```

モード特有の制御/ステータス信号

モードに依存する制御/ステータス信号名は、選択されたPCSモードで決定されます。表8-15は24ビット・デ

ータバスの使い方を説明します。表8-16はすべてのIPexpress GUIで選択可能なPCSモードの制御/ステータス信号のポート名を示します。

表8-15 データバスの用途

データバス	I/O	信号名	記 述
送受信データバス； 8ビットバス			
ff_rxddata_ch0[7:0]	O	受信データバス	チャンネルごと、FPGAへの8ビット受信並列データバス
ff_rxddata_ch0[11:8]	O	表8-16参照	モード特有の受信ステータスビット
ff_rxddata_ch0[23:12]	O		16ビットモード用予約
ff_txdata_ch0[7:0]	I	送信データバス	チャンネルごと、FPGAからの8ビット送信並列データバス
ff_txdata_ch0[11:8]	I	表8-16参照	モード特有の送信制御ビット
ff_txdata_ch0[23:12]	I		16ビットモード用予約
送受信データバス； 16ビットバス			
ff_rxddata_ch0[7:0]	O	受信データバス	チャンネルごと、FPGAへの第一8bデータ用受信並列データバス
ff_rxddata_ch0[11:8]	O		モード特有の第一8bデータ用の受信ステータスビット
ff_rxddata_ch0[19:12]	O	受信データバス	チャンネルごと、FPGAへの第二8bデータ用受信並列データバス
ff_rxddata_ch0[23:20]	O		モード特有の第二8bデータ用の受信ステータスビット
ff_txdata_ch0[7:0]	I	送信データバス	チャンネルごと、FPGAからの第一8bデータ用送信並列データバス
ff_txdata_ch0[11:9]	I		モード特有の第一8bデータ用の送信制御ビット
ff_txdata_ch0[19:12]	I	送信データバス	チャンネルごと、FPGAからの第二8bデータ用送信並列データバス
ff_txdata_ch0[23:20]	I		モード特有の第二8bデータ用の送信制御ビット

注: チャンネル0を示します。他のすべてのチャンネルが同様に割り当てられます。

表8-16 モード特有の制御ステータス信号名

受信データステータス信号、8ビットバス

モード	ff_rxddata_ch0[8]	ff_rxddata_ch0[9]	ff_rxddata_ch0[10]	ff_rxddata_ch0[11]
GbE, OBSAI, CPRI	ff_rx_k_cntrl_ch0	ff_disp_err_ch0	ff_cv_ch0	ff_rx_even_ch0
FC, SRIO, G8B10B	ff_rx_k_cntrl_ch0	ff_disp_err_ch0	ff_cv_ch0	NA
XAUI	ff_rxc_ch0	ff_disp_err_ch0	ff_cv_ch0	NA
PCI Express	ff_rx_k_cntrl_ch0			

送信データ制御信号、8ビットバス

モード	ff_txdata_ch0[8]	ff_txdata_ch0[9]	ff_txdata_ch0[10]	ff_txdata_ch0[11]
GbE, OBSAI, CPRI	ff_tx_k_cntrl_ch0	NA	ff_xmit_ch0	ff_correct_disp_ch0
FC, SRIO	ff_tx_k_cntrl_ch0	ff_force_disp_ch0	ff_disp_sel_ch0	NA
XAUI	ff_txc_ch0	NA	NA	NA
PCI Express	ff_tx_k_cntrl_ch0	ff_force_disp_ch0	ff_disp_sel_ch0	ff_pci_ei_en_ch0
G8B10B	ff_tx_k_cntrl_ch0	ff_force_disp_ch0	ff_disp_sel_ch0	ff_correct_disp_ch0

モード特有の制御/ステータス信号記述

表8-17はモード特有の制御/ステータス信号について説明します。

表8-17 制御信号とそれらの機能

信号名	記 述
送信制御信号	
ff_tx_k_cntrl_ch[3:0]	チャンネルごと、Highアクティブの制御キャラクタ・インジケータ。
ff_force_disp_ch[3:0]	チャンネルごと、Highアクティブの信号で、ff_disp_sel_ch(0-3)FPGAインターフェイス入力からのディスパリティ値を受け入れるようPCSに指示する。
ff_disp_sel_ch[3:0]	チャンネルごとで、ディスパリティ値はFPGAロジックから与えられる。 ff_force_disp_ch(0-3)がhighのときに有効。
ff_correct_disp_ch[3:0]	パケット間ギャップが負のディスパリティ・ステートで始まることを保証するために使用される。
受信ステータス信号	
ff_rx_k_cntrl_ch[3:0]	チャンネルごと、Highアクティブの制御キャラクタ・インジケータ。
ff_disp_error_detect_ch[3:0]	チャンネルごと、PCSからドライブされるhighアクティブの信号で、関連するデータにディスパリティ・エラーが検出されたことを示す。
ff_cv_detect_ch[3:0]	チャンネルごと、Highアクティブのコードバイ奥レーション検出。

1. 詳細な情報はこのドキュメントの "ギガビット・イーサネットモードのLatticeECP2M PCS" セクションを参照してください。

Generic 8b10b(G8B10B)モードにおけるデータと制御/ステータスビットのマッピング例:

```

8-bit mode
.FF_TX_D_0_0(ff_txdata_ch0[0]),
.FF_TX_D_0_1(ff_txdata_ch0[1]),
.FF_TX_D_0_2(ff_txdata_ch0[2]),
.FF_TX_D_0_3(ff_txdata_ch0[3]),
.FF_TX_D_0_4(ff_txdata_ch0[4]),
.FF_TX_D_0_5(ff_txdata_ch0[5]),
.FF_TX_D_0_6(ff_txdata_ch0[6]),
.FF_TX_D_0_7(ff_txdata_ch0[7]),
.FF_TX_D_0_8(ff_tx_k_cntrl_ch0[0]),
.FF_TX_D_0_9(ff_force_disp_ch0[0]),
.FF_TX_D_0_10(ff_disp_sel_ch0[0]),
.FF_TX_D_0_11(ff_correct_disp_ch0[0]),
.FF_TX_D_0_12(fpvc_vlo),
.FF_TX_D_0_13(fpvc_vlo),
.FF_TX_D_0_14(fpvc_vlo),
.FF_TX_D_0_15(fpvc_vlo),
.FF_TX_D_0_16(fpvc_vlo),
.FF_TX_D_0_17(fpvc_vlo),
.FF_TX_D_0_18(fpvc_vlo),
.FF_TX_D_0_19(fpvc_vlo),
.FF_TX_D_0_20(fpvc_vlo),
.FF_TX_D_0_21(fpvc_vlo),
.FF_TX_D_0_22(fpvc_vlo),
.FF_TX_D_0_23(fpvc_vlo),
.FF_RX_D_0_0(ff_rxdata_ch0[0]),
.FF_RX_D_0_1(ff_rxdata_ch0[1]),
.FF_RX_D_0_2(ff_rxdata_ch0[2]),
.FF_RX_D_0_3(ff_rxdata_ch0[3]),
.FF_RX_D_0_4(ff_rxdata_ch0[4]),
.FF_RX_D_0_5(ff_rxdata_ch0[5]),
.FF_RX_D_0_6(ff_rxdata_ch0[6]),
.FF_RX_D_0_7(ff_rxdata_ch0[7]),
.FF_RX_D_0_8(ff_rx_k_cntrl_ch0[0]),
.FF_RX_D_0_9(ff_disp_err_ch0[0]),
.FF_RX_D_0_10(ff_cv_ch0[0]),
.FF_RX_D_0_11(fpvc_vlo),),
.FF_RX_D_0_12(fpvc_vlo),
.FF_RX_D_0_13(fpvc_vlo),
.FF_RX_D_0_14(fpvc_vlo),
.FF_RX_D_0_15(fpvc_vlo),

```

```

.FF_RX_D_0_16(fp_sc_vlo),
.FF_RX_D_0_17(fp_sc_vlo),
.FF_RX_D_0_18(fp_sc_vlo),
.FF_RX_D_0_19(fp_sc_vlo),
.FF_RX_D_0_20(fp_sc_vlo),
.FF_RX_D_0_21(fp_sc_vlo),,
.FF_RX_D_0_22(fp_sc_vlo),
.FF_RX_D_0_23(fp_sc_vlo),

16-bit mode
.FF_TX_D_0_0(ff_txdata_ch0[0]),
.FF_TX_D_0_1(ff_txdata_ch0[1]),
.FF_TX_D_0_2(ff_txdata_ch0[2]),
.FF_TX_D_0_3(ff_txdata_ch0[3]),
.FF_TX_D_0_4(ff_txdata_ch0[4]),
.FF_TX_D_0_5(ff_txdata_ch0[5]),
.FF_TX_D_0_6(ff_txdata_ch0[6]),
.FF_TX_D_0_7(ff_txdata_ch0[7]),
.FF_TX_D_0_8(ff_tx_k_cntrl_ch0[0]),
.FF_TX_D_0_9(ff_force_disp_ch0[0]),
.FF_TX_D_0_10(ff_disp_sel_ch0[0]),
.FF_TX_D_0_11(ff_correct_disp_ch0[0]),
.FF_TX_D_0_12(ff_txdata_ch0[8]),
.FF_TX_D_0_13(ff_txdata_ch0[9]),
.FF_TX_D_0_14(ff_txdata_ch0[10]),
.FF_TX_D_0_15(ff_txdata_ch0[11]),
.FF_TX_D_0_16(ff_txdata_ch0[12]),
.FF_TX_D_0_17(ff_txdata_ch0[13]),
.FF_TX_D_0_18(ff_txdata_ch0[14]),
.FF_TX_D_0_19(ff_txdata_ch0[15]),
.FF_TX_D_0_20(ff_tx_k_cntrl_ch0[1]),
.FF_TX_D_0_21(ff_force_disp_ch0[1]),
.FF_TX_D_0_22(ff_disp_sel_ch0[1]),
.FF_TX_D_0_23(ff_correct_disp_ch0[1]),

.FF_RX_D_0_0(ff_rxdata_ch0[0]),
.FF_RX_D_0_1(ff_rxdata_ch0[1]),
.FF_RX_D_0_2(ff_rxdata_ch0[2]),
.FF_RX_D_0_3(ff_rxdata_ch0[3]),
.FF_RX_D_0_4(ff_rxdata_ch0[4]),
.FF_RX_D_0_5(ff_rxdata_ch0[5]),
.FF_RX_D_0_6(ff_rxdata_ch0[6]),
.FF_RX_D_0_7(ff_rxdata_ch0[7]),
.FF_RX_D_0_8(ff_rx_k_cntrl_ch0[0]),
.FF_RX_D_0_9(ff_disp_err_ch0[0]),
.FF_RX_D_0_10(ff_cv_ch0[0]),
.FF_RX_D_0_11(),
.FF_RX_D_0_12(ff_rxdata_ch0[8]),
.FF_RX_D_0_13(ff_rxdata_ch0[9]),
.FF_RX_D_0_14(ff_rxdata_ch0[10]),
.FF_RX_D_0_15(ff_rxdata_ch0[11]),
.FF_RX_D_0_16(ff_rxdata_ch0[12]),
.FF_RX_D_0_17(ff_rxdata_ch0[13]),
.FF_RX_D_0_18(ff_rxdata_ch0[14]),
.FF_RX_D_0_19(ff_rxdata_ch0[15]),
.FF_RX_D_0_20(ff_rx_k_cntrl_ch0[1]),
.FF_RX_D_0_21(ff_disp_err_ch0[1]),
.FF_RX_D_0_22(ff_cv_ch0[1]),
.FF_RX_D_0_23(),

```

ギガビット・イーサネットモードのLatticeECP2M PCS

ギガビット・イーサネット(1000BASE-X)のアイドル挿入

アイドルパターンの挿入がクロック補償と自動ネゴシエーションには必要です。自動ネゴシエーションはFPGAロジックで行われます。このモジュールは自動ネゴシエーションの間、自動的に /I2/ シンボルを受信データストリームの中に挿入します。自動ネゴシエーション中、リンクパートナーは連続して /C1/ と /C2/ の

オーダーセットを送信するでしょう。クロック補償回路はこれらのオーダーセットを削除せず、 $1/2$ オーダーセットのみを挿入するか、または削除するように構成されています。クロック補償回路におけるオーバーランとアンダーランを防ぐため、挿入/削除する機会をクロック補償回路に与えるように定期的に $1/2$ オーダーセットを挿入しなければなりません。

自動ネゴシエーションを実行している間、このモジュールは2048クロックサイクル毎に8回の $1/2$ オーダーセット(それぞれ2バイト)シーケンスを挿入します。このモジュールは8b10bデコーダの後にあるため、この動作はどのようなランニング・ディスパリティ誤りを生起することはありません。GMIIインターフェイスは、自動ネゴシエーションの間、RxステートマシンによってIDLEステートに置かれるので、 $1/2$ オーダーセットがFPGA受信インターフェイスに送られることはありません。自動ネゴシエーションがいったん完了すると、受信データのいかなるエラーをも防ぐために、 $1/2$ 挿入はディセーブルされます。

このステートマシンは自動ネゴシエーションの間だけアクティブであることに留意してください。自動ネゴシエーション・ステートマシンとGbE受信ステートマシンはソフトロジックで実装されています。GbE受信ステートマシンは自動ネゴシエーション・ステートマシンからの (Txデータバス上の) 信号ff_xmit_ch[3:0]に依存します。この信号は、特に自動ネゴシエーションの後には比較的スタティックです。GbE受信ステートマシンからは、FPGAロジック用のrx_even_ch[3:0]信号が受信データバス上に出力されます。

表8-18 主要なGbE IDLEステートマシン制御 / ステータス信号

モジュール信号	方向	記 述
ff_xmit_ch[3:0]	入力	FPGAロジックの自動ネゴシエーション・ステートマシンから
rx_even_ch[3:0]	出力	rx_evenインジケータ入力(GbEリンクステートマシンから) rx_evenインジケータ出力(FPGAロジック内の自動ネゴシエーションと受信ステートマシンへ)

ギガビット・イーサネットのアイドル挿入とff_correct_disp_ch[3:0]信号の使い方

QuadPCSの送信側でff_correct_disp_ch[3:0]信号が用いられ、パケット間ギャップが負のディスパリティ状態で始まることを確実にします。イーサネットフレームのサイズとデータ内容によって、その終わりではトランスミッターのディスパリティ状態は正も負もあり得ることに留意してください。しかしながら、QuadPCSのFPGAソフトロジック側からは、QuadPCSトランスミッターの現在のディスパリティ状態は未知です。これはff_correct_disp_ch[3:0]信号が役割を果たすところです。パケット間ギャップに入るとき、ff_correct_disp_ch[3:0]信号がクロックの1周期間アサートされると、現在のディスパリティが正である場合は、QuadPCSトランスミッターに対して、送信データストリームにIDLE1オーダーセットを挿入するように強制します。しかしながら、現在のディスパリティが負の場合、送信データストリームは何も変更されません。

QuadPCSのFPGAソフトロジック側からは、パケット間ギャップは通常以下の通りIDLE2オーダーセットの連続伝送で特徴付けられます。

```
ff_tx_k_cntrl_ch3:0=1、ff_txdata=0xBC、ff_tx_k_cntrl_ch3:0=0、ff_txdata=0x50
```

PCSチャネルでは、IDLE2は現在のディスパリティが保持されることに留意してください。IDLE1は、現在のディスパリティ状態が反転されるべきであることを意味します。したがって、パケット間ギャップが負のディスパリティ状態で始まることを確実にすることは有り得ます。パケット間ギャップ前のディスパリティ状態が負であるなら、パケット間ギャップギャップの間、連続したIDLE2のストリームを送信します。パケット間ギャップ前のディスパリティ状態が正であるなら、単一IDLE1がIDLE2の連続ストリームの後に送信されます。

QuadPCSのFPGAソフトロジック側では、パケット間ギャップは常にIDLE2と共にQuadPCSにドライブさ

れます。パケット間ギャップが最初に始まる時、k_cntrl=0、data=0x50、そしてff_correct_disp_ch[3:0]信号がクロックの1周期アサートされます。必要であれば、QuadPCSはこのIDLE2をIDLE1に変換します。パケット間ギャップの残りにおいては、QuadPCSにはIDLE2がドライブされるべきで、そしてff_correct_disparity_chx信号はネゲートされたままです。

PCI ExpressモードのLatticeECP2M PCS

IPexpressでPCI Expressモードに設定されたLatticeECP2Mクワッドには、仕様によって必要とされるレシーバ検出などの電氣的機能をイネーブルするために、FPGAインターフェイスに追加ポートがあります。表8-19はPCI Expressモードに特有のポートについて説明します。

表8-19 PCI Express モード特有のポート

信 号	方向	クラス	記 述
ffs_pcie_done_ch[3:0]	○	チャンネル	1 = 遠端レシーバの検出完了 0 = 遠端レシーバ検出の結果が不成功（未完了）
ffs_pcie_con_ch[3:0]	○	チャンネル	1 = 遠端レシーバの検出 0 = 遠端レシーバの不検出
ffc_pcie_det_en_ch[3:0]	┃	チャンネル	FPGAロジック(ユーザロジック)はPCI Expressレシーバ検出動作の要求をSERDESブロックに通知する。 1 = PCI Expressレシーバ検出をイネーブル 0 = 通常動作
ffc_pcie_ct_ch[3:0]	┃	チャンネル	1 = トランスミッターに遠端レシーバ検出を要求 0 = 通常データ動作
ffc_ei_en_ch[3:0]	┃	チャンネル	SERDESトランスミッターから電氣的アイドルの伝送を制御。 1 = 電氣的アイドル状態を出力するようにSERDESトランスミッターを強制 0 = 通常動作

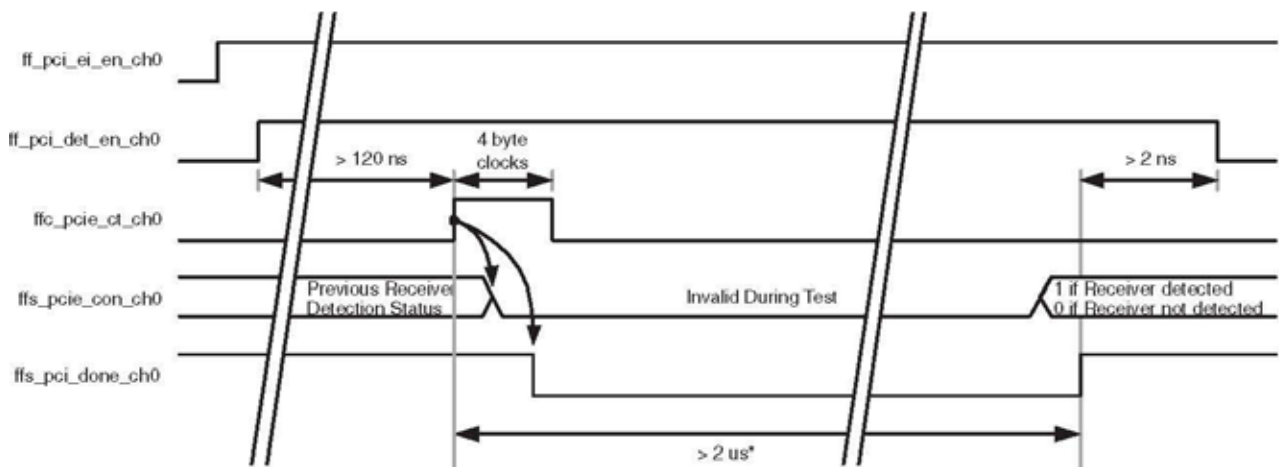
レシーバ検出

図8-29はレシーバ検出シーケンスを示します。

クワッドの各チャンネルでレシーバ検出テストを独立して実行することができます。レシーバ検出テストを始める前に、ff_pci_ei_en_ch入力をHighに設定することによって、トランスミッターを電氣的アイドルにしなければなりません。ffc_pci_det_en_chをHighにすることによって、tx_elec_idleがHighに設定された120ns後にレシーバ検出テストを始めることができます。これはドライバ終端をハイインピーダンスに設定し、またハイインピーダンス・ドライバ終端で差動の両出力をVCCOBにプルすることによって、対応するSERDES送信バッファをレシーバ検出モードにします。

SERDES送信バッファをレシーバ検出ステートに設定するには最大120nsかかります。その後チャンネルのffc_pcie_ct_ch入力を4バイト（ワード）クロック周期Highにドライブすることによって、レシーバ検出テストを開始することができます。対応するチャンネルのffc_pcie_done_chは非同期にクリアされます。レシーバ検出テストが終わるのに十分な時間が経過した後(送信側の時定数で決まる)、ffs_pcie_done_chレシーバ検出ステータスポートはHighになり、ffs_pcie_con_chポートでレシーバ検出ステータスをモニターすることができます。その時ffs_pcie_con_chポートがHighであればレシーバはそのチャンネルで検出され、ポートがLowならそのチャンネルでレシーバは全く検出されていません。レシーバ検出テストがいったん完全すると、ff_pci_ei_en_chをネゲートすることができます。

図8-29 PCI Express モード、レシーバ検出シーケンス(チャンネル0の例)



PCI Express ビーコンのサポート

このセクションではLatticeECP2M PCSがビーコン検出と送信をどのようにサポートすることができるかをハイライトします。PCI Expressのビーコン検出の要件は、PCSによるビーコン送信とビーコン検出のサポートで与えられます。

- ・ ビーコン検出要件 (PCI Express Base Specification, Rev 1.0a, Chapter 4, pp.209-210)
 - L2(P2)ステートから抜けるにはビーコンが必要です。
 - ビーコンは周期的な任意データのDCバランスがとれている信号で、2ns(500Mhz)以上と16us(30Khz)未満のいくつかのパルス幅を含む必要があります。
 - パルス間の最大時間は16us未満でなければなりません。
 - DCバランスは32us未満に回復される必要があります。

- 500ns以上のパルス幅に関しては、出力ピーコン電圧レベルは $V_{TX-DIFFp-p}$ (800mVから1200mV)から6dB低くしなければなりません。
- 500ns以下のパルス幅に関しては、出力ピーコン電圧レベルは $V_{TX-DIFFp-p}$ 以下で、かつ3.5dB低いレベル以上でなければなりません。
- ・ PCSピーコン検出サポート
 - 信号ロス(LOS)閾値検出回路は、指定された電圧レベルが受信バッファに存在するかどうかを探知します。これは`ffs_rlos_lo_ch(0-3)`信号によって示されます。
 - PCI Express電氣的アイドル検出と(パワーステートP2での)PCI Expressピーコン検出にこの設定を用いることができます。
 - リモート送信デバイスは $V_{TX-DIFFpp}$ から6dB低いピーコン出力電圧(即ち201mV)を持つことができます。この信号を検出することができれば、ピーコンが検出されたと言えます。
- ・ PCSピーコン送信のサポート
 - K28.5(IDLE)キャラクタ(5連続の1と続く5連続の0)を送出すると、2nsの周期的なパルス(1.0UI=400psで、5倍すると2 ns)を発生することになります。これは下側の要件を満たします。このとき出力ピーコン電圧レベルは $V_{TX-DIFFp-p}$ であり得ます。これは有効なピーコン送信です。

PCSループバック・モード

LatticeECP2Mファミリ・デバイスは3つのループバック・モードを提供し、PCS/FPGAロジック・インターフェイスの制御信号によって制御されます。これはSERDES外部/ボード・インターフェイスと内部PCS/FPGAロジック・インターフェイスのテストに便利です。3つのループバック・モードは受信したデータを送信データパスに戻すために用意されています。ループバック・モードは、組み込みSERDESやPCSロジックのチェックと共に、高速シリアルSERDESのパッケージピン接続をチェックする為に有用です。

シリアル・ループバック・モード

シリアルの受信クロック/データを、CDRやデシリアライザを通さずに、送信バッファにループバックします。この機能はラティス社内でのテスト目的のために用意されていますが、ユーザも使用できます。

IPexpress GUIのシリアル・ループバック・オプションを選択すると、LB_CTL[3:0]のみが0010に設定されます(表8-82を参照してください)。

また、TDRV_DAT_SEL[1:0]レジスタビットは、シリアル・ループバック・モードをイネーブルするためにSCIを介して11に設定します。また、LB_CTL[3:0]レジスタビットもSCIを通してアクセスできます。

SERDESパラレル・ループバック・モード

PCSロジックを通さずに、パラレル受信データを送信データパスにループバックします。各チャネル用に適切な`ffc_sb_pfifo_lp_ch(0-3)`を 1 に設定することによって、個別にSERDESパラレル・ループバックを選択することができます。

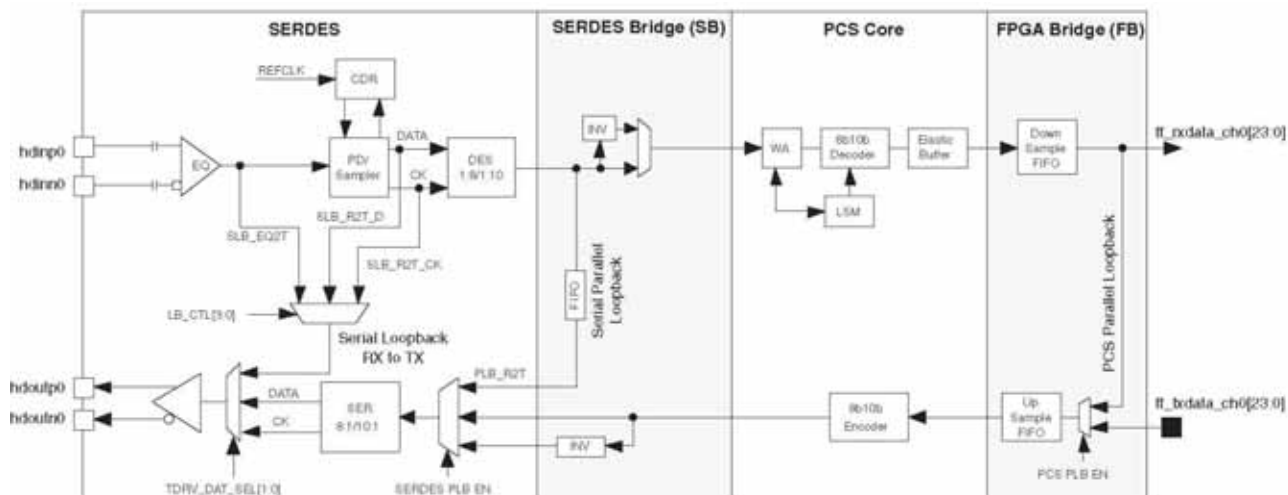
PCSパラレル・ループバック・モード

PCS/FPGAインターフェイス内を通さずに、パラレル受信データを送信データパスにループバックします。SERDES HDINパッケージピンからのシリアルデータ入力は、SERDES受信ロジックを通り、パラレルデータに変換され、PCS受信ロジックパス全体を通り、次いでPCS送信ロジックパス全体を通してループバックされます。その後SERDESトランスミッターによってシリアルデータに再変換され、HDOUT SERDESパッケージピンに送られます。各チャネル用に適切な`ffc_fb_loopback_ch(0-3)`ポートを1に設定することによって、

個別にPCSパラレル・ループバックを選択することができます

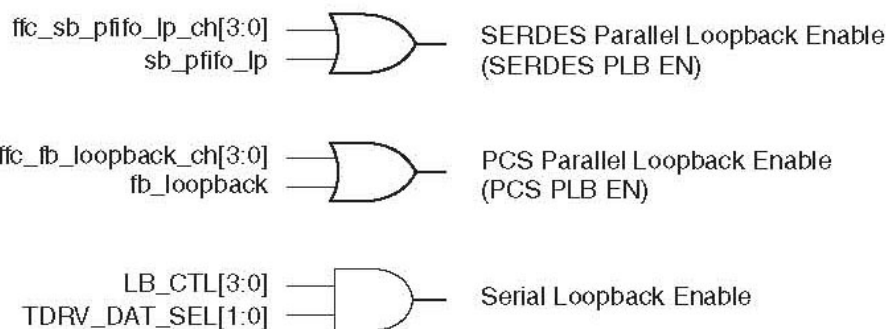
図8-30は単一チャンネル用の3つのループバックモードを図示します。

図8-30 3つのループバック・モード



上述した2つのパラレル・ループバック・モードは、FPGAコアからの制御信号だけでなく制御レジスタビットでも設定できます。制御レジスタビットがループバック・モードに設定されるとき、データパスはFPGAコアからの制御信号でループバック・モードとノーマル・データフローを切り換えることができます。図8-31はこのロジックについて説明します。詳細なレジスタ設定は表8-82と表8-83を参照してください。

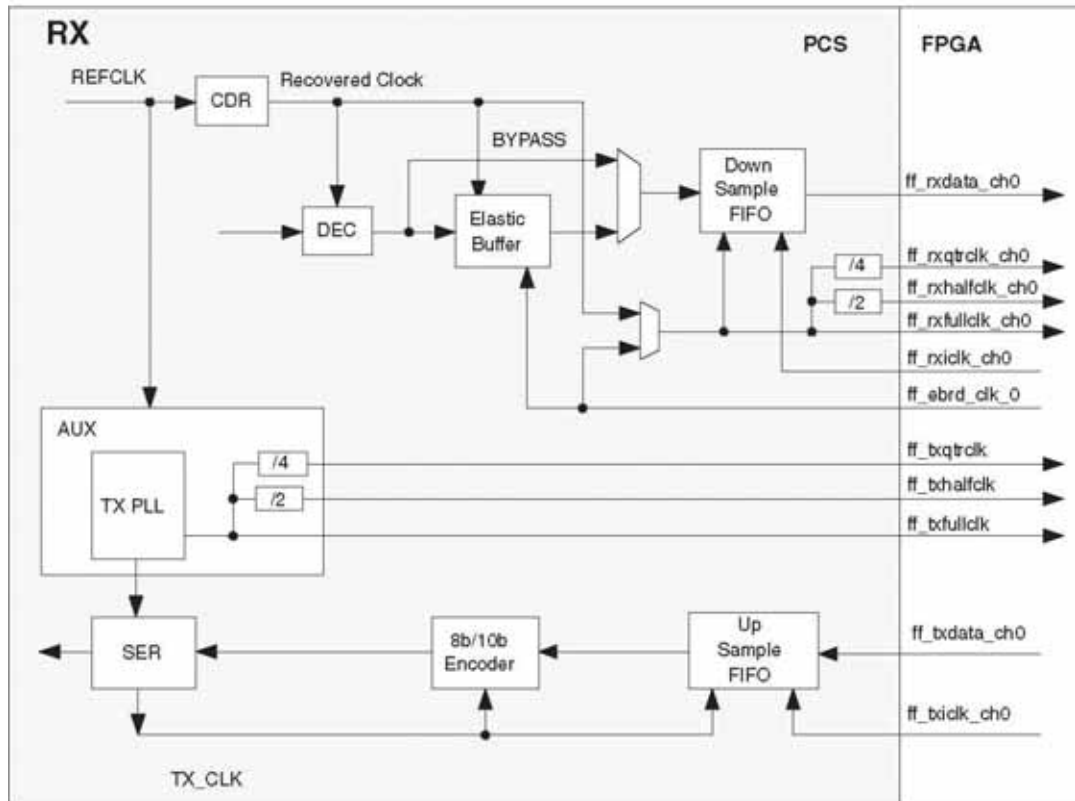
図8-31 ループバック・イネーブル信号



FPGAインターフェイス・クロックの使い方

図8-32は概念図で、PCSコアとFPGAブリッジの後半ステージと、PCSとFPGAの境界にまたがる主要なクロックを示します。

図8-32 概念的なPCS/FPGA クロック・インターフェイス図



上図とこのセクションの各図中、接尾語 “i” がインデックス[0:3]、すなわち各チャネルの1つを示すことに留意してください。

クロックが細くなったりグリッチが出ることを防ぐロジックは、クロック・マルチプレクサのいずれも持っていません。レジスタビットへのライトによってクロック・マルチプレクサへのセクタのどれかが変更された場合、そのマルチプレクスされたクロックによって動作するロジックをリセットすることはソフトウェア・エージェントの要件です。

PCSは15本のクロックを出力します。クワッドあたり3本の送信クロックと、4チャネル分で12本の受信クロックがあります。3本の送信クロックは、フルレート、ハーフレート、およびクォータレート・クロックで、全てTx PLLから生成されます。フルレートとハーフレート送信クロックは、FPGA中央のクロック・マルチプレクサへの専用配線で直接接続される必要があります。また受信チャネルにもそれぞれ3本のクロック(フル/ハーフ/クォータ・レート)があります。FPGAロジックに必要なに応じてローカル(セカンダリ)かグローバル(プライマリ)クロックとして15本のクロックをすべて用いることができます。ギアリング(クワッドベースのみで選択可能)が2:1のモードであるとき、2分周クロックが使用されます。

送信クロックはアップサンプルFIFO(場合によっては位相シフトFIFO)のライトポートで用いられます。2本の受信クロックの1つはダウンサンプルFIFOのリードクロックに接続されます。他方のクロックはエラステック・バッファFIFOのリードポートと、(場合により)潜在的にダウンサンプルFIFOのライトポートのクロ

ックになります。

エラスティック・バッファとアップサンプルFIFOがバイパスされるかどうかと、8b10bモードか16b20bモードかによって、4つのケースが有り得ます。（図8-33～36の各図で）アクティブなパスはハイライトされています。また何本の、どのようなクロックツリーが必要であるかも示されています。一般的にユーザがより好むいくつかのモードがあります。

このセクションはサポートされる4つの異なるケースの動作について、表8-20で概説しています。

表8-20 SERDES/PCS クワッドとFPGA コア間インターフェイスの6つのケース

インターフェイス	データ幅	Rx CTC FIFO	Rx位相シフト/ダウン サンプルFIFO	Tx位相シフト/アップサ ンプルFIFO
Case I-a ¹	8b10b	Yes	Yes	Yes
Case I-b ¹	8b10b	Bypass	Yes	Yes
Case II-a ¹	16b20b	Yes	Yes	Yes
Case II-b ¹	16b20b	Bypass	Yes	Yes

1. すべての場合でTx位相シフト(アップサンプル)FIFOはバイパスされません。SERDES/PCSブロック内部のデータパス幅は8/10ビットであり、バイトクロックはSERDESラインレートの1/10で動作します。例えばSERDESラインレートが3.125Gbpsである場合、バイトクロックは312.5MHzです。

2対1ギアリング

FPGAグローバル・クロックツリーの性能を保証するために、2.5Gbpsより速いSERDESラインレートでは16/20ビットのインターフェイスを用いることを推奨します。このインターフェイスでは、FPGAインターフェイス・クロックはバイトクロック周波数の半分です。

バイトクロック周波数の半分で動作する16/20ビット幅インターフェイスはすべてのSERDESラインレートで用いることができますが、SERDESラインレートが低く(2.5Gbps以下)、許容することができる場合は8/10ビット幅インターフェイスが好ましいでしょう。というのは、これがFPGAコア内のIPに最も効率的な実装という結果をもたらすためです。

6つのインターフェイスケースを決定するためのマトリクスを表8-21で説明しています。

表8-21 6インターフェイス決定のためのマトリクス

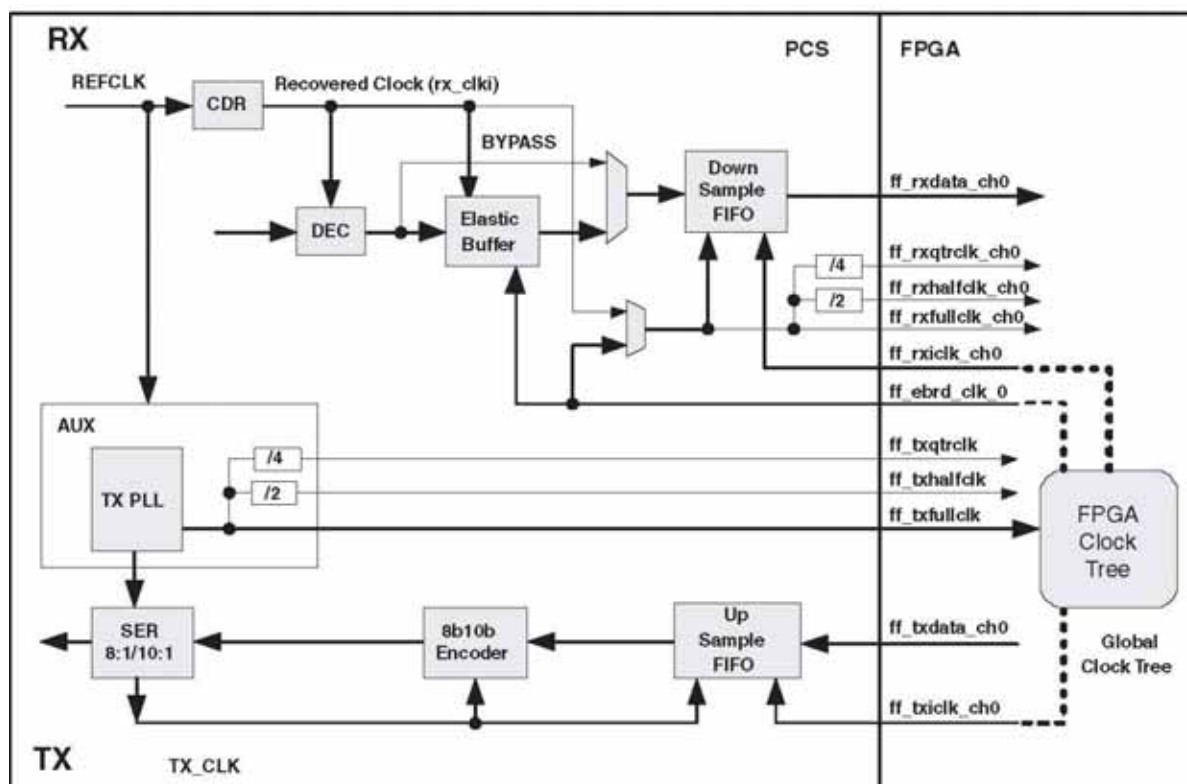
SERDESラインレート	データパス幅	MCA必要性	CTC必要性	インターフェイス・ケース
2.5Gbpsかそれ以下	8/10ビット (1:1ギアリング)	なし, シングルチャネル・リンク	あり	Case I_a ¹
			なし	Case I_b ²
		あり, マルチチャネル・リンク	バイパス必須(CTCなし)	Case I_b ³
3.2Gbpsかそれ以下	16/20ビット (2:1ギアリング)	なし, シングルチャネル・リンク	あり	Case II_a ⁴
			なし	Case II_b ⁵
		あり, マルチチャネル・リンク	バイパス必須(CTCなし)	Case II_b ⁶

1. このケースは、クワッド内でCTCを必要とする2.5Gbpsかそれ以下のラインレート(8/10ビット幅インターフェイス)の単一チャネルリンクを意図します。相互に ± 300 ppm以内にある基準クロックソースがリンクの両端にあるときはCTCが必要です。コア内のIPがRx位相シフトFIFOを必要とするなら、Case I_aが使用されます。コア内のIPがRx位相シフトFIFOを必要としないなら、Case I_bが使用されます。

2. このケースは、クワッド内でCTCを必要としない2.5Gbpsかそれ以下のラインレート(8/10ビット幅インターフェイス)の単一チャネルリンクを意図します。リンクの両端が同じ基準クロックソースに接続されるとき、CTCは不要です。しばしばこれは同一回路基板上のチップ間リンクにみられるケースです。基準クロック間は0ppmの違いなのでCTCは不要で、バイパスすることができます。また、この機能がコア内のIPによって実行されるとき、クワッドにCTCは不要です。
3. このケースは、クワッド内でCTCを必要とする2.5Gbpsかそれ以下のラインレート(8/10ビット幅インターフェイス)のマルチチャネル・リンクを意図します。マルチチャネル・アライメント(MCA)はコア内のIPで対応しなければなりませんが、クワッド内でMCAをしなければならないという規定はありません。CTCの前でMCAをしなければならないので、MCAが必要ときにはクワッド内のCTC FIFOをバイパスしなければなりません。必要であればMCAとCTCの両方をコアのIPによって行われます。
4. このケースは、クワッドとFPGAコア間で2:1ギアリングが必要(16/20ビット幅インターフェイス)で、3.2Gbpsかそれ以下のラインレートの単一チャネルリンクを意図します。CTCはクワッドに含まれています。相互に ± 300 ppm以内にある基準クロックソースがリンクの両端にあるときはCTCが必要です。
5. このケースは、クワッドとFPGAコア間で2:1ギアリングが必要(16/20ビット幅インターフェイス)で、3.2Gbpsかそれ以下のラインレートの単一チャネルリンクを意図します。CTCはクワッドに含まれていません。リンクの両端が同じ基準クロックソースに接続されるとき、CTCは不要です。しばしばこれは同一回路基板上のチップ間リンクにみられます。基準クロック間は0ppmの違いなのでCTCは不要で、バイパスすることができます。また、この機能がコア内のIPによって実行されるとき、クワッドでCTCは不要です。
6. このケースはクワッドとFPGAコア間で2:1ギアリングが必要(16/20ビット幅インターフェイス)で、3.2Gbpsかそれ以下のラインレートのマルチチャネル・リンクを意図します。MCAはコア内のIPで対応しなければなりませんが、クワッド内でMCAをしなければならないという規定はありません。CTCの前でMCAをしなければならないので、MCAが必要ときにはクワッド内のCTC FIFOをバイパスしなければなりません。必要であればMCAとCTCの両方をコアのIPによって行われます。

ケースI_a; 8/10ビット、Elastic Buffer(EB)、およびDown-Sample(DS) FIFOはバイパスしない

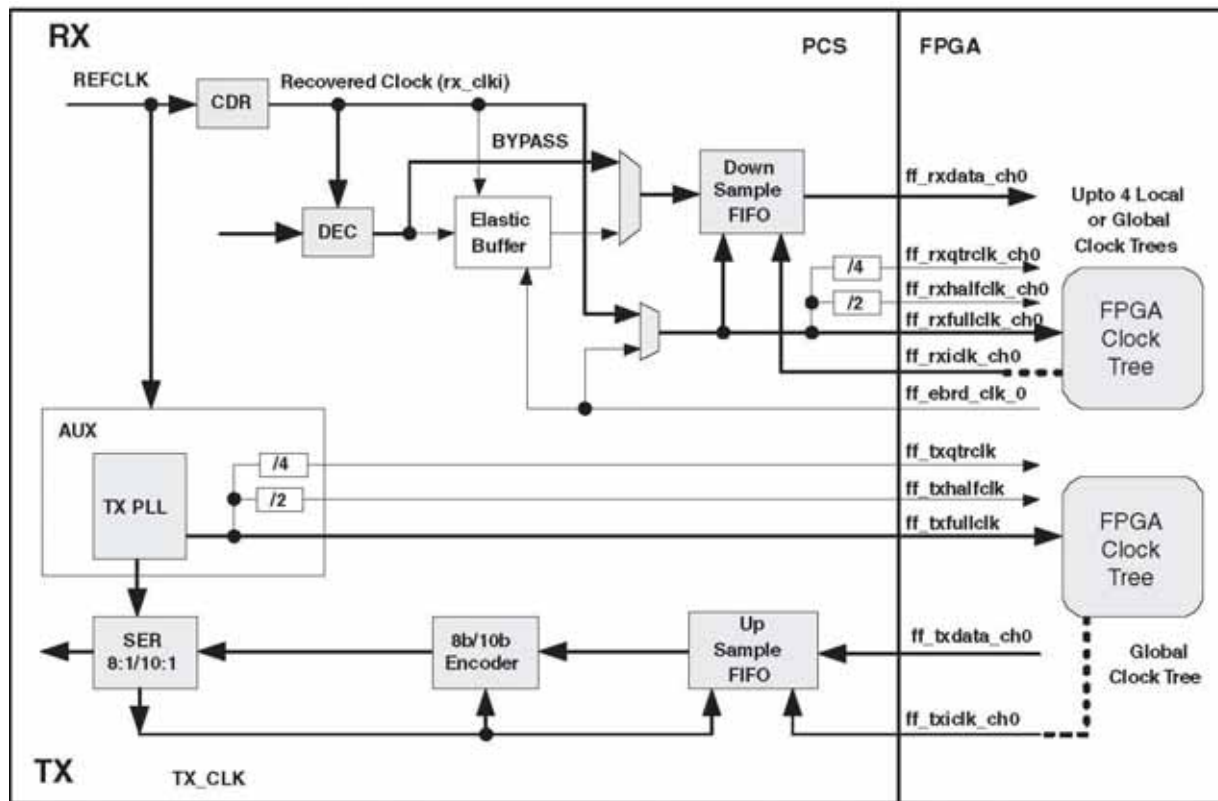
図8-33 8b10b、EB、およびDS FIFOはバイパスしない



1. アップサンプルFIFOはこの場合だけ位相シフトFIFOとして機能します。
2. ダウンサンプルFIFOはこの場合だけ位相シフトFIFOとして機能します。
3. Tx PLLからのクワッドレベル・フルレートクロック(ff_tx_f_clk)は、FPGA中央のクロック・マルチプレクサに直接アクセスできます。これは比較的性能の高いパスです。デバイス中央のクロック・マルチプレクサからのグローバル・クロックツリーは、FPGA内のユーザのインタフェース・ロジックのクロックに用いられます。クロックツリーの幾つかのノードがFPGA送信入力クロック(ff_txi_clki)やCIBクロック入力を介してチャンネル毎(ff_ebrd_clki)のエラスティック・バッファFIFO用リードクロックと、FPGA受信入力クロック(ff_rxi_clki)に接続されます。。これは恐らく最もよく見られる単一チャンネルのユースケースです。

ケースI_b: 8/10ビット、EB FIFOはバイパス

図8-34 8b10b、EB FIFOはバイパス

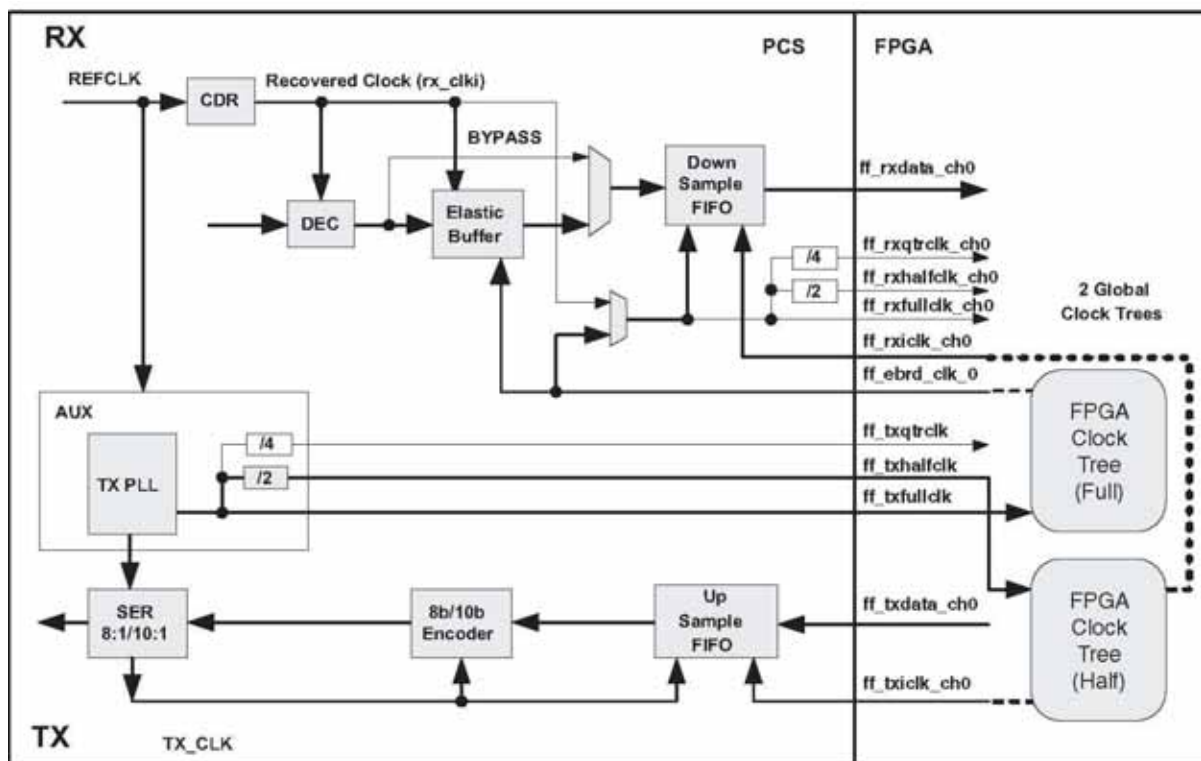


1. アップサンプルFIFOはこの場合だけ位相シフトFIFOとして機能します。
2. ダウンサンプルFIFOはこの場合だけ位相シフトFIFOとして機能します。
3. 前の場合と同様にTx FPGAチャンネル入力クロックは、FPGA中央のクロック・マルチプレクサへ直接接続されるフルレート送信FPGA出力クロックでドライブされるクロックツリーを用います。エラスティック・バッファがいったんバイパスされると、再生受信クロックは、ダウンサンプルFIFOのライトポートを制御する必要があります。各チャンネルの再生受信クロックは、別々の(クワッドあたり最大4本の)ローカル或いはグローバル・クロックツリーをドライブする必要があるかもしれません。そしてクロックツリーは、ダウンサンプルFIFOのリードポートを制御するようにFPGA受信クロック入力をドライブします。この場合エラスティック・バッファFIFOをバイパスする理由は、たぶんFPGAコアでMCAを行うためです。それはエラスティック・バッファを用いるCTCをFPGAコアで行うことを暗示します。CTC FIFOは受信再生クロックがマスター受信再生クロックでライトすることができます。CTC FIFOのリードはTxクロックツリーを通してのTxクロックを

用いて行われます。

ケースII a: 16/20ビット、EB、およびDS FIFOはバイパスしない

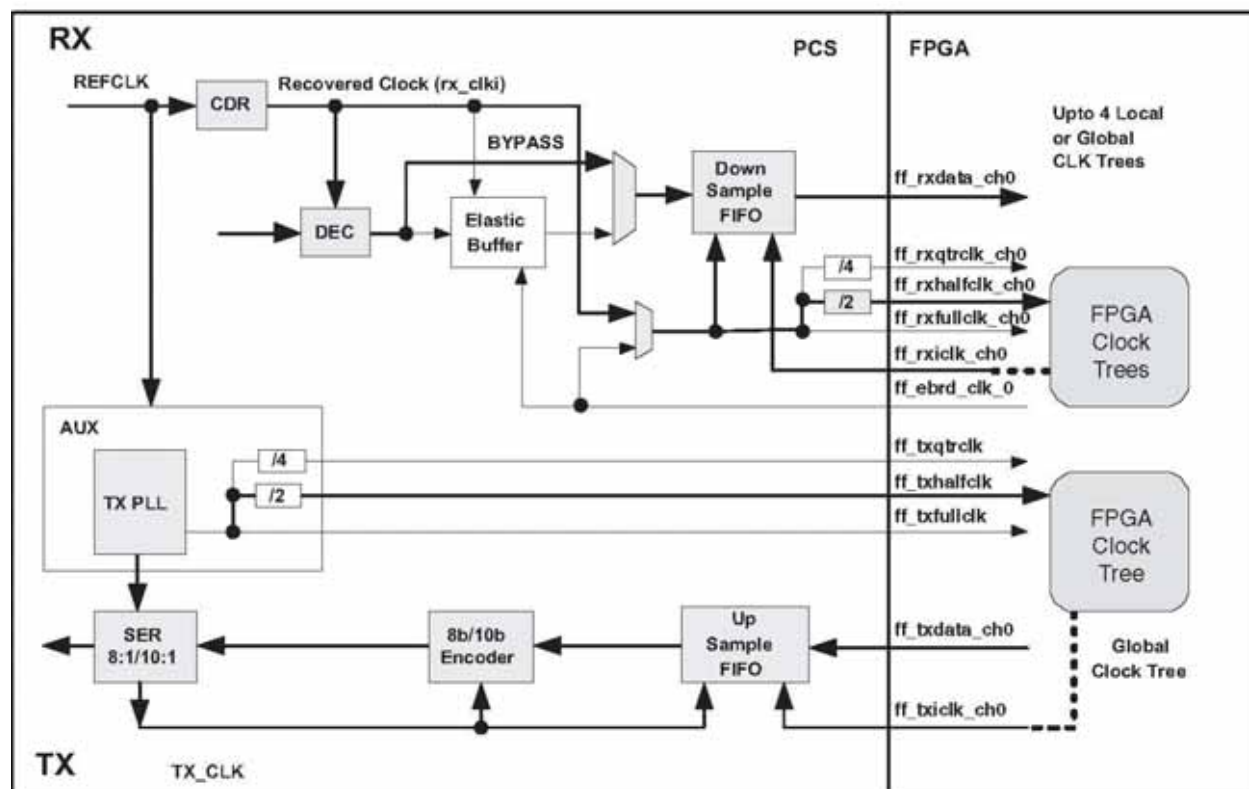
図8-35 16/20ビット、EB、およびDS FIFOはバイパスしない



1. アップサンプルFIFOはこの場合位相シフトFIFOおよびアップサンプルFIFOとして機能します。
2. ダウンサンプルFIFOはこの場合位相シフトFIFOおよびダウンサンプルFIFOとして機能します。
3. FPGAがフルレート・バイト周波数に動作が間に合わない場合、これは非常に一般的な単一チャネルのユースケースです。2本のクロックツリーが必要です。これらのクロックツリーは、FPGA中央のクロック・マルチプレクサに直接アクセスする送信フルレート・クロック/ハーフレート・クロックでドライブされます。フルレート・クロックツリーは、エラスティック・バッファのリードポートと、ダウンサンプルFIFOのライトポートをドライブします。ハーフレート・クロックツリーは、ダウンサンプルFIFOとFPGAロジックをドライブします。

ケースII_b:16/20ビット、DSはバイパスしない

図8-36 16/20ビット、DS FIFOはバイパスしない



1. アップサンプルFIFOはこの場合位相シフトFIFOとアップサンプルFIFOとして機能します。
2. ダウンサンプルFIFOはこの場合位相シフトFIFOとダウンサンプルFIFOとして機能します。
3. FPGAの動作がフルレート・バイト周波数に間に合わない場合、これは非常に一般的なマルチチャネル・アライメント (MCA) のユースケースです。受信クロックツリー(最大4)は、ローカルまたはグローバルで、これらはハーフレートクロックで動作しています。送信クロックツリーは、FPGA中央のクロック・マルチプレクサに直接アクセスする送信ハーフレートクロックでドライブされます。この場合、MCAが完了した後に、FPGAロジックではCTCが必要でしょう。TxクロックツリーはCTCのリードポートを、マスターチャネル受信クロックはCTCのライトポートをクロッキングします。このユースケースでは、MCAとCTCが共に16/20ビット幅で行う必要性に留意してください。

SERDES/PCSブロックのレイテンシ(遅延)

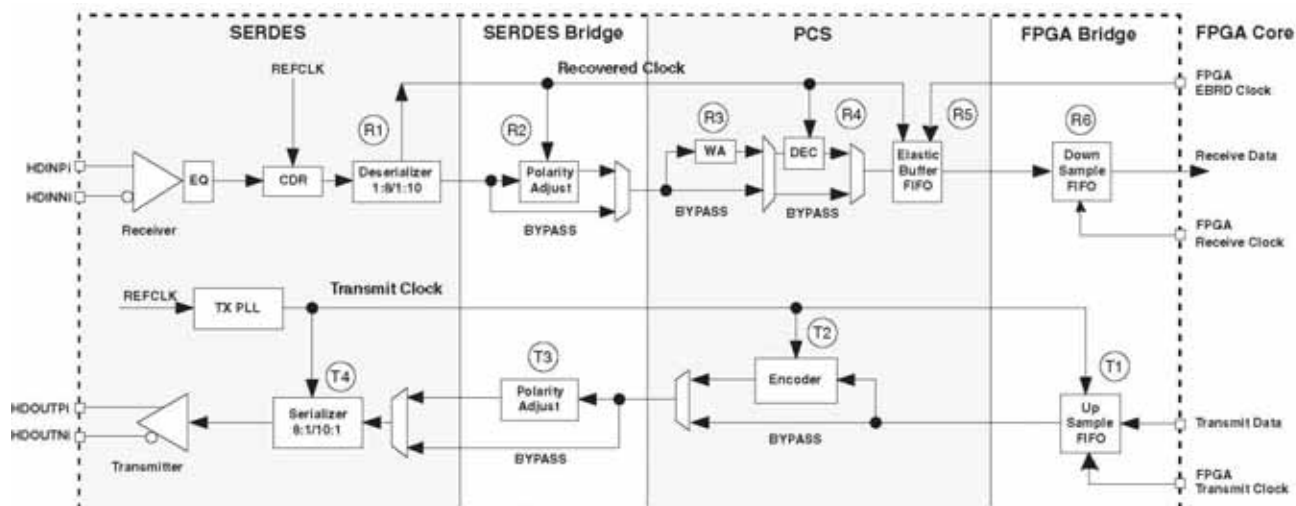
表8-22はトランスミッターとレシーバでそれぞれの機能ブロックの遅延について説明します。遅延はパラレルクロックのサイクル数で与えられています。図8-37はそれぞれのブロックの位置を示します。

表8-22 SERDES/PCS遅延の内訳(パラレル側クロック周期) 1

項目	記述	Min.	平均	Max.	Fixed	Bypass	単位
送信データ遅延							
T1	FPGAブリッジ、送信 ²	1	3	5		1	word clk
T2	8b10bエンコーダ	--	--	--	2	1	word clk
T3	SERDESブリッジ、送信	--	--	--	2	1	word clk
T4 ³	シリアライザ、8ビットモード	--	--	--	15+ $\Delta 1$	--	UI + ps
	シリアライザ、10ビットモード	--	--	--	18+ $\Delta 1$	--	UI + ps
受信データ遅延							
R1 ³	シリアライザ、8ビットモード	--	--	--	10+ $\Delta 2$	--	UI + ps
	シリアライザ、10ビットモード	--	--	--	12+ $\Delta 2$	--	UI + ps
R2	SERDESブリッジ、受信	--	--	--	2	1	UI + ps
R3	ワードアライメント	--	--	--	4	0	UI + ps
R4	8b10bデコーダ	--	--	--	1	1	UI + ps
R5	クロックトレランス補償	7	15	23		1	UI + ps
R6	FPGAブリッジ、受信 ²	1	3	5		1	UI + ps

1. PCS内部の平行クロック。このクロックレートは表8-6のrxfullclkと同じです。
2. FPGAブリッジの遅延はアップ/ダウン・サンプルFIFOのリード/ライトで変わります。これらの数は8/10ビット・インターフェイスで示されます。アップ/ダウン・サンプルFIFOの深さは4です。ダウンサンプルFIFOでは、ライトクロック・サイクル(1クロック)後に、最初のリードが行われます。最後のリードはFIFOがフル(4+1 = 5)になった後に行われます。16b/20bインターフェイスでは、値は倍になります。Min = 2, Max = 10。この遅延はFIFOの内部フラグ動作に依存します。
3. $\Delta 1 = \sim 245\text{ps}$ 、 $\Delta 2 = \sim 700\text{ps}$

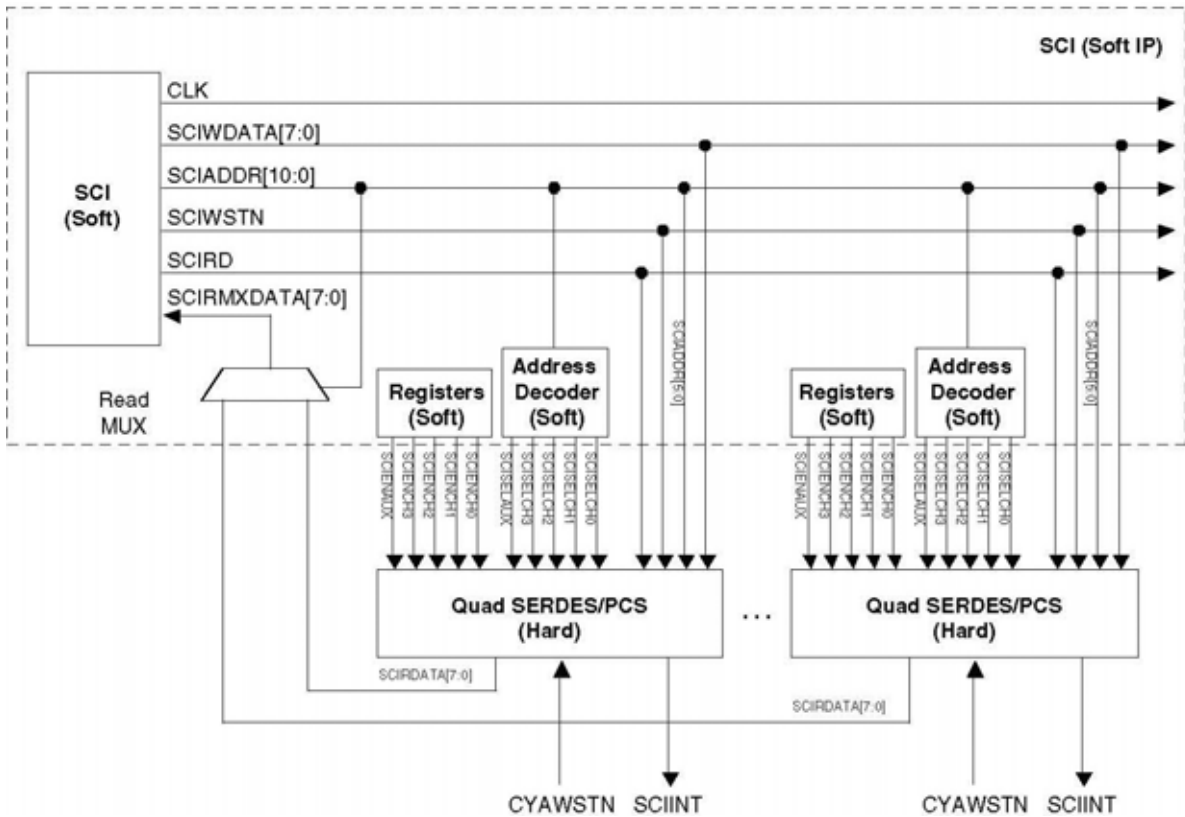
図8-37 トランスミッターとレシーバのブロック図



SERDESクライアント・インターフェイス(SCI)

SCIはFPGAコア内のソフトIPと、SERDES/PCSクワッドに含まれているハードロジックの両方から成ります。SCIは、SERDES/PCSクワッドがコンフィグレーション・メモリセル設定ではなくレジスタによって制御される事を可能する、簡易なレジスタ・コンフィグレーション・インターフェイスです。FPGAコアに置かれるSCIのソフトIP部分のブロック図は図8-38で示され、必要な主要信号を全て示しています。

図8-38 SCIインターフェイス・ブロック図



FPGAコアに置くソフトIPは、そのインターフェイス規定に従ってユーザが開発します。サンプル記述についてはラティス・テクニカルサポート・グループにお問い合わせ下さい。

SCIADDRバスはブロック内では6ビット幅ですが、ブロック境界におけるバス幅は11ビットです。上位5ビットはクワッドブロックとチャネルの選択に用いられます。表8-23はSERDESクワッド用のSCIアドレスマップを示します。

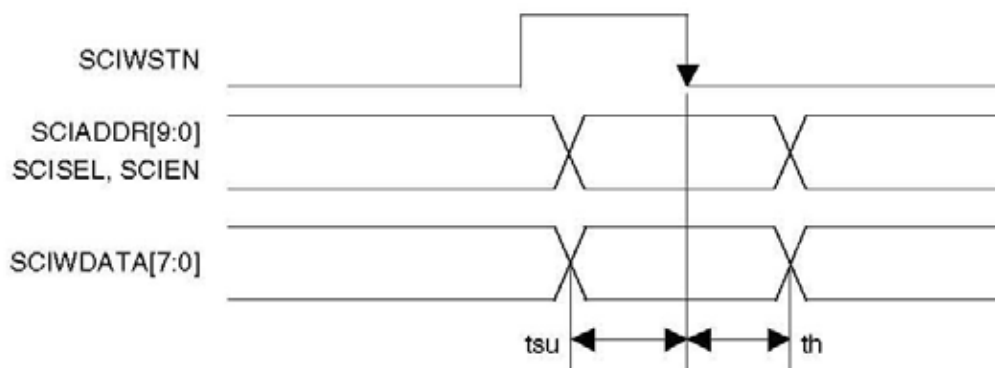
表8-23 最大4つのSERDES/PCSクワッドのSCIアドレスマップ

アドレスビット	記 述
SCIADDR[5:0]	レジスタ・アドレスビット
	000000 = register 0を選択
	000001 = register 1を選択
	...
	111110 = register 62を選択
	111111 = register 63を選択

SCIADDR[8:6]	チャンネル・アドレスビット 000 = channel 0を選択 001 = channel 1を選択 010 = channel 2を選択 011 = channel 3を選択 100 = aux channelを選択 101 = 未使用 110 = 未使用 111 = 未使用
SCIADDR[10:9]	クワッド・アドレスビット 00 = quad 0を選択 01 = quad 1を選択 10 = quad 2を選択 11 = quad 3を選択

このインターフェースを通してのリード/ライト動作は非同期です。ライトサイクル中、ライトデータとライトアドレスは、SCIWSTNの立ち下がりエッジに対してセットアップ時間とホールド時間を満たす必要があります。リードサイクルでは、タイミングはSCIRDパルスに対して規定されます。図8-39と図8-40はそれぞれライトとリードサイクルを示します。

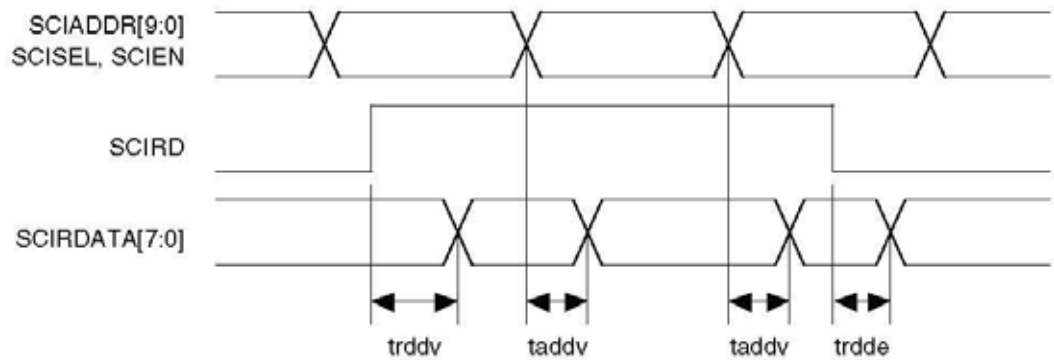
図8-39 SCIライトサイクル、重要なタイミングの注意:



- 1) tsuはライトストロブの立ち下がりエッジの前のアドレスとライトデータのセットアップ時間
- 2) thはライトストロブの立ち下がりエッジの後のアドレスとライトデータのホールド時間

注: 制御レジスタに対する意図しない書き込みを避けるため、レジスタを用いて、パワーアップ・リセット時にSCI入力ポートをLowにドライブするようにします。

図8-40 SCIリードサイクル、重要なタイミングの注意3



- 1) trddvはリードパルスのアサートからリードデータが有効になるまでの時間
- 2) taddvはリードパルスのアサート中にアドレスが変更されてからデータが有効になるまでの時間
- 3) trddeはリードパルスのネゲートからリードデータが有効なホールド時間

表8-24 タイミングパラメータ

パラメータ	Typical値	単位
tsu, trddv, taddv	1.127	ns
th, trdde	0.805	ns

SCIインターフェイスはメモリのリード/ライトと同じ程度に簡単です。これは擬似コードの例です:

ライト

- Cycle 1 : Set sciaddr[5:0], sciwdata[7:0], scien* = 1'b1 scisel* = 1'b1
- Cycle 2 : Set sciwstn from 0 => 1
- Cycle 3 : Set sciwstn from 1 => 0, scien* = 1'b0, scisel* = 1'b0

リード

- Cycle 1 : Set sciaddr[5:0], scisel* = 1'b1
- Cycle 2 : Set scird from 0 => 1
- Cycle 3 : Obtain reading data from scirdata[7:0]
- Cycle 4 : Set scird from 1 => 0

注: ヒューズをロードするとき、scien*の全5ビットをゼロに設定しなければなりません。

割り込みとステータス

ステータスビットはSCIを通して読み出せますが、SCIはバイト幅のため、一度に8つの割り込みステータス信号を読む事ができます。SCIINT信号は割り込みイベントが起こったことを示すためHighになります。その場合ユーザは、割り込みがクワッドからなのか、どれかのチャネルから来たのかを示すQIFステータス・レジスタをリードしなければなりません。このレジスタはリードではクリアされず、クワッドまたはチャネルからのすべての割り込みソースがクリアされることでクリアされます。

割り込みソースがいったん決定されると、ユーザは実際の割り込みソースを決定するために関連するクワッドかチャネルのレジスタを読むことができます。表8-25と表8-26は割り込みのすべてのソースを記載しています。

表8-25 クワッド割り込みソース

クワッド SCI_INTソース	記 述	レジスタ名
int_quad_out	クワッド割り込み。割り込みイベントがクワッドの何処かであると、このレジスタビットはアクティブになります。すべての割り込みイベントがクリアされると、このレジスタビットはクリアされます。	PCS Quad Status Register 1
int_cha_out[0:3]	チャンネル割り込み。割り込みイベントがそれぞれのチャンネルの何処かであると、このレジスタビットはアクティブになります。それぞれのチャンネルにおけるすべての割り込みソースがクリアしてあるとき、これらのレジスタビットはクリアされます。	PCS Quad Status Register 1
ls_sync_statusn_[0:3]_int ls_sync_status_[0:3]_int	Link Status Low(同期はずれ)チャンネル割り込み。 Link Status High(同期獲得)チャンネル割り込み	PCS Quad Status Register 3
~PLOL, PLOL	~PLOLとPLOLで生成されるPLLロック外れ割り込み	SERDES Quad Status Register 2

表8-26 チャンネル割り込みソース

クワッド SCI_INTソース	記 述	レジスタ名
fb_tx_fifo_error_int fb_rx_fifo_error_int cc_overrun_int cc_underrun_int	FPGAブリッジ・アップサンプルTx FIFOエラー割り込み FPGAブリッジ・ダウンサンプルRx FIFOエラー割り込み CTC (エラスティック・バッファ) オーバラン割り込み CTC (エラスティック・バッファ) アンダーラン割り込み	PCS Channel General Interrupt Status Register 4
pci_det_done_int rlos_lo_int ~rlos_lo_int rlos_hi_int ~rlos_hi_int rlol_int ~rlol_int	pci_det_done による割り込み rlos_lo による割り込み ~rlos_lo による割り込み rlos_hi による割り込み ~rlos_hi による割り込み rlol による割り込み ~rlol による割り込み	SERDES Channel Interrupt Status Register 5

SERDESクライアント・インターフェイス (SCI) アプリケーションの例

例えば:

ユーザは、Quad PCS制御レジスタser_ctl_3_qd_13のrefck_mode[1] (D7)とrefck_mode[0] (D6)ビットを変えることによって、基準クロックで1倍モードを10xと20xの間で切り替えることができます。レジスタのSCIアドレスは0x13でORCAstraアドレスは0x113です。この設定を有効にするためには、Quad Reset (QD_18[5])をトグルしなければなりません。

リード/ライト動作は2つの異なる方法で達成することができます。

1. メインウィンドーでは、ユーザは制御レジスタの内容を読み書きでき、ステータス・レジスタを読むことができます。
2. サブウィンドウでは、ユーザは、ON/OFFオプションのプルダウンメニュー・オプションを選択するか、あるいはチェックするかチェックを外すことができます。

表8-27 SCIアドレスマップ

アドレスビット	記 述
SCIADDR[5:0]	レジスタ・アドレスビット 000000 = register 0を選択 000001 = register 1を選択 ... 111110 = register 62を選択 111111 = register 63を選択
SCIADDR[8:6]	チャンネル・アドレスビット 000 = channel 0を選択 001 = channel 1を選択 010 = channel 2を選択 011 = channel 3を選択 100 = aux channelを選択
SCIADDR[10:9]	クワッド・アドレスビット 00 = quad 0を選択 01 = quad 1を選択 10 = quad 2を選択 11 = quad 3を選択

SERDES/PCSクワッドのダイナミック・コンフィグレーション

SERDES/PCSクワッドは、コンフィグレーション・メモリセルかオプションのSCIを通してアクセスされるレジスタによって制御することができます。SCIはFPGAコア内のソフトIPとSERDES/PCSクワッドに含まれているハードロジックの両方から成ります。ソフトIPがFPGAコアに存在する場合にのみSCIが利用できます。

コンフィグレーション完了後、コンフィグレーション・メモリセルが自動的にレジスタにコピーされます。レジスタ値のその後の変更は、もちろんSERDES/PCSクワッドの動作を変えますが、コンフィグレーション・メモリセルに格納された値には影響しません。

コンフィグレーション・メモリセルによって制御される場合、コンフィグレーションが完了後に、ユーザからの干渉なしにSERDES/PCSクワッドが機能状態に至る必要があります。これは、SERDES/PCSクワッドを初期化するために必要な、特別なリセットシーケンスは、自動的にハードウェアで扱わなければならないことを意味します。言い換えると、SCIの使用はオプションのため、SERDES/PCSクワッドはソフトIPがFPGAコアに存在していると想定していません。

SERDESデバッグ機能

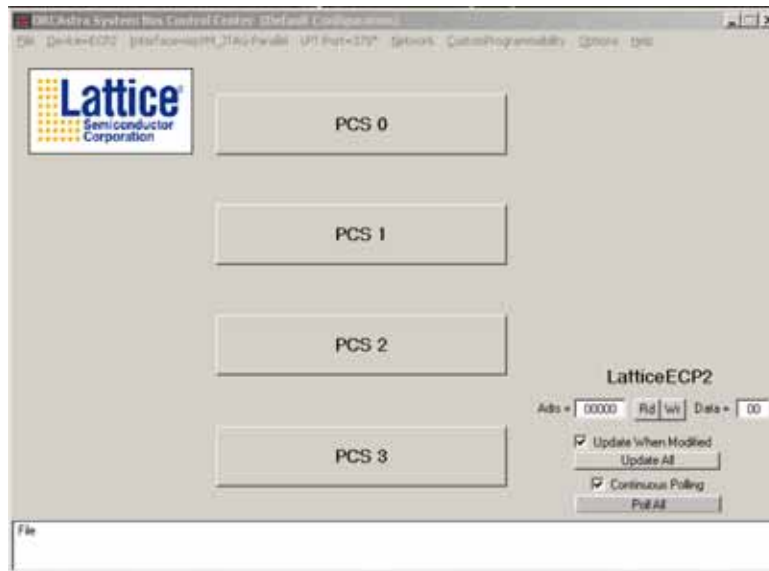
ラティスには、LatticeECP2MデバイスにおけるSERDES/PCS動作のデバッグを手助けするツールがあります。

ラティスORCAstraソフトウェアはPCベースのGUIで、オンチップ・レジスタの制御ビットでLatticeECP2Mデバイスの動作モードを構成・変更するためのツールです。これは、時間の掛かる再コンパイル・プロセスを実行したり、ボードの変更をすることなく、迅速に（最適な）コンフィグレーション・オプションを探ることを助けます。GUIで作成されたコンフィグレーションは、PC上のメモリに保存して、後で使用する時に再ロードすることができます。ORCAstraを用いるためには、ユーザロジックにSCI IPをインスタンスしなければなりません（訳注；ispLEVER7.1SP1から、IPexpressでORCAstraモジュールのRTLを自動生成する機能に対応しています）。

スクリプトベースのコンフィグレーションとテストをサポートするために、マクロ機能も利用できます。また、リアルタイムでステータス情報を表示するためにGUIを用いることができます。ORCAstraソフトウェアの使用はFPGAのプログラミングを妨げません。

図8-41はORCAstra GUIトップレベル・ウィンドウで、各PCSチャンネルのサブウィンドウを用いることなく、ユーザはAdrsセルによってデータをリード/ライトすることができます(右下部)。呼び出されると、ORCAstraは自動的にデバイスタイプを認識します。または、デバイス・プルダウンメニューの下でデバイスタイプを選択することもできます。

図8-41 ORCAstra トップレベル・スクリーンショット

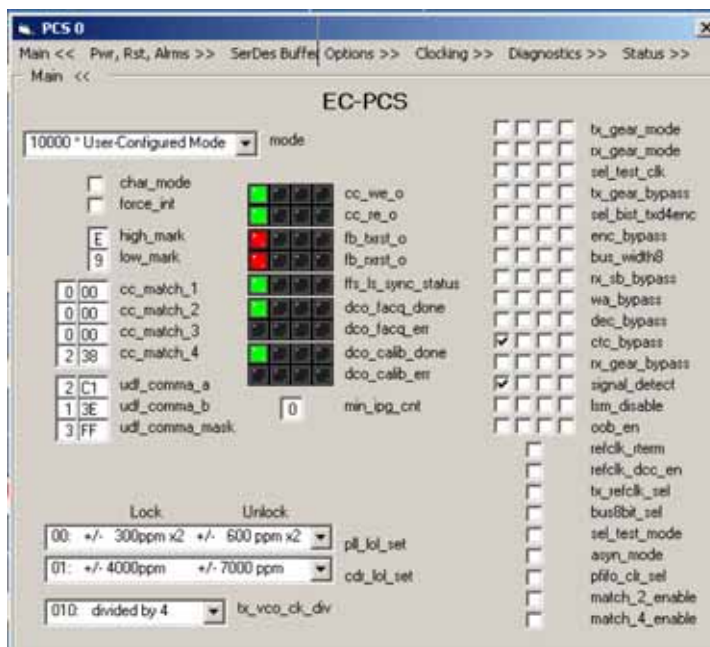


デフォルトで、図8-41で示すData Boxはリトルエンディアン(MSBが右)のビット順に従います。ユーザは、Optionsタブの下でData BoxでDisplay Data reversedを選択することによって、ビッグエンディアンに変更することができます。

PCS0(Quad 0)ボタン上をダブルクリックすると図8-42で示されるようなメインウィンドーが開きます。

これらの標準ウィンドウメニューはデバイスとインターフェイスの選択を制御します。また、格納されたファイルでコンフィグレーションをセットアップしたり、保存すること含む、様々なコンフィグレーション・オプションをサポートします。

図8-42 ORCAstraメインウィンドー

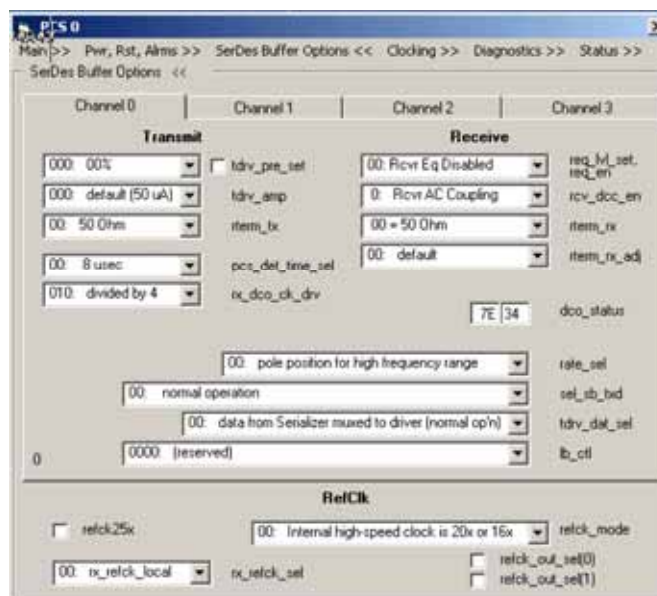


コントロールボックス、ボタン、ステータスボックス、およびテキストウィンドウ

コントロールボックスの上にカーソルを動かして、マウス左ボタンをクリックして、制御ビットを設定します。ともに、選択されたボックスのビット位置と機能は、テキストウィンドウに表示されます。これはデバイス・データシートのレジスタマップ表と合致します。ビット名の上にカーソルがあるとき、機能だけを表示します。ステータスボックスは、コントロールボックスと類似していますが、LED表示と色付きのバックグラウンドになっています。

図8-43はSERDES Buffer Optionsウィンドウを示します。プルダウンメニューからコンフィグレーション・オプションを選択することができます。

図8-43 SERDES Buffer Options ウィンドウ



以下のアドレスのラティスセミコンダクター・ウェブサイトにORCAstraの詳しい情報とダウンロードできるファイルがあります；www.latticesemi.com/products/devtools/software/orcastra/index.cfm

デザインに関する他の考察

LatticeECP2M-35対、他のLatticeECP2Mデバイス

重要な注意:LatticeECP2M-35の(.txtファイル内の)SERDES設定は、他の全LatticeECP2Mデバイスと異なっています。違いは.txtファイルの中に隠されているデフォルト設定を含んでいます。

LatticeECP2M-35から他のいずれかのLatticeECP2Mデバイスに移行する場合、ユーザはIPexpress GUIからPCSモジュールを再生成しなければなりません。

エンジニアリング・サンプル対量産デバイス

エンジニアリング・サンプルと生産デバイスには、それぞれのシミュレーション・モデルとライブラリがあります。エンジニアリング・サンプルのライブラリ名はpcsc_mti_work_revA（訳註；これはModelSim用。Active-HDL用はpcsc_aldec_work_revA）です。

SERDES/PCSのシミュレーション

SERDES/PCSのシミュレーションは、プリコンパイルされたモデルによってModelSim®とNC-Verilog®用に提供されます（訳註；Active-HDL用も提供）。プリコンパイルされたシミュレーション・モデルは、SERDESのビヘービアモデルとPCSのRTLです。表8-28はそれぞれのシミュレーション・モデルの場所を示します。

表8-28シミュレーションモデルの場所

シミュレータ	モデルの場所
Cadence NC-Verilog/VHDL, NC-Sim, Synopsys VCS, Mentor Graphics ModelSim, Aldec Riviera Pro	ispTools¥cae_library¥simulation¥blackbox

なお、配布されているモデルはzipファイルですので、使用する前に解凍する必要があります。

シミュレーション時のリセット方法

PCSリセット信号がいずれもGNDに接続されていない場合、設計者は同じリセット信号がFPGAコアで用いられているかどうか注意しなければなりません。例えば、リセット信号の1つがFPGAコアのカウンタとPCSの両方をリセットするのに用いられて、かつカウンタがtxfullclkを用いていると、カウンタは動作しないでしょう。リセットがアクティブなときにtxfullclkが動作していないので、カウンタはそれ自体をリセットすることができません。

基準クロックソースに依存して、リセットのアサート時間は異なり得ます。シミュレーション時にのみ、PCSリセット信号に最小100nsのアクティブ期間を用いることを推奨します。

16/20ビットのワードアライメント

PCSレシーバは16ビットのワード境界を認識することができません。AUTOモードによるCOMMA_ALIGNと共に、PCSはバイトアライメントのみ行うことができます。16ビット・ワードアライメントは、FPGAファブリックで行うべきで、これはかなり素直な処理です。以下で説明されるようなアライメント機構では、機能アップすることもできます。

例えば、FPGAインターフェイスの前の送信データが、

YZABCDEFGHJKLM.... (それぞれ8-bitか10-bitの文字)

そして、8b10bデコーダ後とrx_gearboxの前のPCSにおける入力データは：

YZABCDEFGHJKLM....

rx_gearboxの後、これは：

1. {ZY}{BA}{DC}{FE}{HG}{JI}{LK}....
または
2. {AZ}{CB}{ED}{GF}{IH}{KJ}{ML}...

明らかに、シーケンス2はアラインされていません。これは1バイトのオフセットがありますが、16/20ビットのアライメントが必要です。特殊文字Aは常に下位バイトに置かれるべきです。

20ビットデータを現在の16/20ビットデータに組み合わせるで32/40ビット・データを形成すると、以下に示すようになります：16/20ビット・アライメント後の出力データは以下です。

1. {DCBA}{HGFE}{LKJI}....

^
| **下位10-bit側にAを検出。オフセットを0に設定してアラインされたデータBAを送出。
次のクロック周期で；
{FEDC}{JIHG}{NMLK}....
^
| **アラインされたデータDCを送出
etc.

20ビット・アライメント後の出力データは以下です。

{ZY}{BA}{DC}{FE}{HG}{JI}{LK}....

2. {CBAZ}{GFED}{KJIH}....

^
| ** 上位10-bit側にAを検出。オフセットを10に設定してアラインされたデータBAを送出。
次のクロック周期で；
{EDCB}{IHGF}{MLKJ}....
^
| **アラインされたデータDCを送出。
etc.

20ビット・アライメント後の出力データは以下です。

{ZY}{BA}{DC}{FE}{HG}{JI}{LK}....

注: 8/10ビットや16/20ビットワードのLSBが常に最初に送信され、最初に受信されます。

SCIを用いた10XH、10X、または20X基準クロックで1倍モードの切り替え

設計者はSCIを用いることで、基準クロックで1倍モードを10XHと、10X、または20Xの中から指定することができます。これは、レシーバが受信信号にロックするように異なるレートを試すようなシステムでは役に立つ機能です。例えばシステムレートが622Mbps、1.244Gbps、2.488Gbpsを取り得る場合、それらを順次切り替えることで、いずれかの1倍モードで受信信号にロックすることができます。

異なるレートに関連するいくつかの制御レジスタビットがあります。

CH_0A[D1]: rate_mode_tx
 CH_0B[D1]: rate_mode_rx
 QD_13[D6]: refclk_mode[0]
 QD_18[D5]: quad_rst

16ビットインターフェイスでの20Xと20XH、または10Xと10XHモード間の切り替え

上記制御レジスタの設定に加えて、送信インターフェイス・クロック(ffc_txiclk_chn)入力は表8-6にあるようにtxhalfclkからtxqtrclkに切り替える必要があります。この場合ff_rxiclkは影響されません。CDR PLLは供給されるデータに自動的にチューニングされ、表8-6のような正しいrxhalfclkを供給します。

オフチップAC結合

チップ外部でAC結合が必要なとき、表8-29に推奨のコンデンサ値を示します。

表8-29 外付けAC結合コンデンサ値

プロトコル	Min.	Typ.	Max.	コメント	単位
8b10b	4.7	10		@3.125Gbps	nF
PCI Express rev1.1	75		200		nF

8b10b符号などDCバランスのとれているパターンが用いられるとき、エッジの劣化を防ぐために最小4.7nFのコンデンサが必要です。より長いランレングスがあるパターンでは、データ依存のジッタを減らすためにより大きいコンデンサ値が必要です。詳しい情報についてはラティステクニカルノートTN1114(Electrical Recommendations for Lattice SERDES)を参照してください。

未使用のクワッド/チャネルと供給電源

未使用のチャネルでもVCCTX、VCCR_X、VCCP、およびVCCAUX33には電力が供給されなければなりません。VCCIB、VCCOB、HDINP/N、HDOUTP/N、およびREFCLKP/Nはフローティングのままにすべきです。未使用のチャネル出力はトライステートにされ、差動出力ペアの間の内部抵抗はおよそ10KΩです。VCCAUX33は終端抵抗に電力を供給します。他の供給電源のようにπ型フィルタを用いることをお勧めします。VCCAUX33の雑音は直接高速I/OのHDIN/HDOUTと結合されるでしょう。VCCAUX(FPGAコア供給電源)が非常にきれいであれば、それをVCCAUX33に接続することができます。

また、未使用のSERDESはデフォルトでパワーダウン・モードにコンフィグレーションされます。

SERDES/PCSリセットのリセットシーケンス

パワーアップとコンフィグレーションの後に、すべてのSERDESリセットとFPGAリセットが与えられます。

1. SERDESリセットシーケンス

- ・SERDESリセットシーケンスを開始として、quad_rstを最低20ns与えます。
- ・quad_rstネゲートの後に、plolとrlol_ch#が非アクティブになるのを待ちます。
- ・plolとrlol_ch#が非アクティブの後にlane_tx_rst_ch#とlane_rx_rst_ch#をネゲートします。

2. SERDES/PCSリセットのネゲート後にFPGA Logic Resetをネゲートします。

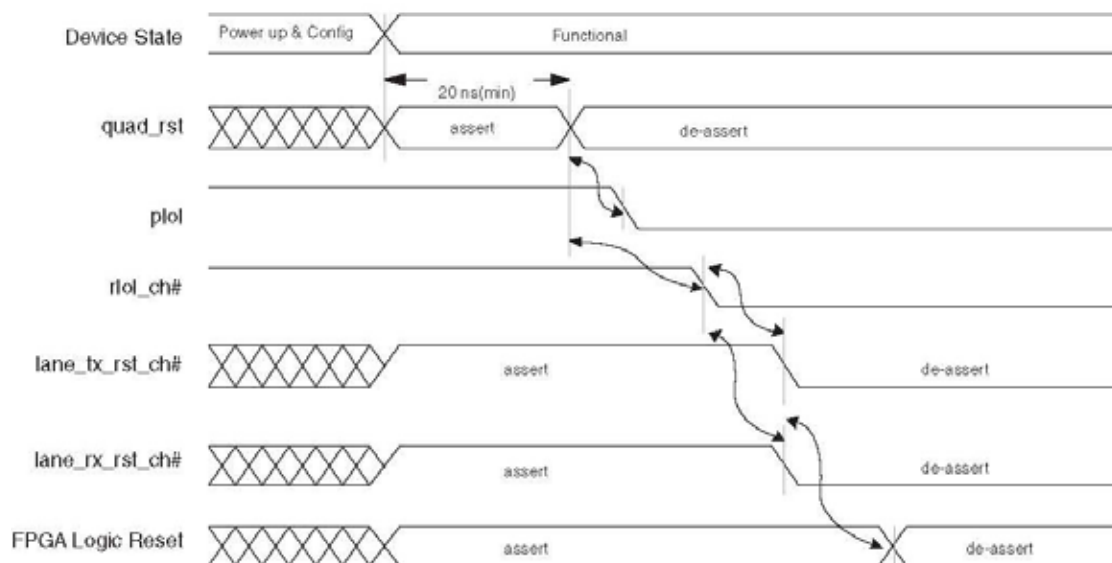
- ・持続時間はプロトコルに依存するかもしれません。

3. デバッグだけに用いられる他のリセット；

- ・macro_rstは常時Lowに接続します。
- ・trstとrrst_ch#は常時Lowに接続します。

リセットシーケンスは図で8-44で説明されています。

図8-44 リセットシーケンス



注；ユーザのシステム環境によっては、CDRプロセス中にrlo_l_ch#信号はトグルする可能性があり、CDRロック後に結果的にLowになります。

通常このプロセスは3ms程度かかります。4msカウンタを用いてquad_rstでカウント開始しrlo_l_ch#がLowからHighになる度にリセットするようにすることで、FPGAログで適切にロック検出を検出するようにできます。

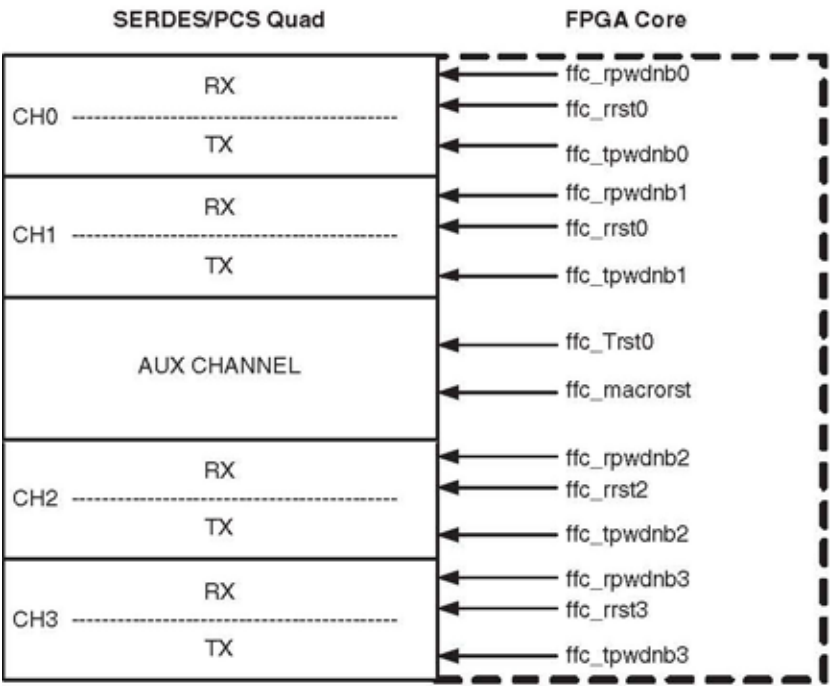
リセットシーケンス・モジュールの例はTN1153 (LatticeECP2M PRBS SERDES Demo User's Guide) に記述されています。

リセットとパワーダウン制御

SERDESクワッドには図8-45で示されるように、マクロ全体のリセットとパワーダウン制御、及び各トランスミッターとレシーバの個別リセットとパワーダウン制御があります。リセット信号はHighアクティブで、パワーダウン信号はLowアクティブです。様々なリセットとパワーダウン制御の動作は以下のセクションで説明されます。

注: デバイスがパワーアップ時にチップレベルのパワーオンリセットがアクティブであるとき、PCSのSERDES制御ビットはクリアされます(または、そのデフォルト値になります)。これはSERDESクワッドをパワーダウン状態にします。

図8-45 SERDES/PCS クワッドリセットとパワーダウン制御



リセットの生成と分配は、クロックとリセットのブロックで扱われます。通常、すべてのリセットがハードリセットと様々なFPGAファブリック・リセットによります。リセットパルス幅は(RTL)ソースの記述によります。しかしながら、少なくとも1クロック幅でなければなりません。リセットロジックは図8-46で示され、そして表8-30は対応表です。

表8-30 SERDES/PCS リセット表

リセット信号		PCS TX	PCS RX	SERDES TX	SERDES RX	PCS CTRL レジスタ	TX PLL	CDR PLL
FPGA	制御レジスタ							
ffc_tx_lane_rst [3:0]	lane_tx_rst[3:0]	X						
ffc_rx_lane_rst [3:0]	lane_rx_rst[3:0]		X					
ffc_quad_rst	quad_rst	X	X	X	X		X	X
ffc_macrorst	macrorst			X	X		X	X
ffc_rrst[3:0]	rrst[3:0] ³				X			
ffc_trst[3:0]	trst[3:0] ³						X	
TRI_ION		X	X	X	X	X		

- 1. SB(SERDES Bidge)、PCSコア、FB(FPGA Bridge)各サブブロックを含む
- 2. ラティスセミコンダクター社内用途のみ。本リセットは常時Lowに接続すること
- 3. TX PLL loss of lock(ffx_pll)のみをリセット

図8-46 SERDES/PCS リセット・ダイアグラム

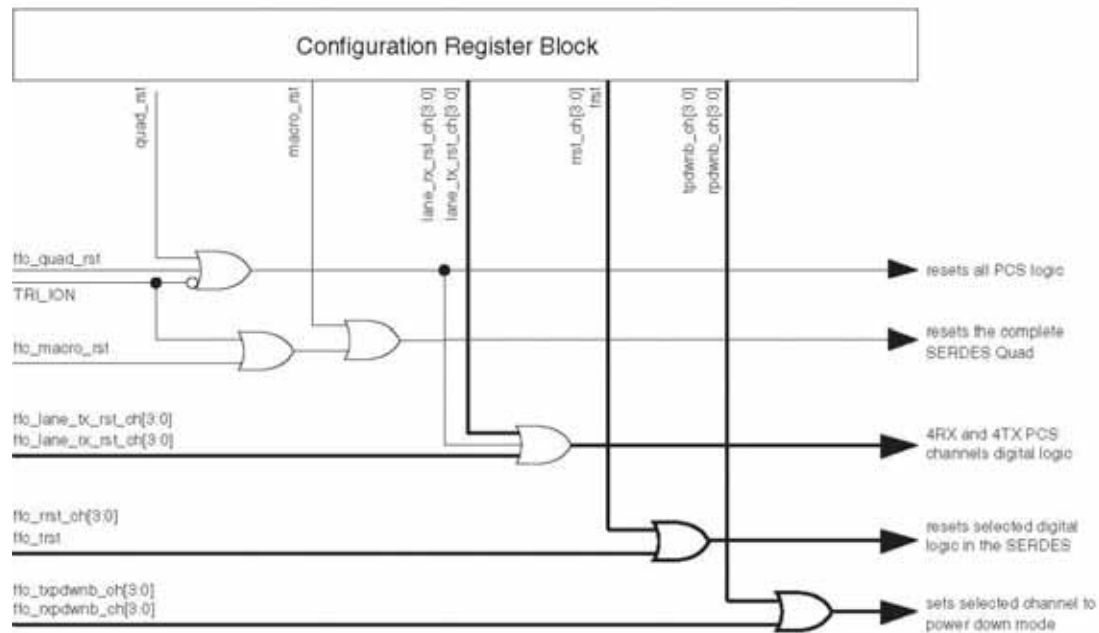


表8-31 リセット制御の記述^{1, 2, 3}

リセット信号		記 述
FPGA	制御レジスタ	
ffc_quad_rst	quad_rst	Highアクティブの非同期入力。SERDESクワッドの補助(auxiliary)チャネルとPCSを含む全チャンネルをリセット。本リセットはmacro_rst、txpll、cdr、lane_tx_rst、lane_rx_rstを含む。
ffc_macro_rst	macro_rst	SERDESクワッドへのHighアクティブの非同期入力。ソフトウェア制御ビットとゲートされる。本リセットはSERDESブロックのみに有効で、TXPLLとCDRPLLも含まれる。
ffc_lane_tx/rx_rst_ch[0:3]	lane_tx/rx_rst_ch[0:3]	Highアクティブの非同期入力。個々のTX/RXチャンネルをリセットし、SB、PCSコア、FB各ブロックに有効。
ffc_rrst_ch[0:3]	rrst[0:3]	loss-of-lock(rlol)回路とloss-of-signal回路をリセット。
ffc_trst	trst	AUX PLLのloss-of-lock(plol)をリセット。

- フルデータレート・モードで動作するクワッドの全チャンネルにとって、パラレル側クロックは同位相であることが保証されます。
- ハーフデータレート・モードで動作するクワッドでは、各チャンネルには個別の2分周回路があります。クワッドにはmacro_rstのネゲート後に、2分周回路が同位相にあることを保証するメカニズムが全くないので、PCSデザインはこれらが(したがってパラレル側クロックも)同位相にないと想定するべきです。
- ハーフデータレート・モードで、これはマルチチャンネル・リンクの送受信側でチャンネル間スキューを与えるかもしれません。

パワーダウン制御の記述

各RxとTxチャンネルは、レジスタビットかFPGAからの制御信号で個別にパワーダウンが可能です。個別チャンネルのパワーダウン制御ビットは、SERDESマクロ内の選択されたブロックと高速I/Oバッファのみをパワーダウンします。

表8-32 パワーダウン制御の記述

信号		記 述
FPGA	レジスタ	
macropdb		SERDESクワッドへのLowアクティブの非同期入力で、補助チャンネルを含む全チャンネルに有効。Lowにドライブされると、送信PLLを含むマクロ全体をパワーダウンする。すべてのクロックが止められ、マクロの電力消費は最少になる。
ffc_txpwdnb_ch[0:3]	tpwdnb[0:3]	Lowアクティブで、各チャンネル用のトランスミッタ・パワーダウン。シリアライザと出力ドライバのみをパワーダウン。本モードではHDOUTは5kΩの抵抗で弱くプルアップされる。
ffc_rxpwdnb_ch[0:3]	rpwdnb[0:3]	Lowアクティブで、各チャンネル用のレシーバ・パワーダウン。CDRと入力バッファ（イコライザとデシリアライザ）、およびLoss-of-Signal検出器をパワーダウン。

表8-33 リセットパルス仕様

パラメータ	記 述	Min.	Typ.	Max.	単位
t _{MACRORST}	マクロリセットHigh時間	1			ns
t _{RRST}	チャンネルRXリセットHigh時間	3			ns
t _{TRST}	クワッドTXリセットHigh時間	3			ns

表8-34 パワーダウン/パワーアップのタイミング仕様

パラメータ	記 述	Min.	Typ.	Max.	単位
t _{PWRDN}	macropdb後のパワーダウン時間	20			ns
t _{PWRUP}	macropdb後のパワーアップ時間	20			ns

供給電源のシーケンス要件

1.5VのVCCIBかVCCOBでSERDESを使用するとき、1.2V電源を供給しないでSERDESを1.5V電源を定常的に供給するべきではありません。1.2Vと1.5V電源の両方が同時にSERDESに供給されなければなりません。ただし供給電源と電圧レギュレータの、通常のランプアップ時の時間差は問題ではありません。

参照ドキュメント

- ラティスセミコンダクター・テクニカルノート
- TN1029, *FPSC SERDES CML Buffer Interface*
- TN1084, *LatticeSC/M and LatticeECP2/M SERDES Jitter*
- TN1114, *Electrical Recommendations for Lattice SERDES*
- TN1159, *LatticeECP2/M Pin Assignment Recommendations*
- HB1003, *LatticeECP2/M Family Handbook*

テクニカル・サポート支援

ホットライン: 1-800-LATTICE (North America)

+1-503-268-8001 (Outside North America)

e-mail: techsupport@latticesemi.com

インターネット: <http://www.latticesemi.com>

変更履歴（日本語版）

Rev.#	日付	変更箇所
2.6J0	---	日本語初版。訳註；次の原文は日本語版には意図的に含めていない（ページ番号は原文）。 • SATAサポートに関する記述全般（オプションとしての説明を除く） • p.8-13 プリエンファシスの記述 • p.8-16 Txレーン間スキューの一部記述 • p.8-27 OOB信号とSATAアプリケーション • p.8-38 コンフィグレーション・ファイル記述 • p.8-57 SERDES Client Interface アプリケーション例の前半 • p.8-58 SERDES/PCSクワッドのダイナミック・コンフィグの第三パラグラフ
2.8J0	Nov.2008	• SATA関連記述を削除（page.8-1, 表8-1, 表8-16, 表8-30） • 表8-5、信号追加(rx/tx_sdi_en)、脚注更新 • 表8-23 Latency更新、表8-31/8-32 Reset Control更新、表8-33 Power Down更新 • (p.8-65)クロックモード切替節記述追加
2.9J	Feb.2009	• PCI PIPEモード関連記述を削除（表8-4 脚注3、表8-11 プロトコル、表8-12 最上部モード、表8-16 受信・送信各モード、pp.8-45～46 “PCS in PIPE Mode”節、表8-105 プロトコル）。表8-20以降の番号を順次繰り上げ。 • 図8-44 の脚注、オリジナル復活（日本語版は対応済み）

付録A.メモリマップ

コンフィグレーション・レジスタ定義

特定のクワッドレベル・レジスタとチャンネルレベル・レジスタがあります。各カテゴリに、SERDESに特有のレジスタとPCSに特有のレジスタがあります。サブカテゴリ中には以下があります。

- ・ 制御レジスタ
- ・ ステータス・レジスタ
- ・ リードでクリアされるステータス・レジスタ
- ・ 割り込み制御レジスタ
- ・ 割り込みステータス・レジスタ
- ・ 割り込みソースレジスタ(リードでクリア)

以下に示すすべてのレジスタビットは、D0がLSBです。

各レジスタがどのようなタイプかをそれぞれ以下の通り定義します(R/W = Read/Write, RO = Read Only, CR = Clear on a read ; リードでクリア)。

これらのレジスタビットはそれぞれのコンフィグレーション・メモリセルがシャドーになっています。これらのメモリセルはビットストリーム制御でのみプログラマブルです。コンフィグレーションが完了後、レジスタに関連づけられたコンフィグレーション・メモリセルは自動的にレジスタにコピーされます。レジスタに関する内容をその後に変更しても、コンフィグレーション・メモリセルに格納された値には影響しませんが、SERDES/PCSクワッドの動作を変えます。

すべてのreserved(予約)ビットにはゼロが書かれます。

クワッドごとのレジスタ概要

表8-35 クワッド・インターフェイス・レジスタマップ

BA 1	レジスタ名	D7	D6	D5	D4	D3	D2	D1	D0
1. クワッド用の制御レジスタ (28)									
クワッド用のPCS制御レジスタ (17)									
00	qd_00	sync_toggle	force_int	char_mode	xge_mode	rio_mode	pcie_mode	fc_mode	uc_mode5
01	qd_01	bist_rpt_ch_sel[1] 6	bist_rpt_ch_sel[0]	bist_res_sel[1]	bist_res_sel[0]	bist_time_sel[1]	bist_time_sel[0]	bist_head_sel[1]	bist_head_sel[0]
02	qd_02	high_mark[3]	high_mark[2]	high_mark[1]	high_mark[0]	low_mark[3]	low_mark[2]	low_mark[1]	low_mark[0]
03	qd_03	min_ipg_cnt[1]	min_ipg_cnt[0]	match_4_enable	match_2_enable		pfifo_clr_sel	asyn_mode	sel_test_clk
04	qd_04	cc_match_1[7]	cc_match_1[6]	cc_match_1[5]	cc_match_1[4]	cc_match_1[3]	cc_match_1[2]	cc_match_1[1]	cc_match_1[0]
05	qd_05	cc_match_2[7]	cc_match_2[6]	cc_match_2[5]	cc_match_2[4]	cc_match_2[3]	cc_match_2[2]	cc_match_2[1]	cc_match_2[0]
06	qd_06	cc_match_3[7]	cc_match_3[6]	cc_match_3[5]	cc_match_3[4]	cc_match_3[3]	cc_match_3[2]	cc_match_3[1]	cc_match_3[0]
07	qd_07	cc_match_4[7]	cc_match_4[6]	cc_match_4[5]	cc_match_4[4]	cc_match_4[3]	cc_match_4[2]	cc_match_4[1]	cc_match_4[0]
08	qd_08	cc_match_4[9]	cc_match_4[8]	cc_match_3[9]	cc_match_3[8]	cc_match_2[9]	cc_match_2[8]	cc_match_1[9]	cc_match_1[8]
09	qd_09	udf_comma_mas k[7]	udf_comma_mas k[6]	udf_comma_mas k[5]	udf_comma_mas k[4]	udf_comma_mas k[3]	udf_comma_mas k[2]	udf_comma_mas k[1]	udf_comma_mas k[0]
0A	qd_0a	udf_comma_a[7]	udf_comma_a[6]	udf_comma_a[5]	udf_comma_a[4]	udf_comma_a[3]	udf_comma_a[2]	udf_comma_a[1]	udf_comma_a[0]
0B	qd_0b	udf_comma_b[7]	udf_comma_b[6]	udf_comma_b[5]	udf_comma_b[4]	udf_comma_b[3]	udf_comma_b[2]	udf_comma_b[1]	udf_comma_b[0]
0C	qd_0c	udf_comma_a[9]	udf_comma_a[8]	udf_comma_b[9]	udf_comma_b[8]	udf_comma_mas k[9]	udf_comma_mas k[8]	bist_mode	bist_en
0D	qd_0d	bist_udf_def_hea der[7]	bist_udf_def_hea der[6]	bist_udf_def_hea der[5]	bist_udf_def_hea der[4]	bist_udf_def_hea der[3]	bist_udf_def_hea der[2]	bist_udf_def_hea der[1]	bist_udf_def_hea der[0]
0E	qd_0e	bist_udf_def_hea der[15]	bist_udf_def_hea der[14]	bist_udf_def_hea der[13]	bist_udf_def_hea der[12]	bist_udf_def_hea der[11]	bist_udf_def_hea der[10]	bist_udf_def_hea der[9]	bist_udf_def_hea der[8]
0F	qd_0f	bist_bus8bit_sel	bist_ptn_sel[2]	bist_ptn_sel[1]	bist_ptn_sel[0]	bist_udf_def_hea der[19]	bist_udf_def_hea der[18]	bist_udf_def_hea der[17]	bist_udf_def_hea der[16]
10	qd_int_10 ²	ls_sync_status_3 _int_ctl	ls_sync_status_2 _int_ctl	ls_sync_status_1 _int_ctl	ls_sync_status_0 _int_ctl	ls_sync_statusn_3 _int_ctl	ls_sync_statusn_2 _int_ctl	ls_sync_statusn_1 _int_ctl	ls_sync_statusn_0 _int_ctl
クワッド用のSERDES制御レジスタ (6)									
11	qd_11	reserved	reserved	tx_refck_sel	refck_dcc_en	refck_rterm	refck_out_sel[2]	refck_out_sel[1]	refck_out_sel[0]
12	qd_12	refck25x	bus8bit_sel	rlos_hset[2]	rlos_hset[1]	rlos_hset[0]	rlos_lset[2]	rlos_lset[1]	rlos_lset[0]
13	qd_13	refck_mode[1]	refck_mode[0]	reserved	reserved	reserved	reserved	cdr_lol_sel[1]	cdr_lol_sel[0]
14	qd_14	reserved	reserved	reserved	pll_lol_sel[1]	pll_lol_sel[0]	tx_vco_ck_div[2]	tx_vco_ck_div[1]	tx_vco_ck_div[0]
15	qd_15	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
16	qd_int_16 ²	lol_int_ctl	~plol_int_ctl	reserved	reserved	reserved	reserved	reserved	reserved
クワッド用のクロックリセット・レジスタ (5)									
17	qd_17	lane_rx_rst3	lane_rx_rst2	lane_rx_rst1	lane_rx_rst0	lane_tx_rst3	lane_tx_rst2	lane_tx_rst1	lane_tx_rst0
18	qd_18	macropdb	macro_rst	quad_rst	trst	rrst[3]	rrst[2]	rrst[1]	rrst[0]
19	qd_19	bist_rx_data_sel	bist_bypass_tx_gate	bist_sync_head_req[1]	bist_sync_head_req[0]	sel_sd_rx_clk3	sel_sd_rx_clk2	sel_sd_rx_clk1	sel_sd_rx_clk0
1A	qd_1a	ff_rx_clk_sel_2[3]	ff_rx_clk_sel_2[2]	ff_rx_clk_sel_2[1]	ff_rx_clk_sel_2[0]	ff_rx_clk_sel_1[3]	ff_rx_clk_sel_1[2]	ff_rx_clk_sel_1[1]	ff_rx_clk_sel_1[0]
1B	qd_1b	ff_tx_clk_sel[2]	ff_tx_clk_sel[1]	ff_tx_clk_sel[0]	reserved	ff_rx_clk_sel_0[3]	ff_rx_clk_sel_0[2]	ff_rx_clk_sel_0[1]	ff_rx_clk_sel_0[0]
2. クワッド用のステータスレジスタ (9)									
クワッド用のPCSステータスレジスタ (5)									
20	qd_20				int_qua_out	int_cha[3]	int_cha[2]	int_cha[1]	int_cha[0]
21	qd_21 ³	ls_sync_status_3	ls_sync_status_2	ls_sync_status_1	ls_sync_status_0	ls_sync_statusn_3	ls_sync_statusn_2	ls_sync_statusn_1	ls_sync_statusn_0
22	qd_int_22 ⁴	ls_sync_status_3 _int	ls_sync_status_2 _int	ls_sync_status_1 _int	ls_sync_status_0 _int	ls_sync_statusn_3 _int	ls_sync_statusn_2 _int	ls_sync_statusn_1 _int	ls_sync_statusn_0 _int
23	qd_23	bist_report[7]	bist_report[6]	bist_report[5]	bist_report[4]	bist_report[3]	bist_report[2]	bist_report[1]	bist_report[0]

24	qd_24	bist_report[15]	bist_report[14]	bist_report[13]	bist_report[12]	bist_report[11]	bist_report[10]	bist_report[9]	bist_report[8]
クワッド用のSERDESステータスレジスタ (4)									
25	qd_25 3	plol	~plol	reserved	reserved	reserved	reserved	reserved	Reserved
26	qd_int_26 4	plol_int	~plol_int	reserved	reserved	reserved	reserved	reserved	Reserved
27	qd_27	reserved	reserved	pll_calib_status[5]	pll_calib_status[4]	pll_calib_status[3]	pll_calib_status[2]	pll_calib_status[1]	pll_calib_status[0]
28	qd_28	reserved	reserved	reserved	reserved	reserved	reserved	reserved	Reserved

1. BA=ベースアドレス(Hex)
 2. 割り込み制御レジスタは、割り込み可能なステータス(int_sts_x)レジスタに関連します。
 3. ステータス・レジスタは関連づけられる割り込み可能なステータス(int_sts_x)レジスタを持ちます。
 4. 割り込み可能なステータス・レジスタ; リードでクリア(関連づけられる制御レジスタとステータス・レジスタがある)
 5. uc_mode ; 8ビットSERDES Onlyと10ビットSERDES Only
 6. BISTはラティスセミコンダクター社内での使用のための内蔵のPRBSジェネレータとチェッカーです。
- 注: 特に明記しないかぎり、デフォルト値は0です。

クワッド単位のSERDES制御レジスタの詳細

表8-36 PCS制御レジスタ1 (QD_00)

Bit	Name	記 述	タイプ	デフォルト
7	sync_toggle	遷移 = Txレーン間スキューを最少化するためにTxシリアルライザをリセット レベル = Txシリアルライザの通常動作		
6	force_int	1 = 割り込み信号を強制的に生成する 0 = 通常動作	R/W	0
5	char_mode	1 = SERDES キャラクタライズ・モードをイネーブル 0 = SERDES キャラクタライズ・モードをディセーブル	R/W	0
4	xge_mode	1 = 10Gb Ethernetモードを選択 0 = 1Gb Ethernetモードを選択	R/W	0
3	rio_mode	1 = Rapid-IOモードを選択 0 = その他モード(10GbE, 1GbE)を選択	R/W	0
2	pcie_mode	1 = PCI Expressモードの動作 0 = その他モード(RapidIO, 10GbE, 1GbE)を選択	R/W	0
1	fc_mode	1 = Fibre Channelモードを選択 0 = その他モード(PCI Express, RapidIO, 10GbE, 1GbE)を選択	R/W	0
0	uc_mode	1 = User Configuredモードの選択 0 = その他モード(PCI Express, RapidIO, 10GbE, 1GbE)を選択		

注: 明示される場合を除いて、いづれかのSERDES制御レジスタにライトした後は、全チャンネルをリセットしなければなりません。

表8-37 PCS制御レジスタ2 (QD_01)

Bit	Name	記 述	タイプ	デフォルト
7:6	bist_rpt_ch_sel [1:0]	00 = BISTレポート、チャンネル0から 01 = BISTレポート、チャンネル1から 10 = BISTレポート、チャンネル2から 11 = BISTレポート、チャンネル3から	R/W	00
5:4	bist_res_sel [1:0]	BIST resolution の選択 00 = エラーなし 01 < 2エラー 10 < 16エラー 11 < 128エラー	R/W	00
3:2	bist_time_sel [1:0]	BIST time の選択: 00 = 5e+8 サイクル 01 = 5e+9 サイクル 10 = 5e+6 サイクル 11 = 100K サイクル	R/W	00
1:0	bist_head_sel [1:0]	BIST header の選択 00 = K28_5 (K28_5=10'h305 K28_5_=10'h0FA) 01 = A1A2 (MA1=10'h1F6; MA2=10'h128) 10 = 10'h1BC 11 = ユーザ定義	R/W	00

表8-38 PCS制御レジスタ3 (QD_02)

Bit	Name	記 述	タイプ	デフォルト
7:4	high_mark [3:0]	クロック補償FIFO、high water mark。 平均は 4'b1000	R/W	4'b0111
3:0	low_mark [3:0]	クロック補償FIFO、low water mark。 平均は 4'b1000	R/W	4'b1001

表8-39 PCS制御レジスタ4 (QD_03)

Bit	Name	記 述	タイプ	デフォルト
7:6	min_ipg_cnt [1:0]	最少IPG to enforce	R/W	2'b11
5	match_4_enable	1 = 4キャラクのスキップマッチングをイネーブル (match 4, 3, 2, 1を使用)	R/W	0
4	match_2_enable	1 = 2キャラクのスキップマッチングをイネーブル (match 4,3を使用)	R/W	1
3	Reserved			
2	pfifo_clr_sel	1 = pfifo_clr信号、又はチャネルレジスタビットが FIFOをクリア 0 = pfifo_error内部信号が自らFIFOをクリア	R/W	0
1	asyn_mode	テスト用途のみ。非同期リセットを選択	R/W	0
0	sel_test_clk	テスト用途のみ。テストクロックを選択	R/W	0

表8-40 PCS制御レジスタ5 – CC match1 LO (QD_04)

Bit	Name	記 述	タイプ	デフォルト
7:0	cc_match_1 [7:0]	ユーザ定義クロック補償のスキップパターン1、下位バイト	R/W	8'h00

表8-41 PCS制御レジスタ6 – CC match2 LO (QD_05)

Bit	Name	記 述	タイプ	デフォルト
7:0	cc_match_2 [7:0]	ユーザ定義クロック補償のスキップパターン2、下位バイト	R/W	8'h00

表8-42 PCS制御レジスタ7 – CC match3 LO (QD_06)

Bit	Name	記 述	タイプ	デフォルト
7:0	cc_match_3 [7:0]	ユーザ定義クロック補償のスキップパターン3、下位バイト	R/W	8'hBC

表8-43 PCS制御レジスタ8 – CC match4 LO (QD_07)

Bit	Name	記 述	タイプ	デフォルト
7:0	cc_match_4 [7:0]	ユーザ定義クロック補償のスキップパターン4、下位バイト	R/W	8'h50

表8-44 PCS制御レジスタ9 – CC match HI (QD_08)

Bit	Name	記 述	タイプ	デフォルト
7:6	cc_match_4 [9:8]	ユーザ定義クロック補償のスキップパターン4、上位バイト [9] = ディスパリティエラーerror [8] = Kコントロール	R/W	2'b01
5:4	cc_match_3 [9:8]	ユーザ定義クロック補償のスキップパターン3、上位バイト [9] = ディスパリティエラーerror [8] = Kコントロール	R/W	2'b01
3:2	cc_match_2 [9:8]	ユーザ定義クロック補償のスキップパターン2、上位バイト [9] = ディスパリティエラーerror [8] = Kコントロール	R/W	2'b00
1:0	cc_match_1 [9:8]	ユーザ定義クロック補償のスキップパターン1、上位バイト [9] = ディスパリティエラーerror [8] = Kコントロール	R/W	2'b00

表8-45 PCS制御レジスタ10 – UDF comma mask LO (QD_09)

Bit	Name	記 述	タイプ	デフォルト
7:0	udf_comma_mask [7:0]	ユーザ定義コンママスク、下位バイト	R/W	8'hFF

表8-46 PCS制御レジスタ11 – UDF comma a LO (QD_0A)

Bit	Name	記 述	タイプ	デフォルト
7:0	udf_comma_a [7:0]	ユーザ定義コンマキャラクタ "a"、下位バイト	R/W	8'h83

表8-47 PCS制御レジスタ12 – UDF comma b LO (QD_0B)

Bit	Name	記 述	タイプ	デフォルト
7:0	udf_comma_b [7:0]	ユーザ定義コンマキャラクタ "b"、下位バイト	R/W	8'h7C

表8-48 PCS制御レジスタ13 – UDS comma HI (QD_0C)

Bit	Name	記 述	タイプ	デフォルト
7:6	udf_comma_a [9:8]	ユーザ定義コンマキャラクタ "a"、上位ビット	R/W	2'b10
5:4	udf_comma_b [9:8]	ユーザ定義コンマキャラクタ "b"、上位ビット	R/W	2'b01
3:2	udf_comma_mask [9:8]	ユーザ定義コンママスク、上位ビット	R/W	2'b11
1	bist_mode	1 = 連続BISTモード 0 = "Timed BIST" モード	R/W	0
0	bist_en	1 = PCS BISTをイネーブル 0 = 通常動作	R/W	0

表8-49 PCS制御レジスタ14 – UDF BIST header LO (QD_0D)

Bit	Name	記 述	タイプ	デフォルト
7:0	bist_udf_def_header [7:0]	ユーザ定義BISTヘッダの下位バイト	R/W	8'h00

表8-50 PCS制御レジスタ15 – UDF BIST header MD (QD_0E)

Bit	Name	記 述	タイプ	デフォルト
7:0	bist_udf_def_header [15:8]	ユーザ定義BISTヘッダの中間バイト	R/W	8'h00

表8-51 PCS制御レジスタ16 – UDF BIST headerHID (QD_0F)

Bit	Name	記 述	タイプ	デフォルト
7	bist_bus8bit_sel	1 = 8ビットデータBIST 0 = 10ビットデータBIST R/W 0	R/W	0
6:4	bist_ptn_sel[2:0]	BISTパターンの選択 000 = PRBS11 001 = 最大データレート 010 = PRBS31 011 = PRBS21100 = K28_5 101 = A1A2110 = 5150 (繰り返し) 111 = 2120 (繰り返し)	R/W	00
3:0	bist_udf_def_header [19:16]	ユーザ定義BISTヘッダの上位ビット	R/W	3'b000

表8-52 PCS割り込み制御レジスタ17 (QD_10)

Bit	Name	記 述	タイプ	デフォルト
7	ls_sync_status_3_int_ctl	1 = ls_sync_status_3 割り込みイネーブル(同期中) 0 = ls_sync_status_3 割り込みディセーブル(同期中)	R/W	0
6	ls_sync_status_2_int_ctl	1 = ls_sync_status_2 割り込みイネーブル(同期中) 0 = ls_sync_status_2 割り込みディセーブル(同期中)	R/W	0
5	ls_sync_status_1_int_ctl	1 = ls_sync_status_1 割り込みイネーブル(同期中) 0 = ls_sync_status_1 割り込みディセーブル(同期中)	R/W	0
4	ls_sync_statusn_3_int_ctl	1 = ls_sync_statusn_3 がLowになるときの割り込みイネーブル(同期外) 0 = ls_sync_statusn_3 がLowになるときの割り込みディセーブル(同期外)	R/W	0
3	ls_sync_status_0_int_ctl	1 = ls_sync_status_0 割り込みイネーブル(同期中) 0 = ls_sync_status_0 割り込みディセーブル(同期中)	R/W	0
2	ls_sync_statusn_2_int_ctl	1 = ls_sync_statusn_2 がLowになるときの割り込みイネーブル(同期外) 0 = ls_sync_statusn_2 がLowになるときの割り込みディセーブル(同期外)	R/W	0
1	ls_sync_statusn_1_int_ctl	1 = ls_sync_statusn_1 がLowになるときの割り込みイネーブル(同期外) 0 = ls_sync_statusn_1 がLowになるときの割り込みディセーブル(同期外)	R/W	0
0	ls_sync_statusn_0_int_ctl	1 = ls_sync_statusn_0 がLowになるときの割り込みイネーブル(同期外) 0 = ls_sync_statusn_0 がLowになるときの割り込みディセーブル(同期外)	R/W	0

クワッド単位のSERDES制御レジスタの詳細

注: 明示される場合を除いて、いずれかのSERDES制御レジスタにライトした後は、全チャネルをリセットしなければなりません。

表8-53 SERDES制御レジスタ1 (QD_11)

Bit	Name	記 述	タイプ	デフォルト
7:6	Reserved			
5	TX_REFCK_SEL	TXPLL 基準クロックの選択 0 = refclk (差動入力) 1 = core txrefclk	R/W	0
4	REFCK_DCC_EN	1 = 基準クロックのDC結合をイネーブル 0 = 基準クロックのDC結合をディセーブル(デフォルト)	R/W	0
3	REFCK_RTERM	基準クロック入力バッファの終端 0 = 50 Ω (デフォルト) 1 = ハイインピーダンス	R/W	0
2:0	REFCL_OUT_SEL [2:0]	Refck制御 [0]: 0 = refckバッファのイネーブル、1 = ディセーブル [1]: 0 = refck2coreディセーブル、1 = イネーブル [2]: reserved 図8-47を参照	R/W	3'b000

図8-47 基準クロックの選択制御

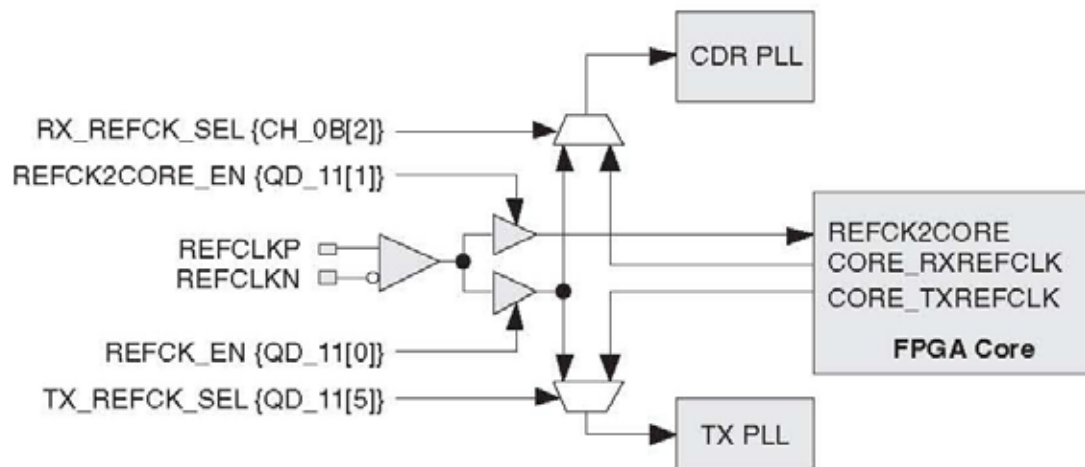


表8-54 SERDES制御レジスタ2 (QD_12)

Bit	Name	記 述	タイプ	デフォルト
7	REFCK25X	1 = 内部高速ビットクロックは25x (基準クロックは100MHzのみ) 0 = REFCK_MODEを参照	R/W	0
6	BUS8BIT_SEL	1 = 8ビットバス幅を選択 0 = 10ビットバス幅を選択(デフォルト)	R/W	0
5:3	RLOS_HSET [2:0]	LOS検出器の大振幅に対する基準電流の調整 000 = デフォルト 001 = +10% 010 = +15% 011 = +25% 100 = -10% 101 = -15% 110 = -25% 111 = -30%	R/W	3'b000
2:0	RLOS_LSET [2:0]	(ラティス社内用途のみ)	R/W	3'b000

表8-55 SERDES制御レジスタ3 (QD_13)

Bit	Name	記 述	タイプ	デフォルト
7:6	REFCK_MODE [1:0]	00 = 内部高速ビットクロックは 20x か 16x 01 = 内部高速ビットクロックは 10x か 8x 1X = Reserved	R/W	2'b00
5:2	Reserved			
1:0	CDR_LOL_SET [1:0]	CDRロックはずれの設定 ロック アンロック 00 = +/-1000ppm x2 +/-1500ppm x2 01 = +/-2000ppm x2 +/-2500ppm x2 10 = +/-4000ppm +/-7000ppm 11 = +/-300ppm +/-450ppm	R/W	2'b10 ¹

1. ispLEVER7.0SP1からのデフォルト値

表8-56 SERDES制御レジスタ4 (QD_14)

Bit	Name	記 述	タイプ	デフォルト
7:5	Reserved			
4:3	PLL_LOL_SET [1:0]	Tx PLLロックはずれの設定 ロック アンロック 00 = +/-300ppm x2 +/-600ppm x2 01 = +/-300ppm +/-2000ppm 10 = +/-1500ppm +/-2200ppm 11 = +/-4000ppm +/-6000ppm	R/W	2'b00
2:0	TX_VCO_CK_DIV[2:0]	VCO出力周波数の選択 000 = 1 分周 001 = reserved 010 = 2 分周 011 = reserved 100 = 4 分周 101 = 8 分周 110 = 16 分周 111 = 32 分周	R/W	3'b000

表8-57 SERDES制御レジスタ5 (QD_15)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

表8-58 SERDES割り込み制御レジスタ6 (QD_16)

Bit	Name	記 述	タイプ	デフォルト
7	PLOL_INT_CTL	1 = PLOLのロックはずれ割り込みをイネーブル 0 = PLOLのロックはずれ割り込みをイネーブルしない	RO CR	0
6	~PLOL_INT_CTL	1 = PLOLのロック時の割り込みをイネーブル 0 = PLOLのロック時の割り込みをイネーブルしない	RO CR	0
5:0	Reserved			

クワッド単位のリセットとクロック制御レジスタの詳細

表8-59 リセットとクロック制御レジスタ1 (QD_17)

Bit	Name	記 述	タイプ	デフォルト
7	lane_rx_rst3	1 = チャンネル3 受信ロジックヘリセット信号をアサート	R/W	0
6	lane_rx_rst2	1 = チャンネル2 受信ロジックヘリセット信号をアサート	R/W	0
5	lane_rx_rst1	1 = チャンネル1 受信ロジックヘリセット信号をアサート	R/W	0
4	lane_rx_rst0	1 = チャンネル0 受信ロジックヘリセット信号をアサート	R/W	0
3	lane_tx_rst3	1 = チャンネル3 送信ロジックヘリセット信号をアサート	R/W	0
2	lane_tx_rst2	1 = チャンネル2 送信ロジックヘリセット信号をアサート	R/W	0
1	lane_tx_rst1	1 = チャンネル1 送信ロジックヘリセット信号をアサート	R/W	0
0	lane_tx_rst0	1 = チャンネル0 送信ロジックヘリセット信号をアサート	R/W	0

表8-60 リセットとクロック制御レジスタ2 (QD_18)

Bit	Name	記 述	タイプ	デフォルト
7	macropdb	0 = パワーダウンをアサート	R/W	1
6	macro_rst	1 = マクロリセットをアサート	R/W	0
5	quad_rst	1 = クワッドリセットをアサート	R/W	0
4	trst	1 = Txリセット	R/W	0
3:0	rrst [3:0]	1 = Rxチャンネルベースのリセット	R/W	0

表8-61 リセットとクロック制御レジスタ3 (QD_19)

Bit	Name	記 述	タイプ	デフォルト
7	bist_rx_data_sel	0 = SERDESからのデータ 1 = 8b10bデコーダの後からのデータ	R/W	0
6	bist_bypass_tx_gate	0 = ヘッダ検出後にBISTデータ送出を開始 1 = ヘッダ検出に係わらずBISTデータ強制送出を開始	R/W	0
5:4	bist_sync_head_req [1:0]	BIST同期ヘッダカウンタの選択 00 = 5'd5 01 = 5'd8 10 = 5'd14 11 = 5'd24	R/W	00
3	sel_sd_rx_clk3	1 = DS FIFOライトクロックにrx_clk3 ¹ を選択 0 = DS FIFOライトクロックにff_ebrd_clk_3を選択	R/W	0
2	sel_sd_rx_clk2	1 = DS FIFOライトクロックにrx_clk2 ¹ を選択 0 = DS FIFOライトクロックにff_ebrd_clk_2を選択	R/W	0
1	sel_sd_rx_clk1	1 = DS FIFOライトクロックにrx_clk1 ¹ を選択 0 = DS FIFOライトクロックにff_ebrd_clk_1を選択	R/W	0
0	sel_sd_rx_clk0	1 = DS FIFOライトクロックにrx_clk0 ¹ を選択 0 = DS FIFOライトクロックにff_ebrd_clk_0を選択	R/W	0

1. rx_clk3はチャネル3の再生クロック

表8-62 リセットとクロック制御レジスタ4 (QD_1A)

Bit	Name	記 述	タイプ	デフォルト
7	ff_rx_clk_sel3_2	1 = チャネル3にff_rx_f_clkをディセーブル	R/W	0
6	ff_rx_clk_sel2_2	1 = チャネル2にff_rx_f_clkをディセーブル	R/W	0
5	ff_rx_clk_sel1_2	1 = チャネル1にff_rx_f_clkをディセーブル	R/W	0
4	ff_rx_clk_sel0_2	1 = チャネル0にff_rx_f_clkをディセーブル	R/W	0
3	ff_rx_clk_sel3_1	1 = チャネル3にff_rx_h_clkをイネーブル	R/W	0
2	ff_rx_clk_sel2_1	1 = チャネル2にff_rx_h_clkをイネーブル	R/W	0
1	ff_rx_clk_sel1_1	1 = チャネル1にff_rx_h_clkをイネーブル	R/W	0
0	ff_rx_clk_sel0_1	1 = チャネル0にff_rx_h_clkをイネーブル	R/W	0

表8-63 リセットとクロック制御レジスタ5 (QD_1B)

Bit	Name	記 述	タイプ	デフォルト
7	ff_tx_clk_sel2	1 = クワッドにff_tx_f_clkをディセーブル	R/W	0
6	ff_tx_clk_sel1	1 = クワッドにff_tx_h_clkをイネーブル	R/W	0
5	ff_tx_clk_sel0	1 = クワッドにff_tx_q_clkをイネーブル	R/W	0
4	Reserved		R/W	0
3	ff_rx_clk_sel3_0	1 = チャネル3にff_rx_q_clkをイネーブル	R/W	0
2	ff_rx_clk_sel2_0	1 = チャネル2にff_rx_q_clkをイネーブル	R/W	0
1	ff_rx_clk_sel1_0	1 = チャネル1にff_rx_q_clkをイネーブル	R/W	0
0	ff_rx_clk_sel0_0	1 = チャネル0にff_rx_q_clkをイネーブル	R/W	0

クワッド単位のPCSステータス・レジスタ詳細

表8-64 PCSステータス・レジスタ1 (QD_20)

Bit	Name	記 述	タイプ	Int?
7:6	Reserved			
5	ion_delay	tri_ionからの遅延されたグローバルresetn	RO	N
4	int_qua_out	クワッド単位の割り込みステータス	RO	N
3:0	int_cha_out [3:0]	チャンネル単位の割り込みステータス	RO	N

表8-65 PCSステータス・レジスタ2 (QD_21)

Bit	Name	記 述	タイプ	Int?
7	ls_sync_status_3	1 = sync_status_3でアラームを生成 0 = sync_status_3でアラームは生成しない	RO	Y
6	ls_sync_status_2	1 = sync_status_2でアラームを生成 0 = sync_status_2でアラームは生成しない	RO	Y
5	ls_sync_status_1	1 = sync_status_1でアラームを生成 0 = sync_status_1でアラームは生成しない	RO	Y
4	ls_sync_status_0	1 = sync_status_0でアラームを生成 0 = sync_status_0でアラームは生成しない	RO	Y
3	Ls_sync_statusn_3	1 = sync_status_3がLowに遷移する際にアラームを生成(同期外) 0 = sync_status_3がLowに遷移する際にアラームを生成しない(同期外)	RO	Y
2	Ls_sync_statusn_2	1 = sync_status_2がLowに遷移する際にアラームを生成(同期外) 0 = sync_status_2がLowに遷移する際にアラームを生成しない(同期外)	RO	Y
1	Ls_sync_statusn_1	1 = sync_status_1がLowに遷移する際にアラームを生成(同期外) 0 = sync_status_1がLowに遷移する際にアラームを生成しない(同期外)	RO	Y
0	Ls_sync_statusn_0	1 = sync_status_0がLowに遷移する際にアラームを生成(同期外) 0 = sync_status_0がLowに遷移する際にアラームを生成しない(同期外)	RO	Y

表8-66 パケット割り込みステータス・レジスタ3 (QD_22)

Bit	Name	記 述	タイプ	Int?
7	ls_sync_status_3_int	1 = sync_status_3で割り込みを生成(同期中) 0 = sync_status_3で割り込みは生成しない(同期中)	RO CR	Y
6	ls_sync_status_2_int	1 = sync_status_2で割り込みを生成(同期中) 0 = sync_status_2で割り込みは生成しない(同期中)	RO CR	Y
5	ls_sync_status_1_int	1 = sync_status_1で割り込みを生成(同期中) 0 = sync_status_1で割り込みは生成しない(同期中)	RO CR	Y
4	ls_sync_status_0_int	1 = sync_status_0で割り込みを生成(同期中) 0 = sync_status_0で割り込みは生成しない(同期中)	RO CR	Y
3	ls_sync_statusn_3_int	1 = sync_status_3がLowに遷移する際に割り込みを生成(同期外) 0 = sync_status_3がLowに遷移する際に割り込みを生成しない(同期外)	RO CR	Y
2	ls_sync_statusn_2_int	1 = sync_status_2がLowに遷移する際に割り込みを生成(同期外) 0 = sync_status_2がLowに遷移する際に割り込みを生成しない(同期外)	RO CR	Y
1	ls_sync_statusn_1_int	1 = sync_status_1がLowに遷移する際に割り込みを生成(同期外) 0 = sync_status_1がLowに遷移する際に割り込みを生成しない(同期外)	RO CR	Y
0	ls_sync_statusn_0_int	1 = sync_status_0がLowに遷移する際に割り込みを生成(同期外) 0 = sync_status_0がLowに遷移する際に割り込みを生成しない(同期外)	RO CR	Y

表8-67 PCS BISTステータス・レジスタ4 (QD_23)

Bit	Name	記 述	タイプ	Int?
7:0	bist_report [7:0]	BISTレポート下位バイト	RO	N

表8-68 PCS BISTステータス・レジスタ5 (QD_24)

Bit	Name	記 述	タイプ	Int?
7:0	bist_report [15:8]	BISTレポート上位バイト	RO	N

クワッド単位のSERDESステータス・レジスタ詳細

表8-69 SERDESステータス・レジスタ1 (QD_25)

Bit	Name	記 述	タイプ	デフォルト
7	PLOL	1 = PLLロックはずれ	RO	Y
6	~PLOL	1 = PLLロック獲得	RO	Y
5:0	Reserved			

表8-70 SERDES割り込みステータス・レジスタ2 (QD_26)

Bit	Name	記 述	タイプ	デフォルト
7	PLOL_INT	1 = PLOLで割り込み生成 0 = PLOLで割り込み生成されない	RO CR	Y
6	~PLOL_INT	1 = ~PLOLで割り込み生成 0 = ~PLOLで割り込み生成されない	RO CR	Y
5:0	Reserved			

表8-71 SERDESステータス・レジスタ3 (QD_27)

Bit	Name	記 述	タイプ	デフォルト
7:6	Reserved			
5:0	PLL_CALIB_STATUS [5:0]	TxPLL VCO キャリブレーション・ステータス出力	RO	N

表8-72 SERDESステータス・レジスタ4 (QD_28)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

チャネル単位のレジスタ概要

表8-73 チャネル・インタフェースのレジスタマップ

BA 1	レジスタ名	D7	D6	D5	D4	D3	D2	D1	D0
1. 制御レジスタ (13)									
チャネル単位の制御レジスタ (7)									
00	ch_00	enable_cg_align	prbs_enable 6	prbs_lock	ge_an_enable			invert_tx	invert_rx
01	ch_01	pfifo_clr	pcie_ei_en	pcs_det_time_sel[1]	pcs_det_time_sel[0]	rx_gear_mode	tx_gear_mode	rx_ch	tx_ch
02	ch_02	bus_width8	sb_bypass	sb_pfifo_lp	sb_bist_sel	enc_bypass	sel_bist_txd4enc	tx_gear_bypass	fb_loopback
03	ch_03	lsm_sel	signal_detect	rx_gear_bypass	ctc_bypass	dec_bypass	wa_bypass	rx_sb_bypass	sb_loopback
04	ch_int_04 2					cc_underrun_int_ctl	cc_overrun_int_ctl	fb_rx_fifo_error_int_ctl	fb_tx_fifo_error_int_ctl
05	ch_05								los_hi_sel
06	ch_06								
チャネル単位のSERDESレジスタ (6)									
07	ch_07	req_en	req_lv1_set	rcv_dcc_en	rate_sel[1]	rate_sel[0]	rx_dco_ck_div[2]	rx_dco_ck_div[1]	rx_dco_ck_div[0]
08	ch_08	lb_ctl[3]	lb_ctl[2]	l b_ctl[1]	lb_ctl[0]	rterm_rxadj[1]	rterm_rxadj[0]	rterm_rx[1]	rterm_rx[0]
09	ch_09	tdrv_amp[2]	tdrv_amp[1]	tdrv_amp[0]	tdrv_pre_set[2]	tdrv_pre_set[1]	tdrv_pre_set[0]	tdrv_dat_sel[1]	tdrv_dat_sel[0]
0A	ch_0a				tdrv_pre_en	rterm_tx[1]	rterm_tx[0]	rate_mode_tx	tpwdnb
0B	ch_0b				oob_en	rx_refck_sel[1]	rx_refck_sel[0]	rate_mode_rx	rpwdnb
0C	ch_int_0c 2		pci_det_done_int_ctl	rlos_lo_int_ctl	~rlos_lo_int_ctl	rlos_hi_int_ctl	~rlos_hi_int_ctl	rlol_int_ctl	~rlol_int_ctl
2. ステータス・レジスタ (13)									
チャネル単位の汎用レジスタ (6)									
20	ch_20 3					cc_underrun	cc_overrun	fb_rx_fifo_error	fb_tx_fifo_error
21	ch_21 5	prbs_error_cnt[7]	prbs_error_cnt[6]	prbs_error_cnt[5]	prbs_error_cnt[4]	prbs_error_cnt[3]	prbs_error_cnt[2]	prbs_error_cnt[1]	prbs_error_cnt[0]
22	ch_22	fb_tx_fifo_error_int	fb_rx_fifo_error_int			cc_underrun_int	cc_overrun_int	fb_rx_fifo_error_int	fb_tx_fifo_error_int
23	ch_int_23		ffs_ls_sync_stat	us fb_rrst_o	fb_txrst_o			cc_re_o	cc_we_o
24	ch_24								
25	ch_25								
チャネル単位のSERDESレジスタ (7)									
26	ch_26 3		pci_det_done	rlos_lo	~rlos_lo	rlos_hi	~rlos_hi	rlol	~rlol
27	ch_27	dco_calib_err	dco_calib_done	dco_facq_err	dco_facq_done				pci_connect
28	ch_28	dco_status[7]	dco_status[6]	dco_status[5]	dco_status[4]	dco_status[3]	dco_status[2]	dco_status[1]	dco_status[0]
29	ch_29	dco_status[15]	dco_status[14]	dco_status[13]	dco_status[12]	dco_status[11]	dco_status[10]	dco_status[9]	dco_status[8]
2A	ch_ch_2a 4		pci_det_done_int	rlos_lo_int	~rlos_lo_int	rlos_hi_int	~rlos_hi_int	rlol_int	~rlol_int
2B	ch_2b								
2C	ch_2c								

1. Ba=ベースアドレス(Hex)
2. 割り込み制御レジスタは割り込み可能なステータス(int_sts_x)レジスタに関連づけられます。
3. ステータス・レジスタには関連づけられる割り込み可能なステータス(int_sts_x)レジスタがあります。
4. 割り込み可能なステータス・レジスタ 1 ; (関連づけられる制御レジスタとステータス・レジスタ) リードでクリア。
5. ステータス・レジスタ、リードでクリア。
6. PRBSは内蔵のロジックによって生成されて、ラティスセミコンダクター社内のテスト用のためのものです。
注: 特に明記しないかぎり、デフォルト値は0です。

表8-74 PCS制御レジスタ1 (CH_00)

Bit	Name	記 述	タイプ	デフォルト
7	enable_cg_align	uc_modeで動作時のみ有効。 1 = 連続コンマアライメントをイネーブル 0 = 連続コンマアライメントをディセーブル	R/W	0
6	prbs_enable	1 = PRBSジェネレータとチェッカーをイネーブル 0 = 通常動作	R/W	0
5	prbs_lock	1 = Lock 受信PRBSチェッカーをロック 0 = Lock 受信PRBSチェッカーをアンロック	R/W	0
4	ge_an_enable	1 = GigE自動ネゴシエーションをイネーブル 0 = GigE自動ネゴシエーションをディセーブル	R/W	0
3:2	Reserved			
1	invert_tx	1 = 送信データを反転 0 = 送信データを反転しない	R/W	0
0	invert_rx	1 = 受信データを反転 0 = 受信データを反転しない	R/W	0

表8-75 PCS制御レジスタ2 (CH_01)

Bit	Name	記 述	タイプ	デフォルト
7	pfifo_clr	1 = クワッドレジスタ・ビット pfifo_clr_sel が 1 にセットされているとPFIFOをクリアする。これは内部信号 pfifo_clr と OR されている。 0 = 通常動作	R/W	0
6	pcie_ei_en	1 = PCI Express 電気的アイドル 0 = 通常動作	R/W	0
5:4	pcs_det_time_sel[1:0]	PCS接続検出時間 11 = 16us 10 = 4us 01 = 2us 00 = 8us	R/W	0
3	rx_gear_mode	1 = 全チャネルの受信パスで 2:1 ギアリングをイネーブル 0 = 全チャネルの受信パスで 2:1 ギアリングをディセーブル(ギアリングなし)	R/W	0
2	tx_gear_mode	1 = 全チャネルの送信パスで 2:1 ギアリングをイネーブル 0 = 全チャネルの送信パスで 2:1 ギアリングをディセーブル(ギアリングなし)	R/W	0
1	rx_ch	1 = 受信出力がテスト・キャラクタライズ用ピンでモニタ可能にする。テスト・キャラクタライズ・モード (pcs_ctl_4_qd_03[6])は 1 にする。	R/W	0
0	tx_ch	1 = トランスミッタPCS入力をテスト・キャラクタライズ用ポートから供給可能にする。テスト・キャラクタライズ・モードをイネーブルにする。	R/W	0

表8-76 PCS制御レジスタ3 (CH_02)

Bit	Name	記 述	タイプ	デフォルト
7	bus_width8	1 = PCSとSERの間は8ビットバス 0 = PCSとSERの間は10ビットバス	R/W	0
6	sb_bypass	1 = Tx SERDESブリッジをバイパス 0 = 通常動作	R/W	0
5	sb_pfifo_lp	1 = RxからTxへFIFOを介してパラレルループバックをイネーブル 0 = 通常データ動作	R/W	0
4	sb_bist_sel	1 = BISTデータを選択 0 = 通常データを選択	R/W	0
3	enc_bypass	1 = 8b10bエンコードをバイパス 0 = 通常動作	R/W	0
2	sel_bist_txd4enc	1 = 8b10bエンコードの前で送信側にBISTデータをイネーブル 0 = 通常動作	R/W	0
1	tx_gear_bypass	1 = PCS Txギアボックスをバイパス 0 = 通常動作	R/W	0
0	fb_loopback	1 = PCS内FPGAブリッジの前で、RxからTxのループバックをイネーブル 0 = 通常データ動作	R/W	0

表8-77 PCS制御レジスタ4 (CH_03)

Bit	Name	記 述	タイプ	デフォルト
7	lsm_sel	1 = ワードアライメントに外部LSMを選択 0 = ワードアライメントに内部LSMを選択	R/W	0
6	signal_detect	1 = Rx LSMを強制的にイネーブル 0 = ffc_signal_detect 信号に依存してRx LSMをイネーブル	R/W	0
5	rx_gear_bypass	1 = PCS Rxギアボックスをバイパス 0 = 通常動作	R/W	0
4	ctc_bypass	1 = CTCをバイパス 0 = 通常動作	R/W	0
3	dec_bypass	1 = 8b10bデコードをバイパス 0 = 通常動作	R/W	0
2	wa_bypass	1 = ワードアライメントをバイパス 0 = 通常動作	R/W	0
1	rx_sb_bypass	1 = Rx SERDESブリッジをバイパス 0 = 通常動作	R/W	0
0	sb_loopback	1 = PCSのSERDESブリッジ内でTxからRxへのループバックをイネーブル 0 = 通常データ動作	R/W	0

表8-78 PCS 割り込み制御レジスタ5 (CH_04)

Bit	Name	記 述	タイプ	デフォルト
7:4	Reserved			
3	cc_underrun_int_ctl	1 = cc_underrunの割り込みをイネーブル 0 = cc_underrunの割り込みをディセーブル	R/W	0
2	cc_overrun_int_ctl	1 = cc_overrunの割り込みをイネーブル 0 = cc_overrunの割り込みをディセーブル	R/W	0
1	fb_rx_fifo_error_int_ctl	1 = 受信FPGAブリッジFIFOのエンプティ/フル条件での割り込みをイネーブル	R/W	0
0	fb_tx_fifo_error_int_ctl	1 = 送信FPGAブリッジFIFOのエンプティ/フル条件での割り込みをイネーブル	R/W	0

表8-79 PCS 制御レジスタ6 (CH_05)

Bit	Name	記 述	タイプ	デフォルト
7:1	Reserved			
0	low_hi_sel	1 = rlos_hi を選択 0 = rlos_lo を選択(ラティス社内での用途のみ)	R/W	0

表8-80 PCS 制御レジスタ7 (CH_06)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

チャネル単位のSERDES制御レジスタ詳細

注；特に明記しない限り、SERDES制御レジスタにライトアクセスした値を有効にするには、全チャネルをリセットすること。

表8-81 SERDES制御レジスタ1 (CH_07)

Bit	Name	記 述	タイプ	デフォルト
7	REQ_EN	1 = 受信イコライザをイネーブル 0 = 受信イコライザをディセーブル	R/W	0
6	REQ_LVL_SET	イコライザのレベル設定 1 = ロングリーチ用等化 0 = 中距離ルート用の等化	R/W	0
5	RCV_DCC_EN	1 = レシーバDCカップリングをイネーブル 0 = ACカップリング (デフォルト)	R/W	0
4:3	RATE_SEL [1:0]	イコライザの極配置を選択 00 = 高域周波レンジ用の極配置 01 = 中域周波レンジ用の極配置 10 = 低域周波レンジ用の極配置 11 = 未使用	R/W	2'b00
2:0	RX_DCO_CK_DIV [2:0]	VCO出力周波数選択 000 = 分周比 1 001 = reserved 010 = 分周比 2 011 = reserved 100 = 分周比 4 101 = 分周比 8 110 = 分周比 16 111 = 分周比 32	R/W	3'b000

表8-82 SERDES制御レジスタ2 (CH_08)

Bit	Name	記 述	タイプ	デフォルト
7:4	LB_CTL [3:0]	ループバック制御 [3] = slb_r2t_dat_en, rx --> tx シリアルループバックをイネーブル (CDRデータ) [2] = slb_r2t_ck_en, rx --> tx シリアルループバックをイネーブル (CDRクロック) [1] = slb_eq2t_en, イコライザ --> ドライバ・シリアルループバックをイネーブル [0] = slb_t2r_en, tx --> rx シリアルループバックをイネーブル	R/W	4'h0
3:2	RTERM_RXADJ [1:0]	終端抵抗の補償(compensation) 00 = default 01 = -7% 10 = -14% 11 = -20%	R/W	2'b00
1:0	RTERM_RX [1:0]	00 = 50 ohm 01 = 75 ohm 10 = 2K ohm 11 = 60 ohm	R/W	2'b00

表8-83 SERDES制御レジスタ3 (CH_09)

Bit	Name	記 述			タイプ	デフォルト		
LatticeECP2M-35用SERDES制御レジスタ3								
7:5	TDRV_AMP[2:0]	CMLドライバの振幅設定 (mV), VCCOB = 1.2V 000 = 1040(デフォルト) 001 = 1280 010 = 1320 011 = 1360 100 = 640 101 = 760 110 = 870 111 = 990			R/W	0		
4:2	TDRV_PRE_SET [2:0]	Txドライバ・プリエン ファシス・レベル設定	% 0 16 36 40 44 56 80		R/W	0		
1:0	TDRV_DAT_SEL [1:0]	ドライバ出力の選択 00 = シリアルライザからのデータを選択 (通常動作) 01 = シリアルライザからのデータクロックをドライバに選択 10 = slb_r2t_dat_en=1 の場合 Rx --> Tx シリアル・ループ バック(データ) 10 = slb_r2t_ck_en=1 の場合 Rx --> Tx シリアル・ループバ ック(クロック) 11 = slb_eq2t_en=1 の場合、イコライザからドライバにシリ アル・ループバック			R/W	0		
LatticeECP2M-20/50/70/100用SERDES制御レジスタ3								
7:5	TDRV_AMP[2:0]	GUIの設定(3-bit) 0 1 2 3 4 5 6 7	レジスタの設定(5-bit) 00101 00001 00010 00011 11100 10100 10110 01110	mV 990 1250 1300 1350 610 730 820 940	R/W	0		
4:3	TDRV_PRE_SET [4:0]	Txドライバ・プリエンファシス・レベル設定 GUIの設定(3-bit) 0 1 2 3 4 5 6			レジスタの設定(5-bit) 00000 00001 00010 01010 10010 00011 00100	% 0 12 26 30 33 40 53	R/W	0

表8-84 SERDES制御レジスタ4 (CH_0A)

Bit	Name	記 述	タイプ	Int?
7				
6:5	TDRV_DAT_SEL [1:0]	ドライバ出力の選択 00 = シリアルライザからのデータを選択 (通常動作) 01 = シリアルライザからのデータクロックをドライバに選択 10 = slb_r2t_dat_en=1 の場合 Rx --> Tx シリアル・ループバック (データ) 10 = slb_r2t_ck_en=1 の場合 Rx --> Tx シリアル・ループバック (クロック) 11 = slb_eq2t_en=1 の場合、イコライザからドライバにシリアル・ループバック	R/W	2'b00
4	TDRV_PRE_EN	1 = Txドライバ・プリエンファシスをイネーブル 0 = Txドライバ・プリエンファシスをディセーブル	R/W	0
3:2	RTERM_TX [1:0]	Tx終端抵抗の選択。PCI Expressが選択されるとディセーブル 00 = 50 ohm (デフォルト) 01 = 75 ohm 10 = 5K ohm	R/W	2'b00
1	RATE_MODE_TX	0 = トランスミッタにフルレートを選択 1 = トランスミッタにハーフレートを選択	R/W	0
0	tpwddb	0 = 送信チャネルをパワーダウン 1 = 送信チャネルをパワーアップ	R/W	0

1. LatticeECP2M-20/50/70/100のみ

表8-85 SERDES制御レジスタ4 (CH_0B)

Bit	Name	記 述	タイプ	デフォルト
7:6	TDRV_AMP[4:3]	送信ドライバ振幅設定上位ビット	R/W	0
5	Reserved	LatticeECP2M-20/50/70/100のみ適用R/W 2'b00		
7:5	Reserved	LatticeECP2M35のみ適用		
4	oob_en	1 = バウンダリスキャン用入力パスを配線としてイネーブルし、FPGA内の低速SERDESに受信入力を供給する(OOB用途)	R/W	0
3:2	rx_refck_sel [1:0]	Rx CDR基準クロックの選択 00 = refclk (差動入力) 01 = core_rxrefclk 1x = reserved	R/W	2'b00
1	RATE_MODE_RX	0 = レシーバにフルレートを選択 1 = レシーバにハーフレートを選択	R/W	0
0	rpwddb	0 = レシーバチャネルをパワーアダウン 1 = レシーバチャネルをパワーアップ	R/W	0

表8-86 TDRV_AMP設定レジスタ ; LatticeECP2M-20/50/70/100

Bit	Name	記 述	タイプ	デフォルト
CH_0B[7:6]	TDRV_AMP[4:3]	表8-86	R/W	0
CH_09 [7:5]	TDRV_AMP[2:0]	表8-84	R/W	0

表8-87 SERDES 割り込み制御レジスタ1 (CH_0C)

Bit	Name	記 述	タイプ	デフォルト
7	Reserved			
6	pci_det_done_int_ctl	1 = PCI Expressでfar-endレシーバ検出割り込みをイネーブル	R/W	0
5	rlos_lo_int_ctl	1 = レシーバ入力レベルがLOW閾値(rlos_lset設定)より小さい場合のRx LOS割り込みをイネーブル	R/W	0
4	~rlos_lo_int_ctl	1 = レシーバ入力レベルがLOW閾値(rlos_lset設定)に等しいか大きい場合のRx LOS割り込みをイネーブル	R/W	0
3	rlos_hi_int_ctl	1 = レシーバ入力レベルがHIGH閾値(rlos_hset設定)より小さい場合のRx LOS割り込みをイネーブル	R/W	0
2	~rlos_hi_int_ctl	1 = レシーバ入力レベルがLOW閾値(rlos_hset設定)に等しいか大きい場合のRx LOS割り込みをイネーブル	R/W	0
1	rlol_int_ctl	1 = レシーバのLOL割り込みをイネーブル	R/W	0
0	~rlol_int_ctl	1 = レシーバがLOLから復帰した際の割り込みをイネーブル	R/W	0

チャンネル毎のPCSステータス・レジスタ詳細

表8-88 PCSステータス・レジスタ1 (CH_20)

Bit	Name	記 述	タイプ	Int?
7:5	Reserved			
4	pfifo_error	1 = パラレルFIFOエラー 0 = パラレルFIFOエラーなし	RO	Y
3	cc_underrun	1 = CC FIFOアンダーラン 0 = CC FIFOアンダーランなし	RO	Y
2	cc_overrun	1 = CC FIFOオーバーラン 0 = CC FIFOオーバーランなし	RO	Y
1	fb_rx_fifo_error	1 = FPGAブリッジ(FB) Rx FIFOオーバーラン 0 = FB Rx FIFOオーバーランなし	RO	Y
0	fb_tx_fifo_error	1 = FB Tx FIFOオーバーラン 0 = FB Tx FIFOオーバーランなし	RO	Y

表8-89 PCSステータス・レジスタ2 (CH_21)

Bit	Name	記 述	タイプ	Int?
7:0	prbs_errors	PRBSエラーカウント数。リードするとクリアされる。FFに達すると値を保持。	RO CR	N

表8-90 PCSステータス・レジスタ3 (CH_22)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

表8-91 PCS一般割り込みステータス・レジスタ4 (CH_23)

Bit	Name	記 述	タイプ	Int?
7:4	Reserved			
3	cc_underrun_int	1 = cc_underrunで割り込みを生成 0 = cc_underrunで割り込みを生成しない	RO CR	Y
2	cc_overrun_int	1 = cc_overrunで割り込みを生成 0 = cc_overrunで割り込みを生成しない	RO CR	Y
1	fb_rx_fifo_error_int	1 = fb_rx_fifo_errorで割り込みを生成 0 = fb_rx_fifo_errorで割り込みを生成しない	RO CR	Y
0	fb_tx_fifo_error_int	1 = fb_tx_fifo_errorで割り込みを生成 0 = fb_tx_fifo_errorで割り込みを生成しない	RO CR	Y

表8-92 PCSステータス・レジスタ5 (CH_24)

Bit	Name	記 述	タイプ	Int?
7	Reserved			
6	ffs_ls_sync_status	1 = LSMはSyncステート 0 = LSMはSyncステートにない	RO	N
5	fb_rxrst_o	1 = 通常動作 0 = FPGAブリッジRxリセット	RO	N
4	fb_txrst_o	1 = 通常動作 0 = FPGAブリッジTxリセット	RO	N
3	Reserved			
2	Reserved			
1	cc_re_o	1 = エラスティックFIFOリードイネーブル 0 = エラスティックFIFOリードディセーブル	RO	N
0	cc_we_o	1 = エラスティックFIFOライトイネーブル 0 = エラスティックFIFOライトディセーブル	RO	N

表8-93 PCSステータス・レジスタ6 (CH_25)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

チャンネル毎のSERDESステータス・レジスタ詳細

表8-94 SERDESステータス・レジスタ1 (CH_26)

Bit	Name	記 述	タイプ	Int?
7	Reserved			
6	pci_det_done	1 = SERDESトランスミッタがレシーバ検出プロセスを完了 0 = SERDESトランスミッタはレシーバ検出プロセスを未完了	RO CR	Y
5	rlos_lo	1 = レシーバで検出した入力信号レベルが設定されたLOW閾値より低いことを示す	RO CR	Y
4	~rlos_lo	1 = レシーバで検出した入力信号レベルが設定されたLOW閾値に等しいか大きいことを示す	RO CR	Y
3	rlos_hi	1 = レシーバで検出した入力信号レベルが設定されたHIGH閾値より低いことを示す	RO CR	Y
2	~rlos_hi	1 = レシーバで検出した入力信号レベルが設定されたHIGH閾値に等しいか大きいことを示す	RO CR	Y
1	rlol	1 = CDRがデータにロックしていない(LOL)で、基準クロックにロックしていることを示す	RO	Y
0	~rlol	1 = CDRがデータにロックしていることを示す	RO	Y

表8-95 SERDESステータス・レジスタ2 (CH_27)

Bit	Name	記 述	タイプ	Int?
7	DCO_CALIB_ERR	1 = DCOキャリブレーションが不正の可能性があることを示す(選択されたL/H境界帯域)	RO	N
6	DCO_CALIB_DONE	1 = DCOキャリブレーション完了を示す	RO	N
5	DCO_FACQ_ERR	1 = DCO周波数アクイジション・エラーを示す(>300ppm)	RO	N
4	DCO_FACQ_DONE	1 = DCO周波数アクイジション完了を示す	RO	N
3:1	Reserved		RO	N
0	pci_connect	1 = SERDESトランスミッタがレシーバを検出(Tx側) 0 = SERDESトランスミッタがレシーバを不検出(Tx側)	RO	N

表8-96 SERDESステータス・レジスタ3 (CH_28)

Bit	Name	記 述	タイプ	Int?
7:0	DCO_STATUS[7:0]	setdcoidac[7:0]	RO	N

表8-97 SERDESステータス・レジスタ4 (CH_29)

Bit	Name	記 述	タイプ	Int?
7:0	DCO_STATUS[15:8]	[1:0] = setdcoidac[9:8] [7:2] = setdcoband[5:0]	RO	N

表8-98 SERDES 割り込みステータス・レジスタ5 (CH_2A)

Bit	Name	記 述	タイプ	Int?
7	Reserved			
6	pci_det_done_int	1 = pci_det_doneで割り込みを生成	RO CR	Y
5	rlos_lo_int	1 = rlos_loで割り込みを生成	RO CR	Y
4	~rlos_lo_int	1 = ~rlos_loで割り込みを生成	RO CR	Y
3	rlos_hi_int	1 = rlos_hiで割り込みを生成	RO CR	Y
2	~rlos_hi_int	1 = ~rlos_hiで割り込みを生成	RO CR	Y
1	rlol_int	1 = rlolで割り込みを生成	RO CR	Y
0	~rlol_int	1 = ~rlolで割り込みを生成	RO CR	Y

表8-99 SERDES ステータス・レジスタ6 (CH_2B)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

表8-100 SERDES ステータス・レジスタ7 (CH_2C)

Bit	Name	記 述	タイプ	デフォルト
7:0	Reserved			

表8-101 異なる規格ごとのK キャラクタ

キャラクタ	Gbe	XAUI	1xFC	PCI Express	RapidIO
K23.7 (F7)	Carrier extend			PAD	
K27.7 (FB)	SOP	ST		Start TLP	A (align)
K28.0 (1C)		SKIP R		SKIP	SC
K28.1 (3C)				FTS	
K28.2 (5C)		SoS		Start DLLP	
K28.3 (7C)		ALIGN A		IDLE	PD
K28.4 (9C)		SEQ			
K28.5 (BC)	+D5.6 or D16.2=IDLE	SYNC K	+D21.4 +D21.5 +D21.5 = IDLE	COMMA (used for alignment)	K
K28.6 (DC)					
K28.7 (FC)					
K29.7 (FD)	EOP	T		END	R (skip)
K30.7 (FE)	ERR	ERR		END BAD	

付録B. 8B10B符号語

表8-102 8B10B符号語

シンボル名(モード組み合わせ表)	シンボル符号(8b10b符号)	10-Bit GUI表記
K28.0	8`b000_11100	0100011100
K28.5	8`b101_11100	0110111100
K29.7	8`b111_11101	0011111101
D16.2	8`b010_10000	0001010000
D21.4	8`b100_10101	0010010101
D21.5	8`b101_10101	0010110101
K28.5+	10`b110000_0101	1100000101
K28.5-	10`b001111_1010	0011111010

表8-103 符号語のラティスマスク

シンボル名(モード組み合わせ表)	シンボル符号(8b10b符号)	10-Bit GUI表記
28.5 (Mask)	10`b111111_1111	1111111111

付録C. 属性のクロスリファレンス

表8-104 属性のクロスリファレンス一覧表

英語版オリジナルを参照して下さい。

付録D. プロトコル依存のSERDES設定オプション

表8-105 プロトコル依存のSERDES設定オプション

プロトコル	DATARATE	DATARATE 範囲	REFCK乗数	DATA WIDTH	Rx等化
GbE	1.25	MED	20X, 10X, 5X	8, 16	DISABLE, MID_MED, LONG_MED
PCI Express	2.5	HIGH	25X, 20X		16
XAUI	3.125	HIGH	20X	8, 16	
Generic 8b10b	ANY_VALUE	LOW, MEDLOW, MED, MEDHIGH, HIGH	20X, 10X, 5X		
8-bit SERDES_Only			16X, 8X, 4X		
10-bit SERDES_Only			20X, 10X, 5X	10, 20	