

概要

LatticeECP3™ FPGA ファミリは、高性能 FPGA ファブリック、高性能 I/O、及び最大 16 チャンネルの組み込み SERDES 及び関連する PCS (Physical Coding Sublayer) ロジックを集積しています。多数の業界標準高速データ転送プロトコルに対応するよう、PCS ロジックを構成することができます。

PCS ロジックの各チャンネルに専用の送受信 SERDES を含み、最大データレート 3.2Gbps 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、GbE、XAUI、SONET/SDH、PCI Express、SRIO、CPRI、OBSAI、SD-SDI、HD-SDI、3G-SDI を含む一連のデータプロトコルに対応するよう、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 ロジックへは 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。それぞれの SERDES ピンは個別に DC カップリングに設定することで、シリアル・デジタルビデオ (SDI) 用途では高速動作と低速動作を、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
 - 汎用 (Generic) 8b10b、10 ビット SERDES 及び 8 ビット SERDES モードで 150Mbps ~ 3.2Gbps。表 8-1 を参照
 - 他のプロトコルでは 230Mbps ~ 3.2Gbps
 - 3.2Gbps 動作時にチャンネルあたり 110mW の低消費電力
 - 小フォームファクタのバックプレーン動作のための受信イコライザと送信プリエンファシス
 - PCI Express、ギガビット・イーサネット (1GbE と SGMII)、XAUI、及び他の複数の規格に対応
 - ユーザ指定の汎用 (Generic) 8b10b モードに対応
 - 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に対応
 - 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に対応
 - デバイス当たり最大 16 チャンネルの全二重データ通信に対応
 - 単一チップで複数プロトコルに対応
 - 8b10b ベースの広く採用されているパケットプロトコルに対応
 - 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
- ・ 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の周波数偏差を補償
 - 1/2/4 バイト長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

LatticeECP2M™ SERDES/PCS の新機能

- ・ 単一の SERDES クワッドで複数のプロトコル規格に対応。表 8-1 に示す対応規格ではフルレートまたはハーフレートで公称周波数となっている必要があります。構成の柔軟さは、様々なプロトコルや規格の混在の障害にはなりません。マルチプロトコル・グルーピングでは、PCI Express、ギガビット・イーサネット、SGMII、及び Serial RapidIO モードに対応します。
- ・ XAUI 準拠機能に対応し、SERDES の最大性能を 3.2Gbps まで拡張
- ・ SONET/SDH OC-3/STM-1、OC-12/STM-4、OC-48/STM-16 の各レートに対応
- ・ SD-SDI、HD-SDI、及び 3G-SDI 用の送受信 DIV11 への対応を追加。マルチレート SDI 対応

本テクニカルノートの使用方

Lattice Diamond® 設計ツールは、PCS の全モードに対応しています。ほとんどのモードは、特定の業界の標準データプロトコル用の用途に専用のものとなっています。それ以外のモードは汎用性があり、設計者は独自のカスタム用途設定が可能です。Diamond 設計ツールでは、ユーザが自分のデザイン内でクワッドごとにモードを定義できます。このテクニカルノートでは、ツールが対応する全てのモードに対する、SERDES 及び PCS の動作について説明します。付録 D を参照してください。

本ドキュメントでは、組み込み SERDES 及び関連する PCS ロジックの全機能について詳しく説明します。組み込み SERDES の電気及びタイミング特性は、LatticeECP3 ファミリー・データシートに記載されています。PCS ロジックの動作は、本ドキュメントの PCS の項で説明します。SCI バスからアクセス可能な SERDES 及び PCS ロジックに関連する、全てのステータス及びコントロールレジスタの一覧表は付録に示してあります。パッケージのピン配列情報は、LatticeECP3 ファミリー・データシートのピン配列情報の項に記載されています。

対応規格

対応する規格を表 8-1 に示します。

表 8-1. SERDES が対応する規格

規格	データレート (Mbps)	システム基準ク ロック (MHz)	FPGA クロック (MHz)	一般的なリンク幅	エンコード方式
PCI Express 1.1	2500	100	250	x1, x2, x4	8b10b
ギガビットイーサネット、SGMII	1250	125	125	x1	8b10b
	2500	125	250	x1	8b10b
	3125	156.25	156.25	x1	8b10b
XAUI	3125	156.25	156.25	x4	8b10b
Serial RapidIO Type I, Serial RapidIO Type II, Serial RapidIO Type III	1250, 2500, 3125	125, 125, 156.25	125, 250, 156.25	x1, x4	8b10b
OBSAI-1, OBSAI-2, OBSAI-3, OBSAI-4	768, 1536, 2304, 3072	76.8, 76.8, 153.6, 115.2, 153.6	76.8, 153.6, 230.4, 153.6	x1	8b10b
CPRI-1, CPRI-2, CPRI-3, CPRI-4	614.4, 1228.8, 2457.6, 3072.0	61.44, 61.44, 122.88, 122.88, 153.6	61.44, 122.88, 122.88, 153.6	x1	8b10b
SD-SDI (259M, 344M)	143 ¹ , 177 ¹ , 270, 360, 540	14.3 ¹ , 17.7 ¹ , 27, 36, 54	143, 177, 27, 36, 54	x1	NRZI/Scrambled

表 8-1. SERDES が対応する規格 (Continued)

規格	データレート (Mbps)	システム基準ク ロック (MHz)	FPGA クロック (MHz)	一般的なリンク幅	エンコード方式
HD-SDI (292M)	1483.5, 1485	74.175, 148.35, 74.25, 148.50	74.175, 148.35, 74.25, 148.5	x1	NRZI/Scrambled
3G-SDI (424M)	2967, 2970	148.35, 148.5	148.35, 148.5	x1	NRZI/Scrambled
SONET STS-3 ² SONET STS-12 ² SONET STS-48 ²	155.52 622.08 2488	15.552 62.208 248.8	15.552 62.208 248.8	x1	N/A
10 ビット SERDES	150 – 3125	15 – 312.5	15 – 312.5	x1, x2, x3, x4	N/A
8 ビット SERDES	150 – 3125	15 – 312.5	15 – 312.5	x1, x2, x3, x4	N/A
汎用 8b10b	150 – 3125	15 – 312.5	15 – 312.5	x1, x2, x3, x4	8b10b

1. 低速時は SERDES がバイパスされ、信号が直接 FPGA コアに送られます。
2. SONET プロトコルには 8 ビット SERDES モードで対応しています。詳細については、本ドキュメントの SONET のセクションを参照してください。

アーキテクチャ概要

SERDES/PCS ブロックは独立する 4 系統の全二重データチャ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。図 8-1 は LatticeECP3-150 FPGA の SERDES/PCS クワッドの配置を示しています (他のデバイスではクワッド数は少なくなります)。

図 8-1. LatticeECP3-150 ブロック図

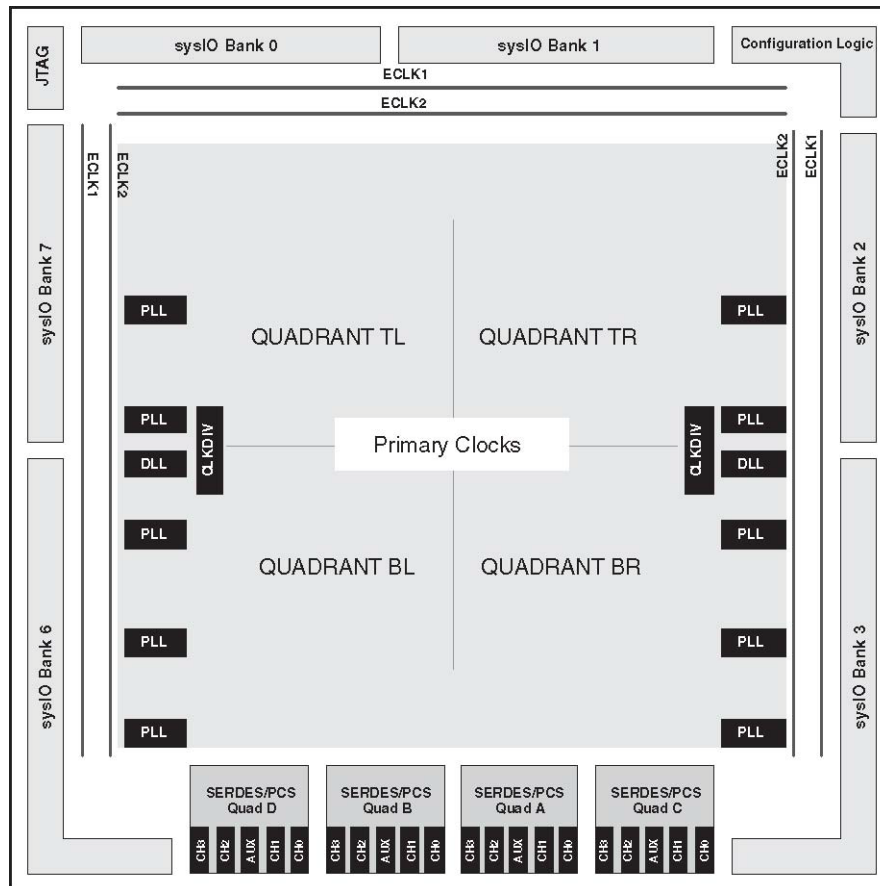


表 8-2 は LatticeECP3 ファミリの各デバイスにある SERDES/PCS クワッドの数を示しています。

表 8-2. LatticeECP3 デバイスごとの SERDES/PCS クワッド数

パッケージ	ECP3-17	ECP3-35	ECP3-70	ECP3-95	ECP3-150
256 ボール ftBGA	1	1	—	—	—
328 ボール csBGA	2 ch *	—	—	—	—
484 ボール ftBGA	1	1	1	1	—
672 ボール ftBGA	—	1	2	2	2
1156 ボール ftBGA	—	—	3	3	4

1. チャンネル 0 と 3 が使用できます

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、あるいは内部 FPGA ロジックから供給することができます。

各クワッドは、公称周波数を持つ選択プロトコルによってプログラム可能で、チャンネルごとにフルレート及びハーフレート・オプションを使用できます。例えば、2.5Gbps の PCI Express x1 とギガビット・イーサネット・チャンネルを同一のクワッドで使用し、ギガビット・イーサネット・チャンネルでハーフレート・オプションを使用できます。あるクワッドで PCI Express x1 チャンネルと PCI Express 以外のチャンネルを共有する場合は、そのクワッドの基準クロックがそのクワッド内の全てのプロトコルと互換性を持つ必要があります。例えば、PCI Express スペクトラム分散基準クロックは、多くのギガビット・イーサネット用途と互換性がありません。

各クワッドにはそれぞれの基準クロックがあるため、クワッドが異なれば同一チップでも違う規格に対応することができます。この機能により、LatticeECP3 ファミリは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となっています。

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、ワードアライメントとクロックトレランス補償は、ユーザ定義の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

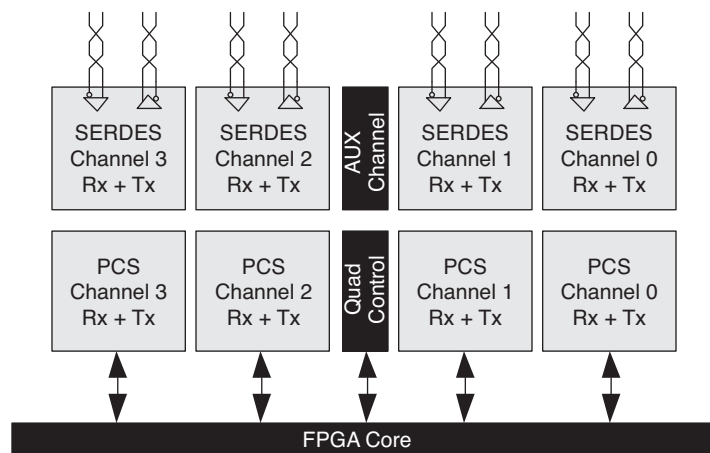
デバイスの各クワッドは最大 4 チャンネルの全二重データをサポートします。アプリケーションによっては、単一クワッドで 1 ～ 4 チャンネルのいずれも利用することができます。どのクワッドであれ、ユーザがチャンネルごとに独立して設定できる多くのオプションがあります。

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。

チャンネル単位の PCS/FPGA インターフェイス・ポート

使用されるモードにかかわらず、全ての 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 PCS モードと FPGA ロジック間の I/O ポートには、それぞれのクワッ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。図 8-2 に SERDES/PCS クワッドの概略を示します。

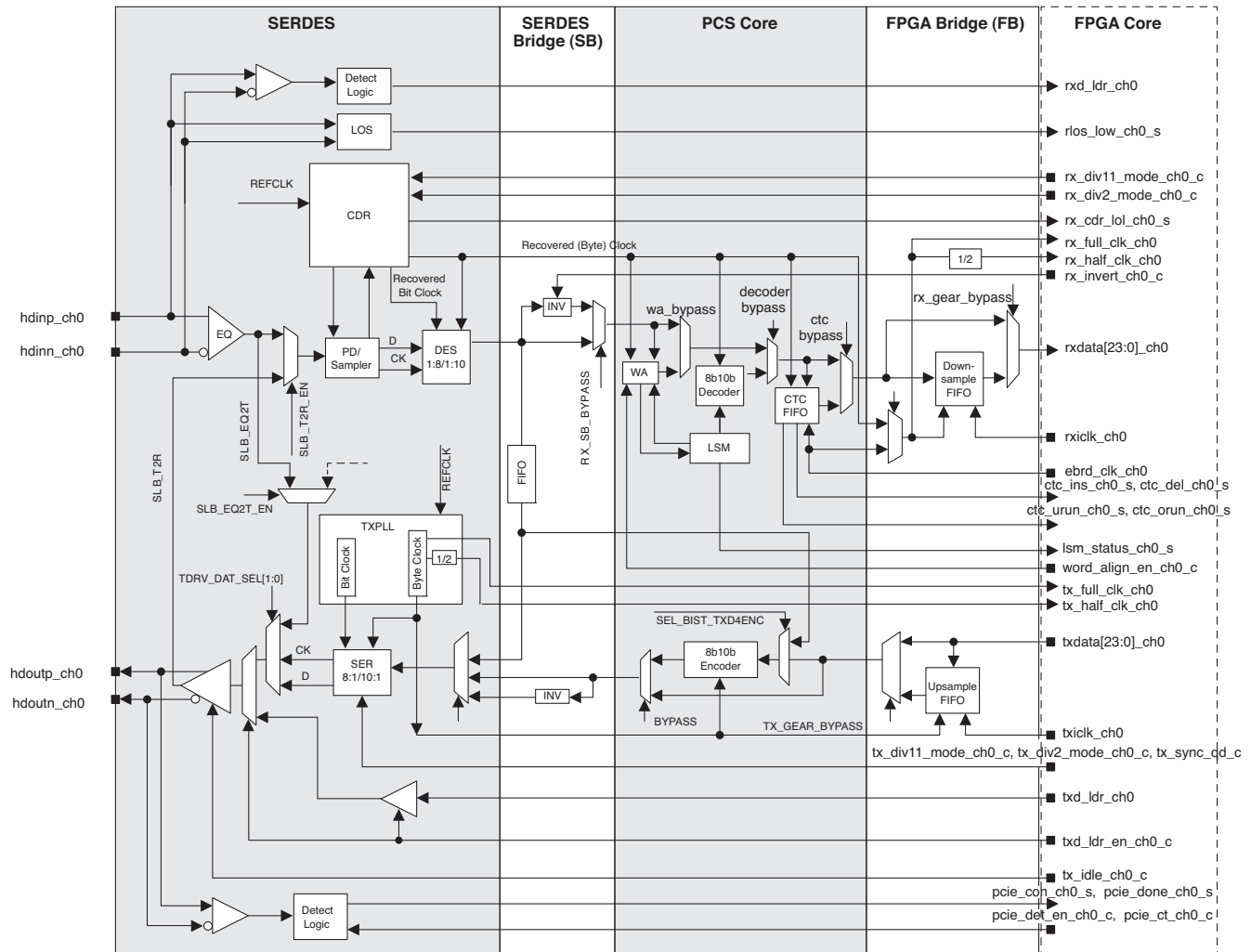
図 8-2. SERDES/PCS クワッドのブロック図



チャネルの詳細ブロック図

図 8-3 は LatticeECP3 SERDES/PCS 単一チャネルの主要機能の詳細ブロック図を表しています。この図には FPGA でユーザロジックから見える全ての主要ブロック、及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 内の主要なサブブロックである SERDES ブリッジ、PCS コア、及び FPGA ブリッジも示しています。

図 8-3. LatticeECP3 SERDES/PCS チャンネルの詳細ブロック図



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及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 ロジックの両方か、あるいは 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チャネルごとにあります。

送信データバス

送信データバスの信号は 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 データバス用に 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 ビット幅データバス) で、最大速度は 156.25MHz (XAUI 4x チャネルモード用) です。SERDES と PCS は、160MHz (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 3.2Gbps データレートまでサポートします。

受信データバス

受信バスの信号は、PCS ブロック内の FPGA ブリッジから FPGA までです。データバスは、8 ビット幅の内部 PCS データバス用に 2:1 ギャリングできます。FPGA インターフェイスのデータバス幅は 16 ビットです。ソフトウェア・レジスタビットで 2:1 ギャリングを禁止することが可能で、その場合バス幅は半分（8 ビット幅）にされます。データが 2:1 ギャリングされる時、下位ビット（rxdata[9:0]）は最初に受信されたワードに対応し、上位ビット（rxdata[19:10]）は 2 番目に受信されたワードに対応します。データが 2:1 ギャリングなしの場合、下位ビット（rxdata[9:0]）が有効なビットであり、上位ビットは使用されるべきではありません。表 8-3 に各プロトコルモードにおけるデータバスの用途を示します。

表 8-3. モード別のデータバスの用途

データバスの PCS セル名 ⁴	G8B10B	CPRI	OBSAI	PCI Express	SRIO	Gigabit Ethernet	XAUI	8-Bit SERDES	10-Bit SERDES	SDI	
FF_TX_D_0.0	txdata_ch0[0]										
FF_TX_D_0.1	txdata_ch0[1]										
FF_TX_D_0.2	txdata_ch0[2]										
FF_TX_D_0.3	txdata_ch0[3]										
FF_TX_D_0.4	txdata_ch0[4]										
FF_TX_D_0.5	txdata_ch0[5]										
FF_TX_D_0.6	txdata_ch0[6]										
FF_TX_D_0.7	txdata_ch0[7]										
FF_TX_D_0.8	tx_k_ch0[0]						txc_ch0[0]	GND	txdata_ch0[8]		
FF_TX_D_0.9	tx_force_disp_ch0[0] ¹				GND				txdata_ch0[9]		
FF_TX_D_0.10	tx_disp_sel_ch0[0] ¹				GND	xmit_ch0[0] ²	GND				
FF_TX_D_0.11	GND			pci_ei_en_ch0[0]	GND	tx_disp_correct_ch0[0]	GND				
FF_TX_D_0.12	txdata_ch0[8]								txdata_ch0[10]		
FF_TX_D_0.13	txdata_ch0[9]								txdata_ch0[11]		
FF_TX_D_0.14	txdata_ch0[10]								txdata_ch0[12]		
FF_TX_D_0.15	txdata_ch0[11]								txdata_ch0[13]		
FF_TX_D_0.16	txdata_ch0[12]								txdata_ch0[14]		
FF_TX_D_0.17	txdata_ch0[13]								txdata_ch0[15]		
FF_TX_D_0.18	txdata_ch0[14]								txdata_ch0[16]		
FF_TX_D_0.19	txdata_ch0[15]								txdata_ch0[17]		
FF_TX_D_0.20	tx_k_ch0[1]						txc_ch0[1]	GND	txdata_ch0[18]		
FF_TX_D_0.21	tx_force_disp_ch0[1] ¹				GND				txdata_ch0[19]		
FF_TX_D_0.22	tx_disp_sel_ch0[1] ¹				GND	xmit_ch0[1] ²	GND				
FF_TX_D_0.23	GND			pci_ei_en_ch0[1]	GND	tx_disp_correct_ch0[1]	GND				
FF_RX_D_0.0	rxdata_ch0[0]										
FF_RX_D_0.1	rxdata_ch0[1]										
FF_RX_D_0.2	rxdata_ch0[2]										
FF_RX_D_0.3	rxdata_ch0[3]										
FF_RX_D_0.4	rxdata_ch0[4]										
FF_RX_D_0.5	rxdata_ch0[5]										
FF_RX_D_0.6	rxdata_ch0[6]										
FF_RX_D_0.7	rxdata_ch0[7]										
FF_RX_D_0.8	rx_k_ch0[0]						rx_c_ch0[0]	NC	rxdata_ch0[8]		
FF_RX_D_0.9	rx_disp_err_ch0[0]			rxstatus0_ch0[0]	rx_disp_err_ch0[0]				NC	rxdata_ch0[9]	
FF_RX_D_0.10	rx_cv_err_ch0[0] ³			rxstatus0_ch0[1]	rx_cv_err_ch0[0] ³				NC		
FF_RX_D_0.11	NC			rxstatus0_ch0[2]	NC						
FF_RX_D_0.12	rxdata_ch0[8]								rxdata_ch0[10]		
FF_RX_D_0.13	rxdata_ch0[9]								rxdata_ch0[11]		
FF_RX_D_0.14	rxdata_ch0[10]								rxdata_ch0[12]		
FF_RX_D_0.15	rxdata_ch0[11]								rxdata_ch0[13]		
FF_RX_D_0.16	rxdata_ch0[12]								rxdata_ch0[14]		

表 8-3. モード別のデータバスの用途 (Continued)

データベースの PCS セル名 ⁴	G8B10B	CPRI	OBSAI	PCI Express	SRIO	Gigabit Ethernet	XAUI	8-Bit SERDES	10-Bit SERDES	SDI
FF_RX_D_0_17	rxdata_ch0[13]								rxdata_ch0[15]	
FF_RX_D_0_18	rxdata_ch0[14]								rxdata_ch0[16]	
FF_RX_D_0_19	rxdata_ch0[15]								rxdata_ch0[17]	
FF_RX_D_0_20	rx_k_ch0[1]						rx_c_ch0[1]	NC	rxdata_ch0[18]	
FF_RX_D_0_21	rx_disp_err_ch0[1]		rxstatus1_ch0[0]		rx_disp_err_ch0[1]			NC	rxdata_ch0[19]	
FF_RX_D_0_22	rx_cv_err_ch0[1] ³		rxstatus1_ch0[1]		rx_cv_err_ch0[1] ³			NC		
FF_RX_D_0_23	NC		rxstatus1_ch0[2]		NC					

1. force_disp 信号はデータワード [7:0] に対して tx_disp_sel 信号で選択されたカラムのビットにデイ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+' (正のデイスパリティ) カラムから選択されます。tx_disp_sel が 0 の場合、10 ビットコードは '現在の RD-' (負のデイスパリティ) カラムから選択されます。
2. ラティスのギガビット・イーサネット PCS IP コアは、信号 xmit を生成する自動ネゴシエーション・ステートマシンを内蔵します。これはハードロジック内で、ギガビット・イーサネット・アイドルステートマシンと相互にインターフェイスする為に用いられます。
3. コード違反があるとき、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)。
4. FF_TX_D_0_0 : FPGA ファブリック送信データバスのチャネル 0 ビット 0。

モードごとの制御 / ステータス信号記述

表 8-4 にモードごとの制御 / ステータス信号を説明します。

表 8-4. 制御信号とその機能

信号名	記 述
送信制御信号	
tx_k_ch[3:0]	チャンネル別。アクティブ High 制御キャラクタ・インジケータ
tx_force_disp_ch[3:0]	チャンネル別。disp_sel_ch(0-3) FPGA インターフェイス入力からのデイスパリティ値を PCS が受け付けるようにするアクティブ High 信号
tx_disp_sel_ch[3:0]	チャンネル別。FPGA ロジックから供給されるデイスパリティ値。force_disp_ch(0-3) がハイのときに有効
tx_correct_disp_ch[3:0]	チャンネル別。ネイティブ・デイスパリティ状態で開始するよう 8b10b エンコーダを調整することで、アサートされたときにデイスパリティ識別子を修正
受信ステータス信号	
rx_k_ch[3:0]	チャンネル別。アクティブ High 制御キャラクタ・インジケータ
rx_disp_err_ch[3:0]	チャンネル別。関連データでデイスパリティ・エラーが検出されたことを示す、PCS がドライブするアクティブ High 信号
rx_cv_err_ch[3:0]	チャンネル別。関連データでエラーが検出されたことを示すコード違反信号

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。

{signal}_c は、FPGA コアから FPGA ブリッジへの制御信号です。全ての制御信号は、SERDES/PCS 内部で同期化されてから使用されます。

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。Diamond 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。

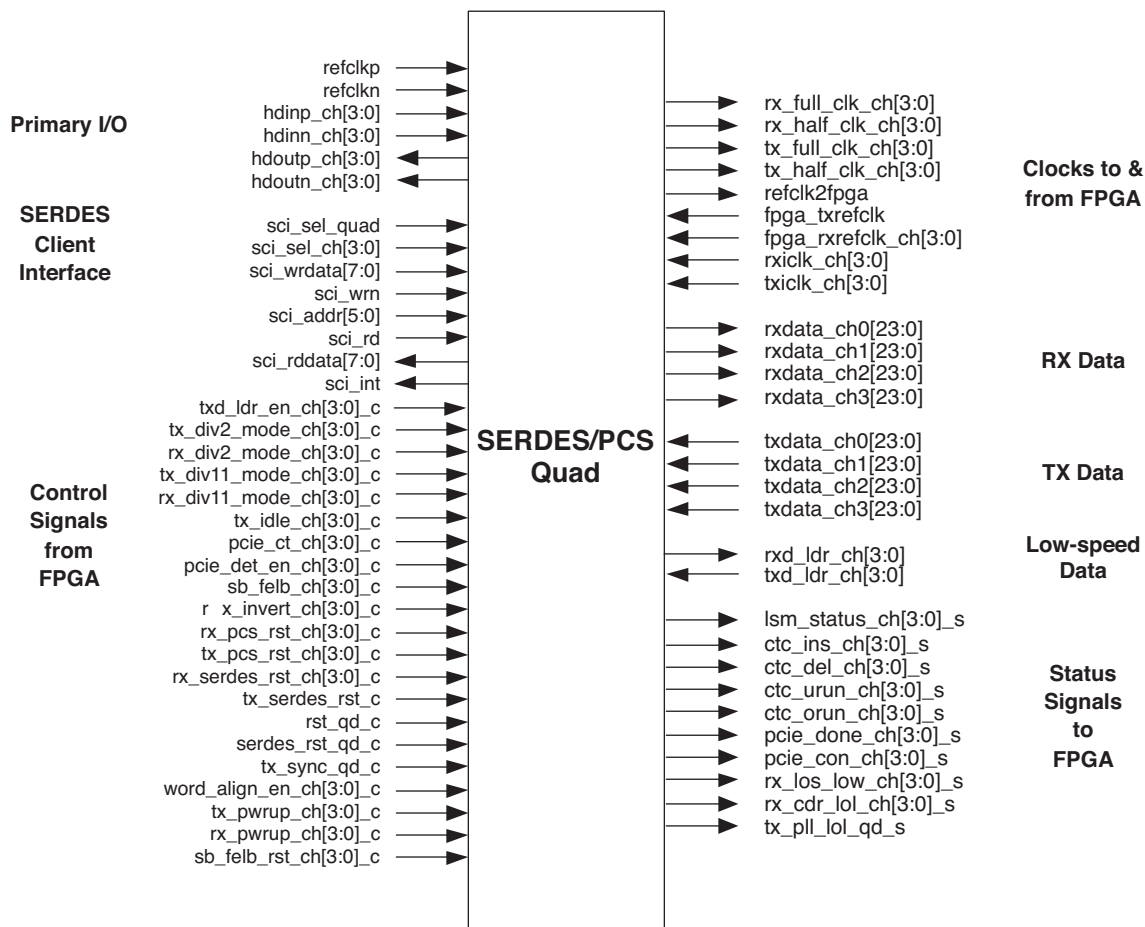
{signal}_s は、FPGA ブリッジから FPGA コアへのステータス信号です。全てのステータス信号は、SERDES/PCS から同期出力されます。これらをクロックドメインと同期してから、FPGA デザインで使用する必要があります。

制御とステータスに関する詳細な情報については “モードごとの制御 / ステータス信号” セクションを参照してください。

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化及びパラレル化機能を実行し、マクロ内の TxPLL 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を図 8-4 に示します。

図 8-4. SERDES/PCS ブロックの信号インターフェイス



I/O 記述

表 8-5 に PCS クワッドに対する全てのデフォルト及びオプションの I/O をリストアップします。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
プライマリ I/O, SERDES クワッド			
hdinp_ch0	I	チャネル	高速 CML 入力、非反転、チャネル 0
hdinn_ch0	I	チャネル	高速 CML 入力、反転、チャネル 0
hdinp_ch1	I	チャネル	高速 CML 入力、非反転、チャネル 1
hdinn_ch1	I	チャネル	高速 CML 入力、反転、チャネル 1
hdinp_ch2	I	チャネル	高速 CML 入力、非反転、チャネル 2
hdinn_ch2	I	チャネル	高速 CML 入力、反転、チャネル 2
hdinp_ch3	I	チャネル	高速 CML 入力、非反転、チャネル 3
hdinn_ch3	I	チャネル	高速 CML 入力、反転、チャネル 3
hdoutp_ch0	O	チャネル	高速 CML 出力、非反転、チャネル 0
hdoutn_ch0	O	チャネル	高速 CML 出力、反転、チャネル 0
hdoutp_ch1	O	チャネル	高速 CML 出力、非反転、チャネル 1
hdoutn_ch1	O	チャネル	高速 CML 出力、反転、チャネル 1
hdoutp_ch2	O	チャネル	高速 CML 出力、非反転、チャネル 2
hdoutn_ch2	O	チャネル	高速 CML 出力、反転、チャネル 2
hdoutp_ch3	O	チャネル	高速 CML 出力、非反転、チャネル 3
hdoutn_ch3	O	チャネル	高速 CML 出力、反転、チャネル 3
refclkp	I	クワッド	基準クロック入力、非反転、専用 CML 入力
refclkn	I	クワッド	基準クロック入力、反転、専用 CML 入力
送受信データバス（詳細なバス使用法については表 8-3 及び 8-4 を参照）			
rxdata_ch0[23:0]	O	チャネル	チャネル 0 受信データバスのデータ信号
rxdata_ch1[23:0]	O	チャネル	チャネル 1 受信データバスのデータ信号
rxdata_ch2[23:0]	O	チャネル	チャネル 2 受信データバスのデータ信号
rxdata_ch3[23:0]	O	チャネル	チャネル 3 受信データバスのデータ信号
txdata_ch0[23:0]	I	チャネル	チャネル 0 送信データバスのデータ信号
txdata_ch1[23:0]	I	チャネル	チャネル 1 送信データバスのデータ信号
txdata_ch2[23:0]	I	チャネル	チャネル 2 送信データバスのデータ信号
txdata_ch3[23:0]	I	チャネル	チャネル 3 送信データバスのデータ信号
制御信号			
tx_idle_ch[3:0]_c	I	チャネル	SERDES トランスミッタによる電氣的アイドルの送信制御 1 = SERDES トランスミッタが電氣的アイドルを強制的に出力 0 = 通常動作
pcie_det_en_ch[3:0]_c	I	チャネル	FPGA ロジック（ユーザロジック）が SERDES ブロックに、PCI Express レシーバ検出動作の要求があることを通知 1 = PCI Express レシーバ検出をイネーブル、0 = 通常動作
pcie_ct_ch[3:0]_c	I	チャネル	1 = トランスミッタにファーエンド・レシーバ検出を要求 0 = 通常データ動作
rx_invert_ch[3:0]_c	I	チャネル	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
word_align_en_ch[3:0]_c	I	チャネル	カンマアライナを制御 1 = カンマアライナをイネーブル 0 = カンマアライナを現在の位置にロック

表 8-5. SERDES_PCS I/O 記述 (Continued)

信号名	I/O	タイプ	記 述
sb_felb_ch[3:0]_c	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 = 通常のデータ動作
sb_felb_rst_ch[3:0]_c	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
tx_sync_qd_c	I	クワッド	シリアルライザのリセット 遷移 = リセット、レベル = 通常動作
rx_div2_mode_ch[3:0]_c	I	チャンネル	レシーバのレートモード選択 (フル / ハーフレート) 1 = ハーフレート、0 = フルレート
tx_div2_mode_ch[3:0]_c	I	チャンネル	トランスミッタのレートモード選択 (フル / ハーフレート) 1 = ハーフレート、0 = フルレート
rx_div11_mode_ch[3:0]_c	I	チャンネル	レシーバのレートモード選択 (DIV11 / フルレート) 1 = DIV11 レート、0 = フルレート
tx_div11_mode_ch[3:0]_c	I	チャンネル	トランスミッタのレートモード選択 (DIV11 / フルレート) 1 = DIV11 レート、0 = フルレート
txd_ldr_en_ch[3:0]_c	I	チャンネル	低データレート TX シリアルパスのイネーブル 1 = イネーブル、0 = ディセーブル
リセット信号			
rx_pcs_rst_ch[3:0]_c	I	チャンネル	アクティブ High、非同期入力。PCS 内の個々の受信チャンネルロジックのみをリセット
tx_pcs_rst_ch[3:0]_c	I	チャンネル	アクティブ High、非同期入力。PCS 内の個々の送信チャンネルロジックのみをリセット
rx_serdes_rst_ch[3:0]_c	I	チャンネル	アクティブ High、SERDES 受信チャンネル内の一部ロジックのみをリセット
tx_serdes_rst_c	I	クワッド	アクティブ High、SERDES 全送信チャンネル内の一部ロジックのみをリセット
rst_qd_c	I	クワッド	アクティブ High、非同期入力。補助チャンネル及び PCS を含む SERDES 全チャンネルをリセット
serdes_rst_qd_c	I	クワッド	アクティブ High、SERDES クワッドへの非同期入力。PCS ロジック以外のクワッドチャンネルを含む SERDES 全チャンネルをリセット
tx_pwrup_ch[3:0]_c	I	チャンネル	アクティブ High 送信チャンネル・パワーアップ。0 = 送信チャンネルパワーダウン
rx_pwrup_ch[3:0]_c	I	チャンネル	アクティブ High 受信チャンネル・パワーアップ。0 = 受信チャンネルパワーダウン
ステータス信号			
pcie_done_ch[0:3]_s	O	チャンネル	1 = ファーエンド・レシーバ検出完了 0 = ファーエンド・レシーバ検出未完了
pcie_con_ch[3:0]_s	O	チャンネル	ファーエンド・レシーバ検出の結果 1 = ファーエンド・レシーバ検出 0 = ファーエンド・レシーバ不検出
rx_los_low_ch[3:0]_s	O	チャンネル	各チャンネル用 LOS (LO THRESHOLD RANGE) 検出
lsm_status_ch[3:0]_s	O	チャンネル	1 = レーンはカンマに同期 0 = レーンはカンマを不検出
ctc_urrun_ch[3:0]_s	O	チャンネル	1 = 受信クロック補償 FIFO のアンダーランエラー 0 = FIFO エラーなし
ctc_orun_ch[3:0]_s	O	チャンネル	1 = 受信クロック補償 FIFO のオーバーランエラー 0 = FIFO エラーなし
rx_cdr_lol_ch[3:0]_s	O	チャンネル	1 = 受信 CDR LOL 0 = ロックを維持
tx_pll_lol_qd_s	O	クワッド	1 = 送信 PLL LOL 0 = ロックを維持
ctc_ins_ch[3:0]_s	O	チャンネル	1 = CTC がスキップキャラクタを挿入

表 8-5. SERDES_PCS I/O 記述 (Continued)

信号名	I/O	タイプ	記 述
ctc_del_ch[3:0]_s	O	チャンネル	1 = CTC がスキップキャラクタを削除
FPGA インターフェイス・クロック			
rx_full_clk_ch[3:0]	O	チャンネル	受信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。10 GbE などの 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送信システムクロックであり、8b10b モードで 8b10b デコーダをディセーブルして (10 ビットまたは 20 ビット・データバス) 実行する "raw" モードの必要あり。
rx_half_clk_ch[3:0]	O	チャンネル	受信チャンネル再生ハーフクロック。2:1 ギアリングモードでは 2 分周された出力
tx_full_clk_ch[3:0]	O	チャンネル	送信 PLL フルレート・クロック。プライマリクロック配線を直接ドライブできるのは tx_full_clk_ch0 のみ。tx_full_clk_ch[3:0] の全信号で、セカンダリクロックのルーティングを直接ドライブ可能 (適用する制約 USE SECONDARY が必要) ^{注3}
tx_half_clk_ch[3:0]	O	チャンネル	送信 PLL ハーフクロック。プライマリクロックのルーティングを直接ドライブできるのは tx_half_clk_ch0 のみ。tx_half_clk_ch[3:0] の全信号で、セカンダリクロックのルーティングを直接ドライブ可能 (適用する制約 USE SECONDARY が必要) ^{注3}
refclk2fpga	O	クワッド	FPGA コアへの基準クロック。選択された場合、このクロックは基準クロックがある限り、クワッドがパワーダウン・モードになっても常に有効
fpga_rxrefclk_ch[3:0]	I	クワッド	FPGA ロジックからの Rx 基準クロック。CDR PLL 用
fpga_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
ebrd_clk_ch[3:0] ²	I	チャンネル	FPGA からの受信チャンネルクロック入力。CTC FIFO リード用
rxiclk_ch[3:0]	I	チャンネル	FPGA からの受信チャンネルクロック入力。基準クロックや受信基準クロックに同期した FIFO と Rx FPGA 部のインターフェイス用クロック
txiclk_ch[3:0]	I	チャンネル	FPGA からの送信チャンネルクロック入力。FPGA からのチャンネルごとの送信クロック入力。基準クロックに同期した FIFO と Tx FPGA 部のインターフェイス用クロック。CTC が用いられる場合は、基準クロックに同期した FIFO と Rx FPGA 部のインターフェイス用クロックとしても用いられる
低速送受信データ及び SERDES クライアント・インターフェイス (SCI) 信号			
rx_d_lch[3:0]	O	チャンネル	FPGA コアへのシングルエンド・シリアル低速データ出力 (RX)
tx_d_lch[3:0]	I	チャンネル	FPGA コアからのシングルエンド・シリアル低速データ入力 (TX)
sci_wrdata[7:0]	I	—	ライトデータ入力
sci_wrn	I	—	ライト入力ストロブ
sci_sel_quad	I	—	クワッドレジスタを選択
sci_sel_ch[3:0]	I	—	チャンネルレジスタを選択
sci_addr[5:0]	I	—	アドレスバス入力
sci_rd	I	—	リードデータ選択
sci_rddata[7:0]	O	—	リードデータ出力
sci_int	O	—	割り込み出力

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. このクロックは、ラッパーモジュールのポートリストで提供されません。CTC モードに応じて、ソフトウェアが自動的にクロックを割り当てます。詳細については、本ドキュメントの "FPGA インターフェイス・クロック" のセクションを参照してください。
3. セカンダリ・クロックネットワークを用いる場合には一般配線を経由します。PAR でウォーニングが生成されるかもしれませんが、殆どのアプリケーションで無視できる程度に遅延値は少ないでしょう。タイミング制約を与えて、タイミング違反がないことをレポートで確認して下さい。

SERDES/PCS の機能記述

LatticeECP3 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ネルを 4 系統サポートします。単一チャ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チャネル 4 つまでに対応できます。

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に対応しています。

以下のそれぞれについては図 8-3 を参照してください。

- ・ SERDES
 - イコライザ
 - CDR (クロックデータ再生)
 - デシリアライザ
 - プリエンファシス
 - シリアライザ
 - 2 つのシリアル・ループバックモード。TX から RX、または RX から TX
- ・ SERDES ブリッジ (SB)
 - インバータ：受信データを反転。PCI Express で必要
 - 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 (WA)
 - 8b10b デコーダ
 - 8b10b エンコーダ
 - リンクステートマシン (LSM)
 - クロックトレランス補償 (CTC)
- ・ FPGA ブリッジ (FB)
 - ダウンサンプル FIFO
 - アップサンプル FIFO

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著しいシンボル間干渉が発生します。したが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。Mid_Low、Mid_Med、Mid_High、Long_Low、Long_Med、Long_High という 6 つの極配置が用意されています。

プリエンファシス

プリエンファシスとは、一部周波数の振幅を他の周波数の振幅に対して増大させる信号処理のことです。目的は、減衰量の差などの現象による悪影響を最小限に抑え、全体としての S/N 比を改善することです。最大 80% のプリエンファシスを選択できます。

基準クロック

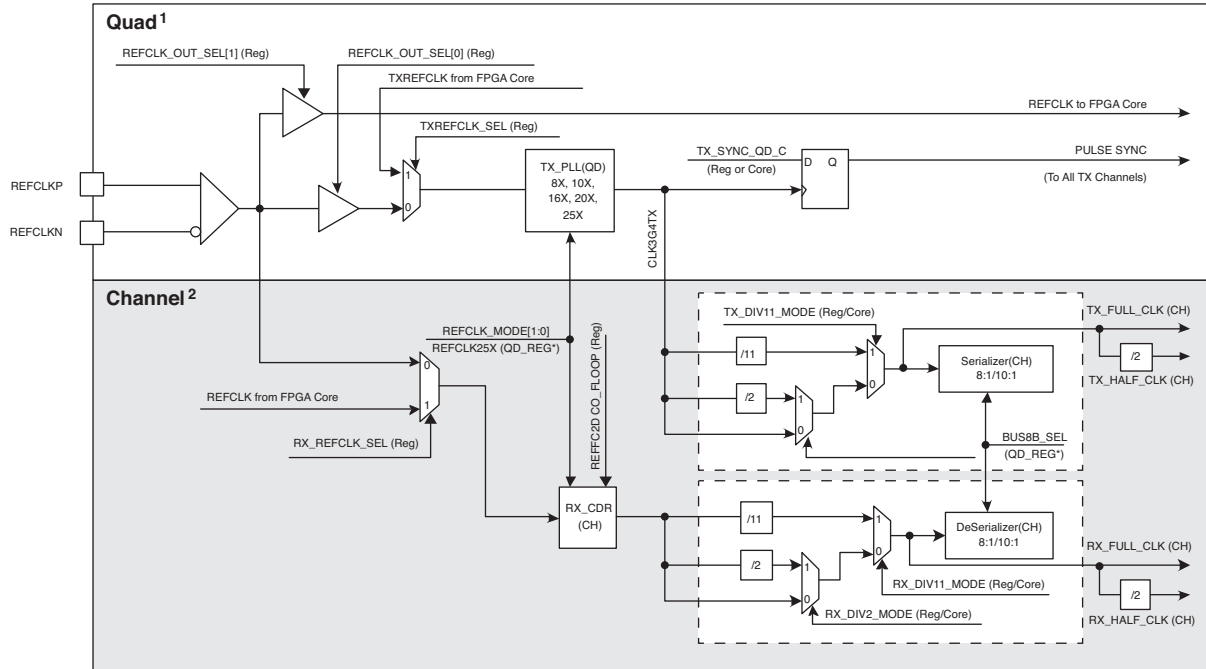
SERDES クワッドは 4 チャネルの Rx と Tx 回路、及び Tx PLL を含む補助チャネルを含んでいます。Tx PLL への基準クロックは、プライマリ差動基準クロックピン、隣接クワッドの基準クロック、または FPGA コアから供給することができます。また、SERDES ブロック内の PLL は出力クロックを提供し、FPGA ファブリックをドライブするシステムクロックとして使用できます。

Rx への基準クロックは、Tx PLL への基準クロックか FPGA コアから供給することができます。FPGA コアから Tx PLL へ、及び Rx への基準クロックは様々なソースから得ることができます。

SERDES クロック・アーキテクチャ

図 8-5 に SERDES のクロック・アーキテクチャ全体を示します。この図はクワッドとチャネルの 2 つの部分に分かれています。簡略化のため、チャネルごとに共通のセクションのみを一系統分示してあります。またブロック用に異なる様々な制御ビットも示してあります。これらはクワッドベースの制御レジスタビット、またはチャネルベースの制御レジスタビットになっています。場合によっては、チャネル制御ポートベースとなっていることもあり、中にはレジスタと制御ポートの両方の組み合わせになっているものがあります。両方のモードを使用すると、特定の機能プロパティの動的制御が可能になります。

図 8-5. SERDES クロック・アーキテクチャ



1. All control bits are quad based.
2. All control bits are channel based, except as indicated (*).
3. These clocks are user-transparent.

クロック・アーキテクチャの主要コンポーネントには、次のものがあります。

- ・ RX ごと及び TX ごとの分周器 (DIV) モード : DIV2、DIV11
- ・ マルチクワッド REFCLK 接続
- ・ FPGA からの tx_sync_qd_c 信号を使用したマルチチャネル送信同期
- ・ OOB 低速データ用途サポート

レートモード

各チャネルの TX は、次のレートで動作するよう個別にプログラムできます。

- ・ FULL_RATE
- ・ HALF_RATE (DIV2)
- ・ DIV11

Rx もチャネルごとに個別の基準クロックを使用できるため、トランスミッタとレシーバが完全に異なるレートで動作できます。

重要な点は、PLL VCO に変更はないことであり、従ってそのプロトコルに対する最速レートに対応できます。そのプロトコルに必要な分周後の全レートは、分周器の MUX 選択をプログラムすることで対応できま

す。その結果、PLL をプログラムし直す必要がないため、非常に高速なデータレート切り替えが可能になります。この機能は多くの用途で有用です。

注：LatticeECP3 PCS は動作中に refclk の周波数を変更するだけでは、SERDES が新しいレートで動作するようにはなりません。（日本語注；本注記は英語オリジナルの表記と異なりますが、いずれ更新予定です。）

通常 TX PLL と 4 つの CDR PLL は、基準クロック周波数の倍数となる同一周波数で動作します。表 8-6 に動作可能なクロックレートの各種モードを示します。ここに示したビットクロックは、基準クロック周波数の倍数になっています。

表 8-6. TXPLL 及び RX CDRPLL が対応可能なモード

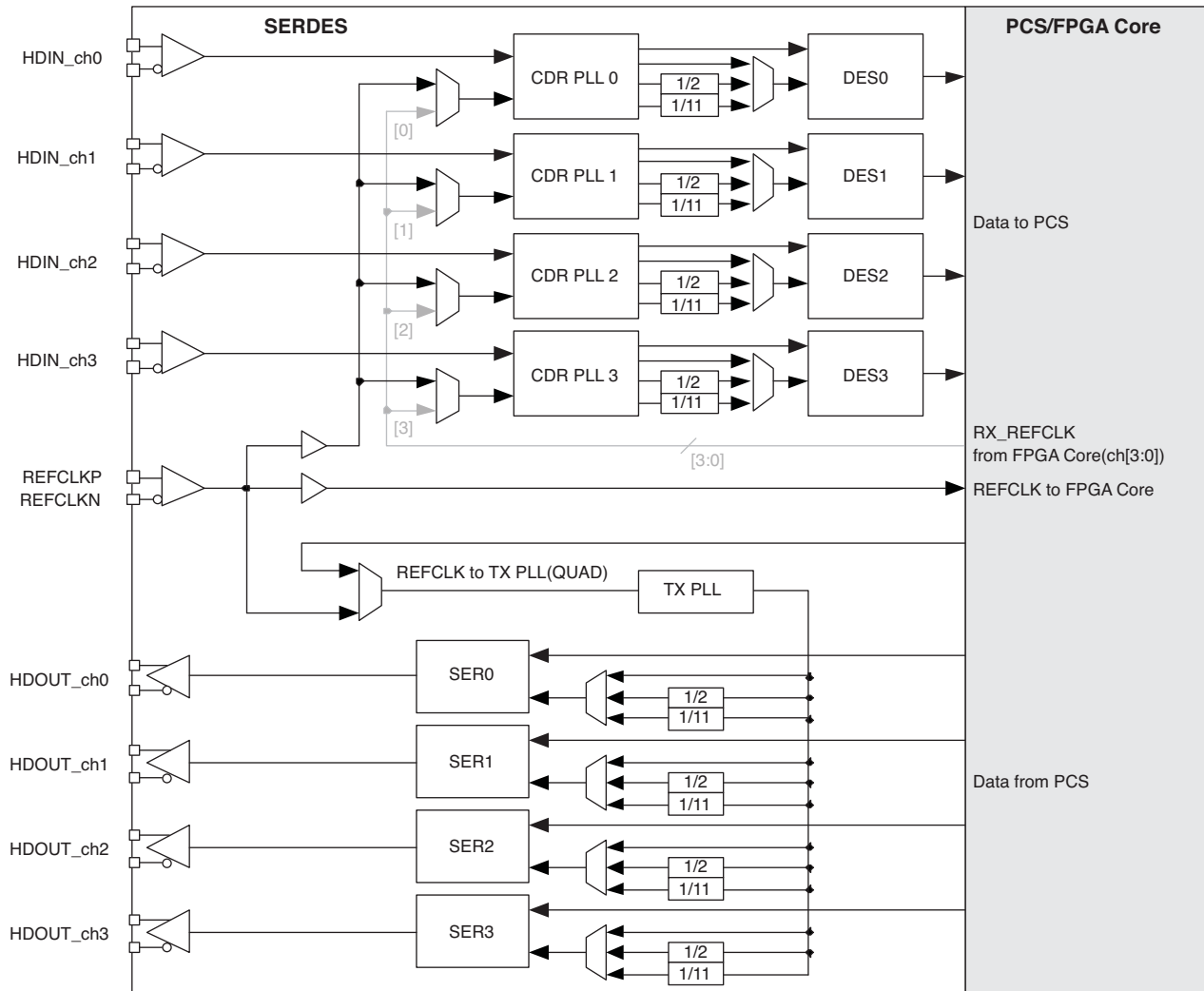
基準クロックのモード	refclkPmode (クワッド)	Bus_width	ビットクロック (フルレート)	ビットクロック (div2, div11)
20x	0	10	Refclk x 20	Refclk x 10
16x	0	8	Refclk x 16	Refclk x 8
10x	1	10	Refclk x 10	Refclk x 5
8x	1	8	Refclk x 8	Refclk x 4
25x	—	8	Refclk x 25	Refclk x 12.5
25x	—	10	Refclk x 25	Refclk x 12.5
20x	0	10	Refclk x 20	Refclk x 20/11 ¹

1. DIV11 モード

FPGA コアからの基準クロック

図 8-5 に示すとおり、TX 基準クロックは FPGA コアからも供給できます。この場合、クロックを SERDES まです配線する FPGA のリソースに起因する、過剰なジッタが送信データに加わり、場合によっては TX ジッタ規格を満たさなくなる可能性があります。FPGA で生成した SERDES TX 基準クロックを使用する場合は注意が必要です。

図 8-6. 基準クロックのブロック図

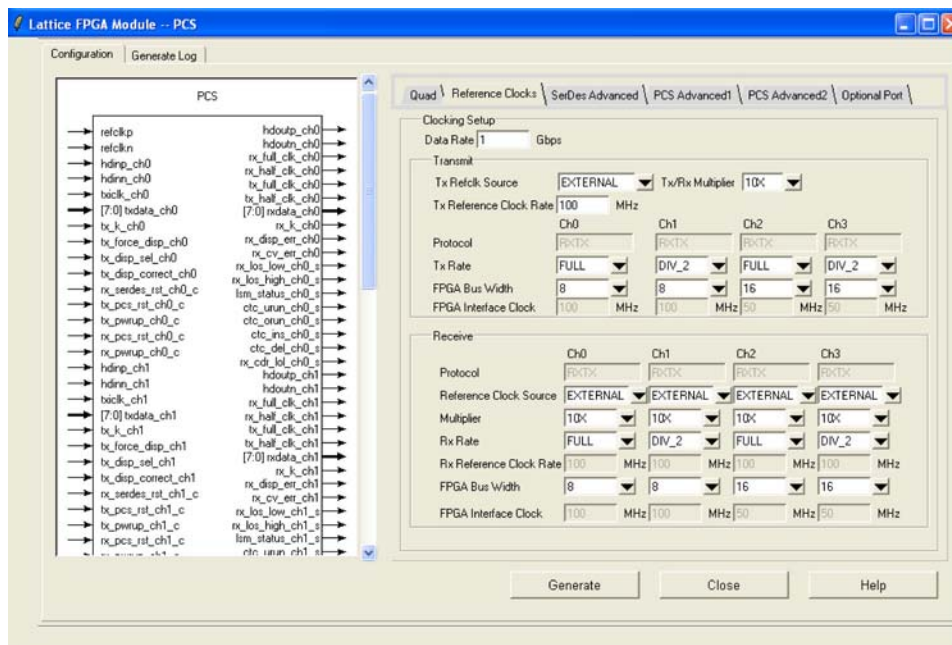


フルデータレート、DIV2、DIV11 データレート

各 TX シリアルライザと RX デシリアルライザは、プロトコルに応じてフルデータレートと DIV2 レートまたは DIV11 レートに分けられるため、方向ごと及びチャンネルごとに異なるデータレートが可能です。詳細については図 8-6 を参照してください。

図 8-7 に示すとおり、4 つ全てのチャンネルで異なる構成が可能です。

図 8-7. IPexpress GUI でのフルデータレートとハーフデータレートの例



この例の実際のデータレート及びFPGA インターフェイスのクロックレートを、表 8-7 に示します。IPexpress GUI については、本ドキュメントで詳しく後述します。

表 8-7. クロックレートの例

チャネル	データレート	基準クロックの乗数	データレート・モード	基準クロックレートの計算値	FPGA インターフェイスのデータバス幅	FPGA インターフェイスのクロックレート	tx_full_clk	tx_half_clk
Channel 0	1 Gbps	10 x	FULL	100 MHz	8 (10) ³	100 MHz	100 MHz	50 MHz
Channel 1	500 Mbps	10 x	DIV2	100 MHz	8 (10)	50 MHz	50 MHz	25 MHz
Channel 2	1 Gbps	10 x	FULL	100 MHz	16 (20)	50 MHz	100 MHz	50 MHz
Channel 3	500 Mbps	10 x	DIV2 ²	100 MHz	16 (20)	25 MHz	50 MHz	25 MHz

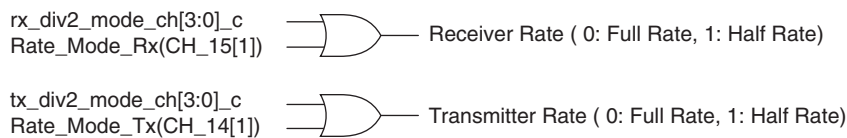
1. グレーで示したセルのクロックは、各モードで FPGA インターフェイス・クロックとして使用されます。
2. DIV2 モードでは、tx_full_clk がハーフレートに調整されます。tx_half_clk は 16 ビットバス・インターフェイスにのみ使用されます。
3. 10 ビット SERDES 専用モードまたは SDI モード。

フルレートとハーフレート (DIV2) の動的切り替え

このセクションでは、フルレートとハーフレート（DIV）を動的に切り替える方法について説明します。

2つのレートモード制御信号は、図 8-8 に示すように OR 接続されます。

図 8-8. レートモード制御信号



tx_div2_mode_chx_c は FPGA ファブリックから TX パスへの入力制御信号です。

rx_div2_mode_chx_c は FPGA ファブリックから RX パスへの入力制御信号です。

Rate_Mode_Tx(CH_14[1]) は TX パス用の制御レジスタビットです。

Rate_Mode_Rx(CH_15[1]) は RX パス用の制御レジスタビットです。

受信レーンには、切り替え後に pcs_rst を与える必要があります。

送信レーンは、リセットなしで新しいレートが有効になります。

基準クロックのソース

refclkp, refclkn

CML 入力専用。送受信に別々のソースが使用されていない限り、第一の選択肢です。クロック信号は CML、LVDS、LVPECL のいずれかです。インターフェイス回路の例については、TN1114 “Electrical Recommendations for Lattice SERDES (ラティス SERDES の電氣的推奨事項)” を参照してください。

fpga_txrefclk, fpga_rxrefclk

FPGA ロジックからの基準クロック。FPGA へのクロック入力ピンとしては、プライマリクロック・パッド (PCLK) を使用する必要があります。クロック信号は CML、LVDS、LVPECL、シングルエンドのいずれかです。

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を専用 PLL 入力パッドに割り当てる必要があります。高速データレート時は、FPGA PLL 出力のジッタがシステム仕様を満たさないことがあります。ジッタが重要な用途では FPGA PLL を使用しないことを推奨します。

スペクトル拡散クロック (SSC) の対応

リンク両端のポートは、常に 600ppm 以内のレートで相互にデータを送信する必要があります。これは $\pm 300\text{ppm}$ のビットレートのクロックソースを許容するための仕様です。最小クロック周期に違反してはいけません。これに適した方法は、公称周波数を上回る変調が発生しないよう、拡散技術を調整することです。データレートは、変調レートの範囲が 30 ~ 33KHz を超えない状態で、公称データレート +0% ~ -0.5% の範囲の変調が許容されます。 $\pm 300\text{ppm}$ というトレランスの上限があるため、SSC でデータを変調する場合は、両方のポートに同じビットレートのクロックが必要です。

PCI Express では、基準クロックを拡散するソースとなるのはルートコンプレックスです。エンドポイントは同じクロックを使用して、拡散スペクトルを返します。そのため、独立した RXREFCLK が不要です。主要な用途としてはアドインカードがあります。アドインカードはコネクタからの REFCLK を使用する必要がありますが、PCI Express コネクタの REFCLK と同じ SSC で送受信する必要があります。

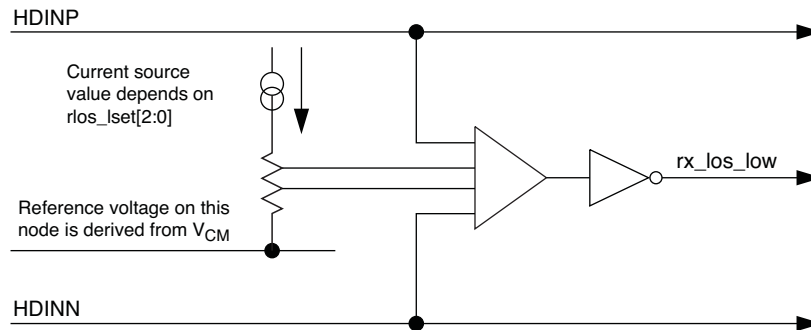
LatticeECP3 のアーキテクチャでは、同一クワッド内で PCI Express チャンネルと、ギガビット・イーサネット、Serial RapidIO、または SGMII チャンネルを混在できますが、送信基準クロックとして PCI Express の SSC を使用する場合は、ギガビット・イーサネット、Serial RapidIO、及び SGMII の送信ジッタ仕様に違反します。

LOS (Loss of Signal、信号の喪失)

各チャネルは、図 8-9 に示すようにプログラマブルな LOS 検出器が含まれています。

LOS 閾値はプログラマブル電流源の値に依存し、電流源の値は rlos_lset[2:0] 制御ビットを使用して選択されます。閾値検出結果は、rx_los_low ステータス信号によって FPGA に通知されます。

図 8-9. 検出器



Note: rx_los_low shows that a signal has been detected for data rates above 1 Gbps with a maximum CID (Consecutive Identical Digits) of 7 bits (i.e., a minimum input signal transition density as is sent by 8b10b).

rx_los_low is supported with a default setting of rlos_lset[2:0] = 2, except in PCI Express mode and SDI mode. In PCI Express mode, 2 and 3 are supported.

In SDI mode, it is recommended to use the carrier detect output signal (/CD) from the external SDI cable equalizer.

表 8-8. LOS 検出器の応答時間

記 述	Min.	Typ.	Max.	単位
信号の喪失を検出するまでの時間 (rx_los_low が 0 から 1)	—	8	10	ns
信号の存在を検出するまでの時間 (rx_los_low が 1 から 0)	—	8	10	ns

LOL (Loss of Lock、ロックはずれ)

送信 PLL と個別チャネルの CDR には、どちら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になります。CDR のロックがはずれると、そのチャネルの 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。CDR の LOL の詳細については、本ドキュメントの "SERDES/PCS のリセット" セクションを参照してください。

表 8-9. LOL 検出器の応答時間

記 述	Min.	Typ.	Max.	単位
ループのロックはずれを検出するまでの時間 (x_pll_lol、rx_cdr_lo が 0 から 1)	—	200	500	us
ループのロックを検出するまでの時間 (x_pll_lol、rx_cdr_lo が 1 から 0)	—	200	500	us

TX レーン間スキュー

制御信号 tx_sync_qd_c は、アクティブな全 TX チャネルをリセットし、bit0 位置からシリアル化を開始します。ほとんどのマルチチャネル・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。

TX シリアライザへのリセットは、tx_sync_qd_c 信号をトグルするか、PLL LOL の遷移によって生成されます。

SERDES PCS 構成のセットアップ

LatticeECP3 PCS は様々な用途で使用するよう構成（パラメータ / オプションの設定）ができます。セットアップは IPexpress モジュール生成ツールで選択され、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、これはデザインツール内のビットストリーム・ジェネレータによって用いられて、選択内容が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、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、自動構成ファイルを手動編集することによって、一部のオプションを変えることもできます。或いはデバイスの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 レジスタにライトすることによって、動的に PCS オプションを変えることもできます。SCI では、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 SERDES/PCS クワッドを制御します。SCI からアクセス可能な制御 / ステータスレジスタの一覧表を、本ドキュメントの付録 A に示してあります。

自動構成（Auto-Configuration）ファイル

IPexpress 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 / チャネルレジスタ設定の自動構成ファイル（<module_name>.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。自動構成ファイルがビットストリームに統合されると、全てのクワッド / チャネルレジスタが、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。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 インターフェイスを含める必要があります。

送信データ

PCS クワッド送信データパスは、チャネル単位の 8b10b エンコーダとシリアルライザで構成されます。

8b10b エンコーダ

本モジュールは 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パリティ規則を維持します。属性（Attribute、アトリビュート）CHx_8B10B（x はチャネル番号）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。

シリアルライザ

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パスは、チャネルごとのデシリアライザ、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、及びオプションの受信クロックトレランス補償（CTC）FIFO で構成されます。

デシリアライザ

データはチップに取り込まれて組み込み SERDES に送られ、シリアルからパラレルに変換されます。

ワードアライメント（バイト境界の検出）

このモジュールはカ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、カンマキャラクタが用いられています。カンマ記述は 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 Figure 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。

- ・ 組み込みリンクステートマシン（LSM）または FPGA 制御からのワードアライメント制御。8b10b パケットモードに加えて、8-bit SERDES Only、10-bit SERDES Only、SDI の各モードが対応しています。

- ・ 2つのプログラマブル・ワードアライメント・キャラクタ（通常正と負のディ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、“XX00000011”（正のランニング・ディスパリティ・カンマキャラクタでコードグループ K28.1、K28.5、及び K28.7 に適合する jhgfiedcba ビット）と “XX01111100”（負のランニング・ディスパリティ・カンマキャラクタでコードグループ K28.1、K28.5、及び K28.7 に適合する jhgfiedcba 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。ただし、任意の 10 ビットパターンを定義できます。
- ・ 最初のアライメント・キャラクタは COMMA_A 属性に割り当てられた 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 2 のアライメント・キャラクタは COMMA_B 属性に割り当てられた 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）。COMMA_M 属性に割り当てられた 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。ワードアライメント属性 CHx_RXWA が “ENABLED”、CHx_ILSM（内部リンクステートマシン）が “ENABLED” にそれぞれ設定されると、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。プロトコルベースのリンクステートマシン動作の詳細については、後述するプロトコルごとのリンクステートマシンのセクションを参照してください。

8b10b デコーダ

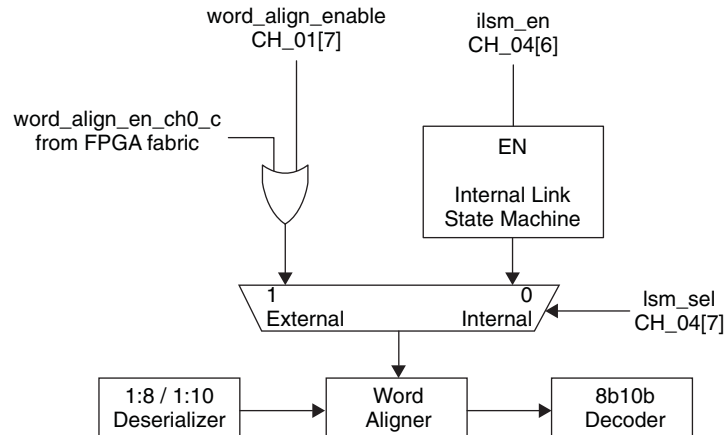
8b10b デコーダは、IEEE 802.3-2002 規格に記述された 8b10b デコーダ動作を実装しています。このデコーダはランニング・ディスパリティの検証と共に、10 ビットから 8 ビットコードへの変換を実行します。コード違反が検出されると、受信データ rxdata は 0xEE に、rx_k_chn は '1' に設定されます。

外部リンクステートマシンのオプション

内部リンクステートマシン属性 CHx_ILSM が “DISABLED”、CHx_RXWA（ワードアライメント）が “ENABLED” にそれぞれ設定されているときに、ワードアライナをイネーブルするためには制御信号 word_align_en_ch(0-3)_c が用いられます。これは FPGA ファブリックに実装される外部リンクステートマシンから生成されなければなりません。この信号が High になると、ワードアライナはアライメントをロックし、ロックしたままになります。受信データとユーザ定義のワードアライメント・キャラクタとの比較をやめ、COMMA_A または COMMA_B との最初の比較一致で現在のアライメントを維持します。再アライメントが必要な場合は、word_align_en_ch(0-3)_c を Low から High のパルスを与えます。ワードアライナは、いずれかのユーザ定義ワードアライメント・キャラクタとの次の一致で再ロックします。必要であれば、PCS クワッドの外部に実装したリンクステートマシンによって word_align_en_ch(0-3)_c を制御でき、特定の条件下でのみワードアライメントを変更することができます。

図 8-10 にリンクステートマシンのオプションを示します。

図 8-10. PCS ワードアライナとリンクステートマシンのオプション

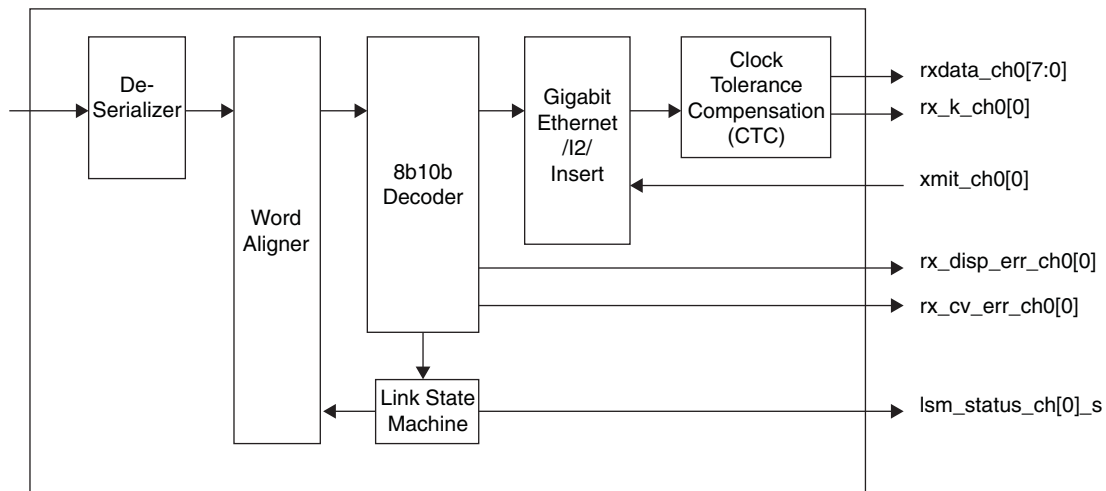


リンクステートマシンが選択され、特定のチャンネル用にイネーブルにした場合、リンクが同期するとそのチャンネルの lsm_status_ch(0-3)_s ステータス信号が High になります。

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 /I2/ 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 /I2/ オーダセット・シーケンスを挿入します。/I2/ 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 xmit_ch(0-3) 入力で制御されます。図 8-11 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 8-11. 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)

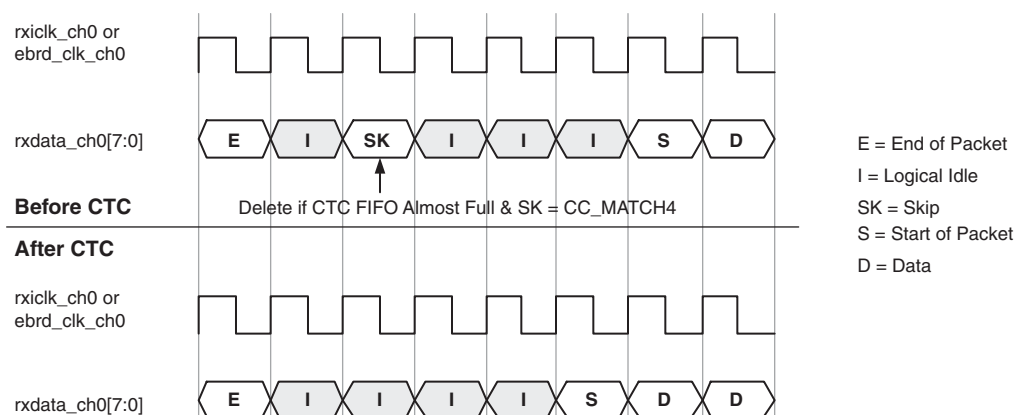
CTC (Clock Tolerance Compensation) モジュールは、受信再生クロックとロックする基準クロックの間のクロックレートの調整を実行します。クロック補償は、パケットデータの損失を引き起こすことなく、事前に定められた位置でバイトを挿入するか、または削除することによって実行されます。16 バイトの CTC FIFO は、2 つのクロックドメイン間のデータ転送に用いられ、LatticeECP3 SERDES のためにクロック偏差の規定

ppm 上限まで対応します (LatticeECP3 ファミリー・データシートの “DC and Switching Characteristics” セクションを参照してください)。

CHx_CTC 属性が “ENABLED” 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、“DISABLED” に設定されるとバイパスされます。

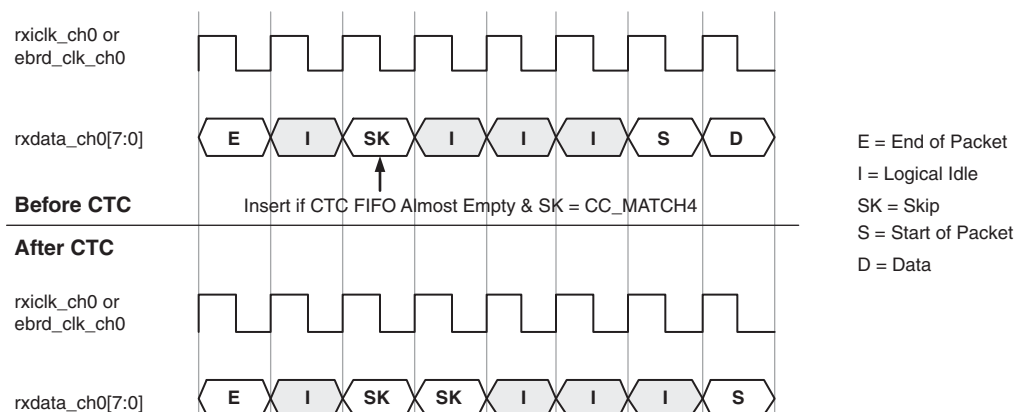
1 バイト削除について図 8-12 に示します。

図 8-12. クロックトレランス補償、1 バイト削除の例



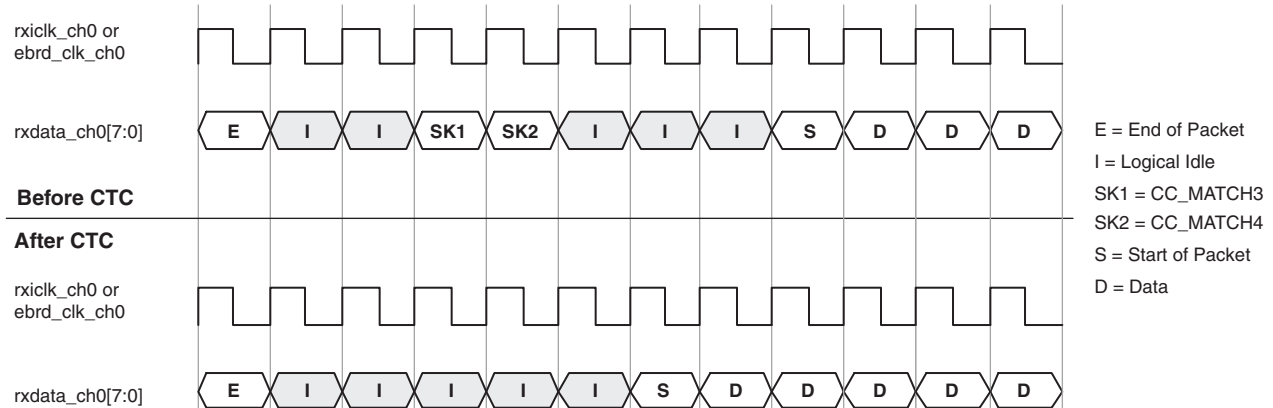
1 バイト挿入について図 8-13 に示します。

図 8-13. クロックトレランス補償、1 バイト挿入の例



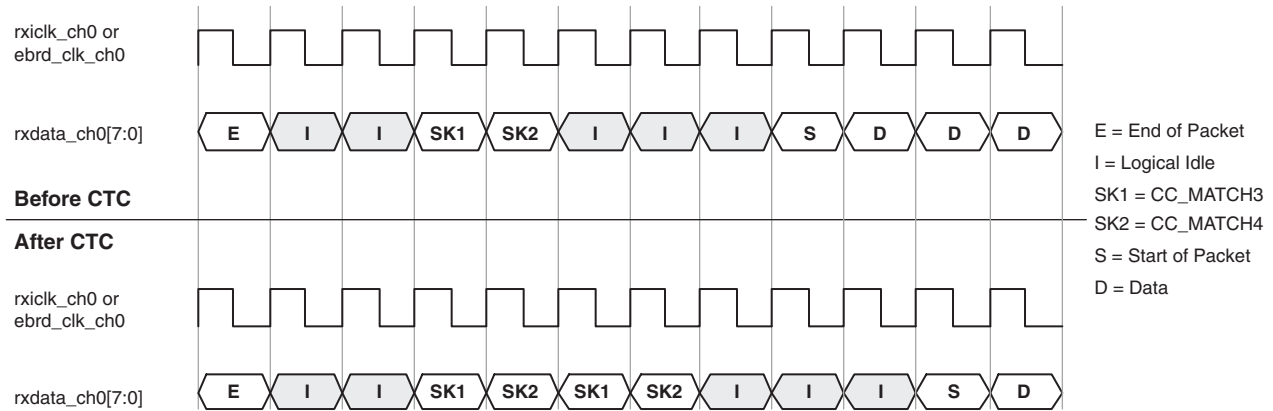
2 バイト削除について図 8-14 に示します。

図 8-14. クロックトレランス補償、2 バイト削除の例



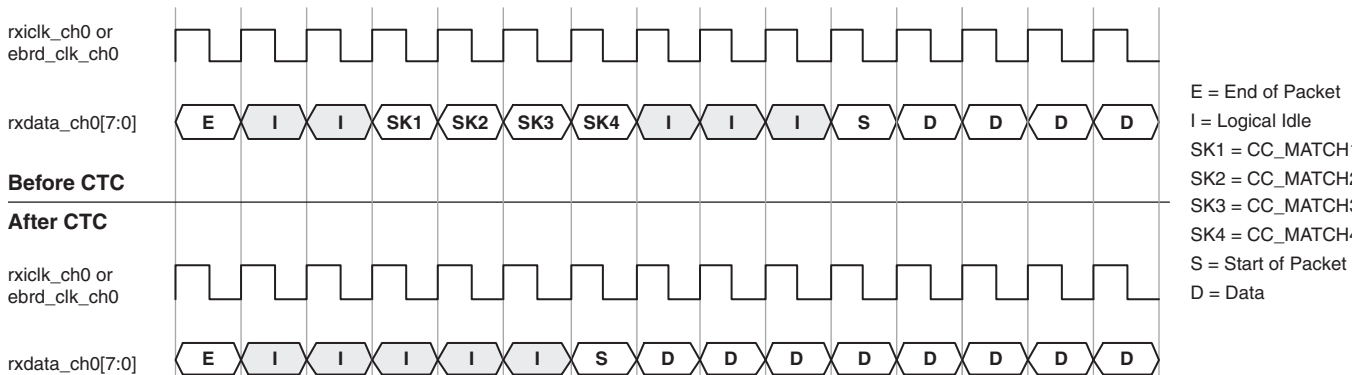
2 バイト挿入について図 8-15 に示します。

図 8-15. クロックトレランス補償、2 バイト挿入の例



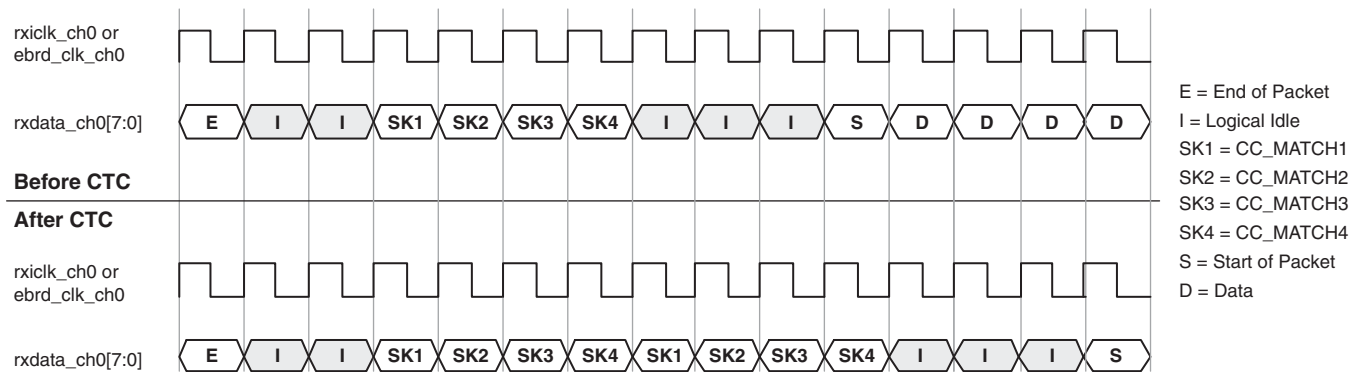
4 バイト削除について図 8-16 に示します。

図 8-16. 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 8-17 に示します。

図 8-17. クロックトレランス補償、4 バイト挿入の例



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、以下の設定を適用しなければなりません。

- ・ CC_MATCH_MODE 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_MATCH_MODE の値は、“1” (1 バイトの挿入 / 削除)、“2” (2 バイトの挿入 / 削除)、及び “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した最小パケット間ギャップを設定する必要があります。パケット間ギャップは CC_MIN_IPG 属性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”、及び “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を表 8-10 に示します。

- ・ 選ばれた CC_MATCH_MODE に対応するように、スキップバイトかオーダセットが設定される必要があります。4 バイトの挿入 / 削除 (CC_MATCH_MODE = “4”) に関しては、CC_MATCH1 属性に最初のバイトを割り当て、CC_MATCH2 属性に第 2 バイト、CC_MATCH3 属性に第 3 バイト、そして CC_MATCH4 属性に第 4 バイトをそれぞれ割り当てる必要があります。割り当てられる値は 2 進の 10 ビット値です。

例：

4 バイトのスキップ・オーダセットが /K28.5/D21.4/D21.5/D21.5 の場合、“CC_MATCH1” は “0110111100”、“CC_MATCH2” = “0010010101”、“CC_MATCH3” = “0010110101”、そして “CC_MATCH4” = “0010110101” となる必要があります。

2 バイトの挿入 / 削除 (CC_MATCH_MODE = “2”) に関しては、CC_MATCH3 に最初のバイトを割り当て、CC_MATCH4 に第 2 バイトを割り当てる必要があります。

1 バイトの挿入 / 削除 (CC_MATCH_MODE = “1”) に関しては、CC_MATCH4 にスキップバイトを割り当てる必要があります。

- ・ 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 ウォーターマークを設定する必要があります。値の範囲は 0 ～ 15 で、High ウォーターマークは Low ウォーターマークよりも高い値に設定する必要があります (等しい値に設定してはいけません)。CCHMARK 属性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。CCLMARK 属性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 の許容値は 16 進値の “0” から “F” です。
- ・ 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 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。
- ・ 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 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 8-10 は CC_MIN_IPG 属性で定義されるパケット間ギャップ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ています。この表は、乗数としてのパケット間ギャップを示しています。パケット間の最小バイト数と、挿入 / 削除のバイト数を表に示す乗数倍したものは等しくなります。例えば、挿入 / 削除あたりのバイト数が 4 (CC_MATCH_MODE を "4" に設定) で、かつ最小パケット間ギャップの CC_MIN_IPG が "2" のとき、最小のパケット間ギャップは 4 (CC_MATCH_MODE = "4" 掛ける 3 (表 8-10 の CC_MIN_IPG = "2")、すなわち 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 を通過するまで、PCS はスキップキャラクタの削除を実行しません。

表 8-10. 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	1x
1	2x
2	3x
3	4x

(注) デバイス品番 TW サフィックスの LatticeECP3-150EA デバイスファミリの CTC 対応について：末尾が TW となっている初期リリースの LatticeECP3-150EA デバイスでは、PCS が CTC に対応していません。CTC 機能がバイパスされ、ソフト IP で実装されています。ラティスの多くの IP コアでは、CTC ロジックがソフト形式で実装されています。

Diamond における IPexpress の使用

IPexpress は SERDES 及び PCS ブロックの生成と構成 (パラメータ等の指定) に使用されます。設計者は GUI を使用して、特定のクワッドまたはチャネルの SERDES プロトコル規格を選択します。IPexpress はこの GUI から入力を受け取り、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、シミュレーション及び論理合成フローで使用されます。自動構成ファイルには属性レベルのマップ情報が格納されます。このファイルは、シミュレーション及び bitgen プログラムの入力となります。設計者は変更と更新を IPexpress で行ってから、自動構成ファイルを再生成するようにしてください。一部の例外的な状況では、ユーザが構成ファイルを修正することもできます。

図 8-18 に IPexpress を使用して SERDES プロトコル規格用の SERDES/PCS ブロックを生成するためのツールフローを示します。

図 8-18. SERDES_PCS、Diamond のユーザーフロー

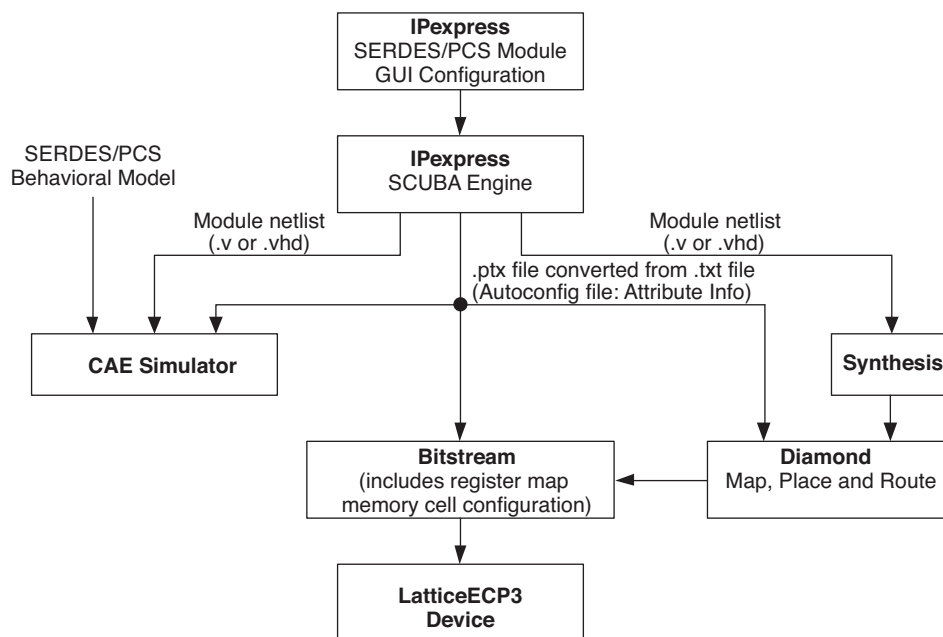
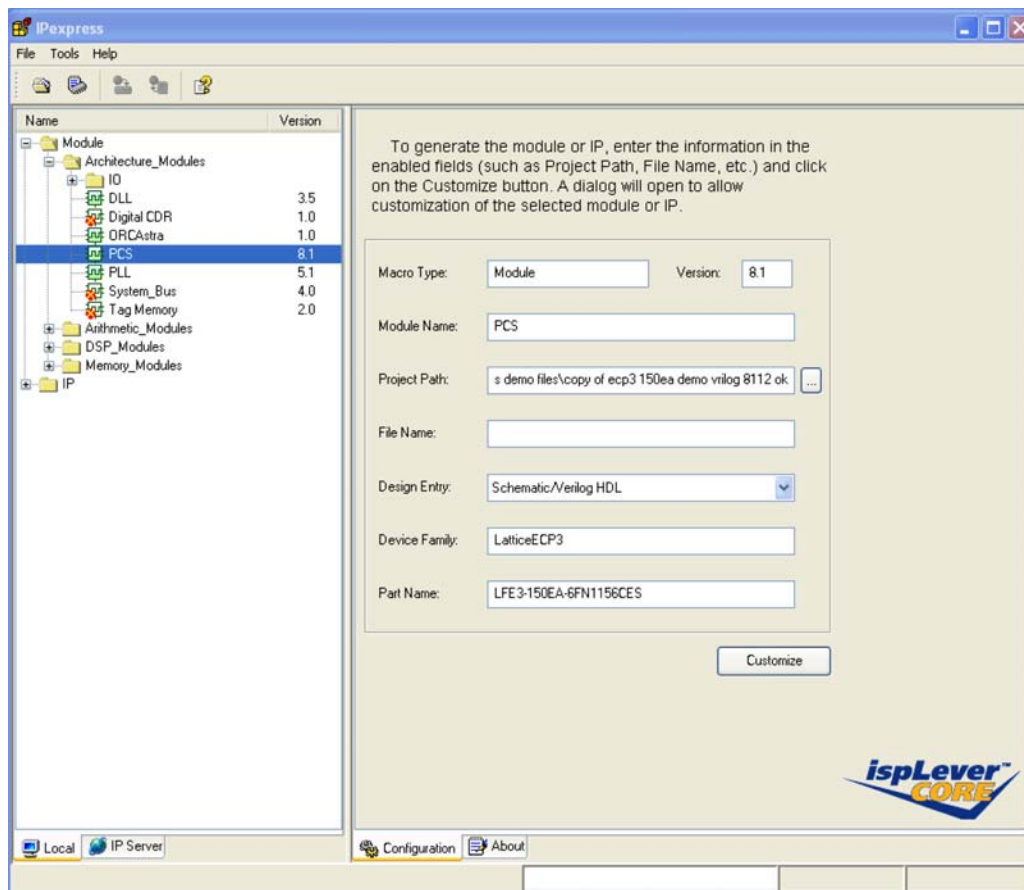


図 8-19 は IPexpress の GUI で PCS を選択したときのメインウィンドウを示しています。

図 8-19. IPexpress の PCS メインウィンドウ



クワッド・セットアップタブ

図 8-20 にファイル名を入力しメインウィンドウで **Customize** (カスタマイズ) ボタンをチェックしたときの、Quad (クワッド) セットアップタブ・ウィンドウを示します。このウィンドウで最初に行うことは、各チャネルのプロトコルモードの選択です。各チャネルは、'RX and TX' (送受信)、'RX Only' (受信のみ)、'TX Only' (送信のみ)、'Disabled' (ディセーブル)、または 'Low Speed Data Port' (低速データポート) として設定できます。

図 8-20. パラメータ設定 GUI ~ Quad (クワッド) セットアップタブ

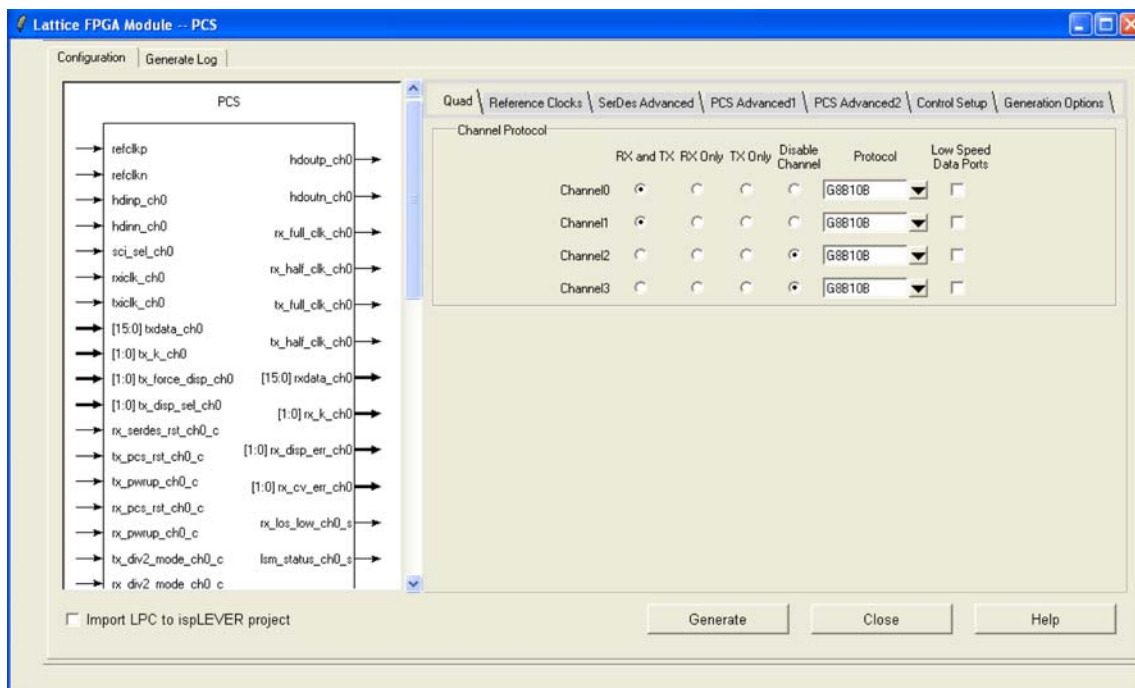


表 8-11. SERDES_PCS GUI の属性 ~ Quad (クワッド) タブの設定

GUI テキスト	属性名	範囲	デフォルト値
Channel Protocol (チャネルのモード)	CHx_MODE	RX and TX, RX Only, TX Only, DISABLE	DISABLE
Disable Channel 注1 (チャネルのディセーブル)	CHx_MODE	ENABLE, DISABLE	DISABLE
Protocol (プロトコル)	CHx_PROTOCOL	GIGE, SGMII, XAUI, SRIO, PCIE, SDI, G8B10B, 10BSER, 8BSER, CPRI, OBSAI	G8B10B
Low Speed Data Port (低速データポート)	CHx_LDR	RX and TX, RX Only, TX Only	DISABLED

1. ECP3-17EA の 328csBGA では、チャンネル 0 と 3 のみが使用できます

基準クロック・セットアップタブ

このタブでは TX 及び RX 基準クロックソースの属性を選択します。基準クロックは EXTERNAL または INTERNAL を選択できます。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が表示されます。

図 8-21. パラメータ設定 GUI ~ Reference Clocks (基準クロック) セットアップタブ

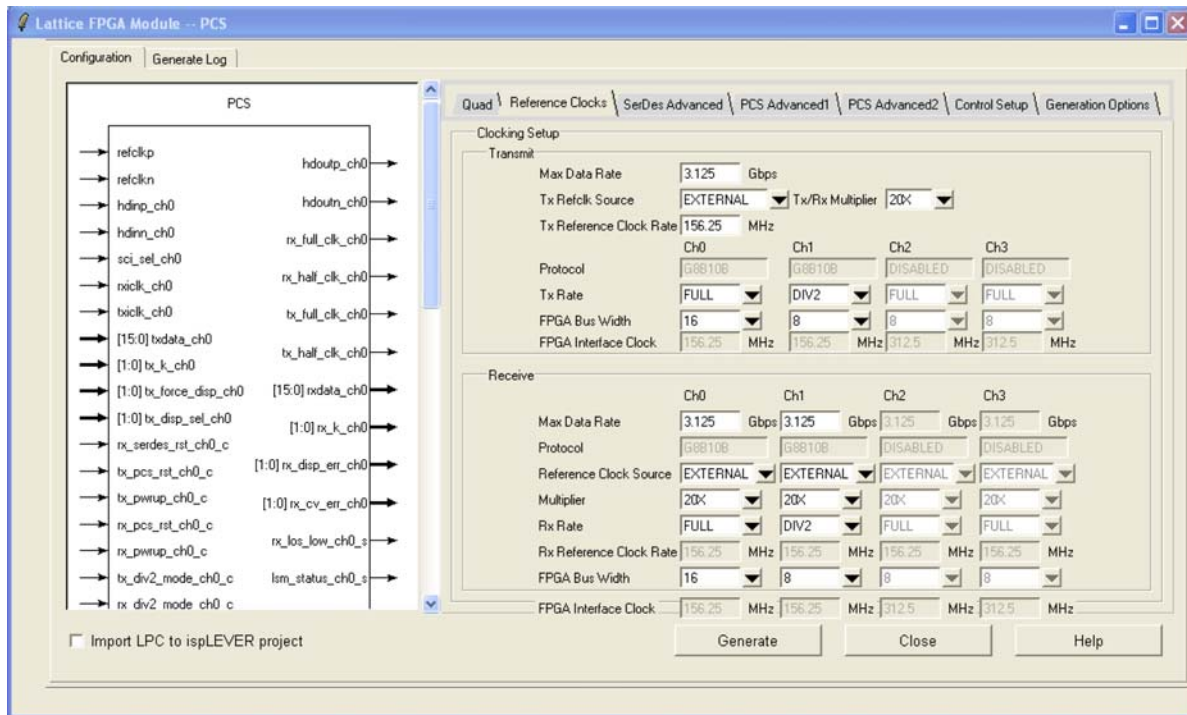


表 8-12. SERDES_PCS GUI の属性 ~ Reference Clocks (基準クロック) タブの設定

GUI テキスト	属性名	範囲	デフォルト値 (GUI)	デフォルト値 (属性)
送信				
Max. Data Rate ¹ (最大データレート)	N/A	0.23 ~ 3.2 Gbps	2.5 Gbps	N/A
TX Refclk Source (送信基準クロックソース)	PLL_SRC	INTERNAL, EXTERNAL	INTERNAL	REFCLK_INT
TX/RX Multiplier (乗数)	REFCK_MULT	8X, 10X, 16X, 20X, 25X	(プロトコル依存)	
TX Reference clock Rate (送信基準クロックレート)	#REFCLK_RATE ²		(プロトコル依存)	
Protocol (プロトコル)	ユーザ設定不可			
TX Rate (送信レート)	CH _x _TX_DATA_RATE	FULL, DIV2, DIV11	FULL	FULL
FPGA Bus Width (FPGA I/F バス幅)	CH _s _TX_DATA_WIDTH	8, 10, 16, 20	(プロトコル依存)	
FPGA Interface Clock (FPGA I/F クロックレート)	#CH0_TX_FICLK_RATE			
受信				
Max. Data Rate ¹ (最大データレート)	N/A	0.23 ~ 3.2 Gbps	2.5 Gbps	N/A
Protocol (プロトコル)	ユーザ設定不可			
Refclk Source (受信基準クロックソース)	CH _x _CDR_SRC	INTERNAL, EXTERNAL	INTERNAL	REFCLK_INT
Multiplier (乗数)	ユーザ設定不可			
RX Rate (受信レート)	CH _x _RX_DATA_RATE	FULL, DIV2, DIV11	FULL	FULL
RX Reference Clock Rate (受信基準クロックレート)	#CH0_RXREFCLK_RATE			
FPGA Bus Width (FPGA I/F バス幅)	CH _x _RX_DATA_WIDTH	8, 10, 16, 20	(プロトコル依存)	
FPGA Interface Clock (FPGA I/F クロックレート)	#CH0_RX_FICLK_RATE			

- レートは自動構成ファイルに反映されません。選択したデータレートに対して DATARATE RANGE の指定は次の通り：
150 Mbps ≤ LOWLOW ≤ 230 Mbps、230 Mbps < LOW ≤ 450 Mbps、450 Mbps < MEDLOW ≤ 0.9 Gbps、0.9 Gbps < MED ≤ 1.8 Gbps、1.8 Gbps < MEDHIGH ≤ 2.55 Gbps、2.55 Gbps < HIGH ≤ 3.2Gbps。
- '#' で始まる属性は、ユーザ情報専用を表しています。このような属性は、参照用として自動構成ファイルにも格納されます。

SERDES アドバンスト・セットアップ

このタブは4チャンネル全ての送受信 SERDES 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。送信属性としてはプリエンファシスや終端、差動出力電圧などを選択します。受信属性ではイコライザ、終端、I/O 結合などを選択します。送信 SERDES クロックと PLL の属性も選択します。

図 8-22. パラメータ設定 GUI ~ SERDES Advanced (アドバンスト) セットアップタブ

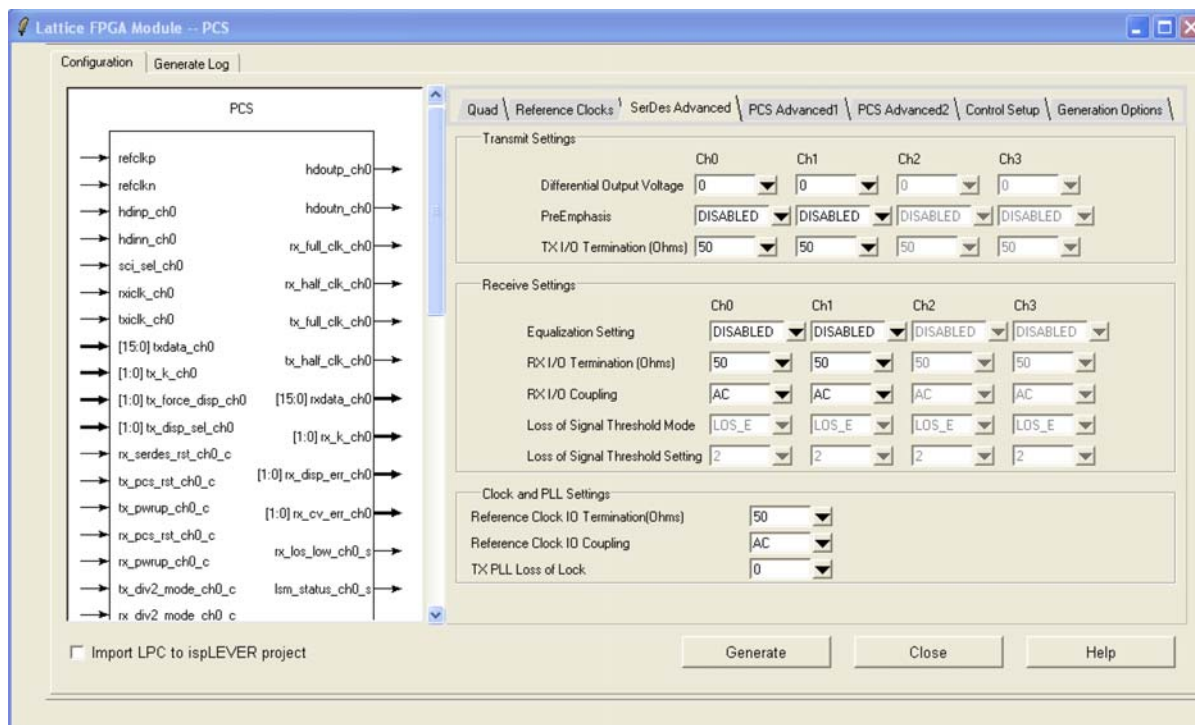


表 8-13. SERDES_PCS GUI の属性 ~ SERDES Advanced (アドバンスド) タブの設定

GUI テキスト	属性名	値	デフォルト値
Differential Output Voltage (差動出力電圧)	CHx_TDRV ⁸	-4 (640mV) ⁵ , -3 (780mV), -2 (870mV), -1 (920mV), 0 (1040mV:default), 1 (1130mV) ⁶ , 2 (1260mV) ⁷ , 3 (1350mV) ⁷ , 4 (1440mV) ⁷	0
PreEmphasis (プリエンファシス)	CHx_TX_PRE	Disabled, 0 (0%), 1 (5%), 2 (12%), 3 (18%), 4 (25%), 5 (33%), 6 (40%), 7 (48%)	DISABLED
TX I/O Termination (Ohms) ³ (送信 I/O 終端抵抗値)	CHx_RTERM_TX	50, 75, 5K	50
Equalization ¹ (イコライザ)	CHx_RX_EQ	Disabled, Mid_Low, Mid_Med, Mid_High, Long_Low, Long_Med, Long_High	DISABLED
RX I/O Termination (Ohms) ³ (受信 I/O 終端抵抗値)	CHx_RTERM_RX	50, 60, 75, High	50
RX I/O Coupling (受信 I/O 結合)	CHx_RX_DCC	AC, DC	AC ²
Loss of Signal Threshold (LOS 閾値)	CHx_LOS_THRESHOLD_LO	2 (+15%), 3 (+25%)	2 ⁴
TX PLL Reference Clock I/O Termination (Ohms) ³ (送信 PLL 基準クロック I/O 終端)	PLL_TERM	50, 2K	50
TX PLL Reference Clock I/O Coupling (送信 PLL 基準クロック I/O 結合)	PLL_DCC	AC, DC	AC ¹⁰
PLL Loss of Lock (PLL LOL 周波数偏差)	PLL_LOL_SET	0: +/- 1350ppm x2 ⁹ 1: +/- 2400ppm x2 2: +/- 6800ppm 3: +/- 400ppm	0

- 詳細については表 8-106 を参照してください。
- 内部オンチップ AC 結合の典型的コンデンサ値は 5pF です。
- 終端抵抗とその使い方：
 - RX I/O 終端：
 - 50：現在のところ、SMTPE 以外の全プロトコルでは 50 Ω 終端抵抗が使用されます。
 - 60：柔軟性のため用意されています。
 - 75：SMTPE では 75 Ω 終端抵抗が使用されます。
 - HIGH：Rx が未使用の場合のデフォルト。
 - TX I/O 終端：
 - 50：現在のところ、SMTPE 以外の全プロトコルでは 50 Ω 終端抵抗が使用されます。
 - 75：SMTPE では 75 Ω 終端抵抗が使用されます。
 - 5K：PCI Express の電気的アイドル、及び PCI Express の R レシーバ検出など。ユーザがレシーバ検出用にこの終端値を設定することはありません。“PCI Express のレシーバ検出”のセクションを参照してください。
 - TX PLL 終端：
 - 50：PCB 上に 50 Ω 終端抵抗がない場合。
 - 2K：PCB 上に 50 Ω 終端抵抗がある場合。
- PCS の設定では、GUI は PCI Express 以外の全てのプロトコルに対して値 2 のみに対応しています。PCI Express に対しては、値 2 と 3 の両方に対応しています。
- この振幅を得るためには TDRV_AMP_BOOST(CH13[3]) を 1 に設定します。
- この設定は PCI Express のデフォルト設定です。PCI Express プロトコルではこのデフォルト設定の使用を推奨します。そのため、IPexpress GUI の TDRV ドロップダウン・ウィンドウはグレー表示となります。他の設定は、自動構成ファイル (.txt ファイル) で CHn_TDRV 属性を編集して使用できます。
- これらの設定では VCCOB を 1.5V にする必要があります。
- 値は典型的数値です。周波数全域に対して約 ±20% のマージンがあります。詳細については、表 8-105 の CHn_TDRV の行を参照してください。
- 'x2' は、内部 LOL カウンタの ppm の正常な 2 倍カウントを示しています。
- 多くの用途では AC 結合を推奨します。DC 結合を使用する必要があるのは、外部 AC 結合コンデンサを組み合わせる場合のみです。

PCS の位置指定 (割り当て)

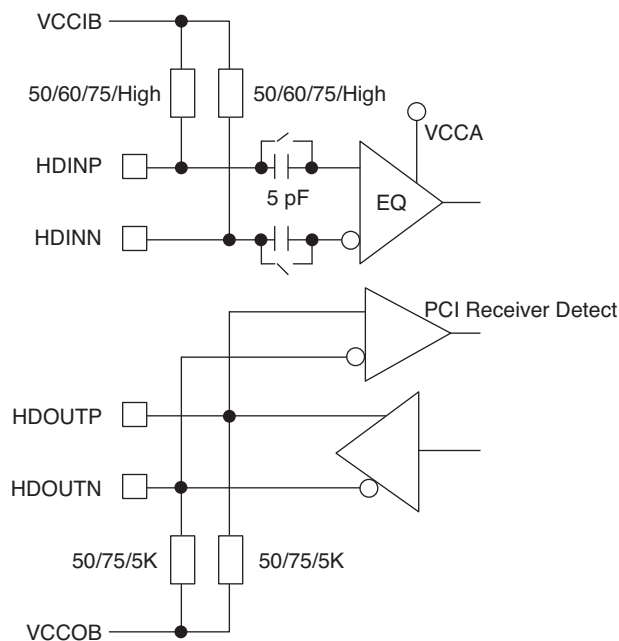
制約ファイル (.lpf) を記述することで、PCS クワッドの希望する位置を指定できます。制約 (プリファレンス) "locate" を使用します。構文例を次に示します。

```
LOCATE COMP "pcs_inst_name" SITE "PCSB" ;
```

クワッド名	サイト名
Quad A	PCSA
Quad B	PCSB
Quad C	PCSC
Quad D	PCSD

高速 I/O 終端トポロジを図 8-23 に示します。

図 8-23. 高速 I/O 終端



PCS アドバンスト 1 セットアップ

このタブは 4 チャンネル全ての送受信 PCS 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 及び RX チャンネルの極性と動作モード (8b10b など) は個別に選択できます。さらに、カンマ値、カンママスク、カンマアラインなどのワードアライメント値も選択できます。

図 8-24. パラメータ設定 GUI ～ PCS Advanced1 (アドバンスト 1) セットアップタブ

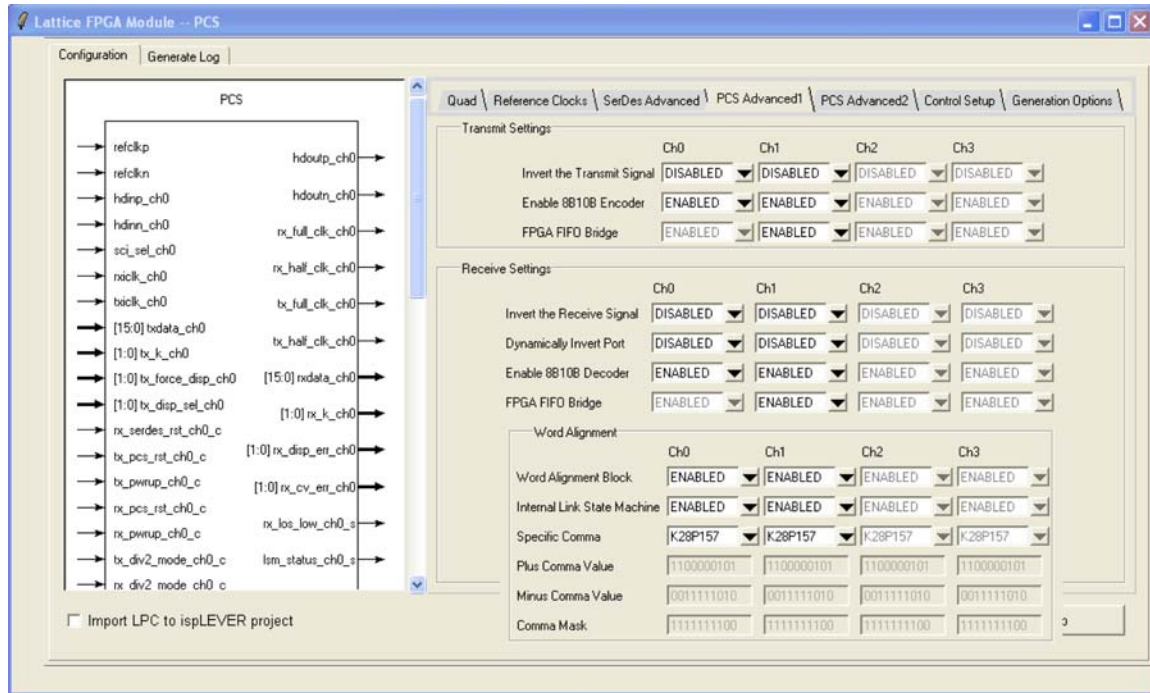


表 8-14. SERDES/PCS GUI ～ PCS Advanced1 (アドバンスト 1) セットアップタブ

GUI テキスト		属性名	デフォルト値
Transmitter (トランスミッタ)	Invert the Transmit Signal (送信信号を反転)	CHx_TX_SB	DISABLED
	Enable 8b10b Encoder (8B10B エンコーダをイネーブル)	CHx_TX_8B10B	(プロトコル依存)
	FPGA FIFO Bridge (FPGA FIFO ブリッジ)	CHx_TX_FIFO	(プロトコル依存)
Receiver (レシーバ)	Invert the Receive Signal (受信信号を反転)	CHx_RX_SB	DISABLED
	Dynamically Invert Port (ダイナミックにポートを反転)	N/A	DISABLED
	Enable 8b10b Decoder (8B10B デコーダをイネーブル)	CHx_RX_8B10B	(プロトコル依存)
	FPGA FIFO Bridge (FPGA FIFO ブリッジ)	CHx_RX_FIFO	(プロトコル依存)
Word Alignment (ワード・アライメント)	Word Alignment Block (WA ブロック)	CHx_RXWA	(プロトコル依存)
	Internal Link (内部リンク)	CHx_ILSM	(プロトコル依存)
	Specific Comma (特定のカンマ)	#CHx_SCOMMA	(プロトコル依存)
	Plus Comma Value (正のカンマ値)	CHx_COMMA_A ¹	1100000101
	Minus Comma Value (負のカンマ値)	CHx_COMMA_B	0011111010
	Comma Mask (カンママスク)	CHx_COMMA_M	(プロトコル依存) ²

- 定義上、COMMA_A と COMMA_B は正と負のランニング・ディスパリティを持つ 1 組の 8b10b コード化制御キャラクタです。プロトコルごとに正しい IDLE シーケンスを指定し、リンクステートマシンを同期する必要があります。例えば、1GbE プロトコルでは IDLE (ワードアライメントと同期ステートマシン) として K28.5+D5.6 または D16.2 が必要です。デフォルト値は、リトルエンディアン形式です。
- 多くの用途では、カンマキャラクタとして K28.5 が使用されます。マスクのデフォルト値は 1111111111 です。G8B10B モードでは任意のカンマを使用でき、マスクを 1111111100 にすると 3 つのカンマキャラクタ K28.1、28.5、28.7 を全て検出できます。

PCS アドバンスト 2 セットアップ

このタブはクロックトレランス補償ブロックの値の設定に使用されます。

図 8-25. パラメータ設定 GUI ～ PCS Advanced2 (アドバンスト 2) セットアップタブ

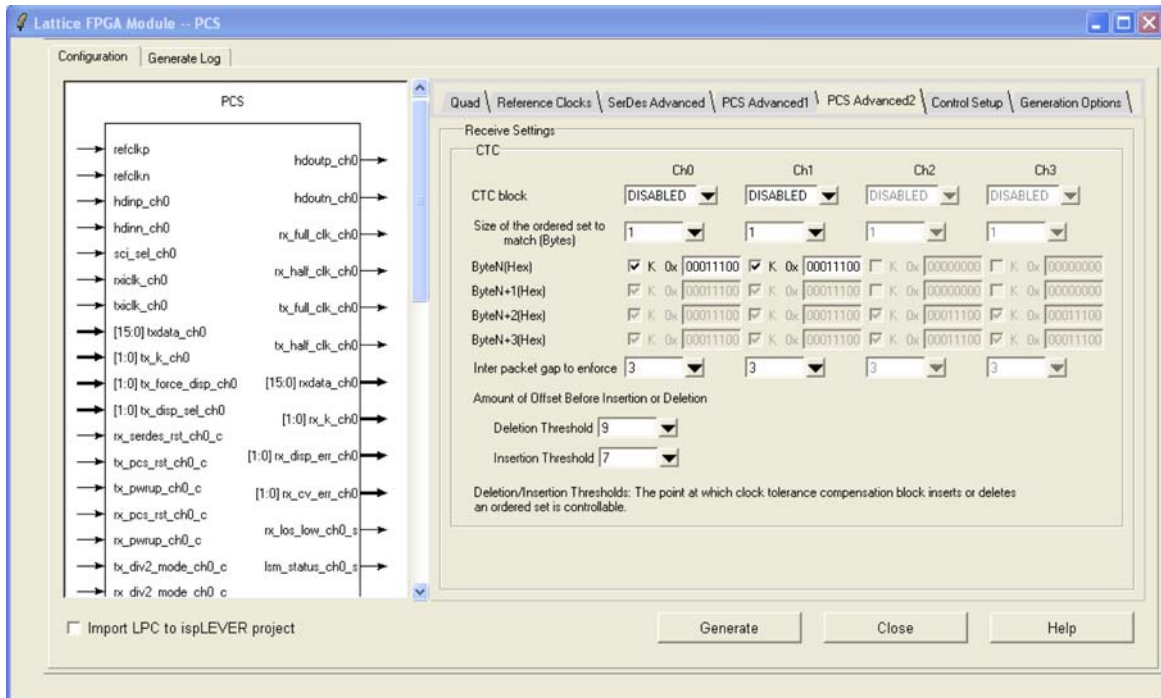


表 8-15. SERDES/PCS GUI ～ PCS Advanced2 (アドバンスト 2) セットアップタブ

GUI テキスト	属性名	デフォルト値
CTC block	CHx_CTC	(プロトコル依存) ¹
Size of ordered set (オーダーセットのサイズ)	CHx_CC_MATCH_MODE	(プロトコル依存)
Byte N	CHx_CC_MATCH1	(プロトコル依存)
Byte N+1	CHx_CC_MATCH2	(プロトコル依存)
Byte N+2	CHx_CC_MATCH3	(プロトコル依存)
Byte N+3	CHx_CC_MATCH4	(プロトコル依存)
Interpacket gap (パケット間ギャップ)	CHx_CC_MIN_IPG	(プロトコル依存)
Deletion threshold (削除の閾値)	CCHMARK	9
Insertion threshold (挿入の閾値)	CCLMARK	7

1. 常にディセーブル : XAUI、SDI、CPRI、OBSAI、10 ビット SERDES、8 ビット SERDES
常にイネーブル : Serial RapidIO
その他の全モード : デフォルトでディセーブル。ほとんどの CTC 機能は IP で提供されます。

制御セットアップ

このタブは SCI インターフェイスとデバッグ及び制御オプションの選択に使用されます。また、SCI、エラーレポート、PLL クォータ・クロック、ループバック機能もイネーブルにできます。

図 8-26. パラメータ設定 GUI ～ Control Setup (制御セットアップ) タブ

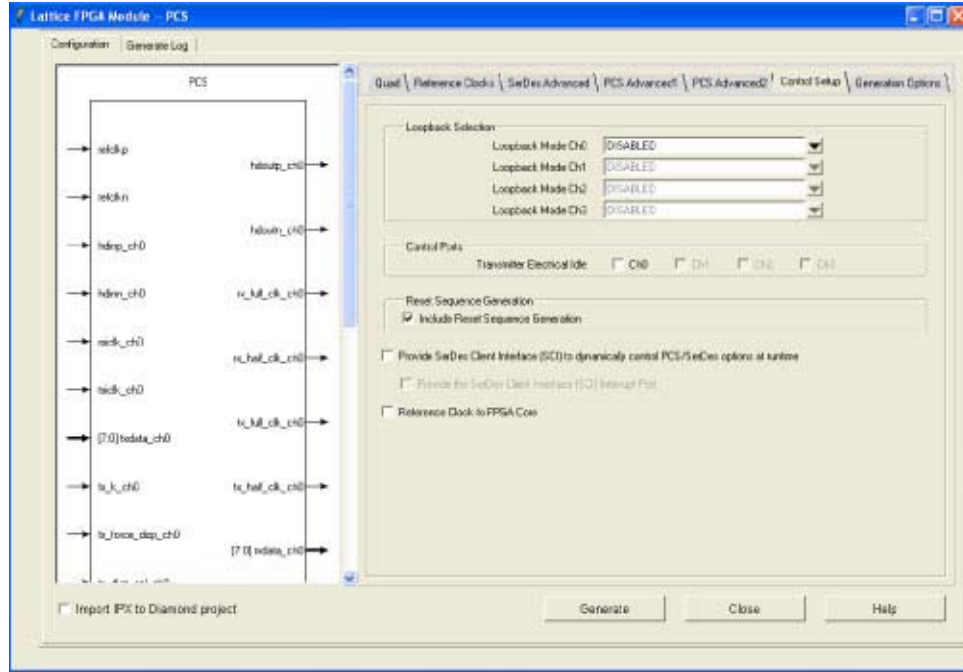


表 8-16. Tab 5, SERDES_PCS GUI 属性 ～ Control Setup (制御セットアップ) タブ

GUI テキスト	属性名	デフォルト値
Loopback Mode (Ch0, Ch1, Ch2, Ch3)	DISABLED Loopback serial data after equalizer Loopback serial data after transmit driver Loopback parallel data after de_serializer	DISABLED ¹
Transmitter Electrical Idle (トランスミッタ電氣的アイドル)	signal tx_idle_ch0_c is provided	DISABLED
Include Reset Sequence Generation ² (リセットシーケンス生成回路を含める)	Include the TX and RX Reset Sequence	ENABLED
Provide SERDES Client Interface (SCI インターフェイスを設ける)	N/A	
Provide the SERDES Client Interface Interrupt Port (SCI 割り込みポートを設ける)	INT_ALL	DISABLED
Reference Clock to FPGA core (基準クロックを FPGA ファブリックに出力する)	QD_REFCK2CORE	DISABLED

- ループバックモードがデフォルトモード (DISABLED) のときは、2つの SERDES ブリッジ・パラレルループバック制御信号 (sb_felb_ch[3:0]_c と sb_felb_rst_ch[3:0]_c) を HDL モジュールで使用し、ループバックモードを動的にオン / オフできます。ループバックモードを使用しない場合は、これらの信号をグラウンドに接続する必要があります。
- リセット・シーケンス生成については、本ドキュメントの "SERDES/PCS のリセット" セクションに説明があります。

FPGA への基準クロックとリセットシーケンス

リセットシーケンスはリセット制御用ステートマシンで基準クロックを使用します。

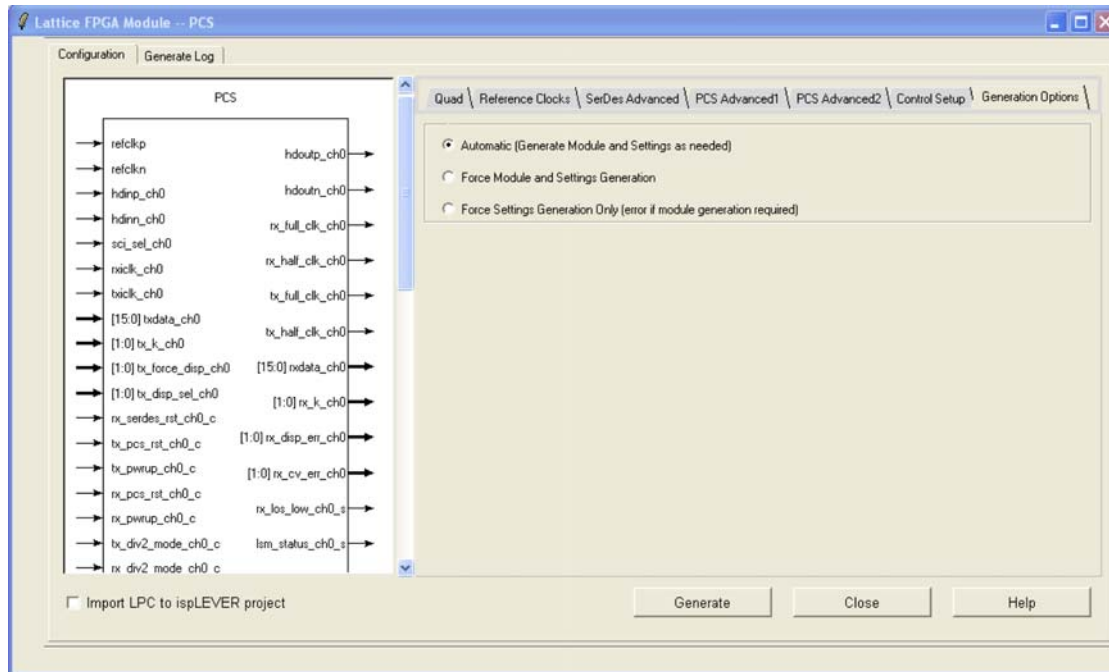
Tx Revclk ソースとして "Internal" が選択されると、リセット・ステートマシンは内部基準クロックを用います。

もし "External" が選択されると、リセット・ステートマシンは FPGA コアへの基準クロックを用います。REFCLK2FPGA 信号がラップモジュールに現れ（制御レジスタビット、QD_0A[1] がセット）、これはユーザも使用できます。

生成オプション

このタブには PCS モジュール生成出力ファイルをユーザが自分で選択するためのオプションがあります。

図 8-27. パラメータ設定 GUI ~ Generation Options (生成オプション) タブ



以前のバージョンの Diamond で作成した古いプロジェクトを最新バージョンに初めて移行する場合は、パラメータ設定を変更する必要がなくても、最新バージョンで PCS モジュールを再生成するようにしてください。そうすることで、最新のプリミティブ・ライブラリが確実に使用されるようになります。再生成しなかった場合は、デザインがフローの途中で停止するか、フローは完了するものの、ボードが予期しない動作となる可能性があります。

既存のプロジェクト内の PCS モジュールを再生成した場合、HDL ファイルには変化がなく、自動構成ファイルのみの再生成が必要になることがよくあります。その場合は、"Generate Bitstream Data"（ビットストリーム・データの生成）及び "Force One Level"（1 レベルのみ）オプションを実行するとコンパイル時間を節約できます。

- ・ Automatic（自動）～ 選択すると、IPexpress で必要なファイルのみが生成されます。HDL ファイルと TXT ファイルの両方が含まれる場合と、TXT ファイルのみの場合があります。これがデフォルト設定です。
- ・ Force Module and Settings Generation（強制的にモジュールと設定を生成）～ 選択すると、HDL ファイルと TXT ファイルの両方が生成されます。Project Navigator の処理が強制的にリセットされ、論理合成に戻ります。
- ・ Force Settings Generation Only（強制的に設定のみを生成）～ 選択すると、TXT ファイルのみが生成されます。HDL の生成が必要な場合は、エラーメッセージが表示されます。
- ・ フロー定義 ～ 生成オプションは 2 つのモジュールフローで異なる機能を持ちます。
 - HDL ソースフロー：プロジェクトナビゲータ内の HDL ファイル
再生成のため、既存の LPC ファイルを IPexpress から開くことができます。この場合、GUI で設定したり

セット点が新しい開始点になります。そのため、ユーザがプロセスをダブルクリックするか、“Force One Level”（1レベルのみ）オプションを実行すると、そのリセット点から開始されます。

- **LPC ソースフロー：プロジェクトナビゲータ内の LPC ファイル**

LPC ファイルを開いて PCS モジュールを再生成すると、HDL モジュールが再生成されたかどうかに関係なく、プロセス全体がリセットされます。

- どちらの場合も、Processes（プロセス）ウィンドウのチェックマークは解除されたままですが、ユーザがプロセスを開始するとすぐに更新されます。

自動構成ファイルについて

IPexpress は属性レベルのマップ情報が格納されたこのファイルを生成します。このファイルはシミュレーション・モデルで使用され、ビットストリーム生成プロセスでも、PCSD クワッドを IPexpress で選択されたモードに自動的に初期化するために使用されます。

自動構成ファイルではファイルタイプ拡張子として “txt” が使用されます。

以下に自動構成ファイルの例を示します。

```
# This file is used by the simulation model as well as the bitstream
# generation process to automatically initialize the PCSD quad to the mode
# selected in the IPexpress. This file is expected to be modified by the
# end user to adjust the PCSD quad to the final design requirements.
```

```
DEVICE_NAME "LFE3-95E"
CH0_MODE "RXTX"
CH1_MODE "DISABLED"
CH2_MODE "DISABLED"
CH3_MODE "DISABLED"
TX_DATARATE_RANGE "HIGH"
PLL_SRC "REFCLK_EXT"
REFCK_MULT "10X"
#REFCLK_RATE 250.0

CH0_PROTOCOL "G8B10B"
CH0_LDR "RXTX"
CH0_RX_DATARATE_RANGE "HIGH"
CH0_TX_DATA_RATE "FULL"
CH0_TX_DATA_WIDTH "8"
CH0_TX_FIFO "DISABLED"
CH0_CDR_SRC "REFCLK_EXT"
#CH0_TX_FICLK_RATE 250.0
CH0_RX_DATA_RATE "FULL"
CH0_RX_DATA_WIDTH "8"
CH0_RX_FIFO "DISABLED"
#CH0_RX_FICLK_RATE 250.0
CH0_TDRV "0"
CH0_TX_PRE "DISABLED"
CH0_RTERM_TX "50"
CH0_RX_EQ "DISABLED"
CH0_RTERM_RX "50"
CH0_RX_DCC "AC"
CH0_LOS_THRESHOLD_LO "2"
CH0_TX_SB "DISABLED"
```

CH0_TX_8B10B	"ENABLED"
CH0_RX_SB	"DISABLED"
CH0_RX_8B10B	"ENABLED"
CH0_RXWA	"ENABLED"
CH0_ILSM	"ENABLED"
#CH0_SCOMMA	"1111111111"
CH0_COMMA_A	"1100000101"
CH0_COMMA_B	"0011111010"
CH0_COMMA_M	"1111111100"
CH0_CTC	"ENABLED"
CH0_CC_MATCH_MODE	"2"
CH0_CC_MATCH1	"0000000000"
CH0_CC_MATCH2	"0000000000"
CH0_CC_MIN_IPG	"3"
CH0_SSLB	"DISABLED"
CH0_SPLBPORTS	"DISABLED"
CH0_PCSLBPORTS	"DISABLED"
PLL_TERM	"50"
PLL_DCC	"AC"
PLL_LOL_SET	"0"
CCHMARK	"9"
CCLMARK	"7"
INT_ALL	"DISABLED"
QD_REFCK2CORE	"ENABLED"

8ビット及び10ビット SERDES-Only モード

このセクションでは SERDES/PCS ブロックの、8ビット SERDES-Only 及び 10ビット SERDES-Only の2つのモードの動作について説明します。これらのモードは、LatticeECP3 PCS ロジックが提供するプロトコルベースの操作なしで、高速 I/O インターフェイスへのアクセスが必要な用途を目的としたものです。

送信パス

- ・ シリアライザ：8ビットまたは10ビットパラレルデータをシリアルデータに変換

受信パス

- ・ デシリアライザ：シリアルデータを8ビットまたは10ビットパラレルデータに変換
- ・ オプションの、ユーザ定義のアライメントパターンへのワードアライメント

Generic 8b10b モード

SERDES/PCS ブロックの Generic 8b10b モードは、追加のプロトコル固有のデータ操作なしで、8b10b エンコード / デコードが必要な用途を目的としたものです。LatticeECP3 SERDES/PCS ブロックは、チャンネルあたり最大 3.2Gbps の Generic 8b10b アプリケーションに対応しています。このモードでは、ワードアライナは組み込み PCS LSM (Link State Machine) から制御できます。この場合、LSM は GbE モードで動作し、RX_SERDES 入力のトラフィックは IEEE802.3-2002 仕様の Figure 36-9 (Synchronizing State Machine, 1000BASE-X) に準拠しなければなりません。

組み込みリンクステートマシンを選択してイネーブルにした場合、リンクが同期すると lsm_status_ch[3:0]_s ステータス信号は High になります。

さもなくば LSM はディセーブルでき、ワードアライナはファブリックから word_align_en_ch[3:0]_c 入力ピンで制御できます。詳細は本ドキュメント 21 ページの ” 外部リンクステートマシンのオプション ” 節、および 32 ページの ”SERDES アドバンスト・セットアップ” 節を参照してください。

送信パス

- ・ シリアライザ
- ・ 8b10b エンコーダ

受信パス

- ・ デシリアライザ
- ・ 組み込み GbE リンクステートマシンから、ユーザ定義ワードアライメント・キャラクタへのワードアライメント
- ・ 8b10b デコード
- ・ クロックトレランス補償（オプション）

LatticeECP3 PCS のギガビット・イーサネット及び SGMII モード

LatticeECP3 SERDES/PCS ブロックのギガビット・イーサネットモードは、シリアル I/O から、IEEE 802.3-2002 1000 BASE-X ギガビット・イーサネット規格の GMII/SGMII インターフェイスまでの完全互換に対応しています。

送信パス

- ・ シリアライザ
- ・ 8b10b エンコーダ

受信パス

- ・ デシリアライザ
- ・ IEEE 802.3-2002 1000 BASE-X で定義されたアライメント・キャラクタに基づくワードアライメント
- ・ 8b10b デコード
- ・ ギガビット・イーサネット・リンクステートマシンは、1 つの例外を除いて IEEE 802.3-2002 の Figure 36-9（同期ステートマシン、1000BASE-X）に準拠しています。Figure 36-9 では、LSM が 1 つの SYNC_ACQUIRED_{N}（N=2,3,4）から SYNC_ACQUIRED_{N-1} に遷移するために、4 つの連続した正しいコードグループを受信することが求められています。それに対して、実際の LSM 実装では遷移するために、5 つの連続した正しいコードグループが必要です。
- ・ ギガビット・イーサネット・キャリア検出：IEEE 802.3-2002（1000BASE-X）のセクション 36.2.5.1.4 では carrier_detect 機能が定義されています。ギガビット・イーサネットモードでは、この機能が PCS に含まれず、FPGA ファブリックに carrier_detect 信号が供給されません。
- ・ クロックドメインの差異に対処するクロックトレランス補償ロジック機能

ギガビット・イーサネット（1000BASE-X）のアイドル挿入

これは、クロック補償と自動ネゴシエーションに必要です。自動ネゴシエーションは FPGA ロジックで行われます。ラティスのギガビット・イーサネット PCS IP コアは、以下で説明する自動ネゴシエーションを提供します。

クロック補償と自動ネゴシエーションにはアイドルパターン挿入が必要です。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/ のオーダセットを送信します。クロック補償回路はこれらのオーダセットを削除せず、/I2/ オーダ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には、定期的に /I2/ オーダセットを挿入し、挿入 / 削除する機会をクロック補償回路に与える必要があります。

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 /I2/ オーダセット（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、この動作によって何らかのランニング・ディスパリティ誤りが発生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 にドライブされるため、/I2/ オーダセットが 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完了すると、受信データの破損を防ぐため、/I2/ 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になります。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 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となり、TX データバスに含まれています。

表 8-17. GbE IDLE ステートマシンの制御/ステータス信号

モジュール信号	方向	記 述
xmit_ch[3:0]	In	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

ギガビット・イーサネットのアイドル挿入と correct_disp_ch[3:0] 信号

PCS の送信側で correct_disp_ch[3:0] 信号が使われることで、パケット間ギャップが負のディスパリティ状態で必ず始まります。イーサネットフレームの終わりでは、イーサネットフレームのサイズとデータ内容に応じて、トランスミッタの現在のディスパリティ状態が正の場合と負の場合があります。

ただし、PCS の FPGA ソフトロジック側からは、PCS トランスミッタの現在のディスパリティ状態が不明です。correct_disp_ch[3:0] 信号はこのために用意されています。パケット間ギャップの開始時に 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、PCS トランスミッタに対して、送信データストリームに IDLE1 オーダセットが強制的に挿入されます。ただし、現在のディスパリティが負の場合は、送信データストリームに変更が加えられません。

PCS 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、次のような IDLE2 オーダセットの連続伝送になります：tx_k_ch=1、txdata= 0xBC tx_k_ch=0、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を表します。IDLE1 は、現在のディスパリティが反転されるべきことを表します。したがって、パケット間ギャップが確実に負のディスパリティ状態で始まるようにすることができます。パケット間ギャップ前のディスパリティ状態が負の場合は、パケット間ギャップの間、IDLE2 の連続ストリームが送信されます。パケット間ギャップ前のディスパリティ状態が正の場合は、IDLE1 が 1 つ送信されてから、IDLE2 の連続ストリームが送信されます。

PCS の FPGA ソフトロジック側では、PCS へのパケット間ギャップは常に IDLE2 にドライブされます。パケット間ギャップの最初の開始時は、correct_disp_ch[3:0] 信号がクロック 1 周期分アサートされ、k_cntrl=0、data=0x50 となります。必要に応じて、PCS はこの IDLE2 を IDLE1 に変換します。残りのパケット間ギャップでは、PCS への IDLE2 がドライブされ、correct_disparity_chx 信号がネゲートされたままになる必要があります。

例えば、512 バイトのイーサネットフレームの連続ストリームと、512 バイトの /I/ が送信された場合は、次のようになります。

- ・ 最初のパケット間ギャップ中は、全ての負のディスパリティ /I2/ となる (K28.5(-) D16.2(+))
- ・ 次のパケット間ギャップ中は、周期が正のディスパリティ /I1/ で始まり (K28.5 (+), D5.6 (+/-) が等しい)、残り全てのオーダセットは負のディスパリティ /I2/ となる
- ・ 次のパケット間ギャップ中は、全ての負のディスパリティ /I2/ となる
- ・ 次のパケット間ギャップ中は、周期が正のディスパリティ /I1/ で始まり (K28.5 (+), D5.6 (+/-) が等しい)、残り全てのオーダセットは負のディスパリティ /I2/ となる
- ・ エンコーダ・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。次のものがある

ります。

- ・ 負または正のディスペリティをワード単位で強制する機能
- ・ FIFO ブリッジから外部マルチプレクサにデータを直接入力する機能
- ・ ランニング・ディスペリティに応じてコードワードを置換する機能 (100BASE-X 及び FC)
- ・ ソフトウェアレジスタで制御するバイパスモード

XAUI モード

ラティス XAUI IP コアでは、XAUI モードの SERDES/PCS ブロックが、シリアル I/O から IEEE 802.3-2002 XAUI 規格の XGMII インターフェイスまで完全互換です。XAUI モードは 10 ギガビット・イーサネットに対応しています。

送信パス

- ・ シリアライザ
- ・ IEEE 802.3ae-2002 仕様に従い、XGMII アイドルから ||A||、||K||、||R|| キャラクタへの変換を実行する送信ステートマシン
- ・ 8b10b エンコード

受信パス

- ・ デシリアライザ
- ・ IEEE 802.3-2002 で定義されたアライメント・キャラクタに基づくワードアライメント
- ・ 8b10b デコード
- ・ XAUI リンクステートマシンは、1 つの例外を除いて IEEE 802.3ae-2002 の図 48-7 - PCS 同期状態図に準拠しています。図 48-7 では、LSM が 1 つの SYNC_ACQUIRED_[N] (N=2,3,4) から SYNC_ACQUIRED_[N-1] に遷移するために、4 つの連続した正しいコードグループを受信することが求められています。それに対して、実際の LSM 実装では遷移するために、5 つの連続した正しいコードグループが必要です。
- ・ PCS 内のクロックトレランス補償ロジックは、XAUI モードでディセーブルとなります。MCA (マルチチャネル・アライメント) 及び CTC は XAUI IP コアで行われます。
- ・ x4 マルチチャネル・アライメントは FPGA コアロジック内で行う必要があります。

LatticeECP3 PCS の PCI Express リビジョン 1.1 (2.5Gpbs) モード

PCI Express モードの SERDES/PCS ブロックは、x1、x2、及び x4 PCI Express アプリケーションに対応しています。

送信パス

- ・ シリアライザ
- ・ 8b10b エンコード
- ・ レシーバ検出
- ・ 電氣的アイドル

受信パス

- ・ デシリアライザ
- ・ 同期コードに基づくワードアライメント
- ・ 8b10b デコード
- ・ リンク同期ステートマシンは、IEEE 802.3ae-2002 10GBASE-X 仕様の PCS 同期ステートマシン (図 48-7) で定義された動作を取り入れて機能します。
- ・ 1 つの PCS クワッドを PCI Express モードに設定して x2 または x4 PCI Express 動作
- ・ クロックドメインの差異に対処するクロックトレランス補償ロジック機能

- ・ x2 または x4 マルチチャネル・アライメントは FPGA コアロジック内で行う必要があります。

表 8-18 に PCI Express モード固有のポートを示します。

表 8-18. PCI Express モード固有のポート

信号	方向	クラス	記 述
pcie_done_ch[3:0]_s	Out	チャネル	1 = ファーエンド・レシーバ検出完了 0 = ファーエンド・レシーバ検出未完了
pcie_con_ch[3:0]_s	Out	チャネル	ファーエンド・レシーバ検出の結果 1 = ファーエンド・レシーバ検出 0 = ファーエンド・レシーバ不検出
pcie_det_en_ch[3:0]_c	In	チャネル	FPGA ロジック（ユーザロジック）が SERDES ブロックに、PCI Express レシーバ検出動作の要求があることを通知 1 = PCI Express レシーバ検出をイネーブル 0 = 通常動作
pcie_ct_ch[3:0]_c	In	チャネル	1 = トランスミッタにファースト・レシーバ検出を要求 0 = 通常データ動作
rxstatus[2:0]	Out	チャネル	チャネルごとの PCI Express レシーバ・ステータスポート。RxStatus# はレシーバ・データパスのエンコード済みステータスです。16 ビット・データバスモードの場合は 2 ビット幅になります。

ステータス信号 rxstatus はレシーバ・データパスのエンコード済みステータスです。次のようにエンコードされます。

表 8-19. rxstatus のエンコード

rxstatus[2:0]			記 述	優先度
0	0	0	受信データ OK	8
0	0	1	CTC が 1 バイト挿入	7
0	1	0	CTC が 1 バイト削除	6
0	1	1	レシーバ検出 (pcie_done, pcie_con)	1
1	0	0	8b10b デコードエラー（コード違反 ~ rx_cv_err）	2
1	0	1	CTC FIFO オーバーフロー (ctc_orun)	3
1	1	0	CTC FIFO アンダーフロー (ctc_urun)	4
1	1	1	受信ディスパリティ・エラー (rx_disp_err)	5

PCI Express の終端

電気レベルにおいて、PCI Express は各レーンに対して 2 つの 2.5Gbps の単方向低電圧差動信号ペアを利用します。送信と受信は個別の差動ペアとなり、レーンごとに合計 4 つのデータ線があります。イコライザをプログラム可能な入力レシーバと、プリエンファシスをプログラム可能な出力トランスミッタによって、リンクを最適化できます。PCI Express 仕様では、差動ラインを受信側でコモンモードで終端することが求められています。各リンクには、ファースト（レシーバ側）に終端抵抗が必要です。使用する抵抗の公称値は 100 Ω です。これは、図 8-28 に示すように、CML 入力の組み込み終端機能を使用して実現されます。仕様では、リンクの送信側に AC 結合コンデンサ（CTX）が必要です。このコンデンサによって、送信デバイスと受信デバイスのコモンモードのバイアス不一致が解消されます。このコンデンサは、ラティス CML 出力に外付けする必要があります。

PCI Express L2 ステート

PCI Express の L2 ステートでは、rx_pwrup_c 信号をネゲートして RX チャネルをパワーダウンしてはいけません。これを行うと RX 終端がハイインピーダンスとなり、ファーストでレシーバを検出できなくなります。rx_pcs_rst_c 信号を使用してチャネルをリセット状態に保ち、電力を節約するようにしてください。

図 8-28. PCI Express のインターフェイス図

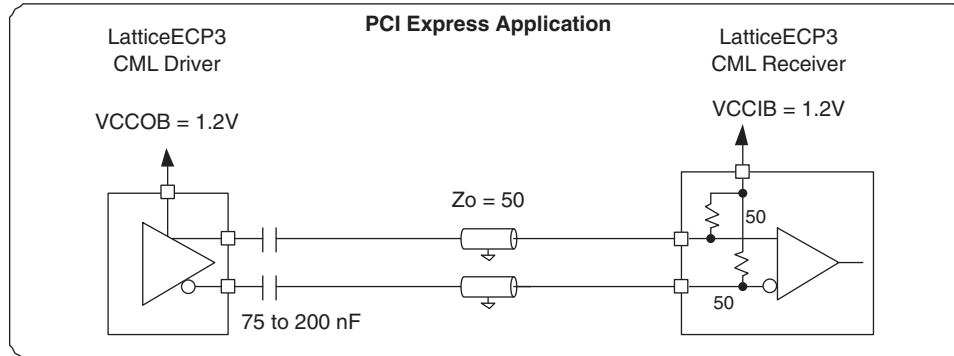


表 8-20. PCI Express の差動仕様

シンボル	パラメータ	Min.	Nom.	Max.	単位	コメント	位置
ZTX-DIFF-DC	DC 差動 TX インピーダンス	80	100	120	Ω	TX DC 差動モード低インピーダンス。ZTX-DIFF-DC は、DC 動作点で測定されたトランスミッタの小さな信号抵抗であり、TX が論理 1 または論理 0 に静的にドライブされているときに、D+ と D- に 100 Ω 抵抗を接続したときの値と同等です。	内部
ZRX-DIFF-DC	DC 差動入ラインピーダンス	80	100	120	Ω	LTSSM の全ステート時の、RX DC 差動モードインピーダンス。基本リセットから検出 (LTSSM の初期状態) に送信する場合、ポートの全ての未構成レーンでレシーバの終端値が満たされるまでに、5ms の遷移時間が発生します。	内部
CTX	AC 結合コンデンサ	75		200	nF	全てのトランスミッタを AC 結合する必要があります。媒体内または送信部品自体の内部で AC 結合する必要があります。	外部

PCI Express 電氣的アイドル送信

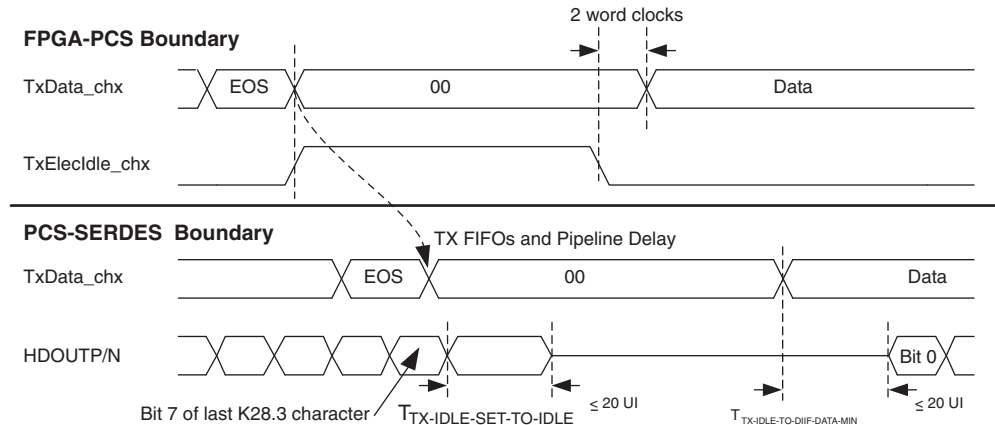
電氣的アイドルは安定状態条件であり、トランスミッタの P 及び N 電圧が同じ値で一定に保たれます (電氣的アイドル差動ピーク出力電圧 VTX-IDLE-DIFFp が 0 ~ 20mV の間)。電氣的アイドルは、節電及び非アクティブ状態で主に使用されます。

PCI Express 基本仕様では、トランスミッタは電氣的アイドルになる前に、必ず電氣的アイドル・オーダセット (EIOS)、K28.5 (COM)、及びそれに続いて 3 つの K28.5 (IDL) を送信する必要があります。電氣的アイドル・オーダセットの最後のシンボル送信後、トランスミッタは TTX-IDLE-SET-TO-IDLE が 20UI 未満と規定された、有効な電氣的アイドル・ステートになる必要があります。

そのため、FPGA コアから PCS への電氣的アイドル・イネーブル (tx_idle_chx_c) が、送信データごとに送信されます。この信号は PCS-SERDES 境界までの全経路で同様にパイプライン化されます。全ての有効なデータでこの信号は Low になります。電氣的アイドルを開始するには、FPGA ロジックが最後の K28.5 (IDL) シンボル送信後のクロックでこの信号を High にプルします。この信号は PCS-SERDES 境界までパイプライン化されるため、送信データとこの信号との関係は、FPGA-PCS 境界とまったく同一になります。

PCS-SERDES 境界での電氣的アイドル・イネーブル信号の立ち上がりエッジから 14UI 後に、最後の K28.3 (IDL) シンボルの最後のビット (ビット 7) が送信されます。16UI (<20UI) 後、送信差動バッファは電氣的アイドル・ステートになります。

図 8-29. 電氣的アイドルの送信



FPGA コアロジックが、トランスミッタが電氣的アイドル・ステートを続ける必要があるとみなしている間、FPGA コアロジックは電氣的アイドル・イネーブル (tx_idle_chx_c) 信号をアクティブ (High) にした状態で、データ (できればオール 0) をクロックと同期する必要があります。トランスミッタは最低 50UI (20ns) の間 (TTX-IDLE-MIN)、電氣的アイドル・ステートを続ける必要があります。

PCI Express 電氣的アイドル検出

クワッドの各チャネルには LOS (信号の喪失) 検出器があります。電氣的アイドルは、電氣的アイドル・オーダセット (EOS) の 3 つの K28.3 (IDL) シンボルの 2 つを受信すると検出されます。電氣的アイドル・オーダセットの受信後、レシーバは最低 50ns (TTX-IDLE-MIN) 待ってから、電氣的アイドル終了検出器をイネーブルする必要があります。

これらの信号 (チャネルごとに 1 つ、クワッドごとに 4 つ) は PCS 経由で送信する必要があり、FPGA コアで利用できるようにする必要があります。そうすることで、電氣的アイドルに対応するために必要なステートマシンを FPGA コアで構成できます。

PCI Express レシーバ検出

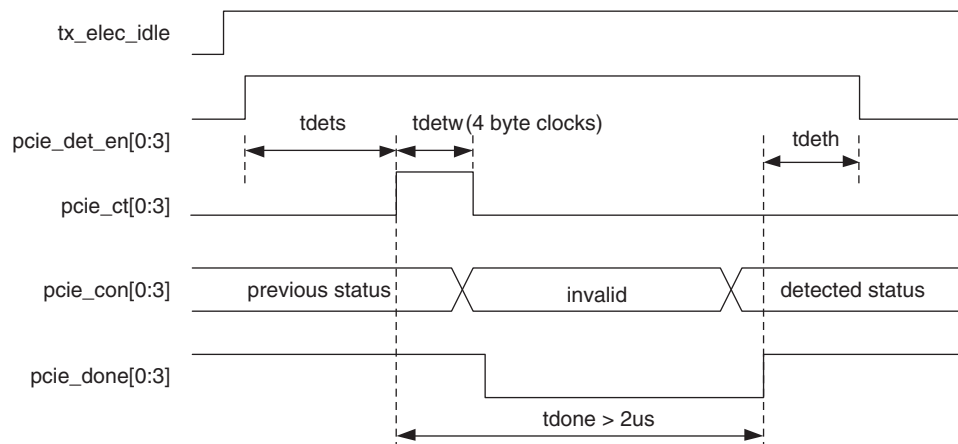
図 8-30 にレシーバ検出シーケンスを示します。レシーバ検出テストは、クワッドの各チャネルで独立して実行できます。レシーバ検出テストを始める前に、tx_idle_ch#_c 入力を High に設定することで、トランスミッタを電氣的アイドル・ステートにする必要があります。レシーバ検出テストは、pci_det_en_ch#_c を High にドライブすることで、対応する tx_elec_idle が High に設定された 120ns 後に開始されます。これはドライバ終端をハイインピーダンスに設定し、またハイインピーダンス・ドライバ終端で差動の両出力を VCCOB にプルすることによって、対応する 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。その後チャネルの pcie_ct_ch#_c 入力を 4 バイト (ワード) クロック周期の間 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。対応するチャネルの pcie_done_ch#_s は非同期的にクリアされます。十分な時間 (送信側の時定数で決まる) が経過してレシーバ検出テストが終了後、pcie_done_ch#_s レシーバ検出ステータスポートは High になり、pcie_con_ch#_s ポートでレシーバ検出ステータスをモニタできるようになります。そのとき、pcie_con_ch#_s ポートが High であればレシーバはそのチャネルで検出され、Low であればそのチャネルでレシーバが検出されません。レシーバ検出テストの完了後は、tx_idle_ch#_c をネゲートできます。

レシーバ検出は次のように行われます。

1. ユーザが pcie_det_en を High にドライブし、対応する TX ドライバをレシーバ検出モードにする。これで、ドライバ終端がハイインピーダンス ($5k\Omega$) に設定され、差動ドライバの両方の出力がハイインピーダンスのドライバ終端を通じてコモンモードにプルされます。TX ドライバは短時間後にこの状態になり、pcie_det_en が最低 120ns の間 High にドライブされてから、pcie_ct がアサートされます。
2. ユーザが 4 バイト・クロックの間 pcie_ct を High にドライブする
3. SERDES が対応する pcie_done を Low にドライブする
4. SERDES が必要な長さ（時定数に基づく）だけ内部信号（pcie_ct に対応するもの）をドライブし、レシーバを検出する
5. SERDES が対応する pcie_con 接続ステータスをドライブする
6. SERDES が対応する pcie_done を High にドライブする
7. ユーザが、このアサート状態の pcie_done を使用して pcie_con ステータスをサンプリングし、レシーバ検出が成功したかどうかを判定する

図 8-30. PCI Express モードのレシーバ検出シーケンス



PCI Express パワーダウン・モード

rx_pwrup_ch[3:0] 信号ではなく、rx_serdes_rst_ch[3:0] リセット信号を使用する必要があります。そうすることで、RX 終端が 50Ω でイネーブルのままとなり、フーエンド・トランスミッタがレシーバの接続を検出できます。

PCI Express ビーコンのサポート

このセクションでは、LatticeECP3 PCS がビーコン検出と送信にどのように対応できるかについて詳しく説明します。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への対応で与えられます。

ビーコン検出の要件

- ・ 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 未満になっている必要があります。
- ・ DC バランスは 32us 未満で回復される必要があります。
- ・ 500 ns を超えるパルス幅に関しては、出力ビーコン電圧レベルは VTX-DIFFp-p (800mV から 1200mV) よりも 6dB 低くなっている必要があります。
- ・ 500 ns 未満のパルス幅に関しては、出力ビーコン電圧レベルは VTX-DIFFp-p 以下で、VTX-DIFFp-p

よりも 3.5dB 低いレベル以上になっている必要があります。

PCS のビーコン検出への対応

- ・ 信号喪失 (LOS) 閾値検出回路は、受信バッファが指定された電圧レベルになっているかどうかを検知します。
- ・ これは rlos_lo_ch(0-3) 信号で示されます。
- ・ PCI Express 電氣的アイドル検出と (パワーステート P2 での) PCI Express ビーコン検出に、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 VTX-DIFFpp から 6dB 低いビーコン出力電圧 (すなわち 201mV) を持つことができます。この信号を検出できた場合は、ビーコンが検出されたことになります。

PCS のビーコン送信への対応

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 間隔で幅 2ns の周期的なパルス (1.0UI=400ps、5 倍すると 2 ns) が発生します。これで、要件の下限が満たされます。そして、出力ビーコン電圧レベルは VTX-DIFFp-p になります。これは有効なビーコン送信です。

SDI (SMPTE) モード

SDI モードの LatticeECP3 SERDES/PCS ブロックは、SD-SDI、HD-SDI、3G-SDI という 3 つの SDI モード全てに対応します。

送信パス

- ・ シリアライザ

受信パス

- ・ デシリアライザ
- ・ オプションで、ユーザ定義のアライメントパターンへのワードアライメント

ビデオ放送業界では、次のデータレートが最も一般的です。

- ・ SD-SDI (SMPTE259M): 270Mbps
- ・ HD-SDI (SMPTE292M): 1.485Gbps, $1.485\text{Gbps}/1.001 = 1.4835\text{Gbps}$
- ・ 3G-SDI (SMPTE424M): 2.97Gbps, $2.97\text{Gbps}/1.001 = 2.967\text{Gbps}$

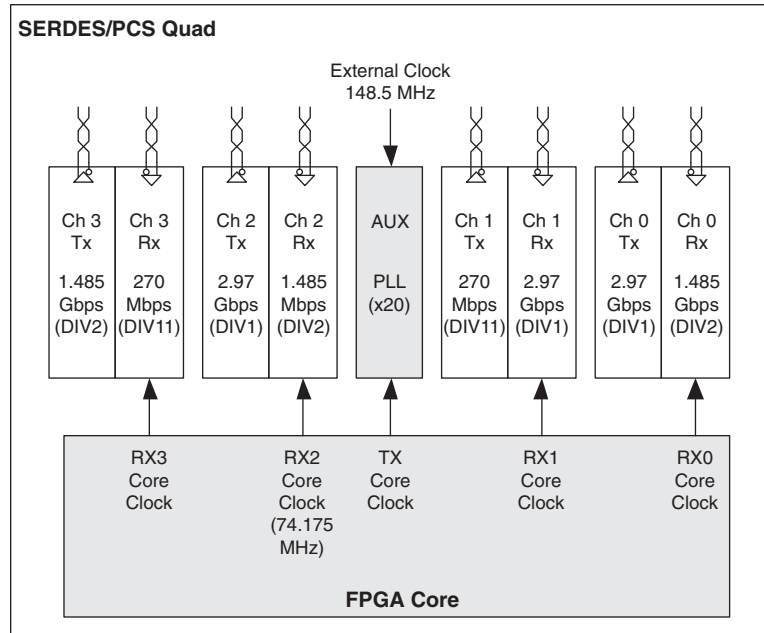
多くの設計者は、これら全てのレートに対応できるようにしようと考えています。その理由は、放送スタジオや、衛星中継局またはケーブル中継局では、事前に受信データレートが分かるとは限らないからです。

異なるレート間の切り替え時間は、可能な限り短くする必要があります。CDR の再ロック時間は避けることができません。LatticeECP3 SERDES では、PLL を再ロックする必要がありません。これは、LatticeECP3 には RX 及び TX ごとに分周器があることで可能になりました。一般的に、ビデオリンクは片方向性です (チャネルごとにレートが異なり、さらに重要なことに、同一チャネルの RX と TX のレートが異なることもあります)。

また、装置が配置される地域によっては、ビデオ送信時にフル HD/3G-SDI レート (ヨーロッパ / アジア) が使用されることも、フラクショナルレート (北米: NTSC) が使用されることもあります。その結果、ラティスではクワッドの使用率の高いマルチレート SMPTE に対応する、2 つのソリューション例を開発しました。

3G/HD Full TX レートとフラクショナル TX レートへの同時対応は、同一 SERDES クワッド内では不可能なことに注意してください。一般的に上記のことがあるため、地理的な使い分けは制限事項として許容されません。

図 8-31. 例 A : 3G/HD/SD フル RX/TX レートへの対応と 3G/HD フラクショナル TX レートへの対応



主要な用途の要件に対応するため、RX 及び TX ごとに DIV が選択可能です。LatticeECP3 には DIV11 が追加されています。マルチレート構成の可能性の1つとして、プライマリピンから TX PLL への 148.5MHz REFCLK の供給があります。TX PLL は x20 モードになります。得られる出力クロックは 2.97GHz です。そして、1.485Gbps 用に DIV2、270Mbps 用に DIV11 を使用すると、PLL の再トレーニング及びロックなしで、非常に短時間の切り替えが可能になります。

Serial RapidIO (SRIO) モード

このセクションでは SERDES/PCS ブロックの Serial RapidIO モードの動作について説明します。LatticeECP3 は、1 つの PCS クワッドで 1x 及び 4x の Serial RapidIO 用途に対応しています。SRIO1.0 は 3.125Gbps と 2.5Gbps、1.25Gbps という複数の周波数に対応していることが特長です。これらのレートの比率は 2.5:2:1 です。同一クワッド内の整数分周器でこれら全てのレートに対応することは不可能ですが、2.5 Gbps と 1.25 Gbps の比率は 2:1 (フルレート : ハーフレート) です。

送信パス

- ・ シリアライザ
- ・ 8b10b エンコード

受信パス

- ・ デシリアライザ
- ・ RapidIO 物理レイヤ 1x/4x LP- シリアル仕様で規定された同期コードグループに基づくワードアライメント
- ・ 8b10b デコード
- ・ クロックドメインの差異に対処するクロックトレランス補償ロジック機能

シリアル・デジタルビデオ及び OOB (Out-Of-Band) 低速 SERDES 動作

LatticeECP3 SERDES/PCS はレシーバ CDR 及び関連する SERDES/PCS ロジックをバイパスすることで、SERDES TX PLL 及び RX CDR がネイティブ・サポートするレート (<250Mbps : OOB 信号) よりも遅いあらゆるデータレート (100Mbps ファースト・イーサネット、143Mbps または 177Mbps の SD-SDI など) に対応します。これらの OOB パスは主に低データレートを使用しますが、他の機能上の理由から高レートも使用できます。詳細については、本ドキュメントのマルチレート SMPTE 対応のセクションを参照してください。

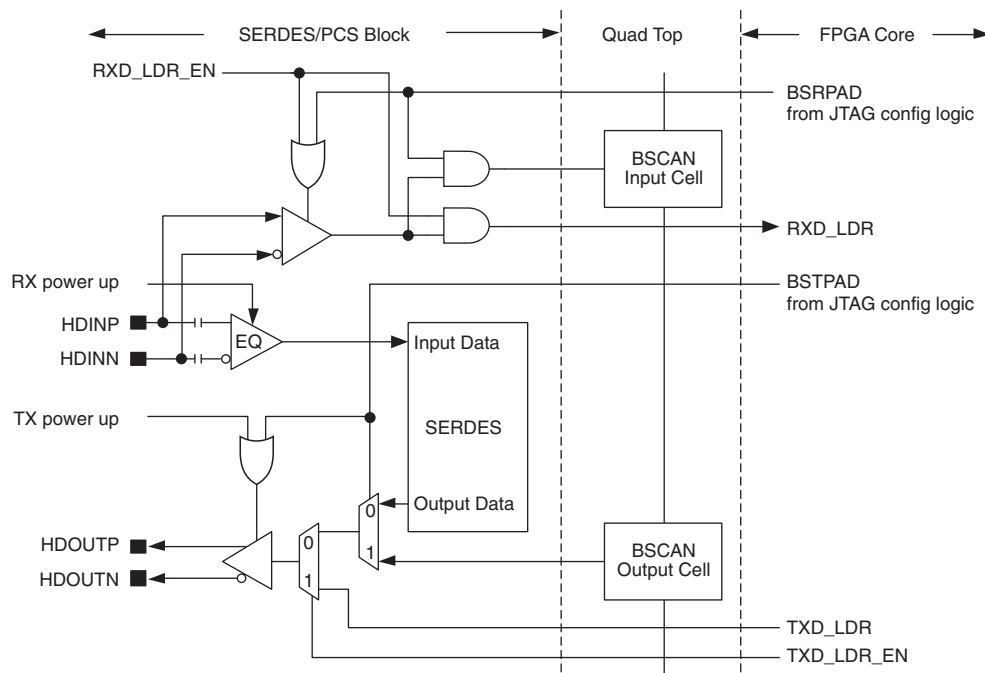
また SD-SDI では、HD-SDI レートの差動 RX ペアと同一のペアに、これらのレートの共存が必要になることがあります (SD-SDI レートがアクティブになってから、データレートを HD-SDI レートに切り替えることがあります)。この 2 つのレートのどちらが有効になるかは予測できないため、高速 SERDES (クワッド内に存在) と低速 SERDES (クワッド外部に実装) の 2 つの SERDES に、入力データストリームを平行に送信する方法があります。可能な実装の 1 つを図 8-32 に示します。

チャンネルごとに、RX バッファから FPGA コアへの低データレート・シングルエンド入力として入力 RXD_LDR があります。コアの低速 CDR (Clock Data Recovery) ブロックまたは DRU (Data Recovery Unit) は、ソフトロジックを使用して作成できます。このデータパスは、チャンネル・レジスタビット RXD_LDR_EN でイネーブルできます。他のレジスタビットでイネーブルされた場合は、FPGA からの信号でも LatticeECP3 内のこのデータパスをイネーブルできます。

送信方向では、FPGA コア内のシリアルライザの内蔵ソフトロジックを使用し、TXD_LDR ピンを使用してデータを SERDES に送信することもできます。通常の高速 SERDES パスが境界スキャンパスで多重化される位置に近い、プリエンファシス・ロジックの直前で送信データが多重化されます。この概念を図 8-32 に示します。低データレート・パスは、チャンネル・レジスタビット TX_LDR_EN を設定して選択できます。

または、出力側で高速 SERDES を使用して高速データを送信したり、デシメーションを使用して低速データを送信します (SERDES は高速で動作し続けますが、出力データはデシメーション係数 n クロックごとに変化します)。

図 8-32. シリアル・デジタルビデオに対応する実装例



OBSAI (Open Base Station Architecture Initiative)

OBSAI は携帯基地局のオープンな市場を対象とした、オープンフォーラムです。

LatticeECP3 SERDES/PCS は、3.84Gbps レートを除くほとんどの OBSAI 機能に対応しています。

送信パス

- ・ シリアルライザ
- ・ 送信ステートマシンをギガビット・イーサネットモードに設定
- ・ 8b10b エンコード

受信パス

- ・ デシリアルライザ
- ・ IEEE 802.3-2002 1000 BASE-X で定義されたアライメント・キャラクタに基づくワードアライメント
- ・ 8b10b デコード

BTS (Basestation Transceiver System) には 4 つのメイン・コンポーネント / モジュールがあり、それらの間に 3 つの主要インターフェイスとして RP (Reference Point、参照点) があります。

- ・ RP3: 携帯デバイス (端末) からの信号を RF モジュールが受信し、それをデジタルデータにダウンコンバートします。
- ・ RP2: エンコード済みの信号をベースバンドモジュールが受け取って処理し、伝送モジュールに送ります。伝送モジュールはそれを地域ネットワークに送出します。
- ・ RP1: 制御モジュールが、この 3 つの機能間の調整を行います。

現在のところ、業界の最大の関心事は低周波 RF モジュールとパワーアンプの提供であるため、OBSAI の主要な活動は RP3 (Reference Point 3) を規定することです。実際、重要な仕様は RRH (Remote Radio Head) を対象とした RP3-01 です。

OBSAI RP3 電気仕様は XAUI 電気仕様に基づいたものであり、基地局トランシーバ・システムのニーズに合わせてカスタマイズされています。XAUI 電気インターフェイスは IEEE 802.3ae-2002 の第 47 節で規定されています。RP3 バージョン 3.1 では 3.84Gbps、3.072Gbps、2.304Gbps、1.536Gbps、0.736Gbps という電氣的レートを規定され、その中の最後の 4 つに対応しています。

RP3 電気仕様ではレシーバのコンプライアンス・マスクが規定され、トランスミッタ出力マスクのサンプルが提供されます。BER は 1×10^{-15} を上回る必要があり、XAUI の要件である 1×10^{-12} よりも厳しいものとなっています。RP3 電気仕様は、UI の規定も XAUI 仕様と異なっています。XAUI では $\pm 100\text{ppm}$ の差が許容されます。BTS は完全な同期システムであることから、この差は OBSAI システムに適用されません。

BTS は同期システムであるため、全てのバス間の遅延の測定と調整が不可欠です。OBSAI でこのことをよく検討した結果、RP3 リンク間でマスタフレームを同期する方法が考案されました。送受信モジュールでは、処理、バッファ遅延、及びリンク間のレイテンシなどのあらゆる要因が遅延調整で考慮されます。

データリンク・レイヤでのもう 1 つの重要な点は、トランスミッタとレシーバの同期です。同期することで、実際のデータがリンク上で正常にデコードされるようになります。エラーの頻度及び同期ステータスは、継続的にモニタされます。

RP3-01 ではそれをさらに推し進め、768Mbps の整数倍で最高 3.84Gbps のラインレートが規定され、OBSAI 互換のラインレートが検討されています。多数のラインレートを利用できるため、リモート RF ユニットとローカルユニットとの間の自動ネゴシエーションが規定されました。この仕様の拡張には、2 つの RP3-01 ノード間のイーサネット伝送、RRH には物理 RP1 リンクがないことに対する RP1 情報の RP3 リンクへのマッピング、遅延測定、RP3-01 ユニット間の同期、RP3-01 リンク間のデータ多重化などがあります。

LatticeECP3 SERDES/PCS 内の各機能ブロックの遅延については、“CPRI” セクションで説明します。

CPRI (Common Public Radio Interface)

CPRI の目標は、基地局メーカ及び部品ベンダが共通プロトコルを共有し、あるユーザのプラットフォームを別のユーザに簡単に適合できるようにすることです。

送信パス

- ・ シリアライザ
- ・ 送信ステートマシンをギガビット・イーサネットモードに設定
- ・ 8b10b エンコード

受信パス

- ・ デシリアライザ
- ・ IEEE 802.3-2002 1000 BASE-X で定義されたアライメント・キャラクタに基づくワードアライメント
- ・ 8b10b デコード

OBSAI とは異なり、CPRI では機械的または電氣的インターフェイス要件が規定されていません。範囲に関して、CPRI は OBSAI よりも対象がはるかに狭く、CPRI は RRH とベースバンドモジュールとの間のリンクだけを対象としています。CPRI の用語では、これらのモジュールがそれぞれ RE (Radio Equipment) 及び REC (Radio Equipment Control) と呼ばれます。すなわち、CPRI では OBSAI RP3 仕様と同じインターフェイスが規定されます。CPRI は、主にインターフェイスの物理及びデータリンク・レイヤを対象としています。また、ユーザプレーン・データ、C&M (制御及び管理) プレーン、及び同期プレーン・データの転送方法も規定されます。

有力企業のバックアップがあること、及び 1 つのインターフェイスリンク (RF モジュールとベースバンドモジュールの間) だけに注目し、しかも物理及びデータリンク・レイヤに主に注目しているという 2 つの理由から、CPRI は優れた “牽引力” を持っています。

CPRI には 614.4Mbps、1.2288Gbps、2.4576Gbps、3.072Gbps という 4 つのライン・ビットレート・オプションがあり、最低でもこの 1 つのレートに対応する必要があります。高いラインレートは、そのすぐ下のレートと常に比較されます。

CPRI には必須の物理レイヤプロトコルがありませんが、使用するプロトコルが 1×10^{-12} の BER 要件を満たしている必要があります (OBSAI ほど厳しくありません)。クロックの安定性と位相ノイズ要件も規定されます。

CPRI では、HV (高電圧) と LV (低電圧) という 2 つの電氣的バリエーションも推奨されています。HV は IEEE 802.3-2002 第 39 節の 1000Base-CX 仕様にガイドがあり、インピーダンスは 100Ω です。LV は XAUI にガイドがあります。LV は全てのレートで推奨され、このデバイスの対象となります。

CPRI 及び OBSAI 仕様を取り扱うときは、次の 2 つのリンクレイヤ要件をよく理解することが重要です。

- ・ リンク遅延精度とケーブル遅延調整
- ・ スタートアップの同期

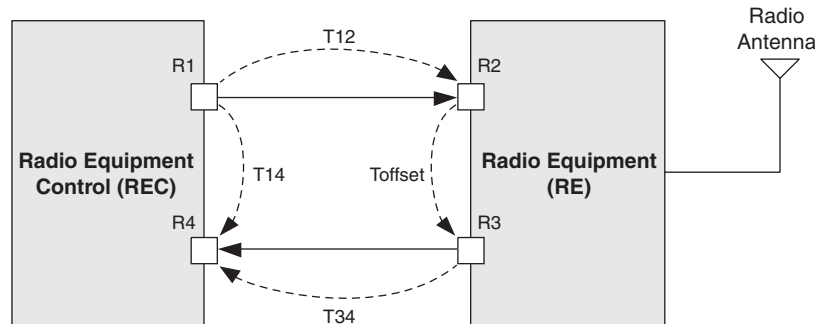
リンク遅延精度とケーブル遅延調整

以下の説明は CPRI 要件を大幅に利用したのですが、OBSAI の実装にも同じ要件が適用されます。

RE または RRH は REC または BTS に周波数ロックされます。そのため、この同期システムでは RRH と BTS との間の全ての遅延を調整し、エアインターフェイスのタイミング要件に適合する必要があります。インターフェイスには、リンクのケーブル遅延と、シングルホップまたはマルチホップ接続のラウンドトリップ遅延を調整できるようにする基本メカニズムが必要です。特に、遅延調整の参照点と、RE (Radio Equipment) に

おける入力信号と出力信号との関係が定義されます。全ての定義と要件は、図 8-33 に示すシングルホップ・シナリオの REC マスタポートと RE スレーブポートの間のリンクに対して記述されます。

図 8-33. REC マスタポートと RE スレーブポート間のリンク (シングルホップ・シナリオ)



参照点 R1-4 は、REC の出力ポイント (R1) と入力ポイント (R4)、及び RE の入力ポイント (R2) と出力ポイント (R3) に対応し、その論理接続を終端します。参考のため、アンテナは Ra で示してあります。

- ・ T12 は REC の出力ポイント (R1) から RE の入力ポイント (R2) までのダウンリンク信号の遅延、特にダウンリンクケーブルの遅延です。
- ・ T34 は RE の出力ポイント (R3) から REC の入力ポイント (R4) までのアップリンク信号の遅延、特にアップリンクケーブルの遅延です。
- ・ Toffset は R2 の入力信号と R3 の出力信号の間のフレームオフセットです。
- ・ T14 は R1 の出力信号と R4 の入力信号の間のフレームタイミング差 (ラウンドトリップ遅延 RTT) です。

遅延の測定はフレームタイミングを使用して行われます。CPRI には、UMTS 無線フレーム番号または BFN (Node B Frame Number) に基づく 10ms フレームがあります。各 UMTS 無線フレームには 150 ハイパーフレーム (各ハイパーフレームは 66.67us) と、それに対応するハイパーフレーム番号 (HFN = $0 \leq Z \leq 149$) があります。各ハイパーフレームには 256 ($0 \leq W \leq 255$) の基本フレームがあります (各基本フレームは 260.42ns = Tchip または Tc)。

RE によって RE の出力信号 (アップリンク) のフレームタイミングが決まり、それが RE の入力信号 (ダウンリンク) のフレームタイミングに対する固定オフセット (Toffset) になります。Toffset は、0 以上で 256 Tc 未満の任意の値です (ハイパーフレームを超えることはできません)。RE ごとに Toffset の値が異なります。REC は各 RE の Toffset 値を事前に認識します (事前設定された値、または上位レイヤのメッセージで RE が REC に通知)。

T14 を判定するため、REC から RE へのダウンリンク BFN 及び HFN が、RE から REC へのアップリンクで返されます。アップリンクでエラー状態が発生した場合、REC はアップリンク BFN 及び HFN を無効と見なします。その結果、 $T14 = T12 + Toffset + T34$ となります。

前述のとおり、システムは同期動作します。また、ハイパーフレームは固定長で、RRH-BTS 内部接続 (ケーブル長) は両方向で等しく ($T12 = T34$ で、両方の光ファイバを 1 つにバンドル)、内部接続遅延は $(T14 - Toffset)/2$ に変化すると見なされます。T14 の判定方法については既に説明しました。そのため、遅延調整に影響を与える主要な成分は Toffset です。したがって、リンクの両端で測定したハイパーフレームの受信時間と送信時間の差が内部接続遅延になります。

遅延調整要件は、3GPP 及び UTRAN の要件、特に 1 リンクのケーブル遅延のラウンドトリップ遅延測定の精度が $\pm Tc/16$ と規定された、CPRI 仕様の要件 R-21 (CPRI v3.0 20 ページ) で決まります。また、要件 R-20 では、伝送媒体 (ケーブル長以外) のラウンドトリップ・グループ遅延を除いて、インターフェイスのラウンドトリップ時間の絶対精度が同程度の要件 (T14 では $\pm Tc/16$) に適合すべきことが規定されています。前

述の説明を考慮すると、ケーブル長を除く、REC マスタポートと RE スレーブポートとの間のダウンリンクにおける絶対リンク遅延精度は、上記要件の半分 ($\pm T_c/32$ または約 8ns (8.138ns)) となります。その結果、T14 と Toffset にはどちらも絶対精度 ± 8 ns が必要です。

次に、各種レートにおいて不確実ビットの数がいくつまで許容されるかを決定することが重要です。基本的には、様々な CPRI 及び OBSAI ビットレートと 8.138ns を乗算することで、不確定 / 不一致と見なすビット数を決定できます。このことによる影響は、後述する SERDES シリアル / パラレル・データパスの説明の中で明確になります。

多くの SERDES は、シリアル化及びパラレル化で発生する一定レベルの不確実性を持っています。したがって、16 ビットバス・アーキテクチャの SERDES は、8 ビット・アーキテクチャの SERDES よりもワードあたりのビット数が 2 倍あるため、遅延の不確実性が 2 倍になります。

TX と RX のそれぞれの遅延を表 8-23 に示します。この表には、遅延のばらつきも示してあります。このばらつきは、既に説明した絶対遅延精度に直接影響を与えます。このばらつきは、TX FPGA ブリッジ FIFO、RX FPGA ブリッジ FIFO、RX クロックトレランス補償 FIFO の 3 つによるものです。CPRI システムは同期システムであるため、RX CTC FIFO がバイパスされ、RX 再生クロックが使用されます。

遅延のばらつきの原因として残るのは FPGA ブリッジ FIFO です。FPGA へのインターフェイスが 8 ビットバス・モードの場合は、この FIFO をバイパスできます。16 ビットインターフェイス・モードでは、FIFO で 2:1 ギアリングが行われるため、FPGA ブリッジ FIFO をバイパスできません。

SONET/SDH

SONET (Synchronous Optical Networking) 及び SDH (Synchronous Digital Hierarchy) は、光ファイバまたは電気インターフェイスを通じてデータを転送する、標準化された多重化プロトコルです。SONET の一般的基準は、“Telcordia Technologies Generic Requirements” ドキュメント (GR-253-CORE) に詳しく記載されています。SONET 及びその他の伝送システム (非同期光ファイバシステムやデジタル無線システム) に適用される一般的基準は、Telcordia の GR-499-CORE に記載されています。SONET と SDH は元来、様々な提供元による回線モード通信 (T1、T3 など) の伝送用に設計されました。SONET 以前は、こうした各種回線の同期ソースが異なることが最大の障害になっていました。すなわち、各回線はそれぞれ少し異なるレートと位相で動作していました。SONET では、送信元が異なる多数の異なる回線を、単一のフレーム化プロトコルで同時伝送できます。

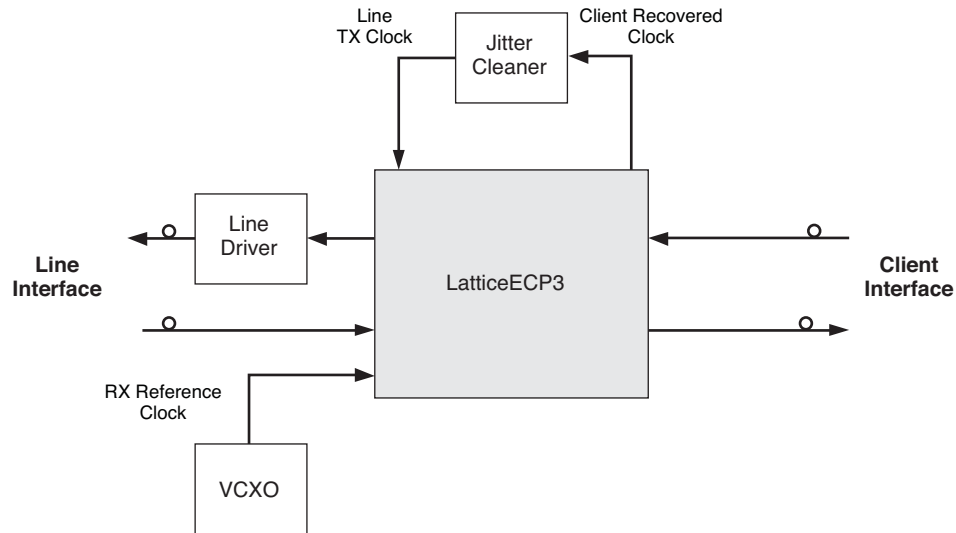
LatticeECP3 SERDES/PCS が提供するトランシーバは、STS-3/STM-1 (155.52 Mbps)、STS-12/STM-4 (622.08 Mbps)、STS-48/STM-16 (2.488 Gbps) という 3 つの SONET/SDH データレートに対応しています。SONET/SDH 用途では 8 ビット SERDES モードが使用されます。

SONET/SDH 回線に準拠するには、LatticeECP3 に外付け部品が必要です。SERDES の出力には外付けラインドライバが必要です。入力データストリームから高周波ジッタを排除するには、受信したクロックにジッタクリーナを適用してから、そのクロックを送信基準クロックとして使用します。

チップ間またはバックプレーン用途では、外付けラインドライバとクロックジッタ・クリーナが不要です。

図 8-34 に外付け部品を使用した回線側ソリューションを示します。

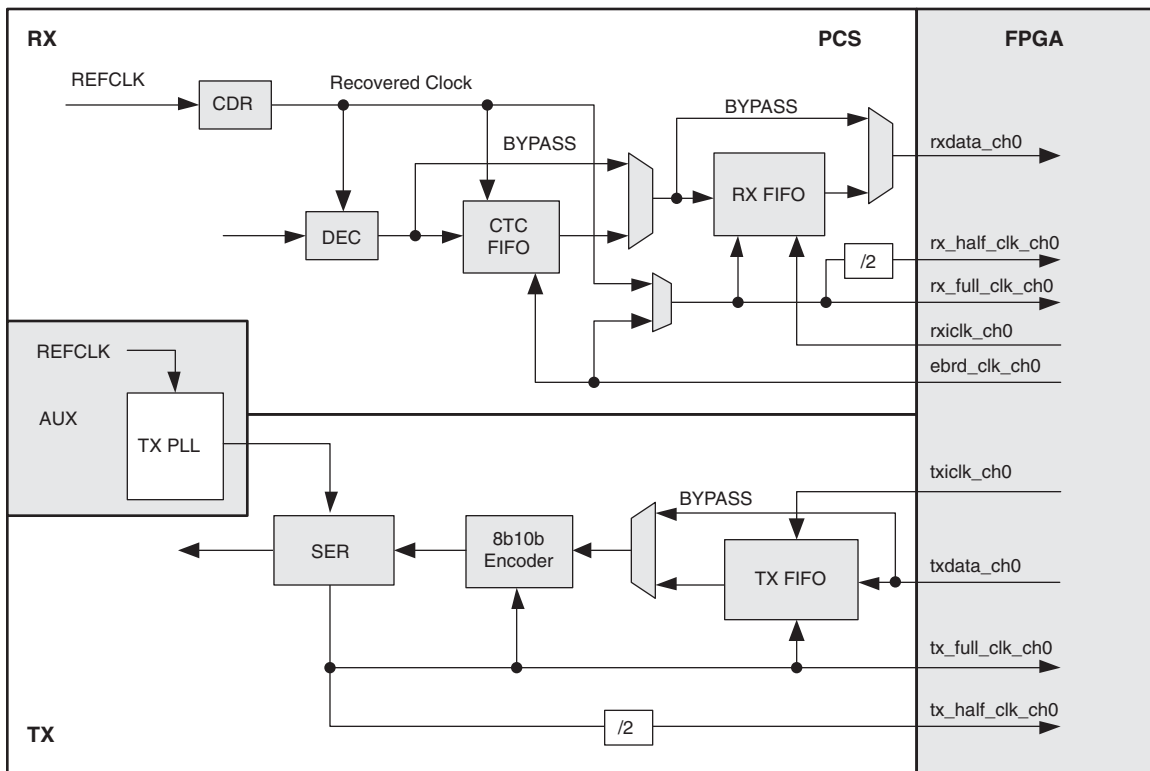
図 8-34. SONET/SDH 回線インターフェイス



FPGA インターフェイス・クロック

図 8-35 に PCS コアの後段ステージと FPGA ブリッジ、及び PCS と FPGA の境界を通過する主要クロックの概念図を示します。

図 8-35. クロック・インターフェイスの概念図



上記の図、及びこのセクションで後述するクロック図では、末尾の "i" がインデックス [3:0] (チャンネルごとに 1 つ) を表しています。

レジスタビットへのライトによって、クロックを多重化するセレクタのいずれかが変化した場合、その多重化クロックでクロックされるロジックを、ソフトウェア・エージェントがリセットするという要件があります。

PCS は 16 本のクロックを出力します。チャンネルごとにそれぞれ 2 本の送信クロックと 2 本の受信クロックがあります。2 本の送信クロックはフルレートとハーフレートのクロックで、全てが TX PLL から得られます。受信チャンネルごとにもフル及びハーフクロックの 2 本があります。16 本の全クロックが、必要に応じて FPGA のローカル（セカンダリ）またはグローバル（プライマリ）クロックとして使用されます。tx_half_clks はギアリングが 2:1 モードのときに使用されます。表 8-5 で記述されているとおり、tx_full_clk_ch0 と tx_half_clk_ch0 はプライマリ・クロックネットを直接ドライブできます。他チャンネルのクロックも全てプライマリ・クロックネットを直接ドライブできますが、一般配線を経由します。全ての tx_full_clk_ch[3:0] と tx_half_clk_ch[3:0] は USE_SECONDARY 制約を用いることでセカンダリ・クロックネットをドライブできますが、やはり一般配線を経由します。

送信クロックは TX FIFO（または状況に応じて位相シフト FIFO）のライトポートで使用されます。2 つの受信クロックの 1 つは RX FIFO のリードクロックに接続されます。その他のクロックは CTC FIFO のリードポートで使用され、RX FIFO のライトポートで使用される可能性があります（状況によります）。CTC 及び TX FIFO がバイパスされるかどうか、及び PCS が 8 ビット /10 ビット・インターフェイスモードか、16 ビット /20 ビット・インターフェイスモードかによって、4 つの使い方があります。アクティブパスは実線で強調してあります。必要なクロックツリーの数と種類も示してあります。モードによっては、ユーザが好んでよく使用するモードがあります。

このセクションでは、サポートする 6 つのケースの動作について説明します。ケースの概要を表 8-21 に示します。

表 8-21. クワッドと FPGA の間の 6 つのインターフェイス・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 ²	8/10 ビット	Yes	Yes	Yes
Case I-b ²	8/10 ビット	バイパス	Yes	Yes
Case I-c ²	8/10 ビット	Yes	バイパス	バイパス
Case I-d ²	8/10 ビット	バイパス	バイパス	バイパス
Case II-a ^{1,2}	16/20 ビット	Yes	Yes	Yes
Case II-b ^{1,2}	16/20 ビット	バイパス	Yes	Yes

- 16/20 ビット・データパス幅で使用了場合は、TX 位相シフト（アップサンプル）FIFO と RX 位相シフト FIFO（ダウンサンプル）が常に使用されます。これらをバイパスすることはできません。RX と TX の両方の FPGA インターフェイスのデータパス幅が、同時に同じになる必要はありません。個別に制御可能です。簡潔にするため、同じ使い方の中ではまとめて示してあります。
- TX 位相シフト（アップサンプル）FIFO と RX 位相シフト FIFO（ダウンサンプル）を互いにバイパスする必要はありません。個別に制御可能です。こちらも簡潔にするため、同じ使い方の中ではここに示してあります。

2:1 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には、SERDES ラインレートが 2.5Gbps よりも高い場合に、16/20 ビット幅インターフェイスの使用を推奨します。このインターフェイスでは、FPGA インターフェイス・クロックがバイトクロック周波数の半分で動作します。

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全ての SERDES ラインレートで使用可能ですが、SERDES のラインレートが十分に低い（2.5Gbps 以下）場合は、FPGA コア内の IP の実装効率が非常に高まるため、8/10 ビット幅インターフェイスが適しています。

6 つのインターフェイス・ケースの決定マトリクスを表 8-22 に示します。

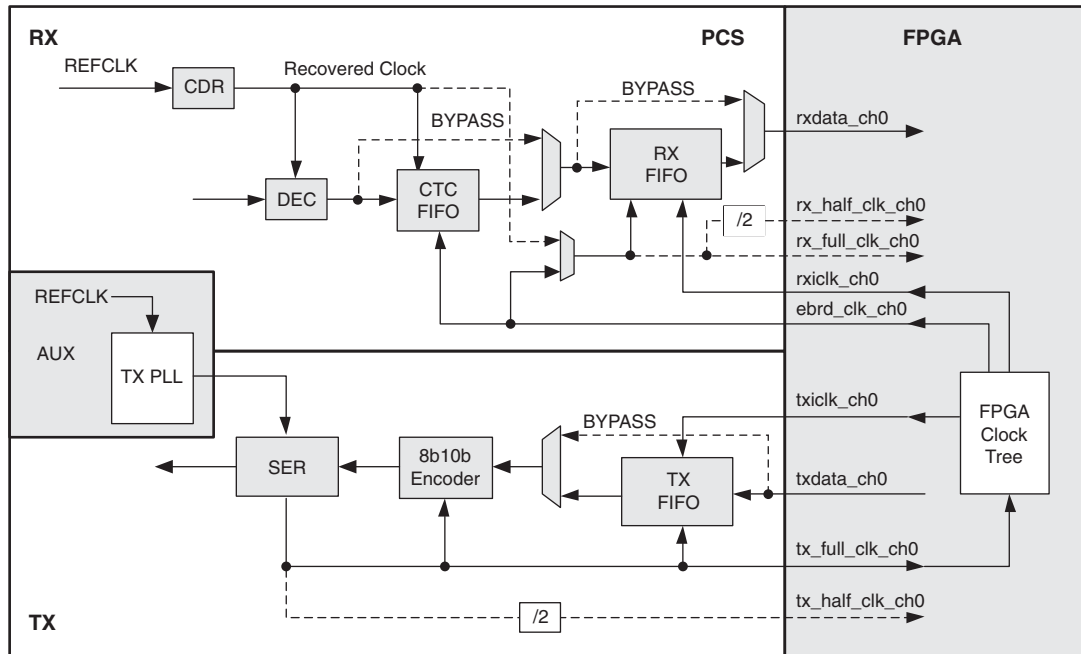
表 8-22. 6 つのインターフェイス・ケースの決定マトリクス

SERDES ラインレート	データパス幅	マルチチャネル・ アライメントの 必要性?	CTC の必要性?	RX FIFO の必要性?	インターフェイス・ ケース
2.5 Gbps 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	Yes	Yes	Case I_a ¹
				No	Case I_c ¹
			No	Yes	Case I_b ²
				No	Case I_d ²
		あり、マルチチャ ネル・リンク	バイパスが必要。 利用不可	Yes	Case I_b ³
				No	Case I_d ³
3.2 Gbps 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	Yes	Yes	Case II_a ⁴
			No	Yes	Case II_b ⁵
		あり、マルチチャ ネル・リンク	バイパスが必要。 利用不可	Yes	Case II_b ⁶

- このケースは、クワッド内でクロックトレランス補償が必要となる、ラインレート 2.5Gbps 以下 (8/10 ビット幅インターフェイス) のシングルチャネル・リンクを目的としています。CTC が必要となるのは、リンクの両端が互いに $\pm 300\text{ppm}$ 以内の異なる基準クロックソースを持つ場合です。ケース I_a は、コア内の IP に RX 位相シフト FIFO が必要な場合に使用されます。ケース I_b は、IP にこの FIFO が不要な場合に使用されます。
- このケースは、クワッド内でクロックトレランス補償が不要な、ラインレート 2.5Gbps 以下 (8/10 ビット幅インターフェイス) のシングルチャネル・リンクを目的としています。リンクの両端が同一の基準クロックソースに接続されている場合は CTC が不要です。同一回路基板上のチップ間リンクに多いケースです。基準クロックの差が完全に 0ppm であるため、CTC は不要でバイパスできます。この機能がコア内の IP で実行される場合も、クワッド内に CTC が不要です。
- このケースは、ラインレートが 2.5Gbps 以下 (8/10 ビット幅インターフェイス) のマルチチャネル・リンクを目的としたものです。FPGA デザイン内でマルチチャネル・アライメントを行う必要があります。マルチチャネル・アライメントは CTC よりも前に行われるため、マルチチャネル・アライメントが必要な場合はクワッド内の CTC FIFO をバイパスし、マルチチャネル・アライメントと CTC (必要な場合) の両方が FPGA デザインで行われる必要があります。
- このケースは、クワッドと FPGA コアの間 (16/20 ビット幅インターフェイス) に 2:1 ギアボックスが必要な、ラインレート 3.2Gbps 以下のシングルチャネル・リンクを目的としたものです。クロックトレランス補償はクワッドに含まれています。CTC が必要となるのは、リンクの両端が互いに $\pm 300\text{ppm}$ 以内の異なる基準クロックソースを持つ場合です。
- このケースは、クワッドと FPGA コアの間 (16/20 ビット幅インターフェイス) に 2:1 ギアボックスが必要な、ラインレート 3.2Gbps 以下のシングルチャネル・リンクを目的としたものです。クロックトレランス補償はクワッドに含まれていません。リンクの両端が同一の基準クロックソースに接続されている場合は CTC が不要です。同一回路基板上のチップ間リンクに多いケースです。基準クロックの差が完全に 0ppm であるため、CTC は不要でバイパスできます。この機能が FPGA デザインで実行される場合も、クワッド内に CTC が不要です。
- このケースは、クワッドと FPGA コアの間 (16/20 ビット幅インターフェイス) に 2:1 ギアボックスが必要な、ラインレート 3.2Gbps 以下のマルチチャネル・リンクを目的としたものです。FPGA デザインでマルチチャネル・アライメントを行う必要があります。マルチチャネル・アライメントは CTC よりも前に行われるため、マルチチャネル・アライメントが必要な場合はクワッド内の CTC FIFO をバイパスし、マルチチャネル・アライメントと CTC (必要な場合) の両方が FPGA デザインで行われる必要があります。

ケース I_a : 8/10 ビット、CTC FIFO 及び RX/TX FIFO バイパスせず

図 8-36. 8/10 ビット、CTC FIFO 及び RX/TX FIFO バイパスせず



1. この場合 TX FIFO は位相シフト FIFO としてのみ機能します。
2. この場合 RX FIFO は位相シフト FIFO としてのみ機能します。
3. TX PLL からのクワッドレベルのフルレート・クロック (tx_full_clk) が、FPGA センタークロック・マルチプレクサに直接アクセスできます。これは比較的高性能なパスです。センタークロック・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ーフェイス・ロジックのクロックに使用されます。クロックツリーの一部のリーフノードは、FPGA 送信入力クロック (txicl)、チャンネルごとの CTC FIFO リードクロック (ebrd_clk)、FPGA 受信入力クロック (rxicl) に接続されます。このケースは、シングルチャンネルの最も一般的な使い方と考えられます。

FPGA ロジック内のクロック及びデータ信号インターフェイスの例 (ケース I_a)

トップモジュール内にインスタンス化された SERDES/PCS モジュールの一部を次に示します。クロック及びデータポートが Verilog にどのようにマップされるかを表しています。

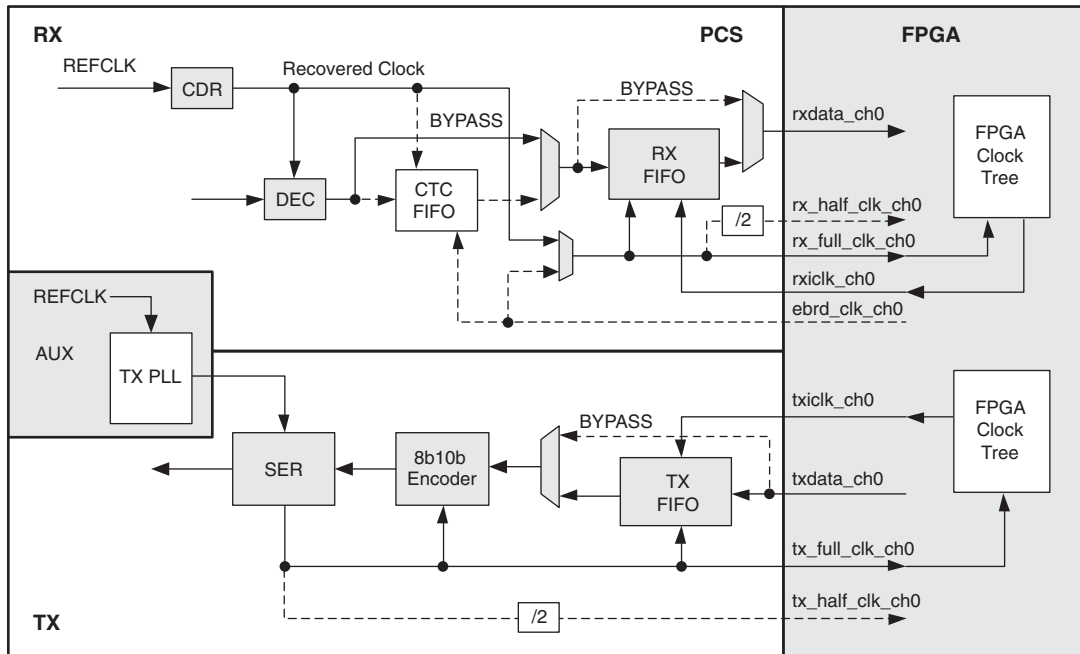
```
.txicl_ch0(txclk),
.rxicl_ch0(txclk),
.rx_full_clk_ch0(),
.tx_full_clk_ch0(txclk),
.tx_half_clk_ch0(),
.txdata_ch0(txdata_2_pcs),
.rxdata_ch0(rxdata_from_pcs),
.tx_k_ch0(txkcntl_2_pcs),
.rx_k_ch0(rxkcntl_from_pcs),
```

ebrd_clk_ch0 はケースに応じてソフトウェアによって自動的に配線されます。

図 8-36 に示すとおり、tx_full_clk_ch0 は 'txclk' という名前のワイヤを使用し、txi_clk_ch0 と rxi_clk_ch0 の両方を送出します。

ケース I_b : 8/10 ビット、CTC FIFO バイパス

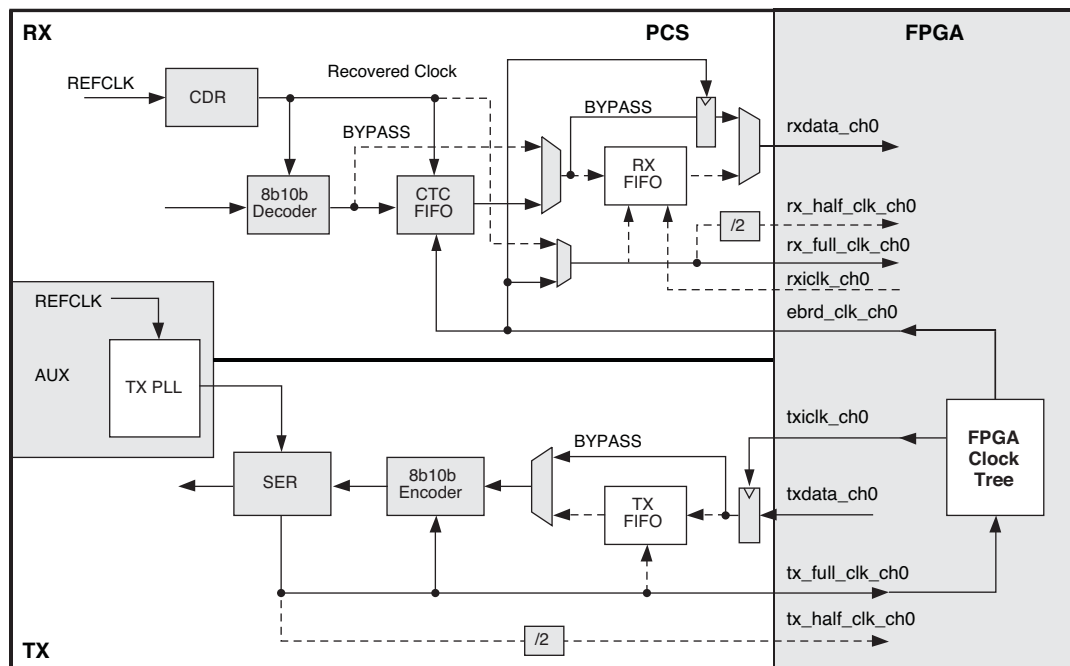
図 8-37. 8b/10 ビット、CTC FIFO バイパス



1. この場合 TX FIFO は位相シフト FIFO としてのみ機能します。
2. この場合 RX FIFO は位相シフト FIFO としてのみ機能します。
3. TX FPGA チャンネル入力クロックは前のケースと同様にクロックされ、フルレート送信 FPGA 出力クロックから FPGA センタークロック・マルチプレクサへの直接接続によってドライブされるクロックツリーを使用します。CTC FIFO がバイパスされると、再生クロックで RX 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クロックによる、個別のローカルまたはグローバル・クロックツリーのドライブが必要になることがあります（クワッドごとに最大4つのローカルまたはグローバル・クロックツリー）。このクロックツリーでは FPGA 受信クロック入力がドライブされ、RX FIFO のリードポートが制御されます。このケースで CTC FIFO をバイパスする最大の理由は、FPGA コアでマルチチャンネル・アライメントを行うためです。エラスティック・バッファを使用した CTC が、FPGA コア内で行われることになります。CTC FIFO は、再生クロックまたはマスタ再生クロックによってライトできます。CTC FIFO のリードは、TX クロックを使用して TX クロックツリーから行われます。

ケース L_c : 8/10 ビット、RX/TX FIFO バイパス

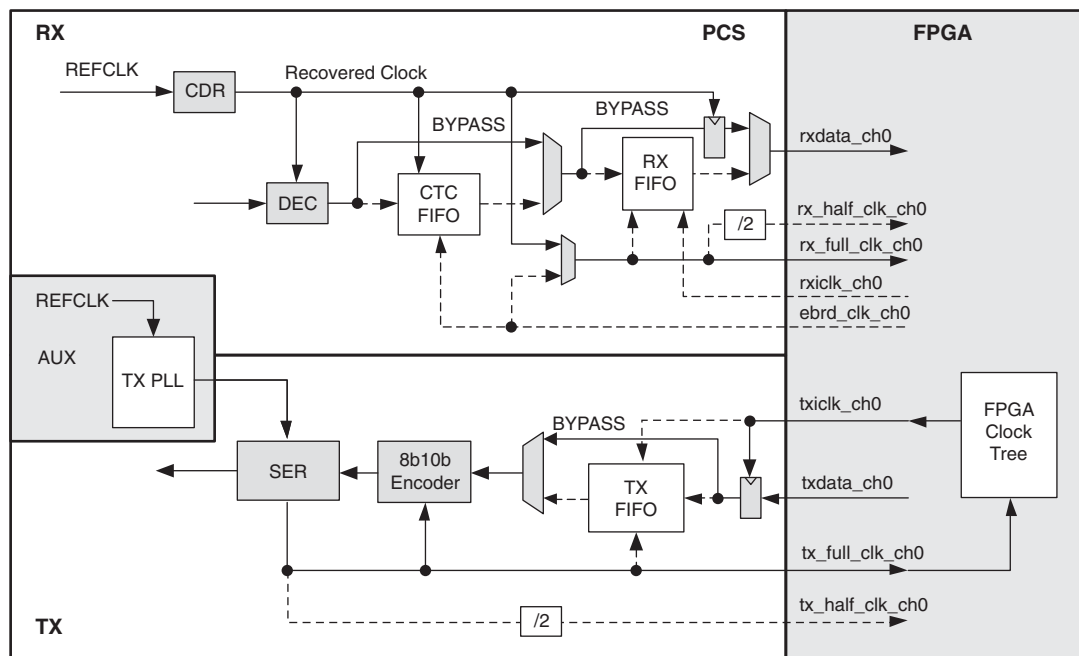
図 8-38. 8/10 ビット、RX/TX FIFO バイパス



1. TX チャンネルは前の 2 つのケースと同様にクロックされます。RX チャンネルでは FPGA 入力クロックが ebrd_clki となります。このクロックは FPGA TX クロックツリーでドライブされます。このケースでは、ebrd_clki がソフトウェアによって自動的に配線されます。

ケース L_d : 8/10 ビット、CTC FIFO 及び RX/TX FIFO バイパス

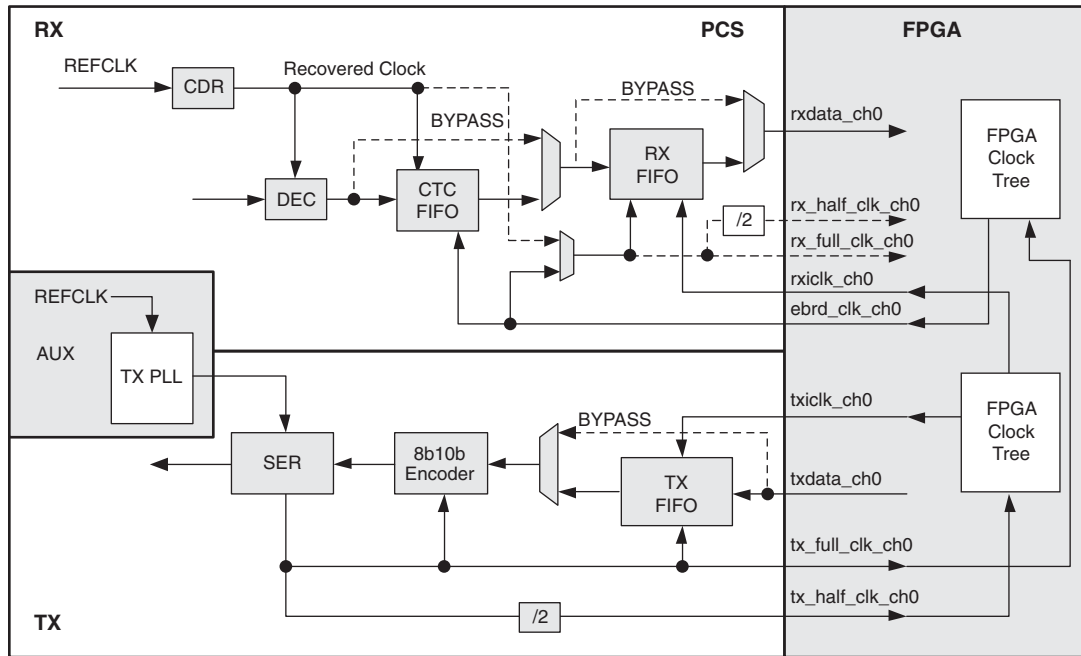
図 8-39. 8b/10 ビット、CTC FIFO 及び RX/TX FIFO バイパス



1. FPGA クロックツリーは、このケースではクロックドメインと考える交換可能です。TX チャンネルは前の 3 つのケースと同様にクロックされます。RX チャンネルでは、再生チャンネル RX クロックが FPGA に送出されます。このケースはビデオ用途への対応に便利です。

ケース II_a : 16/20 ビット、CTC FIFO 及び RX/TX FIFO バイパスせず

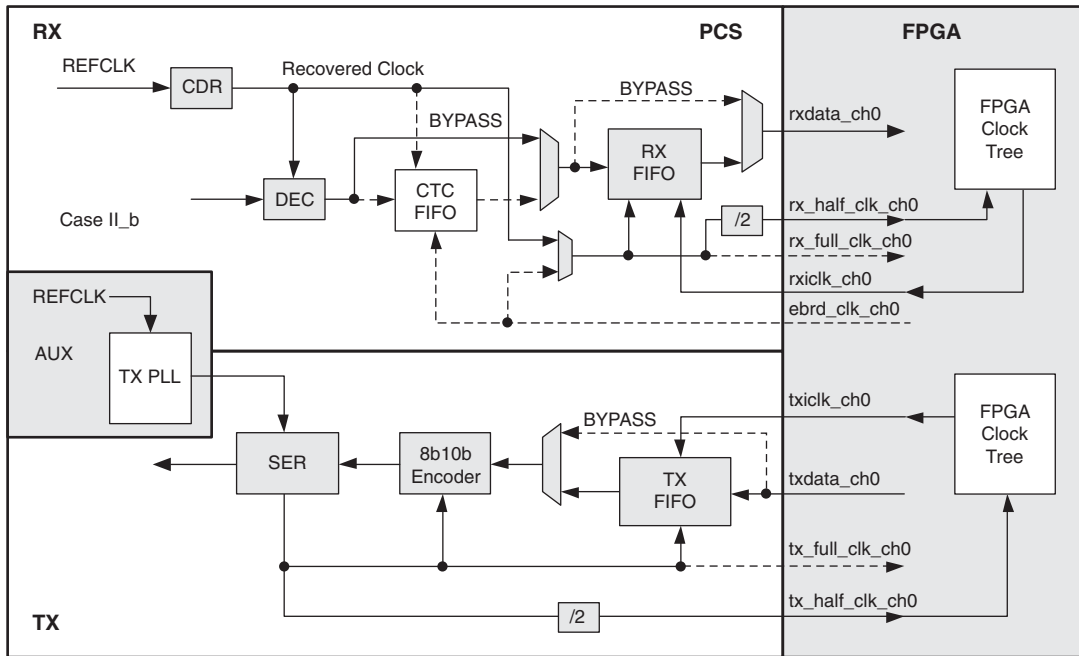
図 8-40. 16/20 ビット、CTC FIFO 及び RX/TX FIFO バイパスせず



1. この場合 TX FIFO は位相シフト FIFO とアップサンプル FIFO の両方として機能します。
2. この場合 RX FIFO は位相シフト FIFO とダウンサンプル FIFO の両方として機能します。
3. これは、FPGA でフルバイト周波数を維持できない場合に非常によくあるシングルチャネルの使い方です。2つ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 クロックセンター・マルチプレクサへの送信フルレート・クロックと送信ハーフレート・クロックの直接アクセスでドライブされます。フルレート・クロックツリーは、CTC FIFO リードポートと RX FIFO 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、RX 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、CTC FIFO バイパス

図 8-41. 16/20 ビット、CTC FIFO バイパス



1. この場合 TX FIFO は位相シフト FIFO とアップサンプル FIFO の両方として機能します。
2. この場合 RX FIFO は位相シフト FIFO とダウンサンプル FIFO の両方として機能します。
3. これは、FPGA でフルバイト周波数を維持できない場合に非常によくあるマルチチャネル・アライメントの使い方です。受信クロックツリー（最大 4 つ）はローカルとグローバルのどちらでもかまいません。ハーフレート・クロックで動作します。送信クロックツリーは、FPGA クロックセンター・マルチプレクサへの送信ハーフレート・クロックの直接アクセスでドライブされます。

SERDES/PCS ブロックの処理遅延

表 8-23 にトランスミッタ及びレシーバ内の各機能ブロックの処理遅延を示します。処理遅延はパラレル・クロックサイクルで与えられます。図 8-42 は各ブロックの位置を示しています。

表 8-23. SERDES/PCS 処理遅延の内訳

項番	記 述	Min.	平均 .	Max.	固定	バイパス	単位
送信データの処理遅延¹							
T1	FPGA ブリッジ ~ 異なるクロックによる 1:1 ギアリング	1	3	5	—	1	word clk
	FPGA ブリッジ ~ 同一クロックによる 1:1 ギアリング	—	—	—	3	1	word clk
	FPGA ブリッジ ~ 2:1 ギアリング	1	3	5	—	—	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	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
T5	プリエンファシス ON	—	—	—	1 + $\Delta 2$	—	UI + ps
	プリエンファシス OFF	—	—	—	0 + $\Delta 3$	—	UI + ps
受信データの処理遅延²							
R1	イコライザ ON	—	—	—	$\Delta 1$	—	UI + ps
	イコライザ OFF	—	—	—	$\Delta 2$	—	UI + ps
R2	デシリアライザ : 8 ビットモード	—	—	—	10 + $\Delta 3$	—	UI + ps
	デシリアライザ : 10 ビットモード	—	—	—	12 + $\Delta 3$	—	UI + ps
R3	SERDES ブリッジ受信	—	—	—	2	1	word clk
R4	ワードアライメント ³	3.1	—	4	—	—	word clk
R5	8b10b デコーダ	—	—	—	1	1	word clk
R6	クロックトレランス補償	7	15	23	1	1	word clk
R7	FPGA ブリッジ ~ 異なるクロックによる 1:1 ギアリング	1	3	5	—	1	word clk
	FPGA ブリッジ ~ 同一クロックによる 1:1 ギアリング	—	—	—	3	1	word clk
	FPGA ブリッジ ~ 2:1 ギアリング	1	3	5	—	—	word clk

1. $\Delta 1 = -245\text{ps}$, $\Delta 2 = +88\text{ps}$, $\Delta 3 = +112\text{ps}$.

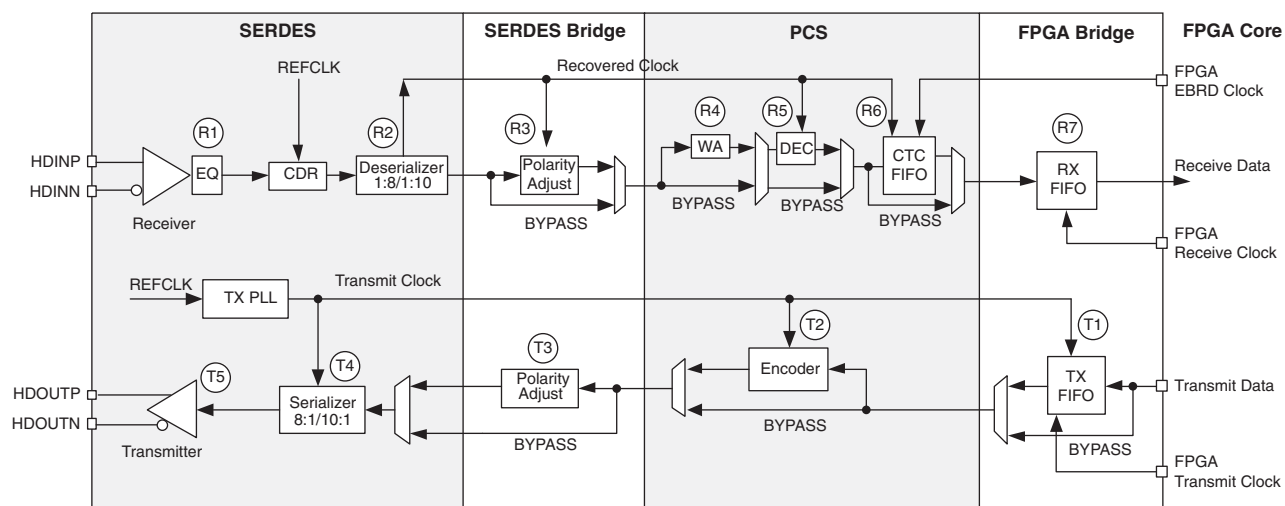
2. $\Delta 1 = +118\text{ps}$, $\Delta 2 = +132\text{ps}$, $\Delta 3 = +700\text{ps}$.

3. 表 8-24 にワードアライメント・オフセットごとのワードアライナ処理遅延を示します。実際のオフセットはチャネル・ステータスレジスタ CH_22, bit [3:0] で確認できます。

表 8-24. ワードアライナ処理遅延とオフセットの関係

wa_offset[3:0] (CH_22[3:0])	処理遅延 (ワードクロック数)
0	4.0
1	3.9
2	3.8
3	3.7
4	3.6
5	3.5
6	3.4
7	3.3
8	3.2
9	3.1

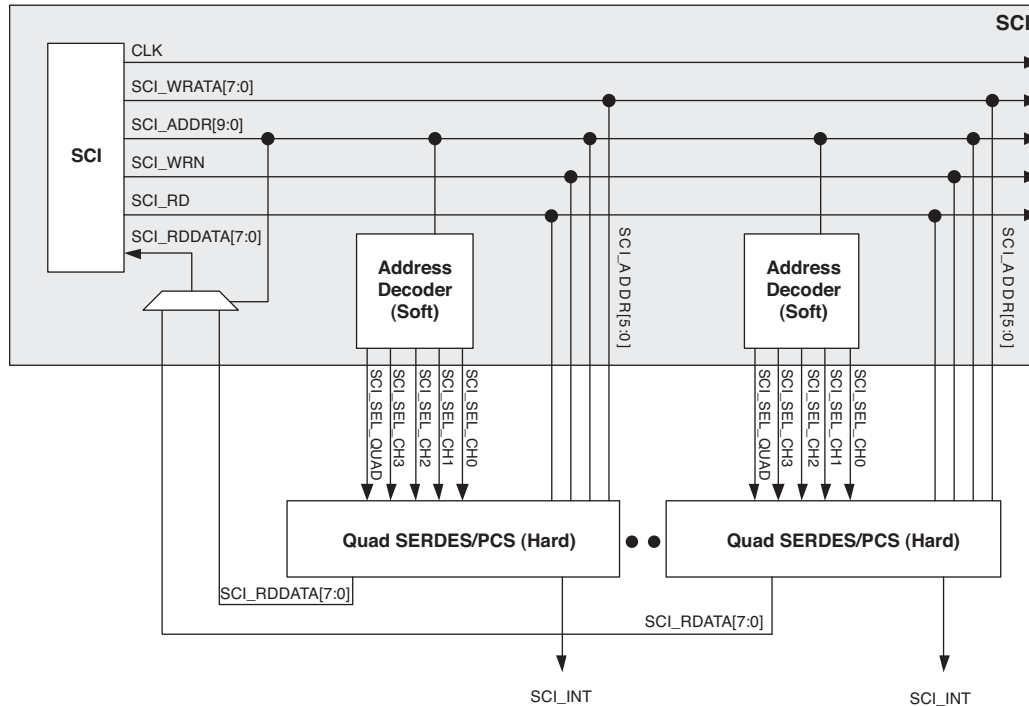
図 8-42. トランスミッタとレシーバの処理遅延ブロック図



SERDES クライアント・インターフェイス (SCI)

SCI では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 SERDES/PCS クラウドを制御できます。こ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。FPGA コア内にある SCI のブロック図を図 8-43 に示します。

図 8-43. SCI インターフェイスのブロック図



FPGA 内にあるインターフェイス・ロジックは、インターフェイス方式に合わせてユーザが開発する必要があります。コード例については、ラティスのテクニカルサポートにお問い合わせください。

SCI_ADDR バスはブロック内で 6 ビット幅になっています。ブロック境界でのバス幅は 11 ビットです。上位 5 ビットは、クワッドブロック選択及びチャンネル選択に使用されます。表 8-25 に、SERDES クワッドの SCI アドレスマップを示します。

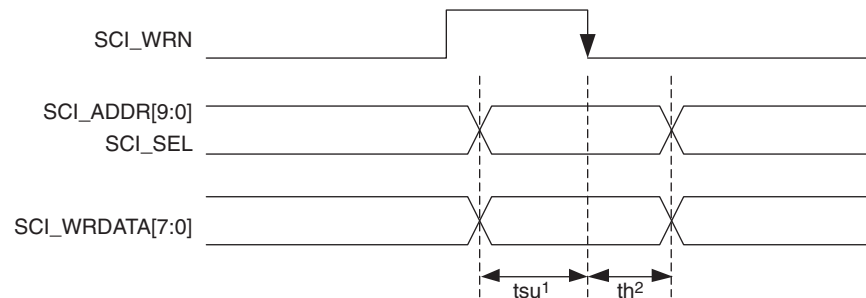
SERDES/PCS のレジスタアドレスとビットについては、付録 A 及び付録 B を参照してください。

表 8-25. SCI 最大 4 つの SERDES/PCS クラウド用の SCI アドレスマップ

アドレスビット	記 述
SCI_ADDR[5:0]	レジスタ・アドレスビット 000000 = レジスタ 0 選択 000001 = レジスタ 1 選択 ... 111110 = レジスタ 62 選択 111111 = レジスタ 63 選択
SCI_ADDR[8:6]	チャンネル・アドレスビット 000 = チャンネル 0 選択 001 = チャンネル 1 選択 010 = チャンネル 2 選択 011 = チャンネル 3 選択 100 = クラウド選択 101 = 未使用 110 = 未使用 111 = 未使用
SCI_ADDR[10:9]	クラウド・アドレスビット 00 = クラウド A 選択 01 = クラウド B 選択 10 = クラウド C 選択 11 = クラウド D 選択

このインターフェイスを通じたリード及びライト動作は非同期で行われます。ライトサイクルではライトデータとライトアドレスのセットアップが必要で、SCI_WR の立ち下がりエッジで保持されます。リードサイクルでは、SCI_RD パルスでタイミングをとる必要があります。図 8-44 と 8-45 に、ライト及びリードサイクルをそれぞれ示します。

図 8-44. SCI WRITE サイクル、重要なタイミング



1. tsu is the setup time for address and write data prior to the falling edge of the write strobe.
2. th is the hold time for address and write data after the falling edge of the write strobe.

Note: To avoid accidental writing to control registers, registers should be used at the SCI input ports to drive them low at power-up reset.

図 8-45. SCI READ サイクル、重要なタイミング

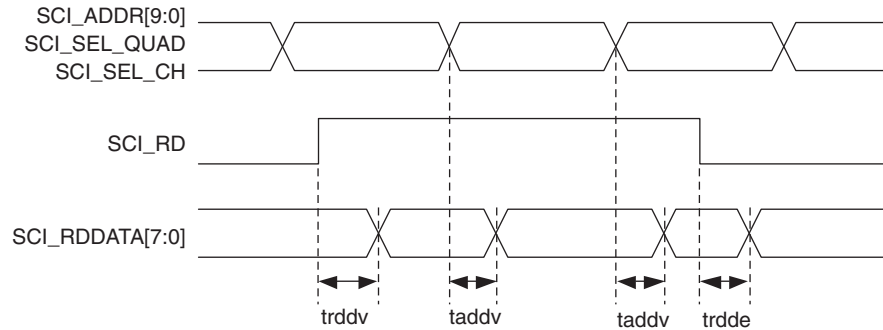


表 8-26. タイミングパラメータ

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/ライトのようにシンプルです。擬似コードの例を示します。

ライト :

- ・ Cycle 1: Set sci_addr[5:0], sciw_data[7:0], sci_sel = 1'b1
- ・ Cycle 2: Set sci_wrn from 0 ≥ 1
- ・ Cycle 3: Set sci_wrn from 1 ≥ 0, sci_sel = 1'b0

リード :

- ・ Cycle 1: Set sci_addr[5:0], sci_sel = 1'b1
- ・ Cycle 2: Set sci_rd from 0 ≥ 1
- ・ Cycle 3: Obtain reading data from sci_rddata[7:0]
- ・ Cycle 4: Set sci_rd from 1 ≥ 0

割り込みとステータス

ステータスビットは 1 バイト幅の SCI からリードできるため、8 つの割り込みステータス信号を同時にリードできます。割り込みイベントが発生すると、SCI_INT 信号が High になることで示されます。その場合は、割り込み元がクワッドなのか、いずれかのチャンネルなのかを示す QIF ステータスレジスタを、ユーザがリードする必要があります。このレジスタはリードしてもクリアされません。クリアされるのは、クワッドまたはチャンネルからの全ての割り込みソースがクリアされたときです。割り込みの集約ソースの判定後は、対応するクワッドまたはチャンネルのレジスタをリードし、割り込みのソースを判定します。表 8-27 と 8-28 に、割り込みの全てのソースを示します。

表 8-27. クワッドの割り込みソース

クワッド SCLINT ソース	記 述	レジスタ名
int_qd_out	クワッド割り込み。クワッド内のどこかで割り込みイベントが発生すると、このレジスタビットがアクティブになります。全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がクリアされます。	PCS クワッド・ステータスレジスタ QD_20
int_ch_out[0:3]	チャンネル割り込み。対応するチャンネル内のどこかで割り込みイベントが発生すると、このレジスタビットがアクティブになります。対応するチャンネル内の全ての割り込みソースがクリアされると、これらのレジスタビットがクリアされます。	PCS クワッド・ステータスレジスタ QD_20
ls_sync_statusn_[0:3]_int ls_sync_status_[0:3]_int	リンクステータス Low（同期外）チャンネル割り込み。リンクステータス High（同期中）チャンネル割り込み	PCS クワッド割り込みステータスレジスタ QD_22
~PLOL, PLOL	PLL ロックはずれで生成される割り込み。~PLOL および PLOL	SERDES クワッド・ステータスレジスタ QD_25

表 8-28. チャンネルの割り込みソース

チャンネル SCLINT ソース	記 述	レジスタ名
fb_tx_fifo_error_int fb_rx_fifo_error_int cc_overnrun_int cc_underrun_int	FPGA ブリッジ TX FIFO エラー割り込み FPGA ブリッジ RX FIFO エラー割り込み CTC FIFO オーバーラン及びアンダーラン割り込み	PCS チャンネル一般割り込みステータスレジスタ CH_23
pci_det_done_int rlos_lo_int ~rlos_lo_int rlo_int ~rlo_int	lpci_det_done 用に生成された割り込み rlos_lo 用に生成された割り込み ~rlos_lo 用に生成された割り込み rlo_int 用に生成された割り込み ~rlo_int 用に生成された割り込み	SERDES チャンネル割り込みステータスレジスタ CH_2A

SERDES クライアント・インターフェイスの用途例

ラティス ORCAstra FPGA コンフィグレーション・ソフトウェアは、ユーザが FPGA レジスタ内の制御ビットをプログラムすることで、ラティス FPGA の動作モードを設定可能な PC ベースの GUI です。

SERDES/PCS ステータス情報がリアルタイムに画面表示され、全てのコンフィグレーションを制御レジスタに保存し、さらなるテストが可能です。GUI を使用しても、FPGA コア部分のプログラミングの妨げにはなりません。ORCAstra の詳細とダウンロード可能なファイルについては、ラティスセミコンダクター社のウェブサイト（www.latticesemi.com/products/designsoftware/orcastra.cfm）を参照してください。

LatticeECP3 の ORCAstra インターフェイス設計は、全て IPexpress から行えます。ORCAstra インターフェイスが格納された HDL ファイル以外に、ORCAstra インターフェイスをラップする“chip.v”（Verilog 用）というファイルも用意されています。

SERDES/PCS クワッドの動的コンフィグレーション

SERDES/PCS クワッドは、オプションの SERDES クライアント・インターフェイスからアクセス可能なレジスタで制御できます。

コンフィグレーション・メモリセルで制御する場合は、SERDES/PCS クワッドがコンフィグレーション完了後に、それ以上のユーザによる介入なしに機能状態になる必要があります。そのため、SERDES/PCS クワッドの初期化に必要な特殊なリセット・シーケンスは、ハードウェアで自動的に処理する必要があります。すなわち、SCI の使用はオプションになります。SERDES/PCS クワッドは、FPGA コア内にソフト IP が存在することを前提としていません。

SERDES デバッグ機能

PCS ループバックモード

LatticeECP3 ファミリーには、PCS/FPGA インターフェイスの制御信号で制御する 3 つのループバックモードがあり、外部 SERDES/ ボード・インターフェイス及び内部 PCS/FPGA ロジック・インターフェイスをテストしやすくなっています。受信データを送信データパスにループする、2 つのループバックモードが用意されています。このループバックモードは、高速シリアル SERDES パッケージのピン接続、及び組み込まれた SERDES や PCS ロジックのチェックに便利です。

RX-to-TX シリアル・ループバックモード

CDRまたはデシリアライザを通過することなく、シリアル受信データを送信バッファにループバックします。IPexpress GUI で RX-to-TX シリアル・ループバック・オプションを選択すると、LB_CTL[1:0] が '10'、TDRV_DAT_SEL[1:0] レジスタビットが '11' に設定されます (表 8-81 及び 8-84 参照)。

TX-to-RX シリアル・ループバックモード

このモードでは、シリアル送信データがレシーバの CDR ブロックにループバックされます。IPexpress GUI で TX-to-RX シリアル・ループバック・オプションを選択すると、LB_CTL[1:0] が '01' に設定されます (表 8-81 参照)。

SERDES パラレル・ループバックモード

PCS ロジックを通過することなく、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。IPexpress GUI でディセーブルした場合は、パラレル・ループバックモードが FPGA コア制御信号 sb_felb_ch[3:0]_c と sb_felb_rst_ch[3:0]_c から動的に制御されます。

このループバックモードの動的機能を使用しない場合は、この 2 つの制御信号をグランドに接続する必要があります。

IPexpress GUI でこのループバックモードをイネーブルすると、制御レジスタビット sb_pffifo_lp(CH_03[5]) が設定され、ループバックモードが設定されます。FPGA コアからの制御信号 sb_felb_ch[3:0]_c と sb_felb_rst_ch[3:0]_c は、PCS モジュール内で利用できません。

以上の説明については、図 8-46 を参照してください。

図 8-46. 3 つのループバックモード

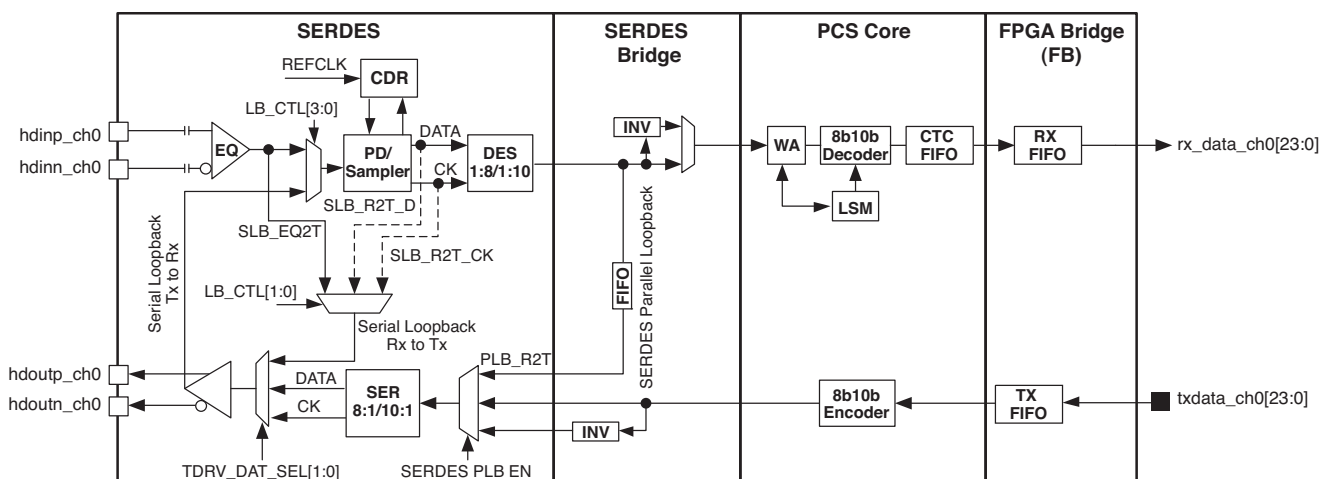
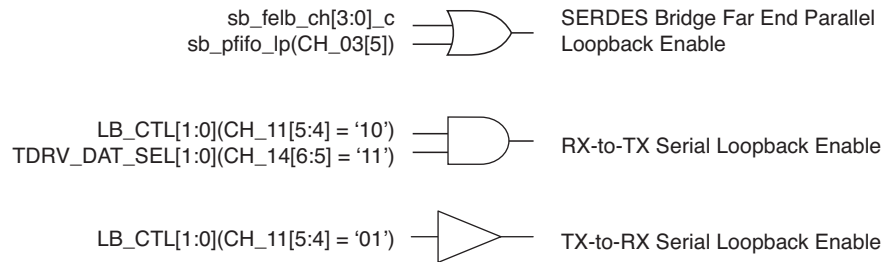


図 8-47. ループバックのイネーブル信号



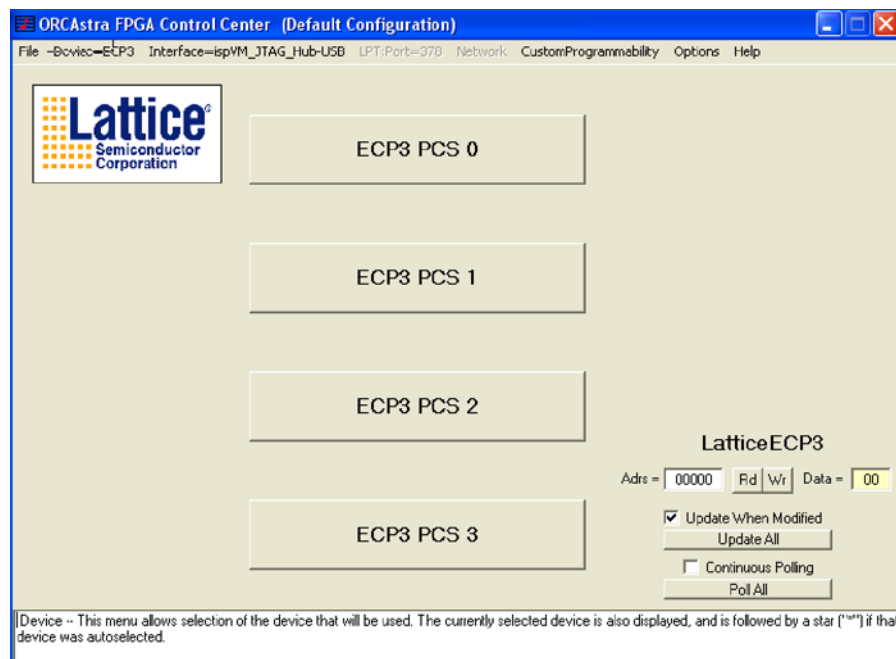
ORCAstra

ラティス ORCAstra ソフトウェアは、プロセスを大幅に再コンパイルしたり、基板に変更を加えることなく、コンフィグレーション・オプションを手軽に推敲するため役立ちます。GUI で作成されたコンフィグレーションは、メモリに保存し、後から再ロードして使用できます。ORCAstra を使用するには、IPexpress で ORCAstra モジュールを作成し、FPGA デザインで使用する必要があります。

マクロ機能もあり、スクリプトベースのコンフィグレーションとテストに対応できます。この GUI は、システムのステータス情報のリアルタイム表示にも使用できます。ORCAstra ソフトウェアを使用しても、FPGA のプログラミングが妨げられることはありません。

図 8-48 に、ORCAstra GUI のトップレベル・ウィンドウを示します。アドレスセルでデータをリード及びライトすることで、各 PCS チャンネル用のサブウィンドウに移動することなく、このウィンドウでリードとライトを行えます。起動すると、ORCAstra は自動的にデバイスタイプを認識します。あるいは、デバイス・プルダウンメニューでデバイスタイプを選択することもできます。

図 8-48. ORCAstra トップレベル・スクリーンショット



デフォルトでは図 8-48 に示すデータボックスがビッグエンディアン・バイトオーダ（最上位ビットが左）になっています。“Options”（オプション）タブにある“Display Data Reversed in Data Box”（データボックスのデータを反転表示）を選択し、リトルエンディアン・オーダにすることもできます。

“Interface=None”（インターフェイスなし）タブをクリックし、ドロップダウンリストから “1 ispVM JTAG Hub USB Interface”（1 ispVM JTAG ハブ USB インターフェイス）を選択します。

次に、“Select Target JTAG Device”（ターゲット JTAG デバイスの選択）ウィンドウで “C2 0A 80 80” を選択します。

図 8-49. JTAG デバイス選択



そして、ORCAstra のハブ I/O ウィンドウで “OK” をクリックします。

図 8-50. ハブ ID 選択

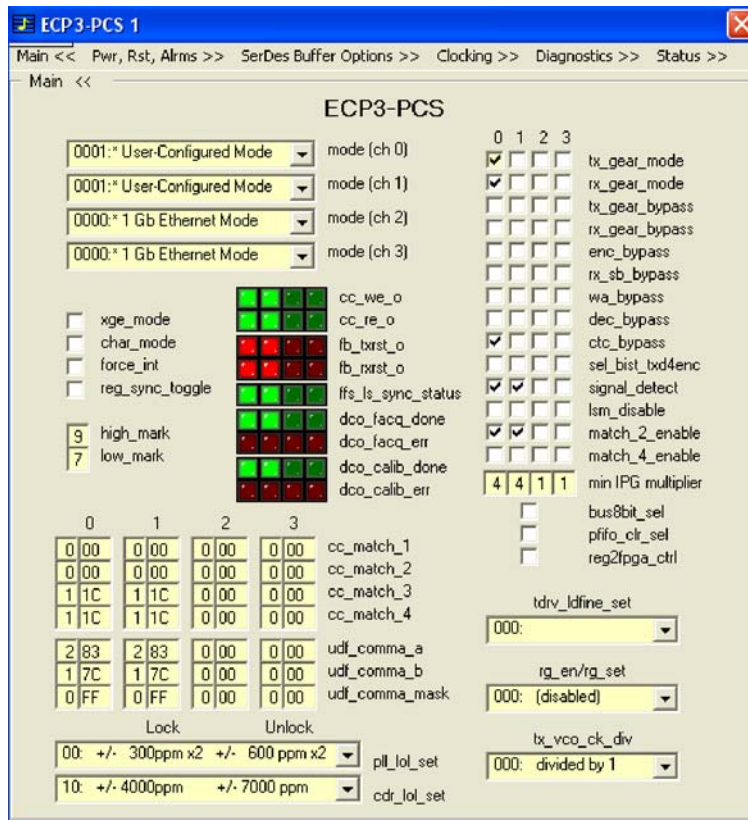


この図ではチャンネル 0 とチャンネル 1 の活動が表示されています。この例はデザイン内で PCS SCI アドレスが クアッド 0 にマップされていることを前提としています。

PCS0（クアッド 0）ボタンをダブルクリックすると、図 8-51 に示すメインウィンドウが表示されます。

この標準ウィンドウメニューでは、デバイスとインターフェイスの選択が制御されます。また、保存済みファイルによるコンフィグレーションのセットアップと保存など、様々なコンフィグレーション・オプションもあります。

図 8-51. ORCAstra メインウィンドウ

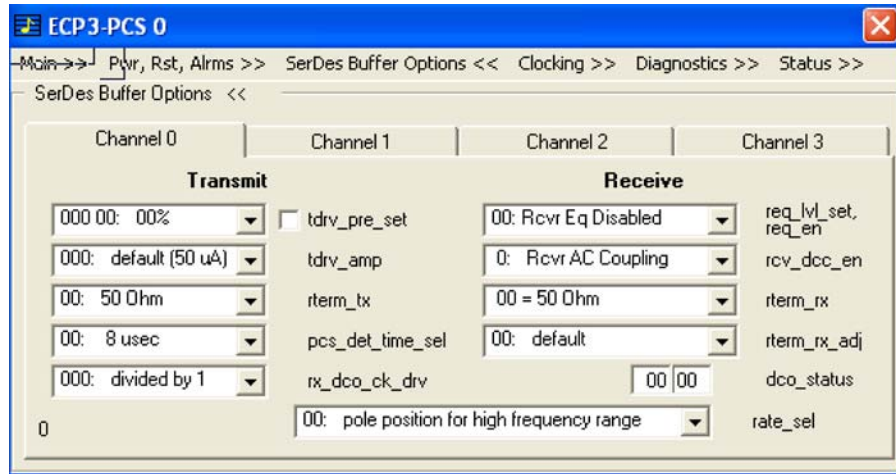


制御ボックスとボタン、ステータスボックス、及びテキストウィンドウ

カーソルを制御ボックスに合わせてマウスの左ボタンをクリックすると、制御ビットが設定されます。選択したボックスのビット位置と機能の両方がテキストウィンドウに表示され、LatticeECP3 ファミリー・データシートのレジスタマップ・テーブルと同じ表示になります。カーソルをビット名に合わせた場合は、機能のみが表示されます。ステータスボックスは制御ボックスと似ていますが、LED が表示され、背景に色が付きます。

図 8-52 に SERDES バッファオプション・ウィンドウを示します。コンフィグレーション・オプションは、プルダウンメニューから選択できます。

図 8-52. SERDES バッファオプション・ウィンドウ



ORCAstra の詳細とダウンロード可能なファイルについては、次のアドレスのラティスセミコンダクターのウェブサイトを参照してください：www.latticesemi.com/products/designsoftware/orcastra.cfm

その他の設計上の注意事項

SERDES/PCS のシミュレーション

表 8-29. シミュレーション・モデルのフォルダ

シミュレータ	モデルのあるフォルダ
Active-HDL	ispTOOLS\cae_library\simulation\blackbox\pcsc-aldec.zip
ModelSim	ispTOOLS\cae_library\simulation\blackbox\pcsd-mti_6.0-V1-1.zip
NC-Verilog	ispTOOLS\cae_library\simulation\blackbox\pcsd-ncv.zip
VCS	ispTOOLS\cae_library\simulation\blackbox\PCSD_sim.vp.zip

16/20 ビット・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できません。ワードアライナがイネーブルの場合、PCS はバイトアライメントのみを実行できます。16 ビット・ワードアライメントは FPGA ファブリックで行う必要があります、それが最も単純な方法です。シミュレーション・モデルも同様に動作します。ユーザが、以下に述べるようなアライメント方式を実装すると、この動作を拡張できます。

例えば、FPGA インターフェイスの送信データが次のようなものとします。

YZABCDEFGH IJKLM... (各文字は 8 ビットか 10 ビット幅のバイトを示す)

デコーダ以降、rx_gearbox 以前の PCS の受信データは次のようになります。

YZABCDEFGH IJKLM...

以降では次のようになります。

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 はアライメントがとれていません。1 バイトのオフセットがありますが、16/20 ビット・アライメントが必要です。例えば、特殊文字 'A' を必ず下位バイトに配置する必要があるとします。

20ビットデータを1つ、現在の16/20ビットデータと組み合わせると、次に示す32/40ビットデータになります。

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

| ** A を下位 10 ビットで検出、オフセットを '0' に設定、アラインされたデータ 'BA' を送出

次のクロックサイクルで：

{FEDC}{JIHG}{NMLK}...

| **send out aligned data 'DC'

etc.

16/20 ビット・アライメント後の出力データ：

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

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

| ** A A を上位 10 ビットで検出、オフセットを '10' に設定、アラインされたデータ 'BA' を送出

次のクロックサイクル：

{EDCB}{IHGF}{MLKJ}...

| ** アラインされたデータ 'DC' を送出

etc.

20 ビットアライン後の出力データは：

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

注：8/10 ビットバイトまたは 16/20 ビットワードの LSB が、常に最初に送受信されます。

未使用のクワッド / チャネルと電源

未使用のクワッドとチャネルには、VCCA を供給する必要があります。VCCIB、VCCOB、HDINP/N、HDOUTP/N、REFCLKP/N はフロート状態のままにする必要があります。未使用チャネルの出力はトライステートになり、差動出力ペア間に約 10KΩ の内蔵抵抗が接続されます。コンフィグレーションの間、HDOUTP/N は VCCOB にプルアップされます。

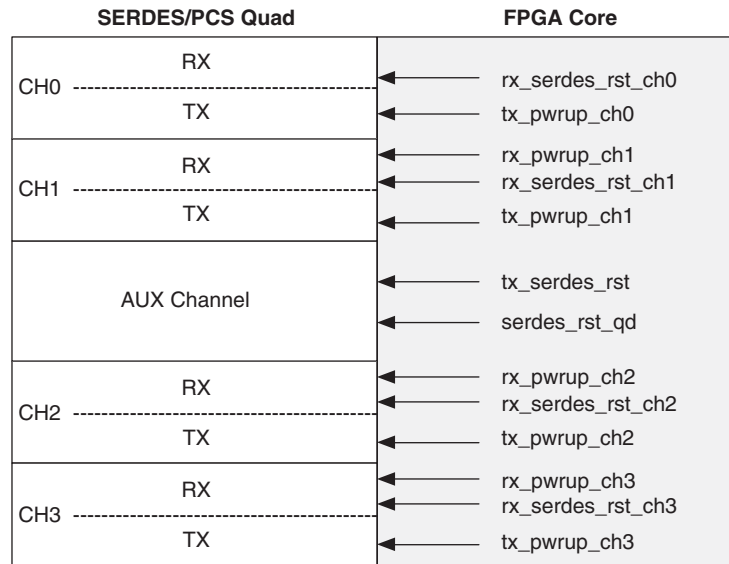
チャネルが送信のみのモードや受信のみのモードの場合でも、そのチャネルの VCCOB および VCCIB には電源を供給する必要があります。未使用 SERDES は、デフォルトでパワーダウン・モードに設定されます。

リセットとパワーダウンの制御

SERDES クワッドには図 8-53 に示すようにマクロ全体、及び各トランスミッタとレシーバ用のリセット及びパワーダウン制御があります。リセット信号はアクティブ High で、パワーダウンは pwrup 信号を Low にドライブすることで行われます。各種のリセット及びパワーダウン制御の動作について、以降のセクションで説明します。

注：デバイスがパワーアップされてチップレベルの電源オンリセットが行われると、SERDES 制御ビット (PCS 内) がクリアされます (またはデフォルト値になります)。その結果、SERDES クワッドがパワーダウン状態になります。

図 8-53. SERDES/PCS クワッドのリセット及びパワーダウン制御



通常全てのリセットは電源オンリセット及び各種の FPGA ファブリック・リセットから行われます。リセットロジックを図 8-54 及び表 8-30 に示します。

図 8-54. SERDES/PCS リセット図

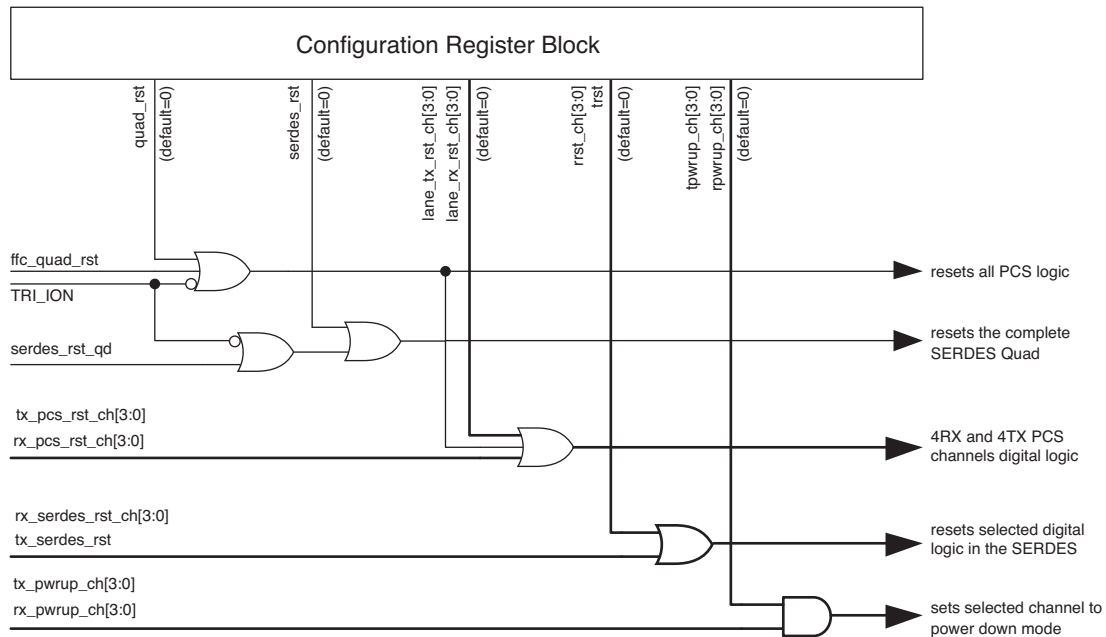


表 8-30. SERDES/PCS リセット表

リセット信号		PCS ¹ TX	PCS ¹ RX	SERDES TX	SERDES RX	PCS CTRL Registers	TX PLL	CDR PLL
FPGA	制御レジスタ							
tx_pcs_rst_ch[3:0]_c	lane_tx_rst[3:0]	X						
rx_pcs_rst_ch[3:0]_c	lane_rx_rst[3:0]		X					
rst_qd_c	quad_rst	X	X	X	X		X	X
serdes_rst_qd_c	serdes_rst			X	X		X	X
rx_serdes_rst_ch[3:0]_c	rrst[3:0] ²				X			X
tx_serdes_rst_c	trst						X ³	
TRI_IION (configuration)		X	X	X	X	X		

1. SB (SERDESブリッジ)、PCS コア、FB (FPGAブリッジ) サブブロックを含みます。
2. 内部使用専用です。CDR PLL をリセットする必要がある場合を除き、リセットは必ず '0' に固定します。
3. tx_serdes_rst_c では TX PLL がリセットされません。tx_pll_lol_qd_s が強制的に High になるだけです。

表 8-31. リセット制御の説明^{1,2,3}

リセット信号		記 述
FPGA	制御レジスタ	
rst_qd_c	quad_rst	アクティブ High、非同期入力。補助チャネル及び PCS を含む SERDES 全チャネルをリセットします。このリセットには serdes_rst、txpll、cdr、lane_tx_rst、lane_rx_rst などがあります。
serdes_rst_qd_c	serdes_rst	アクティブ High、SERDES クワッドへの非同期入力。ソフトウェア・レジスタビットでゲート処理されます。このリセットは SERDES ブロック専用で、TXPLL と CDRPLL が含まれます。
tx/rx_pcs_rst_ch[3:0]_c	lane_tx/rx_rst[0:3]	アクティブ High、非同期入力。SB、PCS コア、及び FB ブロック内の TX/RX チャネルを個別にリセットします。
rx_serdes_rst_ch[3:0]_c	rrst[0:3]	RLOL (Loss-of-Lock)、LOS (Loss-of-Signal)、及び調整回路をリセットします。
tx_serdes_rst_c	trst	AUX PLL の Loss-of-Lock (PLOL) をリセットします。

1. フルデータレート・モードで動作するクワッド内の全てのチャネルで、パラレル側クロックは同相となることが保証されます。
2. ハーフデータレート・モードで動作するクワッド内の全てのチャネルは、チャネルごとに個別に 1/2 分周回路を持ちます。“serdes_rst” のネゲート後に、これら 1/2 分周回路が同相となることを保証するメカニズムがクワッド内に存在しないため、PCS デザインでは分周器（及びパラレル側クロック）が同相ではないことを前提にする必要があります。
3. ハーフデータレート・モードでは、パラレル側クロックが同相であることが保証されないため、マルチチャネル・リンクの送信側と受信側の両方にチャネル間スキューが追加されることがあります。

表 8-32. リセットパルスの仕様

パラメータ	記 述	Min.	Typ.	Max.	単位
t _{SERDES RST QD}	クワッド SERDES リセット High 期間	1			us
t _{RX PCS RST}	チャネル RX PCS リセット High 期間	3			ns
t _{TX PCS RST}	チャネル TX PCS リセット High 期間	3			ns
t _{RX SERDES RST}	チャネル RX SERDES リセット High 期間	3			ns
t _{TX SERDES RST}	クワッド TX SERDES リセット High 期間	3			ns

パワーダウン制御について

各 RX 及び RX チャネルは、ソフトウェア・レジスタビットまたは FPGA からの制御信号によって個別にパワーダウンできます。チャネル別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、及び高速 I/O バッファのみをパワーダウンします。

表 8-33. パワーダウン制御の説明

信 号		記 述
FPGA	レジスタ	
serdes_pd		SERDES クワッドへのアクティブ Low 非同期入力。補助チャネルを含む全チャネルで有効。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します。全クロックが停止し、マクロの消費電力が最小になります。解除後は、TX と RX の両方のリセット・シーケンスに従う必要があります。
tx_pwrup_ch[0:3]_c	tpwrup[0:3]	アクティブ High の送信チャネル・パワーアップ ~ シリアルライザと出力ドライバをパワーアップ。解除後、TX リセット・シーケンスに従う必要があります。
rx_pwrup_ch[0:3]_c	rpwrup[0:3]	アクティブ High の受信チャネル・パワーアップ ~ CDR、入力バッファ（イコライザと増幅器）、及び LOS 検出器をパワーアップ。解除後は、RX リセット・シーケンスに従う必要があります。

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
tPWRDN	serdes_pd 後のパワーダウン時間	20			ns
tPWRUP	serdes_pd 後のパワーアップ時間	20			ns

SERDES/PCS リセット

リセット・シーケンスとリセットステート図

パワーアップ及びコンフィグレーション後は、全ての SERDES リセットと FPGA リセットが適用されます。

リセット・シーケンスの生成

リセット・シーケンスは、IPexpress GUI（Diamond 1.1 以降のバージョンで利用可能）に含まれています。

IPexpress のリセット・シーケンス生成オプションは、Control Setup（制御セットアップ）タブセクションの説明のように選択することを推奨します。SERDES/PCS 用に生成された HDL ファイルには、Tx リセット・ステートマシンと Rx リセット・ステートマシンが含まれます。

ロックステータス信号の定義

tx_pll_lol_qd_s:	: 1 = TX PLL ロックはずれ : 0 = TX PLL ロック TX PLL のロック判定には 1,400,000 UI を要する
rx_cdr_lol_ch[3:0]_s	: 1 = CDR ロックはずれ : 0 = ロックを維持 CDR PLL のロック判定には 400,000 基準クロックサイクル（最悪ケース）を要する
rx_los_low_ch[3:0]_s	: 1 = 各チャネルの LOS（Loss of Signal）検出信号 : 0 = 信号検出

rx_cdr_lol_ch[3:0]_s ステータス信号は、上記のとおり CDR ロック状態のインジケータです。ただし、CDR ロック処理中に入力データが存在しなかった場合、CDR PLL は基準クロックにロックします。そうすることで、入力データが回復時に無視されることを防止します。

CDR ロック状態のチェック時に入力データが確実に存在するようにするには、rx_los_low_ch[3:0]_s 信号を rx_cdr_lol_ch[3:0]_s 信号と組み合わせて使用することを推奨します。

TX リセット・シーケンス

1. QUAD_RESET: パワーアップ時に rst_qd_c と tx_pcs_rst_ch[3:0]_c をアサートする
2. WAIT_FOR_TIMER1: TIMER1 を開始する。最低 20ns 待つ
3. CHECK_PLOL: rst_qd_c を解除する
4. WAIT_FOR_TIMER2: TIMER2 を開始する。TIMER2 が終了し、TX PLL がロックされていない場合は、ステップ 1 に進む
5. NORMAL: tx_pcs_rst_ch#_c を解除する。通常動作中に tx_pll_lol_qd_s が High に遷移した場合は、ステップ 1 に進む

RX リセット・シーケンス

1. WAIT_FOR_PLOL: TX PLL がロックし、受信データが出現するまで待つ。rx_los_low[3:0]_s はアサートされると High になるため、rx_serdes_rst_ch[3:0]_c を 0 に設定する
2. RX_SERDES_RESET: rx_serdes_rst_ch[3:0]_c と rx_pcs_rst_ch[3:0]_c をアサートする
3. WAIT_FOR_TIMER1: 最低 3ns 待つ
4. CHECK_LOL_LOS: rx_serdes_rst_ch_c を解除する。TIMER2 をリセットする
5. WAIT_FOR_TIMER2: cdr_lol_ch[3:0]_s と rx_los_low_ch[3:0] がどちらも Low になるまで待つ。rx_lol_los (rx_cdr_lol_ch_s || rx_los_low_ch_s) が遷移した場合は、ステップ 4 に進む。TIMER2 が終了し、rx_lol_los = 1 となった場合は、ステップ 1 に進む
6. NORMAL: rx_pcs_rst_ch_c を解除する。rx_lol_los が High に遷移した場合は、ステップ 1 に進む

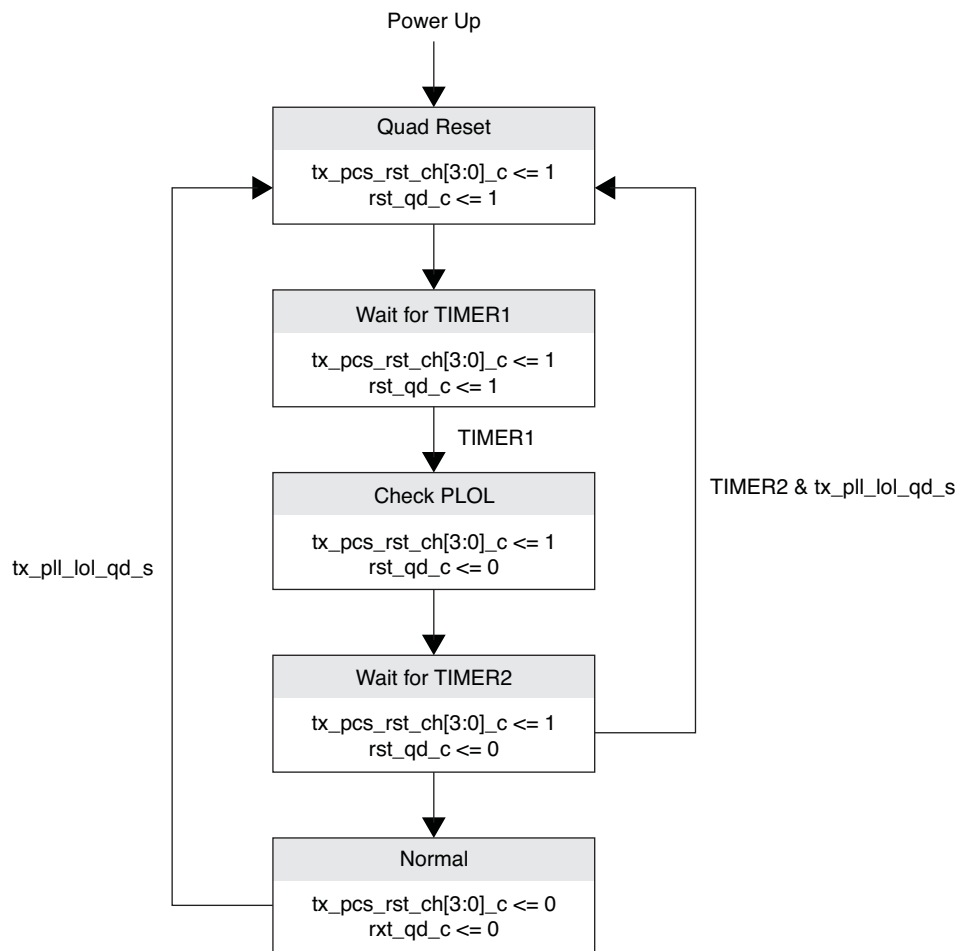
注：RX リセット・シーケンスでは、通常動作中に入力データソースに割り込みがあった場合に CDR 再ロック機能が提供されます。RX リセットはチャンネル単位で適用できます。

リセット・シーケンス状態図を図 8-55 及び 8-56 に示します。

リセット・シーケンスのサンプルコードはラティスのウェブサイトから入手できます（注¹）。

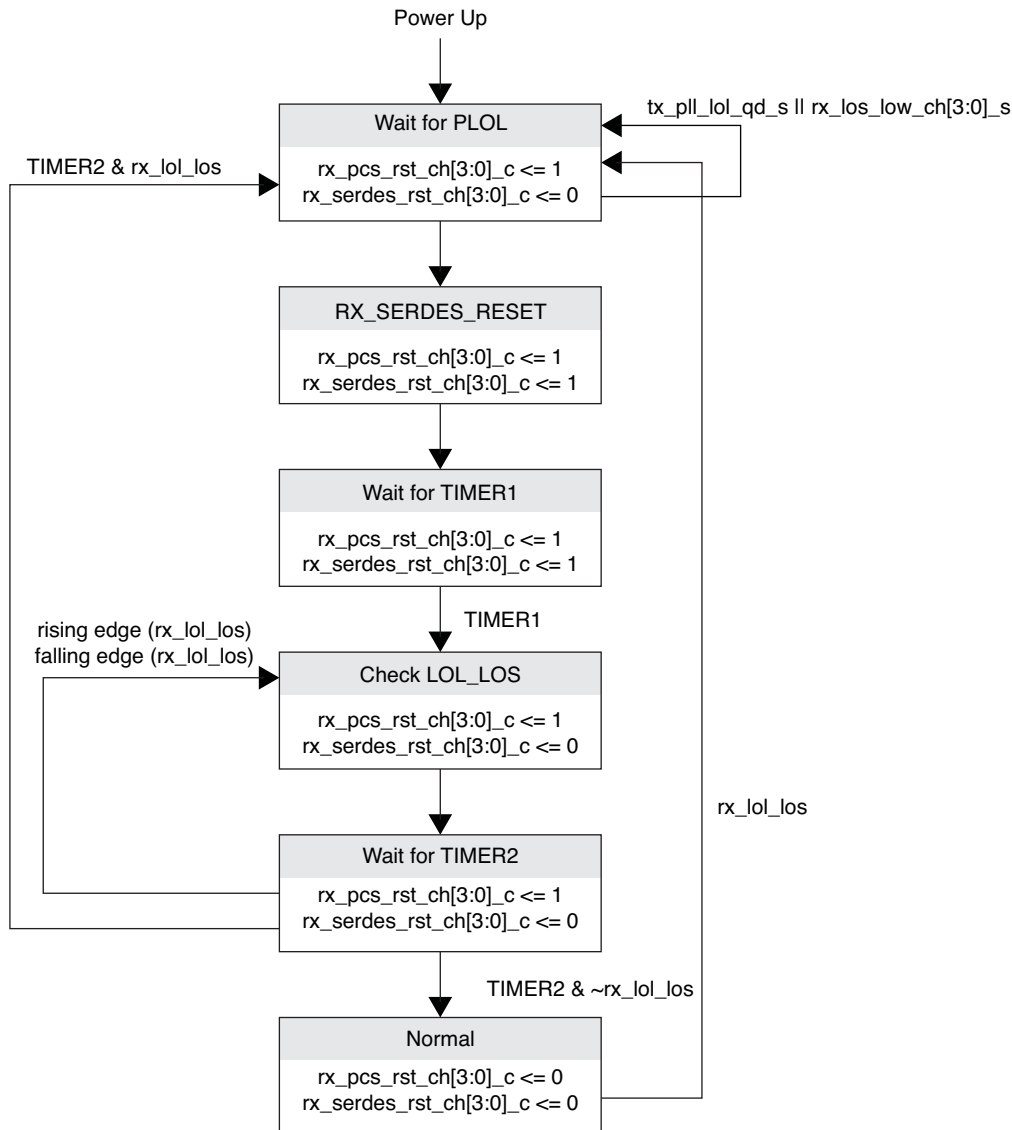
1. 日本語版注：ラティスセミコンダクターでは、日本のお客様向けに別途 RTL を用意しております。ラティス代理店の技術サポートまでお問い合わせください。

図 8-55. TX リセットステート図



Notes:
 TIMER 1: rst_qd_c asserted for a minimum of 20 ns.
 TIMER 2: Time to declare TX PLL lock: 1,400,000 UI.

図 8-56. RX リセットステート図



Notes:

TIMER 1: `rx_serdes_rst_ch[3:0]_c` asserted for minimum 3 ns.

TIMER 2: Time for `rx_lol_los` signal to stay low (400,000 reference clock cycles). Any FPGA clock can be used to satisfy the timer requirement.

In the diagram above, `rx_lol_los` is defined as `rx_cdr_lol_ch[3:0]_s || rx_los_low_ch[3:0]_s`.

The `tx_pll_lol_qd_s` input to the state diagram RTL code should be tied low in Rx Only mode or when the recovered clock is used as the Tx PLL reference clock, as in SDI applications.

When multiple receiver channels `rx_serdes_rst_ch[3:0]_c` are to be asserted, it is recommended to activate the reset signals one channel at a time. Simultaneous resetting of multiple receiver SERDES channels may cause a current surge in the SERDES/PCS quad.

The `rx_los_low` output from SERDES may be triggered for some input streams with continuous zeros, like the SDI pathological pattern. For such applications, the `rx_los_low` input to the reset state machine must be connected to the carrier detect output (must be inverted) of the cable equalizer if available. In general, `CD=1` means carrier is present. So this signal must be inverted to replace `rx_los_low`. If a cable equalizer is not available, users may tie it to zero but in this case, the CDR can lock to a local reference clock when there is no input data present.

供給電源のシーケンス要件

1.5V の VCCIB または VCCOB で SERDES を使用するときは、1.2V 電源を供給しない状態で、SERDES に 1.5V 電源を定常的に供給するべきではありません。通常は、1.2V と 1.5V の両方の電源を SERDES に同時に供給する必要があります。電源と電圧レギュレータの通常の ramp_up 時間の差は問題になりません。

参考文献

- ・ TN1033, [High-Speed PCB Design Considerations](#)
- ・ TN1114, [Electrical Recommendations for Lattice SERDES](#)
- ・ HB1009, [LatticeECP3 Family Handbook](#)
- ・ DS1021, [LatticeECP3 Family Data Sheet](#)

テクニカルサポート支援

ホットライン :1-800-LATTICE (北米)
+1-503-268-8001 (北米以外)

e-mail: techsupport@latticesemi.com

インターネット :www.latticesemi.com

改版履歴（日本語版、v.1.8 以降）

リリース	バージョン	(新) 該当ページ	更新内容
2011 年 10 月	1.8	-	日本語版新規作成
2011 年 11 月	1.9	8-82	表 8-82、QD_02 デフォルト値修正
		8-96	表 8-84、CH_14、tpwrap デフォルト値修正
		8-105	表 8-105、int_all 中 RLOS_HINT、RLOS_HNINT ビット削除
2011 年 11 月	2.0	8-12	表 8-5、tx_full/_half_clk_xh[3:0] に制約指定の記述を追記
		8-19	"TX レーン間スキュー" 節の日本語注を削除（英語版が対応）
		8-20	"ワードアライメント" 節、p.20 最下部 2 行、記述更新
		8-21	"外部ステートマシンのオプション" 節、記述追加
		8-78	表 8-33 の日本語注を削除（英語版が対応）
		8-86	表 8-47 REFCK25X の記述更新
		8-91	表 8-65 word_align_enable の記述更新
		8-105	表 8-105 {CHn_C_ALIGN} の値、更新
2012 年 5 月	2.3 (英語版は 2-1 ~ 2.3 に相当)	-	ロゴ更新
		8-4	表 8-2、328 csBGA 追加
		8-12	表 8-5、refclk2fpga 記述更新
		8-33	表 8-13、脚注 3、Rx I/O 終端、HIGH 記述更新
		8-37	表 8-16、GUI Text 最後の行のタイポ修正
		8-37	"基準クロックとリセットシーケンス" 節追加
		8-40	Generic 8B10B モード節の記述更新
		8-56	第二パラグラフに記述を追加
		8-94	表 8-81、[3:2] 記述追加
		8-94	表 8-82、[4:0]、ビット 4:3 更新
2012 年 8 月	2.4	8-4, 8-29	表 8-11、ECP3-17/328cssBGA のチャンネルについて脚注追記
		8-12	表 8-5、脚注 3 追記
		8-26	図 8-18 差し替え (ispLEVER --> Diamond)
		8-75	未使用チャンネル VCCOB/VCCIB への電源について追記
		8-81	図 8-56、Rx ステートマシン図差し替え（図注追記含む）
		-	ispLEVER --> Diamond、或いは削除

付録 A. コンフィグレーション・レジスタ

クワッドレジスタの概要

表 8-35. クワッドのインターフェイス・レジスタマップ

BA	レジスタ名	D7	D6	D5	D4	D3	D2	D1	D0
クワッド単位の PCS 制御レジスタ									
00	QD_00	reg_sync_toggle	force_int	char_mode	xge_mode				
01	QD_01	(ラティス社用途)							
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						pfifo_clr_sel	internal use only	internal use only
04	QD_04	internal use only							
05	QD_05	internal use only							
06	QD_06	internal use only							
07	QD_07	internal use only							
08	QD_08	internal use only							
09	QD_09	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
クワッド単位の SERDES 制御レジスタ									
0A	QD_0A	internal use only	reserved	tx_refck_sel	refck_dco_en	refck_rterm		refck_out_sel[1]	refck_out_sel[0]
0B	QD_0B	refck25x	bus8bit_sel	reserved	reserved	reserved	reserved	refck_mode[1]	refck_mode[0]
0C	QD_0C	reserved	reserved	reserved	reserved	reserved	reserved	cdr_lol_sel[1]	cdr_lol_sel[0]
0D	QD_0D	internal use only	internal use only	internal use only	pll_lol_sel[1]	pll_lol_sel[0]	tx_vco_ck_div[2]	tx_vco_ck_div[1]	tx_vco_ck_div[0]
0E	QD_0E	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only
0F	QD_0F	plol_int_ctl	~plol_int_ctl	reserved	reserved	reserved	reserved	reserved	reserved
クワッド単位のクロックリセット・レジスタ									
10	QD_10	reserved	reserved	reserved	reserved	serdes_pd	serdes_rst	quad_rst	trst
11	QD_11	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
クワッド単位の PCS ステータスレジスタ									
20	QD_20				int_qd_out	int_ch[3]	int_ch[2]	int_ch[1]	int_ch[0]
21	QD_21	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_22	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	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only
24	QD_24	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only
クワッド単位の SERDES ステータスレジスタ									
25	QD_25	plol	~plol	reserved	reserved	reserved	reserved	reserved	reserved
26	QD_26	plol_int	~plol_int	reserved	reserved	reserved	reserved	reserved	reserved
27	QD_27	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
28	QD_28	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

クワッド単位の PCS 制御レジスタの詳細

表 8-36. PCS 制御レジスタ QD_00

ビット	名称	記 述	タイプ	デフォルト
7	reg_sync_toggle	遷移 = 4 つの TX シリアライザをリセットし、TX レーン間スキューを最小にする レベル = TX シリアライザの通常動作	RW	0
6	force_int	1 = 強制的に割り込み信号を生成 0 = 通常動作	RW	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	RW	0
4	xge_mode	1 = 10Gb イーサネットモードを選択 0 = (チャネルのモード選択によって異なる)	RW	0
3:0	Reserved			

表 8-37. PCS 制御レジスタ QD_01

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-38. PCS 制御レジスタ QD_02

ビット	名称	記 述	タイプ	デフォルト
7:4	high_mark[3:0]	クロック補償 FIFO、High ウォーターマーク。中央値は 4' b1000	RW	4' b1001
3:0	low_mark[3:0]	クロック補償 FIFO、Low ウォーターマーク。中央値は 4' b1000	RW	4' b0111

表 8-39. PCS 制御レジスタ QD_03

ビット	名称	記 述	タイプ	デフォルト
7:3	Reserved			
2	pfifo_clr_sel	1 = pfifo_clr 信号またはチャネル・レジスタビットで FIFO をクリア 0 = pfifo_error 内部信号自体で FIFO をクリア	RW	0
1	Internal use only			
0	Internal use only			

表 8-40. PCS 制御レジスタ QD_04

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-41. PCS 制御レジスタ QD_05

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-42. PCS 制御レジスタ QD_06

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-43. PCS 制御レジスタ QD_07

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-44. PCS 制御レジスタ QD_08

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-45. PCS 制御レジスタ QD_09

ビット	名称	記 述	タイプ	デフォルト
7	ls_sync_status_3_int_ctl	1 = ls_sync_status_3 の割り込みをイネーブル (同期中) 0 = ls_sync_status_3 の割り込みをディセーブル (同期中)	RW	0
6	ls_sync_status_2_int_ctl	1 = ls_sync_status_2 の割り込みをイネーブル (同期中) 0 = ls_sync_status_2 の割り込みをディセーブル (同期中)	RW	0
5	ls_sync_status_1_int_ctl	1 = ls_sync_status_1 の割り込みをイネーブル (同期中) 0 = ls_sync_status_1 の割り込みをディセーブル (同期中)	RW	0
4	ls_sync_status_0_int_ctl	1 = ls_sync_status_0 の割り込みをイネーブル (同期中) 0 = ls_sync_status_0 の割り込みをディセーブル (同期中)	RW	0
3	ls_sync_statusn_3_int_ctl	1 = ls_sync_status_3 が Low に遷移するときに割り込みをイネーブル (同期外) 0 = ls_sync_status_3 が Low に遷移するときに割り込みをディセーブル (同期外)	RW	0
2	ls_sync_statusn_2_int_ctl	1 = ls_sync_status_2 が Low に遷移するときに割り込みをイネーブル (同期外) 0 = ls_sync_status_2 が Low に遷移するときに割り込みをディセーブル (同期外)	RW	0
1	ls_sync_statusn_1_int_ctl	1 = ls_sync_status_1 が Low に遷移するときに割り込みをイネーブル (同期外) 0 = ls_sync_status_1 が Low に遷移するときに割り込みをディセーブル (同期外)	RW	0
0	ls_sync_statusn_0_int_ctl	1 = ls_sync_status_0 が Low に遷移するときに割り込みをイネーブル (同期外) 0 = ls_sync_status_0 が Low に遷移するときに割り込みをディセーブル (同期外)	RW	0

クワッド単位の PCS 制御レジスタの詳細

表 8-46. SERDES 制御レジスタ QD_0A

ビット	名称	記 述	タイプ	デフォルト
7	Internal use only			
6	Reserved		RW	0
5	TX_REFCK_SEL	TxPLL 基準クロック選択 0 = REFCLKP/N 1 = FPGA コア	RW	0
4	Reserved			
3	REFCK_RTERM	基準クロック入力バッファで終端 0 = ハイインピーダンス 1 = 50 Ω	RW	1
2	Reserved		RW	0
1	REFCLK_OUT_SEL[1]	0 = refclk2fpga 出力ディセーブル 1 = refclk2fpga 出力イネーブル	RW	0
0	REFCLK_OUT_SEL[0]	0 = tx_refck_local 出力イネーブル 1 = tx_refck_local 出力ディセーブル	RW	0

注：基準クロック選択制御信号については図 8-9 を参照してください

表 8-47. SERDES 制御レジスタ QD_0B

ビット	名称	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 0 = REFCK_MODE を参照	RW	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	RW	0
5	Reserved		RW	0
4	Reserved		RW	0
3	Reserved		RW	0
2	Reserved		RW	0
1:0	REFCK_MODE[1:0]	REFCK25X = 0 の場合 : 00 = 内部高速ビットクロックは 20x 01 = 内部高速ビットクロックは 10x 10 = 内部高速ビットクロックは 16x 11 = 内部高速ビットクロックは 8x REFCK25X = 1 の場合 : xx = 内部高速ビットクロックは 25x	RW	00

表 8-48. PCS 制御レジスタ QD_0C

ビット	名称	記 述	タイプ	デフォルト
7:2	Reserved			
1:0	CDR_LOL_SET[1:0]	CDR ロックはずれ設定 ロック アンロック 00 = +/-1000ppm x2 +/-1500ppm x2 01 = +/-2000ppm x2 +/-2500ppm x2 10 = +/-4000ppm +/-7000ppm 11 = +/-300ppm +/-450ppm	RW	00

表 8-49. PCS 制御レジスタ QD_0D

ビット	名称	記 述	タイプ	デフォルト
7:6	Internal use only			
5	Internal use only			
4:3	PLL_LOL_SET[1:0]	00 = +/- 1350ppm x2 01 = +/- 2400ppm x2 10 = +/- 6800ppm 11 = +/- 400ppm	RW	0
2:0	TX_VCO_CK_DIV[2:0]	VCO 出力周波数選択 00x = 1 で分周 01x = 2 で分周 100 = 4 で分周 101 = 8 で分周 110 = 16 で分周 111 = 32 で分周	RW	0

表 8-50. PCS 制御レジスタ QD_0E

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-51. PCS 制御レジスタ QD_0F

ビット	名称	記 述	タイプ	デフォルト
7	PLOL_INT_CTL	1 = PLOL ロックはずれに対する割り込みイネーブル 0 = PLOL ロックはずれに対する割り込みディセーブル	RO CR	0
6	~PLOL_INT_CTL	1 = PLOL ロック時の割り込みイネーブル 0 = PLOL ロック時の割り込みディセーブル	RO CR	0
5:0	Reserved			

クワッド単位のリセット及びクロック制御レジスタの詳細

表 8-52. PCS 制御レジスタ QD_10

ビット	名称	記 述	タイプ	デフォルト
7:4	Reserved			
3	serdes_pd	0 = SERDES パワーダウンをアサート	RW	1
2	serdes_rst	1 = SERDES リセットをアサート	RW	0
1	quad_rst	1 = クワッドリセットをアサート	RW	0
0	trst	1 = TX リセット	RW	0

表 8-53. PCS 制御レジスタ QD_11

ビット	名称	記 述	タイプ	デフォルト
7:0	Reserved			

クワッド単位の PCS ステータスレジスタの詳細

表 8-54. PCS ステータスレジスタ QD_20

ビット	名称	記 述	タイプ	デフォルト
7:6	Reserved			
5	ion_delay	0 = tri_ion からの遅延されたグローバル resetn	RO	No
4	int_qd_out	1 = クワッド単位の割り込みステータス	RO	No
3:0	int_ch_out[3:0]	1 = チャネル単位の割り込みステータス	RO	No

表 8-55. PCS ステータスレジスタ QD_21

ビット	名称	記 述	タイプ	デフォルト
7	ls_sync_status_3	1 = sync_status_3 でアラームを生成 0 = sync_status_3 でアラームを生成しない	RO	Yes
6	ls_sync_status_2	1 = sync_status_2 でアラームを生成 0 = sync_status_2 でアラームを生成しない	RO	Yes
5	ls_sync_status_1	1 = sync_status_1 でアラームを生成 0 = sync_status_1 でアラームを生成しない	RO	Yes
4	ls_sync_status_0	1 = sync_status_0 でアラームを生成 0 = sync_status_0 でアラームを生成しない	RO	Yes
3	ls_sync_statusn_3	1 = sync_status_3 が Low に遷移時にアラームを生成（同期外） 0 = sync_status_3 が Low に遷移時にアラームを生成しない（同期外）	RO	Yes
2	ls_sync_statusn_2	1 = sync_status_2 が Low に遷移時にアラームを生成（同期外） 0 = sync_status_2 が Low に遷移時にアラームを生成しない（同期外）	RO	Yes
1	ls_sync_statusn_1	1 = sync_status_1 が Low に遷移時にアラームを生成（同期外） 0 = sync_status_1 が Low に遷移時にアラームを生成しない（同期外）	RO	Yes
0	ls_sync_statusn_0	1 = sync_status_0 が Low に遷移時にアラームを生成（同期外） 0 = sync_status_0 が Low に遷移時にアラームを生成しない（同期外）	RO	Yes

表 8-56. PCS 割り込みステータスレジスタ QD_22

ビット	名称	記 述	タイプ	デフォルト
7	ls_sync_status_3_int	1 = sync_status_3 で割り込みを生成 0 = sync_status_3 で割り込みを生成 しない	RO CR	Yes
6	ls_sync_status_2_int	1 = sync_status_2 で割り込みを生成 0 = sync_status_2 で割り込みを生成 しない	RO CR	Yes
5	ls_sync_status_1_int	1 = sync_status_1 で割り込みを生成 0 = sync_status_1 で割り込みを生成 しない	RO CR	Yes
4	ls_sync_status_0_int	1 = sync_status_0 で割り込みを生成 0 = sync_status_0 で割り込みを生成 しない	RO CR	Yes
3	ls_sync_statusn_3_int	1 = sync_status_3 が Low に遷移時に割り込みを生成 (同期外) 0 = sync_status_3 が Low に遷移時に割り込みを生成 しない (同期外)	RO CR	Yes
2	ls_sync_statusn_2_int	1 = sync_status_2 が Low に遷移時に割り込みを生成 (同期外) 0 = sync_status_2 が Low に遷移時に割り込みを生成 しない (同期外)	RO CR	Yes
1	ls_sync_statusn_1_int	1 = sync_status_1 が Low に遷移時に割り込みを生成 (同期外) 0 = sync_status_1 が Low に遷移時に割り込みを生成 しない (同期外)	RO CR	Yes
0	ls_sync_statusn_0_int	1 = sync_status_0 が Low に遷移時に割り込みを生成 (同期外) 0 = sync_status_0 が Low に遷移時に割り込みを生成 しない (同期外)	RO CR	Yes

表 8-57. PCS ステータスレジスタ QD_23

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-58. PCS ステータスレジスタ QD_24

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

クワッド単位の SERDES ステータスレジスタの詳細

表 8-59. SERDES ステータスレジスタ QD_25

ビット	名称	記 述	タイプ	デフォルト
7	PLOL	1 = PLL ロックはずれ	RO	Yes
6	-PLOL	1 = PLL ロック獲得	RO	Yes
5:0	Reserved			

表 8-60. SERDES 割り込みステータスレジスタ QD_26

ビット	名称	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 しない	RO CR	Y
6	-PLOL_INT	1 = -PLOL で割り込み生成 0 = -PLOL で割り込み生成 しない	RO CR	Y
5:0	Reserved			

表 8-61. SERDES ステータスレジスタ QD_27

ビット	名称	記 述	タイプ	デフォルト
7:0	Reserved			

表 8-62. SERDES ステータスレジスタ QD_28

ビット	名称	記 述	タイプ	デフォルト
7:0	Reserved			

チャネルレジスタの概要

表 8-63. チャネルのインターフェイス・レジスタマップ

BA	レジスタ名	D7	D6	D5	D4	D3	D2	D1	D0
チャネル単位一般制御レジスタ									
00	CH_00					rio_mode	pcie_mode	fc_mode	uc_mode
01	CH_01	word_align_enable	internal use only	internal use only	ge_an_enable		internal use only	invert_tx	invert_rx
02	CH_02	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
03	CH_03		sb_bypass	sb_pfifo_lp	internal use only	enc_bypass	internal use only	tx_gear_bypass	fb_loopback
04	CH_04	lsm_sel	ilsm_en	rx_gear_bypass	ctc_bypass	dec_bypass	wa_bypass	rx_sb_bypass	sb_loopback
05	CH_05	min_ipg_cnt[1]	min_ipg_cnt[0]	match_4_enable	match_2_enable				
06	CH_06	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]
07	CH_07	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]
08	CH_08	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]
09	CH_09	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]
0A	CH_0A	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]
0B	CH_0B	udf_comma_mask[7]	udf_comma_mask[6]	udf_comma_mask[5]	udf_comma_mask[4]	udf_comma_mask[3]	udf_comma_mask[2]	udf_comma_mask[1]	udf_comma_mask[0]
0C	CH_0C	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]
0D	CH_0D	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]
0E	CH_0E	udf_comma_a[9]	udf_comma_a[8]	udf_comma_b[9]	udf_comma_b[8]	udf_comma_mask[9]	udf_comma_mask[8]		
0F	CH_0F					cc_underrun_int_ctl	cc_overrun_int_ctl	fb_rx_fifo_error_int_ctl	fb_tx_fifo_error_int_ctl
チャネル単位SERDES制御レジスタ									
10	CH_10	req_en	req_lv1_set	rcv_dco_en	rate_sel[1]	rate_sel[0]	rx_dco_ck_div[2]	rx_dco_ck_div[1]	rx_dco_ck_div[0]
11	CH_11	internal use only	internal use only	lb_ctl[1]	lb_ctl[0]	internal use only	internal use only	rterm_rx[1]	rterm_rx[0]
12	CH_12	tdrv_amp[2]	tdrv_amp[1]	tdrv_amp[0]	tdrv_pre_set[4]	tdrv_pre_set[3]	tdrv_pre_set[2]	tdrv_pre_set[1]	tdrv_pre_set[0]
13	CH_13	ldr_core2tx_sel				internal use only	internal use only	internal use only	internal use only
14	CH_14	tx_div11_sel	tdrv_dat_sel[1]	tdrv_dat_sel[0]	tdrv_ppre_en	rterm_tx[1]	rterm_tx[0]	rate_mode_tx	tpwrup
15	CH_15	internal use only	internal use only	internal use only	internal use only	ldr_rx2core_en	rx_refck_sel	rate_mode_rx	rpwrup
16	CH_16	rx_div11_sel		rlos_hset[2]	rlos_hset[1]	rlos_hset[0]	rlos_lset[2]	rlos_lset[1]	rlos_lset[0]
17	CH_17		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
チャネル単位クロックリセット・レジスタ									
18	CH_18	internal use only	internal use only				rrst	lane_rx_rst	lane_tx_rst
19	CH_19				tx_f_clk_dis	tx_h_clk_en	rx_f_clk_dis	rx_h_clk_en	sel_sd_rx_clk
チャネル単位一般ステータスレジスタ									
20	CH_20					cc_underrun	cc_overrun	fb_rx_fifo_error	fb_tx_fifo_error
21	CH_21	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					wa_offset[3]	wa_offset[2]	wa_offset[1]	wa_offset[0]
23	CH_23	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
24	CH_24		ffs_ls_sync_status	fb_rxrst_o	fb_txrst_o			cc_re_o	cc_we_o
25	CH_25								
チャネル単位SERDESステータスレジスタ									
26	CH_26		pcie_det_done	rlos_lo	~rlos_lo	rlos_hi	~rlos_hi	rlol	~rlol
27	CH_27	internal use only	internal use only	internal use only	internal use only			cdr_train_done	pci_connect
28	CH_28	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only
29	CH_29	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only	internal use only
2A	CH_2A		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								

チャンネル単位の PCS 制御レジスタの詳細

表 8-64. PCS 制御レジスタ CH00

ビット	名称	記 述	タイプ	デフォルト
7:4	Reserved			
3	rio_mode	1 = RapidIO モード選択 0 = 他のモード選択 (10GbE, 1GbE)	RW	0
2	pcie_mode	1 = PCI Express モード選択 0 = 他のモード選択 (RapidIO, 10GbE, 1GbE)	RW	0
1	fc_mode	1 = Fibre Channel モード選択 0 = 他のモード選択 (PCI Express, RapidIO, 10GbE, 1GbE)	RW	0
0	uc_mode	1 = ユーザ構成モード選択 (G8B10B, 8BSER only, 10BSER only) 0 = 他のモード選択 (Fibre Channel, PCI Express, RapidIO, 10GbE, 1GbE)	RW	0

表 8-65. PCS 制御レジスタ CH01

ビット	名称	記 述	タイプ	デフォルト
7	word_align_enable	1 = 連続カンマアライメントをイネーブル 0 = 連続カンマアライメントをディセーブル	RW	0
6	Internal use only			
5	Internal use only			
4	ge_an_enable	1 = GbE 自動ネゴシエーションをイネーブル 0 = GbE 自動ネゴシエーションをディセーブル	RW	0
3	Reserved			
2	Internal use only			
1	invert_tx	1 = 送信データを反転 0 = 送信データを反転しない	RW	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	RW	0

表 8-66. PCS 制御レジスタ CH02

ビット	名称	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 に設定された場合に PFIFO をクリア。この信号はインターフェイス信号 pfifo_clr と OR 接続される 0 = 通常動作	RW	0
6	pcie_ei_en	1 = PCI Express 電氣的アイドルをイネーブル 0 = 通常動作	RW	0
5:4	pcs_det_time_sel[1:0]	PCS 接続検出時間 11 = 16us 10 = 4us 01 = 2us 00 = 8us	RW	0
3	rx_gear_mode	1 = 選択したチャンネルの受信パスで 2:1 ギアリングをイネーブル 0 = 選択したチャンネルの受信パスで 2:1 ギアリングをディセーブル		
2	tx_gear_mode	1 = 選択したチャンネルの送信パスで 2:1 ギアリングをイネーブル 0 = 選択したチャンネルの送信パスで 2:1 ギアリングをディセーブル	RW	0
1	rx_ch	1 = 受信出力をテスト・キャラクタライズ用ピンでモニタ可能。テスト・キャラクタライズ・モード (PCS 制御レジスタ QD_03 のビット 6) を '1' に設定する必要がある	RW	0
0	tx_ch	1 = 送信 PCS 入力をテスト・キャラクタライズ用ピンから供給。テスト・キャラクタライズ・モードをイネーブルにする必要がある	RW	0

表 8-67. PCS 制御レジスタ CH_03

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			
6	sb_bypass	1 = TX SERDES ブリッジをバイパス 0 = 通常動作	RW	0
5	sb_pfifo_lp	1 = パラレル FIFO を経由した RX から TX へのパラレルループバックをイネーブル 0 = 通常動作	RW	0
4	internal use only			
3	enc_bypass	1 = 8b10b エンコーダをバイパス 0 = 通常動作		
2	internal use only			
1	tx_gear_bypass	1 = PCS TX ギアをバイパス 0 = 通常動作	RW	0
0	internal use only			

表 8-68. PCS 制御レジスタ CH_04

ビット	名称	記 述	タイプ	デフォルト
7	lsm_sel	1 = 外部 RX リンクステートマシン選択 0 = 内部 RX リンクステートマシン選択	RW	0
6	ilsm_en	1 = リンクステートマシンを強制的にイネーブル 0 = リンクステートマシンを強制的にディセーブル	RW	0
5	rx_gear_bypass	1 = PCS RX ギアをバイパス 0 = 通常動作	RW	0
4	ctc_bypass	1 = クロックトレランス補償をバイパス 0 = 通常データを選択	RW	0
3	dec_bypass	1 = 8b10b デコーダをバイパス 0 = 通常動作		
2	wa_bypass	1 = ワードアライメントをバイパス 0 = 通常動作	RW	0
1	rx_sb_bypass	1 = RX SERDES ブリッジをバイパス 0 = 通常動作	RW	0
0	sb_loopback	1 = ブリッジ内で TX から RX へ PCS のループバックをイネーブル 0 = 通常動作	RW	0

表 8-69. PCS 制御レジスタ CH_05

ビット	名称	記 述	タイプ	デフォルト
7:6	min_ipg_cnt[1:0]	強制する最小 IPG	RW	11
5	match_4_enable	1 = 4 キャラクタ・スキップマッチングをイネーブル (4, 3, 2, 1 マッチを使用)	RW	0
4	match_2_enable	1 = 2 キャラクタ・スキップマッチングをイネーブル (4, 3 マッチを使用)	RW	1
3:0	Reserved			

表 8-70. PCS 制御レジスタ CH_06

ビット	名称	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スキップパターン 1 の下位ビット	RW	8'h00

表 8-71. PCS 制御レジスタ CH₀₇

ビット	名称	記 述	タイプ	デフォルト
7:0	cc_match_2[7:0]	ユーザ定義のクロック補償スキップパターン 2 の下位ビット	RW	8'h00

表 8-72. PCS 制御レジスタ CH₀₈

ビット	名称	記 述	タイプ	デフォルト
7:0	cc_match_3[7:0]	ユーザ定義のクロック補償スキップパターン 3 の下位ビット	RW	8'hBC

表 8-73. PCS 制御レジスタ CH₀₉

ビット	名称	記 述	タイプ	デフォルト
7:0	cc_match_4[7:0]	ユーザ定義のクロック補償スキップパターン 4 の下位ビット	RW	8'h50

表 8-74. PCS 制御レジスタ CH_{0A}

ビット	名称	記 述	タイプ	デフォルト
7:6	cc_match_4[9:8]	ユーザ定義のクロック補償スキップパターン 4 の上位ビット [9] = ディスパリティ・エラー [8] = K コントロール	RW	2'b01
5:4	cc_match_3[9:8]	ユーザ定義のクロック補償スキップパターン 3 の上位ビット [9] = ディスパリティ・エラー [8] = K コントロール	RW	2'b01
3:2	cc_match_2[9:8]	ユーザ定義のクロック補償スキップパターン 2 の上位ビット [9] = ディスパリティ・エラー [8] = K コントロール	RW	2'b00
1:0	cc_match_1[9:8]	ユーザ定義のクロック補償スキップパターン 1 の上位ビット [9] = ディスパリティ・エラー [8] = K コントロール	RW	2'b00

表 8-75. PCS 制御レジスタ CH_{0B}

ビット	名称	記 述	タイプ	デフォルト
7:0	udf_comma_mask[7:0]	ユーザ定義カンママスクの下位ビット	RW	8'hFF

表 8-76. PCS 制御レジスタ CH_{0C}

ビット	名称	記 述	タイプ	デフォルト
7:0	udf_comma_a[7:0]	ユーザ定義カンマキャラクタ 'a' の下位ビット	RW	8'h83

表 8-77. PCS 制御レジスタ CH_{0D}

ビット	名称	記 述	タイプ	デフォルト
7:0	udf_comma_b[7:0]	ユーザ定義カンマキャラクタ 'b' の下位ビット	RW	8'h7C

表 8-78. PCS 制御レジスタ CH0E

ビット	名称	記 述	タイプ	デフォルト
7:6	udf_comma_a[9:8]	ユーザ定義カンマキャラクタ 'a' の上位ビット	RW	2'b10
5:4	udf_comma_b[9:8]	ユーザ定義カンマキャラクタ 'b' の上位ビット	RW	2'b01
3:2	udf_comma_mask[9:8]	ユーザ定義カンママスクの上位ビット	RW	2'b11 ¹
1:0	Reserved			

1. 多くの用途ではカンマキャラクタとして K28.5 が使用されます。マスクのデフォルト値は 111111111 です。G8B10B モードでは任意のカンマを使用でき、マスクを 1111111100 にすると 3 つのカンマキャラクタ K28.1、28.5、28.7 を全て検出できます。

表 8-79. PCS 割り込み制御レジスタ CH0F

ビット	名称	記 述	タイプ	デフォルト
7:4	Reserved			
3	cc_underrun_int_ctl	1 = cc_underrun 割り込みをイネーブル 0 = cc_underrun 割り込みをディセーブル	RW	0
2	cc_overnrun_int_ctl	1 = cc_overnrun 割り込みをイネーブル 0 = cc_overnrun 割り込みをディセーブル	RW	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	RW	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	RW	0

チャネル単位の SERDES 制御レジスタの詳細

特に明記しない限り、SERDES 制御レジスタへのライト後は、全チャネルをリセットする必要があります。

表 8-80. SERDES 制御レジスタ CH10

ビット	名称	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	RW	0
6	REQ_LVL_SET	イコライザのレベル設定 1 = ロングリーチ用イコライザ 0 = 中距離ルート用のイコライザ	RW	0
5	RCV_DCC_EN	1 = レシーバ DC 結合をイネーブル 0 = レシーバ AC 結合をイネーブル	RW	0
4:3	RATE_SEL[1:0]	イコライザの極配置選択 : 00 = 高域周波数レンジ 01 = 中域周波数レンジ 10 = 低域周波数レンジ 11 = 予約済み	RW	00
2:0	RX_DCO_CK_DIV[2:0]	VCO 出力周波数選択 : 00x = 1 分周 01x = 2 分周 100 = 4 分周 101 = 8 分周 110 = 16 分周 111 = 32 分周	RW	000

表 8-81. SERDES 制御レジスタ CH_11

ビット	名称	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] = internal use only [2] = internal use only [1] = slb_eq2t_en、イコライザからドライバへのシリアル・ループバックをイネーブル [0] = slb_t2r_en、TX から RX へのシリアル・ループバックをイネーブル	R/W	4'h0
3:2	Reserved			
1:0	RTERM_RX[1:0]	00 = HiZ, 01 = 50 Ohm, 10 = 60 Ohm, 11 = 75 Ohm	R/W	2'b01

表 8-82. SERDES 制御レジスタ CH_12

ビット	名称	記 述	タイプ	デフォルト
7:5	TDRV_AMP[2:0]	CML ドライバの振幅設定 : 000 = 0% 001 = +8% 010 = +11% 011 = +20% 100 = -17% 101 = -12% 110 = -10% 111 = -6%	RW	000
4:0	TDRV_PRE_SET[4:0]	[2:0] TX ドライバ・プリエンファシス・レベル設定 000 = 0% 001 = 5% 010 = 12% 011 = 18% 100 = 25% 101 = 33% 110 = 40% 111 = 48% [4:3] 微調整 (上記 3 ビットの値に加えられる) 00 = 0% 01 = 2% 10 = 3% 11 = 5%	RW	5'b00000

表 8-83. SERDES 制御レジスタ CH_13

ビット	名称	記 述	タイプ	デフォルト
7	ldr_core2tx_sel	1 = コアからの低速シリアルデータを選択	RW	0
6	pden_sel	予約済み	RW	0
5:4	Reserved			
3	TDRV_AMP_BOOST	TX ドライブ振幅ブースト 0 = 0% 1 = -25%	RW	0
2:0	TDRV_DRVCUR_SET[2:0]	000 = 48% 001 = 30% 010 = 60% 011 = 50% 100 = 0% 101 = -7% 110 = 19% 111 = 8%	RW	100

表 8-84. SERDES 制御レジスタ CH14

ビット	名称	記 述	タイプ	デフォルト
7	TX_DIV11_SEL	0 = 送信にフルレートを選択 (SMPTE、HD) 1 = 送信に 11 分周を選択 (SMPTE、SD)	RW	0
6:5	TDRV_DAT_SEL [1:0]	ドライバ出力選択: 00 = シリアルライザからドライバへの多重化されたデータ (通常動作) 11 = slb_eq2t_en= '1' の場合に、イコライザからドライバへのシリアル・ループバック	RW	00
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	RW	0
3:2	RTERM_TX[1:0]	TX 終端抵抗の選択。PCI Express モード選択の場合はディセーブル 0x = 5K Ω 10 = 50 Ω 11 = 75 Ω	RW	10
1	RATE_MODE_TX	0 = 送信にフルレートを選択 1 = 送信にハーフレートを選択	RW	0
0	tpwrap	0 = 送信チャンネルをパワーダウン 1 = 送信チャンネルをパワーアップ	RW	0

表 8-85. SERDES 制御レジスタ CH15

ビット	名称	記 述	タイプ	デフォルト
7	Internal use only			
6	Internal use only			
5	Internal use only			
4	Internal use only			
3	ldr_rx2core_en	1 = 高速受信入力から FPGA 内の低速 SERDES への配線用にバウンダリスキャン入力パスをイネーブル (OOB 用途)	RW	0
2	rx_refck_sel	RX CDR 基準クロック選択 0 = REFCLKP/N 1 = FPGA コア	RW	0
1	RATE_MODE_RX	0 = 受信にフルレートを選択 1 = 受信にハーフレートを選択	RW	0
0	rpwrap	0 = 受信チャンネルをパワーダウン 1 = 受信チャンネルをパワーアップ	RW	0

表 8-86. SERDES 制御レジスタ CH16

ビット	名称	記 述	タイプ	デフォルト
7	RX_DIV11_SEL	0 = 受信にフルレートを選択 (SMPTE、HD) 1 = 受信に 11 分周を選択 (SMPTE、SD)	RW	0
6	rlos_sel	1 = rlos_hi を選択 0 = rlos_lo を選択	RW	0
5:3	Reserved		RW	000
2:0	RLOS_LSET[2:0]	小振幅に対する LOS 検出基準電流調整 000 = デフォルト 010 = +15% 011 = +25%	RW	010

表 8-87. SERDES 割り込み制御レジスタ CH₁₇

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			
6	pci_det_done_int_ctl	1 = PCI Express ファーエンド・レシーバ検出割り込みをイネーブル	RW	0
5	rlos_lo_int_ctl	1 = 入力レベルがプログラムされた LOW 閾値 (rlos_set を使用) を下回った場合に RX LOS 割り込みをイネーブル	RW	0
4	~rlos_lo_int_ctl	1 = 入力レベルがプログラムされた LOW 閾値と等しいか上回った場合に RX LOS 割り込みをイネーブル	RW	0
3	Reserved		RW	0
2	Reserved		RW	0
1	rlol_int_ctl	1 = レシーバ LOL 割り込みをイネーブル	RW	0
0	~rlol_int_ctl	1 = レシーバが LOL から復帰したときに割り込みをイネーブル	RW	0

チャネル単位のリセット及びクロック制御レジスタの詳細

表 8-88. リセット及びクロック制御レジスタ CH₁₈

ビット	名称	記 述	タイプ	デフォルト
7	Internal use only			
6	Internal use only			
5:3	Reserved			
2	rrst	1 = RX リセット	RW	0
1	lane_rx_rst	1 = 受信ロジックにリセット信号をアサート	RW	0
0	lane_tx_rst	1 = 送信ロジックにリセット信号をアサート	RW	0

表 8-89. リセット及びクロック制御レジスタ CH₁₉

ビット	名称	記 述	タイプ	デフォルト
7:5	Reserved			
4	tx_f_clk_dis	1 = tx_f_clk ディセーブル	RW	0
3	tx_h_clk_en	1 = tx_h_clk イネーブル	RW	0
2	rx_f_clk_dis	1 = rx_f_clk ディセーブル	RW	0
1	rx_h_clk_en	1 = rx_h_clk イネーブル	RW	0
0	sel_sd_rx_clk	1 = sd_rx_clk を選択	RW	0

チャネル単位の PCS ステータスレジスタの詳細

表 8-90. PCS ステータスレジスタ CH₂₀

ビット	名称	記 述	タイプ	デフォルト
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	Yes
3	cc_underrun	1 = CTC FIFO アンダーラン 0 = CTC FIFO アンダーランなし	RO	Yes
2	cc_overnrun	1 = CTC FIFO オーバーラン 0 = CTC FIFO オーバーランなし	RO	Yes
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	Yes
0	fb_tx_fifo_error	1 = FPGA ブリッジ (FB) TX FIFO オーバーラン 0 = FB TX FIFO オーバーランなし	RO	Yes

表 8-91. PCS ステータスレジスタ CH₂₁

ビット	名称	記 述	タイプ	デフォルト
7:0	prbs_errors ¹	PRBS エラーのカウンタ数。リードするとクリアされる。 フリップフロップで保持される	RO CR	No

1. 組み込み PRBS 生成器及びチェッカはラティス社内用途です

表 8-92. PCS ステータスレジスタ CH₂₂

ビット	名称	記 述	タイプ	デフォルト
7:4	Reserved			
3:0	wa_offset[3:0]	ワードアライナのオフセット	RO	No

表 8-93. PCS 割り込みステータスレジスタ CH₂₃

ビット	名称	記 述	タイプ	デフォルト
7:4	Reserved			
3	cc_underrun_int	1 = cc_underrun 割り込みを生成 0 = cc_underrun 割り込みを生成しない	RO CR	Yes
2	cc_overnrun_int	1 = cc_overnrun 割り込みを生成 0 = cc_overnrun 割り込みを生成しない	RO CR	Yes
1	fb_rx_fifo_error_int	1 = fb_rx_fifo_error 割り込みを生成 0 = fb_rx_fifo_error 割り込みを生成しない	RO CR	Yes
0	fb_tx_fifo_error_int	1 = fb_tx_fifo_error 割り込みを生成 0 = fb_tx_fifo_error 割り込みを生成しない	RO CR	Yes

表 8-94. PCS ステータスレジスタ CH₂₄

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			
6	ffs_ls_sync_status	1 = リンクステートマシン LSM は同期している 0 = LSM は同期外していない	RO	No
5	fb_rxrst_o	1 = FPGAブリッジ RX 通常動作 0 = FPGAブリッジ RX リセット	RO	No
4	fb_txrst_o	1 = FPGAブリッジ TX 通常動作 0 = FPGAブリッジ TX リセット	RO	No
3	Reserved			
2	Reserved			
1	cc_re_o	1 = CTC FIFO リードイネーブル 0 = CTC FIFO リードディセーブル	RO	No
0	cc_we_o	1 = CTC FIFO ライトイネーブル 0 = CTC FIFO ライトディセーブル	RO	No

表 8-95. PCS ステータスレジスタ CH₂₅

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			

チャンネル単位の SERDES ステータスレジスタの詳細

表 8-96. SERDES ステータスレジスタ CH₂₆

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			
6	pci_det_done	1 = トランスミッタがレシーバ検出プロセスを未完了 0 = トランスミッタがレシーバ検出プロセスを完了	RO CR	Yes
5	rlos_lo	1 = レシーバで検出した入力信号がプログラムされた LOW 閾値よりも小さいことを示す	RO CR	Yes
4	~rlos_lo	1 = レシーバで検出した入力信号がプログラムされた LOW 閾値と等しいか大きいことを示す	RO CR	Yes
3	Reserved		RO CR	Yes
2	Reserved		RO CR	Yes
1	rlol	1 = がデータにロックしていないことを示す。CDR は基準クロックにロックしている	RO	Yes
0	~rlol	1 = がデータにロックしていることを示す	RO	Yes

表 8-97. SERDES ステータスレジスタ CH₂₇

ビット	名称	記 述	タイプ	デフォルト
7	Internal use only			
6	Internal use only			
5	Internal use only			
4	Internal use only			
3:2	Reserved			
1	cdr_trained	1 = CDR トレーニングが完了したことを示す	RO	No
0	pci_connect	1 = トランスミッタがレシーバを検出した 0 = トランスミッタがレシーバを検出していない	RO	No

表 8-98. SERDES ステータスレジスタ CH₂₈

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-99. SERDES ステータスレジスタ CH₂₉

ビット	名称	記 述	タイプ	デフォルト
7:0	Internal use only			

表 8-100. SERDES 割り込みステータスレジスタ CH2A

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			
6	pci_det_done_int	1 = pci_det_done で生成された割り込み	RO CR	Yes
5	rlos_lo_int	1 = rlos_lo で生成された割り込み	RO CR	Yes
4	-rlos_lo_int	1 = -rlos_lo で生成された割り込み	RO CR	Yes
3	Reserved		RO CR	Yes
2	Reserved		RO CR	Yes
1	rlol_int	1 = rlol で生成された割り込み	RO CR	Yes
0	-rlol_int	1 = -rlol で生成された割り込み	RO CR	Yes

表 8-101. PCS ステータスレジスタ CH2B

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			

表 8-102. PCS ステータスレジスタ CH2C

ビット	名称	記 述	タイプ	デフォルト
7	Reserved			

付録 B. 各種規格用のレジスタ設定

各種規格用チャネル単位のレジスタ設定

表 8-103. 各種規格用のチャネル単位のレジスタ設定

キャラクタ	1GbE	10GbE	1GFC	PCI-Ex	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 DLP	
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)					R (skip)
K29.7 (FD)	EOP	T		END	
K30.7 (FE)	ERR	ERR		END BAD	

各種規格用クワッド単位のレジスタ設定

表 8-104. 各種規格用のクワッド単位のレジスタ設定

レジスタ	1GbE	10GbE	1G, 2G FC	PCI-Ex 1x	PCI-Ex 4x	RapidIO 1x	RapidIO 4x
comma_a_lo	hex 03	hex 03	hex 03	hex 03	hex 03	hex 03	hex 03
comma_b_lo	hex FC	hex FC	hex FC	hex FC	hex FC	hex FC	hex FC
comma_mask_lo	hex 7F	hex 7F	hex 7F	hex 7F	hex 7F	hex 7F	hex 7F

付録 C. 属性のクロスリファレンス一覧表

表 8-105. 付録 C. 属性のクロスリファレンス一覧表

独立属性名	依存する属性名	属性値	レジスタマップ
QUAD_MODE	{Qn_REFCK_NQ_EN} ⁸	SINGLE : {0} MASTER : {1} SLAVE : {1} SLAVE_END : {0}	{QD_0b[2]}
CHn_PROTOCOL	{10G_MODE, CHn_PROT_MODE, CHn_RX_DET, CHn_GE_AN_EN}	GIGE : {0, 0000,00,1} FC : {0,0010,00,0} XAUI : {1,0000,00,0} SRIO : {0,1000,00,0} PCIE : {0,0100,00,0} SDI : {0,0001,00,0} G8B10B : {0,0001,00,0} 10BSER : {0,0001,00,0} 8BSER : {0,0001,00,0} CPRI : {0,0001,00,0} OBSAI : {0,0001,00,0}	{QD_00[4], CH_00[3:0], CH_02[5:4], CH_01[4]}
CHn_MODE	{CHn_TXPWDNB, CHn_RXPWDNB}	RXTX : {11} RXONLY : {01} TXONLY : {10} DISABLED : {00}	{CH_14[0], CH_15[0]}
TX_DATARATE_RANGE	{PLL_DIV}	LOWLOW : {110} LOW : {101} MEDLOW : {100} MED : {010} MEDHIGH : {010} HIGH : {000}	{QD_0D[2:0]}
CHn_RX_DATARATE_RANGE	{CHn_CDR_DIV}	LOWLOW : {110} LOW : {101} MEDLOW : {100} MED : {010} MEDHIGH : {000} HIGH : {000}	{CH_10[2:0]}
REFCK_MULT	{REFCK25X, REFCK_MODE}	8X : {0,11} 10X : {0,01} 16X : {0,10} 20X : {0,00} 25X : {1,00}	{QD_0b[7], QD_0b[1:0]}
CHn_RX_DATA_RATE	{CHn_RX_RATE_MODE, CHn_RX_DIV11}	FULL : {00} DIV2 : {10} DIV11 : {01}	{CH_15[1], CH_16[7]}
CHn_TX_DATA_RATE	{CHn_TX_RATE_MODE, CHn_TX_DIV11}	FULL : {00} DIV2 : {10} DIV11 : {01}	{CH_14[1], CH_14[7]}
CHn_TX_DATA_WIDTH	{CHn_TXCLKF, CHn_TXCLKH, CHn_TX_GEAR}	8 : {0,1,0} 10 : {0,1,0} 16 : {0,1,1} 20 : {0,1,1}	{CH_19[4], CH_19[3], CH_02[2]}
CHn_RX_DATA_WIDTH	{CHn_RXCLKF, CHn_RXCLKH, CHn_RX_GEAR}	8 : {0,0,0} 10 : {0,0,0} 16 : {1,1,1} 20 : {1,1,1}	{CH_19[2], CH_19[1], CH_02[3]}
CHn_TX_FIFO		DISABLED : {1} ENABLED : {0}	{CH_03[1]}
CHn_RX_FIFO		DISABLED : {1} ENABLED : {0}	{CH_04[5]}

表 8-105. 付録 C. 属性のクロスリファレンス一覧表 (Continued)

独立属性名	依存する属性名	属性値	レジスタマップ
PLL_SRC	{TXREFCK_NQ_SEL, TXREFCK_SEL}	REFCLK_EXT : {0, 0} REFCLK_CORE : {0, 1} REFCLK_NQ : {1, 0} ⁸	{QD_0B[3], QD_0A[5]}
CHn_CDR_SRC	{RXREFCK_NQ_SEL, CHn_RXREFCK_SEL, CHn_TRAIN_EN, CHn_TRAIN_DIV }	REFCLK_EXT : {0, 0, 0, 0} REFCLK_CORE : {0, 1, 0, 0} REFCLK_NQ : {1, 0, 0, 0} ⁸ TRAIN_DIV4 : {0, 0, 1, 0} ⁸ TRAIN_DIV8 : {0, 0, 1, 1} ⁸	{QD_0B[4], CH_15[2], CH_15[7], CH_15[6]}
CHn_TDRV ⁷	{CHn_TDRV_AMP, CHn_TDRV_DRVCUR_SET, CHn_TDRV_AMP_BOOST}	-4: {110, 100, 1} -3: {100, 101, 0} -2: {100, 100, 0} -1: {101, 100, 0} 0: {000, 100, 0} 1: {001, 100, 0} 2: {011, 100, 0} 3: {100, 000, 0} 4: {000, 000, 0}	{CH_12[7:5], CH_13[2:0], CH_13[3]}
CHn_TDRV (for PCI Express Protocol only)	{CHn_TDRV_AMP, CHn_TDRV_DRVCUR_SET, CHn_TDRV_AMP_BOOST, CHn_TDRV_PRE_EN, CHn_TDRV_PRE_SET}	2: {100, 000, 0, 1, 00101}	{CH_12[7:5], CH_13[2:0], CH_13[3], CH_14[4], CH_12[4:0]}
CHn_TX_PRE	{CHn_TDRV_PRE_EN, CHn_TDRV_PRE_SET}	DISABLED: {0, 00000} 0: {1, 00000} 1: {1, 00001} 2: {1, 00010} 3: {1, 00011} 4: {1, 00100} 5: {1, 00101} 6: {1, 00110} 7: {1, 00111}	{CH_14[4], CH_12[4:0]}
CHn_RTERM_TX		50: {10} 75: {11} 5K: {0X}	{CH_14[3:2]}
CHn_RX_EQ	{CHn_REQ_EN, CHn_REQ_LVL_SET, CHn_RATE_SEL}	DISABLED : {0, 0, 00} MID_LOW : {1, 0, 10} MID_MED : {1, 0, 01} MID_HIGH : {1, 0, 00} LONG_LOW : {1, 1, 10} LONG_MED : {1, 1, 01} LONG_HIGH : {1, 1, 00}	{CH_10[7], CH_10[6], CH_10[4:3]}
CHn_RTERM_RX	{CHn_RX_RTERM}	50 : {01} 60 : {10} 75 : {11} HIGH: {00}	{CH_11[1:0]}
CHn_RX_DCC		AC: {0} DC: {1}	{CH_10[5]}
CHn_LOS_THRESHOLD_LO ¹	{CHn_RLOS_E}	0: {0000} 1: {0001} 2: {0010} 3: {0011} 4: {0100} 5: {0101} 6: {0110} 7: {0111}	{mc1_ser_ctl_chN[75], CH_16[2:0]}
PLL_TERM		50: {1} 2K: {0}	{QD_0A[3]}

表 8-105. 付録 C. 属性のクロスリファレンス一覧表 (Continued)

独立属性名	依存する属性名	属性値	レジスタマップ
PLL_DCC		AC: {0} DC: {1}	{QD_0A[4]}
PLL_LOL_SET		0: {00} 1: {01} 2: {10} 3: {11}	{QD_0D[4:3]}
CHn_TX_SB	{CHn_TXPOL, CHn_TXSBBYP}	DISABLED: {0,0} ENABLED : {1,0}	{CH_01[1], CH_03[6]}
CHn_RX_SB	{CHn_RXPOL, CHn_RXSBBYP}	DISABLED: {0,0} ENABLED : {1,0}	{CH_01[0], CH_04[1]}
CHn_TX_8B10B		ENABLED : {0} DISABLED: {1}	{CH_03[3]}
CHn_RX_8B10B		ENABLED : {0} DISABLED: {1}	{CH_04[3]}
CHn_COMMA_A		Note 2	{QD_0C[0:7], QD_0E[6:7]}
CHn_COMMA_B		Note 2	{QD_0D[0:7], QD_0E[4:5]}
CHn_COMMA_M		Note 2	{QD_0B[0:7], QD_0E[2:3]}
CHn_RXWA		DISABLED: {1} ENABLED : {0}	{CH_04[2]}
CHn_ILSM		DISABLED: {1} ENABLED : {0}	{CH_04[7]}
CHn_CTC	{CHn_RXRECCLK}	ENABLED : {0,0} DISABLED {1,1}	{CH_19[0] , CH_04[4]}
CHn_CC_MATCH1		Note 2	{QD_0A[1:0], CH_06[7:0]}
CHn_CC_MATCH2		Note 2	{QD_0A[3:2], CH_07[7:0]}
CHn_CC_MATCH3		Note 2	{QD_0A[5:4], CH_08[7:0]}
CHn_CC_MATCH4		Note 2	{QD_0A[7:6], QD_09[7:0]}
CHn_CC_MATCH_MODE	{CHn_MATCH_2_EN, CHn_MATCH_4_EN}	1: {0,0} 2: {1,0} 4: {0,1}	{CH_05[4], CH_05[5]}
CHn_CC_MIN_IPG		0: {00} 1: {01} 2: {10} 3: {11}	{CH_05[7:6]}

表 8-105. 付録 C. 属性のクロスリファレンス一覧表 (Continued)

独立属性名	依存する属性名	属性値	レジスタマップ
CCHMARK		0 : {0000} 1 : {0001} 2 : {0010} 3 : {0011} 4 : {0100} 5 : {0101} 6 : {0110} 7 : {0111} 8 : {1000} 9 : {1001} 10 : {1010} 11 : {1011} 12 : {1100} 13 : {1101} 14 : {1110} 15 : {1111}	{QD_02[7:4]}
CCLMARK		0 : {0000} 1 : {0001} 2 : {0010} 3 : {0011} 4 : {0100} 5 : {0101} 6 : {0110} 7 : {0111} 8 : {1000} 9 : {1001} 10 : {1010} 11 : {1011} 12 : {1100} 13 : {1101} 14 : {1110} 15 : {1111}	{QD_02[3:0]}
CHn_SSLB		DISABLED : {0000, 00} ENABLED_EQ2T : {0010, 11} ENABLED_T2R : {0001, 00}	{CH_11[7:4], CH_14[6:5]}
CHn_SPLBPORTS ⁴	{PFIFO_CLR_SEL, CHn_SB_PFIFO_LP, CHn_SB_LOOPBACK}	DISABLED : {0, 0, 0} ENABLED : {1, 1, 1}	{QD_03[2], CH_03[5], CH_04[0]}
QD_REFCK2CORE		DISABLED : {0} ENABLED : {1}	{QD_0a[1]}
INT_ALL	{PLOLINT, PLOLNINT, CHn_PCIDETINT, CHn_RLOSLINT, CHn_RLOSLNINT, CHn_RLOLINT, CHn_RLOLNINT, CHn_LSSYNCINT, CHn_LSSYNCNINT, CHn_TXFIFOINT, CHn_RXFIFOINT, CHn_CCORUNINT, CHn_CCURUNINT}	DISABLED : { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0000, 0 000, 0, 0, 0, 0} ENABLED : { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1111, 1 111, 1, 1, 1, 1}	{QD_0F[7], QD_0F[6], CH_17[6], CH_17[5], CH_17[4], CH_17[1], CH_17[0], QD_09[7:4], QD_09[3:0], CH_0F[0], CH_0F[1], CH_0F[2], CH_0F[3]}
CHn_LDR	{CHn_LDR_RX_EN, CHn_LDR_TX_SEL}	DISABLED : {0, 0} RXTX : {1, 1} RXONLY : {1, 0} TXONLY : {0, 1}	{CH_15[3], CH_13[7]}
{PLL_SRC, CHn_CDR_SRC}	{REFCKLOCAL}	注 5	{QD_0A[0]}

表 8-105. 付録 C. 属性のクロスリファレンス一覧表 (Continued)

独立属性名	依存する属性名	属性値	レジスタマップ
{CHn_TX_8B10B, CHn_RX_8B10B}	{BUS8BIT_SEL}	注 6	{QD_0B[6]}
{CHn_RXWA, CHn_ILSM}	{CHn_SIG_DET}	{DISABLED, X} ³ {0} {ENABLED, DISABLED}{0} {ENABLED, ENABLED}{1}	{CH_04[6]}
{CHn_RXWA, CHn_ILSM}	{CHn_C_ALIGN}	{DISABLED, X} ²³ {0} {ENABLED, DISABLED}{0} {ENABLED, ENABLED}{0}	{CH_01[7]}

- rx_los_low は、データレートが 1 Gbps よりも大きく、最小 CID (Consecutive Identical Digits、即ち 8b10b から送信される入力信号が遷移しない連続ビット数の最小) が 7 ビットの信号が検出されたことのみを示すようになりました。rx_los_low は、PCI Express 及び SDI を除く全プロトコルでのデフォルト設定 rlos_lset[2:0] = 2 の場合のみに対応しています。PCI Express に対しては、2 と 3 に対応しています。SDI モードでは、外部 SDI ケーブルイコライザからのキャリア検出力信号 (/CD) の使用を推奨します。rlox_hset には対応していません。
- デフォルトまたは、“PCS Advanced Setup” (PCS 詳細設定) 構成 GUI でユーザが指定した 10 ビット・シンボルコード。GUI では 10 ビット・シンボルコードが LSB から MSB の順で表現され、ソフトウェアが使用するテーブルではビット表現が適切に入れ替えられます。
- X = 無視。
- いずれかのチャンネルがイネーブルされると、クワッドビットが 1 に設定されます。
- PLL_SRC 及び、イネーブルされた全ての CHn_CDR_SRC が REFCLK_CORE に設定された場合、このビットは 1 となり、それ以外の場合は 0 となる必要があります。
- イネーブルされたチャンネル CHn_TX_8B10B または CHn_RX_8B10B のいずれかが DISABLED となった場合、このビットは 1 となり、それ以外の場合は 0 となる必要があります。
- TDRV_AMP 属性は TDRV に置き換えられました。.lpc ファイルを開くと、ソフトウェアが自動的に TDRV_AMP を TDRV に置換します。ユーザが PCS モジュールを再生成することなく再コンパイルしようとすると、ソフトウェアによって automake.log ファイルにエラーが生成されます。.txt ファイルを編集するか、IPexpress を使用して PCS モジュールを再生成する必要があります。
- refclk_to/from_nq 信号はラティス社用途 (internal Use Only) です。

表 8-106. プロトコル固有の SERDES 設定オプション

プロトコル	DATARATE	DATARATE 範囲	REFCK 乗数	データ幅	受信イコライザ 設定
GbE	1.25	MED	10x, 20x, 25x	8, 16	DISABLE, MID_MED, LONG_MED
SGMII	1.25	MED	10x, 20x, 25x	8	
PCI Express	2.5	HIGH	25x, 20x	8, 16	MID_LOW, DISABLE, MID_HIGH LONG_HIGH
XAUI	3.125	HIGH	20x, 10X	16	
G8B8B	ANY_VALUE	LOWLOW, LOW MEDLOW, MED, MEDHIGH, HIGH	10x, 20x, 25x	8, 16	DISABLE, MID_LOW, MID_MED, MID_HIGH, LONG_LOW, LONG_MED, LONG_HIGH
8-Bit SERDES Only			8x, 16x	8, 16	
10-Bit SERDES Only			10x, 20x, 25x	10, 20	
USER_DEF			8x, 10x, 16x, 20x, 25x	8, 10, 16, 20	

- MID : 長さ約 20 インチ
LONG : 長さ約 40 インチ
LOW : 1.2Gbps 未満
MED : 1.2Gbps から 2Gbps
HIGH : 2 Gbps 以上

付録 D. Lattice Diamond の概要

この付録では LatticeECP2M SERDES/PCS モジュールを含んだ設計プロジェクトに対する、Lattice Diamond ソフトウェアの使い方について説明します。

Lattice Diamond の一般的な使い方については、“Lattice Diamond User Guide”（ユーザガイド）を参照してください（日本語注：日本語版オリジナルマニュアルがウェブからダウンロード可能です）。

FPGA 設計プロジェクトで ispLEVER ソフトウェアを使用した経験がある場合、Lattice Diamond は大きく違って見えるかもしれません。しかし、Lattice Diamond は ispLEVER と同じツールセットとワークフローに基づいているため、よく見ると類似点が多数見つかります。このような変更を行った目的は、統合されたツール群と、さらに拡張されたユーザインターフェイスを提供するためです。

ispLEVER プロジェクトの Lattice Diamond への変換

ispLEVER で作成された設計プロジェクトは、容易に Lattice Diamond にインポートできます。ispLEVER のプロセスプロパティ（Diamond のストラテジ設定に似ています）と PCS モジュールを除いて、このプロセスは自動的に行われます。プロジェクトのインポート後は、そのストラテジを設定し、全ての PCS モジュールを生成する必要があります（“ストラテジ”についてはマニュアルを参照）。

ispLEVER 設計プロジェクトのインポート

ispLEVER プロジェクトのバックアップコピーを作成するか、Diamond プロジェクトにする新しいプロジェクトを作成します。

1. Diamond で **File > Open > Import ispLEVER Project**（プロジェクトのインポート）と辿る
2. ispLEVER プロジェクト・ダイアログボックスで、プロジェクトの .syn ファイルを参照して開く
3. 必要に応じて Diamond プロジェクトの基本ファイル名や場所を変更する。場所を変更すると、新しい Diamond のファイルは新しい場所に格納されるが、元のソースファイルは移動もコピーもされない。 Diamond プロジェクトは元の場所にあるソースファイルを参照する。

プロジェクトファイルは、デフォルトのストラテジ設定を持つ Diamond フォーマットに変換されます。

PCS モジュールの調整

IPexpress で作成された PCS モジュールは特殊なファイル構造を持ち、ispLEVER からプロジェクトをインポートするときに調整が必要です。この調整方法は 2 通りあり、望ましい方法は Diamond でモジュールを再生成することです。ところが、この方法ではモジュールが最新バージョンにアップグレードされます。一般的にアップグレードは好ましいことですが、何らかの理由によって PCS モジュールをアップグレードしたくない場合は、モジュールの .txt ファイルを実装フォルダにコピーして手動で調整する方法もあります。この方法を使用する場合は、.txt ファイルを将来の全ての実装フォルダに忘れずにコピーしてください。

PCS モジュールの再生成

1. File List（ファイルリスト）ビューの Input Files（入力ファイル）フォルダで、PCS モジュールを探します。モジュールは .lpc、.v、または .vhd ファイル。
2. File List（ファイルリスト）ビューでモジュールが Verilog または VHDL ファイルとして表示され、モジュールを再生成する場合は、そのモジュールの .lpc ファイルをインポートする。
 - a. File List ビューで実装フォルダ () を右クリックし、**Add > Existing File** を選択
 - b. モジュールの .lpc ファイル `<module_name>.lpc` を参照して選択
 - c. **Add** をクリック。 .lpc ファイルが File List ビューに追加される
 - d. モジュールの Verilog または VHDL ファイルを右クリックし、**Remove**（削除）を選択

3. File List でモジュールの .lpc ファイルをダブルクリックする。そのモジュールの IPexpress ダイアログボックスが表示される
4. ダイアログボックス下部の **Generate** (生成) をクリック。Generate Log (生成ログ) タブが表示されたら、エラーがないか調べて閉じる

File List の .lpc ファイルが .ipx に置き換えられます。IPexpress のマニフェストファイル (.ipx) は Diamond で新しく導入されました。.ipx ファイルでは、複合モジュールに必要なファイルが常に追跡されます。

Lattice Diamond での IPexpress の使用

Lattice Diamond での IPexpress の使い方は、基本的に ispLEVER と同じです。

図 8-57 に示す Generation Options (生成オプション) タブを除いて、構成 GUI タブは全て同じです。

図 8-57. Generation Options (生成オプション) タブ

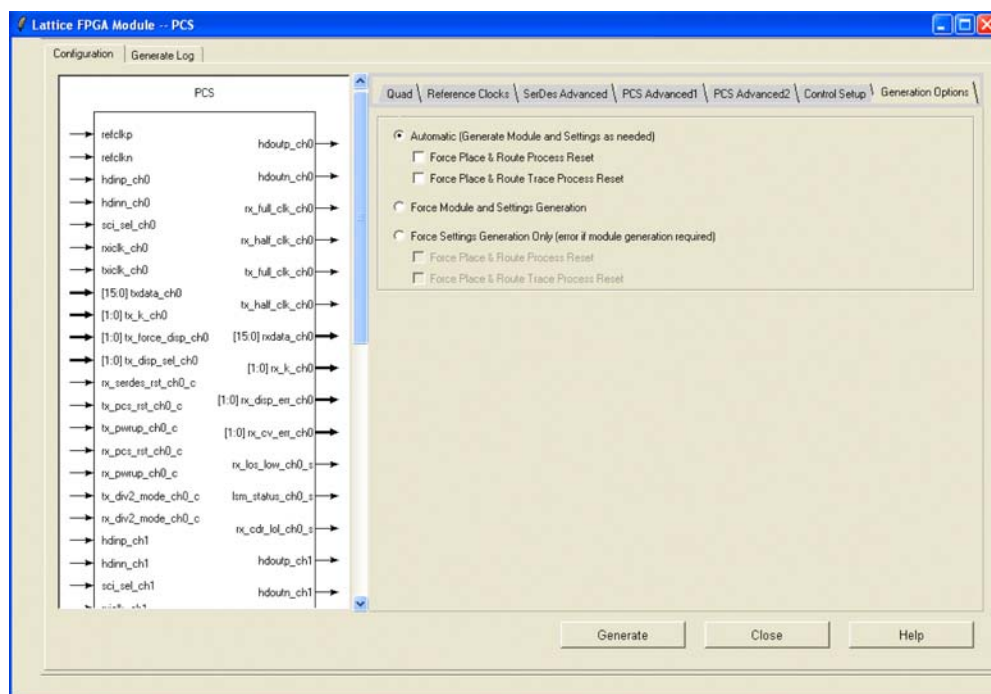


表 8-107. SERDES_PCS GUI 属性 ～ Generation Options (生成オプション) タブ

GUI テキスト	記 述
Automatic (自動)	必要に応じて HDL 及び構成 (.txt) ファイルを自動的に生成する。変更内容によっては、どちらのファイルも再生成が不要な場合がある。
Force Module and Settings Generation (モジュールと設定ファイルの生成を強制)	HDL ファイルと自動構成ファイルの両方を生成する。
Force Settings Generation Only (設定ファイルの生成のみ強制)	属性ファイルのみを生成する。HDL ファイルも生成が必要な場合はエラーメッセージが表示される。
Force Place & Route Process Reset (配置配線プロセスのリセットを強制)	配置配線設計プロセスをリセットし、新たに生成した PCS モジュールで強制的に再実行する。
Force Place & Route Trace Process Reset (配置配線とトレース・プロセスのリセットを強制)	配置配線とトレース (タイミング解析) プロセスをリセットし、新たに生成した PCS モジュールで強制的に再実行する。

注：

Automatic はデフォルトオプションとして設定されます。Automatic または Force Settings Generation Only が選択され、サブオプション Process Reset Options (プロセスのリセット・オプション) が全てオフで、HDL モジュールが生成されなかった場合、リセットポインタは自動的に Bitstream generation (ビットストリーム生成) に設定されます。

生成終了後、それに応じてプロセスウィンドウのリセットマークがリセットされます。

シミュレーション・ウィザードを使用した新規シミュレーション・プロジェクトの作成

このセクションでは、シミュレーション・ウィザードを使用してシミュレーション・プロジェクト (.spf) ファイルを作成し、それをスタンドアロン・シミュレータにインポートできるようにする方法について説明します。

1. プロジェクトナビゲータで、**Tools > Simulation Wizard** (シミュレーション・ウィザード) をクリックします。シミュレーション・ウィザードが表示されます。
2. Preparing the Simulator Interface (シミュレータ・インターフェイスの準備) ページで **Next** をクリックします。
3. Simulator Project Name (シミュレータ・プロジェクト名) ページで Project Name (プロジェクト名) テキストボックスにプロジェクト名を入力し、Project Location (プロジェクトの場所) テキストボックスと Browse (参照) ボタンを使用して、シミュレーション・プロジェクトの格納先ファイルパスに移動します。
4. このウィザードページでプロジェクト名を指定すると、選択したファイルパスに対応するフォルダが作成されます。新しいフォルダを作成するかどうかを確認するポップアップダイアログで **Yes** をクリックします。
5. Active-HDL® または ModelSim® シミュレータのどちらかのチェックボックスをクリックし、**Next** をクリックします。
6. Process Stage (プロセス・ステージ) ページで、作成するシミュレーション・プロジェクトのプロセス・ステージのタイプを選択します。有効なタイプは、RTL、Post-Synthesis Gate-Level、Post-Map Gate-Level、Post-Route Gate-level+Timing です。選択可能なプロセス・ステージのみがアクティブ表示されます。
7. プロジェクト内に複数のストラテジを定義した場合は、現在のストラテジに対して新しい選択を行うことができます。
8. このソフトウェアはプロジェクト実装ごとに複数のストラテジに対応しているため、ソースファイルの共通セットを異なる最適化オプションで実験できます。各ストラテジは異なるステージに処理されていることがあるため、このダイアログではロードするステージを指定できます。
9. Add Source (ソースの追加) ページで、Source Files (ソースファイル) リストボックスに表示されたソースファイルを選択するか、右の参照ボタンを使用して目的のソースファイルを選択します。作成したローカル・シミュレーション・プロジェクト・ディレクトリのソースファイルを残しておく場合は、**Copy Source to Simulation Directory** (ソースをシミュレーション・ディレクトリにコピーする) オプションをオンにします。
10. **Next** をクリックして Summary ページを表示し、選択したプロジェクトに関するシミュレーション・ライブラリなどの情報を指定します。デフォルトでは Run Simulator (シミュレータの実行) チェックボックスがオンになっていて、ウィザードの Simulator Project Name (シミュレータ・プロジェクト名) ページで選択したシミュレーション・ツールが起動されます。
11. **Finish** をクリック

ウィザードの実行後、シミュレーション・ウィザード・プロジェクト (.spf) ファイルと、シミュレーション・スクリプトの DO ファイルが生成されます。必要に応じて、DO ファイルは現在のプロジェクトにインポートできます。Active-HDL を使用した場合はウィザードで .ado ファイルが生成され、ModelSim を使用した場合は .mdo ファイルが作成されます。

注：ステップ 6 で SERDES/PCS 構成ファイル (.txt) を追加する必要があります。