

**CertusPro-NX**  
**PRBS7 Serdes Loopback**  
**リファレンスデザインユーザーマニュアル**

**macnica**

**June-2022**

# 免責事項

本ドキュメントに含まれる情報、及び本ドキュメントの対象であるリファレンスデザインの内容、動作、特性、品質に対して、マクニカはいかなる保証も行いません。

また、本ドキュメントに含まれる情報、及び本ドキュメントの対象であるリファレンスデザインは全て現状有姿にて提供され、これに対する改版や技術サポートのご依頼に関しては理由の如何を問わずお控え頂くようお願いしております。お客様ご用途における使用可否の判断、使用の際の動作確認、お客様製品への実装における適合性や安全性の確認、法的要件の確認はお客様にて実施頂きますようお願いいたします。これらに対してもマクニカは一切の責任を負うことが難しく、いかなる保証もいたしかねます。また、本ドキュメントの情報、及びドキュメントの対象であるリファレンスデザインはマクニカの所有物であり、予告なしに変更を加えることがございますので予めご了承ください。

# Table of Contents

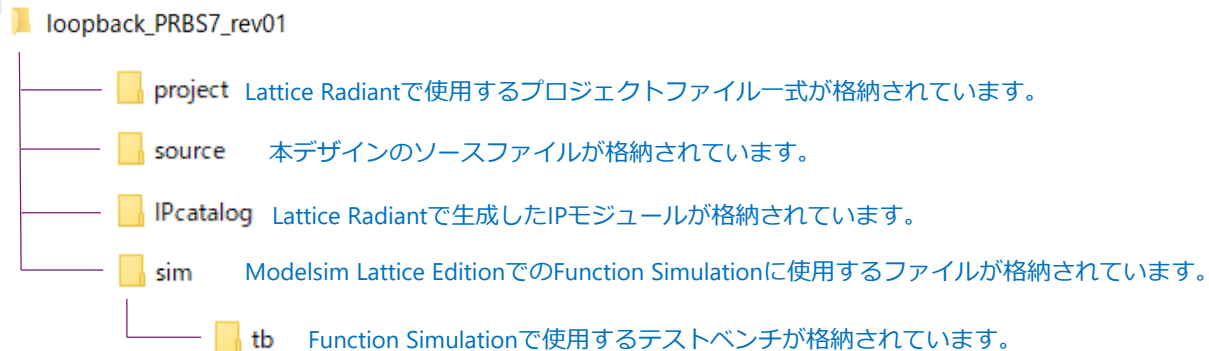
1. デザイン概要
2. デザインフォルダ構成
3. 回路ブロック図
4. デザインポート説明
5. 各モジュール概要
6. ファンクションシミュレーション

# 1. デザイン概要

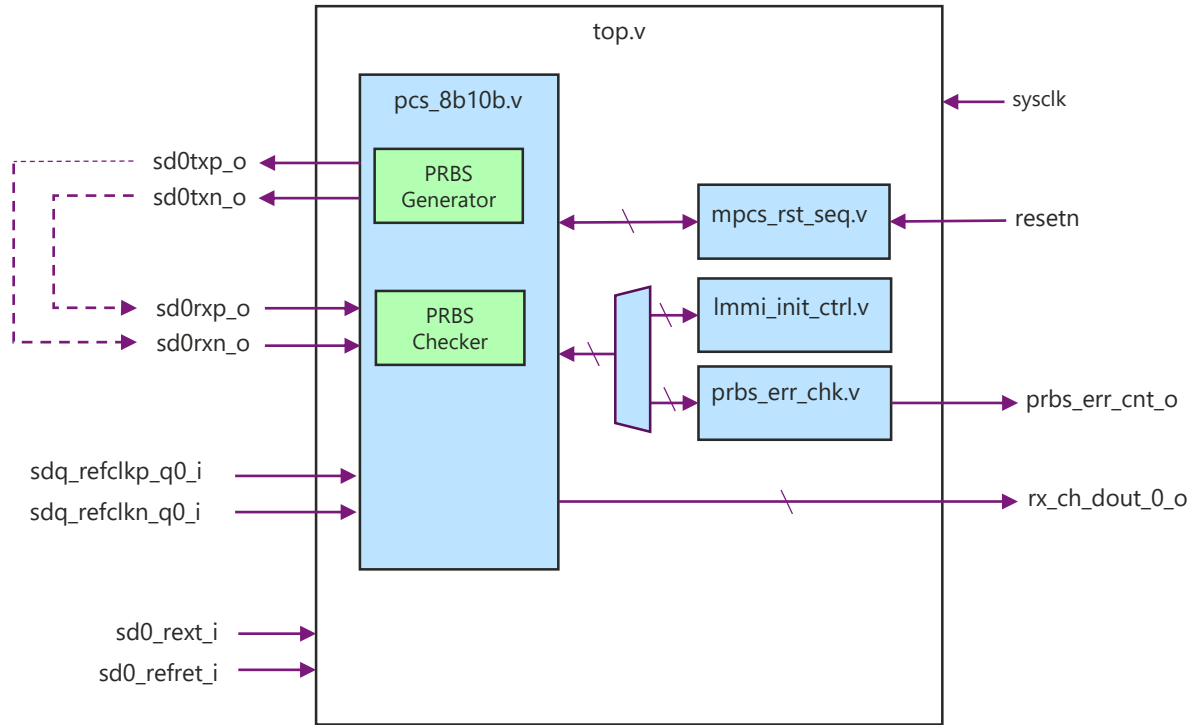
- 本デザインはPRBS Generator/Checkerを使用したCertusPro-NXのSerdesループバックリファレンスデザインです。
- Serdesブロック内のPMA Controllerに内蔵されたPRBS GeneratorからPRBS7のデータを出力してループバックし、同じく内蔵のPRBS Checkerを用いてビットエラーの検出を行います。  
PMA Controllerに内蔵のPRBS Generator/Checkerについては以下の資料を参照してください。  
[https://www.latticesemi.com/view\\_document?document\\_id=53257](https://www.latticesemi.com/view_document?document_id=53257)
- MPCSRリファレンスクロック周波数は156.25MHz、シリアルデータレートは1.5625Gbpsに設定されています。また、リファレンスクロック入力はQuad内チャネル共用の専用クロックSDQ\_REFCLKを使用しています。  
リファレンスクロックの詳細は以下のテクニカルノートを参照してください。  
[https://www.latticesemi.com/view\\_document?document\\_id=53257](https://www.latticesemi.com/view_document?document_id=53257)
- デザイン動作はModelSim Lattice Editionでのファンクションシミュレーションにて確認しています。実機上での確認はしておりません。
- 本デザインはRadiant3.1.1でコンパイルされており、論理合成ツールはSynplify Proを使用しています。
- MPCSR IPはVersion 1.3.0を使用しています。

## 2. デザインフォルダ構成

本デザインのフォルダ構成を以下に示します。



### 3. 回路ブロック図



## 4. デザインポート説明

ポート名	入出力方向	説明
sysclk	入力	システムクロック入力。本デザインではリセットシーケンサやMPCSの制御クロックとして使用
resetn	入力	リセット入力 (Active-Low)
sdq0_refclkp_i	入力	MPCSリファレンスクロック (ポジティブ)
sdq0_refclk_n_i	入力	MPCSリファレンスクロック (ネガティブ)
sd0rxp_i	入力	Serdesシリアルデータ入力 (ポジティブ)
sd0rxn_i	入力	Serdesシリアルデータ入力 (ネガティブ)
sd0_rext_i	入力	PMA PLL用のアナログリファレンスリターン信号への外部抵抗接続信号です。基板上で処理を行います
sd0_refret_i	入力	PMA PLL用のアナログリファレンスリターン信号です。基板上で処理を行います
sd0txp_o	出力	Serdesシリアルデータ出力 (ポジティブ)
sd0txn_o	出力	Serdesシリアルデータ出力 (ネガティブ)
rx_ch_dout_0_o	出力	MPCS出力データ。本デザインでは特に意味のあるデータ出力ではありません
prbs_err_cnt_o	出力	PRBS Checkerのエラーカウント値出力

# 5. 各モジュール概要

## ■ top.v

本デザインのトップモジュールです。

## ■ mpcs\_rst\_gen.v

MPCS (pcs\_8b10b.v) が要求するリセットシーケンス生成モジュールです。resetn信号をトリガとして以下のシーケンスを生成しています。

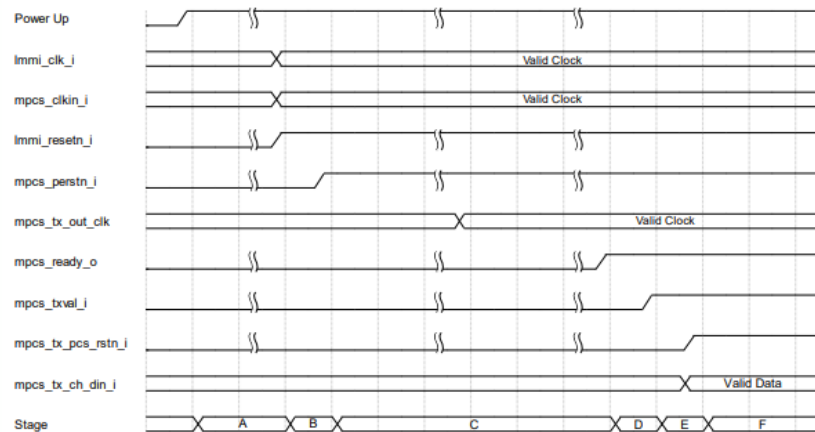


Figure 7.18. MPCS Mode Reset Sequence (Tx Path)

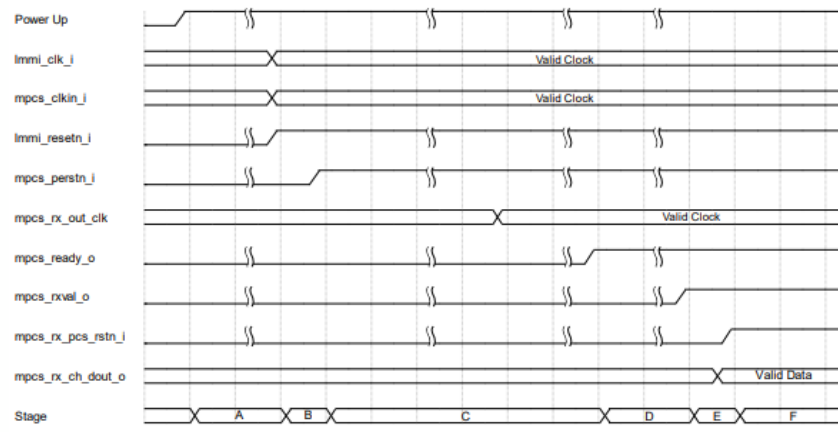


Figure 7.19. MPCS Mode Reset Sequence (Rx Path)



# 5. 各モジュール概要

## ■ lmmi\_init\_ctrl.v

LMMI (Lattice Memory Mapped Interface) を介したPMA/PCSのレジスタ初期設定用のモジュールです。PMA Controller内のPRBS Generator/Checkerを有効化するレジスタを操作するために使用しています。LMMIやレジスタマップについての詳細は次のテクニカルノートを参照してください。

[https://www.latticesemi.com/view\\_document?document\\_id=53257](https://www.latticesemi.com/view_document?document_id=53257)

本モジュールをカスタマイズして使用する場合のソース改版例は以下を参照してください。

```
//-----↓
// PMA/PCS register access sequence↓
//-----↓

always@(posedge lmmi_clk_i or negedge lmmi_resetsn_i)begin
  if(!lmmi_resetsn_i)begin
    pc_ff <= {CNT_WIDTH{1'b0}};
    request_ff <= 1'b0;
    wr_rdn_ff <= 1'b0;
    offset_ff <= 9'h000;
    wdata_ff <= 8'h00;
    lmmi_active <= 1'b0;
    init_done <= 1'b0;
  end
  else begin
    if(lmmi_ready_i && lmmi_state == IDLE)begin
      if(pc_ff == {CNT_WIDTH{1'b1}})begin
        pc_ff <= pc_ff;
      end
      else if(pc_ff == 0) begin
        lmmi_idle ();
      end
      //-----↓
      else if(pc_ff == 1)begin // 1st access↓
        if(lmmi_active)begin
          lmmi_idle ();
        end
        else begin
          lmmi_write ( PMA, 8'h64, 8'h41 ); // PMA Reg 64 : PRBS Generator/Checker
          // 8'h41(8'b01000001)
        end
      end
      //-----↓
      else if(pc_ff == 2)begin // 2nd access↓
        if(lmmi_active)begin
          lmmi_idle ();
        end
        else begin
          lmmi_read ( PMA, 8'h64 ); // PMA Reg 64 : PRBS Co
        end
      end
      //-----↓
      :
    end
  end
end
```

“PMA/PCS register access sequence”のコメント以下のalways文でこの部分より下を改版します。

アクセス順に沿ってpc\_ffの条件式に1から順に設定します。

この構文は変えず  
コピー&ペースト。

レジスタへのライトはlmmi\_write(), リードはlmmi\_read()のタスクを使用します。構文は以下の通りです。  
lmmi\_write (レジスタ領域, アドレス, ライトデータ)  
lmmi\_read (レジスタ領域, アドレス)  
レジスタ領域はPMAレジスタ、PCSレジスタに分かれており、PMAレジスタへのアクセスの場合は“PMA”、PCSレジスタへのアクセスの場合は“PCS”と記載します。ライトデータはlmmi\_wdata\_oから出力されます。リードデータはlmmi\_rdata\_valid\_oがHighのタイミングでlmmi\_rdata\_oから出力されます。

# 5. 各モジュール概要

## ■ prbs\_err\_chk.v

LMMIを介してPMA Controller内のPRBS Checkerに定期的にアクセスし、エラーカウントを読み出すモジュールです。resetn解除後はPMA Controller内のPRBS Generator/Checkerを有効化するためにlmmi\_init\_ctrl.vがLMMIを占有してレジスタの初期設定を行いますが、初期設定が完了するとprbs\_err\_chk.vにLMMIへのアクセスを引き渡されます。その後、MPCSのReady信号（mpcs\_ready\_i）がHighになるのを待った後に動作を開始し、エラーカウントを定期的に読み出します。

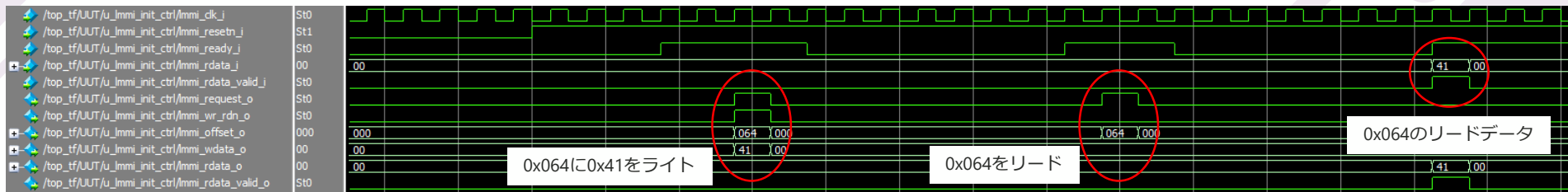
## ■ pcs\_8b10b.v

156.25MHzリファレンスクロックから1.5625Gbpsのシリアルデータを生成するMPCSモジュールです。プロトコルとしてGeneric 8B10Bが選択されていますが、実際の送信データはPMA Controllerに内蔵されたPRBS Generatorで生成されたPRBS7のデータのため、8B10B符号化の機能は使用しておらず、Serdesブロックの実装のためだけに用意されています。



# 6. ファンクションシミュレーション

## ■リセット解除～PRBS7 Generator/Checker有効化



resetsn解除後にmpcs\_rst\_gen.vによって生成されたlmmi\_resetn\_igがHighになると、lmmi\_init\_ctrl.vが動作を開始します。

まずPMA Controller内のアドレス (offset) 0x064に0x41をライトし、PRBS7 Generator/Checkerを有効化します。レジスタに正しくライトされたことを確認するため、アドレス0x064に対してリードアクセスを行い、0x41がリードされることを確認しています。アドレス0x064の詳細については以下のテクニカルノート抜粋を参照してください。最上位bitはPMAレジスタ領域 or MPCSレジスタ領域かの選択で、0x064はPMAレジスタ領域のreg64となります。

Table A. 20. PRBS Control Register [reg64]

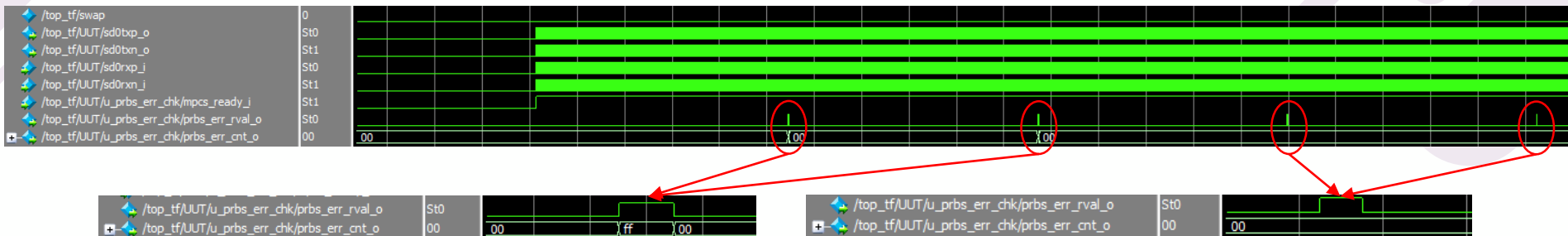
Field	Name	Access	Width	Reset	Description
[7]	reserved	RSVD	1	1'b0	—
[6]	prbs_chk	RW	1	1'b0	PRBS pattern checker control. <ul style="list-style-type: none"><li>• 1'b1 – enabled.</li><li>• 1'b0 – disabled.</li></ul>
[5:4]	reserved	RSVD	2	2'b00	Internal usage.
[3:2]	prbs_typ	RW	2	2'b00	Defines the type of PRBS pattern which is applied. <ul style="list-style-type: none"><li>• 2'b11 – PRBS31</li><li>• 2'b10 – PRBS23</li><li>• 2'b01 – PRBS11</li><li>• 2'b00 – PRBS7</li></ul>
[1]	lpbk_en	RW	1	1'b0	Near-End loopback (serial loopback from Tx back to Rx) control. <ul style="list-style-type: none"><li>• 1'b1 – the PMA is put in Near-End loopback.</li><li>• 1'b0 – the PMA is not put in Near-End loopback.</li></ul>
[0]	prbs_gen	RW	1	1'b0	PRBS pattern transmission control. <ul style="list-style-type: none"><li>• 1'b1 – starts the PRBS pattern transmission.</li><li>• 1'b0 – not start the PRBS pattern transmission.</li></ul>

Note: This register can be reprogrammed any time but has functional impact on the functionality as it can configure the SerDes in loopback or generate the PRBS pattern.

- 8-bit width of write and read data.
- 9-bit address (offset), the highest bit is used to select register space.
  - offset[8] == 1'b1, MPCS register space
  - offset[8] == 1'b0, PMA register space

# 6. ファンクションシミュレーション

## ■ MPCs Ready ~ CDR Lock



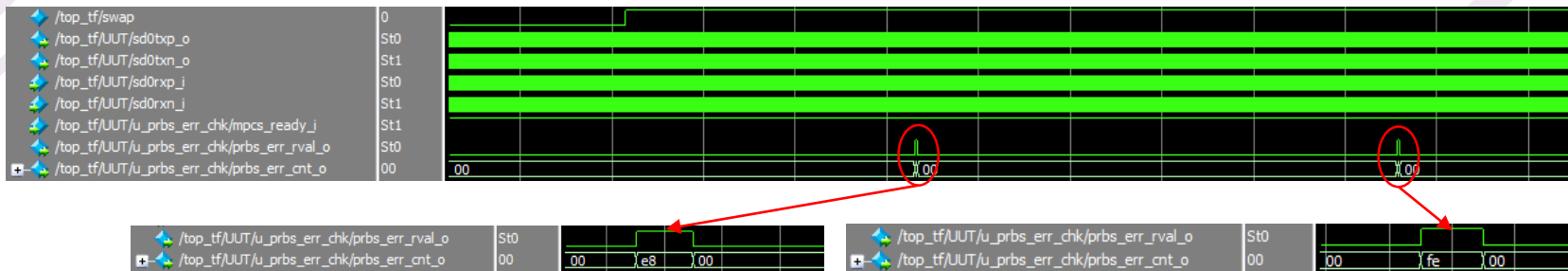
MPCsがReady状態になった後、prbs\_err\_chk.vがPRBS Checkerのエラーカウント値の定期的な読み出しを開始します。読み出し開始後しばらくはCDRがロックしていない状態が続き、リードデータとしてCDRアンロックを示す0xFFを示します。PRBS Checkerのエラーカウント最大値は0xFEとなっており、0xFFはCDRアンロックを示す特別なコードとして用意されています。CDRがロックした後はエラーカウント値は0x00を示しています。

Table A. 21. PRBS Error Counter Register [reg65]

Field	Name	Access	Width	Reset	Description
[7:0]	prbs_errcnt	RO	8	8'h00	Reports the number of PRBS error detected when the PRBS test is applied. This register is automatically cleared when the prbs_chk bit gets cleared. The PRBS error counter saturates at 254 errors, the 255-count value corresponding to an error code where the CDR PLL is not locked to incoming data. In case of such error code detected, the PRBS test must either wait for longer time for CDR PLL to synchronize on input data before enabling PRBS checker, or simply times out reporting that no data is received at all. Note that the PRBS error counter logic is also with count error, when the PRBS invariant (all zero value) is obtained, considering input data as error data.

# 6. ファンクションシミュレーション

## ■ Bit Error Insertion



本デザインの実ベンチでは、ビットエラーの確認のためにループバックシリアルデータの差動極性をswapという信号で反転させています。swap信号がLowからHighに遷移し、シリアルデータの差動極性が反転するとPRBS Checkerがビットエラーのカウントを開始します。上図では極性反転後の一度目の読み出しでエラーカウント0xE8がリードされ、次の読み出しで最大値の0xFEがリードされていることを示しています。

# Revision History

Date	Revision	Page	Change Information
2022/07/14	1.0		First Revision