

CrossLink-NX
GDDR4 GDDR5
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
7. 実機動作

1. デザイン概要

- 本デザインはCrossLink-NXのI/O部に搭載された8:1, 10:1用シリアライズ/デシリアライズロジックGDDR4 (8:1) 及び、GDDR5 (10:1)を使用したループバックリファレンスデザインです。
- 内部RAMに保持された8bit (GDDR4向け) または10bit (GDDR5向け) のデータパターンを4レーンに分配後にシリアライズして送信し、そのデータを同様にGDDR4またはGDDR5で受信してデシリアライズを行います。
- 受信したデータは送信側と同じデータパターンを用いて照合し、正しくデータが受信されているかチェックします。
- 送信データレートは800Mbps (400MHzソースシンクロナスクロックを並走) に設定されています。
- GDDR4, GDDR5のアーキテクチャ含む詳細は以下の資料を参照してください。
https://www.latticesemi.com/view_document?document_id=52783
http://www.latticesemi.com/view_document?document_id=52454
- 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いません。
- 本デザインはRadiant3.2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。
- 回路構成はGDDR4ループバックデザインとGDDR5ループバックデザインで差分がありません。デザインポート説明、各モジュール概要については両方のデザインに対して共通の内容となります。

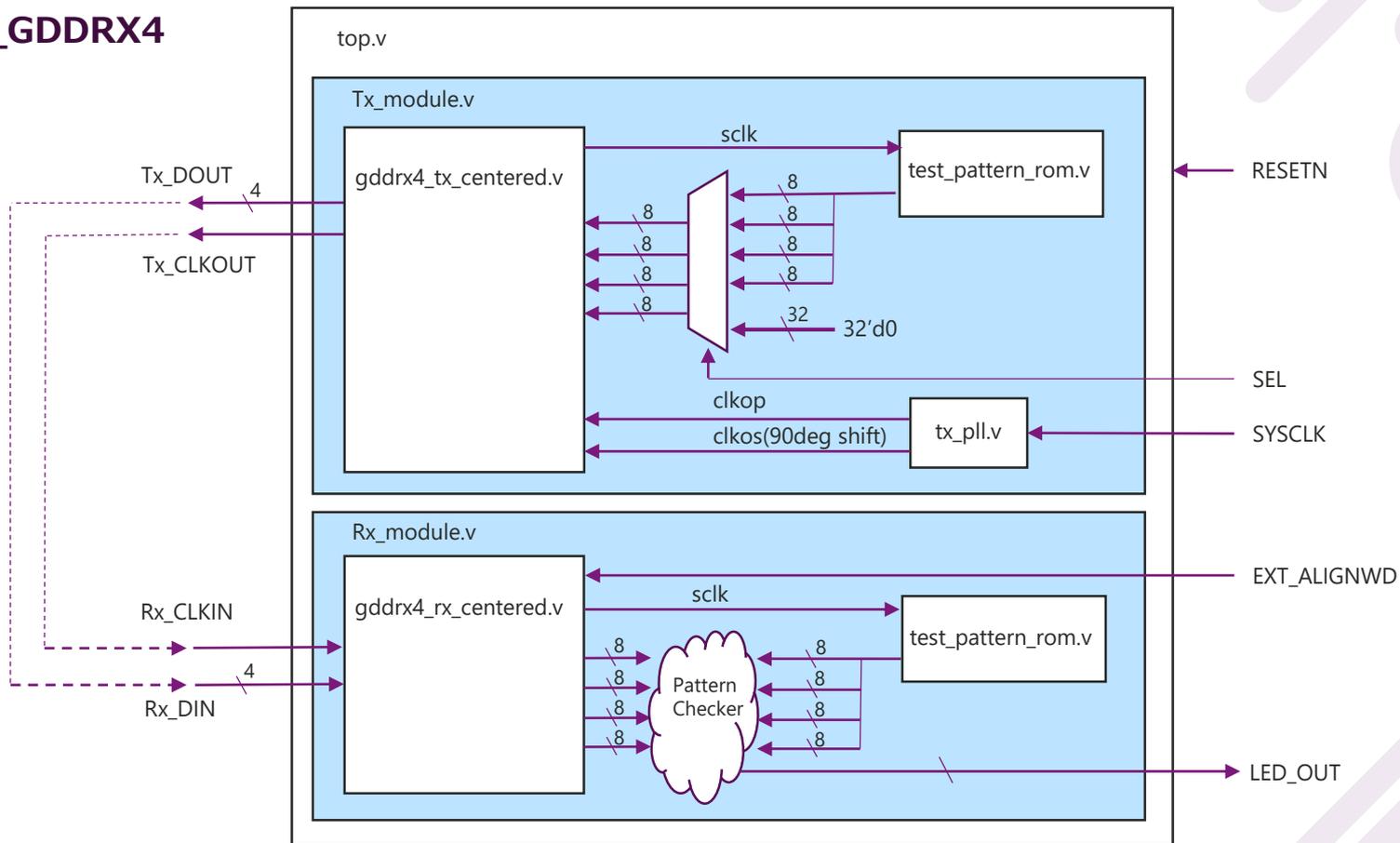
2. デザインフォルダ構成

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

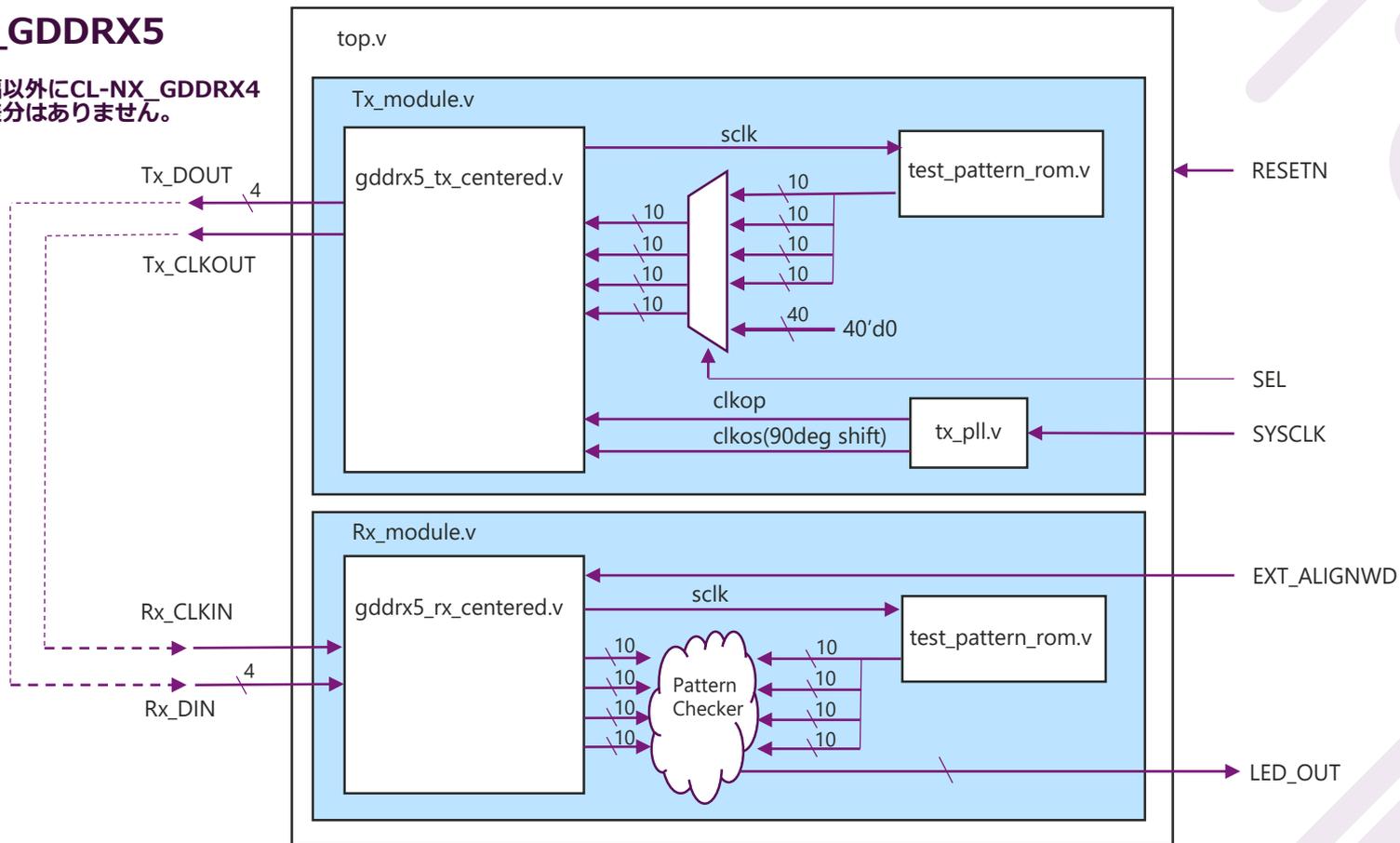
CL-NX_GDDR4



3. 回路ブロック図

CL-NX_GDDR5

※内部バス幅以外にCL-NX_GDDR4と構成の差分はありません。



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
RESETN	入力	リセット入力 (Active-Low)
SEL	入力	出力テストパターン選択信号 (High : 内部RAMパターン出力、Low : All 0出力)
EXT_ALIGNWD	入力	受信データ手動ワードアライメント用入力
RX_DIN	入力	シリアルデータ入力
RX_CLKIN	入力	シリアルデータ受信用ソースシンクロナスクロック入力
TX_DOUT	入力	シリアルデータ出力
TX_DIN	出力	シリアルデータ送信用ソースシンクロナスクロック出力
LED_OUT	出力	送信パターンと受信パターンの一致検出信号

5. 各モジュール概要

■ top.v

本デザインのエントリポイントモジュールです。

■ Tx_module.v

test_pattern_rom.vから出力されるテストパターンをGDDR4（またはGDDR5）を介してシリアル化して出力します。シリアル化用のクロックを生成するtx_pll.vもこのモジュール内に存在します。簡易的な送信パターンエラー挿入用途として、出力パターンをtest_pattern_rom.vのデータとするか、ALL 0とするか切り替えるセレクタも有しています。

■ test_pattern_rom.v

送信パターンを初期値として持つ内部RAMモジュールです。送信パターンは512wordのデータとなっており、GDDR4では8bit、GDDR5では10bitのデータを1クロック毎に出力します。512wordの先頭パターンがGDDR4用は0xAA、GDDR5用は0x2AAとなっており、これらのデータは512wordの中で唯一先頭ワードにのみ出現する仕様になっています。受信回路側ではこの先頭ワードを検出することでワードアライメント完了を検出することができます。

■ tx_pll.v

GDDR4（またはGDDR5）によるシリアル化に使用する高速クロック生成用PLLモジュールです。50MHzのクロックから400MHzクロック2本を生成しています。2本のうち1本は送信クロック出力用に90°位相シフトしています。

■ gddrx4_tx_centered.v (gddrx5_tx_centered.v)

gddrx4_tx_centered.vは8:1シリアル化、gddrx5_tx_centered.vは10:1シリアル化を行うモジュールです。送信クロックエッジはデータの中心位置に来るように設定されています。設定の詳細は次ページを参照してください。

5. 各モジュール概要

Property	Value
General	
Interface Type	Transmit
I/O Standard for this Interface	LVDS
Gearing Ratio	X4
Bus Width for this Interface [1 - 256]	4
Clock to Data Relations on the Pins	Centered
Interface	GDDR4_TX_ECLK_Centered
Data Path Delay	Bypass
Fine Delay Value for User Defined [0 - 126]	0
Coarse Delay Value for User Defined	0NS
Clock Path Delay	Fixed
Include GDDR SYNC	<input checked="" type="checkbox"/>
Enable Tri-State Control	<input type="checkbox"/>
Clock Frequency for this Interface (MHz) [100 - 750]	400
Bandwidth for this Interface(Mbits/s)	3200
Enable PLL Instantiation	<input type="checkbox"/>
PLL Input Clock Frequency (MHz) [10 - 400]	25
PLL Output Clock Frequency Actual Value (MHz) [100 - 750]	150
Reference Clock from I/O Pin	<input type="checkbox"/>
I/O Standard for Reference Clock	LVDS
PLL Output Clock Tolerance (%)	0.0

No DRC issues are found.

Generate Cancel

■ Interface Type

Transmit (送信)を選択しています。

■ I/O Standard for this interface

I/Oスタンダードの選択です。LVDSを選択しています。

■ Gearing Ratio

gddrx4_tx_centered.vではX4 (8:1シリアライズ)、gddrx5_tx_centered.vではX5 (10:1シリアライズ)を選択しています。

■ Data Path Delay

データ出力への遅延付与設定です。本デザインでは遅延付与せず、Bypassを設定しています。

■ Clock Frequency for this Interface (MHz)

送信クロック周波数の設定です。Gearing Ratioに応じてこの周波数から分周されたクロックが出力データ同期用のソースクロック (sclk_o) として出力されます。本デザインでは400MHzと設定しており、ソースクロックはX4の場合は1/4の100MHz, X5の場合は1/5の80MHzとなります。シリアルデータはDDR出力のため、400MHzの倍の800Mbpsとなります。

■ Enable PLL Instantiation

シリアライズ用の高速クロックを生成するPLLを本モジュールに組み込んだ状態でGDDRモジュールを生成するかどうかの設定です。チェックを外しており、このモジュールとは別に実装する方法を取っています (tx_pll.vを別途実装しています)。

5. 各モジュール概要

■ Rx_module

送信側から送られてきたシリアルデータをGDDR4（またはGDDR5）で受信し、デシリアライズしてファブリックへと出力し、受信データのワードアライメントとデータパターンチェックを行います。受信データのワードアライメントはgddrx4_rx_centered.v（またはgddrx_rx_centered.v）に用意されているalignwdポートをトグルして行います。テストパターンに含まれるユニークコードの0xAA（GDDR4向け）、または0x2AA（GDDR5向け）を検出するまでalignwdポートを内部ロジックから定期的にトグルします。ユニークコードを検出すると、本モジュール内に実装されたtest_pattern_rom.vからテストパターンの出力と比較を開始し、一致し続けている間LED_OUTをLowにドライブします。ユニークコードの検出ができなくなると、再度alignwdポートを定期的にトグルし、ワードアライメントを取り直す動作を行います。

■ gddrx4_rx_centered.v (gddrx5_rx_centered.v)

gddrx4_rx_centered.vは1:8デシリアライズ、gddrx5_rx_centered.vは1:10デシリアライズを行うモジュールです。ソースシンクロナスクロックのエッジはデータの中心に位置する設定になっています。設定の詳細は次ページを参照してください。

5. 各モジュール概要

Module/IP Block Wizard

Configure Component from Module gddr Version 1.5.0
Set the following parameters to configure this component.

Diagram gddrx4_rx_centered

Configure IP

Property	Value
General	
Interface Type	Receive
I/O Standard for this Interface	LVDS
Gearing Ratio	X4
Bus Width for this Interface [1 - 256]	4
Clock to Data Relations on the Pins	Centered
Interface	GDDRX4_RX_ECLK_Centered
Data Path Delay	Bypass
Fine Delay Value for User Defined [0 - 126]	0
Coarse Delay Value for User Defined	0NS
Clock Path Delay	Fixed
Include GDDR SYNC	<input checked="" type="checkbox"/>
Enable Tri-State Control	<input type="checkbox"/>
Clock Frequency for this Interface (MHz) [100 - 750]	400
Bandwidth for this Interface(Mbits/s)	3200
Enable PLL Instantiation	<input type="checkbox"/>
PLL Input Clock Frequency (MHz) [10 - 400]	25
PLL Output Clock Frequency Actual Value (MHz) [100 - 750]	150
Reference Clock from I/O Pin	<input type="checkbox"/>
I/O Standard for Reference Clock	LVDS
PLL Output Clock Tolerance (%)	0.0

No DRG issues are found.

Generate Cancel

■ **Interface Type**
Receive(受信)を選択しています。

■ **I/O Standard for this interface**
I/Oスタンダードの選択です。LVDSを選択しています。

■ **Gearing Ratio**
gddrx4_tx_centered.vではX4 (1:8デシリアライズ)、
gddrx5_tx_centered.vではX5 (1:10デシリアライズ) を選択しています。

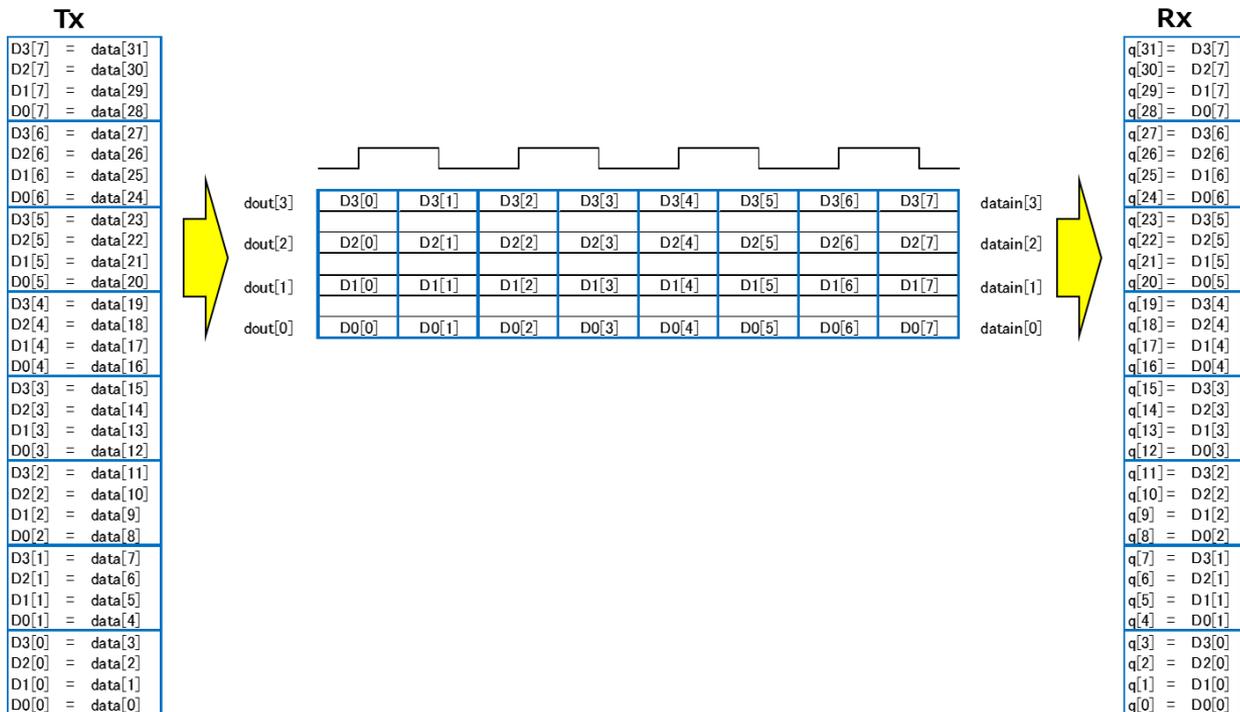
■ **Data Path Delay**
データ出力への遅延付与設定です。本デザインでは遅延付与せず、Bypassを設定しています。

■ **Include GDDR SYNC**
GDDR Rxモジュールの同期化ロジックをモジュール内に内蔵するかどうかの設定です。内蔵する設定にしています。

■ **Clock Frequency for this Interface (MHz)**
受信クロック周波数の設定です。Gearing Ratioに応じてこの周波数から分周されたクロックが受信データと同期したソースクロック (sclk_o) として出力されます。本デザインでは400MHzと設定しており、ソースクロックはX4の場合は1/4の100MHz, X5の場合は1/5の80MHzとなります。シリアルデータはDDR出力のため、400MHzの倍の800Mbpsとなります。

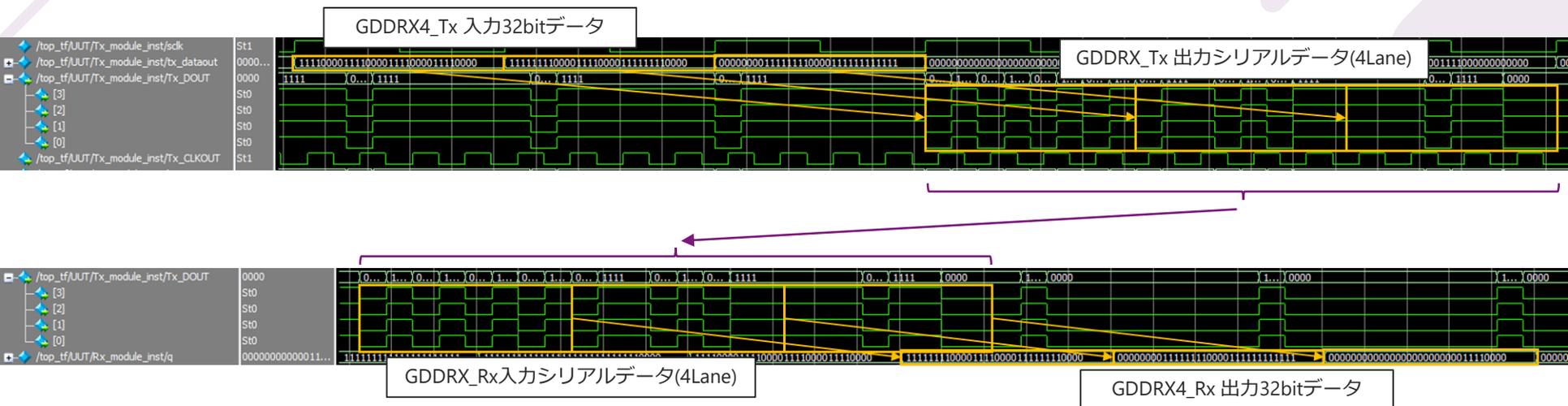
(参考) GDDR送受信データ並び

CrossLink-NXのGDDR Tx/Rx入出力データ並びは、レーン毎に纏まって入出力されるのではなくビット順毎に纏まっています。例えばGDDR4で4レーン出力を行う場合、レーン3がbit[31:24]、レーン2がbit[23:16]...というような形ではなく、bit[31:28]が各レーンの7bit目、bit[27:23]が各レーンの6bit目...という形で割り当てられます。



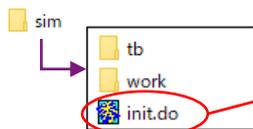
(参考) GDDR送受信データ並び

以下は本デザイン (CL-NX_GDDR4) のシミュレーション波形から抜粋した送受信データの抜粋です。



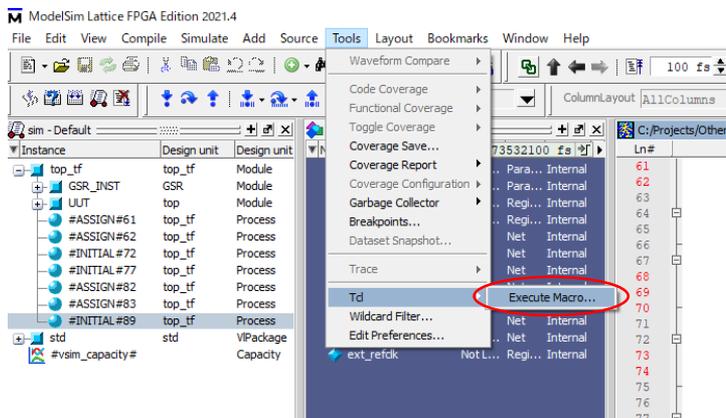
6. ファンクションシミュレーション

ファンクションシミュレーションを実施するには、ModelSim Lattice Editionを使用します。
シミュレーション開始前にinit.doファイル内の以下のディレクトリ指定をユーザー環境のsimフォルダのパスに変更する必要があります。
Init.doファイルはsimフォルダの中に格納されています。



```
6 |#-----↓
7 |# Directry definition↓
8 |#-----↓
9 |set WORK_DIR "C:/Projects/Others/CrossLink-NX_loopback/CL-NX_GDDR4/sim"↓
10 |set SRC $WORK_DIR/./source↓
11 |set IPcat $WORK_DIR/./IPcatalog↓
12 |set TB $WORK_DIR/tb↓
```

Model Simを起動後、Tools > Tcl > Execute Macro からinit.doファイルを起動します。

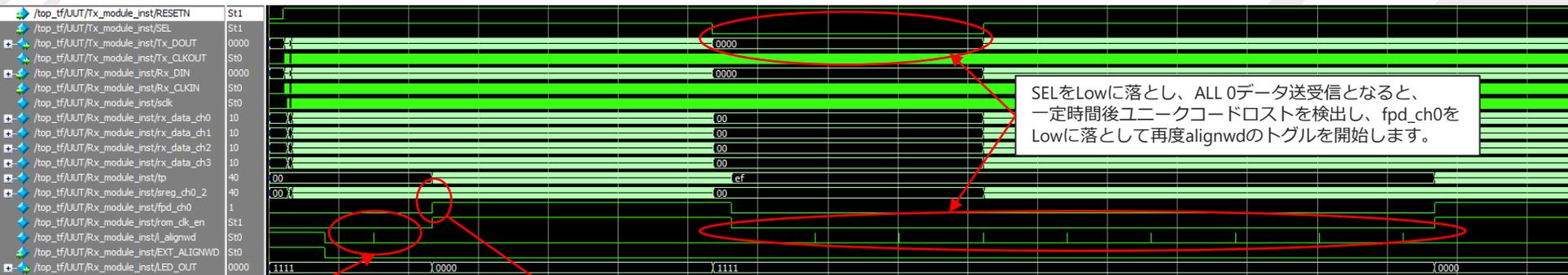


Init.doの編集方法、doファイルでのシミュレーション詳細については、以下のページの「ModelSim Lattice Edition DO マクロ ユーザーガイド」を参照してください。

<https://www.macnica.co.jp/business/semiconductor/articles/lattice/132003/>

6. ファンクションシミュレーション

CL-NX_GDDR4



ユニークコード検出前は、alignwdをトグルし、ユニークコードが見つかるまでデータ並びをシフトします。(本シミュレーションでは1回のトグルでユニークコードの検出ができています。)

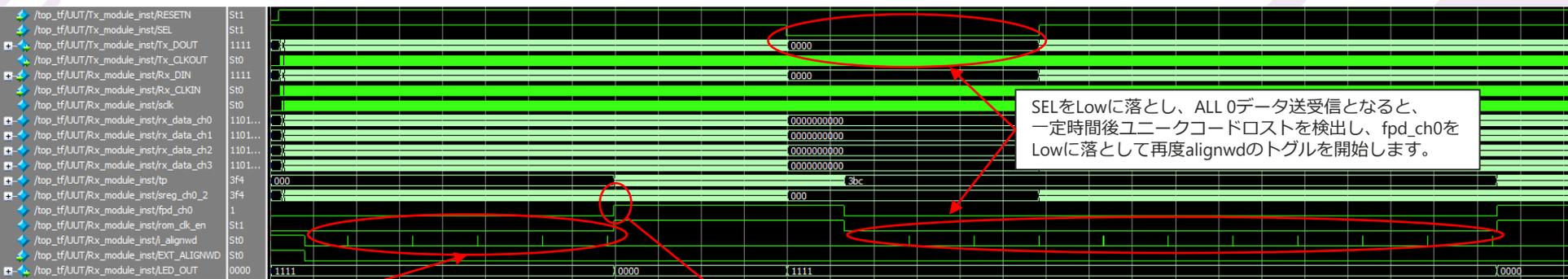
+/top_tf/UUT/Rx_module_inst/tp	40	00			aa	d6	37	02
+/top_tf/UUT/Rx_module_inst/sreg_ch2_2	40	fb	fd	fe	aa	d6	37	02
/top_tf/UUT/Rx_module_inst/fpd_ch0	1							

ユニークコード検出後 (fpd_ch0 = High後) 受信データとテストパターン (tp) は一致しています。

上図はCL-NX_GDDR4デザインのシミュレーション波形全体像です。送信側からデータ出力が始まり、ループバックしたデータを受信し始めた後、Rx_module内でワードアライメントのためにalignwdのトグルを開始します。ユニークコードが検出されて受信データとテストパターンが一致するとLED_OUTがAll 0になります。その一定時間後にSEL入力をLowに落としてALL 0パターンを送信側から出力し、ユニークコード検出をロストさせています。すると再度ユニークコード検出のためにalignwdのトグルが開始され、SELをHighに戻した後はしばらくするとアラインされたデータからユニークコードを再度検出し、LED_OUTが再びAll 0となります。

6. ファンクションシミュレーション

CL-NX_GDDR5



ユニークコード検出前は、alignwdをトグルし、ユニークコードが見つかるまでデータ並びをシフトします。

<code>/top_tf/UUT/Rx_module_inst/tp</code>	3f4	000			2aa	002	004	008	
<code>/top_tf/UUT/Rx_module_inst/sreg_ch0_2</code>	3f4	3ec		3f4	3f8	2aa	002	004	008
<code>/top_tf/UUT/Rx_module_inst/fpd_ch0</code>	1								

ユニークコード検出後 (fpd_ch0 = High後) 受信データとテストパターン (tp) は一致しています。

上図はCL-NX_GDDR5デザインのシミュレーション波形全体像です。動作はCL-NX_GDDR4と同様になります。

Revision History

Date	Revision	Page	Change Information
2022/07/22	1.0		First Revision