

PCI-Express ハード IP を使用した DMA 転送の実現  
for Cyclone V GT FPGA 開発キット  
(ハードウェア編)

ver.1

2015 年 4 月 Rev.3

# PCI-Express ハード IP を使用した DMA 転送の実現 for Cyclone V GT FPGA 開発キット (ハードウェア編)

## 目次

1.	はじめに .....	4
1-1.	概要 .....	4
1-2.	対象デバイス・ボード .....	4
1-3.	環境 .....	4
1-4.	対象デザイン .....	4
2.	ハードウェア概要 .....	5
2-1.	ハードウェア構成 .....	5
2-2.	ハードウェア・デザイン構成 .....	5
2-3.	PCI-Express のハード IP .....	7
2-3-1.	System Setting .....	8
2-3-2.	PCI Base Address Registers .....	8
2-3-3.	Device Identification Registers .....	9
2-3-4.	Link Capabilities .....	9
2-3-5.	Error Reporting .....	9
2-3-6.	Avalon-MM System Settings .....	10
2-3-7.	Address Translation .....	11
2-4.	mSGDMA .....	12
2-5.	DDR3 SDRAM コントローラ .....	13
2-6.	オンチップ・メモリ .....	14
2-7.	XCVR リコンフィギュレーション .....	15
2-8.	クロック .....	16
3.	PCI-Express Avalon-MM ブリッジ .....	17
3-1.	PCI-Express ⇒ Avalon-MM アドレス変換テーブル .....	17
3-2.	Avalon-MM ⇒ PCI-Express アドレス変換テーブル .....	18
3-3.	割り込み .....	19

**PCI-Express ハード IP を使用した DMA 転送の実現  
for Cyclone V GT FPGA 開発キット  
(ハードウェア編)**

---

4.	SignalTap II を使用した実機動作確認.....	20
4-1.	Qsys 内のアドレス設定.....	20
4-2.	SignalTap II の波形設定.....	22
4-3.	オンチップ・メモリへのターゲット・ライト.....	24
4-4.	外部 DDR3 SDRAM へのターゲット・リード.....	26
4-5.	外部 DDR3 SDRAM への DMA リード.....	28
5.	まとめ.....	35
	改版履歴.....	36

## 1. はじめに

### 1-1. 概要

以前にアルテラ社 PCI-Express ハード IP を使用した Qsys システム統合ツール(以下、Qsys)での DMA 転送の実現についての記事を書きました。今回はその続編となるハードウェア編となります。今回は、Qsys での PCI-Express ハード IP リファレンス・デザインの紹介から始まり、弊社実機での動作確認までの一連の操作方法について触れました。今回は、Qsys 上への PCI-Express<sup>®</sup>、メモリ・コントローラ、DMA エンジン等の各種ペリフェラルの実装方法、また、PC 上で動作する転送レート計測用ソフトウェアと SignalTap<sup>®</sup> II を使用したターゲット・リード/ライト、DMA 転送時の詳細なメカニズムについて解説します。

今回のハードウェア編を通して、Qsys での PCI-Express ハード IP を使用したデザイン実装のノウハウを理解し、ユーザ様独自のシステム構築の助けになることを目的とします。

### 1-2. 対象デバイス・ボード

今回対象としたアルテラ社 FPGA デバイスとボードの情報は以下となります。

Cyclone<sup>®</sup> V GT FPGA 開発キット

PCI-Express Gen2 x1, x2, x4 に対応、最大ペイロード・サイズ 512Bytes

[https://www.altera.co.jp/products/boards\\_and\\_kits/dev-kits/altera/kit-cyclone-v-gt.html](https://www.altera.co.jp/products/boards_and_kits/dev-kits/altera/kit-cyclone-v-gt.html)

### 1-3. 環境

PC: ASUS P8Z77-M PRO

チップセット: インテル社製 Z77 Express

OS: Windows<sup>®</sup> 7 64bit

ドライバ: Jungo 社 WinDriver<sup>™</sup> (<http://www.jungo.com/>) v10.21

ソフトウェア: アルテラ社 AN 431 対応転送レート計測用ソフトウェア

FPGA 開発ツール: Quartus<sup>®</sup> II 開発ソフトウェア 14.0

### 1-4. 対象デザイン

Altera Wiki の以下 URL よりダウンロードした Cyclone V GT FPGA 開発キット用のデザインをベースとして、Quartus II 開発ソフトウェア v14.0 用にカスタマイズしたデザインを使用しました。本資料の紹介ページに Quartus II 開発ソフトウェア v14.0 用のデザインがありますので、参考にしてください。

なお、Altera Wiki のデザインは Quartus II 開発ソフトウェア v13.0 対応となっており、下記 URL から取得できます。

[http://www.alterawiki.com/uploads/9/95/C5gt\\_gen2x4\\_hmc\\_13\\_0\\_0.zip](http://www.alterawiki.com/uploads/9/95/C5gt_gen2x4_hmc_13_0_0.zip)

## 2. ハードウェア概要

### 2-1. ハードウェア構成

本デザインは、FPGA 内に実装された DMA エンジンを使用して DMA 転送を行います。DMA リードの場合は、PC ソフトウェアからの操作により PCI-Express リンク経由でコマンドが発行されると DMA エンジンの設定レジスタに必要な情報が設定され、DMA エンジンがマスタとして動作します。DMA エンジンは、メモリ・リード・コマンドで PC 側のメモリからデータを読み出し、FPGA 内部のオンチップ・メモリ、または、FPGA に接続されている外部 DDR3 SDRAM メモリにデータを書き込みます。DMA ライトの場合は逆に、DMA エンジンは、FPGA 内部のオンチップ・メモリ、または、FPGA に接続されている外部 DDR3 SDRAM メモリからデータを読み出し、メモリ・ライト・コマンドで PC 側のメモリにデータを書き込みます。

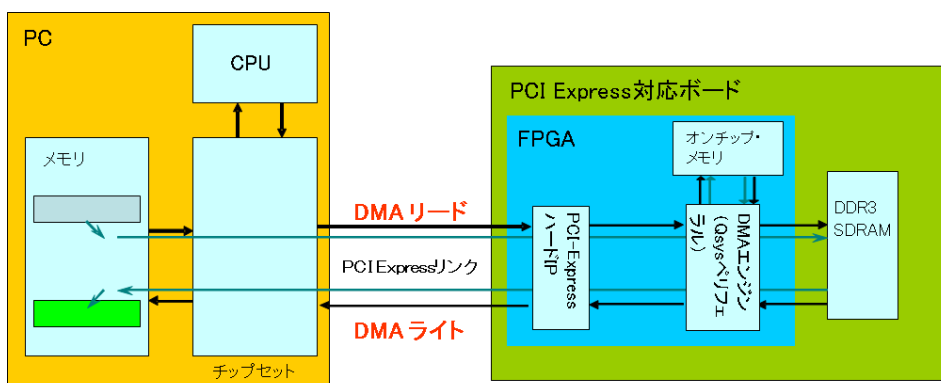


図 2-1. AN 431 ハードウェア構成ブロック図

### 2-2. ハードウェア・デザイン構成

本デザインでは、Qsys のコンポーネントとして、PCI-Express のハード IP、オンチップ・メモリ、DMA エンジン、外部メモリ (DDR3 SDRAM) コントローラをインプリメントしています。

ダウンロードした該当のデザインを Quartus II 開発ソフトウェア v14.0 で開き、Qsys を起動します。q\_sys.qsys ファイルを Qsys から開くと図 2-2 のような画面が表示されます。Qsys 上には、既に上記のコンポーネントが実装済みであることが確認できます。

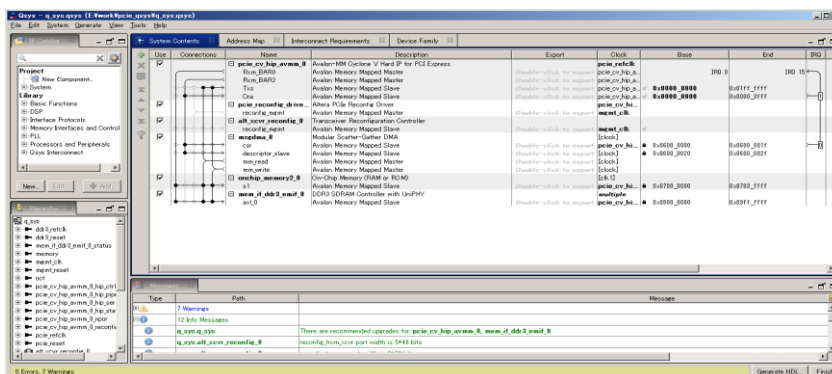


図 2-2. Qsys 設定画面

さらに、図 2-3 では、理解しやすいように、Qsys の接続内容をブロック図として示しました。

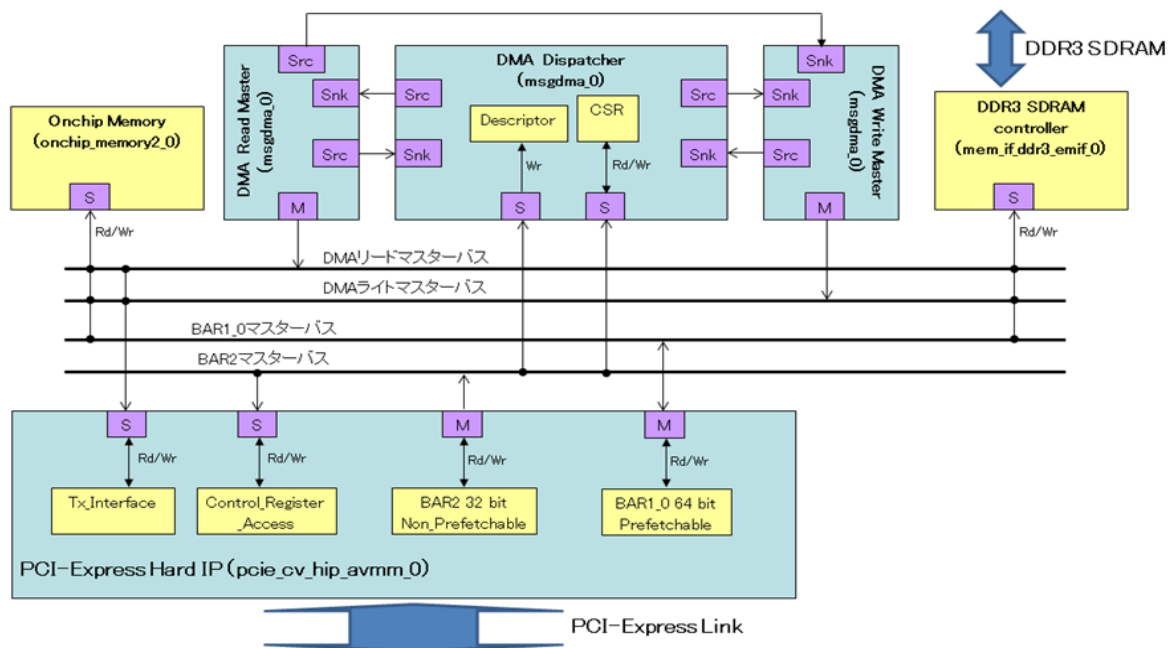


図 2-3.ハードウェア・デザイン・ブロック図

PCI-Express のハード IP は、PCI-Express リンクとの接続を Gen2 の x4 モードで行っています。ベース・アドレスは、BAR1\_0 で 64 ビット・プリフェッチャブル・アドレス空間を、BAR2 で 32 ビット・ノン・プリフェッチャブル・アドレス空間を実装しています。Control\_Register\_Access では PCI-Express (PC 側) のアドレス空間と Avalon-MM (Qsys 側) のアドレス空間の変換を行う設定など、ブリッジ機能としての各種設定を行うためのレジスタ・アクセス用スレーブ・ポートとなっています。Tx\_Interface は、DMA リード/ライト・マスタからの要求を受け付けるためのスレーブ・ポートの役割を担います。

オンチップ・メモリ(図 2-3 では、Onchip\_Memory)は、FPGA 内部のメモリですが、転送レート計測用ソフトウェアの転送モードを“OnChipMemory”とした場合に転送のターゲットとなります。これは、PCI-Express の BAR1\_0 マスタと DMA リード/ライト・マスタからアクセスされます。

DDR3 SDRAM コントローラは、UniPHY ベースのメモリ・コントローラが実装されています。転送レート計測用ソフトウェアの転送モードを“DDR”とした場合に転送のターゲットとなります。バス幅 32 ビットでビットあたり 600Mbps の転送レートで該当ボードに実装されている DDR3 SDRAM にアクセスしています。

DMA エンジン、mSGDMA (モジュラー・スキヤッタ・ギャザ DMA) を使用しています。この mSGDMA は、Quartus II 開発ソフトウェア v14.0 から Qsys のペリフェラルとして付属しています。v14.0 より前のバージョンでは Qsys に付属されているペリフェラルではなく、Altera Wiki に掲載されているリファレンス・デザインに付属しています。Qsys には、これ以外に DMA (Simple DMA)、SGDMA の 2 種類の DMA エンジンがありますが、広帯域を要求する DMA 転送を行う場合には、mSGDMA が適しています。mSGDMA は、DMA Dispatcher、DMA Read Master、DMA Write Master から構成されます。DMA Dispatcher が PC から BAR2 経由で DMA 転送に必要なディスクリプタ・テーブル (転送に必要な転送元/先アドレス、転送サイズ、など) 情報とコントロール情報を受け取ると DMA 転送が開始されます。DMA Read Master、DMA Write Master は、DMA Dispatcher 間と DMA Read Master、DMA Write Master 間のインタフェースを Avalon-ST とすることで転送効率の最適化を行い、広帯域を必要とするデータ転送に対応しています。

### 2-3. PCI-Express のハード IP

PCI-Express ハード IP の設定について説明します。図 2-2 の状態から Qsys の pcie\_cv\_hip\_avmm\_0 をダブルクリックすると Avalon-MM Cyclone V Hard IP for PCI Express が起動します。

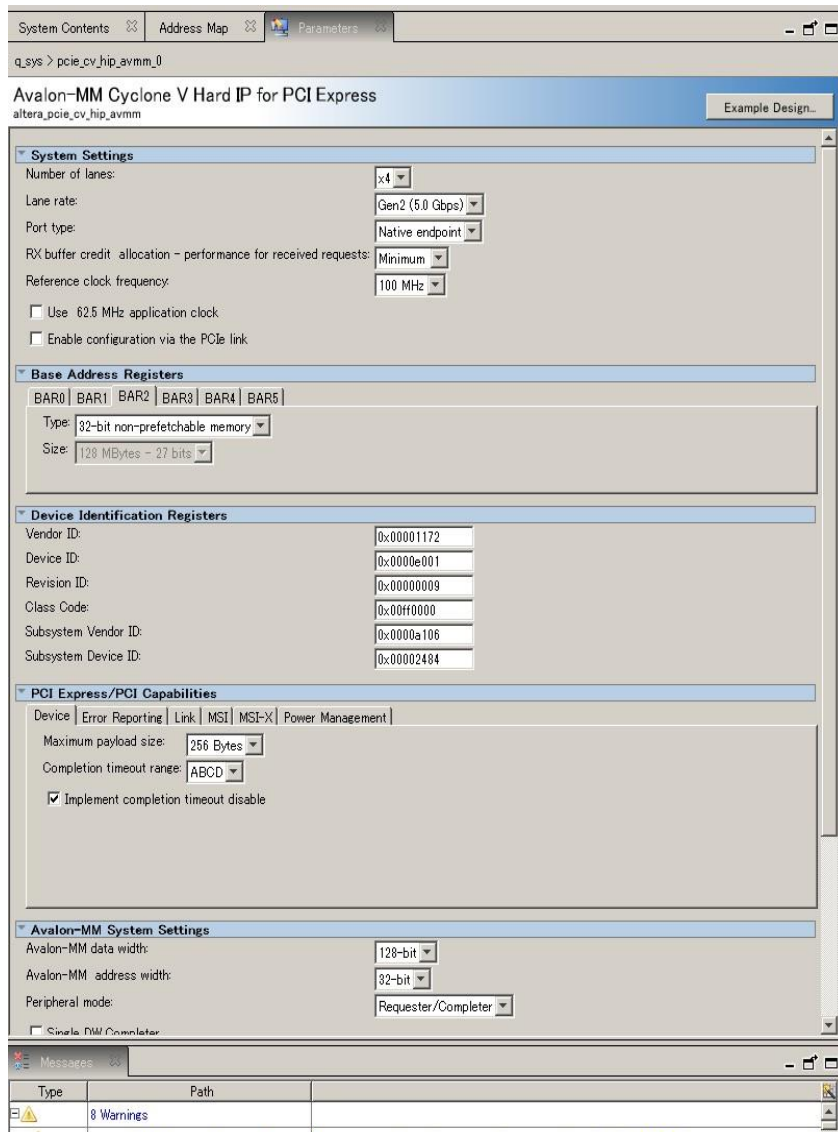


図 2-4. Avalon-MM Cyclone V Hard IP for PCI Express

2-3-1. System Setting

System Setting では、Gen2 対応の設定、レーン数の設定、リファレンス・クロックの設定、アプリケーション・クロック、受信バッファなどの設定を行います。各設定に共通ですが、設定できる値はデバイスによって異なるため、設定できない値を選択した場合にはエラーが表示されます。リファレンス・クロック設定は通常では、対向のルート・コンプレックスからの 100MHz を設定しますが、125MHz を設定することもできます。PCI-Express の規格では、±300ppm の周波数偏差までを IP 部分で吸収可能であるため、エンドポイント側で周波数偏差 ±300ppm 以内の精度のよい固定クロックをリファレンス・クロックとして使用することもできます。アプリケーション・クロック(FPGA ローカル・インタフェース)として 62.5MHz を設定できますが、Cyclone V GT デバイスでは、Gen1 x1 レーン設定時のみ選択できます。

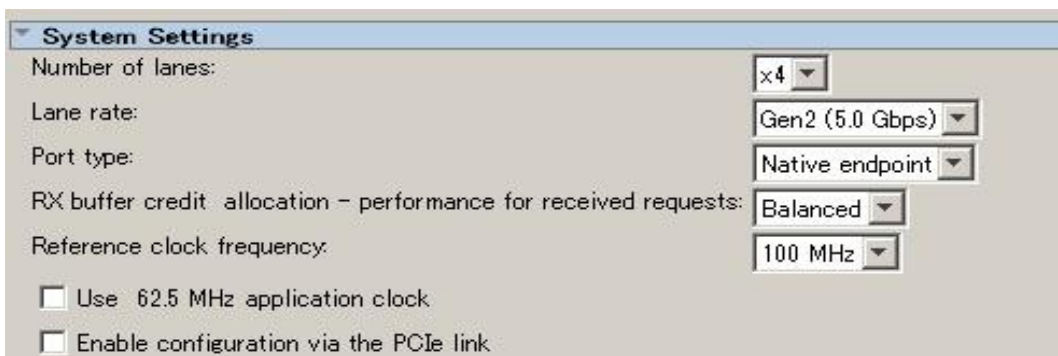


図 2-5. System Setting

2-3-2. PCI Base Address Registers

PCI のベース・アドレスを設定します。このデザインでは、BAR0\_1 に 64bit Prefetchable タイプが 28 ビット(256MByte)のアドレス空間で、BAR2 に 32bit Non-Prefetchable タイプが 27 ビット(128MByte)のアドレス空間で設定されています。空間の大きさは手動では設定できず、Qsys 側で他のコンポーネントの空間の大きさを自動的に計算して適切な値を設定します。



図 2-6. PCI Base Address Registers



### 2-3-3. Device Identification Registers

Vendor ID、Device ID 等の設定を行います。デフォルトでは、アルテラ社の Vendor ID が設定されています。そのままアルテラ社の ID を使用しても構いませんが、オープン・システムで使用する場合には、Vendor ID を PCI-SIG で取得し設定することをお勧めします。ただし、これには、PCI-SIG のメンバー登録が必要となり、\$3000(US)/年(2014 年 10 月現在)の費用が掛かります。

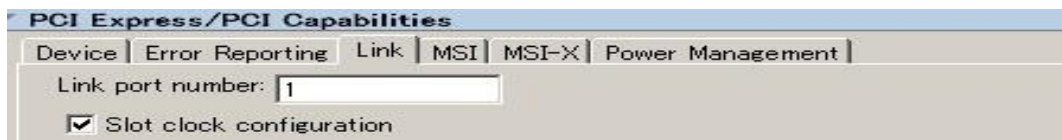


Device Identification Registers	
Vendor ID:	0x00001172
Device ID:	0x0000e001
Revision ID:	0x00000009
Class Code:	0x00ff0000
Subsystem Vendor ID:	0x0000a106
Subsystem Device ID:	0x00002484

図 2-7. Device Identification Registers

### 2-3-4. Link Capabilities

リンク・ケーパビリティ・レジスタの Port Number の指定を行います。通常は、デフォルトで問題ありません。



PCI Express/PCI Capabilities

Device | Error Reporting | **Link** | MSI | MSI-X | Power Management

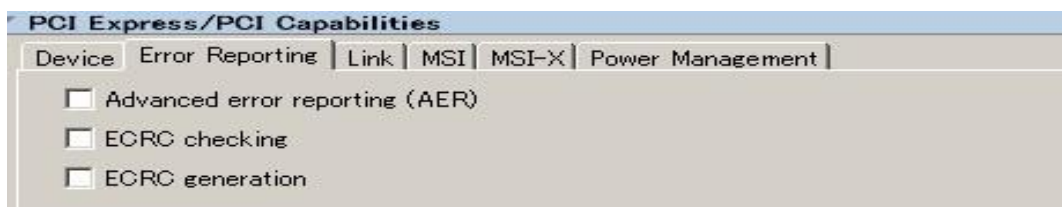
Link port number:

Slot clock configuration

図 2-8. Link Capability

### 2-3-5. Error Reporting

オプションのエラー・レポート機能の設定を行います。ここでは使用しないため、すべてチェックは外れています。



PCI Express/PCI Capabilities

Device | **Error Reporting** | Link | MSI | MSI-X | Power Management

Advanced error reporting (AER)

ECRC checking

ECRC generation

図 2-9. Error Reporting

### 2-3-6. Avalon-MM System Settings

Avalon-MM 設定を行います。Peripheral Mode では、以下のいずれかを選択します。

Requester/Completer : データライト/リードの要求と応答に対応

Completer-Only : データライト/リードの応答のみに対応、DMA 不可

DMA 転送を行うためには、エンドポイント側がマスタとして動作する必要があります。したがって、ここでは、Requester/Completer を選択しています。

Avalon-MM data width, Avalon-MM address width は、それぞれ Avalon バスとのデータ幅とアドレス幅を指定します。この値は PCI-Express の Lane rate と Lane 数に応じて決まります。

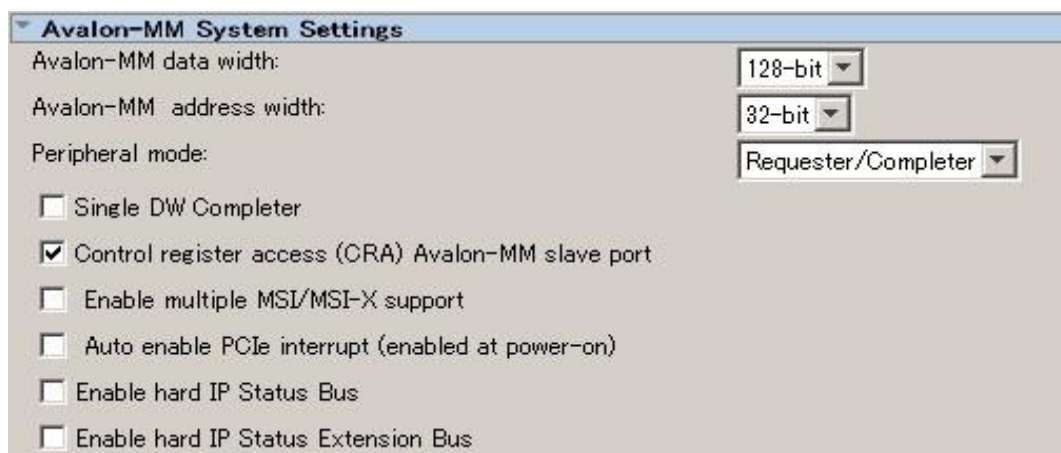


図 2-10. Avalon-MM Settings

### 2-3-7. Address Translation

Qsys を使用した場合、PCI-Express(PC 側)のアドレス空間と Avalon(Qsys システム内)のアドレス空間の 2 つのアドレス空間が存在するため、お互いの変換処理が必要となります。ここでの設定は、Avalon-MM (Qsys 側) ⇒ PCI-Express(PC 側)アドレス変換時の各種設定となります。

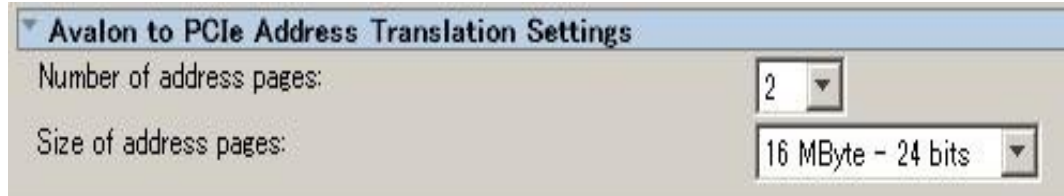


図 2-11. Address Translation

Dynamic translation table を指定した場合には、Number of address pages は 512 が最大値となります。ここでは、Number of address pages が 2、Size of address pages が 16MByte (24bit) となっているので、16MByte のアドレス空間を変換できるページを 2 つ指定しています。この指定により、PCI-Express コア内の Control\_Register\_Access のアドレス変換テーブルが有効となります。Avalon-MM(Qsys 側) ⇒ PCI-Express(PC 側)アドレス変換については、さらに 3-2 章で詳しく説明してありますので、そちらを参照ください。

また、PCI-Express(PC 側) ⇒ Avalon-MM(Qsys 側)のアドレス変換については、ここでは設定の必要はなく、ベース・アドレスの設定時に自動的に行われます。こちらの詳細については、3-1 章で詳しく説明してありますので、そちらを参照ください。

クローズド・システムで固定アドレスの指定が可能な場合には、Fixed translation table が指定できます。Fixed translation table を指定した場合は、Number of address pages は 16 が最大値となります。すなわち、固定のアドレス値を持った最大 16 ページのアドレス変換テーブルを持つことができます。変換するアドレス値は、Address Translation Table に手動で登録します。

## 2-4. mSGDMA

このモジュールは DMA 転送の制御を行います。v14.0 からこの mSGDMA が Qsys に付属されるようになりました。v14.0 より前のバージョンで mSGDMA を使用する場合は、従来通り Altera Wiki よりリファレンス・デザインをダウンロードする必要があります。また、Altera Wiki には、mSGDMA 専用の説明ページもありますので、併せて参照ください。

[http://www.alterawiki.com/wiki/Modular\\_SGDMA](http://www.alterawiki.com/wiki/Modular_SGDMA)

今回の構成では、DMA の転送元/先がともに Avalon-MM インタフェースであるため DMA Mode では、Memory-Mapped to Memory-Mapped を選択しています。Descriptor FIFO Depth は、64 を選択していますが、これは DMA のディスクリプタ・テーブルを FIFO に何個まで溜められるかを指定します。

CSR が PCI-Express リンクから BAR2 経由で DMA 転送のための各種設定を行うインタフェースです。Descriptor\_Slave が DMA 転送用のディスクリプタ・テーブルを送るためのインタフェースです。

mm\_write は外部メモリ、あるいはオンチップ・メモリへデータを書きこみます。mm\_read は外部メモリ、またはオンチップ・メモリからデータを読み出します。csr\_irq が割り込み完了時に PC に対して割り込みを発行するためのインタフェースです。

アルテラ社の mSGDMA ユーザ・ガイドがリリースされておりますので、詳細情報はこちらを参照ください。

[https://www.altera.com/en\\_US/pdfs/literature/ug/ug\\_embedded\\_ip.pdf](https://www.altera.com/en_US/pdfs/literature/ug/ug_embedded_ip.pdf)

※ “Altera Modular Scatter-Gather DMA” で検索してください。

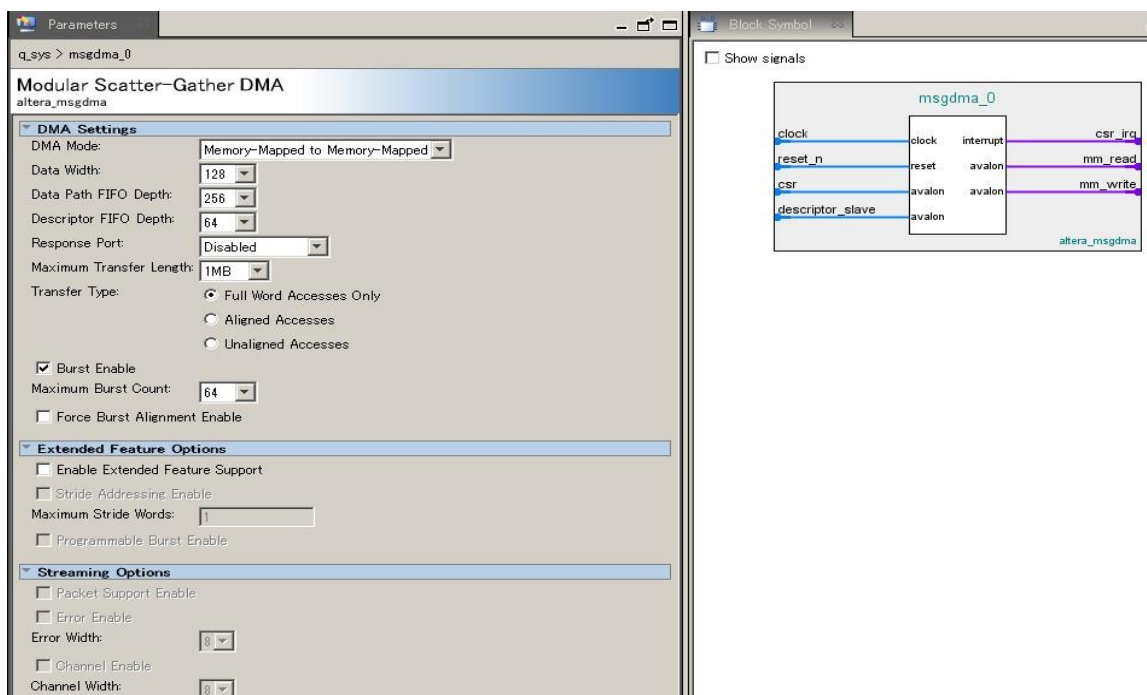


図 2-12. Modular SGDMA

## 2-5. DDR3 SDRAM コントローラ

DDR3 SDRAM コントローラは、UniPHY ベースのコントローラを実装しています。100MHz のリファレンス・クロックから外部メモリへのクロック 300MHz を生成し、32 ビット幅の 64MByte のメモリ・アクセスを構成しています。リファレンス・デザインの DDR3 SDRAM コントローラの設定ではデータ幅が 40bit となっていますが、このうち 8bit は ECC の機能を実現しているため、実際のデータ幅は 32bit となります。1 ビットあたりのビット・レートは、300MHz x 2 = 600MHz となります。また、Controller data rate を Full にしているため、FPGA 内部のバス幅は、32 ビット x 2 = 64 ビットとなります。DDR3 SDRAM メモリ・コントローラについての詳細については、本資料では趣旨から外れるため、これ以上立ち入りません。詳しく知りたい方は、アルテラ社の以下のページなどに詳しく情報がありますので、そちらを参照ください。

アルテラ社 外部メモリ・インタフェース

<https://www.altera.co.jp/solutions/technology/external-memory/overview.html>

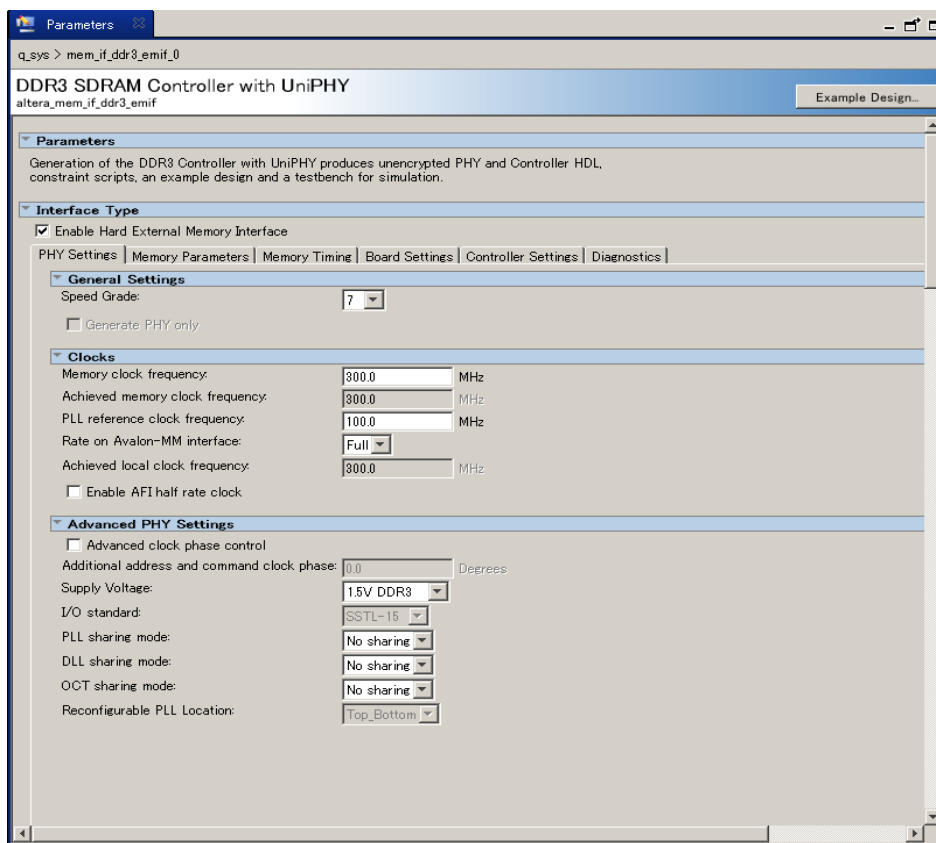


図 2-13. DDR3 SDRAM コントローラ

## 2-6. オンチップ・メモリ

オンチップ・メモリのバス幅は 128 ビットとしています。容量は、262Kbyte です。今回の設計では、DMA 転送の元/先の設定を外部メモリ(DDR3 SDRAM)かオンチップ・メモリのどちらかに設定できる仕様であるためオンチップ・メモリを実装しています。

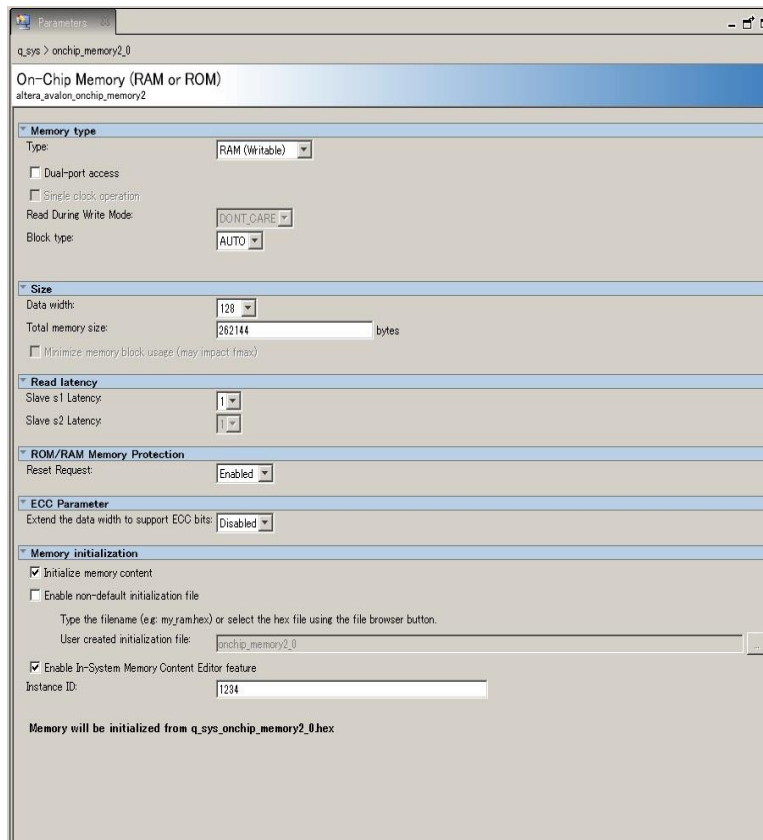


図 2-14. オンチップ・メモリ

## 2-7. XCVR リコンフィギュレーション

PCI-Express コンポーネントのインターフェース一覧を見ると、reconfig\_to\_xcvr、reconfig\_from\_xcvr などのインターフェースがあります。これは、トランシーバ・ブロックを使用する際に必要な XCVR リコンフィギュレーション用のインスタンスを接続するためのインターフェースとなります。本来、XCVR リコンフィギュレーションはトランシーバ・ブロックの動作を動的に変更するための機能です。例えば、転送レートや各種 PMA 設定などを変更する際に使用します。今回は、このような動的な変更は行いませんが、オフセット・キャンセレーションを有効にするために必須となります。Cyclone IV GX、Arria® II GX、Stratix® IV GX 以降のデバイスでは、トランシーバ・ブロックの使用に際してオフセット・キャンセレーションの使用が必須となっているため、Transceiver Reconfiguration Controller のインスタンスを行わないと Quartus II 開発ソフトウェアにてエラーとなります。オフセット・キャンセレーションの詳細については、各種デバイスのデータシートを参照ください。(Cyclone V GT の場合は、[https://www.altera.com/en\\_US/pdfs/literature/hb/cyclone-v/cv\\_53007.pdf](https://www.altera.com/en_US/pdfs/literature/hb/cyclone-v/cv_53007.pdf) の “Offset Cancellation” を参照ください。)

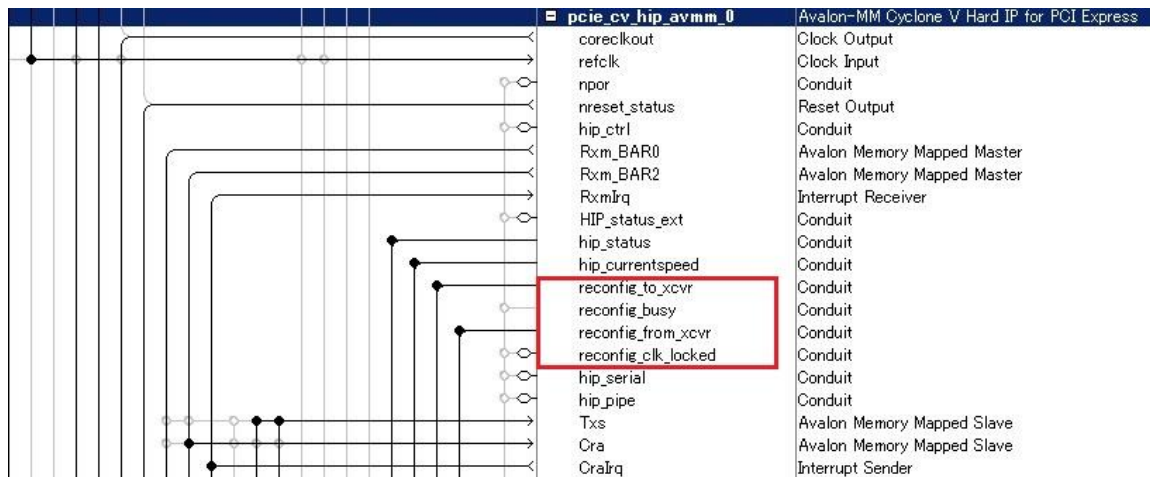


図 2-15. GXB リコンフィギュレーション用インターフェース

## 2-8. クロック

このトップ・デザインには、100MHz のクロックが 3 つ(入力信号名: clkintop\_p, clkinbot\_p, pcie\_refclk\_p)、存在します。このうち、pcie\_refclk\_p は PCI-Express のリファレンス・クロックであり、トップ・ファイル内で直接 PCI-Express ハード IP のリファレンス・クロック入力(ポート名: pcie\_cv\_hip\_avmm\_0\_refclkt)に接続されています。clkintop\_p は、Qsys 内のトランシーバ・リコンフィギュレーション・ブロック(モジュール名: alt\_xcvr\_reconfig\_0)の入力クロック(ポート名: mgmt\_clk\_clk)に接続されています。clkinbot\_p は、DDR3 SDRAM コントローラのリファレンス・クロックとして使用されています。Qsys 内のポート名は ddr3\_refclk となっており、トップ・ファイルで接続されていることが確認できます。

PCI-Express ハード IP に必要な入力クロックとしては、リファレンス・クロックの 100MHz(または 125MHz)のみです。2-7 章でも記載した通り、トランシーバ・リコンフィギュレーション・ブロックが必要となり、Cyclone V GT ではそのリファレンス・クロックの周波数範囲が 75MHz ~ 100MHz と規定されていることに注意が必要です。



### 3. PCI-Express Avalon-MM ブリッジ

Qsys でシステムを構成した場合、Qsys 内部がローカルにアドレス空間を持つことになります。このアドレス空間は、PC 側から発行されるアドレスとは独立なものであるため、PCI-Express Avalon-MM ブリッジを介してアドレス変換を行う仕組みが用意されています。アドレス変換は、PCI-Express(PC 側) ⇒ Avalon-MM(Qsys 側)用のベース・アドレス・レジスタ(BAR)を使用した仕組みに基づくテーブルと Avalon-MM(Qsys 側) ⇒ PCI-Express(PC 側)のテーブルの 2 種類があります。この章では、それぞれのアドレス変換テーブルについて説明します。

また、割り込みに関しては、MSI、またはレガシーの割り込みに対応していますが、その設定についても簡単に説明します。

#### 3-1. PCI-Express ⇒ Avalon-MM アドレス変換テーブル

PCI-Express(PC 側)から Avalon-MM(Qsys 側)にアクセスする場合、起動時に BAR の設定がされているので、アクセスは有効な BAR のいずれかにされます。PCI-Express Avalon-MM ブリッジでは、それぞれの BAR に対応する Qsys のローカル・アドレスを割り当ててあるので、以下の図のように上位アドレスを変換することで PCI-Express ⇒ Avalon-MM アドレス変換を行います。たとえば、BAR0 がヒットした場合、下位アドレスに BAR0 に該当する Avalon アドレス B0 を連結して Avalon-MM アドレスを生成します。これについては、PCI-Express Avalon-MM ブリッジにて自動的に行われるため、ユーザが意識することなく実行されます。

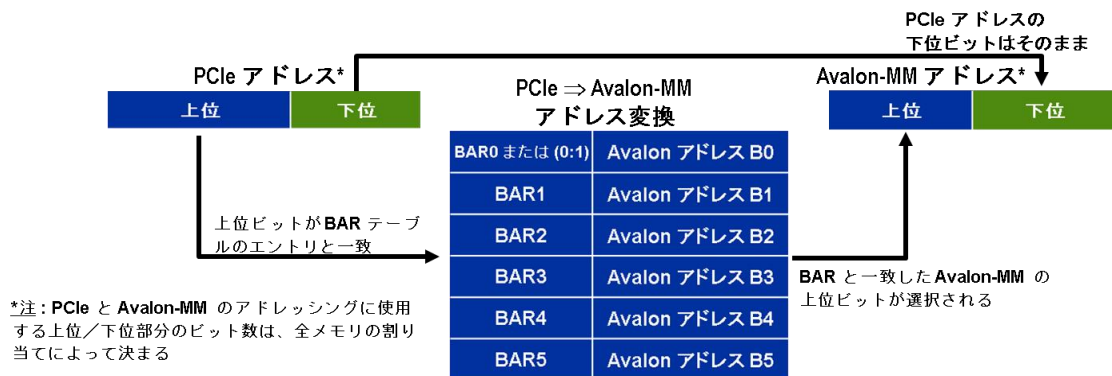


図 3-1. PCI-Express ⇒ Avalon-MM アドレス変換テーブル

### 3-2. Avalon-MM ⇒ PCI-Express アドレス変換テーブル

対して、Avalon-MM(Qsys 側)から PCI-Express(PC 側)にアクセスする場合は少々複雑です。基本的な仕組みは同じですが、このアドレス変換テーブルを使用するためには、事前にユーザが意図的に Control Register Access Module (CRA) レジスタ内の該当のテーブルに PC 側の物理アドレスを登録する必要があります。今回のデザインでは、図 2-12 にあるようにアドレス変換テーブルを Dynamic translation table としてアドレス・ページを 2 つ、サイズを 16MByte と設定しましたので、有効なテーブル数は 2 つとなり、PC 側から CRA レジスタの該当箇所事前にアクセス先のアドレス値を登録する必要があります。例えば、DMA ライト(FPGA から PC の共有メモリへの書き込み)を行う場合、まず PC 側で連続した 16MByte のアドレス空間をメモリ・アロケートして領域を確保します。次にその先頭アドレスをアドレス変換テーブルの該当箇所(Avalon-MM Control and Status Register の 0x1000 番地、詳細はユーザ・ガイドを参照)に書き込みます。その後 DMA 転送用の各種設定を行うと、Qsys 内部から PC の共有メモリへのライト・アクセスを行う時点で下図のようにアドレス変換テーブルを使用して、Avalon-MM アドレスから PCI-Express アドレスへの変換を行います。

このアドレス変換テーブルの数をいくつにするのか、またサイズをどのくらいにするのかは、一度に転送する最大データ量と同時にアクセスする転送数によります。また、このテーブルはアクセスのたびに動的に PC 側のソフトウェアから書き換えてもよいし、システム起動時、または最初のアクセス時にアロケートし設定した領域をそのまま残しておくような使用方法もできます。いずれにしても、注意点としては、この割り当てが正しくないと PC 側の不正アクセス領域にライトなどを行いシステムがフリーズするなどの現象が起こりうるということです。したがって、この部分の仕組みについては十分理解の上、実装をすることをお勧めします。

さらに細かい情報については、ユーザ・ガイド

([https://www.altera.com/en\\_US/pdfs/literature/ug/ug\\_c5\\_pcie\\_avmm.pdf](https://www.altera.com/en_US/pdfs/literature/ug/ug_c5_pcie_avmm.pdf)) の Avalon-MM-to-PCI Express Address Translation Algorithm for 32-Bit Addressing の項目にありますので、そちらを参照ください。

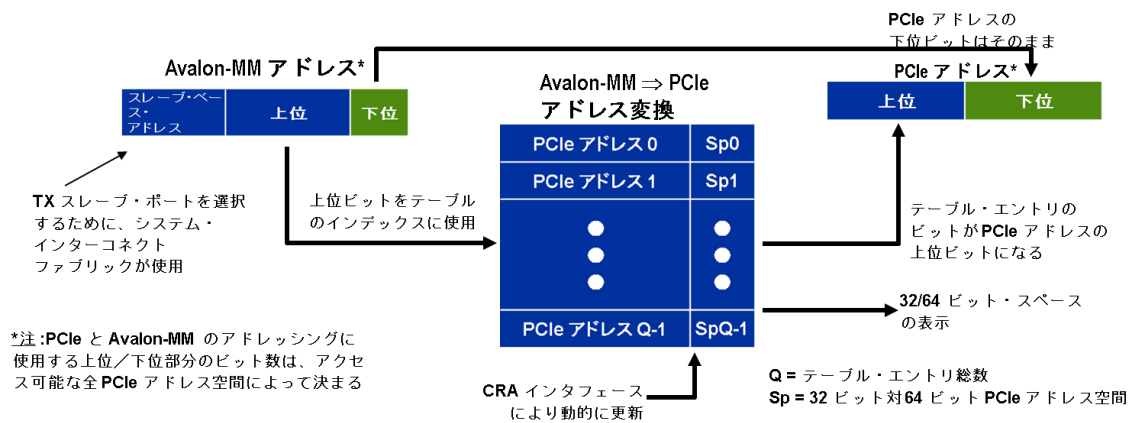


図 3-2. Avalon-MM ⇒ PCI-Express アドレス変換テーブル

### 3-3. 割り込み

アルテラ社の PCI-Express ハード IP は、レガシー、MSI いずれの割り込みにも対応しています。これらは排他的に設定されるため、どちらか一方を選択することになります。

割り込みに関するハードウェア(FPGA)側の設定は、BAR2 経由でアクセスする CRA のオフセット 0x0050 番地にある Avalon-MM to PCI Express Interrupt Enable Register の AVL\_IRQ[15:0] の該当ビットを '1' にすることで、Avalon-MM ⇒ PCI-Express の DMA 転送完了時の割り込みが正常に行われるようになります。

Table 5-14: Avalon-MM to PCI Express Interrupt Enable Register, 0x0050

Bits	Name	Access	Description
[31:24]	Reserved	N/A	N/A
[23:16]	A2P_MB_IRQ	RW	Enables generation of PCI Express interrupts when a specified mailbox is written to by an external Avalon-MM master.
[15:0]	AVL_IRQ [15 : 0]	RW	Enables generation of PCI Express interrupts when a specified Avalon-MM interrupt signal is asserted. Your Qsys system may have as many as 16 individual input interrupt signals.

図 3-3. 割り込み有効レジスタ

割り込みに関するソフトウェア(PC)側の設定は、PCI コンフィギュレーション空間にアクセスすることで行います。Command レジスタの 10 ビット目に “Interrupt Disable” ビットがあります。また、MSI Capability レジスタの Message Control レジスタの 0 ビット目に “MSI Enable” ビットがあります。これらを制御することで、割り込みの設定を行います。レガシーの割り込みを有効にするには、“Interrupt Disable” ビットを '0' に設定し、“MSI Enable” ビットを '0' に設定します。また、MSI の割り込みを有効にするには、“MSI Enable” ビットを '1' に設定し、次に “Interrupt Disable” ビットを '1' に設定します。

## 4. SignalTap II を使用した実機動作確認

この章では、今まで説明したことを踏まえて SignalTap II を使用した Qsys 内部の各インタフェースの動作について検証します。検証は、アルテラ社から提供されている転送レート計測用ソフトウェアを使用しています。SignalTap II での波形観測は、オンチップ・メモリへのターゲット・ライトと外部 DDR3 SDRAM へのターゲット・リードを確認し、それぞれのターゲットに対して正しくアドレスの指定が行われていることと、基本的な操作方法を確認した後、DMA 転送については、外部 DDR3 SDRAM と PC 側の共有メモリ間の DMA リード(PC 側の共有メモリ ⇒ 外部 DDR3 SDRAM)の動作について細かく確認していきます。DMA ライト動作については、転送方向が逆になるだけで、基本的な操作フローは同一であるため、割愛します。

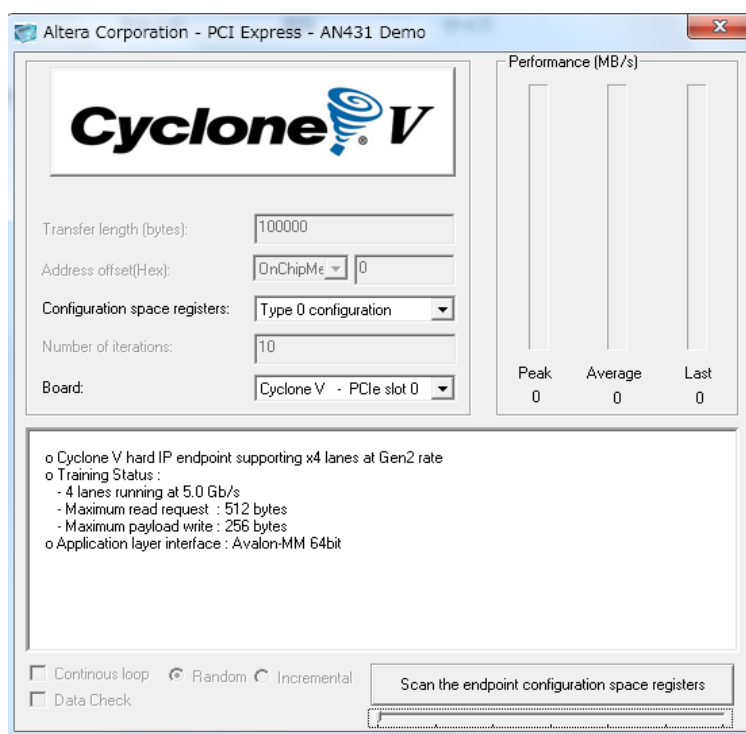


図 4-1. AN 431 対応転送レート計測用ソフトウェア

### 4-1. Qsys 内のアドレス設定

図 4-2 に Qsys 内で設定されたアドレス設定についてブロック図内に記載しました。また、表 4-3 に同じものを表でまとめました。表中の○は接続、×は非接続を示しています。pcie\_cv\_hip\_avmm\_0.Rxm\_BAR0 と onchip\_memory2\_0.s1 が交差するセルは○となっているので、ここは接続状態であることを示します。

一見、Tx\_Interface(pcie\_cv\_hip\_avmm\_0.Txs)のアドレスと Control\_Register\_Access

(pcie\_cv\_hip\_avmm\_0.Cra)のアドレスは、ともに 0x00000000 で始まっており、重複しているように見えますが、Tx\_Interface のアドレス空間は PC 側のアドレス空間を指しているので実際にはこれらは同一アドレス空間の値ではないことに注意してください。Tx\_Interface のアドレスは、mSGDMA モジュールからのアクセスにより PC の共有メモリへのリード/ライトを発行します。このとき、3-2 章で説明した Avalon-MM ⇒ PCI-Express アドレス変換テーブルを経由して PC 側のアドレス空間に変換されてアクセスされることになります。

Tx\_Interface 以外のアドレスは、Qsys 内部のローカルな Avalon-MM アドレス空間であるため、値が重複した場合には、エラーとなります。

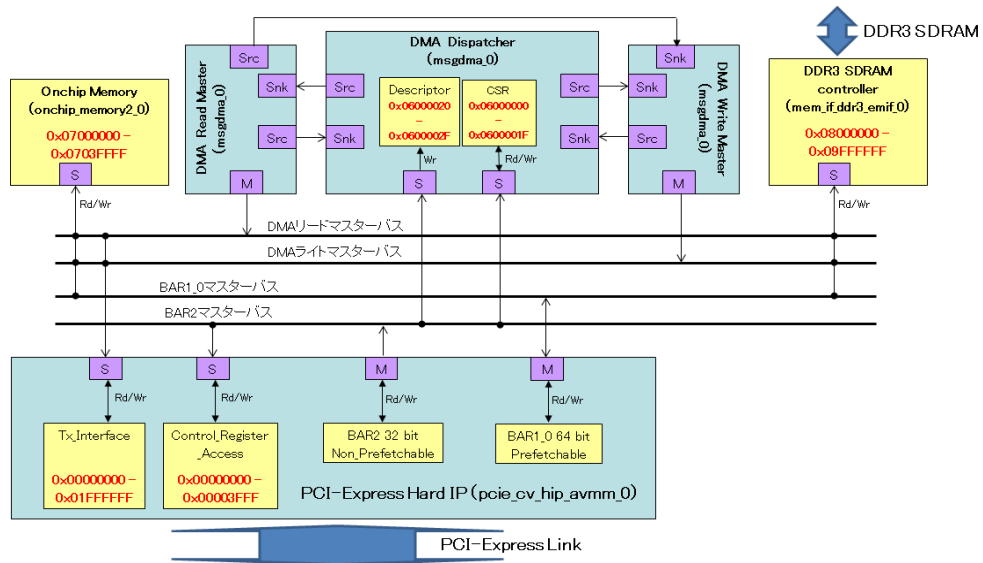


図 4-2. Qsys 内のアドレス設定

Slave Interface	Address	Size (Byte)	Master Interface			
			pcie_cv_hip_avmm_0.Rxm_BAR0	pcie_cv_hip_avmm_0.Rxm_BAR2	msgdma_0.mm_read	msgdma_0.mm_write
onchip_memory2_0.s1	0x07000000 - 0x0703FFFF	16K	○	×	○	○
mem_if_ddr3_emif_0.av1_0	0x08000000 - 0x09FFFFFF	32M	○	×	○	○
pcie_cv_hip_avmm_0.Cra	0x00000000 - 0x00003FFF	1K	×	○	×	×
msgdma.csr	0x06000000 - 0x0600001F	32	×	○	×	×
msgdma.descriptor_slave	0x06000020 - 0x0600002F	32	×	○	×	×
pcie_cv_hip_avmm_0.Txs	0x00000000 - 0x01FFFFFF	32M	×	×	○	○

表 4-3. Qsys 内のアドレス設定

### 4-2. SignalTap II の波形設定

SignalTap II の波形観測は、PCI-Express コアの Avalon-MM ブジツリ側のポートを観測することで行いました。図 4-4 の“Cra”で始まる Control\_Register\_Access インタフェース、Bar0 で始まる BAR0 64 bit Prefetchable インタフェース、Bar2 で始まる BAR2 32 bit Non\_Prefetchable インタフェース、Txs で始まる Tx\_Interface インタフェースをそれぞれ SignalTap II 上に Node Finder で展開しました。展開後の状態が、図 4-5 となります。

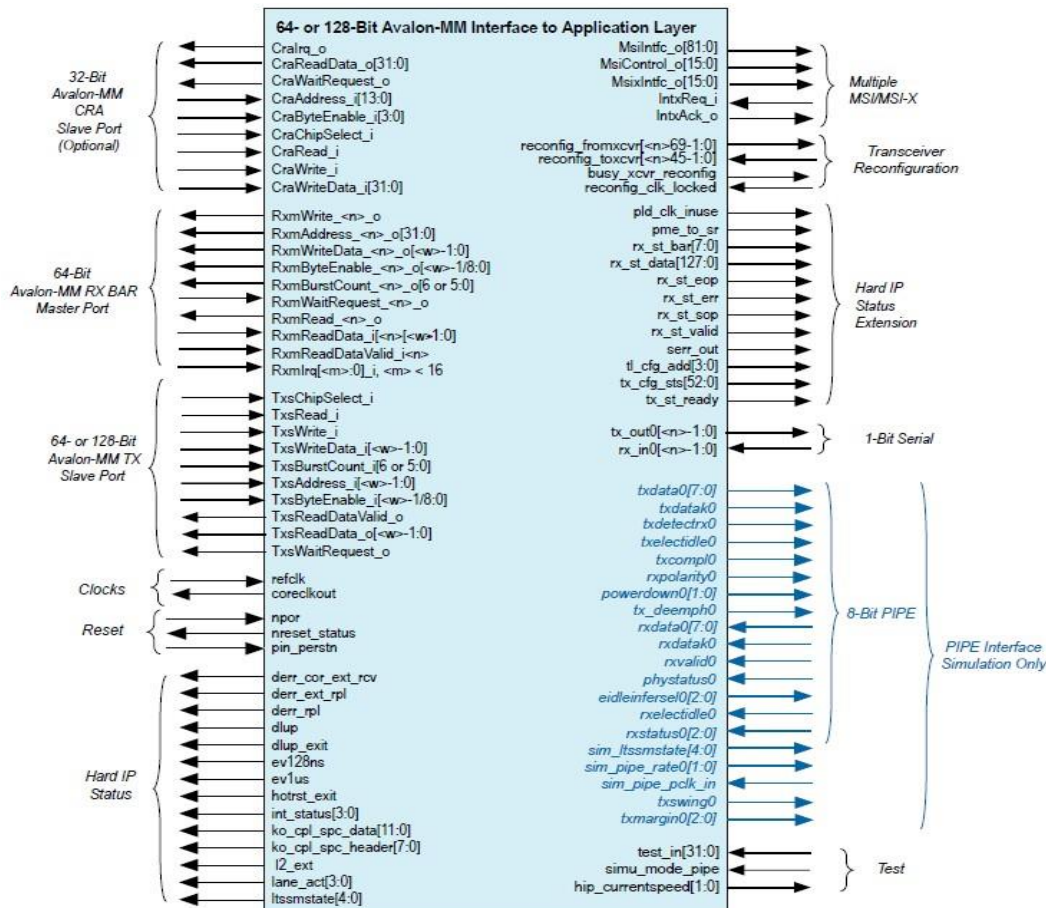


図 4-4. PCI-Express コア Avalon-MM インタフェース



### 4.3. オンチップ・メモリへのターゲット・ライト

オンチップ・メモリへのターゲット・ライトは、BAR1 64 bit Prefetchable インタフェース経由で PC からアクセスされるので、“RxmWrite\_2\_o” 信号にトリガをかけます。

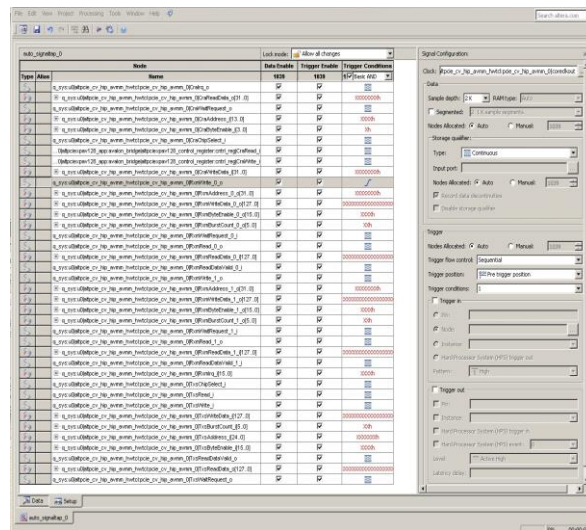


図 4-6. オンチップ・メモリへのターゲット・ライト発行トリガ信号

ソフトウェアの下部にあるスライド・バーを“Run target write”に変更して、endpoint address（実際には、アドレスではなくデータサイズ）を 0x10(16Byte)に設定して、“Address offset (Hex)”の項目で“OnChipMemory”を選択します。“endpoint data value”を 0x1 とすることで、書き込むデータを 1 から始まるインクリメント・データにします。そして、“Run target write”をクリックします。

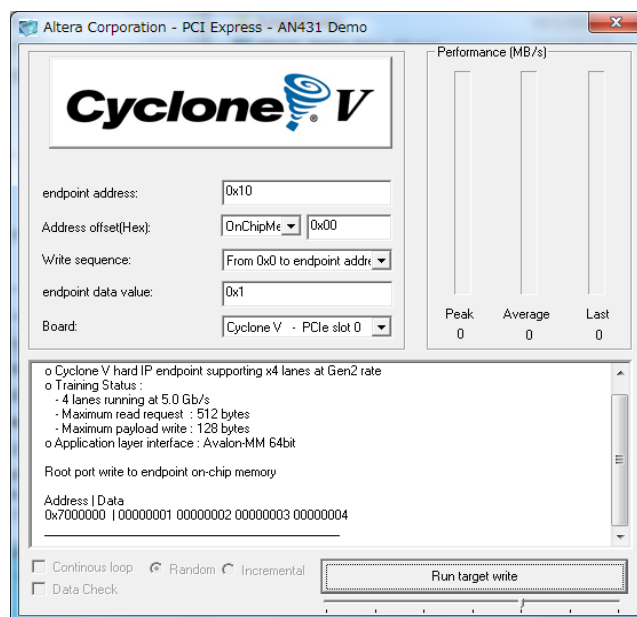


図 4-7. オンチップ・メモリへのターゲット・ライト発行



トリガ信号を検出すると、図 4-8 のように 16 回のライト・アクセスが確認できます。ターゲット・ライト動作では、1 バイト単位でアクセスが行われるため、これで、0x10(16)バイトのアクセスとなります。

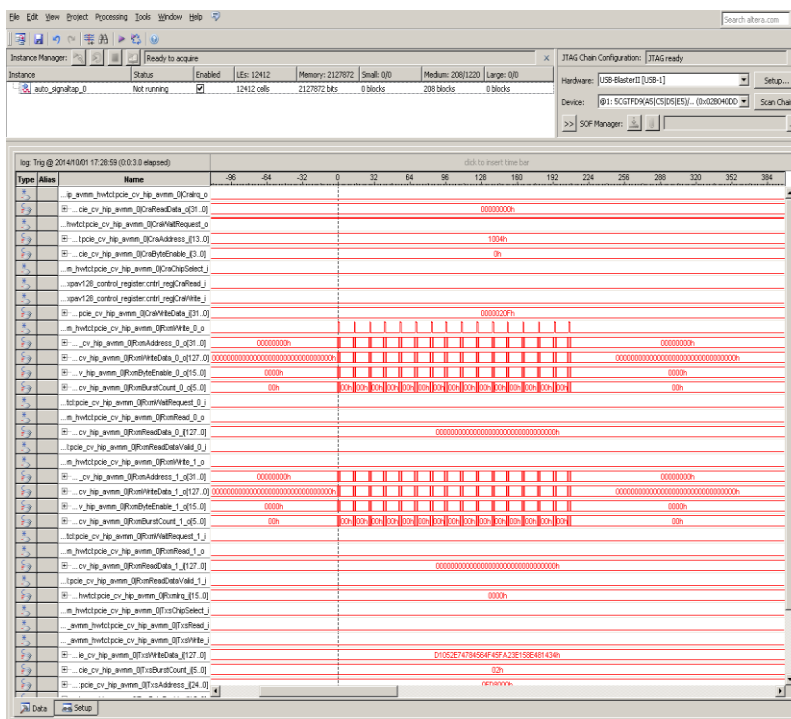


図 4-8.オンチップ・メモリへのターゲット・ライト波形

1 アクセス分の波形を拡大したのが、図 4-9 です。これを見ると、RxmAddress\_0\_o = 0x7000000 番地(オンチップ・メモリのアドレス)に、RxBurstCount\_0\_o = 0x1 でアクセスしているのが分かります。RxmByteenable\_0\_o = 0x01 となっているので、有効データは、RxmWriteData\_0\_o のバス幅が 8 バイトあるうちの最下位バイトとなり、0x01 を書き込んでいることが分かります。

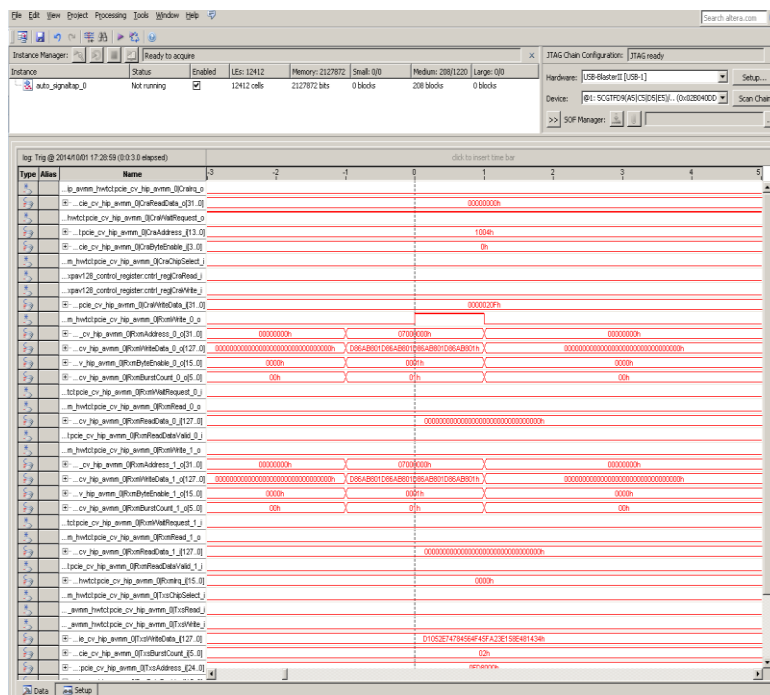


図 4-9.オンチップ・メモリへのターゲット・ライト拡大波形

#### 4.4. 外部 DDR3 SDRAM へのターゲット・リード

オンチップ・メモリへのターゲット・ライトと同じ手順で、外部 DDR3 SDRAM へのターゲット・リードを実行します。

外部 DDR3 SDRAM へのターゲット・リードは、BAR1 64 bit Prefetchable インタフェース経由で PC からアクセスされるので、“RxmRead\_0\_o” 信号にトリガをかけます。

次に、ソフトウェアの下部にあるスライド・バーを“Run target read”に変更して、endpoint address(実際には、アドレスではなくデータサイズ)を 0x10 に設定して、“Address offset (Hex)” の項目で“DDR”を選択します。そして、“Run target read” をクリックします。

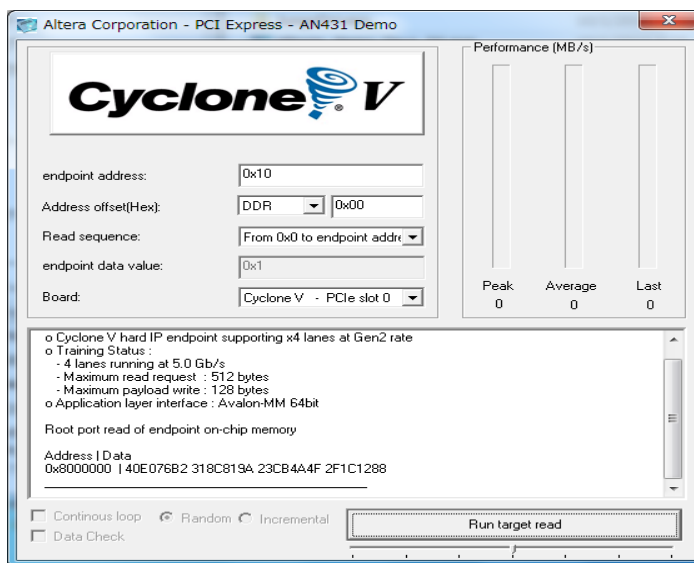


図 4-10. 外部 DDR3 SDRAM へのターゲット・リード発行

こちらは、ライトとは異なり 4 バイト単位でのアクセスとなります。図 4-11、4-12 は、DDR3 SDRAM へのターゲット・リードの拡大波形です。RxmAddress\_0\_o = 0x8000000 番地 (DDR3 SDRAM のアドレス) に RxBurstCount\_0\_o = 0x01 でアクセスしています。しかし、RxmByteEnable\_0\_o = 0x0F となっているので、4 バイトアクセスであることが分かります。読み出したデータは、RxmReadDataValid\_0\_o = 1 のところが有効データとなります。下位 4 バイトが有効となるので、0x00000001 が読み出したデータであることを示しています。

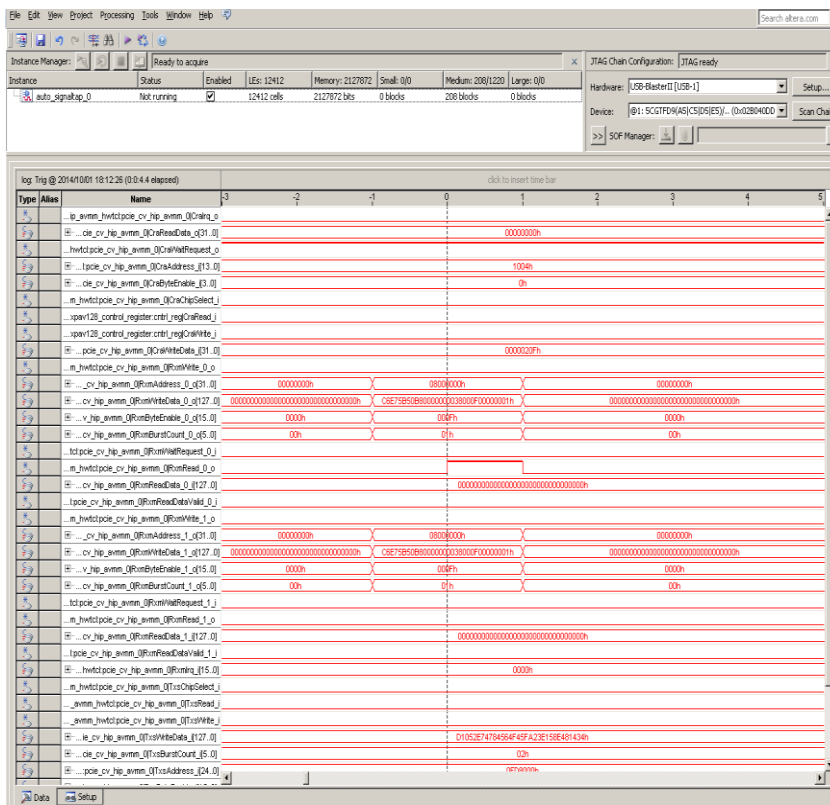


図 4-11. 外部 DDR3 SDRAM へのターゲット・リード拡大波形(リード要求部)

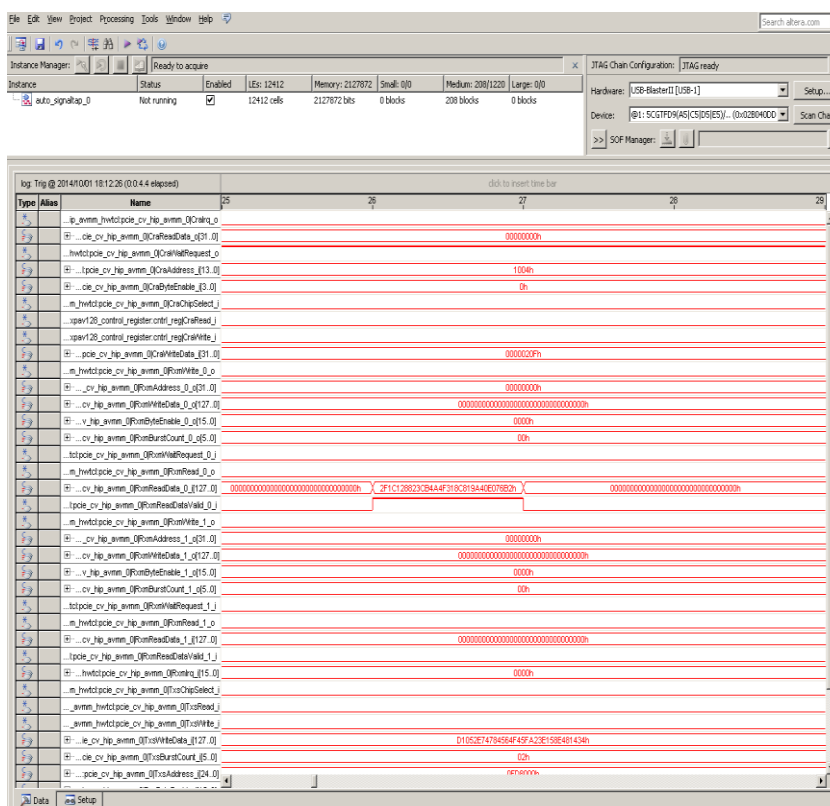


図 4-12.外部 DDR3 SDRAM へのターゲット・リード拡大波形(データ応答部)

### 4.5. 外部 DDR3 SDRAM への DMA リード

外部 DDR3 SDRAM への DMA リードを実行します。転送レート計測用ソフトウェアの右下のスライダーを“Run DDR DMA Test”に変更し、転送サイズを 0x40000Byte、方向を PC ⇒ FPGA とすることで DMA リードの転送を行うモードになります。“Run DDR DMA Test” ボタンをクリックすると DMA 転送が開始され、転送レートが表示されます。

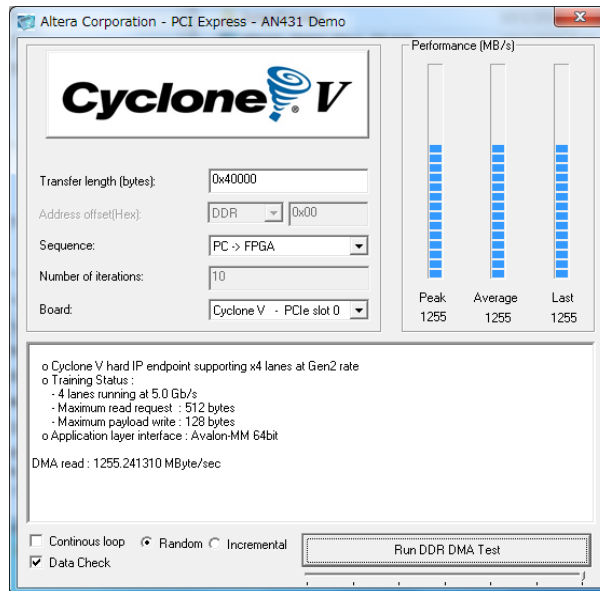


図 4-13.外部 DDR3 SDRAM への DMA リード発行

図 4-14 が DMA 転送を行った際の全体波形となります。前半部分で Control\_Register\_Access の Avalon-MM ⇒ PCI-Express アドレス変換テーブルへの設定を行っています。中間部分では、DMA Dispatcher の CSR, Descriptor への各種設定を行っています。設定完了後、後半部分で DMA リードの転送が行われます。



図 4-14.外部 DDR3 SDRAM への DMA リード波形

各レジスタ設定の詳細などは細かくなり過ぎるので、各ペリフェラルのマニュアルを参照ください。

Avalon-MM ⇒ PCI-Express アドレス変換テーブルのレジスタ情報については、Cyclone V Avalon-MM Interface for PCIe Solution User Guide の Chapter 9. IP Core Architecture の Avalon-MM-to-PCI Express Address Translation for 32-bit Addressing を参照ください

[https://www.altera.com/en\\_US/pdfs/literature/ug/ug\\_c5\\_pcie\\_avmm.pdf](https://www.altera.com/en_US/pdfs/literature/ug/ug_c5_pcie_avmm.pdf)

mSGDMA のレジスタ情報などの詳細については、Qsys ペリフェラルのユーザ・ガイドの Modular Scatter-Gather DMA (mSGDMA) のページを参照ください。

[https://www.altera.com/en\\_US/pdfs/literature/ug/ug\\_embedded\\_ip.pdf](https://www.altera.com/en_US/pdfs/literature/ug/ug_embedded_ip.pdf)

以下に、図 4-14 の波形内容に基づいて Control\_Register\_Access、および、DMA Dispatcher の CSR、Descriptor への各種設定を箇条書きで示します。

#### 【アドレス変換テーブル設定】

1. BAR2 ライト: アドレス=0x1000、データ=0xFFFFFFFFC

BAR2 の 0x1000 番地は、Control\_Register\_Access の Avalon-MM ⇒ PCI-Express アドレス変換テーブルを示しています。RxmByteEnable\_2\_o = 0x0F なので、RxmWriteData\_2\_o の 8 バイトのうち下位の 4 バイトの 0xFFFFFFFFC(上位側の 0xFFFFFFFFC は無効データ)を書き込んでいます。ここでは、アドレス変換テーブルの領域の大きさを読み出すために、0xFFFFFFFFC を書き込んでいます。

2. BAR2 リード: アドレス=0x1000、データ=0xFF000000

Control\_Register\_Access の Avalon-MM ⇒ PCI-Express アドレス変換テーブルから 0xFF000000 を読み出しています。1で 0xFFFFFFFFC を書き込み、読み出した結果が 0xFF000000 であることから、この領域の大きさは、0x000000~0xFFFFFFFF の 16MByte であることが確認できました。

3. BAR2 ライト: アドレス=0x1000、データ=0x00000000

Control\_Register\_Access の Avalon-MM ⇒ PCI-Express アドレス変換テーブルに 0x00000000 を書き込んでいます。

4. BAR2 ライト: アドレス=0x1004、データ=0x00000000

RxmAddress\_2\_o = 0x1000、RxmByteEnable\_2\_o = 0xF0 となっているので、書き込むデータは RxmWriteData\_2\_o の 8 バイトのうち上位 4 バイトとなり、この場合、アドレスは、0x1004 となります。

したがって、ここでは、Control\_Register\_Access の 0x1004 番地の Avalon-MM ⇒ PCI-Express アドレス変換テーブルに 0x00000000 を書き込んでいます。

3、4より、Avalon-MM ⇒ PCI-Express アドレス変換テーブルには、0x0000000000000000 が書き込まれます。したがって、図 3-2 で説明したように、Avalon-MM アドレスから Tx\_Interface を経由して PC 側にアクセスする場合には、下位 24 ビットの 16MByte のアドレス値に上位の 0x0000000000 が付加されてアクセスされることとなります。

#### 【DMA コントローラとステータスのクリア処理】

##### 5. BAR2 リード: アドレス=0x6000000、データ=0x00000002

BAR2 の 0x6000000 番地は、DMA Dispatcher の CSR Status Register を示しています。ここでは、DMA Dispatcher の CSR Status Register から 0x00000002 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であることを示しています。

##### 6. BAR2 ライト: アドレス=0x6000000、データ=0x00000200

DMA Dispatcher の CSR Status Register に 0x00000200 を書き込んでいます。これは、IRQ ビットを "1" に設定して、割り込み発生ステータスをクリアしています。

##### 7. BAR2 リード: アドレス=0x6000000、データ=0x00000002

DMA Dispatcher の CSR Status Register から 0x00000002 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であることを示しています。

##### 8. BAR2 ライト: アドレス=0x6000004、データ=0x00000002

BAR2 の 0x6000004 番地は、DMA Dispatcher の CSR Control Register を示しています。ここでは、DMA Dispatcher の CSR Control Register に 0x00000002 を書き込んでいます。これは、DMA Dispatcher のリセットの実行を示しています。

##### 9. BAR2 ライト: アドレス=0x6000000、データ=0x00000000

DMA Dispatcher の CSR Status Register に 0x00000000 を書き込んでいます。ここでは、DMA Dispatcher の CSR Status Register の全てのステータスをクリアしています。

#### 【割り込みの有効化処理】

##### 10. BAR2 ライト: アドレス=0x6000004、データ=0x00000010

DMA Dispatcher の CSR Control Register に 0x00000010 を書き込んでいます。ここでは、Global Interrupt Enable Mask を有効にして、割り込みを利用可能にしています。

##### 11. BAR2 リード: アドレス=0x6000000、データ=0x00000002

DMA Dispatcher の CSR Status Register から 0x00000002 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であることを示しています。

##### 12. BAR2 リード: アドレス=0x6000004、データ=0x00000010

DMA Dispatcher の CSR Control Register から 0x00000010 を読み出しています。これは、Global Interrupt Enable Mask の有効を示しています。

##### 13. BAR2 ライト: アドレス=0x6000004、データ=0x00000010

DMA Dispatcher の CSR Control Register に 0x00000010 を書き込んでいます。ここでは、Global Interrupt Enable Mask を有効にして、割り込みを利用可能にしています。

**【DMA コントローラとステータスのクリア処理】**

## 14. BAR2 リード: アドレス=0x6000000、データ=0x00000002

DMA Dispatcher の CSR Status Register から 0x00000002 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であることを示しています。

## 15. BAR2 ライト: アドレス=0x6000000、データ=0x00000000

DMA Dispatcher の CSR Status Register に 0x00000000 を書き込んでいます。ここでは、DMA Dispatcher の CSR Status Register の全てのステータスをクリアしています。

## 16. BAR2 リード: アドレス=0x6000000、データ=0x00000002

DMA Dispatcher の CSR Status Register から 0x00000002 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であることを示しています。

## 17. BAR2 リード: アドレス=0x6000000、データ=0x00000002

DMA Dispatcher の CSR Status Register から 0x00000002 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であることを示しています。

※ ソフトウェア側では、2ビット目の“Descriptor Buffer Full”が”1”(0x00000004)であるかを確認し、”0”になるまで繰り返し処理を行っています。

### 【データ転送処理】

#### 18. BAR2 ライト:アドレス=0x6000020、データ=0x00FFB000

BAR2 の 0x6000020 番地は、DMA Dispatcher の Descriptor を示しています。DMA Dispatcher の Descriptor に 0x00F63000 を書き込んでいます。これは、Descriptor Format の 0x0 番地に Read Address(このケースでは、PC 側のデータ読み出し元アドレス)を指定していることを示しています。なお、この場合のアドレス値 0x00FFB000 は、PC 側がアロケートしたメモリ・アドレスなので、実行毎に値が変わります。

#### 19. BAR2 ライト:アドレス=0x6000024、データ=0x08000000

DMA Dispatcher の Descriptor に 0x08000000 を書き込んでいます。これは、Descriptor Format の 0x4 番地に Write Address(このケースでは、外部 DDR3 SDRAM の先頭アドレス)を指定していることを示しています。

#### 20. BAR2 ライト:アドレス=0x6000028、データ=0x00004000

DMA Dispatcher の Descriptor に 0x00040000 を書き込んでいます。これは、Descriptor Format の 0x8 番地に Length(転送データサイズ)を指定していることを示しています。

#### 21. BAR2 ライト:アドレス=0x600002C、データ=0x81000000

DMA Dispatcher の Descriptor に 0x81000000 を書き込んでいます。これは、Descriptor Format の 0xC 番地に Control(DMA 制御情報)を指定していることを示しています。最上位ビットが DMA スタート・ビットとなるため、DMA 転送を開始していることを示しています。

#### 22. Tx Interface リード:アドレス(TxsAddress\_i)=0x00FFB000、バースト長(Txs\_BurstCount\_i)=0x20

18 で設定したアドレス値 0x00FFB000 に対して、バースト・リード要求が発行されていることを示しています。このアドレス値は、Avalon-MM ⇒ PCI-Express アドレス変換テーブルで 0x0000000000FFB000 の 64 ビットのアドレス値に変換されてメモリ・リード要求としてコマンド発行されます。

#### 23. Tx Interface リード:アドレス=0x00FFB000~、バースト長=0x20

その後、続けてアドレス 0x00FFB200、0x00FFB400、0x00FFB800 番地でバースト長=0x20 のバースト・リード要求が続けて発行されます。これは、トータル・データ・サイズが 0x4000 になるまで継続されます。

### 【データ転送完了処理】

#### 24. BAR2 リード:アドレス=0x6000000、データ=0x00000003

22、23 と並行して、DMA Dispatcher の CSR Status Register から 0x00000003 を読み出しています。これは、ディスクリプタ・テーブル格納用バッファがエンプティ状態であり、DMA エンジンが Busy であることを示しています。すなわち、18~21 で指定した DMA ライト転送の処理が完了し、ディスクリプタ・テーブル格納用バッファがエンプティ状態になり、次の DMA 転送要求が受け付け可能状態になったことを意味します。ここでいう DMA エンジンの Busy 状態は DMA の Read Master がホスト(PC 側)からデータ読み出しているため、Busy 状態となります。

※ 17 と同様、ソフトウェア側では、2 ビット目の “Descriptor Buffer Full” が ”1” (0x00000004) であるかを確認し、”0” になるまで繰り返し処理を行っています。



**【DMA 転送完了判定用データ転送処理】****25. BAR2 ライト:アドレス=0x6000020、データ=0x08000000**

DMA Dispatcher の Descriptor に 0x08000000 を書き込んでいます。これは、Descriptor Format の 0x0 番地に Read Address(外部 DDR3 SDRAM の先頭アドレス)を指定していることを示しています。

**26. BAR2 ライト:アドレス=0x6000024、データ=0x00FFF000**

DMA Dispatcher の Descriptor に 0x00FA3000 を書き込んでいます。これは、Descriptor Format の 0x4 番地に Write Address(PC 側のデータ書き込み先アドレス)を指定していることを示しています。

**27. BAR2 ライト:アドレス=0x6000028、データ=0x00000020**

DMA Dispatcher の Descriptor に 0x00000020 を書き込んでいます。これは、Descriptor Format の 0x8 番地に Length(転送データサイズ)を指定していることを示しています。

**28. BAR2 ライト:アドレス=0x600002C、データ=0x80004000**

DMA Dispatcher の Descriptor に 0x80004000 を書き込んでいます。これは、Descriptor Format の 0xC 番地に Control (DMA 制御情報)を指定していることを示しています。最上位ビットが DMA スタート・ビットとなるため、DMA 転送を開始していることを示しています。また、14ビット目の“Transfer Complete IRQ Mask”が Hi となっているため、この DMA ライト転送が完了した時点で割り込みが発行されます。

この 25~28 の処理は、DMA リードで外部 DDR3 SDRAM に書き込んだデータの先頭部分の 0x20 バイト分のデータを PC 側に DMA ライトで書き戻しています。PC 側では、このデータを使用して DMA 転送が完了したことを判定しています。図 4-15 は、この DMA ライト完了時点の波形となります。一連の DMA リード転送が完了した後に、割り込み信号 RxmIrq\_i[0] がアサートされ 0x20 バイト分のデータを PC 側へ DMA ライトで転送しています。



## 5. まとめ

今回は、アルテラ社 PCI-Express ハード IP を使用した Qsys システム統合ツールでの DMA 転送の実現(ハードウェア編)ということで、Cyclone V GT 評価用ボードベースのリファレンス・デザインを用いて Qsys 内部の各コンポーネントの実装方法の詳細や PCI-Express Avalon-MM ブリッジの詳細、SignalTap II を使用した実機での動作確認を行いました。この手法によるメリットは、何と云ってもこれまでいろいろな意味でハードルの高かった PCI-Express の FPGA 設計を簡単にしたことに尽きると思います。主なメリットを以下に挙げておきます。

### 1. 簡単な設計手法

ハードウェア側の設計に限りますが、今回紹介したリファレンス・デザイン程度のデザインならば1時間程度の時間で実現できるレベルです。ユーザは、システム内部の細かな接続やクロックの載せ替えなどのロジックを作りこむ必要がなく、簡単に実現できます。

### 2. PCI-Express の細かい規格を知る必要がない

アルテラ社のハード IP を IP Catalog(旧 MegaWizard)ベースで設計した場合、Avalon-ST 形式のデータのやり取りになりますが、この場合、実際にはストリーム・パケットの内容は PCI-Express のトランザクション・レイヤ・パケット(TLP)となっているので、結局、TLP の詳細を理解する必要があります。Qsys ベースの設計手法では、この TLP の処理は、PCI-Express Avalon-MM ブリッジ内で吸収されており、ユーザは TLP の詳細を一切知らなくてもデータのやり取りができます。

### 3. 高いパフォーマンス

PCI-Express などの高帯域の要求に答えるために Qsys では、ネットワーク・プロセッサなどで使用されているパケットベース・アーキテクチャを採用し、現時点では、Stratix V GX デバイスを用いて PCI-Express の Gen3、x8 の接続までができます。

### 4. DMA 転送のソリューションを無償で提供

今回紹介した mSGDMA は、無償で高パフォーマンスの DMA 転送のソリューションを実現できます。また、今回は、Avalon-MM to Avalon-MM での転送タイプでしたが、Avalon-ST to Avalon-MM の転送にも対応しているので、様々なアプリケーションに適用できる DMA エンジンとなっております。

## 改版履歴

Revision	年月	概要
1	2014 年 10 月	初版
2	2014 年 12 月	「4-5. 外部 DDR3 SDRAM への DMA リード」の「22. TxS_Interface リード : アドレス ( TxSAddress_i ) = 0x00FFB000 、バースト長 ( TxS_BurstCount_i ) = 0x20 」の説明内のアドレス値を 0x0000000000F63000 から 0x0000000000FFB000 へ誤記訂正
3	2015 年 4 月	アルテラ社の Web サイトのリニューアルに伴う URL 変更 誤記訂正 P.12 誤) v14.0 以前のバージョン ⇒ 正) v14.0 より前のバージョン

### 免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。  
株式会社アルティマ ホームページ: <http://www.altima.co.jp> 技術情報サイト EDISON: <https://www.altima.jp/members/index.cfm>  
株式会社エルセナ ホームページ: <http://www.elsena.co.jp> 技術情報サイト ETS : <https://www.elsena.co.jp/elspear/members/index.cfm>
4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる場合は、英語版の資料もあわせてご利用ください。