

# SATA-IP ホスト・リファレンス・デザイン説明書

Rev2.0J 2017/03/28

## 1. SATA について

シリアル ATA (SATA)は従来のパラレル ATA(PATA)に替わる革新的なストレージ・インターフェイスです。最新の SATA インターフェイスにおける転送速度は、SATA-I 規格の 1.5Gbps から SATA-III 規格の 6.0Gbps に高速化されています。SATA プロトコルによる通信システム全体としては、図 1 に示すように、アプリケーション・レイヤ、トランスポート・レイヤ、リンク・レイヤ、物理 (PHY)レイヤ、の4レイヤにより実装されるアーキテクチャとなります

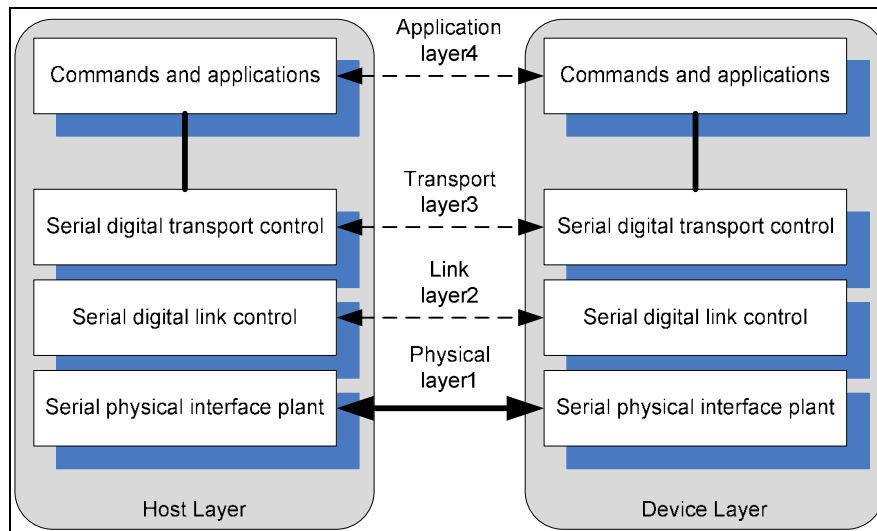


図 1-1: SATA のレイヤ構造

アプリケーション・レイヤはコマンド・ブロック・レジスタの制御を含む ATA コマンドの実行を担当します。トランスポート・レイヤではパケットや FIS(Frame Information Structure)と呼ばれるフレームによってホスト~デバイス間で転送される制御情報やデータを管理します。リンク・レイヤにおいては、生成されたフレームをもとにバイトごとの 8b/10b エンコード/デコードの実行や、10 ビットのデータ・ストリームが受信側で正しくデコードされるよう制御キャラクタの挿入を行います。PHY(物理)レイヤは、シリアル・データとして外部信号線上に流れるエンコード情報を送受信します。

本リファレンス・デザインでは、ホスト側において SATA-III または SATA-II デバイスと高速データ転送を行うため、SATA-IP を含めた全 SATA 通信レイヤの具体的な実装方法例を紹介します。SATA-IP コアは下位 PHY レイヤにて 7 シリーズや UltraScale デバイスの GTP/GTX/GTH トランシーバと、上位レイヤに MicroBlaze と組み合わせることで SATA チャネルを構築します。より詳細については以下で説明します。

## 2. 動作環境

本リファレンス・デザインは図 2-1 に示す以下の環境で動作します。

- Xilinx 製 AC701/KC705/ZC706/VC707/VC709/KCU105 各評価ボード
- Bit ファイルをダウンロードするための Vivado/ISE ツール
- AB09-FMCRAID アダプタ基板  
注: 本アダプタ基板の入手方法については DesignGateway 社までお問い合わせください。
- SATA-III または SATA-II 対応の HDD または SSD (ドライブ用 ATX 電源も必要)
- Xilinx 評価ボードに標準添付の USB ケーブル 2 本 (JTAG およびシリアル通信) シリアル通信は、  
ボーレート=115,200 / データ=8bit / パリティ=なし / Stop ビット=1bit、に設定してください。

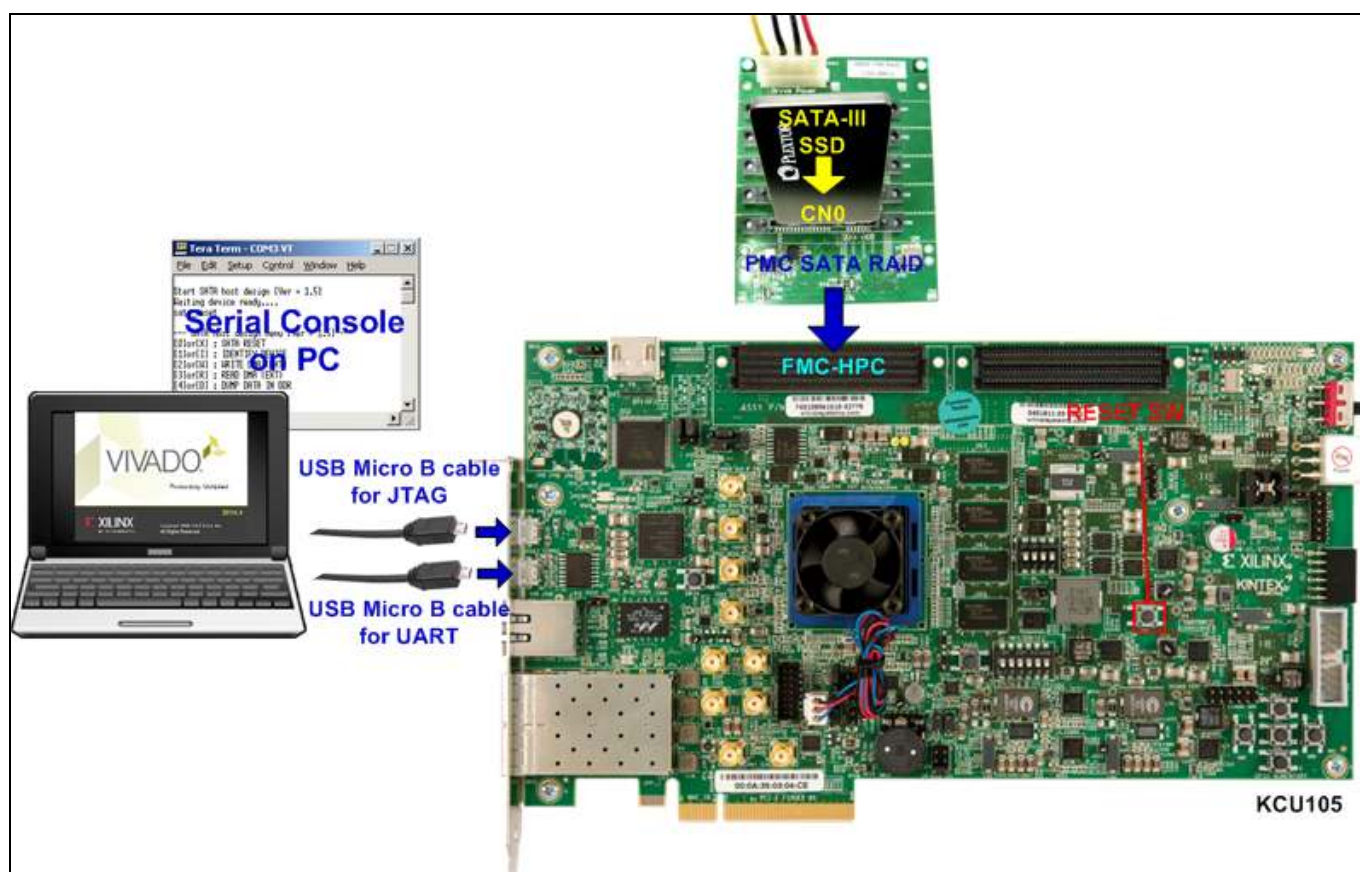


図 2-1: リファレンス・デザインの動作環境

### 3. ハードウェアの説明

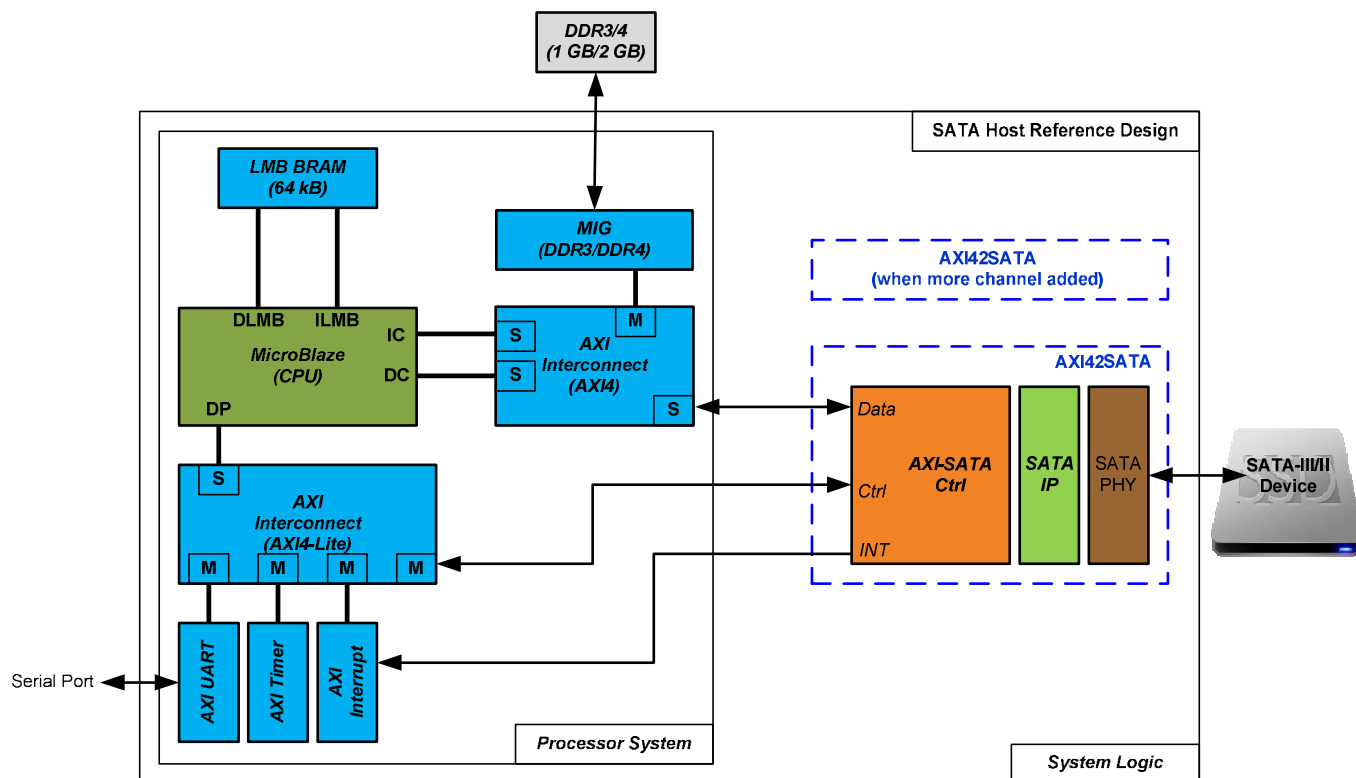


図 3-1: ZC706 以外のホスト・リファレンス・デザインのブロック図

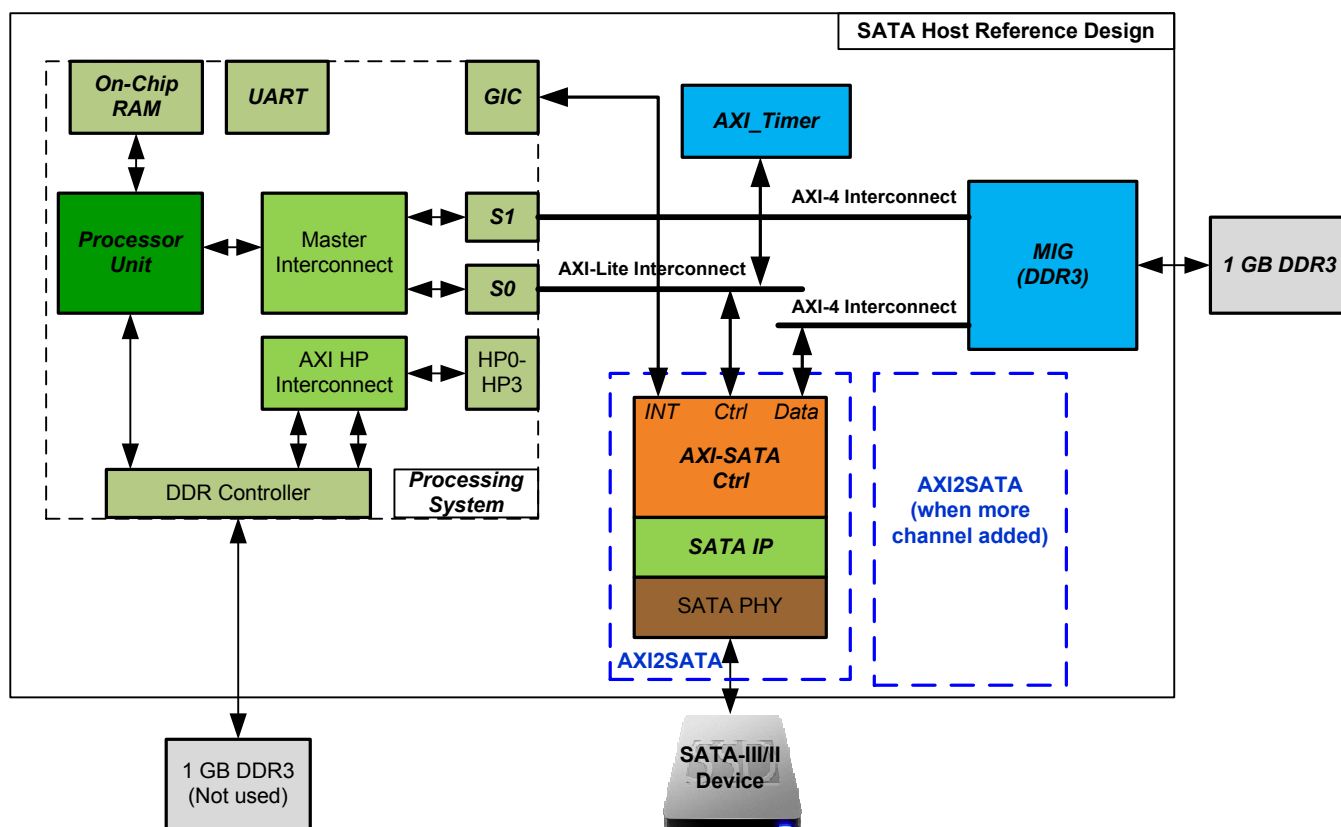


図 3-2: ZC706 のホスト・リファレンス・デザインのブロック図

### ● FPGAによるSATA-IPのホスト機能

図 3-1 または図 3-2 に示す通り、本デザインは大きくは 2 つのブロックで構成されます、すなわち一つはプロセッサや周辺モジュールを含むプロセッサ・システムで、もう一つは本ハードウェアで説明する AXI42SATA モジュールです。

プロセッサ・システムは UART、タイマー、割り込みコントローラ、DDR3/4 コントローラ、プロセッサのファームウェアやデータを格納するためのブロック RAM などの基本コンポーネントを含みます。そしてプロセッサの AXI4-Lite および AXI4 インターフェイスによってトップ階層でハードウェア・ロジックと接続します。

システム・ロジックはプロセッサ・システムのラップと 2 種類のバス・インターフェイスを介して接続する AXI42SATA ロジックを含む HDL コードです。AXI4-Lite バスは AXI42SATA を制御するプロセッサ (MicroBlaze または ARM) のインターフェイスです。一方 AXI4 バスはプロセッサと AXI42SATA モジュール間のデータ・バッファとして使われる DDR3/DDR4 のインターフェイスです。AXI42SATA モジュールをシステム・ロジック上で複製しプロセッサ・システムと接続する AXI4-Lite と AXI4 バスのポートを拡張することで SATA チャンネルを増やすことができます。より詳細には以下で説明します。

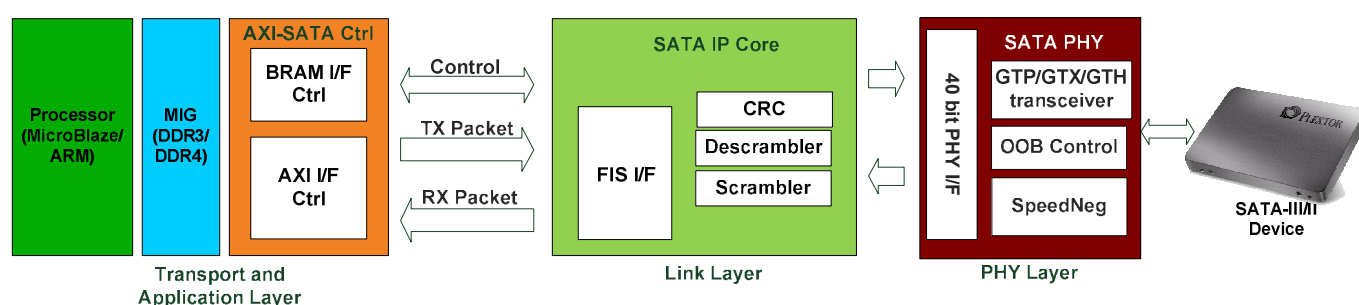


図 3-3: ホスト・リファレンス・デザインの各 SATA レイヤ構成

### ● PHY (物理) レイヤ

このレイヤは 7 シリーズ/UltraScale デバイスの GTP/GTX/GTH モジュール (ビルトイン高速シリアル回路) でデザインされ、物理レイヤの OOB (Out-of-band) シーケンスおよび初期化シーケンスを生成するロジックと合わせて動作します。OOB シーケンスを制御するステート・マシンは "oob\_control.vhd" 内で実装され、PHY レイヤのソースコードにてトップ階層の "sata2phy\_xx.vhd" (xx は FPGA のモデル名が入ります) のサブ・モジュールです。SATA-II と SATA-III の両スピードに対応するため、AC701/KC705/ZC706/VC707 の PHY ソースコードには "speed\_neg\_control.vhd" が内蔵され、SATA3 と SATA2 の速度を自動ネゴシエーションで選択するモジュールも含まれます。

リファレンス・デザインには PLL およびトランシーバのリセット・シーケンスが含まれますが、それは Xilinx のトランシーバ・ユーザ・ガイドの章に沿ったデザインです。AC701 ボードの場合リセット・シーケンスは "gtp\_rxreset\_seq.vhd" モジュール内で記述され、その他のデバイスでは "sata2phy\_xx.vhd" 内で記述されています。

ユーザ自身が新規にボードをデザインする場合、事前に以下のトランシーバ・デザイン・ガイドの "Board Design Guidelines" の章を熟読し、必ずその指示に沿って設計しなくてはなりません。各トランシーバのユーザ・ガイドは以下のドキュメントとなります。

- UG482: 7 Series FPGAs GTP Transceivers User Guide (AC701)
- UG476: 7 Series FPGAs GTX/GTH Transceivers User Guide (KC705/ZC706/VC707/VC709)
- UG576: Ultrascale GTH Transceivers User Guide (KCU105)

● SATA-IP によるリンク層

SATA-IP はリンク・レイヤ全てと SATA プロトコルにおけるトランスポート・レイヤの一部を含みます。 FIS パケットのフォーマットはCRCの追加とスクランブル処理が行われ下位層に向けて変換されます。また、物理層から送られてきたパケットはデコードされ FIS パケット・フォーマットとして構築されてからユーザ・インターフェイスに転送されます。SATA-IP コアのインターフェイス信号詳細仕様や信号波形については、SATA-IP データシート“dg\_sata\_ip\_datasheet\_7series\_jp.pdf”を参照してください

● AXI-SATActrl によるトランスポート・レイヤ

AXI-SATActrl モジュールのブロック図を下図 3-4 に示し、以下に説明します。

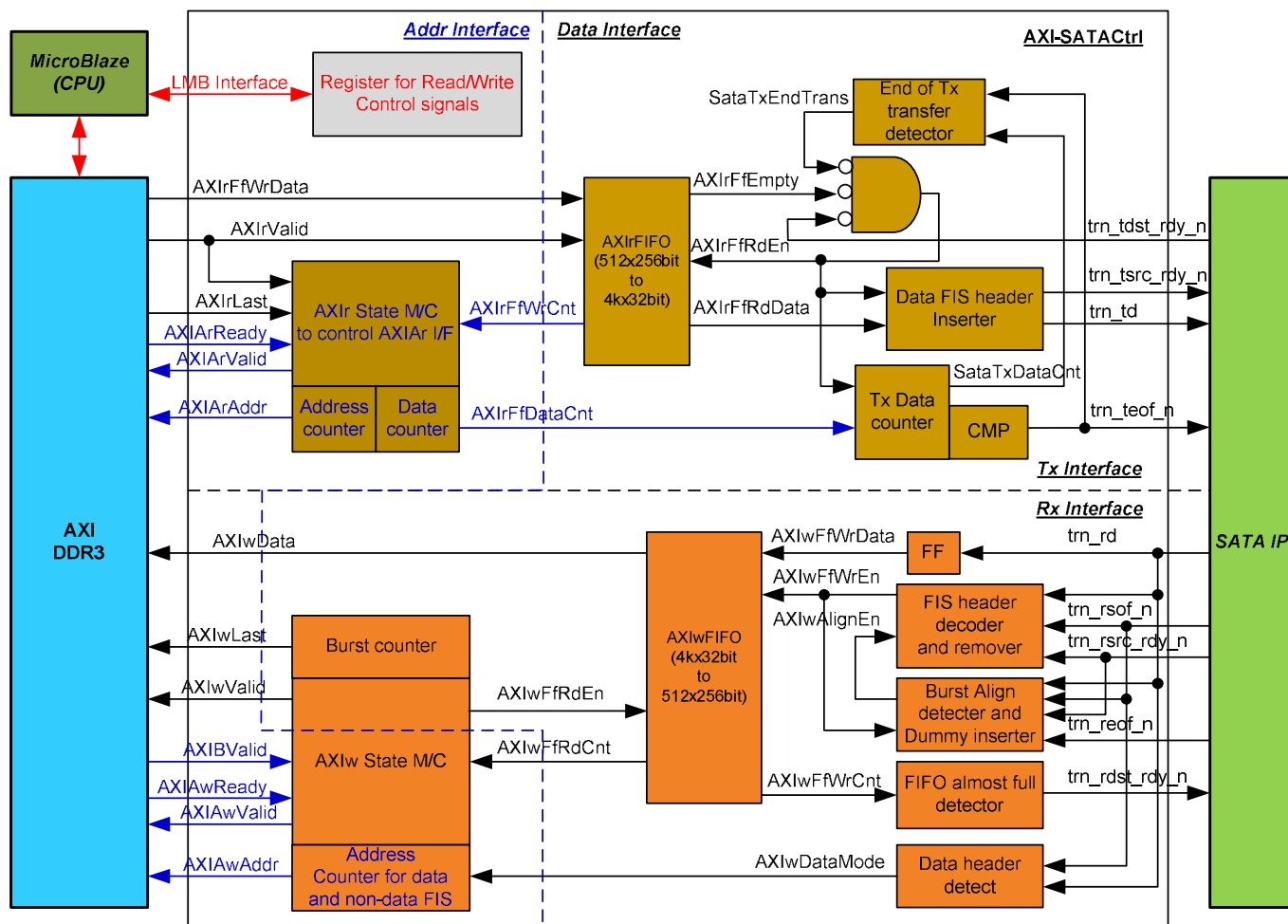


図 3-4: AXI-SATActrl のブロック図

AXI-SATActrl は SATA-IP をプロセッサ・システムの標準インターフェイスすなわち制御インターフェイス用の AXI4-Lite バスおよびデータ・インターフェイス用の AXI4 バスと接続するために設計されたモジュールです。

プロセッサ(MicroBlaze/ARM)から制御するレジスタのマッピングを表 3-1 に示します。これらのレジスタは主として転送する FIS データの DDR アドレスや転送長や FIS タイプ(すなわちデータか非データかの種別)を指定するためのものです。また、プロセッサから AXI-SATA Ctrl モジュールのステータスをリードするためのレジスタも実装されています。

アドレス Rd/Wr	レジスタ名 ("sata_host.c" でのラベル名)	説明 (Bit ならばはリトル・エンディアンで表記)
BA+0x04 Rd	エラー・コード・レジスタ (ERROR_CODE)	SATA IP のエラーコード。送信完了および受信完了時にセットされる。CRC エラーや FIS エラーはここで判別できる。
BA+0x0C Rd	受信ワード数レジスタ (RX_COUNT)	[31] : 割り込み発生時の受信 FIS タイプ、'1'は非データ FIS、'0'はデータ FIS。このビットは INT_CLEAR レジスタの bit[29]に'1'をライトすることでクリアされる。 [23:0]: 受信した FIS データの総ワード数。コントロール・レジスタにて次の転送開始が指示されるとゼロクリアされる。
BA+0x00 Wr	送信データ格納アドレス (TX_ADDR)	送信する DDR3/4 データを格納している先頭のアドレスを指定する。 本レジスタの Bit[8:0]はセクタのアライメントに合うよう必ずゼロにセットすること。
BA+0x04 Wr	受信データ格納アドレス 1 (RX_ADDR)	受信したデータのうち、データ FIS 以外のものが格納される DDR3/4 のアドレス。 本レジスタの Bit[8:0]はセクタのアライメントに合うよう必ずゼロにセットすること。
BA+0x08 Wr	コントロール・レジスタ (CONTROL)	[31]ハードウェア・リセット [30]送信要求 [29]FIS タイプ(データ FIS を送る場合'1'それ以外の FIS タイプの場合'0') [15:0]送信データワード数。 本レジスタの書き込みで RX_COUNT レジスタがリセットされる。
BA+0x0C Wr	受信データ格納アドレス 2 (RX2_ADDR)	受信したデータのうち、データ FIS が格納される DDR3/4 のアドレス。 本レジスタの Bit[10:0]は必ずゼロにセットすること。
BA+0x10 Wr	割り込みクリア・レジスタ (INT_CLEAR)	[31] : ip2host 割り込みをクリアするときに'1'をセット [30] : host2ip 割り込みをクリアするときに'1'をセット [29] : 受信 FIS タイプをクリアするときに'1'をセット

(BA : ベース・アドレス)

表 3-1: CPU からのレジスタ・マッピング

DDR3/4 とデータを転送するため、AXI4 バスは 4 グループに分類されます、すなわちリード命令要求である AXIAr、リード転送データである AXIr、ライト命令要求である AXIAw、ライト転送データである AXIw の 4 種類です。ライト・リードそれぞれの転送方向での命令要求は内部ステート・マシンが制御します。AXI4 のデータ・バス幅は 256 ビットですが SATA-IP コアは 32 ビット幅です、このためデータ・バス幅変換とフロー制御を兼ねて FIFO が使われています。リード・ライトの動作シーケンスを以下に説明します。

AXI バスから SATA へのリード動作ではプロセッサによって動作が開始されます。プロセッサが FIS やデータを DDR3/4 に準備した後、制御レジスタをパケット・サイズと合わせてセットしデータ FIS または非データ FIS の SATA-IP コアへの転送を開始します。データ FIS が送信された場合データ FIS は内部ロジックにより自動的に付加され DDR3/4 からのデータが続きます。データ・バス幅変換とフロー制御のため AIXrFIFO が使われます。FIFO がほぼ一杯になったことを示す Almost full 状態になると、ステート・マシンは DDR/4 からの次のデータ要求を保留し FIFO 空間が十分に空くの待ちます。また、SATA 側のロジックも FIFO が空になったことを示す Empty フラグをモニタし SATA-IP コアへとデータを送信します。このようなシーケンスでプロセッサは指定の FIS サイズでどのような FIS タイプでも簡単に生成できます。

SATA-IP コアから AXI へのライト動作では SATA から動作が開始されます。SATA からの FIS を格納する DDR3/4 アドレスはプロセッサによって予めセットされます。図 3-5 に示す通り、データ FIS と非データ FIS それぞれのパケットごとに異なる 2 つの DDR3/4 空間が定義されます。動作モードによりデータ FIS は RX\_DATA\_ADDR 空間に格納される一方、非データ FIS は RX\_FIS\_ADDR 空間に格納されます。データ・フロー制御とデータ・バス幅変換を兼ねて AXIwFIFO が使われます。FIFO が Almost full 状態になると SATA-IP コアから転送される SATA パケットは一時停止します。SATA-IP コアからの SATA パケットの全データ・サイズが小さく 256 ビットに満たずワードのアライメントに合わない場合、内部ロジックによりダミー・データが FIFO に補填されます。リード側では常に FIFO カウンタがモニタされ、データが十分に用意されていることをチェックし、AXI4 バスを通じた DDR3/4 へのデータ転送を要求します。各 SATA パケットの FIS ヘッダはデコードされ、FIS タイプが非データ型かデータかをチェックし、SATA FIS を DDR3/4 の正しい格納位置にセットします。

データ・トランザクションの両方向ともデザインをシンプルにし高い転送パフォーマンスを実現するため 256 ビットで 16 ビート(512 バイト)の固定となります。SATA-IP コアや PHY 層合わせて動作する AXI-SATActrl ロジックの HDL コードは SATA-IP コア納品物の"AXI42SATA.vhd"内に格納されています。

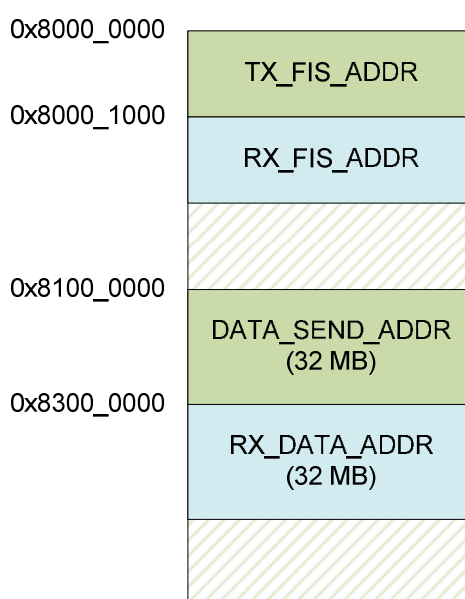


図 3-5: リファレンス・デザインの DDR3/4 メモリ・マップ

- TX\_FIS\_ADDR は、プロセッサから SATA デバイスへ送信する非データ FIS を格納します。
- RX\_FIS\_ADDR は、SATA デバイスからプロセッサへ受信される非データ FIS を格納します。
- DATA\_SEND\_ADDR は、プロセッサから SATA デバイスへ送信するデータ FIS を格納します。
- RX\_DATA\_ADDR は、SATA デバイスからプロセッサへ受信されるデータ FIS を格納します。

## 4. ソフトウェアの説明

- FIS を介した SATA デバイスへのアクセス

SATA によるホストとデバイス間の通信は FIS(Frame Information Structure)データ構造によって実行されます。ホストデザインの MicroBlaze はメインメモリ上に FIS データを構築し、バス・マスタとなる DMA によってデバイスに転送されます。また、デバイスからの FIS データも同じように DMA によってメインメモリに転送されます。

従って MicroBlaze は以下の手順で SATA デバイスへのアクセスを実行します。

- (1) FIS データ・ストラクチャを作成します。最初の FIS コマンドは RegH2D FIS とする必要があります。
- (2) FIS データを転送します。
- (3) デバイスからの FIS データ受信を待ちます。
- (4) 受信した FIS データを読み取り、解析します。
- (5) 必要に応じて追加の FIS データの送受信を行います。

プロトコルによって送信する FIS の数や受信する FIS の数は異なりますが、ほぼこのような流れになります。

- リファレンス・デザインのソフトウェア

本リファレンス・デザインのソフトウェアは一般的な 3 コマンドを実装しており、それは IDENTIFY DEVICE, READ DMA EXT/READ DMA, WRITE DMA EXT/WRITE DMA となります。本リファレンス・デザインは 48 ビット LBA (LogicalBlock Address)モードと 28 ビット LBA モードの両方をサポートしています。

デバイスがパワー ON したとき、デバイスは必ず Register –Device to Host FIS を最初に送ります。従って、ホストは最初のコマンドを発行する前にデバイスからの RegD2H FIS を待つ必要があります。

- IDENTIFY DEVICE

表 4-1 は SATA デバイスからデバイス情報を取得するための IDENTIFY COMMAND の FIS 構造です。コマンドは ECh で、あとはデバイス番号を設定するだけで実行できます。SATA の場合、デバイス番号は通常 '0' になります。

なお、Device レジスタの 5 ビットと 7 ビットは obsolete(廃止)ビットですが、慣例では常に 1 にセットするようのでここは A0h を設定します。また C ビットを 1 にします。コマンドを送信する場合はこれを必ずセットしますが、以後のコマンドも同様です。

これらの値を Register – Host to Device FIS に格納してリンク・レイヤに送信します。するとデバイスから PIO SetupFIS が送られてきたあとに Data FIS が送られてきます。この中にデバイスの情報が格納されています。デバイス情報の詳細については ATA 規格書 (<http://www.t13.org/> から入手可能)を参照してください。本リファレンス・デザインではデバイス型番、48 ビット LBA の対応情報、ディスク容量等を表示します。

0	Features 00h	command ECh	C R R R PM Port 1 0 0 0  0h	FIS Type (27h)
1	Device A0h	LBA High 00h	LBA Mid 00h	LBA Low 00h
2	Features (exp) 00h	LBA High (exp) 00h	LBA Mid (exp) 00h	LBA Low (exp) 00h
3	Control 00h	Reserved(0)	sector Count (exp) 00h	Sector Count 00h
4	Reserved(0)	Reserved(0)	Reserved(0)	Reserved(0)

表 4-1: IDENTIFY COMMAND の FIS 構造



- READ DMA EXT

表 4-2 は SATA デバイスから 48 ビット LBA でデータを読み出す READ DMA EXT 命令の FIS 構造です。28 ビット LBA では READ DMA コマンドが使われます。データ転送は大きく分けて PIO と DMA がありますが、SATA にとっては若干 FIS の手順が違っただけで、どちらもそれほど変わりません。実は PIO 転送を使っても、DMA と変わらないくらいの速度が出ますが、リードに関しては READ DMA の手順が簡単なので、こちらを使います。

コマンドは 25h (28 ビット LBA の READ DMA コマンドでは C8h)、Device レジスタ 6 ビットの LBA ビットを 1 にし、あとは LBA アドレスと読み出したいセクタ数を Register - Host to Device FIS に格納して送信します。するとデバイスから Data FIS が要求したデータ分だけ送られてきた後、Register - Device to Host FIS が送られてきて完了です。

0	Features 00h	command 25h	CR 10	RR 00	PM Port 0h	FIS Type (27h)
1	Device E0h	LBA High LBA[23:16]	LBA Mid LBA[15:8]		LBA Low LBA[7:0]	
2	Features(exp) 00h	LBA High(exp) LBA[47:40]	LBA Mid(exp) LBA[39:32]		LBA Low(exp) LBA[31:24]	
3	Control 00h	Reserved(0)	sector Count(exp) sector_count[15:8]		Sector Count sector_count[7:0]	
4	Reserved(0)	Reserved(0)	Reserved(0)		Reserved(0)	

表 4-2: DMA READ EXT の FIS 構造

- WRITE DMA EXT

表 4-3 は SATA デバイスへデータを書き込む WRITE DMA EXT 命令の FIS 構造です。(28 ビット LBA では READ DMA コマンドが使われます。) コマンドは 35h (28 ビット LBA の場合 CAh)、LBA ビットや LBA アドレス、セクタ数の設定は DMA READ EXT と同じです。その後、デバイスから DMA Activate FIS が返ってきます。それを受けてホストは最初の Data FIS を送信します。これを繰り返し全てのデータを送信し終わったら、デバイスから Register-Device to Host FIS が送られてきて完了です。

0	Features 00h	command 35h	CR 10	RR 00	PM Port 0h	FIS Type (27h)
1	Device E0h	LBA High LBA[23:16]	LBA Mid LBA[15:8]		LBA Low LBA[7:0]	
2	Features(exp) 00h	LBA High(exp) LBA[47:40]	LBA Mid(exp) LBA[39:32]		LBA Low(exp) LBA[31:24]	
3	Control 00h	Reserved(0)	sector Count(exp) sector_count[15:8]		Sector Count sector_count[7:0]	
4	Reserved(0)	Reserved(0)	Reserved(0)		Reserved(0)	

表 4-3: DMA WRITE EXT の FIS 構造

● リファレンス・デザインの動作について

本リファレンス・デザインのソフトウェアのソースコードは“sata\_host.c”内に格納されています。ただし本デザインはエラーチェックや異常発生時のリカバリなどの処理は含まれていません。従ってユーザが開発するソフトウェアにおいては、デバイスから Register- Device to Host FIS が送られたときにステータスやエラーをチェックし、必要な処理を追加する必要があります。

図 4-1 に本リファレンス・デザインを動作したときの PC 上のシリアル・ターミナル画面サンプルを示します。

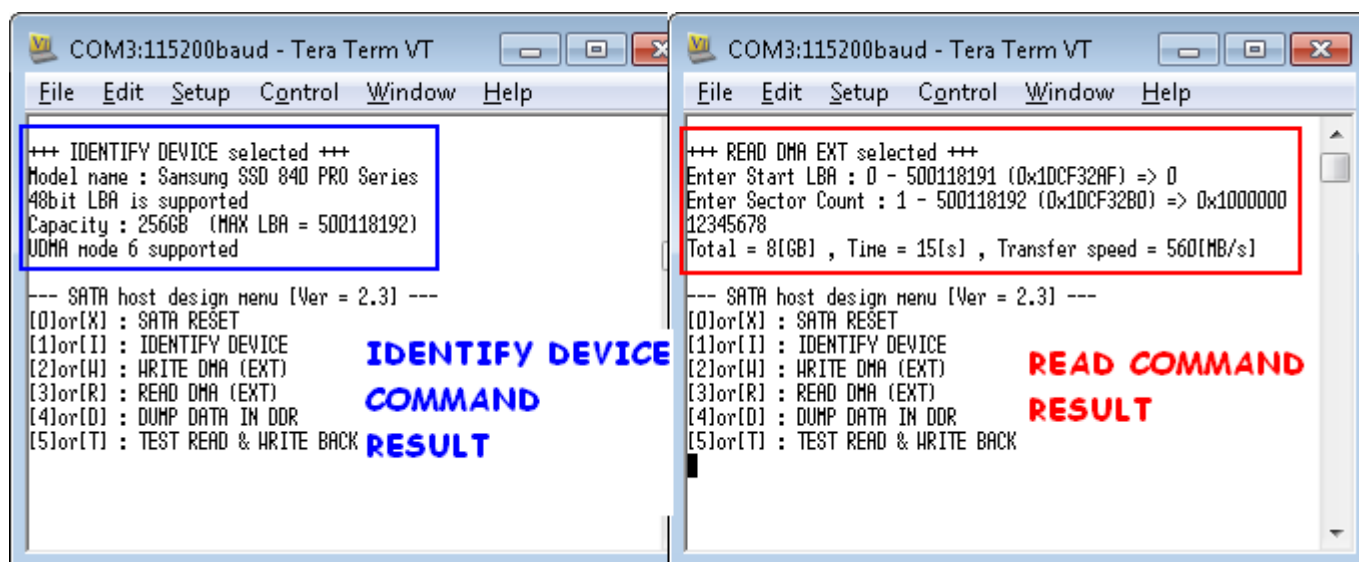


図 4-1: 実機動作時のシリアル・ターミナル画面例

## 5. 改版履歴

リビジョン	日付	履歴
1.0	21-Apr-14	Initial release
1.1	1-Jul-14	Update to Vivado2014.2 version
2.0	20-Jan-16	Support 7-series and KCU105 board
2.0J	2017/03/28	Rev2.0 英語版に対応した日本語翻訳版を作成

Copyright: 2014 Design Gateway Co.,Ltd.