

# ML605 対応 SATA-IP RAIDx4 リファレンス・デザイン説明書

Rev1.1 2010/05/26

このドキュメントは Xilinx 製 ML605 評価ボードで動作する SATA-IP ホスト向けのリファレンス・デザインに関して説明したものです。

## 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-II 規格の 3.0Gbps 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。

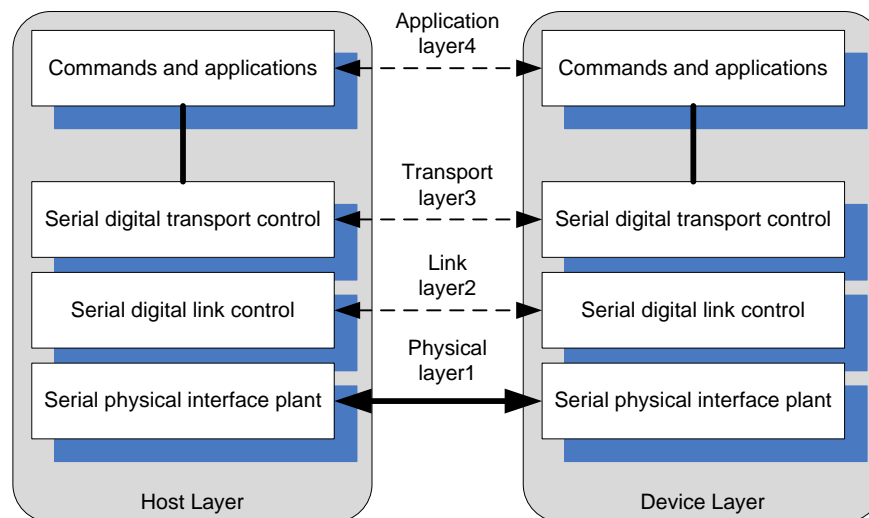


図 1: SATA のレイヤ構造

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-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。この評価システムでは外付けの SATA-II HDD または SSD 4 台を使っての RAID0 システムを構築します。SATA-IP コアは Virtex-6 デバイスの GTX トランシーバと組み合わせで動作し、本 RAID リファレンス・デザインは Xilinx 製 ML605 評価ボードを使って実装されます。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 に示される以下の環境で動作します。

- Xilinx 製 ML605 評価ボード
- ISE 11.5 / EDK 11.5
- ML605 ボードの PCIe を SATA に変換する PCIe2SATA 8 チャンネル・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-II 対応の HDD または SSD 4 台 (SATA-II 対応ケーブル 4 本とドライブ用電源も必要)
- USB ミニケーブル 2 本 (JTAG およびシリアル通信、ML605 評価ボードキットに同梱されています。) 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。 .

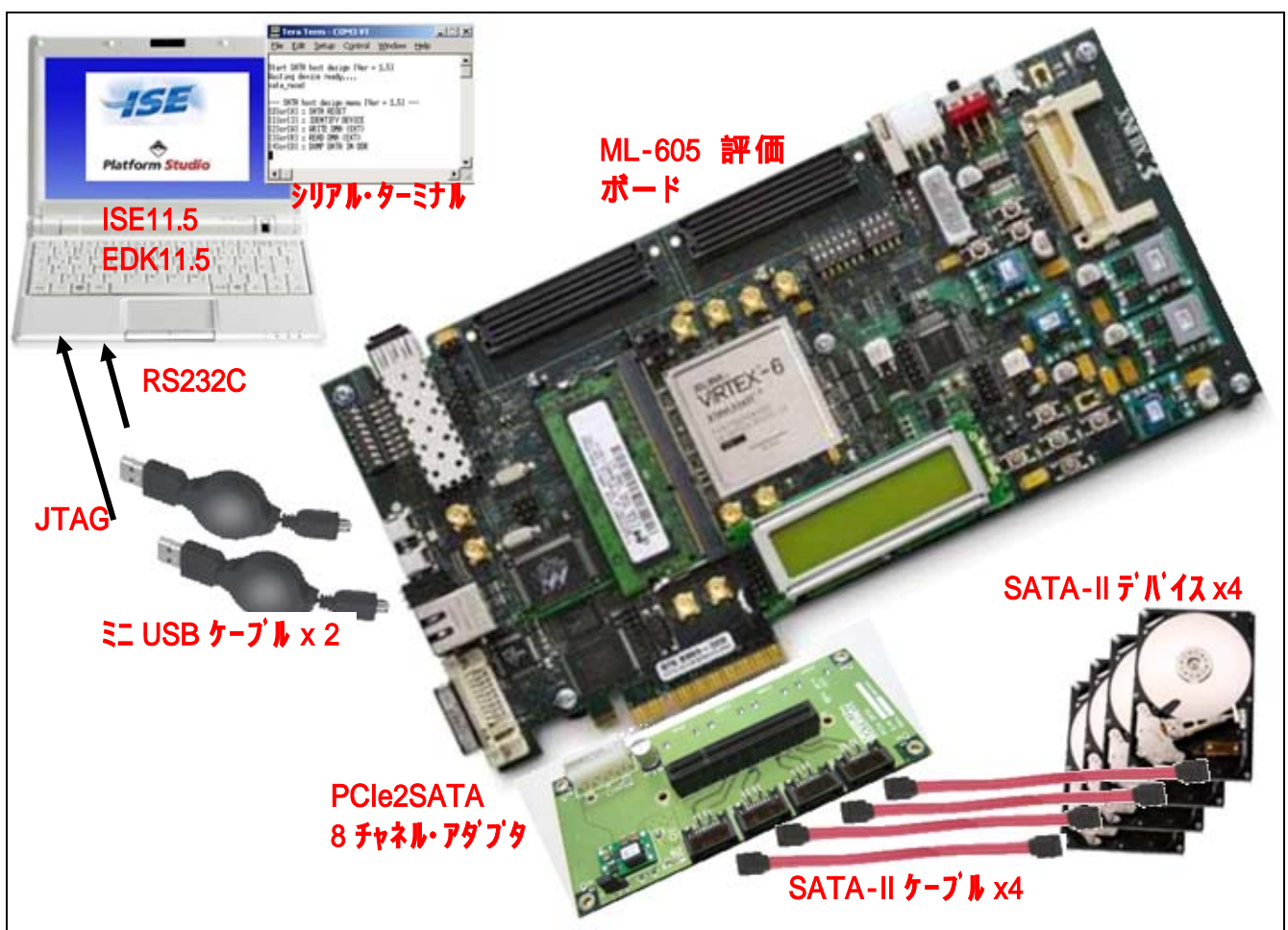


図 2: リファレンス・デザインの動作環境

本リファレンス・デザインの実機での操作方法については、デモ・マニュアル “SATA-IP RAIDx4 Demo Instruction on ML605” を参照してください。また、評価版のデザインにおいては 1 時間の動作制限があるため、1 時間経過後に動作が自動停止します。

### 3. ハードウェアの説明

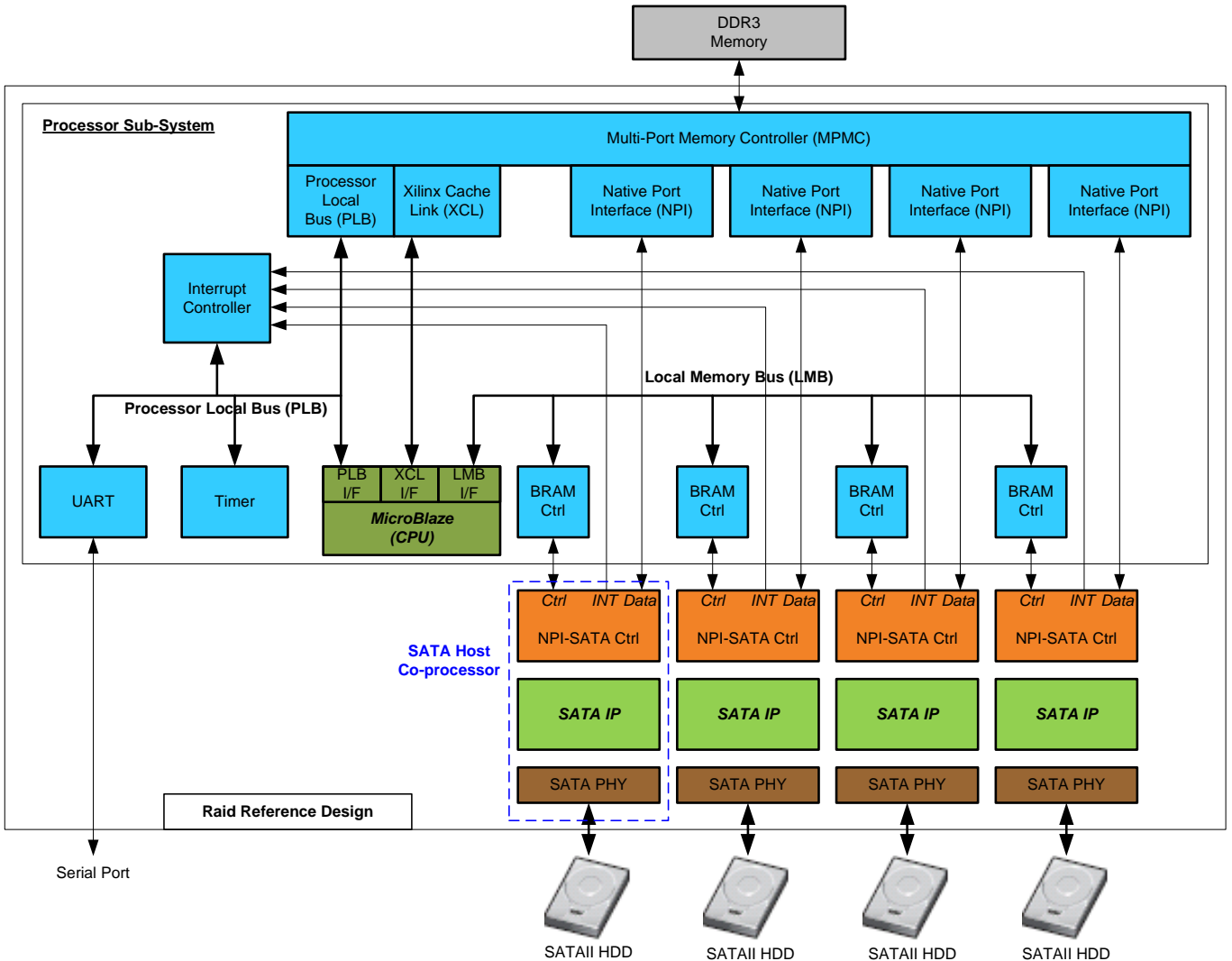


図 3: ML605 による 4 チャンネル RAID0 リファレンス・デザインのブロック図

- Virtex-6 FPGA による SATA IP の 4 チャンネル RAID0 機能

本 RAID リファレンス・デザインは図 3 に示すように、MicroBlaze、MPMC、4 チャンネルの SATA ホスト・コプロセッサ、の 3 つをメイン・モジュールとして構成します。MicroBlaze は外部の DDR3 メモリ上へのテスト・パターン・データの構築を行い、各 SATA ホスト・コプロセッサは DDR3 メモリのデータを読み出してそれぞれの SATA デバイスに転送します。

従って DDR3 メモリを制御する MPMC は、同時にアクセスする可能性のある MicroBlaze および 4 つの SATA ホスト・コプロセッサからのデータ転送に対して調停・制御を行います。結果として MPMC は、MicroBlaze からのデータアクセスとなる PLB、MicroBlaze キャッシュのための XCL、そして各 SATA チャンネル接続の 4 つの NPI、の合計 6 チャンネルのポートと接続します。

また、MicroBlaze が各 SATA ホスト・コプロセッサに対して制御する接続ロジックを簡略化しリソースを節約するため、BRAM Ctrl を使ったレジスタアクセスを使います。割り込み機能を使うことによって、MicroBlaze 上のソフトウェアは RAID0 システムとして機能する 4 つの SATA チャンネルに対して非常に優れたパフォーマンスを実現します。シリアル・ポートは RS232C 接続したパソコンからユーザ命令を受信するために使われます。

それぞれの SATA ホスト・コプロセッサは、NPI-SATA Ctrl、SATA-IP、SATA-PHY から構成されそれぞれ SATA プロトコルにおいて、トランスポート・レイヤ、リンク・レイヤ、物理レイヤを構築します。より詳細については下図 4 をもとに以下に説明します。

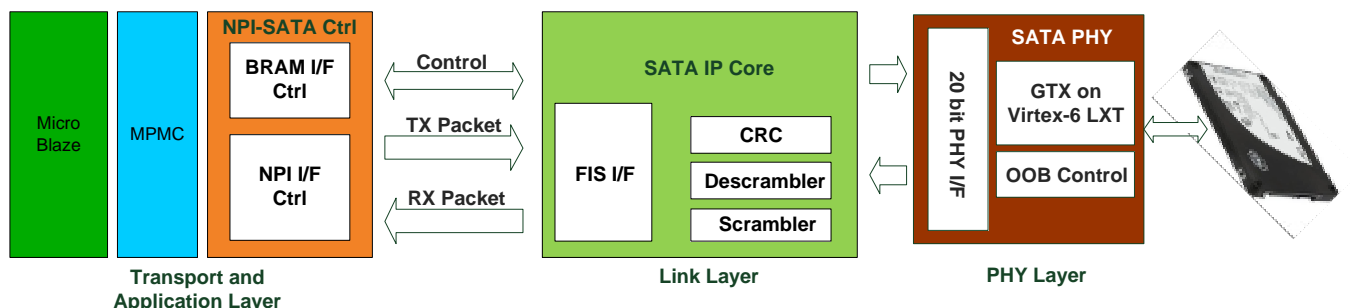


図 4: SATA プロトコルでの各レイヤ間接続

#### ● SATA-PHY

本デザインの SATA-PHY モジュールは OOB(Out-of-Band)制御ロジック(oob\_control.v)により制御される Virtex-6LXT デバイスのビルト・イン高速シリアル・ロジック (GTX トランシーバ) を使って構成されます。SATA-PHY と IP コアの入出力は 20 ビット幅 (16 ビット幅データと 4 ビット幅制御信号) のインターフェイスとなります。本デザインにおいては FPGA 内部のクロック・リソース節約のため自動速度ネゴシエーションは実装されず SATA-II 速度 (3.0Gbps) でのみ接続可能です。ユーザにて新規に基板を設計する場合は、UG366 (Virtex-6 FPGA GTX Transceivers User Guide) に記載のデザイン・ガイドを遵守する必要があります。PHY レイヤのソースコードは IP コア製品版にのみ添付され、購入前の評価版には添付されません。

#### ● SATA IP

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。リンク・レイヤのデザインは CRC、デスクランブル、スクランブルモジュールを含みます。FIS インターフェイスはユーザ回路と簡単に接続できるように設計されユーザ回路側のクロックで動作します。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-IP004 データシート ([http://www.dgway.com/products/IP/SATA-IP/dg\\_sata\\_ip\\_data\\_sheet\\_vt6\\_jp.pdf](http://www.dgway.com/products/IP/SATA-IP/dg_sata_ip_data_sheet_vt6_jp.pdf)) を参照してください。

### ● NPI-SATA Ctrl

このモジュールは SATA-IP と上位レイヤ間のデータおよび制御/ステータスを接続するブリッジ回路です。ソースコードは“pcores/np\_i\_sata\_v1\_00\_a/hdl/vhdl/np\_i\_sata.vhd”に格納されています。図 5 に示されるように、NPI-SATA Ctrl は NPI インターフェイスを介して FIS データを MPMC と送受信し、BRAM インターフェイスを介してコアへの制御信号やコアからのステータス信号を MicroBlaze と送受信します。

接続相手の SATA デバイスに対してデータを書き込む場合、MicroBlaze は先に FIS データを DDR3 内に用意し、次に転送を開始するスタート信号を本モジュールに対して出力します。すると本モジュールは予め設定された数量分のデータ・リードを NPI に対して要求します。Data FIS Header Inserter(データ FIS ヘッダ挿入回路)は NPI\_RdFIFO\_Empty 信号を監視し、全てのデータが転送されるまで MPMC 側からリードしたデータを SATA-IP コアへ送信しますが、このとき FIS タイプが DATA FIS であった場合は FIS ヘッダを自動的に挿入します。

SATA デバイスからの読み出しにおいては、MicroBlaze が SATA-IP コアに対してのリード転送開始を指示すると FIS データがユーザ・ロジックに対して出力されます。このときデータ FIS のヘッダやエラー・コードは FIS データから取り除かれ、純粋なリード・データのみが FIFO16x32 に書き込まれ、さらにシステムのデータ・バッファに転送されます。本リファレンス・デザインにおいてはデザインの簡略化のため、NPI のライト要求時でのバースト・サイズは 32 ワード固定です。従って FIS パケットの実データ・サイズが 32 ワードの倍数でない場合、FIS データのリード完了後にダミー・データが 32 ワードの倍数となるよう、自動的に補填されます。NPI Command Control は 32 ワード・データが MPMC に対して送信されるごとに NPI 転送要求を出力します。リードプロセスは、SATA-IP からこれ以上転送するデータが残っていない状態となると自動的に停止します。

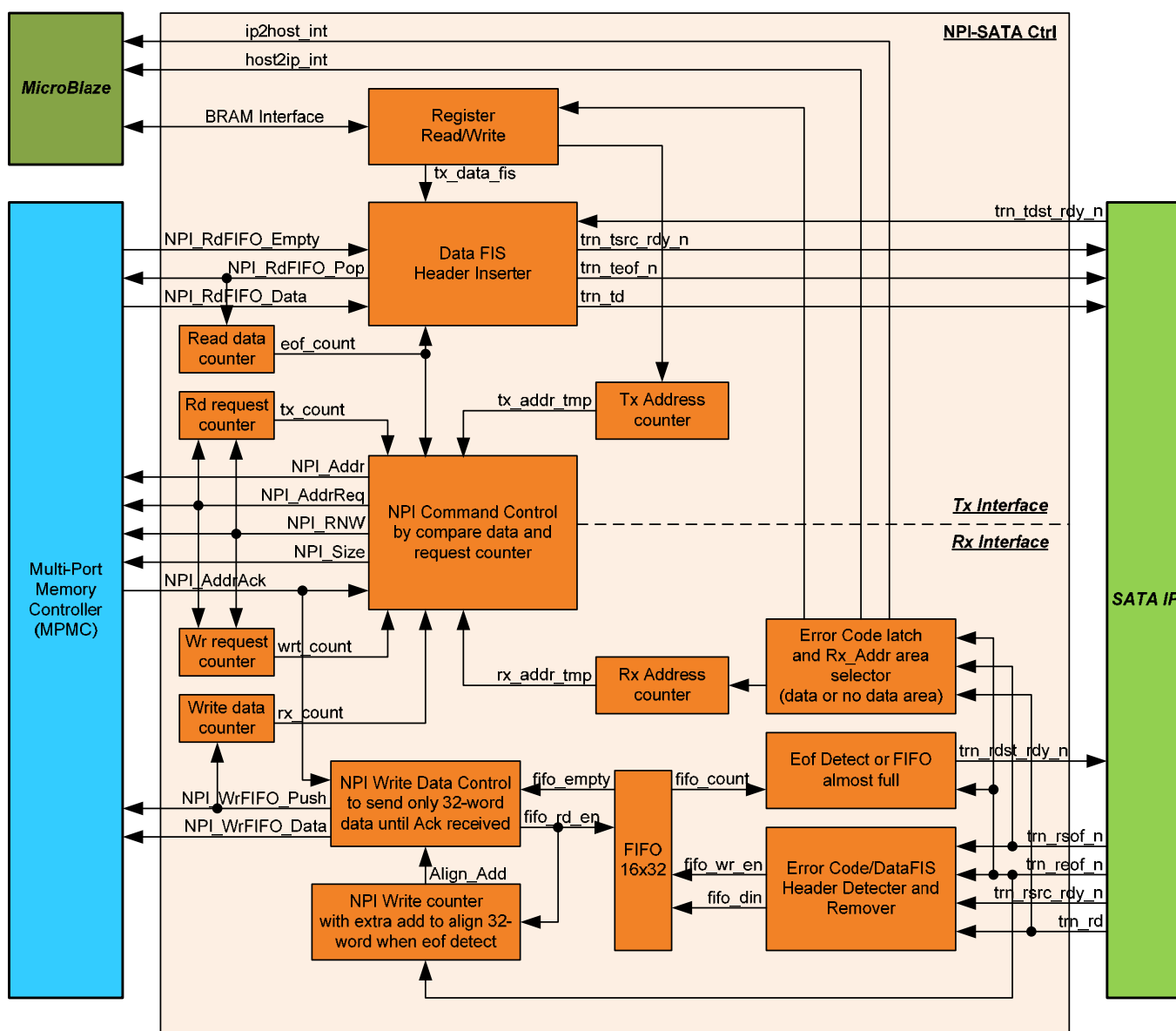


図 5: NPI-SATA Ctrl ブロック図

### ● MicroBlaze

MicroBlaze 上のソフトウェアは、レジスタ・インターフェイスによって制御の実行とステータスの読み出しを行います。制御レジスタの書き込みによって、MPMC の DMA 機能を使ったメイン・メモリ上の FIS データから SATA-IP リンク・レイヤへの転送を実行したり、あるいは逆にリンク・レイヤから受け取った FIS データのメイン・メモリへの転送を実行します。

FIS データ受信時には、NPI-SATA Ctrl 内の FIS ヘッダ・でコードによって検出された FIS タイプによって、受信 FIS データはメインメモリ内の異なるアドレス先に転送されます。FIS データ送信時は全ての FIS タイプで同じアドレス先を使いますが、Control レジスタの FIS タイプ・フラグ(ビット 29)が 1 にセットされた場合、純粋な送信データに対してデータ FIS ヘッダが自動的に付加されます。MicroBlaze からみた制御/ステータス・レジスタの詳細については下表 1 を参照してください。また、MicroBlaze による具体的な制御シーケンスについてはファームウェアのソースコードを参照してください。

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+0x00 Rd	ステータス・レジスタ (STATUS)	[8]MPMC 準備完了 [7:4]OOB 状態データ [3]GTP PLL がロック [1]'1' 固定 [0]SATA IP コアがリンク
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)	受信したワード数、次にコントロール・レジスタへの書き込みが発生するまで全ての FIS データの受信数が積算される。
BA+0x00 Wr	送信データ格納アドレス (TX_ADDR)	送信する DDR3 データを格納している先頭のアドレスを指定する。
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 のアドレス。本レジスタの Bit[7:0]は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]SATA 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)	データ FIS を受信したときに格納される DDR3 の先頭アドレス。
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

(BA : ベース・アドレス)

表 1 MicroBlaze 側からのレジスタ・マップ

注: BA (ベース・アドレス)はそれぞれの SATA デバイス番号ごとに以下のように異なります。

- BA = 0x10000000 デバイス番号#0
- BA = 0x20000000 デバイス番号#1
- BA = 0x30000000 デバイス番号#2
- BA = 0x40000000 デバイス番号#3

## 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 の数は異なってきますが、おおむね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 2 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 にセットするようですのでここは 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 R 1 0 0 0	PM Port 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 2: IDENTIFY COMMAND の FIS 構造

- READ DMA EXT

表3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の手順が違うだけで、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の方が手順が簡単なので、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	C R R R PM Port 1 0 0 0 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3: DMA READ EXTのFIS構造

- WRITE WRITE EXT

表4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。これを繰り返しデータをすべて送信し終わったら、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	C R R R PM Port 1 0 0 0 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: 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 6 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。

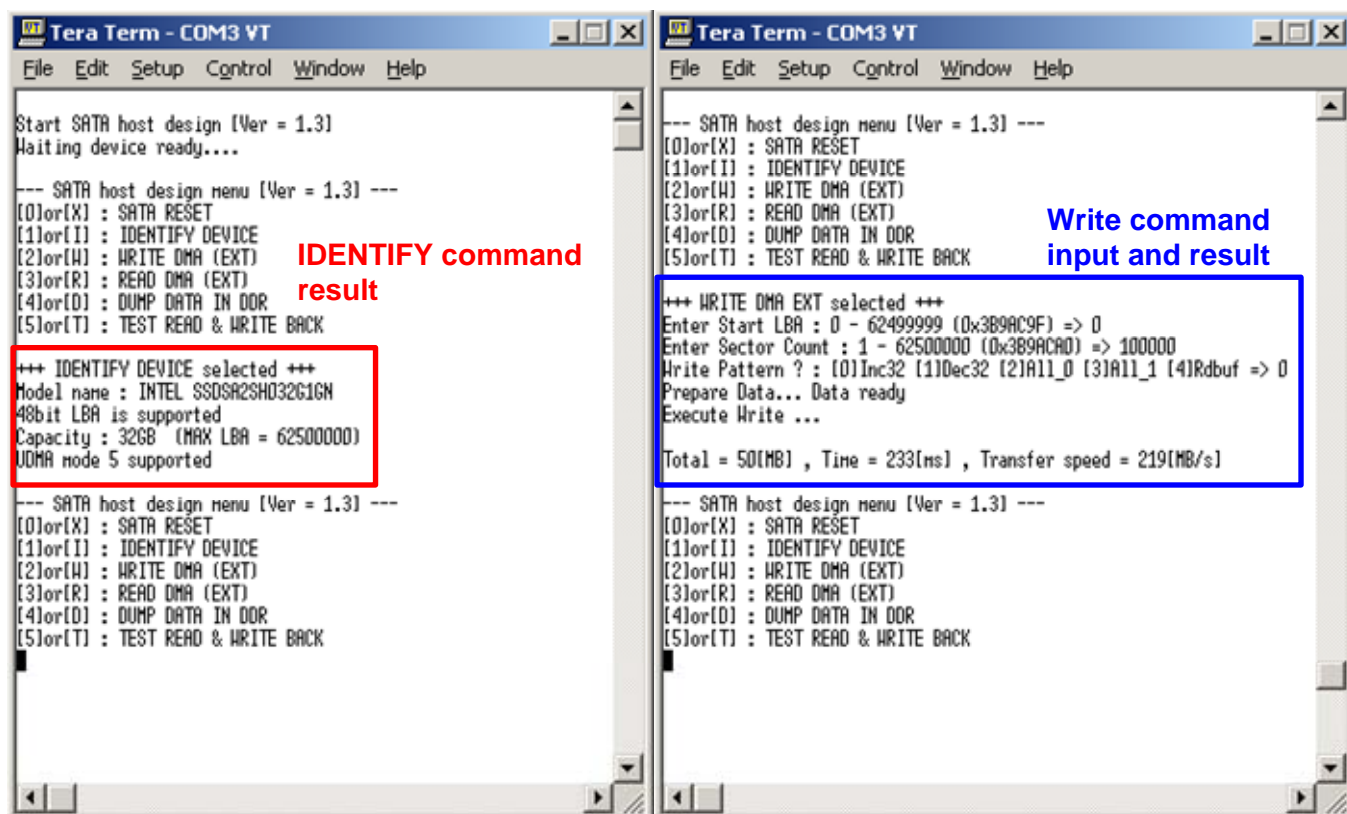


図 6 動作実行時のシリアル・ターミナル画面サンプル

## 5. 改版履歴

Revision	Date	Description
1.0	2010/5/10	日本語第 1 版作成
1.1	2010/05/26	図 2 の修正

Copyright: 2010 Design Gateway Co,Ltd.