

SATA AHCI-IP コア・リファレンス・デザイン説明書

Rev1.3J 2017/03/22

本ドキュメントは AHCI-IP コア実機デモ・システムのリファレンス・デザインを説明したものです。SATA-IP コアの上位に AHCI-IP コアを実装することで、アプリケーション・レイヤのドライバを介して Linux 等の OS から接続 SATA デバイスを直接ドライブとして認識でき、ファイル・システムを使ったリード・ライトが可能となります。

1. 概要

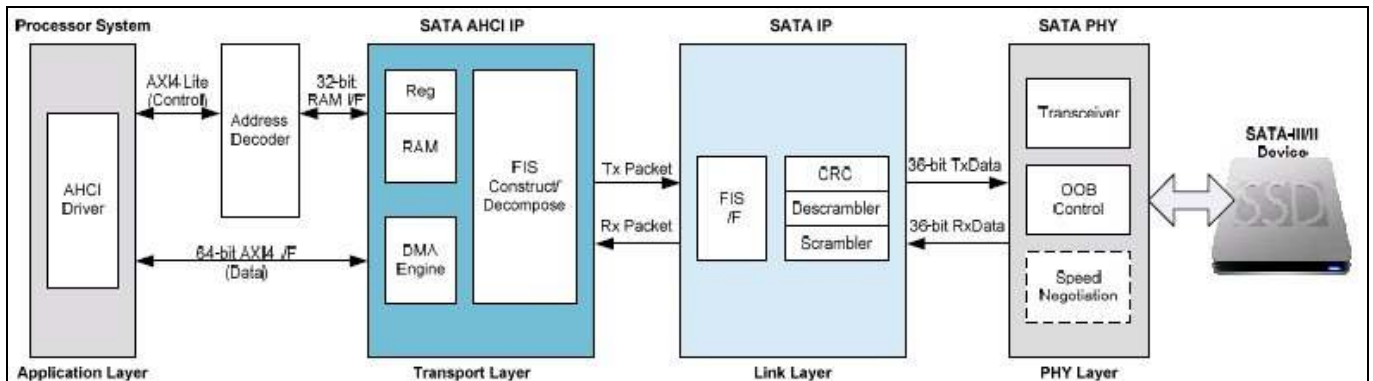


図 1-1: SATA AHCI-IP デモの全体ブロック図

SATA 応用システムにおいては、実装される SATA プロトコルは4レイヤに分けて実装されます。SATA AHCI-IP コアはこのうちトランスポート・レイヤを実装し、図 1-1 に示すように上位アプリケーション・レイヤのプロセッサ・システムや下位リンク・レイヤの SATA-IP コアと接続します。トランスポート・レイヤは送信における FIS と呼ばれるフレーム・インフォメーション・ストラクチャを構築し、受信した FIS を解析する役割を担います。FIS の処理シーケンスは ATA コマンドごとによって異なり、プロセッサのソフトウェアで制御されます。SATA-IP コアや PHY(物理)レイヤの詳細に関しては、SATA-IP コアの技術資料を参照してください。

プロセッサは Linux OS を動作し AHCI ドライバを介して SATA AHCI-IP コアにアクセスします。IP コア内の DMA エンジンとメイン・メモリ間の DMA データ転送には 64 ビット AXI4 バスが使われますが、IP コア内部レジスタは RAM インターフェイスを通して制御されます。AXI4-Lite バスから RAM インターフェイスへの変換にアドレス・デコーダが使われます。本デザインのより詳細については本ドキュメントの次章以下を参照してください。

2. ハードウェア

下図 2-1 に SATA AHCI-IP コアの詳細ブロック図を示します。

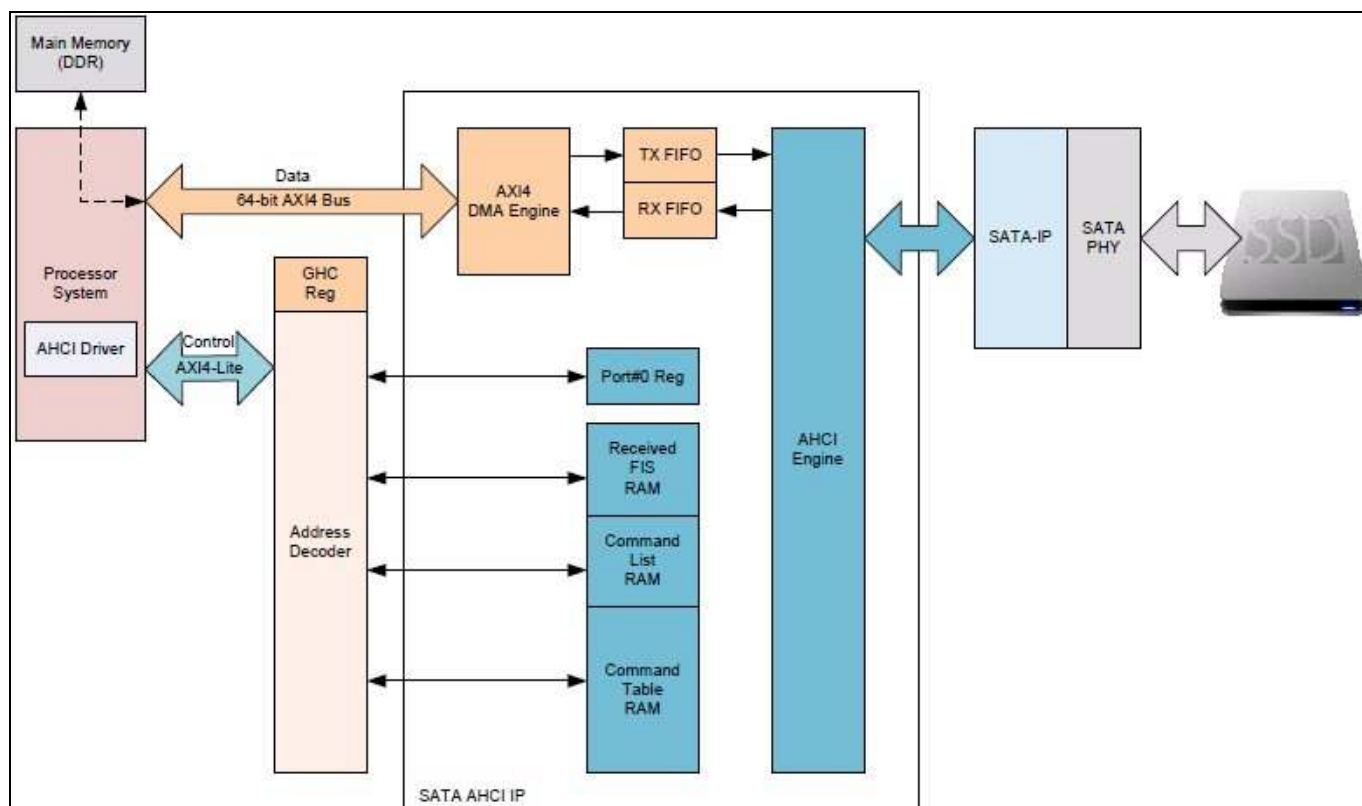


図 2-1: SATA AHCI-IP コアの詳細ブロック図

ホスト動作の SATA-IP コアに対して転送する FIS パケットには、コマンド FIS とデータ FIS の 2 種類があり、プロセッサによってコマンド・テーブル RAM 内に用意されます。プロセッサからスタート信号が発行されると AHCI エンジン は図 2-2 に示すようにそれぞれのコマンド・スロットにおけるコマンド・テーブル RAM の先頭エリアから 5 ダブル・ワード (=20 バイト) のデータをリードし、SATA-IP コアへ FIS パケットとして転送します。コマンド・キュー機能に沿うため、AHCI エンジン は最大 32 コマンドを同時に SATA-IP コアへ発行でき、SATA デバイスからの応答を待って DMA Setup FIS により対象コマンドが選択されます。

データ FIS においては AXI4 DMA エンジン は PRD に設定された値に従ってメイン・メモリからデータを読み出し TX(送信)FIFO へ転送します。PRD はバースト転送の転送サイズとメイン・メモリの先頭アドレスを定義するために使われます。送信 FIFO 内のデータは SATA デバイスからの DMA Activate FIS が戻された後に AHCI エンジン によって SATA-IP コアへとバースト転送されます。AHCI エンジン は SATA-IP コアへデータ・パケットを送信する前に FIS ヘッダを挿入します。スキップ・ギャザー機能をサポートするために 1 コマンド当たり 120PRD スロットが用意されています。

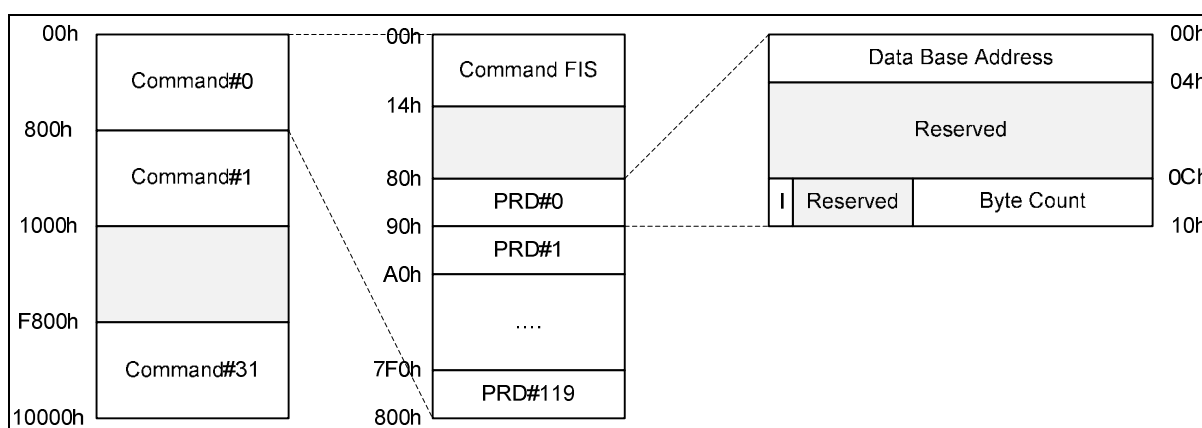


図 2-2: コマンド・テーブル RAM の構造

受信 FIS パケットは非データ FIS かデータ FIS かの2種類に振り分けられます。AHCI エンジン は SATA-IP コア から出力される受信 FIS タイプをチェックしデータ FIS の場合は受信 FIFO へ格納します。このときパケットを受信 FIFO へ転送する前に FIS ヘッダは除去されデータのみが転送されます。送信側と同様 AXI4 DMA エンジン は PRD にセットされた値をもとに受信 FIFO からメイン・メモリへと転送されます。一方非データ FIS の場合、受信 FIS は FIS タイプ番号ごとに異なった受信 FIS RAM 空間へ格納されます。

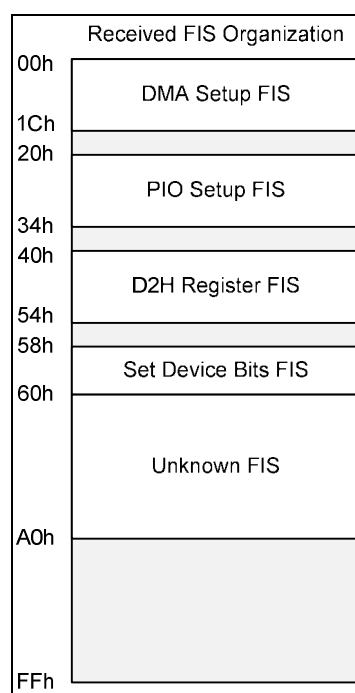


図 2-3: 受信 FIS RAM のアドレス・マップ

AHCI-IP コア内の AHCI レジスタは SATA AHCI 3.1 規格に準拠しており以下からダウンロード可能です。

[AHCI3.1 規格書ダウンロード先]

<http://www.intel.com/content/www/us/en/io/serial-ata/serial-ata-ahci-spec-rev1-3-1.html>

AHCI 規格において受信 FIS、コマンド・リスト、コマンド・テーブルはメイン・メモリにマッピングされます、しかし SATA AHCI-IP コアはパフォーマンス改善のためこれらは BRAM にマッピングされます。メモリ・マップ・アドレスの詳細については AHCI-IP コアデータ・シートを参照してください。

3. ソフトウェア

本章では5種類の ATA コマンドをサポートするファームウェアの基本的な動作について説明します。説明するコマンドは IDENTIFY DEVICE (ECh), WRITE DMA EXT (35h), READ DMA EXT (25h), WRITE FPDMA QUEUED (61h), READ FPDMA QUEUED (60h)です。キュー・コマンドをサポートしない SATA デバイスではホストとデバイス間のデータ転送には WRITE/READ DMA EXT コマンドを使う必要があります。キュー・コマンドをサポートする SATA デバイスでは WRITE/READ FPDMA QUEUED コマンドを使ってパフォーマンスを上げることが推奨されます。

WRITE/READ DMA コマンドにおいてホストはデバイスに対して同時には1コマンドしか送信することができないためコマンド・スロットは1つだけ使います。各コマンドで 120 箇所の異なる空間にデータを配置できるようにするため 120PRD までサポートします。

WRITE/READ QUEUED コマンドではホストはデバイスに対してデータの実処理を待たずに最大 32 コマンドまで送信することができます。そしてコマンド・スロットを参照するために 0~31 の値を持つタグ番号が使われます。タグ番号はホストがデバイスに対して新しいコマンドを発行する際にコマンド FIS 内にセットされ、デバイスもまた DMA Setup FIS においてホスト側に対象となるコマンド・スロットを指定するためにセットします。

ホスト側においてはコマンド・スロットに空きがあれば新しいコマンドを送信できますが、特にデバイスが現在処理中のコマンド動作の最後に Set Device Bits FIS を戻してきた直後で可能となります。デバイス側においては現在のコマンド動作が完了しホストからのコマンドがまだキュー内に残っている場合、新しいコマンドでのデータ転送のため DMA Setup FIS をホストに送信することができます。コマンドとデータはそれぞれ独立して動作するため、コマンド・キューのパフォーマンスは DMA コマンドにより向上します。

各コマンドにおける GHC とポート#0 レジスタの設定値については以下により詳細を説明します。

3.1 初期化

システムの電源投入後、グローバル HBA コントロール(GHC)レジスタと Port#0 レジスタを初期化する必要があります。以下に AHCI アプリケーションを起動するための手順について説明します。

- 1) GHC_HR='1'をセットしハードウェア・システムをリセットします。
- 2) PxCMD.FRE='1'をセットし FIS 受信を許可します。
- 3) PxCMD.ST='1'をセットしコマンド・リストの処理を許可します。
- 4) 受信 FIS による以下の割り込みを許可します。
 - PxIE.DHRE='1': D2H レジスタ FIS 割り込み許可
 - PxIE.PSE='1': PIO Setup FIS 割り込み許可
 - PxIE.DSE='1': DMA Setup FIS 割り込み許可
 - PxIE.SDBE='1': Set Device Bits FIS 割り込み許可
 - PxIE.UFE='1': Unknown FIS 割り込み許可
 - PxIE.DPE='1': PRD に'1'をセットする動作完了の割り込み許可

システムの初期化の最後にて SATA デバイスは D2H レジスタ FIS(シグネチャ FIS)をホストに転送します。

- 5) PxIS.DHRS フラグのアサートおよびステータス・レジスタのエラー・ビットがデフォルト値でセットされていると PxIS.TFES もアサートされます。
- 6) ホストは PxIS.DHRS='1'をセットすることで割り込みをクリアし、さらに PxIS.TFES='1'をセットすることでタスク・ファイル・エラーによる割り込みを防止します。
- 7) PxIE レジスタに 0xFFFF_FFFF. をセットすることで全ての割り込みを許可します。

3.2 IDENTIFY DEVICE コマンド

ホストは接続ドライブの情報を取得するため IDENTIFY DEVICE コマンドを発行します。このコマンドのシーケンス例を以下に説明します。

- この例ではコマンド・スロット#0 が使われるので最初のステップではコマンド・ヘッダ 0 エリア内の PRDTL に'1'をセットします。SATA デバイスから返送される Identify Device データは 512 バイトなのでひとつの PRD のみで十分です。

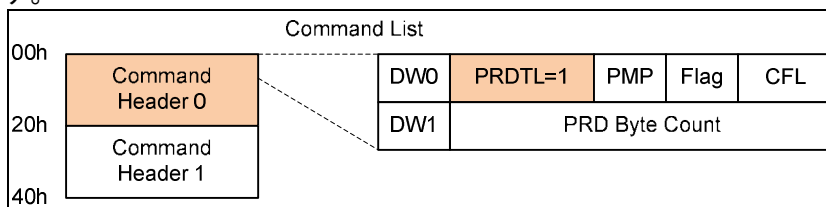


図 3-1: Identify Device コマンドのコマンド・リスト設定

- コマンド・テーブル RAM のスロット#0 に 5 DWord のコマンド FIS をセットします。

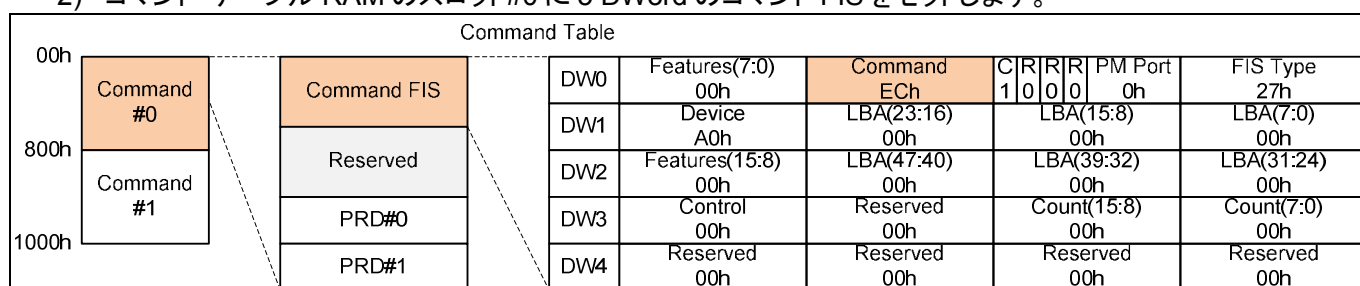


図 3-2: Identify Device コマンドのコマンド FIS 設定

- コマンド・テーブル RAM の PRD#0 において、512 バイトの IdentifyDevice データを格納するためメイン・メモリの空き領域番地をデータ・ベース・アドレスに、'1'ビットに 1 を、バイト・カウント=0x1FF をセットします。

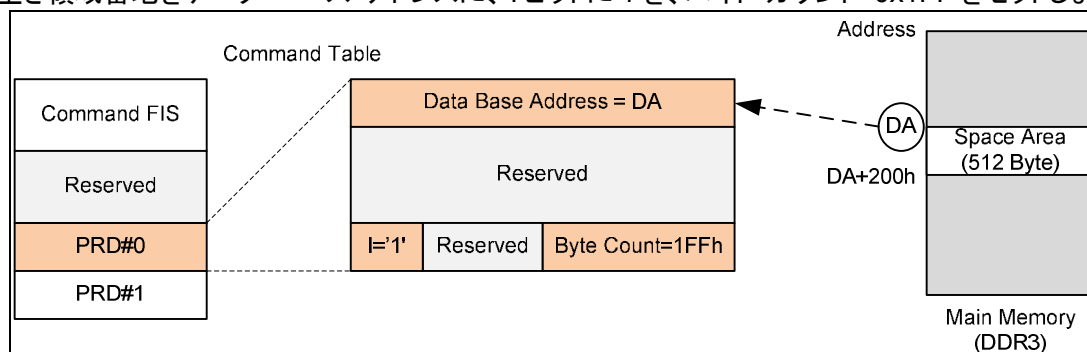


図 3-3: Identify Device コマンドの PRD 設定

- スロット#0 のコマンド FIS を送信するため PxCI のビット 0 = '1' をセットします。

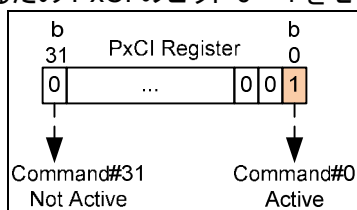


図 3-4: Identify Device コマンドの PxCI レジスタ設定

- 5) PIO セットアップ FIS の割り込み(PxIS.PSS='1')を待ちます、このとき Identify device データの PxIS DPS が'1'となります。

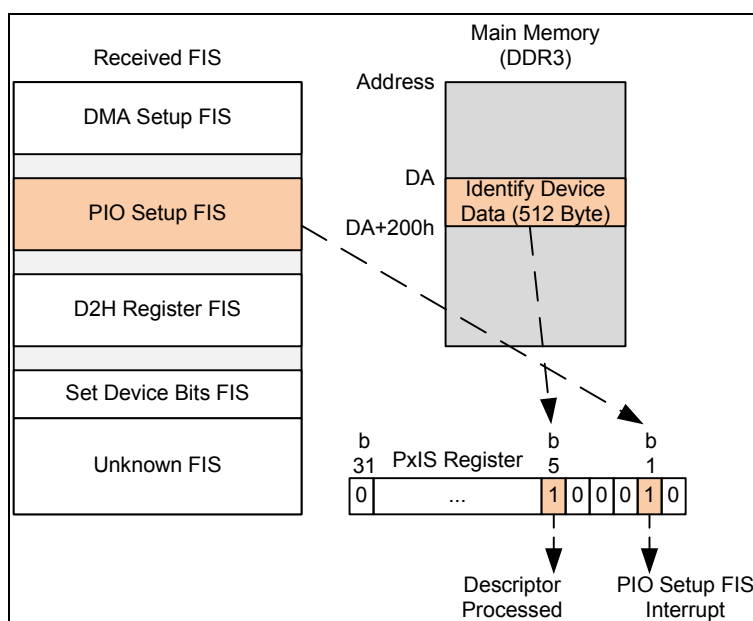


図 3-5: Identify Device コマンドで返信された FIS

- 6) PxIS.PSS='1'をセットし、割り込みをクリアするため PxIS.DPS='1'をセットします。
- 7) コマンド処理が終わったことを PxCI のビット 0='0'で確認します。
- 8) メイン・メモリ内のデータ・ベース・アドレス空間に Identify データが格納され読み出せる状態になります。

3.3 WRITE DMA EXT コマンド

このコマンドのシーケンス例を以下に説明します。

- 1) メイン・メモリ内に記録データを準備します。このとき記録データは図 3-6 に示すように n+1 セグメントに分割されメモリ内に格納されているものとします。

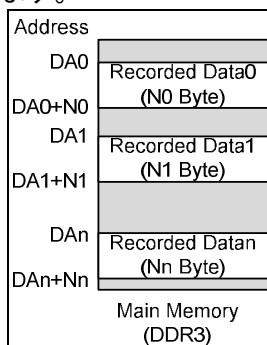


図 3-6: メモリ内の Write DMA Ext コマンドのライト・データ格納状態例

- 2) この例ではコマンド・スロット#0 を使います。コマンド・ヘッダ 0 エリアの PRDTL に全 PRD 数 (n+1) をセットします。

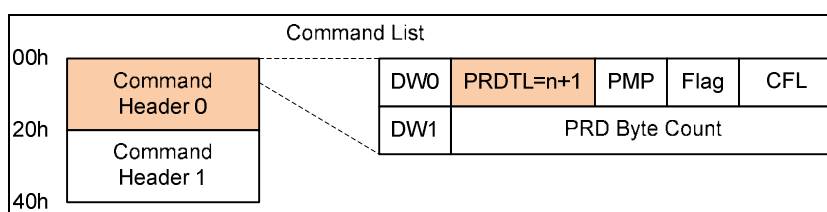


図 3-7: Write DMA Ext コマンドのコマンド・リスト設定

- 3) コマンド・テーブル RAM のスロット#0 に 5 DWord のコマンド FIS をセットします。

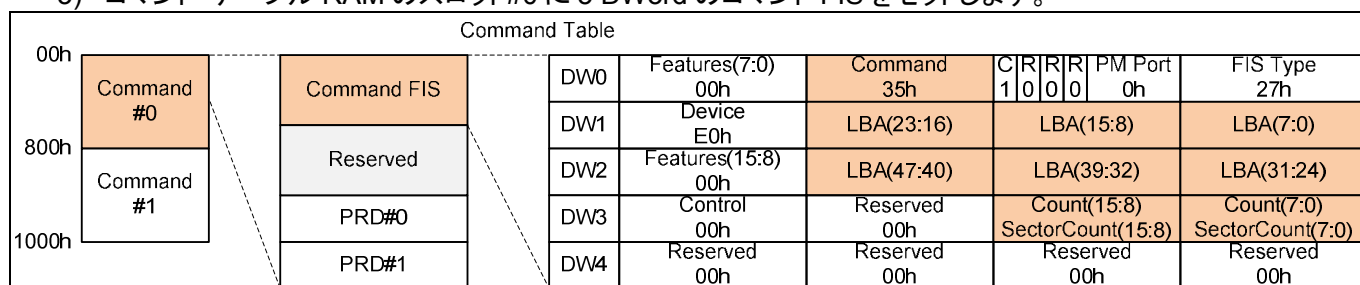


図 3-8: Write DMA Ext コマンドのコマンド FIS 設定

- 4) 図 3-9 に示すように、各 PRD のデータ・ベース・アドレスにメイン・メモリ内に格納されているそれぞれの記録データの先頭番地をセットし、またバイト・カウントに各セグメントでのデータ・サイズをセットします。また、最後の PRD では '1' ビットに '1' をセットします。

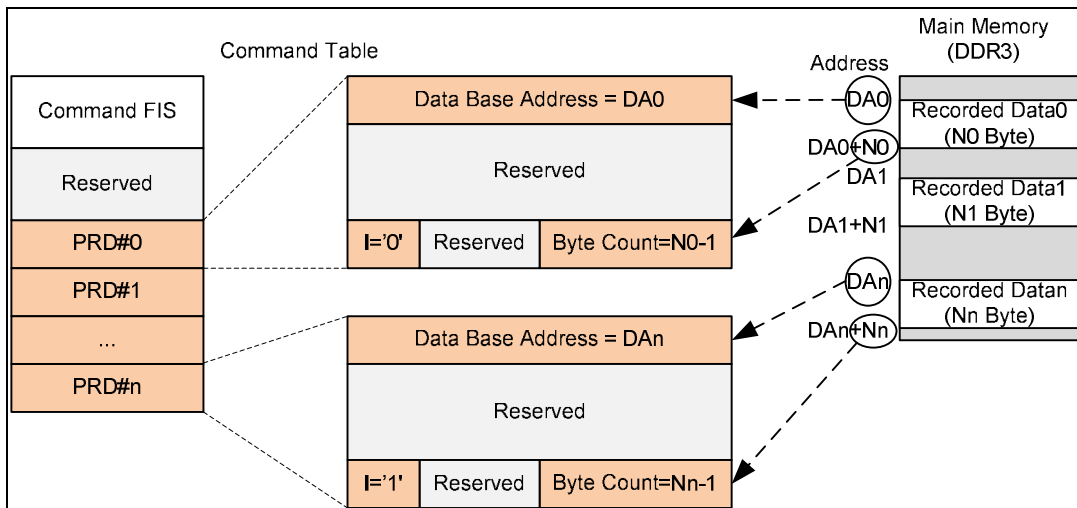


図 3-9: Write DMA Ext コマンドの PRD 設定

- 5) PxCI.b0='1'をセットし、スロット#0 のコマンド FIS を送信します。
- 6) D2H レジスタ FIS 受信の割り込み (PxIS.DHRS='1') を待ちます、このとき最終 PRD のデータが転送されて PxIS.DPS が'1'となります。

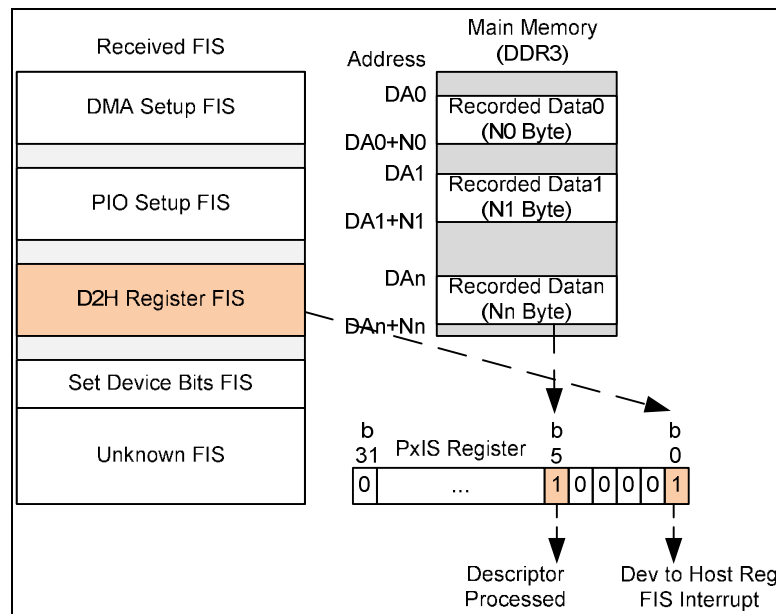


図 3-10: Write DMA Ext コマンドで返信された FIS

- 7) PxIS.DHRS='1'をセットし、割り込みをクリアするため PxIS.DPS='1'をセットします。
- 8) コマンド処理が終わったことを PxCI のビット 0='0'で確認します。

3.4 READ DMA EXT

このコマンドのシーケンス例を以下に説明します。

- 1) SATA デバイスから読み出すデータを格納するメイン・メモリの空き領域を準備します。

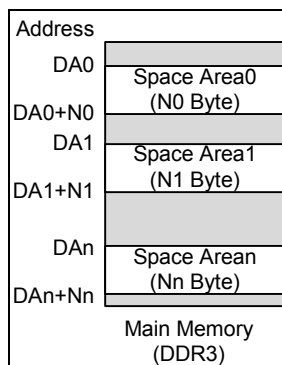


図 3-11: Read DMA Ext コマンドのリード・データ格納先例

- 2) Write DMA Ext コマンドのステップ 2)から 8)と同じ手順を進め、コマンドを送信します。Read DMA Ext コマンドでの 5DWord のコマンド FIS を図 3-12 に示します。

Command Table		DW0	Features(7:0) 00h	Command 25h	CIR	RR	RR	PM Port 0h	FIS Type 27h
00h	Command #0	DW1	Device E0h	LBA(23:16)	1	0	0	0	LBA(7:0)
800h	Command #1	DW2	Features(15:8) 00h	LBA(47:40)					LBA(31:24)
		DW3	Control 00h	Reserved 00h				Count(15:8) SectorCount(15:8)	Count(7:0) SectorCount(7:0)
1000h		DW4	Reserved 00h	Reserved 00h				Reserved 00h	Reserved 00h

図 3-12: Read DMA Ext コマンドのコマンド FIS 設定

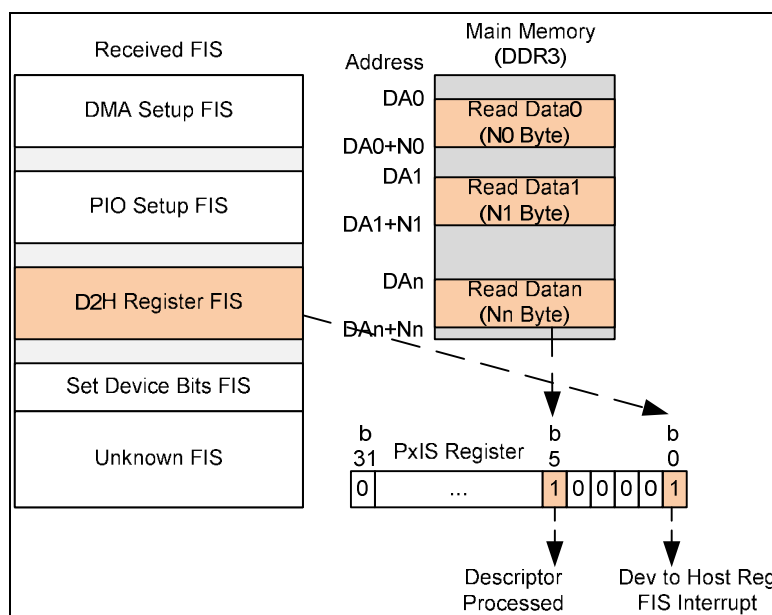


図 3-13: Read DMA Ext コマンドで返信された FIS

- 3) メイン・メモリのリード・データを読み出すことができますようになります。

3.5 WRITE FPDMA QUEUED コマンド

基本的なシーケンスを説明するため、以下ではまず1スロット・コマンドを例として説明します。

- 1) Read PxSACTレジスタをリードし使用可能なスロットをチェックします。ここではスロット#0が利用可能とします。そして(空いていた)スロット#0のビット0を'1'にセットしこのスロットを確保します。

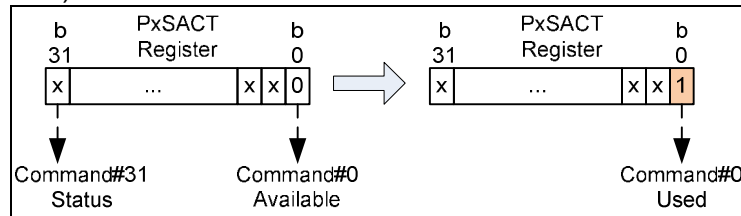


図 3-14: PxSACT レジスタでコマンド・スロット#0が使用可能

- 2) Write DMA Extコマンドのステップ 1)から 5)と同じ手順を進め、コマンドやデータを準備してコマンドを送信します。Write FPDMA Queuedコマンドでの 5DWord のコマンド FIS を図 3-15 に示します。

Command Table		DW0	Features(7:0) SectorCount(7:0)	Command 61h	C R R R PM Port 1 0 0 0 0h	FIS Type 27h
00h	Command #0	DW1	Device 40h	LBA(23:16)	LBA(15:8)	LBA(7:0)
800h	Command #1	DW2	Features(15:8) SectorCount(15:8)	LBA(47:40)	LBA(39:32)	LBA(31:24)
	PRD#0	DW3	Control 00h	Reserved 00h	Count(15:8) Reserved	Count(7:3) Tag(4:0) RSV
1000h	PRD#1	DW4	Reserved 00h	Reserved 00h	Reserved 00h	Reserved 00h

図 3-15: Write FPDMA Queued コマンドのコマンド FIS 設定

- 3) D2H レジスタ FIS 受信(PxIS.DHRS='1')による割り込みが発生し、SATA デバイスによってコマンドが認識されます。そして PxIS DHRS='1'をセットして割り込みをクリアします。
- 4) コマンド処理が終わったことを PxCI のビット 0='0'で確認します。
- 5) DMA Setup FIS による割り込み(PxIS.DSS='1')が発生し上位ホストに対してアクティブなコマンド・スロットを通知します。この例ではアクティブ・スロットはスロット#0です。そして IP コア内の DMA エンジン PRD の内容に従ってメイン・メモリから SATA デバイスへのデータ転送を開始します。そして割り込みをクリアするため PxIS DPS='1'をセットします。
- 6) デスクリプタ処理完了による割り込み(PxIS.DPS='1')が、最終 PRD の全データが SATA デバイスへ転送された後に発生します。
- 7) Set Device Bits FIS による割り込み(PxIS.SDBS='1')が発生しライト動作が終了します。そして割り込みをクリアするため PxIS.SDBS='1'をセットします。
- 8) PxSACT の bit0 が'0'になりスロットのステータスが使用可能状態に復帰します。

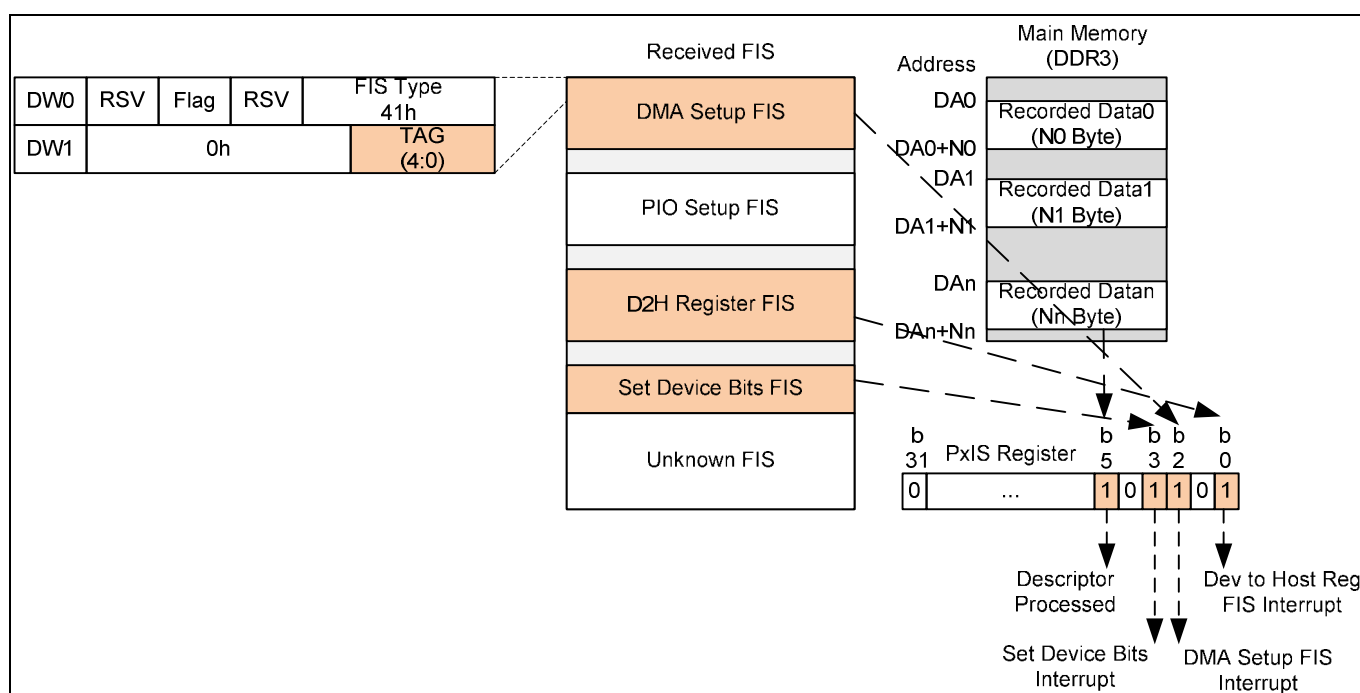


図 3-16: Write FPDMA Queued コマンドで返信された FIS

32 個のキュー・コマンドを並列に走らせるため、全8ステップが2フェーズすなわちステップ 1)~4)のコマンド・フェーズとステップ 5)~8)のデータ・フェーズに分割され同時に実行します。コマンド・フェーズでは PxCI レジスタから使用可能な(空き)スロットをスキャンし空きスロットがあればそこに新しいコマンドで埋めます。新しいコマンドは全 32 スロットが全て埋まるまで発行可能です。一方データ・フェーズでは割り込みを管理しハードウェアが正常動作していることを監視します。

3.6 READ FPDMA QUEUED コマンド

このコマンドのシーケンスは Write FPDMA Queued コマンドとほとんど同じです。異なる点として、Read DMA Ext コマンドと同様 SATA デバイスからのリード・データ用にメイン・メモリ空間がアロケートされていることです。Read FPDMA Queued コマンドでの 5DWord のコマンド FIS を図 3-17 に示します。

Command Table	
00h	Command #0
800h	Command #1
1000h	

DW	Field	Value	Field	Value	Field	Value
DW0	Features(7:0) SectorCount(7:0)	60h	C R R R	1 0 0 0	PM Port	0h
DW1	Device	40h	LBA(23:16)	LBA(15:8)	LBA(7:0)	
DW2	Features(15:8) SectorCount(15:8)		LBA(47:40)	LBA(39:32)	LBA(31:24)	
DW3	Control	00h	Reserved	Count(15:8) Reserved	Count(7:3) Tag(4:0)	RSV
DW4	Reserved	00h	Reserved	Reserved	Reserved	00h

図 3-17: Read FPDMA Queued コマンドのコマンド FIS 設定

4. 改版履歴

リビジョン	日付	履歴
1.0	7-Nov-14	Initial Release
1.1	17-Jul-15	Update IP block diagram
1.2	8-Jan-16	Add SATA-II device support in Figure 1-1
1.3	1-Mar-16	Update Figure1-1 to support fixed speed mode
1.3J	2017/03/22	Rev1.3 英語版に対応した日本語版リリース

Copyright: 2014 Design Gateway Co.,Ltd.