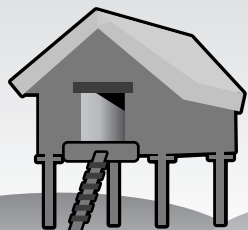


FPGAでハード・ディスクを制御してみた



大牧正知



筆者はFPGAボードを使ってハード・ディスクを制御する装置を自作した。設計したのは主にFIS (Frame Information Structure) と呼ばれるデータをメイン・メモリに作成し、リンク層に相当するブロックに転送するようなFPGAデザインである。物理層はFPGAの高速トランシーバで、リンク層はSATAのIP (Intellectual Property) コアで、トランスポート層はVHDL記述などで実現し、全体の制御にはCPUコアであるMicroBlazeを使用した。ここでは本製作事例を使って、

SATAホストの動作や仕組みを解説する。本稿で掲載するVHDLやC言語のソース・ファイルは本誌Webサイト (<http://www.cqpub.co.jp/dwm/>) からダウンロードできる。(編集部)



1. SATAホストのハードとソフト

● 組み込み機器でSATAを実現するためのハードウェア
組み込み機器でSATAを実現する場合、大きく分けて以下に示すようなハードウェア構成が考えられます(図1)。

(1) SATA内蔵のCPU(SOC)を使う場合

米国AMCC (Applied Micro Circuits Corporation) の「PPC460EX」や米国Freescale Semiconductor社の「MPC8379E」などのCPUは、1チップでSATAインターフェースを備えています。これらを使えばそのままSATAが使えるので簡単です。

これらのCPUの欠点としては、比較的値段が高いことと、5チャンネル以上のSATAポートを持っているデバイスが少ないことがあります。また、多くのCPUはハード・ディスクを制御する側であるSATAホスト機能が内蔵されており、制御される側のSATAデバイス機能はほとんど内蔵されていません。

(2) 外付けSATAコントローラICを使う場合

米国SiliconImage社や米国Marvell Technology社などから、PCIやPCI-X, PCI Express接続のSATA制御(コントローラ)ICが提供されています。ホスト・プロセッサがSATA制御ICで使われているバス・インターフェース

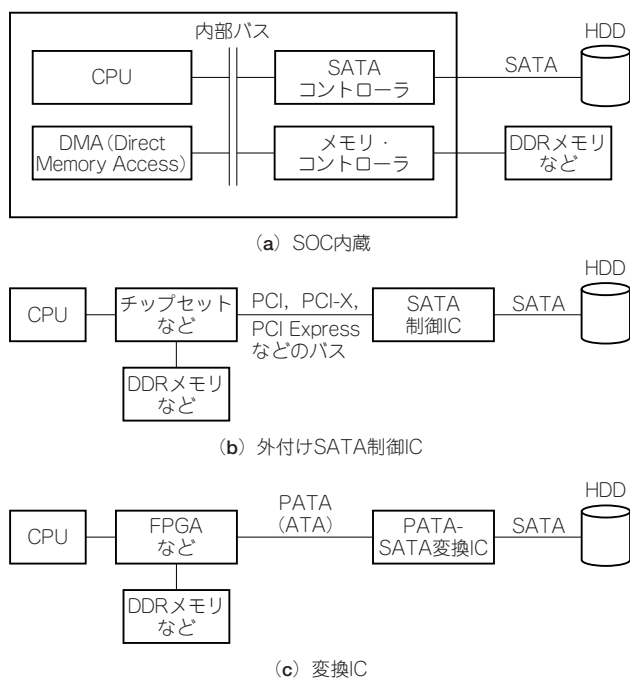


図1 組み込み機器におけるSATA実現例

Keyword

SATAコントローラ, トランスポート層, リンク層, 物理層, ホスト, デバイス, FIS, DMA転送, MicroBlaze, VHDL, C言語, IDENTIFY DEVICE, DMA READ EXT, DMA WRITE EXT



を持っていれば、そのままつなぐだけで使用できます。

欠点としては、システム・コストが高くなってしまいます。PCIではSATAに対して十分な帯域がないので、PCI-XやPCI Expressのバス・インターフェースを持った製品が多いのですが、これらのインターフェースを持ったCPUは比較的値段が高くなるからです。また、これもSATAデバイスの機能を持っているものは少ないようです。

(3) PATA-SATA 変換チップを使う場合

従来のPATA(ATAのこと)のインターフェースに、PATA-SATA変換チップを接続して、SATAを実現できます。この方法だと従来のソフトウェアがそのまま使えます。設計資産をそのまま流用できるので簡単です。また、PATA-SATA変換チップはかなり安くなってきており、システム・コストも低く抑えられます。PATA-SATA変換チップは、SATAデバイス側にも使用できます。

欠点は、1.5Gbpsまで(実際はUltra DMAモード5の133Mバイト/sまで)の速度しかサポートできないことと、多チャンネルになると基板上の配線が増えてしまう(1チャンネル当たり30本程度)ことです。

(4) FPGAにSATAのIPコアを実装する場合

そのほか、FPGAにSATAのIPコアを入れて実現するという方法があります。これはSATA内蔵のCPU(SOC)を使う場合と近い形になります。SATAデバイスとしても使用できます。本稿では、この方法でSATAを実現する方法について解説します。

● FISというデータを使ってSATAデバイスにアクセス

SATA内蔵CPUを使う場合や外付けのSATA制御ICを使う場合と、PATA-SATA変換チップを使う場合とでは、ソフトウェアが大きく異なります。

PATA-SATA変換チップを使う場合は、従来のPATAを使うときとソフトウェアはなんら変わりません。SATA内蔵CPUを使う場合や外付けSATA制御ICを使う場合は、通常FIS(Frame Information Structure)と呼ばれるデータ構造を用いてアクセスします。

ハード・ディスクなどのSATAデバイスにアクセスするときは、FISデータを作成し、SATAコントローラに送信します。一般的にはメイン・メモリ上にFISデータを作成し、コントローラがバス・マスタになってそのFISデータを読み取り、SATAデバイスへ送信するようです。

SATAデバイスから受信したFISデータは、SATAコン

トローラがバス・マスタになり、メイン・メモリに展開します。

従って、ホスト・プロセッサは、以下のような手順でSATAデバイスへアクセスを行うことになります。

(1) FISデータの作成

(初めはホストからデバイスにデータを渡すRegH2Dコマンドを作成)

(2) FISデータの送信

(3) FISデータの受信待機

(4) 受信したFISデータを読み出し

(5) 必要に応じてFISの送受信

プロトコルによって送受信するFISの数などが異なりますが、おおむねこのような流れになります。



2. SATAホストをFPGAに実装

● SATA IPコアはリンク層のみの場合が多い

FPGAあるいはASIC(Application Specific Integrated Circuit)でSATAを実現する場合、ゼロから作るのは大変なため、IPコアを使用することが多いかと思います。現在提供されているSATA IPコアは、リンク層部分のみで、物理層やトランスポート層を含んでいない場合が多いようです。

物理層は、デバイス自体の作りこみにより制御方法が異なってくるほか、ホストとデバイスでも動きが異なります。トランスポート層は、上位との接続方法やアプリケーションで作りが異なるため、IPコア化しにくいものと思われます。

● FPGA高速トランシーバと接続できるIPコアを採用

米国Xilinx社のFPGAの場合、Virtex-5 LXT/SXT/FXTなどを用いると、物理層も含めてSATAを実現できます。同社のWebサイトからは、SATAホスト用の物理層のサンプル・デザイン(XAPP870)がダウンロードできるほか、SATAの特性レポート(rpt087)も入手でき、参考になります。

SATA IPコアはいくつかのIPベンダからリリースされています。今回はデザイン・ゲートウェイのSATA IPコアを使用した事例を紹介いたします。一般的なSATA IPコアのPHY(物理層)との接続は、sapisと呼ばれるインターフェースを用いています。本IPコアはVirtex-5 LXT/SXTの高速トランシーバ(GTP)向けに最適化されています。物