

第4章 SATAコネクタがなくてもPCI Express評価ボードがあればつながる

オープン・ソースSATAコアで
高速アクセスを体験！

三好 健文 Takefumi Miyoshi

OpenCoresに公開されているSATAホスト・コントローラIPコアを使って、FPGAによるSATAアクセスを手軽に体験してみます。公開されているコアはリード/ライトに特化したシンプルなコアなので、SATAアクセスの各層でどのような処理をしているか、動かしながら仕組みを学ぶことができます。最終的な目標はオリジナルのIPコア作成！実験に使ったコード一式は本書サポート・ページからダウンロードできるので、環境がある方は試してみてくださいと思います。

1. OpenCoresのSATAコアを
使ってみよう● オープン・ソースをお手本にしたSATAホスト・
コアの開発

SATAは、大容量で高速なハード・ディスク(HDD)にアクセスするためのインターフェースです。手軽にFPGAからSATAが扱えれば一気に扱えるデータの量を増やすことができます。SATAの仕様⁽¹⁾や参考文献^{(2)~(4)}などを見ていると、実装してみたいという気になります。

しかし、SATAを扱うには、高速な物理層に加えスクランブル/デスクランブルやCRCといった複雑な処理が必要になるため、仕様書だけを読みながら実装していても正しい実装なのか不安になることもあるでしょう。SATAプロトコル・アナライザがあればよいのですが、趣味的に開発を始めてみたものの全く動かないときには、何をしていたか分からなくなってしまう。

そんなときに使える手法の一つが、オープン・ソースのIPコアを使って実際に動作する様子から学習する方法です。

● Serial ATA Host Bus Adapter Core for Virtex 6とは

OpenCores.orgでSATAを検索してみるといくつかプロジェクトを見付けることができます。今回は、プロジェクトの更新日が2013年5月と比較的新しく、またSATA Gen2に対応している、Serial ATA Host Bus Adapter Core for Virtex 6⁽⁵⁾を試してみることにします。

名前の通り、このコアはXilinx社のFPGA Virtex-6用のSATAホスト・コアです。Virtex-6用というのは、物理層のトランシーバ部にVirtex-6のGTXを使うことを前提としているコアだということです。他のシリーズのFPGAに移植する場合には、トランシーバに依存する部分のコードを変更する必要があります。

このコアは、FPL2012というFPGAなどの再構成可能デバイスに関する国際会議で、作者らが発表している論文⁽⁶⁾に関連するものだそうです。そしてこのコアは図1のように32ビット幅のFIFOインターフェースに直結したデータ入出力ポートといくつかの制御信号の操作のみで、ユーザ・ロジックからSATAを介してハード・ディスクとデータのやり取りができるように作られています。例えば、FIFOにデータを書き込んだ後で、cmd_typeにWRITE_DMA_CMD("10")を指定し、new_cmdをアサートするとDMAライトでFIFOに書き込んだデータがディスクに転送される、というような使い方が考えられています。SATAにアクセスするコアをMicroBlazeで制御できるようなベース・システムとソフトウェアも一緒に公開されていますが、今回はロジックで簡単なステートマシンを作ってアクセスしてみることにします。

● SATAコアの内部構造

まずは、IPコアを構成するファイルとモジュール構造を見てみましょう。

OpenCores.orgからダウンロードしたアーカイブを展開すると、sata_controller_core/sata2_fifo_v1_00_aの下に関連ファイルを見付けることができます。doc/READMEにIPコアを構成するファイルの説明や注意書きが記載されています。図2はREADMEに示されているファイルの階層関係を書き起こしたものです。sata_core.vhdをトップ・モジュールとして、SATAのプロトコル層が素直に分割された分かりやすい構造になっています。使う場合には、sata_core.vhdに定義されたsata_coreをインスタンス化してラッパ・モジュールを実装すればよさそうです。