

第2章 Vivado HL WebPACK Editionで高位合成にチャレンジ(AXI4-Liteスレーブ編)

AXI4-Liteバスに接続できる乗算IPコアの作り方

小野 雅晃 Masaaki Ono

特設記事第2章と第3章は、本書前号(FPGA マガジン No.14)の特集第6章の続編となります。Xilinx社の高位合成ツールVivado HLSを使って、もっとも簡単な回路構成のIPコアを作成します。今回作成するIPコアは、AXI4バスの仲間であるAXI4-Liteと呼ばれるバスに接続して、ARMコアから読み書きができるレジスタを実装します。これによりCPUから制御できる本格的なIPコアを構成できます。

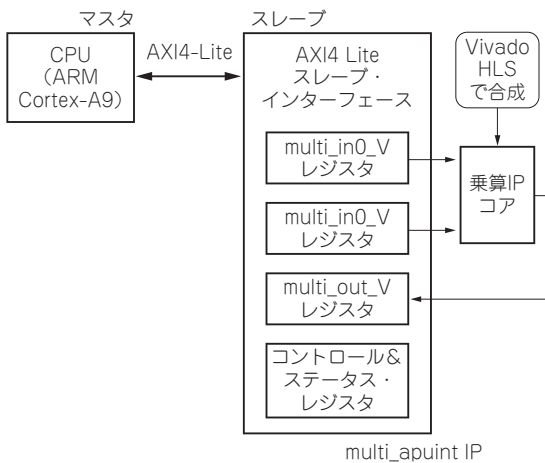


図1 AXI4-Lite版乗算回路のブロック図

● FPGA マガジン No.14の特集のおさらい

筆者は本書前号(FPGA マガジン No.14)の特集第2章から第6章まで、Vivado HLSの使い方について基本的な手順を詳しく解説しました。

特集第2章ではツールのインストールについて、第3章ではC言語でソースを記述し、組み合わせ回路として実装した乗算回路をVivado HLSでIPとしてパッケージ化するまでを解説しました。そして第4章ではVivadoを使って、そのIPにスイッチ入力やLED出力を接続し、実際にZYBOボードで動作を確認しました。

第5章では、乗算回路の入出力にフリップフロップを追加して動作を確認し、次にそれをパイプライン処理できるようにPIPELINE指示子を追加しました。そして第6章では、ディスプレイ・コントローラをC言語で作成し、PIPELINE指示子のrewindオプションの使用方法を確認しました。

● 特設記事第2章と第3章でやること

今回は、今までやってきた乗算回路をAXI4-Liteバス(以下バスは省略)対応させます。そして、その乗算回路をARMプロセッサにつないで、ARMプロセッサから使用します。つまりARMプロセッサからデータを乗算器に与えて、乗算結果を返す乗算アクセラレータのような構成とします。この場合、ARMプロセッサはマスタ(Master)となり、乗算回路はスレーブ(Slave)としてARMのアドレス空間にマップされます。

アクセラレータといっても、AXI4-Liteで接続した乗算器IPは、I/Oアクセスにかかる時間が長く、ARMプロセッサで乗算を実行したほうが圧倒的に早いのですが、AXI4-Liteに接続するIPコアの事例として解説していきます。

● Vivado HLSはドライバも自動生成してくれる

またVivado HLSにはドライバ作成機能があり、とても便利です。AXI4-Liteを使用して回路をIPパッケージ化すると、IPだけではなくドライバ・ソフトウェア(以下ドライバ)もVivado HLSが生成してくれます。そのドライバは、OS無しのベアメタル・アプリケーションだけでなく、Linuxでも使用できます。このドライバを使用すると、自分でドライバを作ることなしに、IPをアプリケーション・ソフトウェア(以下アプリケーション)から制御ができます。その生成されたドライバの使い方も学んでいきましょう。

ただし、ここで解説するのはベアメタル・アプリケーション時のみです。Linuxのドライバの使用方法は、筆者のブログを参照してください。

なお、作成したCソースコードは、本書サポートページまたは、筆者のブログの「Vivado HLS勉強会3 (AXI4-Lite Slave) を公開しました」(<http://marsee101.blog19.fc2.com/blog-entry-3344.html>)に置いてあるので、それをコピーして使用してください。