

第3章 GPIO経由のI²C制御Cプログラムが、そのままハードウェア化できちゃう!?

Vivado HLSとZynqで カスタムI²Cコントローラを作る

早乙女 勝昭 Katsuaki Saotome

筆者はFPGAマガジンNo.6の特集で、Vivado HLSを色補間処理へ適用する事例を紹介しました。色補間処理はアルゴリズム系データ・バス設計で、いかにも高位合成ツールの得意そうな領域です。しかし「システムの上位(アルゴリズム系)だけではなく、もっと下位(プロトコル系)にも使えたら…」と思われる方もいるのではないのでしょうか。そこで今回は、外部信号制御インターフェース系としてI²C通信処理への適用方法を紹介します。

1. なぜ今さらカスタムI²C コントローラか?

昨今、データ伝送の高速化が目覚ましく、最新のFPGAデバイスでは数十Gps程度までも扱い可能な時代になりました。ならば数百kHz程度の通信など何ってこともないと思われるでしょう。

実際にI²Cインターフェースは非常に簡単なくみ(コラム1)で、いまだに非常に多くのシステムで活用されていて枯れた技術ともいえます。

通常、デバイス間1対1(マスター-スレーブ)のみで使用されることも多いのですが、マルチスレーブ(マスター)構成も可能です。

データ伝送方式は単純(2本の双方向信号線)で、その(電氣的な)仕様はかなり厳密に規格されています。しかし、フロー制御やプロトコル(データ・パケット構成)などはかなり自由度が高く、解釈の範囲が広い仕様となっています。

このあたりが原因で、市販のコントローラ(IP)とその添付APIドライバではうまく通信できないことがあります。ローテクなのに手強い…と言われるゆえんで、筆者も実務で何度も経験しています(コラム2)。

“FPGAエバンジェリスト”を自称(?)する筆者としては、カスタム・コントローラを設計することはやぶさかではないのですが、HDLでRTLベースの設計はそれなりに大変です(内部シーケンサの調整、バス・インターフェースの付加など)。またマネージメント(経営者)の立場としては、工数コストの観点からも承諾できません。

そこで開発工数短縮のため、ソフトウェア(CPU処理)で機能を実装・検証して、その後に高位合成ツール(Vivado HLS)でカスタム・ハードウェア化ができないか試してみました。

2. まずはGPIOでソフトウェア実装

今回使用する実機動作検証環境(FPGAボード)は、Xilinx社のZynqデバイスを搭載したZedBoard(Avnet社)です(写真1)。筆者は普段、もう一回り小さいZybo(Avnet社)を愛用しているのですが、ZedBoardには通信相手として適当なI²Cデバイス・

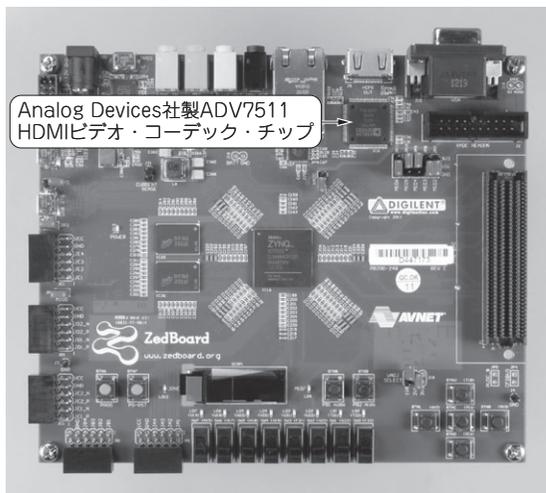


写真1 Zynq搭載評価ボード ZedBoard (Avnet社)

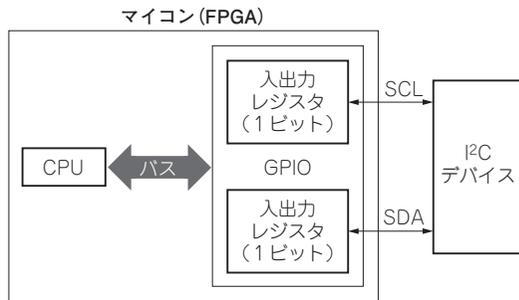


図1 システム構成