

Linuxドライバの基本から実際のドライバ作成、カスタム・ディストリビューション構築まで JPEGデコーダをLinux上から制御する ドライバ&アプリケーションの作成

石原 ひでみ Hidemi Ishihara

筆者は本書で、Zynq評価ボードZedBoard向けのオリジナル・ディストリビューションLinuxを作成し(FPGAマガジンNo.5掲載)、さらにZynqのFPGA部分にオリジナルのJPEGデコーダを実装しました(FPGAマガジンNo.6掲載)。今回はそれらの続編として、前回実装したJPEGデコーダをLinux上から制御するデバイス・ドライバと、そのアプリケーションを作成してみます。

1 Zynqに追加した回路モジュールをLinux上で制御する

● 今回はARMから制御できるようにする！

前回(FPGAマガジンNo.6)は、図1のようにZynqでAXI接続によるJPEGデコーダを紹介しました。

今回はCモデルの作成からHDLへトランスレーションし、Zynq上でデバッグするまでの紹介をしただけとなりました。単純に行うために、u-bootのメモリ・リード/ライト機能を使用して作成したJPEGデコーダをZynqに接続しはじめた初期段階でのデバッグを紹介したにすぎません。今回は実際にJPEGデコーダを使用する部分までは紹介していませんでした。せっかくZynqにはARMコアが載っているので、ARMコアから制御できなければPL部(FPGA部分)の回路モジュールを作成した意味がありません。

今回は組み込み業界の非リアルタイムOSの代表格になっているLinuxをZYBO(Digilent社)に実装し、Linux上でデバイス・ドライバを開発し、自身が作成した回路モジュール(JPEGデコーダ、HDMIエンコーダ)を制御します。

LinuxディストリビューションはFPGAマガジンNo.5で紹介したLinaro/Yoctoプロジェクトを使用し、自前でカスタムLinuxディストリビューションを構築して、ZYBOに適用しています。今回、紹介するZYBOのプロジェクト、Linux環境、およびドライバなどは本書サポート・ページなどからダウンロードできるようにしてあるので参考してください。

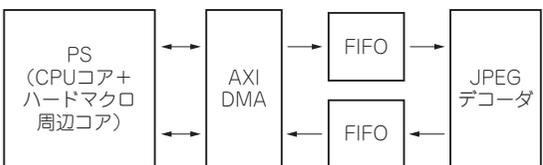


図1 前回(FPGAマガジンNo.6)で作成したJPEGデコーダの回路ブロック

● ソフトウェアの実装形態

まずはZynqのARM上で動作するソフトウェアをおさらいします。Zynq上で動作するソフトウェア構成には、大きく分けると次の三つがあります。

- ベアメタル・アプリ(OSのないアプリケーション)
- リアルタイムOS(FreeRTOSやT-Kernelなど)
- 非リアルタイムOS(Linuxなど)

特に今の時代、いろいろなアプリケーションを使用したシステム化のためにLinuxで実装する機会や要求は多いと思います。単純にZynqにLinuxを実装するだけであれば、Xilinx社がサポートしていたPetaLinuxを使用したり、各ベンダがサポートしているLinuxディストリビューションを使うことが多いのではないのでしょうか？

今回はZynqのPL部に追加している回路モジュールを制御する、Linuxのドライバとアプリケーションを解説します。本書はFPGAマガジンですが、今回の記事は少しFPGAから外れたところの話題になります。

● FPGA開発者はハードウェア屋？

餅は餅屋ということで、ソフトウェア屋にお任せになっているケースが多いと思います。FPGA技術者でFPGA以外にもデバイス・ドライバやアプリケーションなどを設計・開発している方はさほど多くないと思います。デバッグ・アプリケーションにしても作成する人は少ないと思います。それは、次のようなケースが多いからではないのでしょうか？

(1) ドライバを作成するのが難しい

ドライバはカーネル空間で動作し、直接ハードウェアが制御できたりすることから、ユーザ空間で作成するアプリケーションとは違って制約があります。これらを理解しながらアプリケーションを組んでいくのはなかなか難しいものです。アナログ回路やFPGAの設計がそれだけで閉じている職人芸のように言われることがあります。デバイス・ドライバの領域もこれに近いところがあります。そのため、デバイス・ドライバを作成するのが困難で、しいてはアプリケーション