SDSoCで使い慣れたLinuxが動く

SDSoCのLinuxディストリビューションを自在に切り替え可能！

アプリケーションを自由に開発できるSDSoC対応Linux環境の構築

石原 ひでみ Hidemi Ishihara

SDSoCは非常に簡単にアプリケーションの一部機能をハードウェア化できます．しかしSDSoCで標準的に用意されているLinuxは，カーネルのバージョンが古かったり，欲しいライブラリや機能がインストールされていないなど，アプリケーション作成時に不満も出てきます．ここではSDSoCのLinuxを，必要なライブラリなどを組み込んだLinuxディストリビューションに切り替える方法について解説します．

1 SDSoCでLinuxアプリケーションを作成する場合の問題点

本誌No.16のSDSoC特集を読まれた方は，ソフトウェアからシームレスにFPGAを使用できる環境が整いつつあると感じられたと思います．実際にソフトウェアの一部をそのままFPGAにオフロードすることが，現実的になってきています．

ただし実際に使用してみると，現状のSDSoCの環境でアプリケーションを作成することが難しいと感じた人もいると思います．それは現状のSDSoCで開発すると，最少限のLinuxシステムしか組み込まれないためです．

● SDSoCの生成ファイル

まず，SDSoCでアプリケーションをコンパイルして実機で実行するまでのおさらいです．SDSoCはベアメタル，FreeRTOS，Linuxの3つの環境で，ソースコードの一部分をFPGAにオフロードして使用することができます．今回環境を整備するのは，OSとしてLinuxを使う場合の環境です．

SDSoCでFPGAを活用していくには，SDSoCでプロジェクトを作成しソフトウェアを書いて，FPGAにオフロードしたい関数を選択してコンパイルします．SDSoCでコンパイルすると，プロジェクト・フォルダのDebug（またはRelease）の下にsd_cardフォルダが作成され，実機で実行可能なバイナリ・ファイルが生成されます．実機ではこのうち，次の2つのファイルを使用します．

- BOOT.BIN
- Image.ub

BOOT.BINは第1段階ブート・ローダ（FSBL），FPGAコンフィグレーション・ファイル（bitファイル），第2段階ブート・ローダ（U-Boot）の3つが1つのバイナリに結合されたファイルです．Image.ubはLinuxカーネル・イメージとルート・ファイル・システム（Linux環境），デバイス・ツリーの3つが結合されたファイルになっています．

● 最少限のLinuxシステムが起動する

Linux環境であるImage.ubは，/opt/Xilinx/SDx/2016.3/platforms（本記事はSDSoCを/opt/Xilinx/SDx/2016.3にインストールしたことを前提にしている）に用意されているイメージがそのまま使用されており，LinuxディストリビューションとしてはXilinx社純正のディストリビューションでもあるPetaLinuxが採用されています．

ターゲット・ボードがZYBOであれば，/opt/Xilinx/

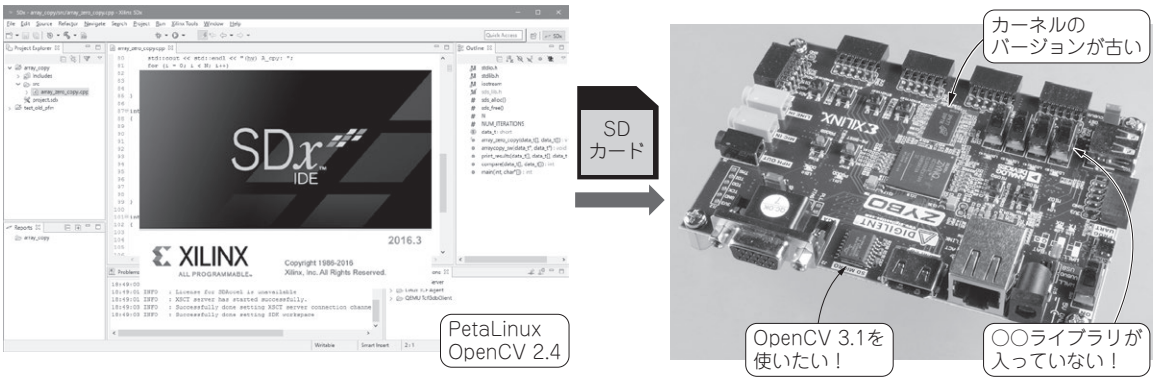

図1 SDSoCで開発すると最少限のLinuxシステムが起動する