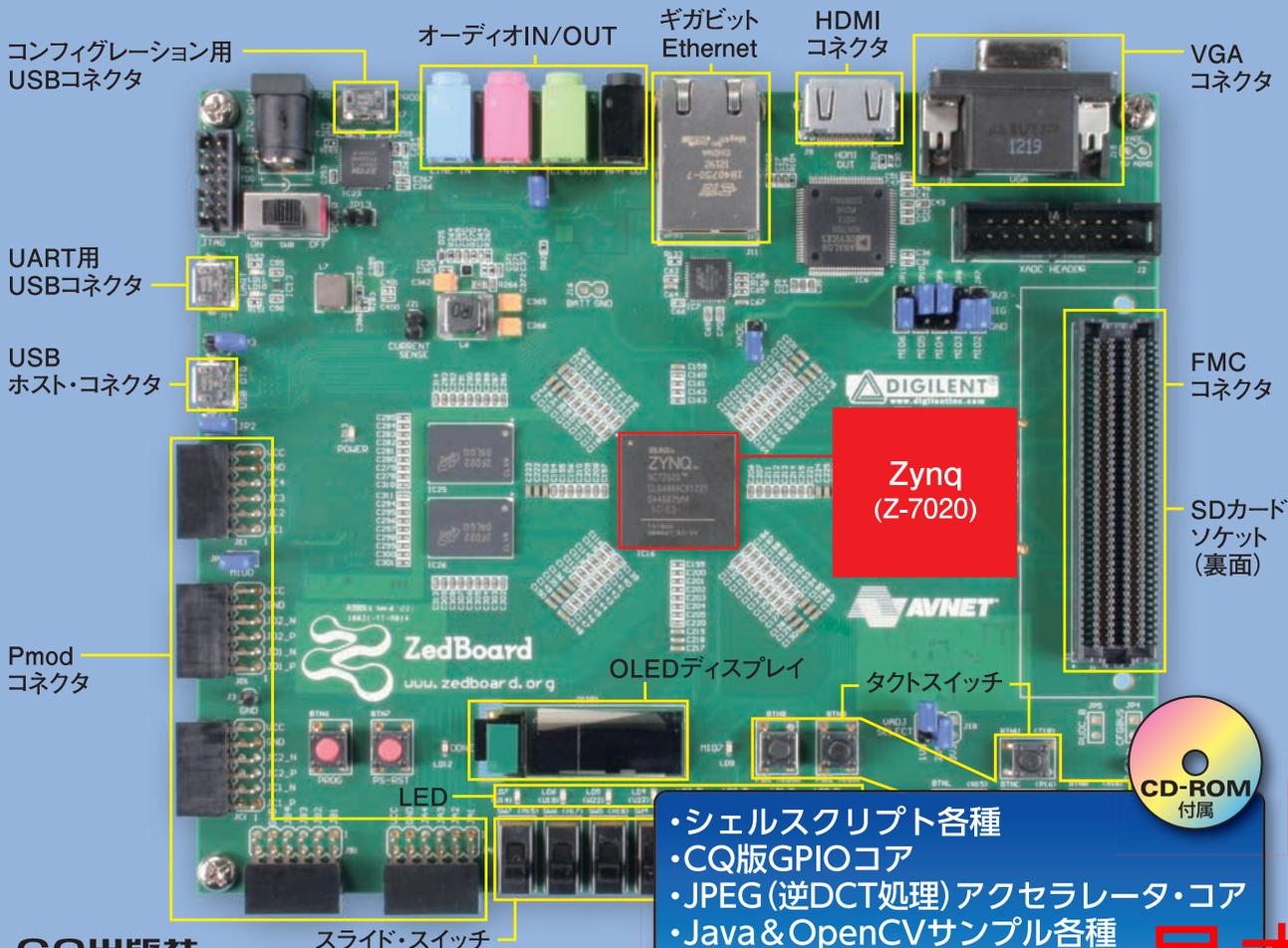


ARM Cortex-A9×2! Zynqでワンチップ Linux on FPGA

動作
クロック
最高
866MHz

エントリ・キット ZedBoard で高速画像処理 IC開発を初体験

鈴木 量三郎, 片岡 啓明 共著



- ・シェルスクリプト各種
- ・CQ版GPIOコア
- ・JPEG (逆DCT処理) アクセラレータ・コア
- ・Java & OpenCV サンプル各種



CQ出版社

第 1 章

ソフトもハードもプログラミング! ARM Cortex-A9搭載FPGA Zynq

ナンテいい時代! こんなに高機能なのに
今すぐキットで誰でも試せる

1.1

Zynq の概要

●単なる ARM コア内蔵FPGA ではない

Xilinx 社が提供する新しいデバイス“Zynq”（ジンクと発音）は、ARM Cortex-A9のデュアルコア・プロセッサとFPGAを搭載した新しいタイプのSoC(System On a Chip)です(写真1)。Zynqの内部を大ざっぱに示すと図1.1のようになります。同社ではARMコアや周辺コントローラ、メモリ・コントローラ部分を、PS(プロセッシング・システム)、FPGA部分をPL(プログラマブル・ロジック)と呼びます。

Zynqというデバイスを、FPGAを搭載したSoC、あるいはその逆のSoCを搭載したFPGAとだけ見るのは少々早合点かもしれません。今後の市場への展開と浸透度にもよるので慎重に評価しなければなりません。筆者は新しい分野のSoC が出現したという印象を持っ

ています。

実際にZynq搭載評価ボードを動作させてみた感じでは、Zynq自身はデバイスとしては完全にARMコアSoCととらえることができます。電源投入後、FPGAをコンフィグレーションせずに先にARMコアが立ち上がる仕様であったり、FPGA部をPLと呼ぶことから、Xilinx社からの「FPGAという枠を超えたSoC」であるというメッセージが受け取れます。

DSPを搭載したSoCは市場にいくつもあり、確かに、そのアプローチは多くの柔軟性を持っています。

ZynqはFPGAを高いレベルでSoCに統合しており、PL部には、既存のFPGAと同様ハードウェアを構成できます。うたい文句にあるようにまさにZynqは“All Programmable SoC”であり、ソフトウェアもハードウェアさえもプログラムできる、プログラマとしては腕の見せ所の多いデバイスといえます。

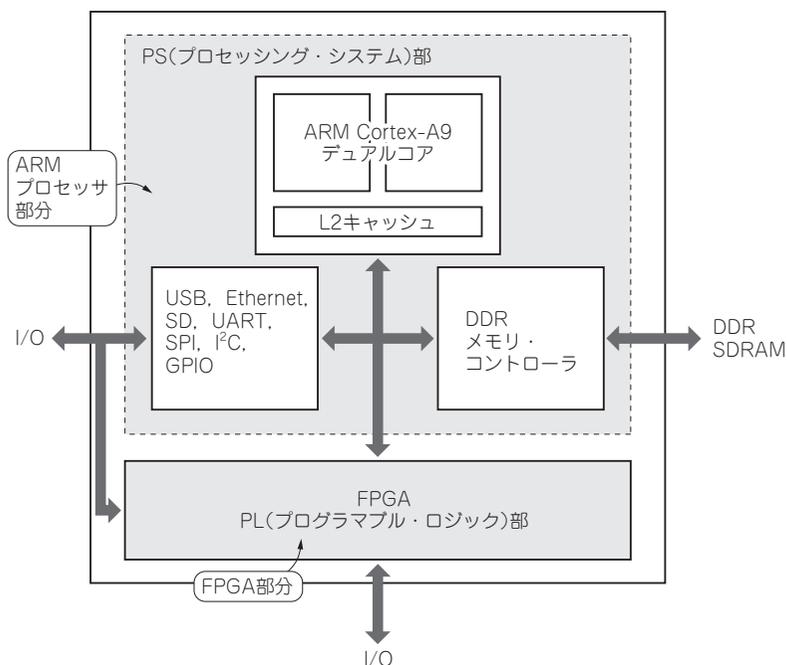


図1.1 ARMコアとFPGAを内蔵するZynq

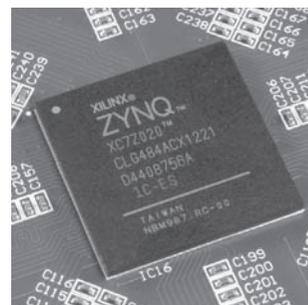


写真1.1 Zynqの外観

第2章

まずはZedBoardを動かしてみよう

付属のSDカードからLinuxを起動して、
コンソールから操作する

2.1 ZedBoardのセットアップとLinuxの起動

それでは、早速ZedBoardを動かしてみましょう。ZedBoardには標準でLinuxが書き込まれたSDカードが付属しています。これを使ってLinuxを起動してみます。

●ZedBoardのセットアップ

まずは必要な機材を用意します。

- ZedBoard一式
- ホスト・パソコン (Windowsが動くPCが良い)
- ZedBoard用ディスプレイ (DVI-D対応のもの)
- USBフラッシュ・メモリ (用意できれば)

PCには、TeraTermなどのターミナルソフトもインストールしておいてください。なお、とりあえずLinuxが起動することを確認する程度であれば、ディスプレイは無くてもかまいません。

さらに次のケーブルを用意して、ZedBoardと接続してください(写真2.1)。

- (1) HDMI↔DVI-Dケーブル (ディスプレイと接続)
- (2) Ethernetケーブル (PCと接続。ハブ経由でも直結でもOK)
- (3) マイクロUSBケーブル (コネクタJ14とPCを接続)

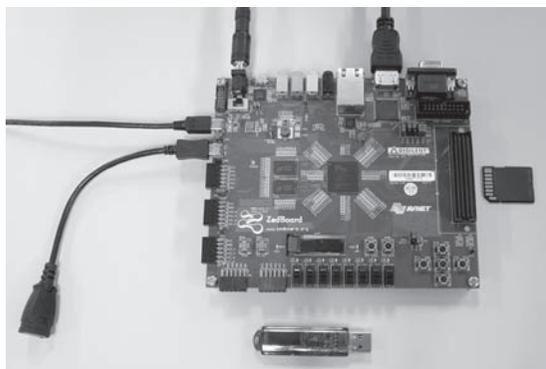


写真2.1 ZedBoardと各種ケーブルの接続

- (4) 付属のUSBホスト用コネクタ (コネクタJ13に接続)
- (5) 付属の電源ケーブル (付属の電源アダプタを接続)

写真2.1に、ZedBoardと各種ケーブルを接続した様子を示します。とりあえずLinuxが起動することを確認したいという場合は、(3)と(5)を用意して、ZedBoardとPCを接続してください。

●ジャンパ設定と電源ON

電源ONの前に、各種ジャンパの状態を確認してください。まず起動モードの設定ジャンパは、写真2.2に示すように、

- MIO2をGND側
- MIO3をGND側
- MIO4を3V3側
- MIO5を3V3側
- MIO6をGND側

に設定にしてください。またジャンパJP6をショート (MIO [0] をGND) し、さらにジャンパJP2とJP3をどちらもショートしてください (USBホスト電源供給)。

以上を確認してからZedBoardのSDカード・スロットに付属のSDカードを差し込み、ZedBoardの電源を入れてください。すると電源が入ったことを示す緑色のLED (LD13) が点灯します。

そして十数秒すると、Zynqが起動したことを示す青

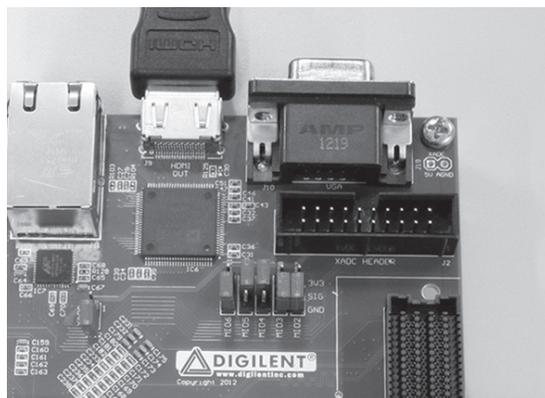


写真2.2 ZedBoardをSDカードから起動させるジャンパ設定

3.1

開発ツールの入手方法とセットアップ手順

ZedBoard を使っての開発には Xilinx 社の開発ツールが必要です。ここでは ISE Design Suite 14.4 を使用します (図 3.1)。Zynq の開発には無償版の ISE WebPACK が使用可能です。ISE Design Suite をインストールすると次のツールが同時にインストールされます。

• Xilinx Platform Studio (XPS)

プロセッサを使用したエンベデッド・システムを構築する際に使用するツールセットです。Zynq プロセッサに対してバスや IP コアを選択し接続することでハードウェアの構築を可能にします。以下 XPS と呼びます。

• Xilinx SDK (Software Developer Kit)

Xilinx 社が用意している C/C++ の統合開発環境です。Eclipse をベースにしています。以下 SDK と呼びます。

• PlanAhead

システム全体の設計・解析を可能とする統合ツールです。今回は XPS で設計したハードウェアのデザインの合成や、後述の XPS と SDK の橋渡しに使います。

各自のシステムに ISE Design Suite 14.4 (またはそれ以降) をインストールしてください。筆者は 64 ビット版 Windows 7 に ISE をインストールし使用しました。また、Linux のクロス・コンパイル環境として、必要に応じて 64 ビット版の Ubuntu サーバを使用しています。

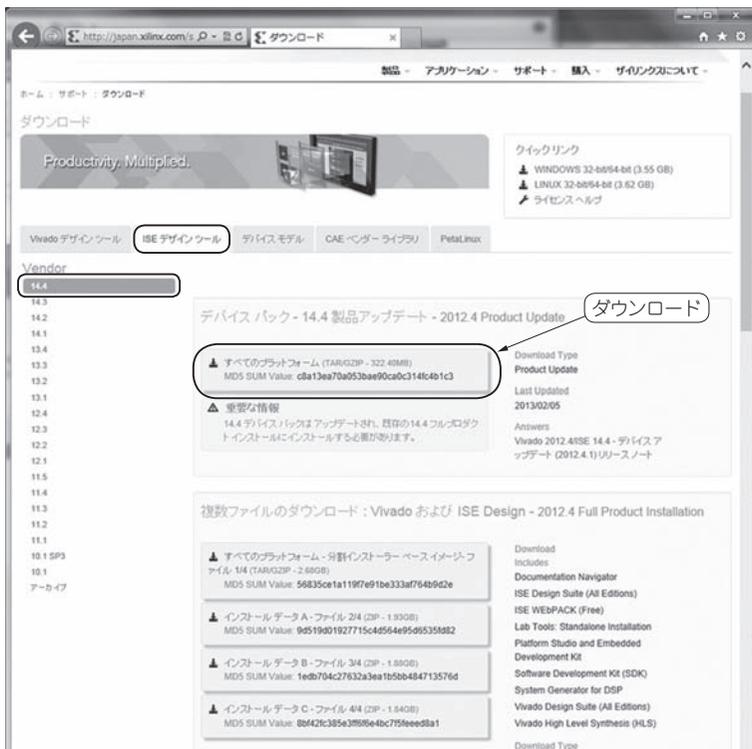


図 3.1 ISE Design Suite 14.4 のダウンロード

第4章

次世代ツールVivadoを使ってみよう

Zynq 対応開発ツールの最新版 Vivado を使った Zynq の開発手順

Xilinx社は新しいFPGA開発ツールVivadoを発表しました。登場当初はZynqに対応していなかったのですが、バージョンアップによりZynq対応になりました。ZedBoardのBSPも初めから用意されたので、実に簡単にVivadoを使えることがわかりました。

Vivadoは、今までのツール群をさらに統合を進めた形になっています。まず、プロセッサを使用した開発を中心に既存のIPコアを組み合わせてIPコアをインテ

グレートできます(IP Integrator)。また、VHDLやVerilog HDLを使った今まで通りのFPGAの開発もシミュレーション(XSIM)ももちろんサポートされています。できあがった回路図はインターフェースをつけてやればIPコアとしてパッケージ化できます。作成されたパッケージはツール非依存・業界標準のIP-XACTで、IP Integratorで汎用的に使うことができるようになります。



図 4.1 Vivado のダウンロード

Xylon社の リファレンス・デザインを使う

グラフィックス・アクセラレータが組み込まれた Zynq デザイン

前章までで、Linuxの立ち上げができるようになりました。この章ではちょっと寄り道をして、Linuxでの応用方法をのぞいてみたいと思います。

Linuxを使う大きな理由の一つは「OSを意識したくない」「既存のライブラリを有効に利用したい」など、ある目的に一直線に向かって行くための近道として利用するためでしょう。

この章ですでにあるリファレンス・デザインを、応用的に使う方法を示します。またこのリファレンス・デザインには、Linuxのフレーム・バッファ用ドライバが用意されているため、容易にLinuxから利用することができます。

この章ではリファレンス・デザインを使い、SDカードから簡単にLinuxを立ち上げることから始め、SDカードのパーティションを切り直して、よりLinuxらしい使い方へと変えていき、最終的にはLinux上でのプログラミングを可能にするところまでを説明します。

用意されたリファレンス・デザインを使用することで、IPコアのパワーをLinuxを通して簡便に利用できます。

5.1

Zynqのリファレンス・デザイン

● Zynqにはグラフィックス・コントローラがない

ZynqではさまざまなI/Oが最初から使えるようになっています。UART、SDカード・コントローラ、USB、ギガビットEthernetなど、一通りのインターフェースがすぐに使えるようになっています。ただ1種類だけ、それも意図的に固定的なハードウェアとして実装していないものがあります。それはグラフィックス機能、とりわけビデオ入出力に関する機能です。

近年のSoCはカメラ入力やDVI-DやHDMIなどの表示機能を一通り持っているものが多く、それはそれで便利なのですが、いざ使ってみるとその制約の多さなどから、「帯に短し褌に長し」となることもあるようです。

例えばビデオの入出力はインターフェースの規格だけで、単純なRGBの平行のタイプから、HDMI、Camera-Link、FPD-Link I/II/III、MIPI、MDDIなどなど、さまざまな規格が存在し、扱う形式もARGBの32ビット、16ビット、ベイヤーパターンなど、サイズにおいてはQVGA、VGA、SVGA、XGA、HD…、さらに各規格は年々バージョンアップされ、扱うサイズは大きくなる傾向にあります。

そんな中で、自分の欲しい機能をパーフェクトに備えているSoCがあるとは限りません。例えばQVGA/16ビット・カラーで十分なのに高機能で複雑なインターフェースしか用意されていなかったり、あるいは10ビットの高解像特殊用途に使いたくてもSoCが持っていない、持っていないも今度は他のインターフェースがないなどのケースです。運よく見つかったり、チップが特殊であればあるほど常にディスコン(生産中止)を気にすることになります。

Zynqの基本コンセプトは「すでにインターフェースとして確立され地位のあるUSB 2.0やEthernetなどはあらかじめ用意します。ユーザがフレキシビリティを必要とするビデオの入出力や画像処理部分はPL部で柔軟に対応しましょう。それがZynqというSoCです。」というところにあるようです。

● ZynqはFPGA部分に自分の望む回路を実装できる

図5.1にZynqのリファレンス・デザインとして公開されている、ソベル・フィルタ(輪郭抽出)の例を示します。入力はHDMI相当でFMC-IMAGEONボードを通して内部メモリに展開され、さらにその画像をハードウェアでソベル・フィルタを実行し、表示用のマルチレイヤ(OSDともいう)に対応したIPコアにより、HDMIからフルHD相当(1920×1080/60フレーム/秒)の出力をしています。OSはLinux、GUIなどの制御部分はQtで書かれています。

デモはクリック一つで、ハードウェアでのソベル・フィルタ処理とソフトウェアでのフィルタ処理を切り替えることができるようになっています。実際に試し

第6章

Linuxのカスタマイズ手順

Linux カーネルを最新のバージョンにしたり、
ドライバの追加も自由自在!

ここまでで、すでに用意されたZynqのLinuxシステムを自分なりに変更して、アプリケーションを使い、IPコアを有効利用するまでの簡単な方法を見てきました。より深く使いこなすなら、ライブラリの構築やカーネルの構築を自分で行う必要があることもわかりました。

この章では、ZynqのI/O (Input/Output)をおさらいした後、Zynqの初期化からルート・ファイル・システムの構築まで、よりディープに使えるシステムに仕上げていきます。

- Zynqの初期化の詳細
- Linuxの再構築
- デバイス・ツリー
- クロス・コンパイル
- ルート・ファイル・システムの構築

6.1

ZynqのMIO/EMIO

ここではMIOとEMIOを中心に、ARM SoCとしての構造と組み込みLinuxでの関係を整理しておきましょう。

●MIOとは

Zynqを含めSoCのチップは多くの機能を盛り込んでいます。それらの機能を、ユーザが全部使い切るということはまずないでしょう。Ethernetを二つ使うこともあれば、別のユーザでは一切使わないかもしれません。使わない機能は無駄にはなりません。

この時、Zynqを含めたほとんどのSoCは、全ての機能を同時に使えるような構造にはしていません。使わない機能のためにSoCのピン (SoCから出ている足)を割り当てるのは無駄なので、いくつかの機能を各ピンで共有させています (図6.1)。

ZynqではMIO (Multiplexed I/O) と命名していますが、要はマルチプレクサのことで、複数のI/Oが一

つのピンで共有されています。ZynqでのMIOのピンとしてどのように共有されているかを図6.2に示します。

例えば、MIOの28番はEthernet 1のtx ck, USB 0のdata, SPI 0のck, SDIO 0のck, SDIOのカード検出トライとプロテクト, SDIO 電源制御 0, NOR/SRAM のアドレス 13, CAN 1 の tx, UART 1 の tx, I²C 1 の ck, TTC1 の w, GPIO の 28 で共有されています。実際のピンは一つしかないので、そのうちの1機能を使うようにします。特定のレジスタに値を入れることで、その機能が選択されます。

ここで、USB 0を選択してしまうと他の機能が使えなくなるかというところではありません。共有されている機能の中には、他のMIOピンにアサインすることが可能なペリフェラルがあります。例えばCAN 1のtx/rxは、12組のMIOから選択可能です。

●EMIOとは

Ethernetは他のMIOピンにアサインされていません。それでは、CAN1を使うとEthernet1は使えなくなるのでしょうか？ Zynqには大きな特徴として

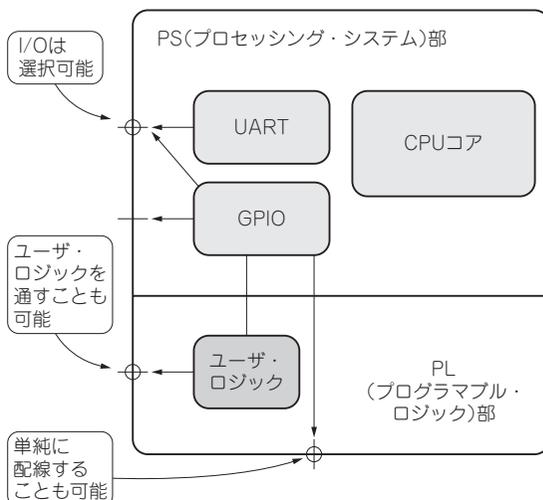


図6.1 I/Oの出し方

第7章

ハードウェア・ロジックの追加

標準で用意されている GPIO の追加から、独自のハードウェア CQ 版 GPIO の作成&組み込みまで

この章からは Zynq の FPGA 部分に焦点を当てます。ついに「やわらかいハード」の核心部分に到達です。ハードウェア・ロジック（ここでは IP コアと同義）を追加して自分のスーパー SoC を構築することが可能です。

ハードウェア・ロジック (IP コア) の追加方法にはいくつかあります。一つはツールである Xilinx Platform Studio (XPS) 上にすでに用意されているロジックを選択して追加する方法です。Xilinx 社が用意しているものからサードパーティが用意しているもの、無償のもの、有償のものと同様な選択肢の中から必要なロジックを選んで自分のシステムの中に組み込むことができます。まさに自分だけの ARM SoC を作ることができるようになります。

この手法はビジネスを加速させるうえでも重要です。必要なロジック (IP コア) を調達して SoC を組み立て、市場の開発スピードに負けないようにするためにも、あるものを有効に使い、一番上のアイデア部分で特徴のあるものを開発して差別化していくという方法は今後の主流になるでしょう。

一方、必要なロジックが特殊であるために自分で開発しなければならないこともあるでしょう。昔からある秋葉原的手法と言ってもよいかもかもしれません。アマチュアの方にとっては世界のどこにもない一品物がくれる魅力的な手法と言えます。

ここではその二つの方法を紹介します。さらに作成した IP コアを、別のデザインに再利用する方法も紹介します。

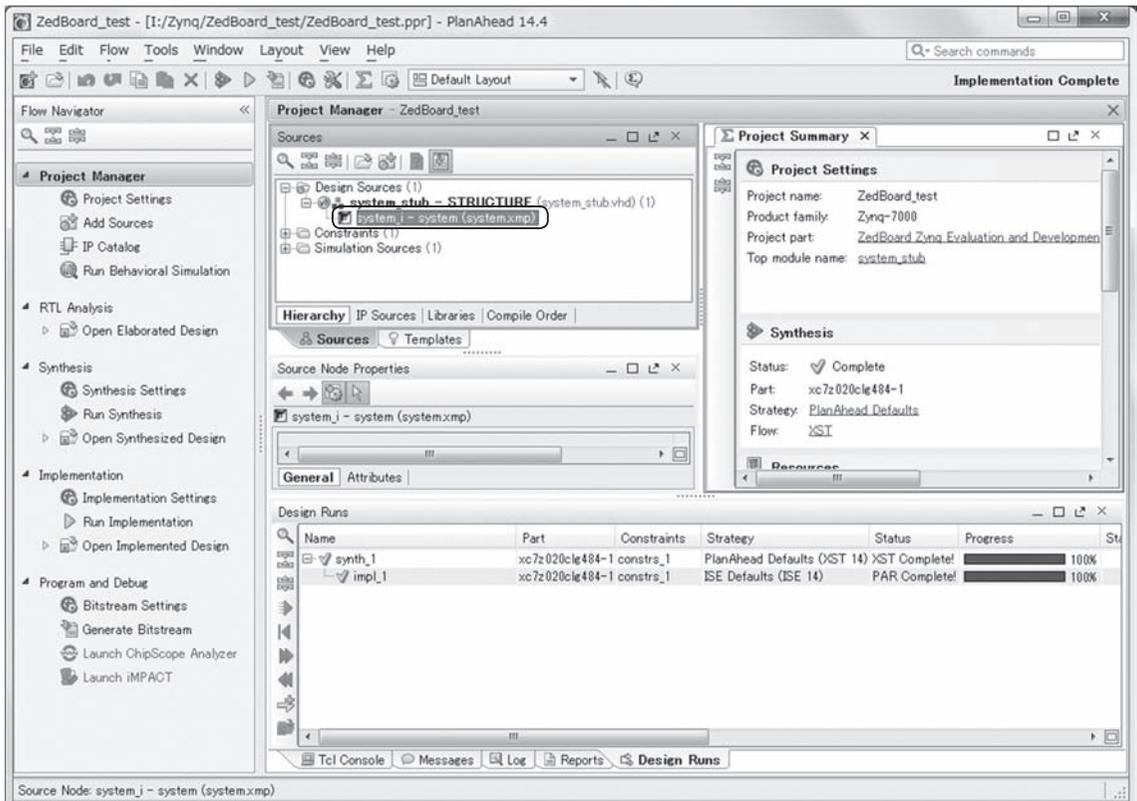


図7.1 PlanAheadからXPSを呼び出す

ISBN978-4-7898-4609-7

C3055 ¥5400E

CQ出版社

定価：本体5,400円（税別）



9784789846097



1923055054004

このPDFは、CQ出版社発売の「ARM Cortex-A9×2！ZynqでワンチップLinux on FPGA」の一部見本です。

内容・購入方法などにつきましては以下のホームページをご覧ください。

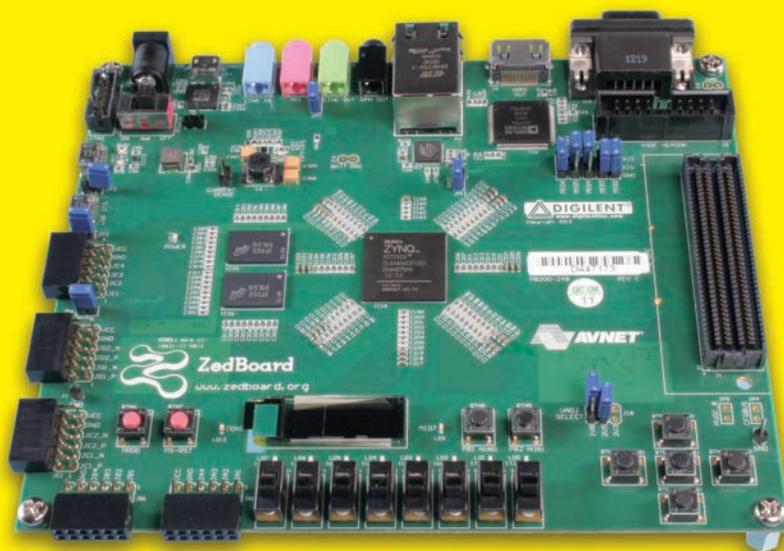
内容 <http://shop.cqpub.co.jp/hanbai/books/46/46091.htm>

購入方法 <http://www.cqpub.co.jp/order.htm>

Design Wave

ARM Cortex-A9×2! ZynqでワンチップLinux on FPGA

エントリキット ZedBoard で高速画像処理 IC 開発を初体験



本書には ZedBoard は付属しておりません。
ZedBoard の購入は、CQ 出版 WebShop まで！
<http://shop.cqpub.co.jp/hanbai/books/I/1000125.html>

見本