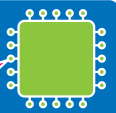


音声&画像処理/画像転送/測定器…
なんでも高速・高安定・高機能

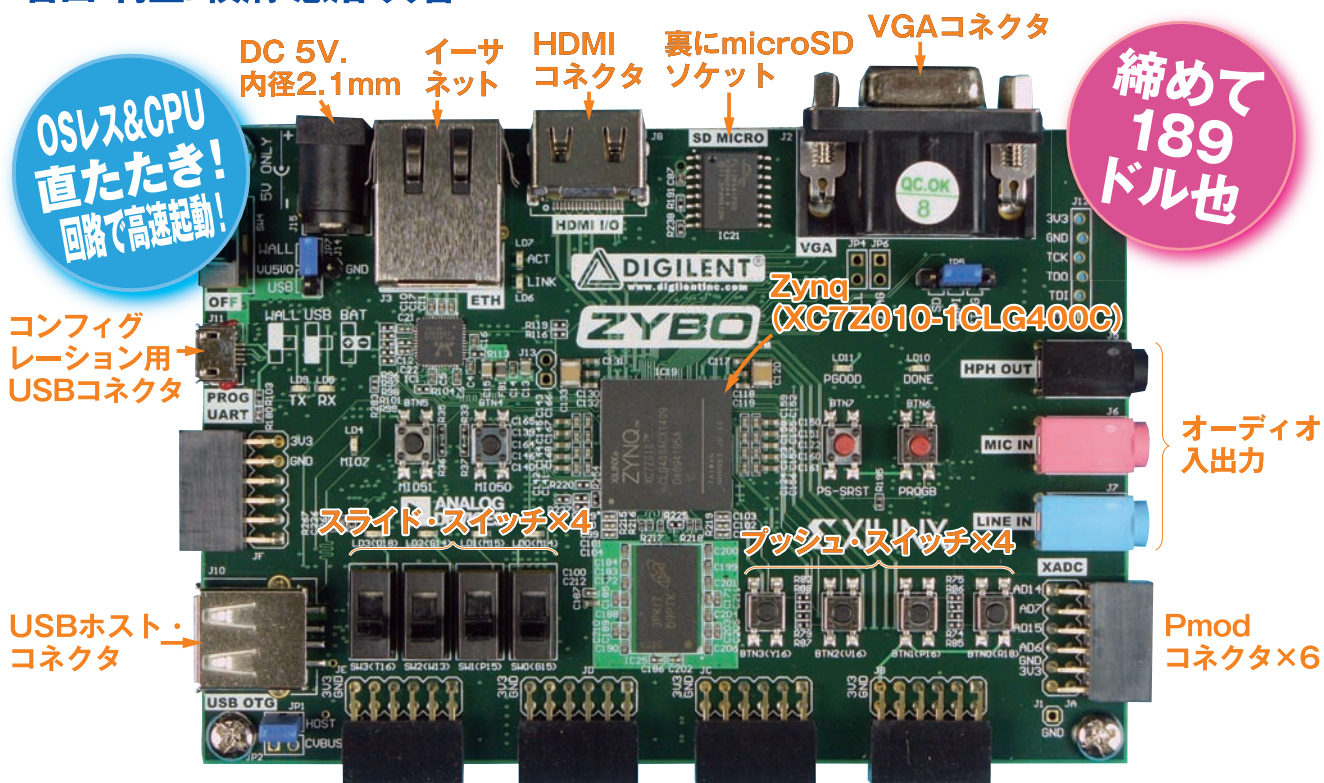
トライアル
シリーズ



IoT時代のハイパフォーマンス電子工作

FPGAパソコン ZYBOで作る Linux I/Oミニコンピュータ

岩田 利王/横溝 憲治 共著



縮めて
189
ドル也

ご購入はこちら .

<http://shop.cqpub.co.jp/hanbai/books/48/48091.htm>

ZYBOの主なスペック

- ザイリンクス Zynq-7000 (XC7Z010-1CLG400C)
- オンボードJTAGプログラミングとUART-USB変換回路
- 512MB×32 DDR3, 1050Mbps帯域幅
- 128Mbit QSPIシリアル・フラッシュ
- microSDスロット (Linuxファイル・システムをサポート)
- 高帯域幅周辺コントローラ: 1Gイーサネット, USB 2.0, SDIO
- 低帯域幅周辺コントローラ: SPI, UART, I²C
- デュアル・ロール (ソース/シンク) HDMIポート
- 16bit/ピクセル VGAポート

- 3モード (1Gbit/100Mbit/10Mbit) イーサネットPHY
- OTG USB 2.0 PHY (ホスト, デバイスをサポート)
- 外部EEPROM (48bitグローバルでユニークなEUI-48/64互換識別番号でプログラム済み)
- ヘッドホン, マイク, ライン入力付きオーディオ・コーデック
- GPIO: 6×プッシュ・ボタン・スイッチ, 4×スライド・スイッチ, 5×LED
- Pmodコネクタ×6 (1プロセッサ専用, 1デュアル・アナログ/デジタル)

見本

CD-ROM
付き

サンプル・プログラム収録

CQ出版社

CQ出版社

見本

第1章 ARM ベース SoC Zynq とは何か

1.1 リッチな GUI とリアルタイム信号処理をこなすシステムを作るには

図 1-1 のような画像処理システムを作りたい... みなさんならどうしますか？

PC を使えば、図 1-1 のような画像のエッジ検出は比較的簡単にできそうです。でももっと小型で安価なボードで実現しなければならない...そんな状況での解決策を探ってみます。

✓ ワンチップ・マイコンではパワー不足

図 1-1 の GUI (Graphical User Interface) はまるで PC のようです。その実現には DDR (Double Data Rate) などの外部メモリが必要になるでしょうし、クロックも数百 MHz は必要でしょう。従って、Cortex-M0 クラスのワンチップ・マイコンではどう見ても間に合わなさそうです。

✓ パワフルなシングル・ボード・コンピュータなら間に合いそう

写真 1-1 は、AM335x (ARM Cortex-A8 コア) 搭載の BeagleBone Black (BeagleBoard.org 製) です。クロックは 1GHz, 外部メモリは 512Mbyte DDR3 なので何とかなりそうです。実際、参考文献 (1) ではこのボードに Linux を搭載した例がいくつか紹介されています。

そして、Linux 上に OpenCV などをインストールすれば、図 1-1 のような画像処理も可能になりそうです。

並行してリアルタイム信号処理も必要... さあ困った

さらに図 1-2(a) のようなシステムを考えます。Linux を動かし、さらにリアルタイム信号処理をや

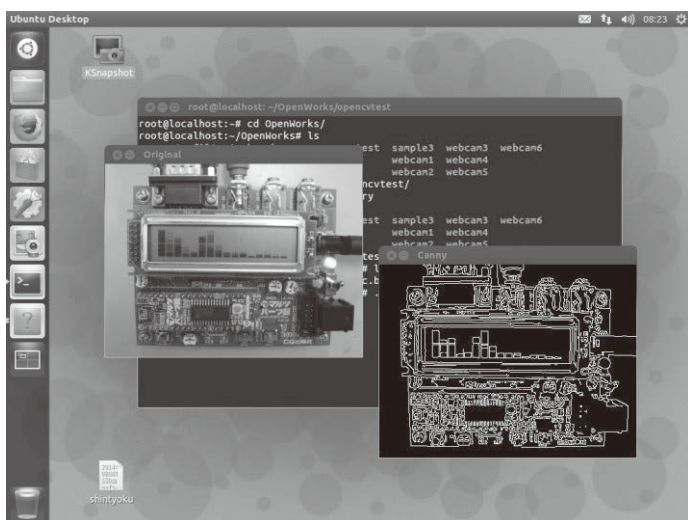
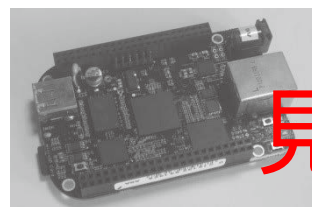


図 1-1 こんなリッチな GUI でエッジ検出したい！



見本

写真 1-1 BeagleBone Black

ろうというものです。

✓ 次から次へとデータが入って来る

扱うデータはリアルタイム、つまり音声信号のように A-D 変換されたデータが途切れることなく入ってきます。そして、信号処理したデータを途切れることなく D-A コンバータに渡します。

✓ Linux のような OS を載せてしまうとリアルタイム信号処理は難しい

そのような処理を図 1-2(a) のように OS で行うことを考えます。この場合、A-D、D-A のサンプリング周波数が低ければ間に合いそうですが、高くなると間に合わなくなる可能性があります。仮に間に合ったとしても、OS の状態（何を走らせているか、何をインストールしたか）によっては出力が不安定になるかもしれません。

✓ OS から独立して動くロジックを考える

そこで信号処理の部分は OS と切り離すことを考えます。図 1-2(b) のようにロジックで信号処理を行い、OS はロジックにパラメータを渡すだけにします。こうすれば OS の状態に左右されることはなく、リアルタイム性を保持できそうです。

✓ 小型/安価なボードでそんなことができるの？

実現の可能性があることは分かりました。しかし...そういえば一つ条件がありました。小型で安価なボードで実現すること...そんな無茶な要求に応える解決策なんてあるのでしょうか？

✓ シングル・ボード・コンピュータではロジックを組むことができない

BeagleBone Black なら Linux を載せ、C 言語などでアプリケーションを作って OS 上で走らせることができます。

しかし、残念ながら、HDL (Hardware Description Language) でロジックを組むことはできません。

✓ FPGA では Linux のような OS は厳しい

逆に、FPGA (Field Programmable Gate Array) はロジックを自由自在に使うことができますが、

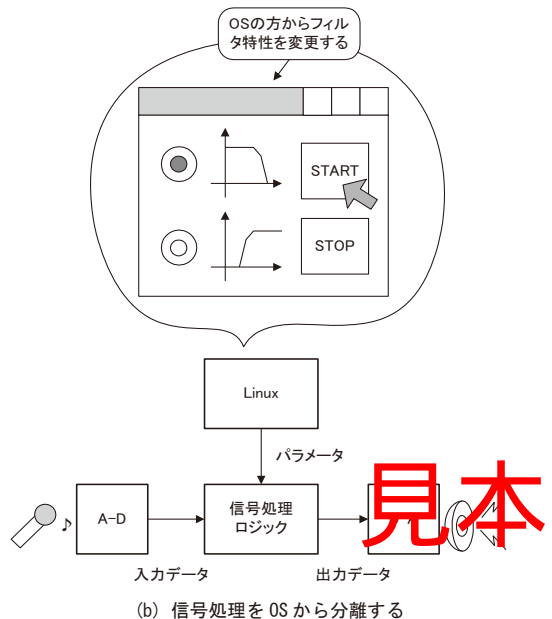
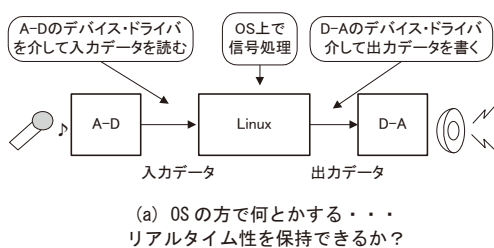


図 1-2 リッチな GUI と高速でリアルタイムな信号処理を兼ね備えるには？

Linux のような本格的な OS に耐え得るようなプロセッサを持っていません。

✓ 高速な ARM Cortex-A9 を内蔵した FPGA デバイス Zynq

しかし、そのような要望に応えるデバイスがあります。ザイリンクスの提供する「Zynq」というデバイスは「ARM Cortex-A9」を2個内蔵しています。スマートフォンなどにも採用されているプロセッサなので、Linux のような OS を載せても十分動作します。

✓ プロセッシング・システムとプログラマブル・ロジックを備え持つ最新デバイス Zynq

Zynq はもちろん従来の FPGA 的な部分も持ち合わせています。すなわちプロセッシング・システム (PS) に OS を任せ、プログラマブル・ロジック (PL) にはリアルタイム信号処理を任せることができるのです。

✓ Zynq を搭載した小型で安価なボード ZYBO 誕生！

Zynq を搭載したボードでは「ZedBoard」が有名ですが、最近「ZYBO」という、より小型で安価なボードが登場しました（原稿執筆時点）。

本書ではこの ZYBO ボードを使って、図 1-1 のような画像処理や図 1-2 (b) のようなリアルタイム信号処理システムなどを実現します。

1.2 従来の FPGA を進化させた SoC の誕生！

ゲート・アレイ→FPGA→SoC (System-on-a-Chip) とデバイスが進化するにつれ、設計の効率や自由度が飛躍的に増してきました。ここではその経緯や背景について説明します。

ちょっとした PC+α のシステムを作れる SoC

「SoC」とは、従来複数の LSI で行われていた機能を一つの LSI に統合する方式、またはそのようなデバイスのことを示す言葉です。なお、これは一般的な定義であり、何を SoC と呼ぶかはそのメーカーによって微妙に違います。

✓ マイクロプロセッサを進化させた SoC

例えば、テキサス・インスツルメンツは従来の ARM プロセッサ・システムに GPU (Graphics Processing Unit) やディスプレイ・コントローラ、その他周辺回路を加えてワンチップ化したもの

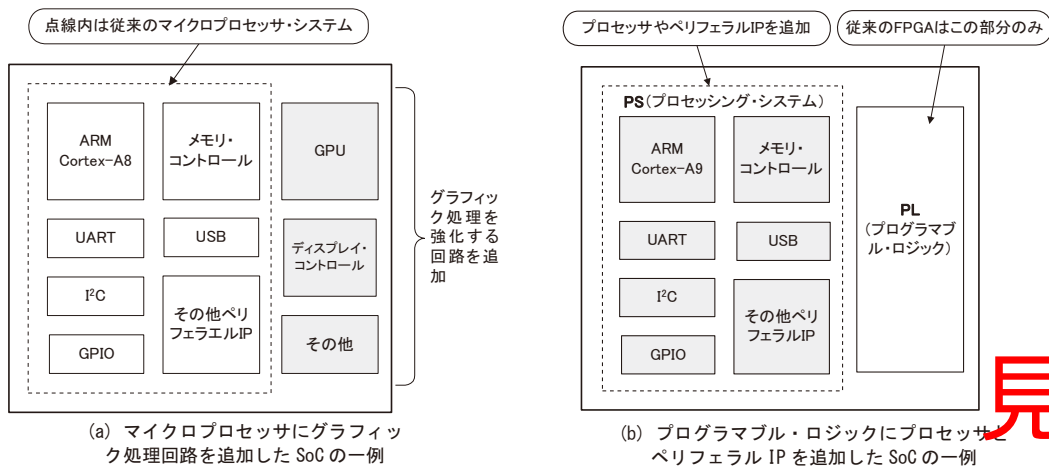


図 1-3 二つのアプローチによる SoC

を SoC と呼んでいます。それは BeagleBone Black (写真 1-1) に搭載されており、Linux などの OS を載せることができます。この例は、従来のマイクロプロセッサ・システムに周辺回路を追加したアプローチによる SoC です [図 1-3(a)]。

✓ FPGA を進化させた SoC... 本書ではこれを紹介する

それに対して図 1-3(b) のように、従来プログラマブル・ロジックだけだったものにプロセッサおよびペリフェラル IP を追加してワンチップ化する、というアプローチによる SoC もあります。ザイリンクスの Zynq がその典型例で、従来の FPGA に ARM Cortex-A9 という高速プロセッサを 2 個追加したものです。同社はこの Zynq ファミリを「SoC」と呼んでいます。

✓ SoC≒Linux を載せることができるデバイス

SoC へのアプローチを 2 通り紹介しましたが、どちらも Cortex-A シリーズの高速プロセッサを搭載しているため、Linux のような何らかの OS を載せるような使い方が大半のようです。

FPGA とは何か。そして FPGA はどのように SoC へと進化したのか

そもそも「FPGA」とはどのようなデバイスなのでしょう。ここではその歴史的背景をもとにイメージを伝えたいと思います。また、FPGA メーカーが声高に SoC と叫び出した経緯も推し量ってみたいと思います。

✓ FPGA はゲート・アレイの進化形

FPGA は Field Programmable Gate Array の略で、「現場でプログラムできるゲート・アレイ」という意味です。それでは「ゲート・アレイ」とはなんなのでしょう。

✓ FPGA がなかった頃によくあった話

1990 年代のまだ FPGA が普及していなかった頃、デジタル回路を設計する際は「74 シリーズ」という AND ゲート、OR ゲートやフリップフロップが数個入った IC を並べて配線していました。回路が小さければそのまま製品化できます。しかし、回路が大きくなって IC が 100 個にもなると、基板サイズや信頼性、製造コストや保守といった問題が噴出してきます。

✓ ゲート・アレイ化してワンチップにしていた

そこで設計者はその回路図を IC メーカーに渡し「ゲート・アレイ化」を依頼します。ゲート・アレイというデバイスにはあらかじめ AND ゲート、OR ゲートやフリップフロップといった論理素子(ゲート)が多数敷き詰められており、IC メーカーは回路図を見ながら依頼通りに適宜配線してワンチップ化します(図 1-4)。

✓ 夢も悪夢も見させてくれたゲート・アレイ

IC メーカーはみな大企業で、当時はゲート・アレイ化を依頼するのに何百万円も支払っていました。出来上がった物がちゃんと動作すればよいのですが、動かなかったらまた作り直し、しかも自分のせいで製品のスケジュールが何カ月も遅れる...それを想像すると恐ろしくて夜も寝られなくなったことを著者は思い出します。

✓ FPGA は「ゲートの大平原」を自分で配線するイメージ

そこで「現場でプログラムできるゲート・アレイ」の登場です。設計者は「ゲートの大平原」である FPGA を自分で配線してゲート・アレイを自作できるようになりました。しかも失敗したらすぐにやり直しが利きます。以前のように大金を払った上に悪夢にうなされるようなことはなくなったのです。

見本

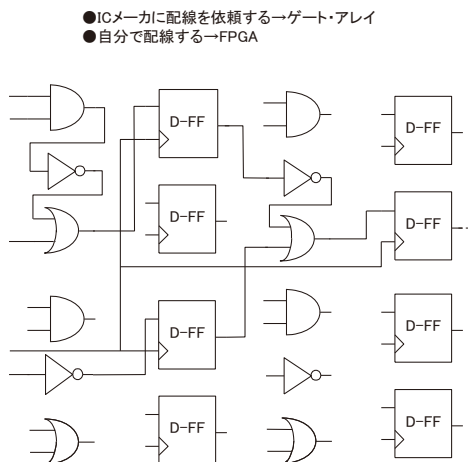


図 1-4 あらかじめ敷き詰められた論理素子（ゲート）を適宜繋ぐ

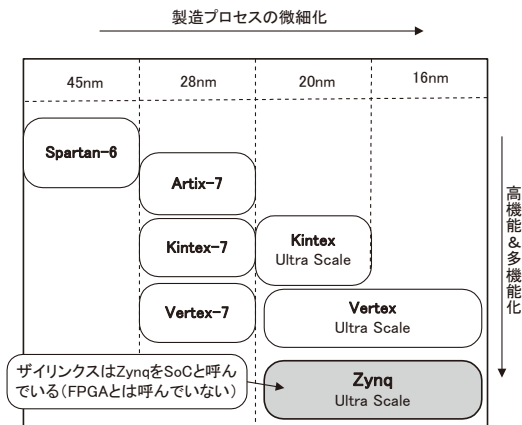


図 1-5 ザイリンクスの FPGA のラインアップ

✓ 行けども行けども果てしなく続く大平原となった最近の FPGA

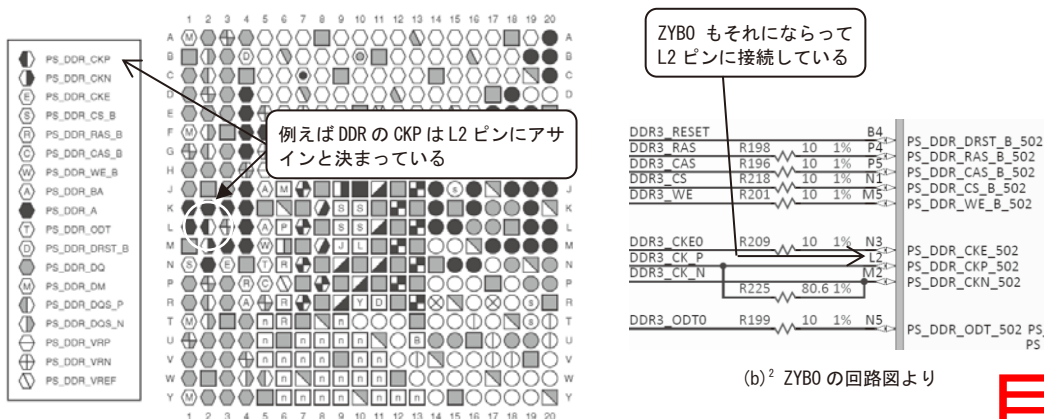
FPGA は「ゲートの大平原」であると書きましたが、近年はデバイスの製造プロセスの微細化が進み過ぎて大平原がとてつもなく広がってしまいました。有名な「ムーアの法則」によると「デバイス上のゲート数は 18 カ月で倍」と言われていますが、設計する側のレベルがそのペースで上がるわけではないため、FPGA 上のゲートを全然使い切れないといった非効率性が露呈してきました。

✓ ゲートの大平原を埋めるロジックが不足する事態に

この問題は FPGA メーカーにとっても頭の痛い問題だと思います。熾烈な微細化競争により高密度なデバイスを製品化しても、ユーザから「そんなに大きくても使い道がない」と言われてしまいます。

✓ 微細化競争の行き着く先とも言える SoC

そこで FPGA メーカーはゲートの大平原に「ARM Cortex-A9」という巨大な建物とその周辺施設を建てたのではないかと想像します。そうすることによりゲート使用の非効率性が解消できますし、さらに SoC という名前を付けて差別化と高付加価値化を図れるからだだと思います。



(a)¹ XC7Z010 CLG400 のピン・アサイン規定

見本

図 1-6 DDR2 を繋ぐ I/O ピンは決まっている

1.3 Zynq の用途とアーキテクチャ

Zynq は普通の FPGA とどう違うのか

ザイリンクスが供給する FPGA には図 1-5 のようなものがあります。ただし、同社は Zynq のことを FPGA とは呼んでおらず、これだけは SoC と呼んでいます（原稿執筆時点）。

✓ Zynq では PS 部がメイン、PL 部がサブとなるイメージ

従来の FPGA との決定的な違いは ARM Cortex-A9 が入っていることです。またそれに伴い、以下のような違いが派生しています。

- 内部が PS 部（プロセッシング・システム）と PL 部（プログラマブル・ロジック）に分かれている。
- UART, I²C, SPI, USB, GPIO などのペリフェラルはハード・マクロ IP¹として PS 部に置かれている。
- 外部メモリ（DDR3 など）を使うことが前提になっており、それら専用の I/O ピンはあらか

Vivado の Diagram から Zynq をダブルクリックして開く

ここがハイライトされている。MIO で SDIO が選択されて MIO40~MIO45 から入出力される

microSD カードを繋ぐことが可能なピン

(a) Vivado でペリフェラル IP のピン・アサインをする

microSD カードは MIO40~MIO45 に繋がっている

```

PS_MIO36_501  A11 OTG_CLK
PS_MIO37_501  A10 OTG_DATA5
PS_MIO37_501  E13 OTG_DATA6
PS_MIO38_501  C18 OTG_DATA7
PS_MIO39_501  D14
PS_MIO40_501  R226 SD_CCLK 40.2 1%
PS_MIO41_501  C17 SD_CMD
PS_MIO42_501  E12 SD_D0
PS_MIO43_501  A9 SD_D1
PS_MIO44_501  B13 SD_D2
PS_MIO45_501  B15 SD_D3
PS_MIO46_501  D16 OTG_RESETN
PS_MIO47_501  B14 SD_CD
PS_MIO48_501  B12 UART_RXD_OUT
PS_MIO49_501  C12 UART_TXD_IN
PS_MIO50_501  B13 PBO
PS_MIO51_501  B9 PBI
PS_MIO52_501  C10 ETH_MDC
PS_MIO53_501  C11 ETH_MDIO
  
```

図 1-7 microSD カードを繋ぐピンはある程度決まっている

(b)² ZYB0 の回路図より

見本

¹ ハード・マクロ IP とはあらかじめ作り込んだ IP で、ユーザはそれを変更/削除することはできない。それに対しソフト・マクロ IP とはゲートを配線することで作る IP。Spartan など従来の FPGA の場合、UART などはソフト・マクロ IP として提供されている。

じめ決まっている (図 1-6) .

- ペリフェラル用の I/O ポートもある程度決まっている. 例えば microSD カードのピン・アサインは図 1-7 のように開発ツール Vivado (第 3 部第 1 章などで言及) で設定する.
- コンフィグレーションは基本的に PS 部が行う.

✓ Zynq なら Linux を載せたい!

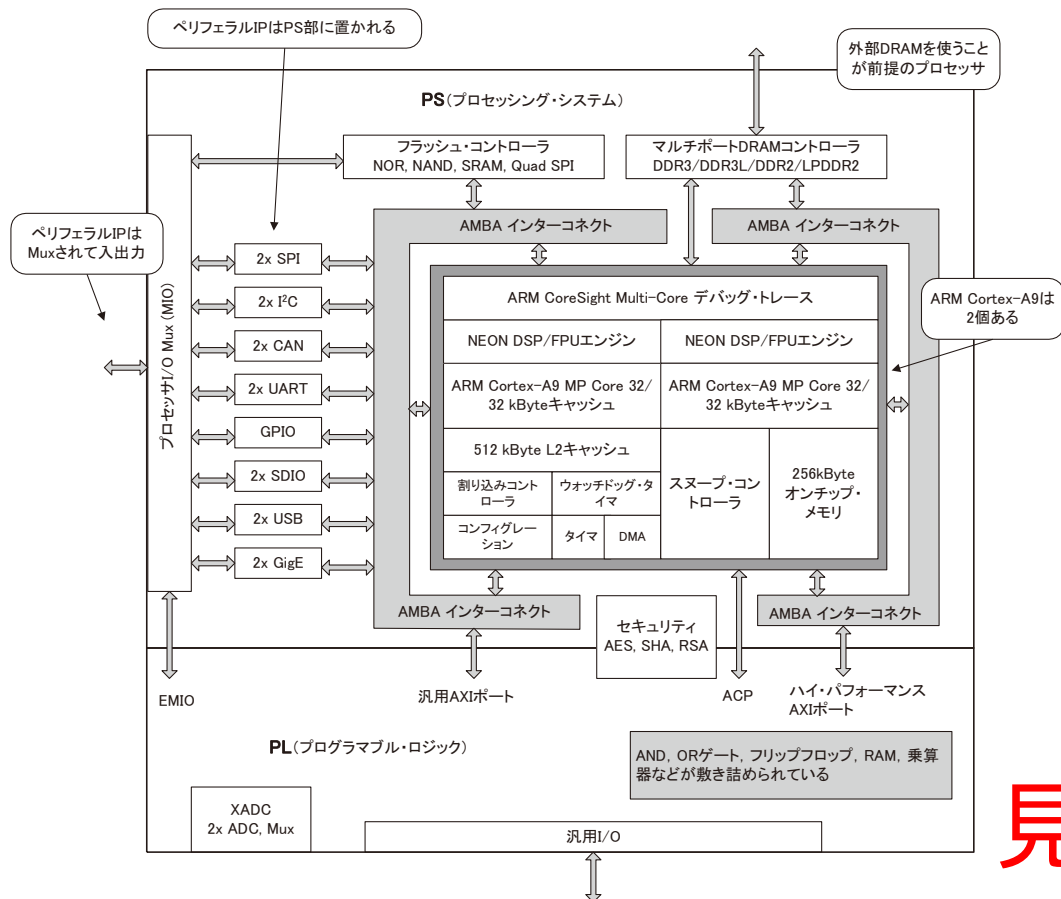
ARM Cortex-A9 はとても高速でパワフルなプロセッサです. それに DDR3 など高速/大容量のメモリを接続して走らせるわけですから, ベア・メタル (ARM を裸で, OS なしで使う) ではちょっともったいない気がします. Zynq の本当の実力を体感するためには, やはり Linux のような OS を載せたいところです.

Zynq のアーキテクチャ

Zynq は PS 部と PL 部に分かれているのが特徴です. 図 1-8 は ZYBO ボード搭載 XC7Z010-1CLG400C のアーキテクチャです.

✓ プロセッサとペリフェラルの PS 部

図 1-8 の上の部分が PS 部で, その中心に ARM Cortex-A9 が 2 個あります. ZYBO 搭載のデバイスでは動作速度は最高 650MHz になります.



見本

図 1-8 Zynq のアーキテクチャ

ARM Cortex-A9 は DDR3 など外部メモリを使うことが前提とされており、それらを制御するための DRAM コントローラがあります。外部メモリとして DDR3/DDR3L/DDR2/LPDDR2 がサポートされています。また、DRAM を接続するピンはデバイスごとに決まっています (図 1-6 参照)。

UART, I²C, SPI, USB, GPIO などのペリフェラルがハード・マクロ IP としてここに置かれています。これらは MIO (Multiplexed I/O) というピンから選択的に入出力されます (図 1-7 参照)。

フラッシュ・メモリ・コントローラや SDIO コントローラもあり、フラッシュ・メモリや SD メモリーカードからブートすることもできます。

✓ 従来の FPGA と同じ PL 部

PL 部はザイリンクスの 7 シリーズである Artix FPGA とほぼ同等です。しかし、コンフィグレーションは PS 部または JTAG ポートから行われ、従来の FPGA ように PROM からこの部分を直接コンフィグレーションすることはできないようです。

このエリアに AND, OR など各種ゲート、フリップフロップ、RAM、乗算器などが多数敷き詰められており、ユーザはそれらをプログラマブルに使用することができます。

1.4 ZYBO ボードの仕様と開発環境

SoC デバイス Zynq を搭載するボードとして ZYBO は安価であり小型でもあります。また国内の複数の代理店が扱っており、その入手性も問題なさそうです (2016 年 2 月時点)。機能的、容量的にも SoC/FPGA の入門としては十分でしょう。開発ツールも無償/無期限で使用できます。

これが SoC 入門の定番ボード ZYBO !

Zynq を搭載した定番ボードといえば「ZedBoard」がありますが、本書で使用するのはより小型で安価な「ZYBO」です (写真 1-2)。

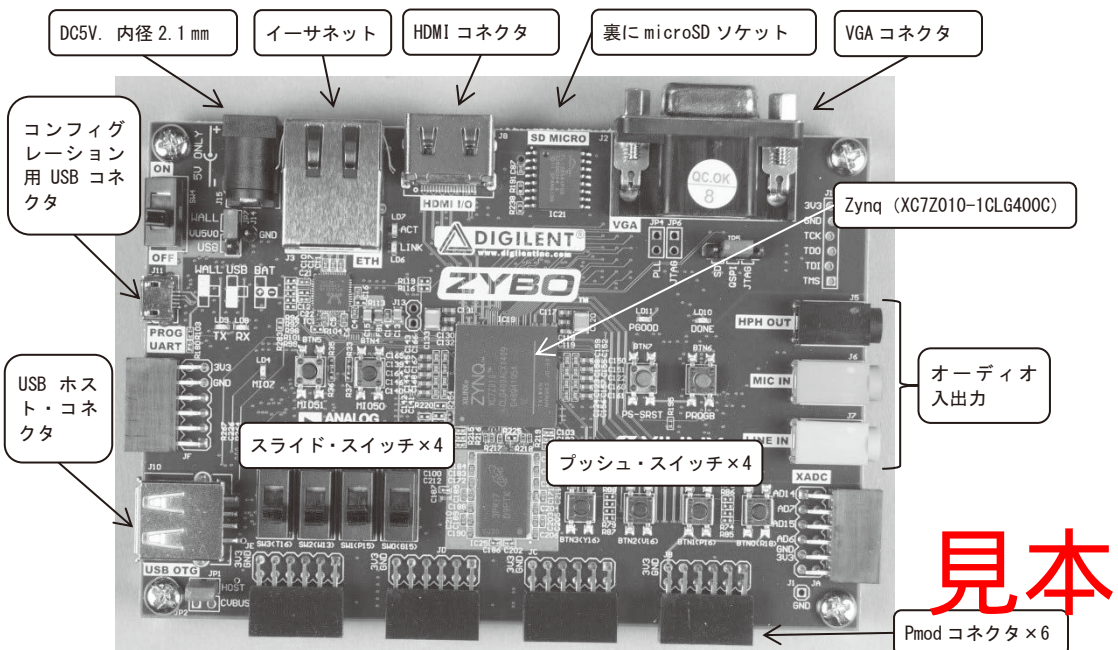


写真 1-2 コンパクトなボードにすべてが詰まった感じの ZYBO

✓ SoC の入門なら ZYBO がおすすめ

表 1-1 は ZedBoard と ZYBO の比較です。サイズも価格も半分程度になっています。メインの Zynq デバイスは 1 グレード下がりますが、周辺デバイス、周辺機能はほぼ遜色ないと思います。

✓ Zynq デバイスの性能も SoC 入門には十分

それでは Zynq デバイスの性能はどうでしょうか。表 1-2 に示すように ZedBoard 搭載の Zynq デバイスと比べて、ZYBO のそれは速度的には遜色ありません。プログラマブル・ロジックの容量は小さくなりますが、それでも SoC の入門としては十分すぎるほどだと思えます。

ZYBO を入手して火入れしてみよう！

ZYBO を購入して箱を開けてみるとボードしか入っていないと思いますが、MicroUSB ケーブルを用意して繋ぐと PC から電源が供給されます。しかし、SoC のような複雑な使い方をする場合、消費電流が増えるので AC アダプタ (5V, 内径 2.1 mm) を入手して電源をとるのがよいでしょう。その際は JP7 (DC ジャックの隣にあるジャンパ) を「WALL」側にしてください。

✓ 出荷時の設定では Quad SPI フラッシュからブートされる

VGA または HDMI コネクタからディスプレイに繋いで電源を入れると ZYBO は Quad SPI フラッシュからブートアップされます。写真 1-3 のような格子模様が出ると思います。そうならない場合は JP5 (VGA コネクタの隣にあるジャンパ) が「QSPI」になっているか確認してください。

表 1-1 ZedBoard より小型で安価な ZYBO

ボード名	ZedBoard	ZYBO (本書で使用)
搭載 Zynq	XC7Z020-1CLG484	XC7Z010-1CLG400C
搭載メモリ (揮発性) [Byte]	512M DDR3	512M DDR3
搭載メモリ (不揮発性) [bit]	256M Quad SPI フラッシュ	128M Quad SPI フラッシュ
基板サイズ [mm]	160×134	122×84
価格 (税込)	54,000~67,000 円	22,000~25,000 円
購入先 (一例)	アヴネットなど	秋月電子通商, アヴネットなど
製造元	Digilent 社	Digilent 社
搭載メモリ (揮発性) [Byte]	512M DDR3	512M DDR3
搭載メモリ (不揮発性) [bit]	256M Quad SPI フラッシュ	128M Quad SPI フラッシュ

表 1-2 SoC の入門には十分なデバイス速度と容量

Zynq デバイスの型番	XC7Z020-1CLG484 (ZedBoard 搭載)	XC7Z010-1CLG400C (ZYBO 搭載)
プロセッサ・コア	Dual ARM Cortex-A9	Dual ARM Cortex-A9
プロセッサ周波数 [Hz]	667M (MAX)	650M (MAX)
プログラマブル・ロジック・セル	85k (ASIC ゲート換算 1300k)	28k (ASIC ゲート換算 430k)
フリップフロップ	106,400	35,200
ブロック RAM [Byte]	560K	240K
DSP スライス	220	80
外部メモリ (揮発性)	DDR3/DDR2/LPDDR/SDRAM	DDR3/DDR3L/DDR2/LPDDR2
外部メモリ (不揮発性)	2×Quad SPI フラッシュ, NAND, NOR	2×Quad SPI フラッシュ, NAND, NOR
パッケージ	484 ピン BGA	400 ピン BGA
ユーザ I/O 数	54	54
デバイス単価 [円]	17,395 (Digikey, 執筆当時)	8,483 (Digikey, 執筆当時)

見本

✓ ブートは3種類. LinuxはmicroSDカードから

図1-9にZynqのコンフィグレーション方法を示します. デフォルトではJP5の設定によりQSPIフラッシュからコンフィグレーション・データが読み込まれ, ブートアップされます.

JP5をJTAGにするとPCからコンフィグレーション・データをダウンロードすることができます. JP5を「SD」にするとmicroSDカードからのブートになります. LinuxのようなOSを走らせる場合はこの設定になります.

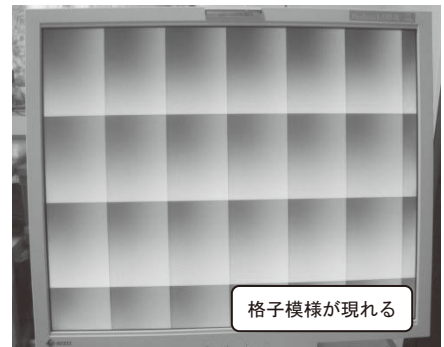


写真1-3 出荷時はQuad SPIからブート

ZYBO ボードの開発環境

Zynq 内蔵の ARM Cortex-A9 を使った設計を行う場合, 開発ツールは以下の2通り考えられます.

- PlanAhead + Xilinx Platform Studio + Xilinx SDK
- Vivado + Xilinx SDK

本書では無償/無期限で使える後者 ([Vivado WebPACK](#)) を採用することにします.

✓ 従来のFPGAの開発ステップとはだいぶ様相が変わっている

Vivado はザイリンクスの最新開発ツールですが, 著者が使ってみたところ, 従来の同社のツールとは以下のように様相が変わっています.

- 操作がグラフィカルになった. IP ブロックを配線して設計するのが基本 (ほぼ自動配線なのがうれしい).
- 以前は ISE Project Navigator, PlanAhead, Xilinx Platform Studio, iMPACT など, ツールが散在して分かりにくかったが, Vivado に統一された感があり, だいぶすっきりした.
- だいぶ軽くなった印象. 特に Xilinx Platform Studio は重かったが, Vivado は論理合成や配置配線が速くなった気がする.

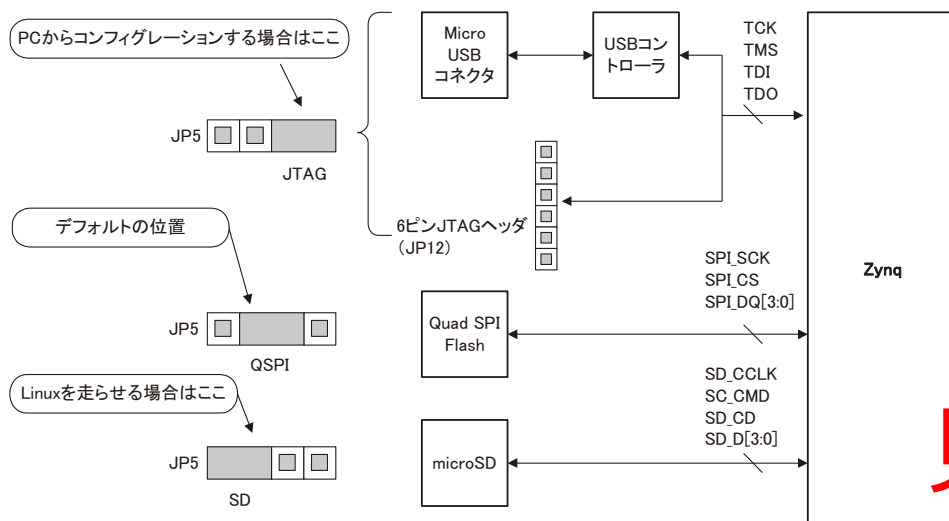


図1-9 JP5のジャンパ位置でコンフィグレーションが切り替わる

第2章 PSでHello World, PLだけでLED点滅

●本章で使用する Vivado
Vivado WebPACK 2015.4

ZYBO に搭載されている SoC Zynq の設計には、ザイリンクスの Vivado と SDK (Software Development Kit) が必要です。ここでは、これらをインストールして、簡単な設計例を題材に Zynq の設計手順について説明します。

最初は、Zynq の PS 部 (Processing System) で “Hello World” を表示するプログラムを作成します。次に、PL 部 (Programmable Logic) の回路のみで、LED を点滅させるプログラムを作成します。

2.1 Vivado のインストール

Vivado には有償版と無償版があります。無償版は WebPACK と呼ばれ、設計できるデバイスと機能に制約があります (表 2-1)。

32bit OS 用の Vivado は 2014.4 が最終バージョンで、2015.1 からは 64bit OS 用のみがリリースされています。

また、2015.4 からは WebPACK でもロジック・アナライザ (波形観測) と HLS (C/C++ 言語を使用する高位合成) が利用可能になっています。

WebPACK のインストール手順を図 2-1~図 2-4 に示します。

ツールをダウンロードするには、ザイリンクスのサイトのアカウントが必要です。アカウントがない場合はユーザ登録してアカウントを作成する必要があります。

初めにインストーラをダウンロードします。そのインストーラを使って Vivado と SDK をインストールします。

インストールが完了したらライセンスを申請して、ライセンスを PC に登録します。

表 2-1 Vivado 有償版と WebPACK の違い

対象/デバイス	Vivado 有償版	WebPACK (2015.4 以降)	WebPACK (2015.3 以前)
対象デバイス	7 シリーズ, Zynq の全デバイス	Artix-7 (7A15T - 7A200T) Kintex-7 (7K70T, 7K160T) Zynq (XC7Z7010 - XC7Z7030)	←同じ
論理合成	○	○	○
配置配線	○	○	○
シミュレータ	○	○	○
デバイス・プログラマ	○	○	○
IP Integrator	○	○	○
ロジック・アナライザ	○	○	×
シリアル I/O アナライザ	○	○	×
高位合成 (HLS)	○	○	×

見本

SDK 使用の Windows 用 Vivado HLx 2015.4 Web Install

↓ SDK 使用の Windows 用 Vivado HLx 2015.4 Web Install (EXE - 49.32 MB)
MD5 SUM Value: cd91c8758f9ea474ad68ea794254c357

↓ SDK 使用の Linux 用 Vivado HLx 2015.4 Web Install (BIN - 76.98 MB)
MD5 SUM Value: c599ca2e5cae8721efc1841465a46fce

ザイリンクスのダウンロード・ページ (<http://japan.xilinx.com/support/download.html>) でインストールする OS 用のインストーラをクリック。
2015.1 以降のバージョンは 64bit OS のみ対応している。32bit OS を使う場合は 2014.4 のインストーラを選択する



Q
Sign In | (0)

APPLICATIONS
PRODUCTS
DEVELOPER ZONE
SUPPORT
ABOUT

Sign in to Download File

User ID

Password

Forgot your password?

Note: Xilinx employees should Sign In using their Xilinx ID and password.

Why register?

Get easy access to all your design needs when you register

- Sign up for newsletters
- Join Xilinx forums
- Download software
- Order products
- Get technical help
- Gain access to restricted content
- Register for training

[Create Account](#)

ログインを求められ、アカウント登録済みの場合はログインする。登録してない場合は「Create Account」をクリックしてアカウント登録する



APPLICATIONS

Download Center

Primary...net*

[Wi]

ログインすると Download Center のページが表示される。ここで [Next] をクリックすると、インストーラ (Xilinx_Vivado_SDK_2015.4_1118_2_Win64.exe) がダウンロードされるので、インストーラを実行する。
実行時に「MSVCR110.dllがないため、プログラムが開始できません」と表示された場合は
<http://www.microsoft.com/ja-jp/download/details.aspx?id=30679> から
VSU4¥vcredist_x86.exe
VSU4¥vcredist_x64.exe
をダウンロードして実行してから、再度インストーラを実行する



Vivado 2015.4 Installer - Welcome
Welcome

We are glad you've chosen Xilinx as your platform development par Environment, Software Development Kit and Documentation Nav...

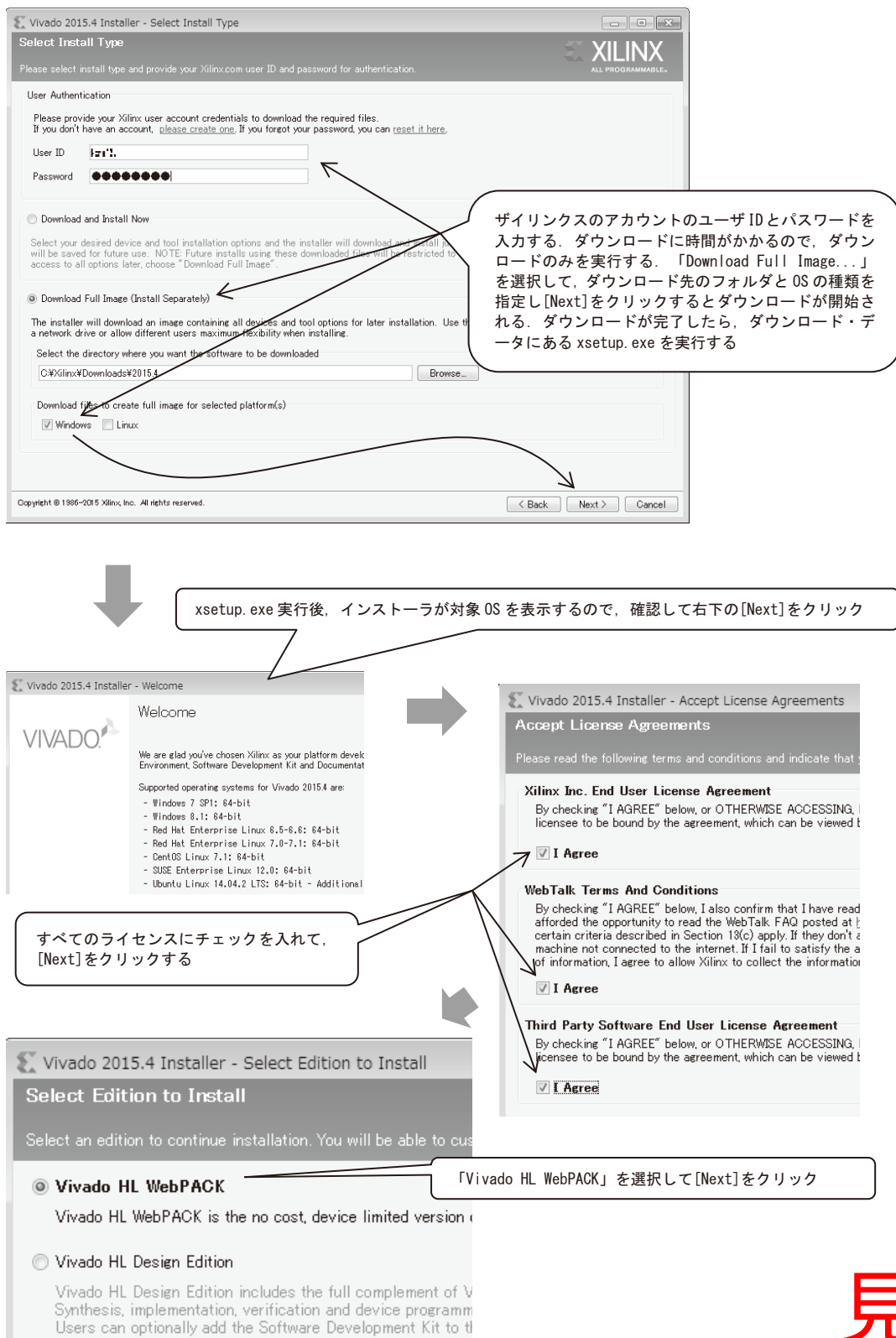
Supported operating systems for Vivado 2015.4 are:

- Windows 7 SP1: 64-bit
- Windows 8.1: 64-bit
- Red Hat Enterprise Linux 6.5-6.8: 64-bit
- Red Hat Enterprise Linux 7.0-7.1: 64-bit
- CentOS Linux 7.1: 64-bit
- SUSE Enterprise Linux 12.0: 64-bit
- Ubuntu Linux 14.04.2 LTS: 64-bit - Additional library

インストーラの「Welcome」画面で対象 OS が表示される。確認して右下の[Next]をクリック

見本

図 2-1 Vivado, SDK のインストール手順 1



見本

図 2-2 Vivado, SDK のインストール手順 2

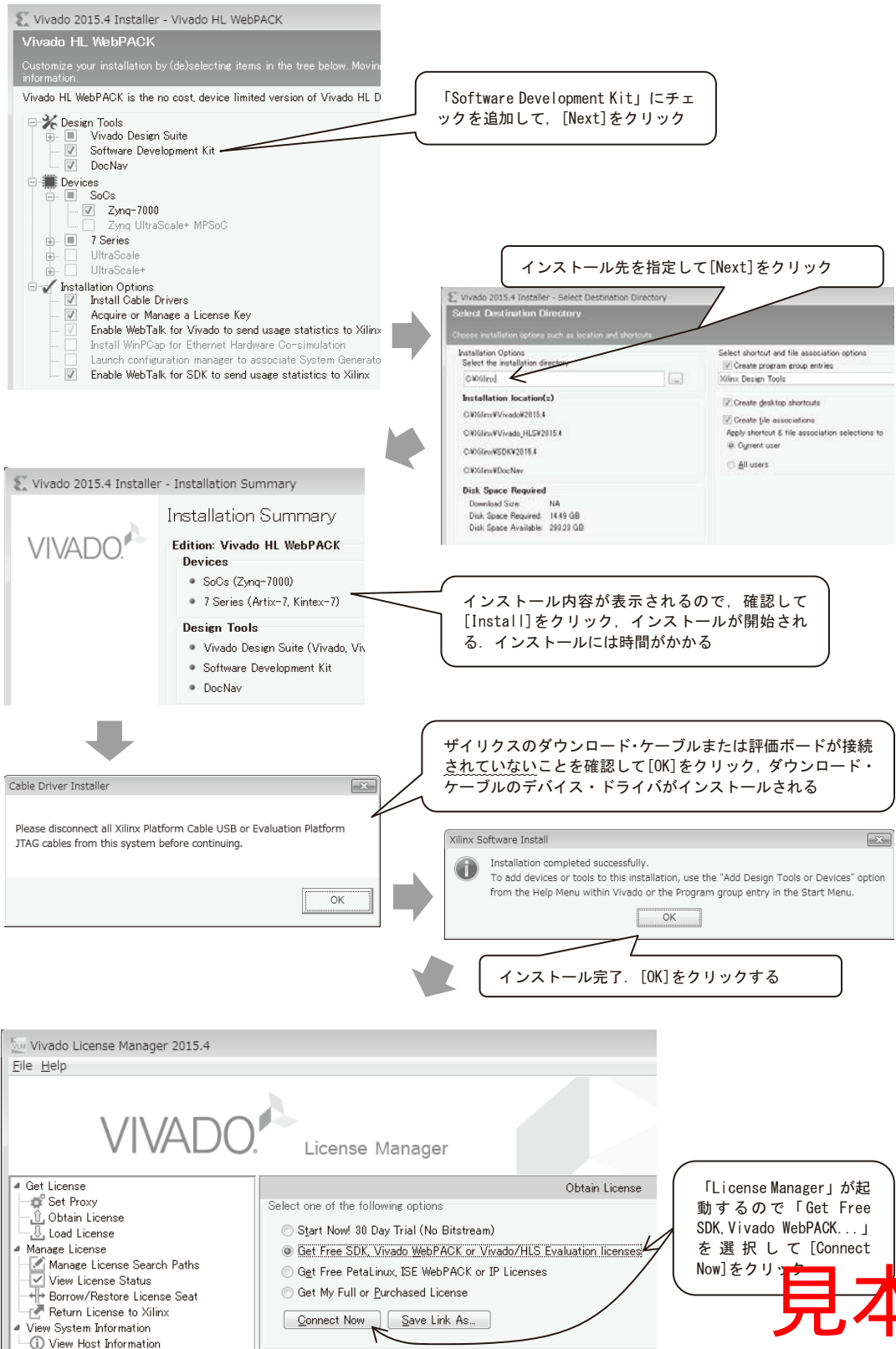


図 2-3 Vivado, SDK のインストール手順 3

Product Licensing

Account: **HIMAWARI CORPORATION**

Create New Licenses | Manage Licenses | Orders | Manage Users

Have a Voucher to Redeem? Redeem Now

Evaluation and Search the Eucres catalog table below

Create a New License File for Account: HIMAWARI CORPORATION

Create a new license file by making your product selections from the table below. Floating an same license file.:

Certificate Based Licenses

Product	Type
<input checked="" type="checkbox"/> Vivado Design Suite: HL WebPACK, Node-Locked License Certificate-Based Lic...	Certificate - No Charge
<input type="checkbox"/> ISE WebPACK License	Certificate - No Charge
<input type="checkbox"/> Peltiour Tools License	Certificate - Evaluation
<input type="checkbox"/> Peltiour Tools License, Floating License	Certificate - Evaluation
<input type="checkbox"/> Vivado HLS Evaluation License	Certificate - Evaluation

Generate Floating License | **Generate Node-Locked License**

PRODUCT SELECTION

Product Selections	Product	Type	Available Seats	Subscription End Date	Revised Seat	Remaining Seats
<input checked="" type="checkbox"/>	Vivado Design Suite -	No Charge	1/1	None	1	

2 SYSTEM INFORMATION

License: Node
Host ID: Any

3 COMMENTS

Comments:

Next Cancel

[Next]をクリックする

[Next]をクリックする

ライセンスが発行されてメールで送付される

[Load License] をクリック, [Copy License] をクリックして, メールで送られてきたライセンス・ファイル (Xilinx.lic) を指定する

Vivado License Manager 2015.4

File Help

VIVADO License Manager

- Get License
 - Set Proxy
 - Obtain License
 - Load License**
- Manage License
 - Manage License Search Paths
 - View License Status
 - Borrow/Restore License Seat
 - Return License to Xilinx
 - View System Information
 - View Host Information

Certificate Based Licenses

Click the 'Copy License' button to copy a certificate-based license. It will automatically detect valid, node-locked licenses. (*1)

Copy License...

Activation Based Licenses

Click the 'Activate License' button to load a response

Waiting for Activation License

You have chosen to generate a license on the Xilinx Licensing Site.

Vivado License Manager is now attempting to automatically download and install an Activation-based license record (xml) from the license you generated on the Xilinx Licensing Site.

If you are using Certificate-based licenses (.lic) or if you wish to load the Activation-based license record at a later time manually, please press Cancel.

Retrieving activation license...

Cancel

[License Manager]に戻り, 「Waiting for Activation License」で[Cancel]をクリック

Vivado License Manager 2015.4

File Help

VIVADO License Manager

View License Status

Certificate Based Licenses:

License Name	Tools/IP	Expiration Date	Version Limit
Analyzer	Tools	Permanent	2016.12
HLS	Tools	Permanent	2016.12
SDK	Tools	Permanent	2016.12
V.WebPACK	Tools	Permanent	2016.12
Vivado_WebPACK	Tools	Permanent	2016.12
VIVADO_HLS	Tools	Permanent	2016.09

[View License Status] をクリック, 「Vivado WebPACK」「SDK」のライセンスが有効になっていることを確認して「License Manager」を終了

Vivado License Manager

License installation was successful: C:\Xilinx\Xilinx.lic

OK

ライセンスのコピーが完了。[OK]をクリック

見本

図 2-4 Vivado, SDK のインストール手順 4

ISBN978-4-7898-4809-1

C3055 ¥6000E

CQ出版社

定価：本体6,000円（税別）



9784789848091



1923055060005

ZYBOに搭載しているFPGA Zynqの 主な仕様

- 650MHz デュアル・コア ARM Cortex-A9 プロセッサ
- 28000 ロジック・セル
- 240KB ブロック RAM
- 80 DSPスライス
- オンチップ・デュアル・チャンネル, 12bit, 1MSps A-D コンバータ (XADC)
- 8 DMA チャンネル付き DDR3 メモリ・コントローラ

対応ホストPCおよびOS

- 対応PC：特に指定はありませんが、なるべく高いスペックのPCがよいでしょう。
- 対応OS：Vivadoが動作するWindows, またはLinuxが必要です。

本書の主な内容

基礎編

- ARMベースSoC Zynqとは何か
- PSでHello World, PLだけでLED点滅
- PS+PLのGPIOでLED点灯
- microSD, QSPIからの起動

ベア・メタル編

- IP Integratorで作る画像表示回路
- AXI回路の作成とIPパッケージの登録
- HDMI→VGA変換回路の作成
- 画像メモリ・インターフェースの作成
- HLSを使った画像処理回路の作成
- LEDマトリクス表示制御回路の作成
- Gbitイーサネット画像転送

Linux編

- ZYBOにLinuxを載せて使ってみる
– OSからFPGAのロジックを制御!
- ロジック×ARMで実現!堅牢で柔軟なデジタル・フィルタ
- ネットに繋がるFPGA!ZYBOで作る遠隔操作システム
- LinuxのGUIでロジックの動作検証
– ZYBOで作るジェネレータ&ロジアナ
- ドライバの知識がなくてもOK
– Linux on ZYBOで制御する加速度センサ
- OpenCVで画像処理を試す
- XilinxからLEDマトリクス表示制御
- ネットワーク経由でZYBOをリモート制御

見本