

FPGAで組み込みOSを活用する

— Nios と μ Clinix を実装したシステムLSIの実現

中根隆康, 福島雅史

ここでは、FPGAに実装したソフト・マクロのCPUを使ってLinuxを動作させる。CPUコアとして米国Altera社の「Nios II」、Linuxディストリビューションとしては「 μ Clinix」を利用する。FPGAボードでLinuxカーネルを動作させた後、CompactFlashから起動できるようにする。 (編集部)

最近、組み込みシステムの世界でもLinuxを活用する場面が増えています。Linuxの良いところは、オープン・ソースで誰でも自由に使えることと、コミュニティで作られているドライバなどを利用できることです。しかし、コーディング・ルールや試験方法などが明確に定められていないため、これらのソース・コードは、一貫性に欠けているように見えます。また、Linuxをビルドするためのmakefileも構造が複雑になり過ぎていて、初めてLinuxを導入しようとする方は少々戸惑うかもしれません(同じオブジェクトを何度もコンパイルするような構造のものもある)。

それでもLinuxが備えているネットワーク系のプロトコル・スタックやサーバ・アプリケーションを利用できるのは、かなりの魅力といえます。

1. ソフト・マクロのCPUと組み込みLinux

今回は、米国Altera社のソフト・マクロのCPU「Nios II」で、組み込みLinuxを動作させます^{注1}。

Nios IIは32ビットの命令セット、32ビットのデータ・バス、32個の外部割り込みソース、最大2Gバイトの外部

アドレス空間、最大256個のカスタム命令を持ったRISCプロセッサです。高速版(Nios II/f)、標準版(Nios II/s)、エコノミ版(Nios II/e)の3種類の構成を選択できます。今回は、Nios II/sを使用しました。

ソフトウェア開発は、EclipseとGNUのコンパイラをベースとした総合開発環境(IDE)で行います。

組み込みLinuxには、いろいろなディストリビューションがあります。しかしNios II用となると限られてしまいます。今回は一番実績がありそうな μ Clinixを搭載します。

● MMUを持たないマイコン向けのLinux

μ Clinixは、Embedded Linux / Microcontroller Project (<http://www.uclinux.org/>)によって、米国Motorola社(現在は米国Freescale Semiconductor社)のDragon Ball (MC68328)を搭載するPDA「PalmPilot」向けに開発されたものです。

Linuxは仮想メモリを使うことが前提になっているため、MMU(Memory Management Unit)を持たないプロセッサでは動作させられませんでした。この仮想メモリを物理ページ管理に変更して、MMUを持たないプロセッサでも動作できるようにしたものが μ Clinixです。そのため、残念ながらfork(プロセスのクローン生成)というUNIX系ではよく使われる機能が削除されています。

μ Clinixの構造を図1に示します。

注1: 米国Xilinx社のMicroBlazeを使った組み込みLinuxシステムには、例えば、アットマークテクノの「SUZAKU(朱雀)」がある。

Keyword

FPGA, ソフト・マクロ, CPUコア, Nios II, μ Clinix, カーネル, CompactFlash

● Nios II 対応の μ Clinux

Nios II に対応する μ Clinux は2種類あります。

- uclinux.org(<http://www.uclinux.org/>)からダウンロードできるフル・ソース・ディストリビューション (μ Clinux-dist-20070130.tar.gz)
- Nios Forum の μ Clinux Forum からダウンロードできる Nios IDE のプラグイン(nios2linux-1.4.zip)
Nios Forum のほうは、本稿執筆時には Nios IDE バージョン5.0をターゲットにしたものです^{注2}。このため、

アプリケーション				
システム・コール・インターフェース				
ターミナル	ファイル・システム	ネットワーク	IPC	プロセス管理
仮想メモリ管理	ページ管理	スケジューラ		
ハードウェア依存部				
ハードウェア(MMUなし)				

図1 μ Clinux システムの構造

灰色の部分が μ Clinux カーネル。

最新のバージョン7.1では利用できません。そこで今回は、uclinux.orgのものを利用します(p.58のコラム「 μ Clinux の入手と開発環境のセットアップ」を参照)。

uclinux.org の μ Clinux を実装する際には、開発環境に注意しなければなりません。ハードウェアの設計はWindowsで動作するQuartus IIを使えますが、ソフトウェアはLinux環境を使う必要があります(p.59のコラム「開発環境について」を参照)。Windows環境しかない場合は、coLinuxが必要⁽²⁾。

Windowsのファイル・システムがファイル名の大文字、小文字を区別しないためです。uclinux.orgの μ Clinux のディストリビューションの中には、同じ名前で大文字と小文字のファイルが存在しています。

注2 Microtronix Datacom社のホームページ(<http://www.microtronix.com/>)にバージョン7.0版が一度リリースされたが、すでになくなっている。理由は不明。

コラム

μ Clinuxの入手と開発環境のセットアップ

GNU 開発環境

μ ClinuxのビルドにはGNUの開発環境を使用します。通常ではNios II用にクロス開発環境を構築する必要がありますが、Nios Community Wikiにバイナリが用意されています(<http://nios.wiki.jot.com/WikiHome/OperatingSystems/BinaryToolchain> の nios2gcc.tar.bz2)。

μ Clinux ディストリビューション

今回使用するディストリビューションはuclinux.orgで公開されているものです(<http://www.uclinux.org/pub/uclinux/dist/> の uclinux-dist20070130.tar.gz)。

さらにNios Community WikiにあるNios II用のパッチ・ファイルを適用します(<http://nioswiki.jot.com/WikiHome/OperatingSystems/UclinuxDist> の uclinux-dist-20070130-nios2-02.diff.gz)。

```
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
[fukushima@linux ~]$ nios2-linux-uclibc-gcc -v
Reading specs from /opt/nios2/11b/gcc/nios2-linux-uclibc/3.4.6/specs
Configured with: /root/buildroot/toolchain_build_nios2/gcc-3.4.6/configure --pre
fix=/opt/nios2 --build=i386-pc-linux-gnu --host=i386-pc-linux-gnu --target=nios2
-linux-uclibc --enable-languages=c --enable-shared --disable-__cxa_atexit --enab
le-target-optspace --with-gnu-ld --disable-nls --enable-threads --disable-multil
ib --enable-cxx-flags=static
Thread model: posix
gcc version 3.4.6
[fukushima@linux ~]$
```

図A nios2-linux-uclibc-gcc バージョン情報

GNU ツールのセットアップ

次にダウンロードしたツールをインストールします。ツールを保存したディレクトリへ移動し、管理者権限でログインし、以下のコマンドを実行します(# はコンソールのプロンプト)。

```
# tar jxf nios2gcc.tar.bz2 -C/
ツールは/opt/nios2以下に展開されます。
```

次に環境変数の設定を行います。 .bash_profile に、
export PATH = \$PATH:/opt/nios2/bin
を追加します。また、よく使用する sdk_shell(Altera社が提供する開発ツールに付属する Nois のコマンド・シェル)も追加します。

```
export PATH = $PATH:/opt/Altera71/NiosIIEDS
```

環境変数の設定を適用させるためには、Linuxパソコンの再起動が必要です。確認のため、

```
# nios2-linux-uclibc-gcc -v
```

を実行します。図Aのようなメッセージが表示されれば、インストールは完了です。

ディストリビューションの解凍

カーネルの解凍とビルドは一般ユーザ権限で実行します。まず、 μ Clinux のディストリビューションを解凍するディレクトリを作成します。筆者は/home/nios2_uclinuxディレクトリを用意しました。 μ Clinuxディストリビューション・ファイルを nios2_uclinuxへ移動し、解凍します。

```
# tar zxvf uclinux-dist20070130.tar.gz
/home/nios2_uclinuxの下にuclinux-distディレクトリが作成され、その下にディストリビューション一式が解凍されます。
```