

Das U-Boot にみる CPU 起動シーケンス

川本 泰久

Das U-Boot は、最近では一般的に使われるようになったオープン・ソースの汎用的なブートローダである。本章では Das U-Boot を Blackfin に移植して動作させるために行った作業手順を追うことにより、CPU ボードの起動シーケンスを解説する。 (編集部)

Das U-Boot(以下 U-Boot)とは複数のアーキテクチャに対応したブートローダで、GPL でリリースされています。ここでは、実際に U-Boot を動作させるには何をしたらよいの

かを、CQ 出版社の「組み込みシステム開発評価キット(以下評価キット)」^{注1}やデバイスドライバズの「E!Kit-BF532」といった学習用組み込みボードを使って説明します。「E!Kit-BF532」は、Analog Devices 社製の Blackfin プロセッサ「ADSP-BF532」を搭載した、uClinux 2.6 系カーネルを移植済みの評価用ボードです(写真1,表1)。組み込みシステム開発評価キットと接続して使用できます。なお、Blackfin プロセッサの詳細仕様やデータブックなどは、同社の Web サイト(<http://www.analog.com/processors/japan/blackfin/>)から入手可能です。

開発環境とソース・ファイルを準備する

ここでは、図1のような機器構成を例に説明します。E!Kit-BF532 は評価キットに実装してあり、ホスト・パソコンとはシリアル・ケーブルと JTAG ケーブルで接続されています。

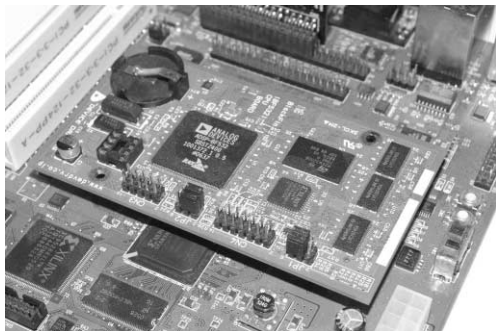


写真1 E!Kit-BF532 の外観(現在開発中)

「組み込みシステム開発評価キット」の上に実装したところ。本ボードは、組み込みシステム開発評価キットのオプション CPU カード用コネクタを備えている。

表1 E!Kit-BF532 の概略仕様(予定)

CPU	Analog Devices 社製 ADSP-BF532 400MHz 16K バイト 命令キャッシュ/ 32K バイト データ・キャッシュ
SDRAM	64M バイト, 16 ビット
フラッシュ ROM	4M バイト, 16 ビット
シリアル・インターフェイス	UART x 1
CPLD	Xilinx XC2C64A 44 ピン
JTAG コネクタ	14 ピン
拡張ソケット	120 ピン x 2 (CQ 出版社製「組み込みシステム開発評価キット」 に対応した独自拡張ボードを開発予定)
CPU 温度特性	動作時: 0 ~ 50 (周辺温度)
電源	+3.3V ± 5%, max 1.0A
大きさ	110mm x 60mm
対応 OS	uClinux 2.6 系カーネル(2.6.11 を移植, 動作確認済み) TOPPERS/JSP(μITRON4.0 仕様準拠)にも対応予定

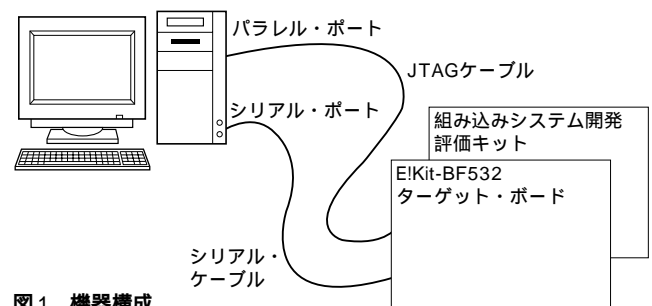


図1 機器構成

注1: 「組み込みシステム開発評価キット」は、FPGA や SDRAM, フラッシュ ROM, グラフィックス出力やオーディオ入出力などの A/V 機能, PCI, LAN, IDE, COM, USB, PS/2 などの I/O 機能を搭載した教育用ボードである。

第4章 Das U-Boot にみる CPU 起動シーケンス



```

$ su
Password

# cd /
# tar xzf blackfin-toolchain-07r1.1-3.i386.tar.gz
# exit

$ cd ~
$ vi .bash_profile
PATH=$PATH:$HOME/bin:/export/local/bin:/opt/uClinux/bfin-uClinux/bin/
    
```

この部分を追加する

```

$ cd ~
$ tar jxf u-boot-1.1.6.tar.bz2
    
```

図3 U-Boot ソース・ファイルの設定

図2 GNU Toolchain の設定

開発環境としては、Linux 上で動作する GNU Toolchain を使用することにします。GNU Toolchain は、Blackfin Linux Project (<http://blackfin.uclinux.org/>) から入手可能です。「GNU Toolchain(Releases, SVN, Snapshots, Help, Lists, Bugs)」の Releases 部分をクリックすることで該当ページにたどり着けます。ダウンロード後、図2のように解凍し、パスを設定すればインストールが完了します。

U-Boot のソース・ファイルは、DENX Software Engineering の Web サイト (<http://www.denx.de/wiki/UBoot/WebHome>) からオリジナルのものを入手するか、Blackfin Linux Project から入手します。筆者は Blackfin Linux Project で公開されているものを使用しています。「Das U-boot(Bootloader)(Releases, SVN, Help, Lists, Bugs)」の Releases 部分をクリックすることで該当ページにたどり着けます。ソース・ファイルを入手後、図3のように解凍すれば準備完了です。

開発環境の準備が整ったので、E!Kit-BF532 に U-Boot を実装します。今回は、搭載しているプロセッサ(ADSP-BF532)と構成が近い bf533-stamp のソースを基に実装することにします。

リセット直後の動作を理解しよう

ADSP-BF532 には幾つかのブート・モードがあります

表2 ADSP-BF532 のブート・モード

ブート・モード	解説
0	16 ビット・バス幅の外部メモリ (0x20000000) から実行
1	オンチップ・ブート ROM が実行され、8 ビット/16 ビット・バス幅の外部メモリからコードを読み込み実行
2	オンチップ・ブート ROM が実行され、SPI 経由でコードが書き込まれ実行
3	オンチップ・ブート ROM が実行され、SPI シリアル EEPROM からコードを読み込み実行

(表2)。例えばモード0では、リセット直後にリセット・イベントが発生して、0x20000000 から実行します。図4のメモリ・マップを見ると分かりますが、ここにはフラッシュ ROM が接続されており、U-Boot のコード領域がここに配置されています。

リンク・スクリプト(リスト1)を見ると、リセット直後に実行されるリセット・イベント・コード(リスト2)をコード領域の先頭に配置するように定義しています。リンク・スクリプトで定義されている先頭アドレスはメイン・メモリ上のアドレス(0x03FC0000)になっていますが、これは最終的に U-Boot はメイン・メモリ上で実行されるからです。Blackfin の JUMP/CALL 命令はすべて相対アドレスなので、このままフラッシュ ROM 上で実行可能です。

リセット・イベント・コード内では、まず各種 DAG (データ・アドレス発生器)、DATA、ADDRESS レジスタの初期化、リセット要因の確認、イベント・ベクトル・

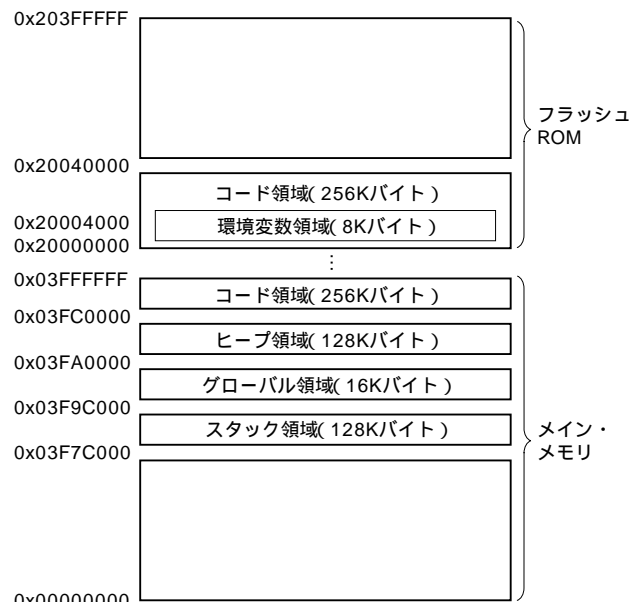


図4 U-Boot のメモリ・マップ