

XMOS の XS1-G4,L1 であそぶ - その 5

テスト基板 XC-1,XC-2,XK-1,XC-1A などであそんで気がついたことのメモを書いています

ツールチェーン 10.4 (2010/5)

問題点

・9.9.1 で xflash の内部エラーで焼けなかったソース (XK-1 を 2 台接続してお互いに通信) は正常に焼けるように見えるが、ブートしなかった。(xrun での実行は可能)

2 台の XK-1 を接続してスタンドアロンブートする方法 (10.4.2 ツールチェーン) (2011/1)

サポートから情報をいただきました。

- ・10.4.2 でも前のコードは正常にブートしません (xrun では動作する)。
- ・これは次期リリース (早ければ 2011/2 末) で修正される見込み。それまでは、2 台連結であれば、以下の手順でスタンドアロン動作可能とのことで、動作を確認しました。

実際には、2 つ FLASH を焼くのではなく、二台の XK-1 を [XK-1(A)],[XK-1(B)] とした場合、[XK-1(A)] 側を SPI ブート、[XK-1(B)] 側を XLINK ブートとする方法になります。

```
[XK-1(A)]--(XLINK)--[XK-1(B) boot from XLINK]
[SP|ROM]
```

まず、XLINK からブートさせるために [XK-1(B)] に以下のコードをビルドして、焼きます。これは、XK-1 の OTP->SPI ブート状態から XLINK ブート待ちにするためのコードです。

```
.text
.align 2
_start:
main:
mkmsk    r1, 16
shl     r1, r1, 16
ldc     r11, 0xc344
or      r11, r11, r1
set     dp, r11

ldc     r3, 0x80
ldw     r4, dp[0xb]
ldc     r11, 0xc0f4
or      r11, r11, r1
getr    r0, 2
bla     r11
freer   res[r0]

ldc     r7, 0xc076
or      r7, r1, r7
bau     r7
```

上記のコードをとしてセーブし、10.4.2 のコマンドプロンプトから、下記のようにビルドします。

```
xcc f2l.S -target=XK-1 -nostdlib -Xmapper --nochaninit -o f2l.xe
```

次に、[XK-1(B)] だけを接続し、xflash で f2l.xe を焼きます。

```
(Flash XLINK boot code to [XK-1(B)])  
xflash f2l.xe
```

これで、[XK-1(B)] は XLINK からのブート待ちになります。

次は [XK-1(A)]、[XK-1(B)] 両方のコアで動作させるプログラム (たとえば foo.xc とする) を、添付の を用いてビルドし、

```
(If main program is "foo.xc")  
xcc foo.xc 2XK-1.xn -o foo.xe
```

できたバイナリを、同様に [XK-1(A)] に焼きます。

```
(Flash binary foo.xe to [XK-1(A)])  
xflash foo.xe
```

肝は、2XK-1.xn の Source Location="XMOSLINK" の箇所 (下記) と思われます。

```
33: <Node Id="1" Type="XS1-L1A" Oscillator="20MHz" SystemFrequency="400MHz">  
34:   <Boot>  
35:     <Source Location="XMOSLINK"/>  
36:   </Boot>
```

次のリリースでは直る予定だそうです、XLINK からブートするコードは参考になるかと思いません。

11.2.0 ツールチェインリリース (2011/3)

サポートが言っていた通り、このバージョンでは2台の XK-1 を使った SPI ブートは正常に動作するようになりました。これでスタンドアロンでたくさんつないで (データシートによると Max16 コア) 実験ができます。

と思って3台つないでみたところ、2台でコケたときと同じような現象 (xrun では走るが SPI ブートしない) が出たので、現在問い合わせをしています ...orz

さて、L1 系のコアで SPI ブートするコアに接続された隣接コアで XMOSLINK からブートする (10.4.2 の workaround と同様の構成にする) 場合の .xn の記述が XMOS のサイトにある Tools-User-Guide の 8.2 節、Describing a Simple Hardware Platform に詳しく例として記載されています。

具体的には、SPI のあるほうの Node の Boot セクションで、XMOSLINK からのブート待ちにした

隣接コアを Bootee タグで指定するようになり

```
(e.g. if Slave Boot NodeId="1" Core="0")  
<Boot>  
  <Source Location="SPI:bootFlash"/>  
  <Bootee NodeId="1" Core="0"/>  
</Boot>
```

XMOSLINK で待つほうは 10.4.2 の workaround で行ったのと同様 (このページ前項を参照ください)

```
<Boot>  
  <Source Location="XMOSLINK"/>  
</Boot>
```

と記述すればよいようです。

XK-1 2 台での SPI-XMOSLINK Boot テスト (11.2.0)

10.4.2 での前々項の workaround をした XK-1 2 台接続プロジェクトに、前項後半で説明した Bootee 記述を追加して IDE から焼いてみましたが、残念ながらきちんとブートしませんでした。

そこで、10.4.2 と同様に、を、Bootee 側の XK-1 に下記のようにコマンドラインから焼いてやると、下記のようなワーニングが出るものの、11.2.0 でも同様にブートしました。

```
C:\work\%xk1%\xk1dual2\Release>xcc f2l.S -target=XK-1 -nostdlib -Xmapper --nochaninit -o f2l.xe  
C:\work\%xk1%\xk1dual2\Release>xflash f2l.xe  
Warning: F03098 Factory image and boot loader cannot be write-protected on flash device on node "0".  
Site 0 has finished.
```

XK-1 の OTP にかかっている出荷時ファームウェアが邪魔しているのかもしれませんが。うーむ。

XK-1 複数台での SPI Boot 問題修正 (11.2.1)

Release Note より。

```
* Flash programming  
* The 11.2 version of xflash could not produce a flash image to  
boot a system with more than 2 switches. This is now fixed.
```

10.4 から引きずっていたこの問題が、ようやく修正されたようです (実際に 4 台までは、つないで確認しました)。

これで安心してたくさん買ってつなげます。

OTP の中をのぞいてみる (11.2.2)

知り合いに OTP の中がどうなっているか聞かれたが、いままで気にしたことがなかったので、

のぞき方を知りませんでした。教えてもらったのが xburn を使う方法。

XK-1 (via XTAG2) の場合

```
C:\Program Files\Xilinx\DevelopmentTools\11.2.2>xburn --read --target XK-1
Reading device...
Core 0:
0x00000000: 0x00000018
0x00000001: 0xdbfef007
...
SR      : 0x00000020
```

のような感じですか。すでに書かれていますね。

XC-1A (via 内蔵 JTAG) の場合

```
C:\Program Files\Xilinx\DevelopmentTools\11.2.2>xburn --read --target XC-1A
Reading device...
Core 0:
SR      : 0x00000000
Core 1:
SR      : 0x00000000
Core 2:
SR      : 0x00000000
Core 3:
SR      : 0x00000000
```

どうやら空のようです。

XS1-L1-64 (SparkFun 製 旧ロット :FTDI つき via XTAG2) の場合。
(XC-5 と同一チップなので、XC-5 を指定しています)

```
C:\Program Files\Xilinx\DevelopmentTools\11.2.2>xburn --read --target XC-5
Reading device...
Core 0:
*
0x0000020e: 0x00800000
SR      : 0x00000000
```

SR = 0x00000000 ですが、空ではないようです。

XC-1 (via 内蔵 JTAG) の場合、普通にブートしてしまうとエラーになりました。

```
C:\Program Files\Xilinx\DevelopmentTools\11.2.2>xburn --read --target XC-1
Reading device...
xburn: error: reading device failed
```

SPI/JTAG の切り替え (選択 SW を取り付け) を SPI ブートにしてやると、

```
C:\Program Files\Xilinx\DevelopmentTools\11.2.2>xburn --read --target XC-1
Reading device...
Core 0:
0x00000000: 0x000007dd
0x00000001: 0xfed1684b
0x00000002: 0xf00417ec
0x00000003: 0xfec2688b
...
0x000007dd: 0x00000000
0x000007de: 0x0d15ab1e
SR      : 0x00000020
```

Core 1:
SR : 0x00000000
Core 2:
SR : 0x00000000
Core 3:
SR : 0x00000000

読めました。

< < XMOS の XS1-G4,L1 であそぶ - その 4 | XMOS の XS1-G4,L1 であそぶ - その 6 > >