

Keystone 1 SPI NOR 启动步骤以及注意事项

本文例程和文件可在

http://www.devisupport.com/question_answer/dsp_arm/c6000_multicore/f/53/t/100597.aspx 下载

1. DSP 启动的 Rom Bootloader (RBL) 只接受 Boot table 文件格式，编译生成的 elf 或者 coff (.out) 文件需要转换成 boot table 格式才能使用，转换工具为 hex6x，所在目录一般为 C:\ti\ccsv6\tools\compiler\c6000_7.x.xx\bin
rmd 文件是 hex6x 的输入，格式一般为

```
simple.out
-a
-boot
-e _c_int00

ROMS
{
    ROM1: org = 0x0C000000, length = 0x100000, memwidth = 32, romwidth = 32
    files = { simple.btbl }
}
```

一般只需更改输入 out 和输出的 btbl 文件的名字和路径，支持绝对路径和相对路径，length 需根据用户使用的 memory 的大小进行相应的调整，只要长度大于用户使用的空间即可。起始地址 0x0C000000 无需更改为用户地址，不影响结果。

```
C:\Users\0282268\Documents\TI\Code\Keystone I\Boot\K1\SPI_Bootloader>hex6x.exe
simple.rmd
Translating to ASCII-Hex format...
"simple.out" ==> .text      (BOOT LOAD)
"simple.out" ==> .cinit     (BOOT LOAD)
"simple.out" ==> .const     (BOOT LOAD)
"simple.out" ==> .switch    (BOOT LOAD)
```

多核启动代码应分别使用 hex6x 生成对应的 btbl 文件格式，然后使用 mergebtbl 进行合并。

```
mergebtbl core0.btbl core1.btbl multi_core.btbl
```

2. 使用 b2i2c 工具将第 1 步生成的 btbl 文件转换成 i2c/spi 启动的格式，b2i2c 工具和源代码可以在 C:\ti\mcsdk_2_01_02_06\tools\boot_loader\ibl\src\util\btoccs 里找到。

```
C:\Users\0282268\Documents\TI\Code\Keystone I\Boot\K1\SPI_Bootloader>b2i2c simple.btbl simple.btbl.i2c
```

3. 使用 b2ccs 工具将第 2 步生成的 i2c 文件转换成 CCS 可识别的 ccs 文件

```
C:\Users\0282268\Documents\TI\Code\Keystone I\Boot\K1\SPI_Bootloader>b2ccs simple.btbl.i2c simple.i2c.ccs
```

4. 使用 Romparse 工具将第 3 步生成的 ccs 文件和 spi 启动的参数合并生成 i2crom.ccs，romparse 的源代码 C:\ti\mcsdk_2_01_02_06\tools\boot_loader\ibl\src\util\romparse，此处注意按照自己的 SPI NOR 的模式修改 spi.map 文件，比如片选信号。

```
C:\Users\...\Documents\TI\Code\Keystone I\Boot\K1\SPI_Bootloader>romparse n  
ysh.spi.map
```

在 i2crom.ccs 里修改如下图对应位置的 51 为 00

2 0x00500000	0x00500000
3 0x00320000	0x00320000
4 0x40200002	0x40200002
5 0x00010018	0x00010018
6 0x00040000	0x00040000
7 0x00000000	0x00000000
8 0x03200000	0x03200000
9 0x01f40051	0x01f40000
10 0x04000000	0x04000000
11 0x00000000	0x00000000

5. 使用 byteswapccs 工具将 i2crom.ccs 转换成大端，RBL 只工作在大端模式下，因此，如果需运行在小端模式下，编译生成的小端格式的文件需进行转换。如果编译时使用了大端模式配置并将运行在大端的 DSP 下，可以直接将 i2crom.ccs 更名成第 6 步需要的文件名。

```
C:\Users\...\Documents\TI\Code\Keystone I\Boot\K1\SPI_Bootloader>byteswapccs i2crom.ccs spirom_le.dat
```

6. 使用 SPI NOR writer 在 CCS 下将文件写入 NOR flash，SPI NOR writer 的文件在 mcsdk_2_01_02_06\tools\writer\nor\evmc66xx 目录下，需要注意该烧写器代码仅对应 EVM 板，用户自己使用的 NOR flash 参数和 DSP 的参数可能与 EVM 不一致，需要修改重新编译才能使用，不同的 EVM 板对应的 platform 源文件在 C:\ti\pdk_C66xx_1_1_2_6\packages\ti\platform\evmc6657\platform_lib\src 目录下。

由于 CCS 文件格式有一些变化，mcsdk_2_01_02_06\tools\writer\nor\evmc66xx 下的 writer 需要做一些修改后才能使用，下载修改后的 norwriter.c 覆盖原工程的文件重新编译工具。

然后更改 nor_writer_input.txt 内容如下，可以根据自己的文件名进行修改，注意需要用.dat 文件格式；修改完成以后在 no boot 的启动模式下启动 DSP，运行 nor_writer 读取 dat 文件并烧写到 NOR flash，务必注意此时需加载 gel 文件初始化 DDR3，因为用户 dat 文件会被读取到 DDR3 缓存中。

```
file_name = spirom_le.dat  
start_addr = 0
```

7. 烧写完成以后，将 boot 模式置于 SPI NOR flash 启动，如不成功，可考虑按下述方式检查
 - 确定 DSP 是否正常启动，测量 SYSCLKOUT 管脚是否稳定输出时钟信号，连接 JTAG 读取 DEVSTAT 确定 boot mode 设置是否正确；

- b. 连上 JTAG 后可检查 PC 指针的值看是否仍停留在 RBL (0x20B0000) 还是已经进行用户代码
- c. 测量 SPI 线上时钟是否稳定;
- d. 使用 JTAG 读取 SPI ROM 内容并与写入的数据比较检查是否正确;
- e. 如果修改了 spi.map 的时钟，修改回低速率时钟再做测试。
- f. 多核启动每个核的 cmd 文件务必使用全局地址 0x1x8xxxxx 而不是本地地址 0x008xxxxx，因为 ROM 的读取和写入都由 core 0 完成，使用本地地址无法找到对应的写入地址。
- g. DDR3 必须初始化之后才可使用，如果使用 DDR 的内存作为 boot 段需考虑使用二次 Boot，二次 boot 可以参考 C:\ti\mcSDK_2_01_02_06\tools\boot_loader\examples\srio\srioboot_ddrinit 的代码。
- h. 如仍有问题，请到 deyisupport 多核论坛提出，请将上述步骤中测试的结果都描述清楚，有助于快速定位问题。

文中实例可在附件 SPI_Bootloader.7z 查看。