上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Boot的原理与实例

– DSP培训课件之八

上海交大-TI 联合DSP实验室

主要内容

- boot的原理概述
- 实例分析
- Flashburn工具的使用
- Hex转换工具的使用

TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP ，MSP430等技术培训与技术服务，项目合作等。培训内容有

1）CCS开发环境精解与实例；

2）DSP/SYS BIOS 实例；

3）C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；

4）HPI，EMIF，EDMA，Timer等外设；

5）C6416、DM642， C6678多核EVM开发平台实例；

6）Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），颁发TI授权证书

邮件报名：jhniu@sjtu.edu.cn ，niujinhai@yahoo.com.cn

颁发TI授权的培训证书

引言

如果某电脑中安装有多个操作系统，在电脑加电自检后会出现一个启动菜单，它列出了在这台电脑上安装的所有的操作系统，用户使用“↑”“↓”键可以选择进入哪个系统。

实际上这一切都源于一个名为BOOT.INI的文件，起一个引导作用。

同样在嵌入式操作系统中，也需要一个引导(boot)程序来，做前期的引导工作。

Boot 原理

在DSP开发中，所谓的Boot就是指系统的启动模式。即如何将程序加载到内部memory中去。

C6416中有三种启动模式（boot mode）：

1. 无模式启动(no boot)

2. 外部存储器模式（EMIF boot）

3. 主机启动模式（ Host boot)

bootmode

The C6414T/15T/16T device resets using the active-low signal $\overline{\text{RESET}}$. While $\overline{\text{RESET}}$ is low, the device is held in reset and is initialized to the prescribed reset state. Refer to reset timing for reset timing characteristics and states of device pins during reset. The release of $\overline{\text{RESET}}$ starts the processor running with the prescribed device configuration and boot mode.

The C6414T/C6415T/C6416T has three types of boot modes:

- Host boot

  If host boot is selected, upon release of $\overline{\text{RESET}}$, the CPU is internally "stalled" while the remainder of the device is released. During this period, an external host can initialize the CPU's memory space as necessary through the host interface, including internal configuration registers, such as those that control the EMIF or other peripherals. For the C6414T device, the HPI peripheral is used for host boot. For the C6415T/C6416T device, the HPI peripheral is used for host boot if PCI_EN = 0, and the PCI peripheral is used for host boot if PCI_EN = 1. Once the host is finished with all necessary initialization, it must set the DSPINT bit in the HPIC register to complete the boot process. This transition causes the boot configuration logic to bring the CPU out of the "stalled" state. The CPU then begins execution from address 0. The DSPINT condition is not latched by the CPU, because it occurs while the CPU is still internally "stalled". Also, DSPINT brings the CPU out of the "stalled" state only if the host boot process is selected. All memory may be written to and read by the host. This allows for the host to verify what it sends to the DSP if required. After the CPU is out of the "stalled" state, the CPU needs to clear the DSPINT, otherwise, no more DSPINTs can be received.

- EMIF boot (using default ROM timings)

  Upon the release of $\overline{\text{RESET}}$, the 1K-Byte ROM code located in the beginning of $\overline{\text{CE1}}$ is copied to address 0 by the EDMA using the default ROM timings, while the CPU is internally "stalled". The data should be stored in the endian format that the system is using. In this case, the EMIF automatically assembles consecutive 8-bit bytes to form the 32-bit instruction words to be copied. The transfer is automatically done by the EDMA as a single-frame block transfer from the ROM to address 0. After completion of the block transfer, the CPU is released from the "stalled" state and starts running from address 0.

- No boot

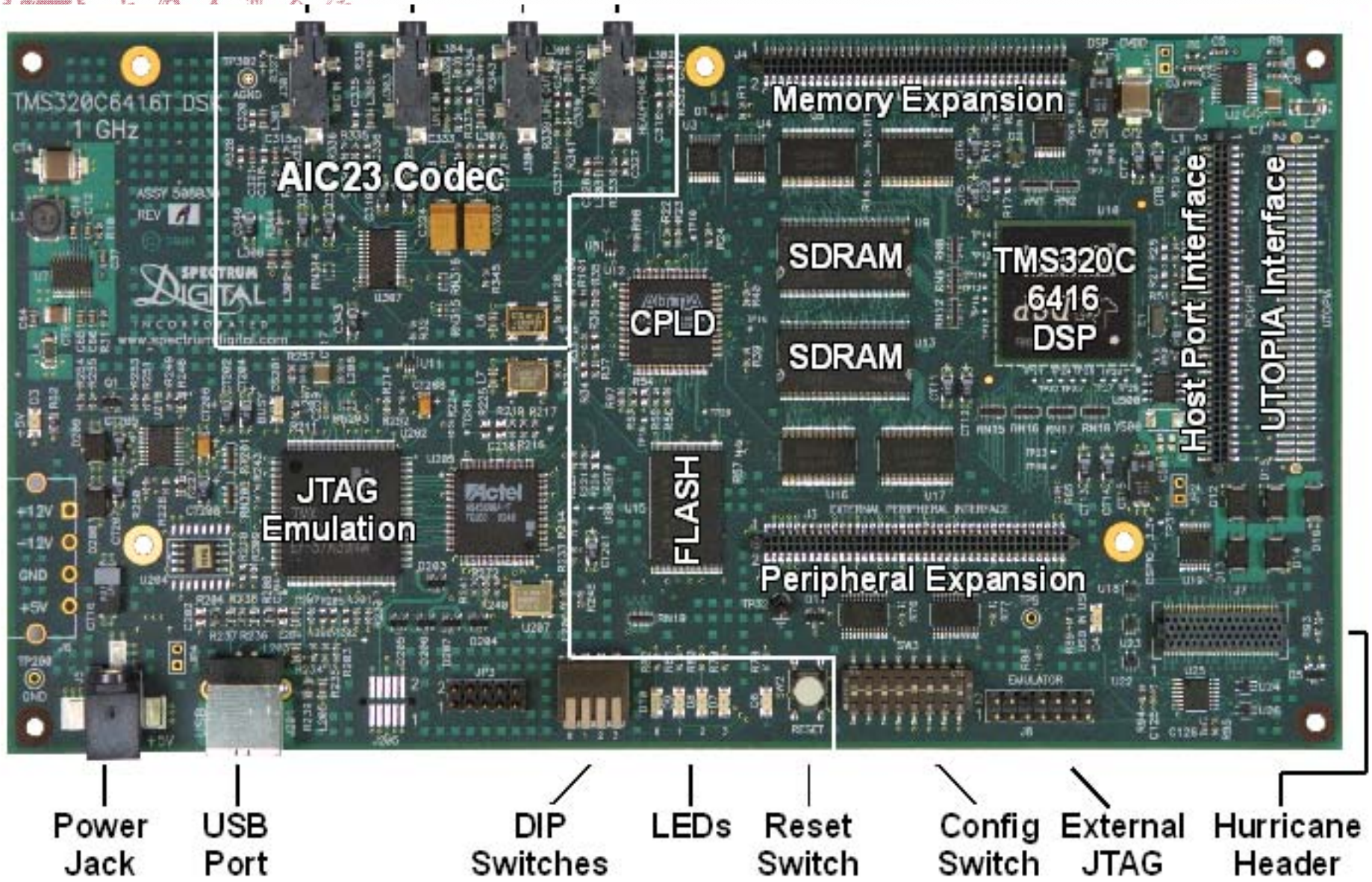
  With no boot, the CPU begins direct execution from the memory located at address 0. Note: operation is undefined if invalid code is located at address 0.

Boot 原理

系统到底采样何种启动模式，取决去与底层硬件的状态，主要是8个结构控制开关中2号、3号的状态，具体如下：

Configuration Switch Settings

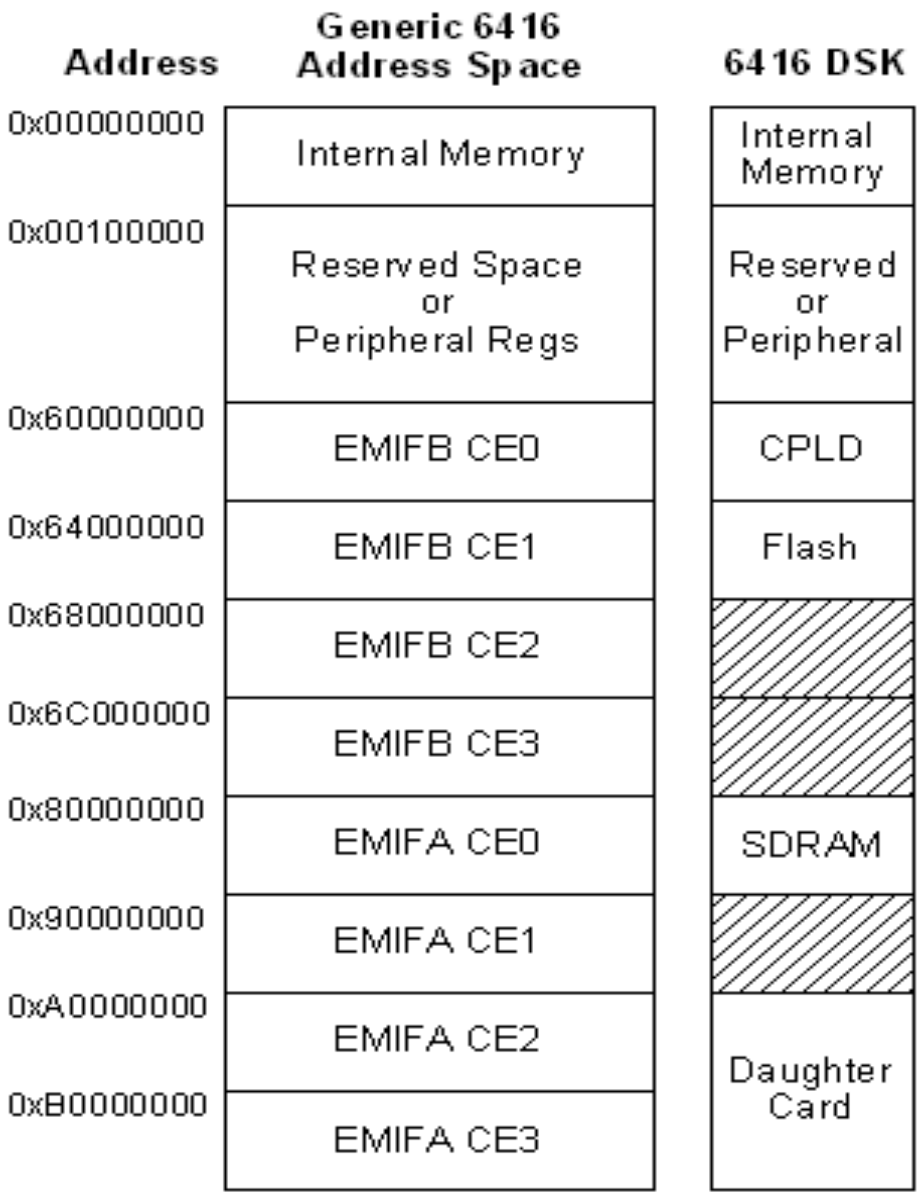
Switch 3	Switch 2	Switch 1	Configuration Description
Off	Off		EMIFB boot from on-board 8-bit Flash
Off	On		No boot
On	Off		Reserved
On	On		HPI boot
		Off	Little endian
		On	Big endian


TMS320C6416T DSK
1 GHz
AIC23 Codec
Memory Expansion
SDRAM
SDRAM
TMS320C
6416
DSP
CPLD
FLASH
JTAG
Emulation
Host Port Interface
UTOPIA Interface
Peripheral Expansion
+12V
-12V
GND
+5V
Power
Jack
USB
Port
DIP
Switches
LEDs
Reset
Switch
Config
Switch
External
JTAG
Hurricane
Header

Boot 原理

No boot

这时CPU直接从地址为0处开始执行。若0处的代码无法执行则，启动失败，指令操作无法进行。


Address
Generic 6416 Address Space
6416 DSK
0x00000000
Internal Memory
Internal Memory
0x00100000
Reserved Space or Peripheral Regs
Reserved or Peripheral
0x60000000
EMIFB CE0
CPLD
0x64000000
EMIFB CE1
Flash
0x68000000
EMIFB CE2
0x6C000000
EMIFB CE3
0x80000000
EMIFA CE0
SDRAM
0x90000000
EMIFA CE1
0xA0000000
EMIFA CE2
Daughter Card
0xB0000000
EMIFA CE3

Boot 原理

Host boot

在这种模式下，当按下 $\overline{RESET}$ 键并释放后,或者重新上电后，CPU处于等待状态。此时，外部的主机（ARM,PC等）就可以对6416DSK经行操作。如，读写memory中的内容、对内部的寄存器进行设置等。要结束boot过程，只要将HPCI寄存器中的DSPINT置位。这样CPU就可以从地址为0处执行指令。

这种模式一般用于系统调试阶段。

注意事项：

1 只有在host boot模式下，DSPINT才对CPU的状态起作用

2 DSPINT并不是一直不变的，只有在CPU处于等待状态时，才能锁定DSPINT状态。

3 CPU解除等待状态后，必须将DSPINT的内容清空，否则后续对DSPINT位的操作无法经行。

Boot 原理

EMIF boot

在这种情况下，当按下 $\overline{RESET}$ 键并释放后，或者重新上电后，在EDMA的控制下，系统自动从地址为0x64000000处(即flash的起始处)，读入1KB的内容到地址为0的处。此时CPU处于等待状态(be stalled)，当搬运完成后，CPU从地址为0处开始执行。

这种模式一般应用于嵌入式系统。

实例分析

在嵌入式系统中，应用程序往往放在，外部的ROM中，就6416而言就是放在flash(512KB)中。

而实际的应用程序经常大于1KB，这样仅仅采用上述的EMIF boot显然是不够的。

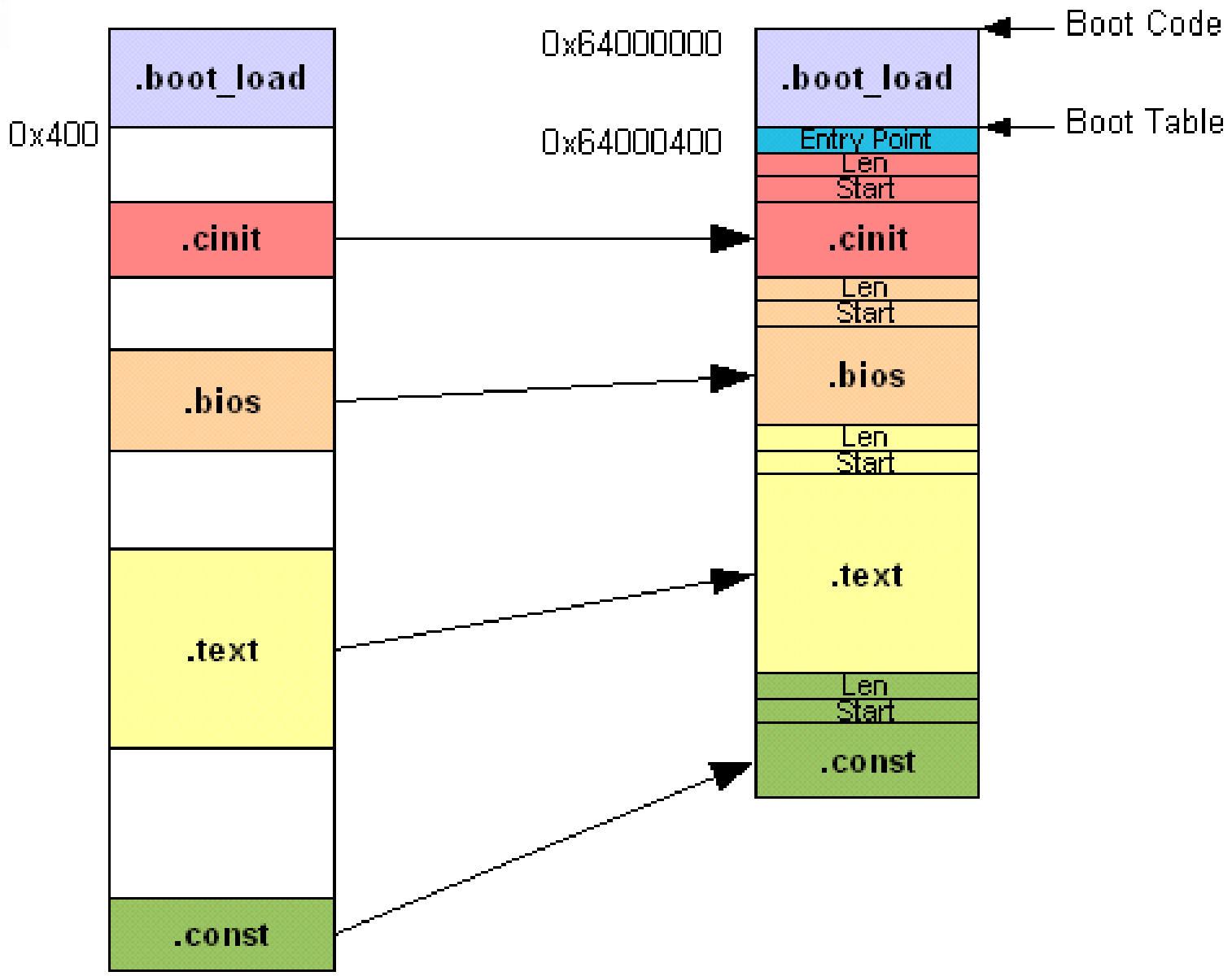
解决的办法是：在flash起始的1KB中存储一段特殊的代码，一般称为boot loader。在之后存储真正的应用程序代码，不妨称为user_code。

实例分析

当EDMA将这1KB的内容搬到内部高速RAM处时，CPU随即执行这段boot_loader代码。

其结果是将flash中后续的全部user_code搬入内部RAM中。

然后，程序的指针跳转到user_code的入口处，即可执行应用程序的代码了。


0x400
.boot_load
.cinit
.bios
.text
.const
0x64000000
0x64000400
.boot_load
Entry Point
Len
Start
.cinit
Len
Start
.bios
Len
Start
.text
Len
Start
.const
Boot Code
Boot Table

实例分析

Boot_loader 代码：

```
    .ref    _c_int00
    .global RESET_RST

FLASH_START   .equ    0x64000400        ;flash start address
CODE_START    .equ    0x00000400        ;start of non boot code
CODE_SIZE     .equ    0x00003000        ;application code size in
byte

      .sect "bootload"

_boot_start:
      mvkl  FLASH_START,B4 ;flash start address ->B4
      mvkh  FLASH_START,B4
```

实例分析

```
mvkl  CODE_START,A4 ;apps code start address ->A4
    mvkh  CODE_START,A4
    zero  A1
_boot_loop1:
    ldb   *B4++,B5        ; flash read
    mvkl  CODE_SIZE-4,B6 ; B6 = BOOT_SIZE -1024

    add   1,A1,A1           ;A1+=1,inc outer counter
    ||    mvkh  CODE_SIZE-4,B6

    cmplt  A1,B6,B0
    nop
    stb   B5,*A4++
    [B0]  b     _boot_loop1
    nop   5
```

实例分析

```
mvkl .S2 _c_int00, B0
mvkh .S2 _c_int00, B0
B    .S2 B0
nop   5
```

FlashBurn的使用

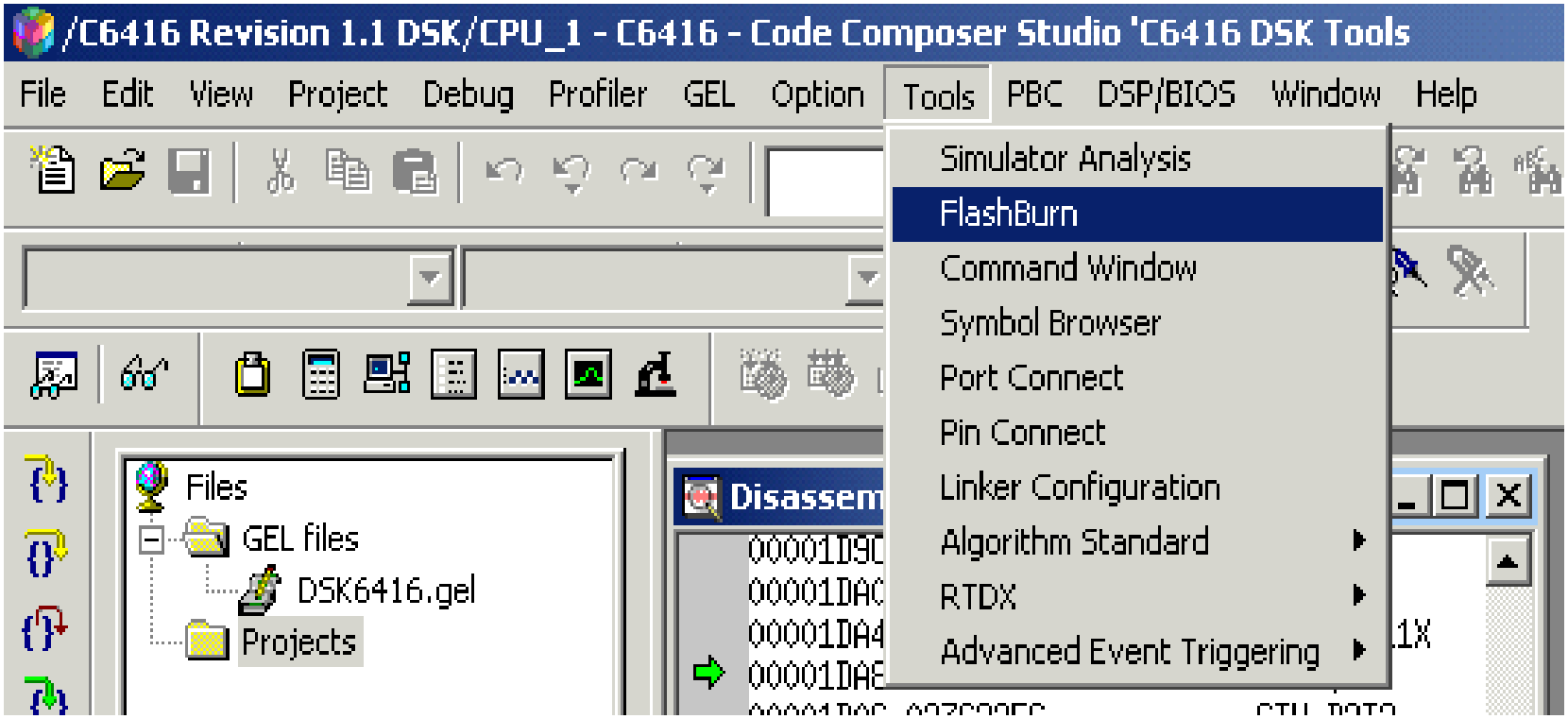
FlashBurn是CCS自带的的工具，用来将用户编写的代码烧写到开发板上flash中，真正帮助开发真实现EMIF boot模式。

使用方法如下：

FlashBurn的使用

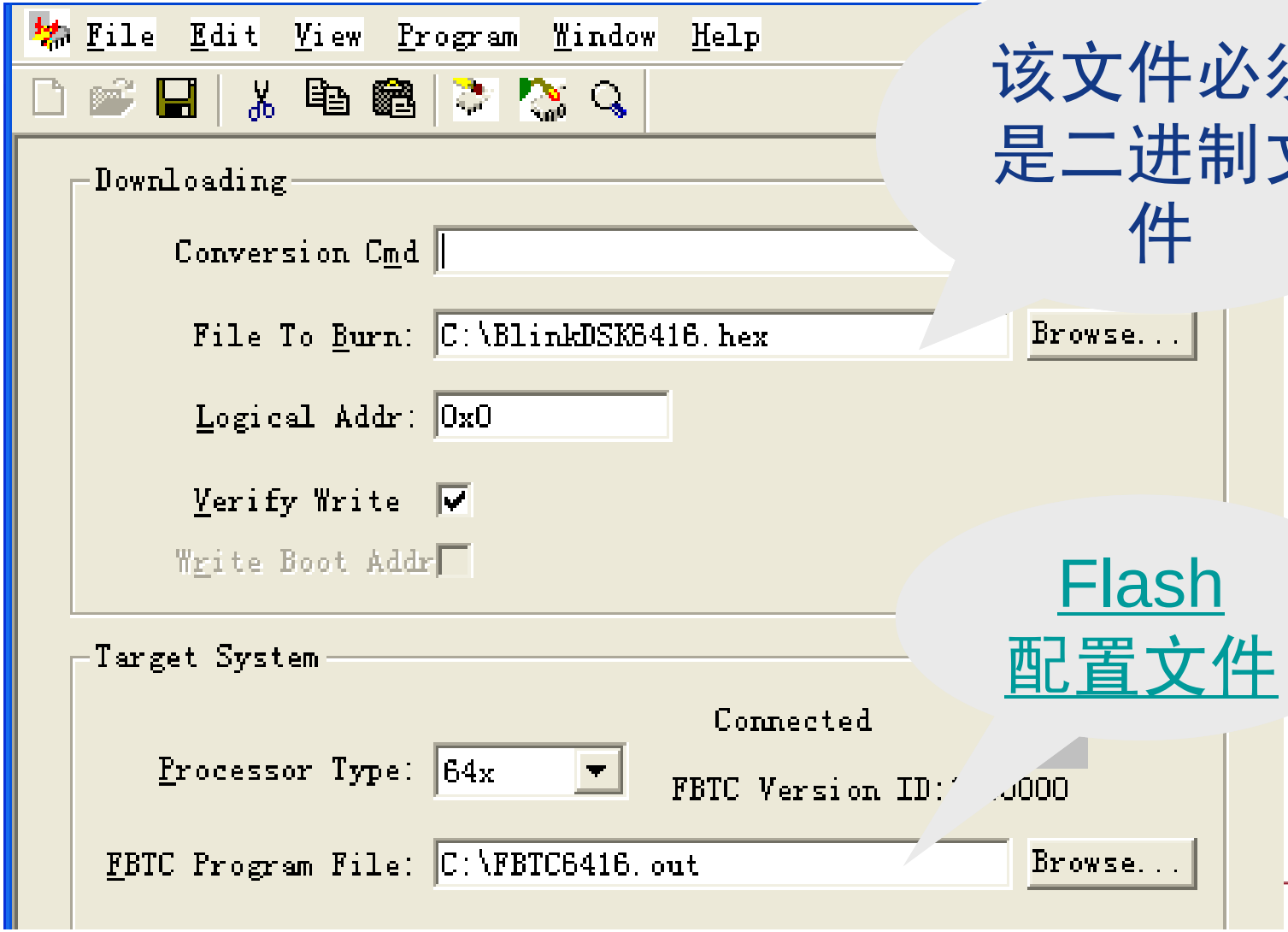
1 先打开CCS,然后点击Tools-->FlashBurn 。或直接双击

• • • \bin\utilities\flashburn\FlashBurn.exe



FlashBurn的使用

2 打开FlashBurn之后，点击File->Open,找到需要烧写的文件(.cdd)。
或者新建一个新的.cdd文件，自己填入.hex文件和.out文件



FlashBurn的使用

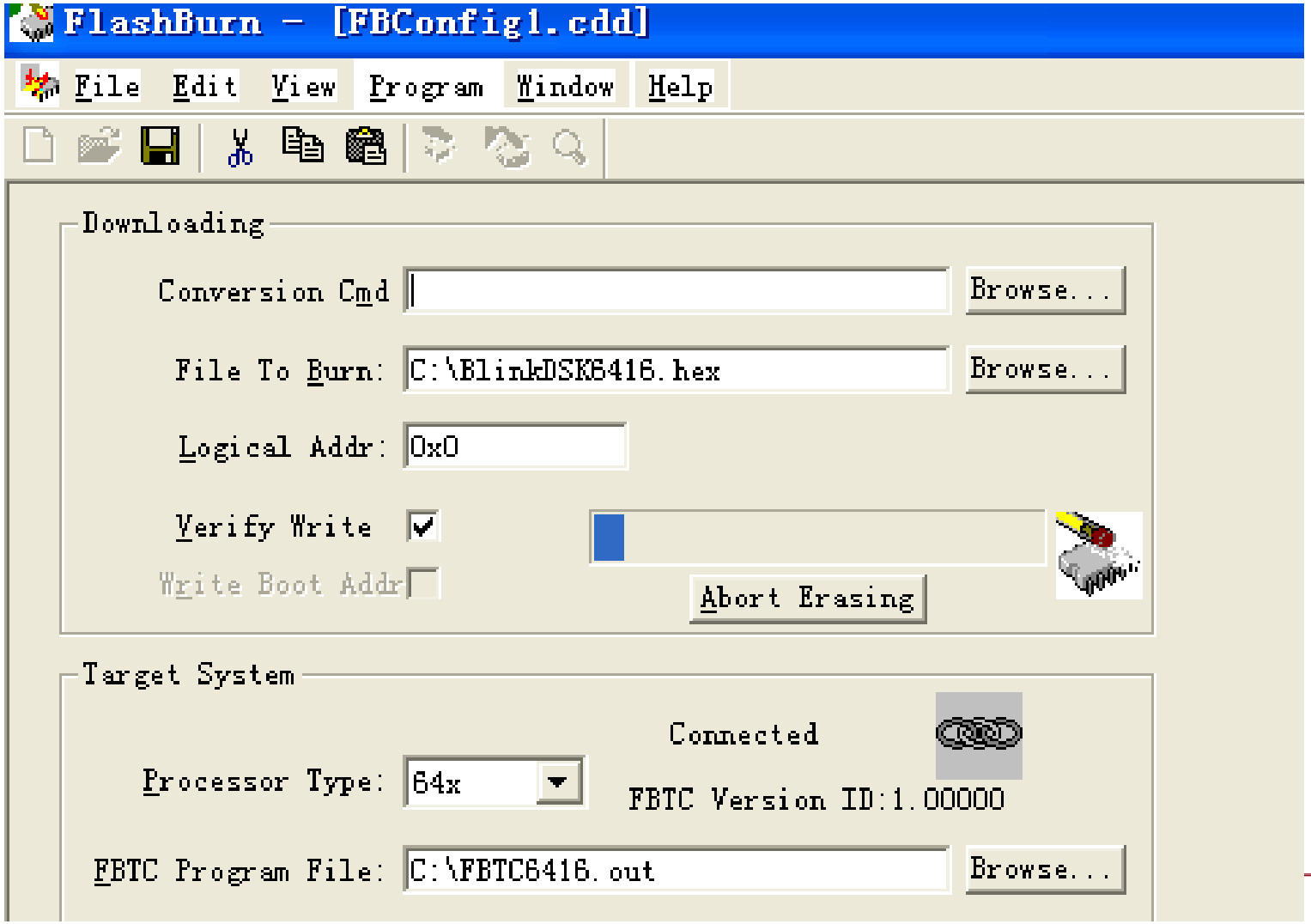
FBTC 项目中的 .out 文件是用来配置底层flash的。FlashBurn就是用这个文件操纵底层flash，将 .hex文件烧写到flash中。

一般位于

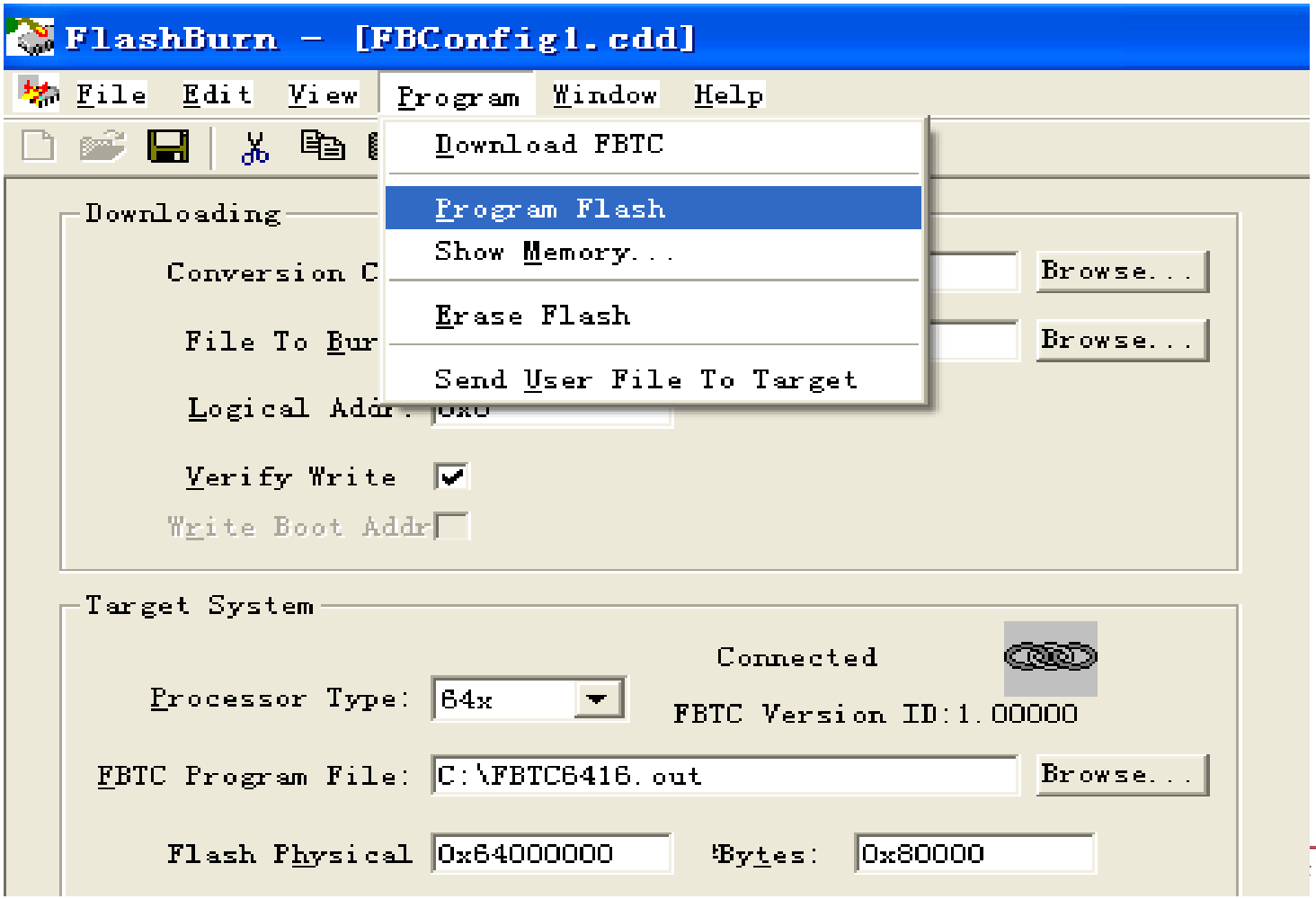
• • •\bin\utilities\flashburn\c6000\dsk6416v3\FBTC6416 \FBTC6416.out

FlashBurn的使用

3 点击Program->Erase Flash，擦除掉flash中的内容。



4 最后，点击Program->Program Flash将程序烧写到flash中。



Hex转换工具

位于・・・\C6000\cgtools\bin\hex6x.exe

用来将.out装换成能被烧写到flash中的.hex

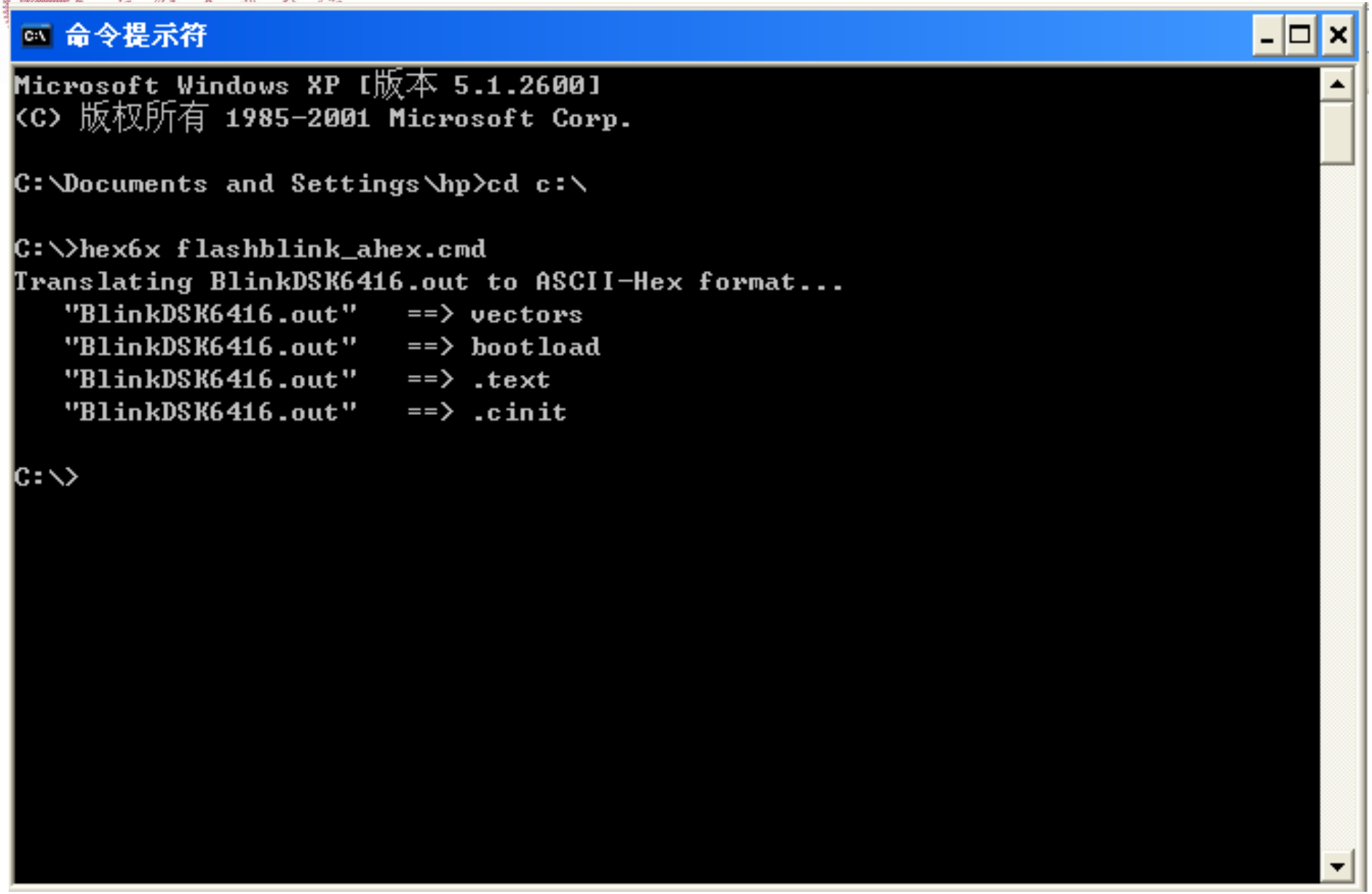
二进制文件。

使用方法是：

将hex6x.exe ， conversion.cmd ，.out文件放在一个目录下。然后在dos命令行中将盘符路径定位到该目录。接着输入：

hex6x conversion.cmd

就可以得到.hex 文件了。


命令提示符
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.
C:\Documents and Settings\hp>cd c:\
C:\>hex6x flashblink_ahex.cmd
Translating BlinkDSK6416.out to ASCII-Hex format...
"BlinkDSK6416.out" ==> vectors
"BlinkDSK6416.out" ==> bootload
"BlinkDSK6416.out" ==> .text
"BlinkDSK6416.out" ==> .cinit
C:\>

Hex转换工具

其中.cmd 文件的内容实际上是.hex6x 的输入参数，在上例中其内容如下：

```
BlinkDSK6416.out  /* Input File: COFF file format (.out) */
-a                /* Output Format: ASCII Hex format (.hex) */
-image                          /* Select image mode */
-memwidth 8                     /* Set memory width */
-o BlinkDSK6416.hex /* Output File:ASCII Hex file (.hex) */

ROMS
{
    FLASH: org = 000h, len = 0x3000, romwidth = 8
}
```

参考文献

- \docs\hlp\c6416dsk.hlp(有关dsk6416介绍)
- \docs\pdf\spru642.pdf(关于boot mode)
- \docs\hlp\spru189.pdf(c600 CPU和指令集)

  sprs226M(关于6416CPU)

TI DSP培训以及技术服务简介

上海交大BME-美国德州仪器联合DSP实验室成立于2007年，是国内最权威的TI技术服务于培训机构。实验室有TI（C6000，C2000，C5000，达芬奇，多核DSP）全系列开发平台，提供DSP ，MSP430等技术培训与技术服务，项目合作等。培训内容有

1）CCS开发环境精解与实例；

2）DSP/SYS BIOS 实例；

3）C6000/C5000/C2000全系列DSP架构以及汇编，C语言，混合编程等；

4）HPI，EMIF，EDMA，Timer等外设；

5）C6416、DM642， C6678多核EVM开发平台实例；

6）Bootloader 原理以及实例等。

常年开班，三人以上集体报名8折优惠，学生5折。

联系电话：13651621236（牛老师），颁发TI授权证书

邮件报名：jhniu@sjtu.edu.cn ，niujinhai@yahoo.com.cn

颁发TI授权的培训证书

DSP实验室介绍

- 美国德州仪器（TI）—上海交通大学（SJTU）联合DSP实验室成立于2007年10月，位于上海交大闵行校区，致力于TI DSP技术的推广，以及相关数字信号处理算法的研究与开发，为客户提供优质的产品与服务，涉及的技术领域有，无线通信，音频/视频信号处理，医学信号/图像处理，数字马达控制等。实验室研发与培训教师主要由上海交通大学青年教师承担，同时聘请了多位有企业工作背景的DSP技术专家为实验室的顾问。

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

DSP实验室介绍

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

End

Thanks