Potential Problem in SPRC074 Relating to KERNEL
Relating to the TMS320LF2406A

by Eric Ribble, AMT Machine Systems
ribble@amtsys.com
June 29, 2010

I was experiencing some problems trying to program the TMS320LF2406A flash memory via the SCI serial port. I subsequently dug deep into device's BOOT ROM source code and the TI-supplied SCI flash programming utilities.

I may have found a potential problem.

From TI web site, I downloaded the "LF240xA Serial Flash Programming Utility" (part number SPRC074) from the link http://focus.ti.com/docs/toolsw/folders/print/sprc074.html . One of the files in this SPRC074 is a file named "krnl24xx.hex" which is the kernel for the SCI flash utility.

Here are the first 8 words of the file "krnl24xx.hex", which I have annotated with comments:

```
0000    ; entry address
007a    ; length
8000    ; memory address where code is to be loaded
0000    ; unused word
bce0    ; first word of kernel code to be loaded at 0x8000
ae29    ; second word of kernel code to be loaded at 0x8001
006f    ; third word of kernel code to be loaded at 0x8002
bc00    ; fourth word of kernel code to be loaded at 0x8003
...
```

I think that the first word in the above file is erroneous. I think it should be 8000 instead of 0000 for the following reasons.

1. The kernel code is supposed to be loaded and executed beginning at address 0x8000. Thus the entry address (the first word in the file) should be 8000.

2. When the TMS320LF2406A BOOT ROM receives (via the SCI serial port) the above "krnl24xx.hex" file, it will properly load the kernel code at address 0x8000. However, after the BOOT ROM has finished loading the kernel code, the BOOT ROM will branch to the ENTRY address of 0x0000 (as indicated in the file above) which is incorrect. It should branch to 0x8000.

On the following page is a cut-and-paste of the appropriate sections of the BOOT ROM source code obtained from Appendix D (page D-18) of the document SPRU357C (TMS320LF/LC240xA DSP Controllers Reference Guide).

If you study the attached BOOT ROM code (beginning with the label MAIN), you can see that it is going to branch to the destination address (DEST) which will be equal to 0x0000 for the kernl24xx.hex code above, which is incorrect.

At the label M00, the call to FETCH_HEADER will fetch the first two words of krnl24xx.hex and return with DEST equal to 0x0000 and LENGTH equal to 0x007a.

Then at the label M01, the call to XFER_SCI_2_PROG will not alter DEST. Then the code branches to DEST.

In my opinion, the kern24xx.hex file should be built so that the entry address (the first word in the file) is 8000 instead of 0000.

(see BOOT ROM source code on following page …)

```
        B     UI01                  ;No! fetch another char

INC_TBL_PTR    LACC   BAUD_TBL_PTR  ;Inc CRC counter
        ADD     #1h
        AND     #0007H               ; BAUD_TBL_PTR is MOD(8)
        SACL    BAUD_TBL_PTR
        SPLK    #0h, CHAR_RETRY_CNTR
        B       UI00
BAUD_DETECTED:
;***************************************************************************
;M A I N   P R O G R A M
;***************************************************************************
MAIN:
                                    ;Load & Execute incoming algorithm.

M00       CALL     FETCH_HEADER
          CALL     CHECK_DEST

M01:      CALL     XFER_SCI_2_PROG
          LACC     DEST
          BACC                     ; Branch to the address where
                                   ; code is loaded.


;***************************************************************************
; Routine Name: F E T C H _ H E A D E R            Routine Type: SR
;***************************************************************************
FETCH_HEADER:     CALL     FETCH_SCI_WORD
        LACC     data_buf
        SACL     DEST
        CALL     FETCH_SCI_WORD
        LACC     data_buf
        SACL     LENGTH
        RET


;***************************************************************************
; Routine Name: X F E R _ S C I _ 2 _ P R O G            Routine Type: SR
;***************************************************************************
XFER_SCI_2_PROG:
        MAR     *, AR0
        LAR     AR0, LENGTH
        LACC     DEST            ;ACC=dest address

XSP0          CALL     FETCH_SCI_WORD
        TBLW     data_buf        ;data_buff-->[*ACC]
        ADD      #01h            ;ACC++
        BANZ     XSP0            ;loop "length" times
        RET
```