Working with Flash

INTRODUCTION

Modifying the contents of Flash memory in the 41CL is not for the faint-of-heart. While the Flash Memory functions do prevent you from corrupting the Operating System of the calculator, they still allow you to erase or modify the rest of the Flash memory. You must be familiar with how Flash memory operates before attempting to use the Flash Memory functions.

FLASH MEMORY OPERATION

Flash memory has limited endurance, typically 100,000 write cycles, and is erased by sectors, which are usually 64K bytes (32K words, or eight pages) in the case of the 41CL. The only exceptions to this sector size are for the bottom eight pages or the top eight pages, and this depends on which version of the Flash memory has been stuffed on the board.

If the 41CL board uses a M29W160EB part the bottom sector (0x000000-0x007FFF) is broken into smaller sectors. This doesn't matter though, because the Flash Memory functions do not allow operations on this address range.

If the 41CL board uses a M29W160ET part the top sector (0x0F8000-0x0FFFFF) is broken into smaller sectors. This will only be important of you attempt to erase or write to addresses in this top sector. The sectors in this case are:

0x0F8000- 0x0FBFFF (a sector of 16K words)
0x0FC000-0x0FCFFF (a sector of 4K words)
0x0FD000-0x0FDFFF (a sector of 4K words)
0x0FE000-0x0FFFFF (a sector of 8K words)

All Beta 41CL boards use the "ET" parts.

An erased Flash sector returns 0xFFFF in every location. Only 0's can be written to any given location in Flash, which means that writes to Flash can only change a "1" to a "0" and never vice-versa.

FLASH PROGRAMMING

During a Flash erase or write, no other accesses of the Flash memory are allowed. This means that the Flash Memory functions must be running out of RAM to work. Both Flash Memory functions check for this, and return with an error message if they are not running in RAM. So, if you want to use either of the Flash memory functions you must copy the entire 41CL Extra Functions image to RAM and then program the MMU to use this RAM copy of these functions.

WARNINGS

Writing and erasing Flash memory requires that the 41CL circuit board draw more current than during normal operation. In addition, Flash memories are not tolerant of power disruptions during write and erase operations.

Do not start any Flash write or erase if the BATT annunciator is on. In fact it is best to have new batteries in the calculator when writing or erasing Flash memory. In addition, make sure that the 41CL circuit board is making good contact with the key-board PC board through the flexible connectors. If there is any play between the case halves make sure to repair it before attempting a write or erase of Flash. The extra current required by a Flash write or erase operation may cause the connector to flex sufficiently to disrupt the power to the 41CL board, which in turn may damage the Flash memory. Never remove the battery during a Flash write or erase operation.

EXAMPLE

As an example of how to replace an image in Flash memory, let's go through the steps necessary to update the YFNS image at address 0x062000. We cannot update the YFNZ image at address 0x007000 because this is in the protected sector of the Flash memory. I recommend that you start with a fresh set of batteries before trying this.

First we need to copy the entire 32K word sector (eight pages) to RAM. Any RAM will work, but I've chosen to use 0x0810000-0x817FFF.

```
ALPHA 060>810 ALPHA
XEQ ALPHA YMCPY ALPHA

ALPHA 061>811 ALPHA
XEQ ALPHA YMCPY ALPHA

ALPHA 062>812 ALPHA
XEQ ALPHA YMCPY ALPHA
```

```
ALPHA 063>813 ALPHA
XEQ ALPHA YMCPY ALPHA

ALPHA 064>814 ALPHA
XEQ ALPHA YMCPY ALPHA

ALPHA 065>815 ALPHA
XEQ ALPHA YMCPY ALPHA

ALPHA 066>816 ALPHA
XEQ ALPHA YMCPY ALPHA

ALPHA 067>817 ALPHA
XEQ ALPHA YMCPY ALPHA
```

Second, apply the patches to the copy of YFNS that is now at address 0x812000.

```
ALPHA 812680-005E ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812694-005E ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812F51-03B3 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812F6A-0343 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812F7A-02C3 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812DFD-0010 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812DFE-0110 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812DFF-0010 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812E00-0150 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812E01-0150 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812E08-02AA ALPHA
```

```
XEQ ALPHA YPOKE ALPHA

ALPHA 812E31-013C ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812E4C-013C ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D1B-0110 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D51-0010 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D52-0110 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D53-0010 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D54-0150 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D55-0150 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D5C-02AA ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D7E-013C ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D88-013C ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 812D97-013C ALPHA
XEQ ALPHA YPOKE ALPHA
```

Third, reprogram the MMU so that the patched copy of YFNS is being used instead of the one in Flash. This is necessary to guarantee that there will be no accesses of the Flash except for the erase or write operations while the Flash functions are executing. I am assuming that you will use Port 1L for the YFNS image.

```
ALPHA 812-RAM ALPHA
XEQ ALPHA PLUG1L ALPHA
```

Fourth, erase the sector of Flash that you want to reprogram. This function requires about two seconds to complete, and the display will be blank during execution. This step permanently destroys the contents of this sector of the Flash memory. Make sure that you really want to do this.

DO NOT REMOVE THE BATTERIES DURING YFERASE!

```
ALPHA 060000 ALPHA
XEQ ALPHA YFERASE ALPHA
```

Fifth, write the 8 pages in RAM, including the modified YFNS, back to the sector that you just erased. The YFWR function requires up to 30 seconds to complete (at 1x speed) and the display will be blank during execution.

DO NOT REMOVE THE BATTERIES DURING YFWR!

```
ALPHA 810>060 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 811>061 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 812>062 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 813>063 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 814>064 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 815>065 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 816>066 ALPHA
XEQ ALPHA YFWR ALPHA

ALPHA 817>067 ALPHA
XEQ ALPHA YFWR ALPHA
```

Sixth, to protect yourself from accidently writing more Flash, reprogram the MMU to use the YFNS that you just modified.

ALPHA YFNS ALPHA
XEQ ALPHA PLUG1L ALPHA
You're done!

EXPERIMENTING WITH FLASH

The 41CL includes several features for adding custom images to Flash that can be used with the PLUG functions. These features also make it easy to experiment with erasing and programming the Flash.

Five uncommitted module identifiers are provided:

XXXA identifies a 4K image located at address 0x0C8000

XXXB identifies a 4K image located at address 0x0D0000

XXXC identifies a 4K image located at address 0x0D8000

XXXD identifies an 8K image located at address 0x0E0000

XXXE identifies an 8K image located at address 0x0E8000

XXXF identifies a 16K image (4 banks) located at address 0x0F0000

Each identifier is located in a separate Flash sector, which makes it much easier to erase and program these locations if the remainder of the sector is not used. Note that the sector starting at address 0x0F0000 also holds the Sandmath-II image, so this one is a little more complicated to use.

If you want to practice using the Flash functions without the possibility of damaging any preprogrammed sectors of Flash, you might try writing the patched YFNS image to address 0x0C8000. You can then plug in the image, using either the direct address or the XXXA identifier.