Application Note

July 28, 2003

How to setup and use the Flash operations with the POD-51T:

Emulator Setup:

1) Replace U8 with PAL that was included with your POD. This PAL has been labeled “FLASH”.
2) On JP1 make sure BM0 and BM1 are installed and BM2 and BM3 are removed.
3) Jumpers labeled BS0 and BS1, located near the 50 pin cable connector, should both be installed.
4) Under the Config | Emulator and the Map Config tab, map all the XDATA memory to the emulator.

Note: ATR and ETR setup can not be set to drive the ANB or FLF signals.

POD Setup:

1) Place the jumpers on JP3 to be in the B0/ and B1/ positions.
2) Install a wire from the JP1, pin labeled with BSW0, to JP4 pin 8 (last pin closest to U7). BSW0 is low when the FCON bits 1, 2 or 3 are active on MOVX or MOVC cycles.
3) Install a wire from JP1, pin labeled with BSW1, to pin 7 (second pin closest to U7). BSW1 is low with FCON bits 1 or 2 are active.
4) The large FPGA (Lattice) should have a program date of 07/16/03 or latter.
5) Make sure the M1 jumper is removed and that you have an 89’series micro installed in the pod.

Operational Function of emulator:

Make sure that the first 4K block of XDATA is mapped to the emulator’s memory and not the target for this operation to work.

When FCON bit 1 and bit 2 are zero and bit 3 (FPS) is one, then MOVX will access code memory. This simulates the normal FLASH programming.

When FCON bit 1 or bit 2 are one, then MOVX and MOVC will access separate memory. This simulates the FLASH programming of extra ROWS. XDATA window will show this separate memory. Screen may have to be refreshed.

With AUXR.5 (ENBOOT) set to a 1 the code from F800h-FFFFh will be loaded from a separate memory. The boot loader must be loaded here.

Notes:

By placing the FCON register in you register display window and selecting it, you can right click on this register and select Change Attributes and setup to Enable reset value and place the value to be written to the FCON register on a chip reset that is done by the emulator.

The FLASH memory in the chip is not programmed. This is not possible in the special emulation mode of the chip. Only the memory in the emulator is altered when FCON bits 1, 2 or 3 are set and MOVX is performed.

The emulator will currently not check if a Programming Launch is performed. The programmer will have to manually check this.

FCON bit 0 (FBUSY) will always be clear, since the actual FLASH memory in the chip is not programmed.

SFR at address FFh should be set to 0Dh at power up.

The ENBOOT bit will be initialized to 1 if the BLOAD option is selected on the POD's mode jumper selections (M1 and M5).

Working with the Boot-loader:

To work with the Atmel **FLIP** program the following procedure must be used. This procedure must also be used if you are using any of the Atmel Bootloader IAP routines.

1) Start the Nohau Emulator
2) Map the XDATA regeion from 0 to FFFh (first 4k block) to the emulator. Also, **all** code memory must be mapped to the emulator.
3) Set the correct FCON value in respect to bits 1 and 2 (valid values would be 2, 4, or 6).
4) Patch the XDATA locations with the values from **table 1**
5) Load the nohau version of the bootloader code for your specific micro, with the ENBOOT bit of AUXR1 set to a 1.
6) Load your application. Make sure your application resides in lower memory and does not overlay the memory range where the boot loader code resides.
7) Run your application.

Nohau's boot loader versions are available from our web at: http://www.nohau.com/bootloaders.html

Boot loader for 89C51RB2 & 89C51RC2		Boot loader for 89C51RD2/ED2/ID2	
XDATA Location	Patch Value	XDATA Location	Patch Value
0000h	FFh	0000h	FFh
0001h	FCh	0001h	FCh
0004h	BBh	0004h	BBh
0005h	FFh	0005h	FFh
0030h	58h	0030h	58h
0031h	D7h	0031h	D7h
0060h	ECH	0060h	ECH
0061h	EFH	0061h	EFH

Table 1

RB2 / RC2 notes on locations:

; CODE address 0FFF0h is Entry Point for API
; CODE address 0F800h is ISP Start
;
; XAF address 00h (def 0FFh) is BSB (Boot Status Byte)
; XAF address 01h (def 0FCh) is SBV (Software Boot Vector)
; XAF address 04h (def 0101x1011b) is HSB Copy (Hardware Security Byte)
; XAF address 05h (def 0FFh) is SSB (Software Security Byte)
; XAF address 30h (def 058h) is Copy of Manufacturer Code for ATMEL
; XAF address 31h (def 0D7h) is Copy of Device ID #1
; XAF address 60h is Copy of Device ID #2
; XAF address 61h is Copy of Device ID #3
; XAF is Extra Row Memory (XROW). FMOD1:0 = 01b
; XAF is address at 0 – 7Fh and not FF80h
; Hardware Byte address 00h (erased 0111x1111b) is HSB. FMOD1:0 = 10b
; For Nohau Emulator, Hardware Byte is at 04h to coincide with HSB copy.