

FlashFlex Microcontroller In-Application Programming Basics



Application Note

August 2008

1.0 INTRODUCTION

For a number of years, In-System Programming (ISP) has been popular as a means to program or reprogram application code into the on-board memory block of MCUs located in user end products. ISP is accomplished by having downloader firmware preprogrammed into the on-board memory, which interfaces to PC software to download user application code. SST introduced In-Application Programming (IAP) in 1998, and its basic operation is similar to ISP. IAP differs from ISP in that it executes separately from the 8051 core controller. It operates between the two blocks of flash memory which is common in current SST MCUs. Specifically, one flash block can program the other flash block using IAP commands, while 8051 core user code is concurrently executing within the programming block.

In more detail, there are two executing controllers within FlashFlex[®] MCUs: the 8051 CPU core controller and the IAP flash memory controller. The two controllers can be operating concurrently, which means that once an IAP command is issued to the IAP controller via its mailbox SFR registers, the 8051 core is free to execute user instructions while the IAP controller is executing its command. For IAP programming, the guidelines to be followed are:

1. To enable IAP programming, set the IAPEN bit in the SFCF register (SFCF[6]=1).
2. To program Block 1, the IAP command code must reside in and be executed from Block 0.
3. To program Block 0, the IAP command code must reside in and be executed from Block 1.
4. To determine if an IAP command has completed, monitor the Flash_busy bit in the SFST register, that is, SFST[2]=0 indicates command completion.

2.0 SFR DESCRIPTIONS

IAP commands are initiated and controlled by the 8051 core controller through the following 6 SFRs: SuperFlash Status (SFST), SuperFlash Configuration (SFCF), SuperFlash Command (SFCM), SuperFlash Data (SFDT), SuperFlash Address Low (SFAL), and SuperFlash Address High (SFAH). A description of each of these SFRs follows:

2.1 SuperFlash Status (SFST)

Only one bit of this 8-bit register applies directly to IAP commands, bit 2 (Flash_busy). This bit is used to indicate a flash operation in progress and can be polled by the 8051 core controller to determine when a flash operation has been completed. This bit will be high while a flash operation is in progress. Flash_busy will only go low after the entire command has been completed.

2.2 SuperFlash Configuration (SFCF)

The important bit in this 8-bit register is bit 6 (IAPEN). The IAP Enable (IAPEN) bit is like an on/off switch for the IAP flash controller. When IAPEN=0, all IAP commands are ignored by the flash controller. When IAPEN=1, the flash controller will execute commands issued through the SFCM register.

2.3 SuperFlash Command (SFCM)

This 8-bit register is used to pass an IAP command word from the 8051 core controller to the IAP flash controller. Bit 7, Flash Interrupt Enable (FIE), indicates whether or not the IAP flash controller should issue an interrupt when the command has finished executing. If FIE=0 then no interrupt will be issued and the 8051 core controller will have to poll the BUSY and Flash_busy bits in the SFST register to determine when the command has been completed. If FIE=1 then the IAP flash controller will generate an interrupt on INT1# when the command is completed. (Note: This will disable external INT1# interrupts.) The remaining 7 bits contain the actual command word of the desired IAP command. (See Table 3-2 for a list of IAP commands.) Placing a new value in this register is the catalyst for starting an IAP operation in the IAP flash controller, therefore all of the other registers must be set up prior to loading the command word.



Application Note

2.4 SuperFlash Data (SFDT)

This 8-bit register is used to pass the data back and forth between the 8051 core controller and the IAP flash controller. During an IAP program operation the value in this register will be programmed by the flash controller into the address indicated by SFAH:SFAL. During an IAP verify operation this register will contain the data located and address SFAH:SFAL after the operation has been completed. During IAP erase operations the value in this register is checked to verify that the erase command is valid.

2.5 SuperFlash Address Low (SFAL)

This 8-bit register contains the 8 low-order address bits of the 16-bit address used by the IAP flash controller to access the internal flash memory.

2.6 SuperFlash Address High (SFAH)

This 8-bit register contains the 8 high-order address bits of the 16-bit address used by the IAP flash controller to access the internal flash memory.

3.0 IN-APPLICATION PROGRAMMING

Figure 3-1 provides the control flow for IAP operations, showing the mailbox SFR register interface between the 8051 Core Controller and the Flash Memory Controller. It also shows the effects of the IAPEN bit on enabling or disabling the Flash Memory Controller. The typical IAP command sequence is provided in Table 3-1, namely, the IAPEN bit is set, followed by loading the 8-bit data and 16-bit address for the specific command operation. Next the appropriate command is issued. The last step in an IAP command, which uses the polling method to detect flash operation completion, is polling the Flash_busy bit (SFST[2]). Typical IAP code examples are provided for Sector-Erase, Byte-Program, Byte-Verify, Prog-SB2, and Enable-Clock-Double. The entire list of available IAP commands is provided in Table 3-2.

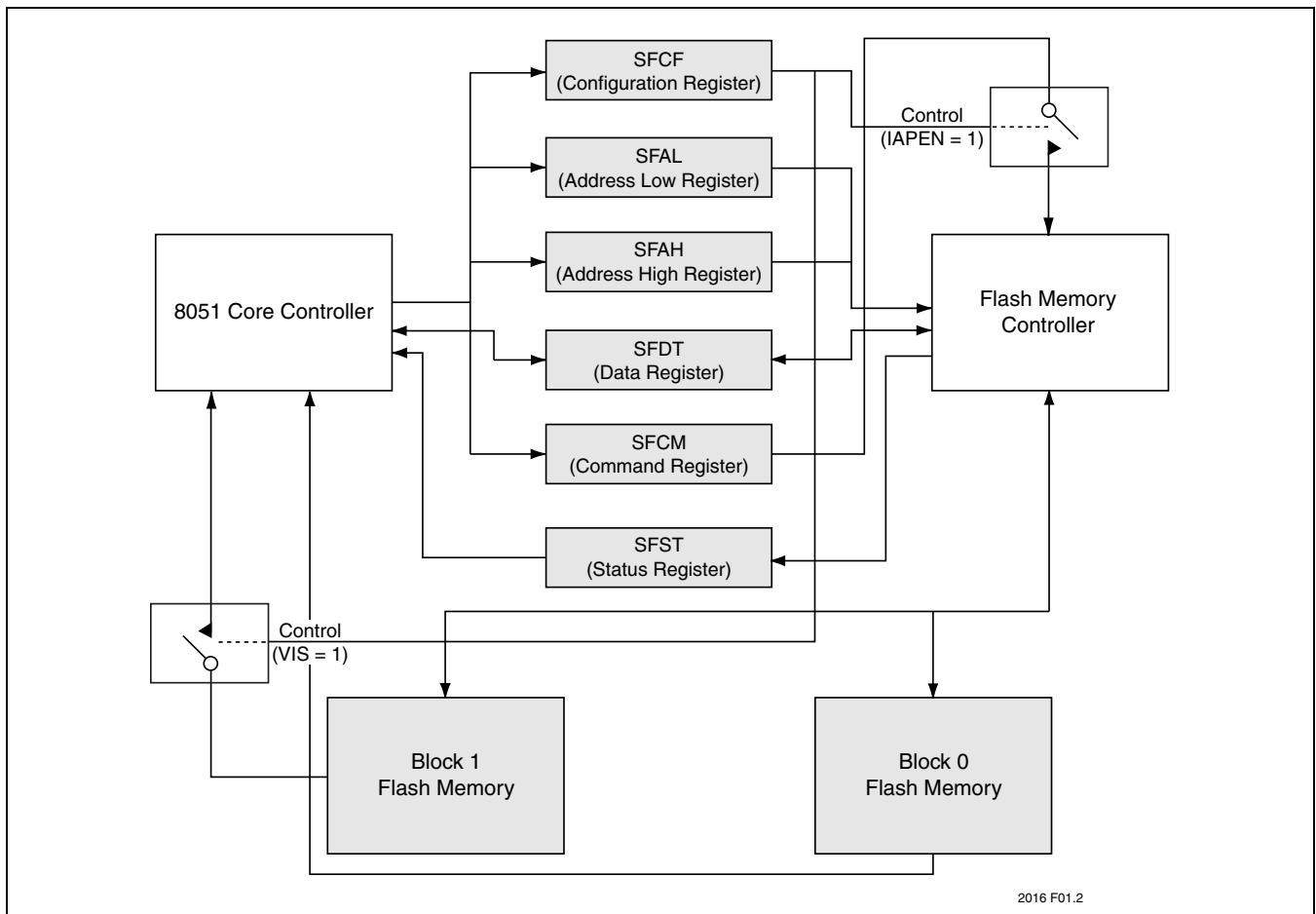


FIGURE 3-1: Control Flow of IAP Operation



TABLE 3-1: Typical IAP Command Sequence

Register	Contains
SFCF	enable IAP and make 4K Flash memory block visible
SFDT	data
SFAH	high order address byte
SFAL	low order address byte
SFCM	command #
SFST	monitor Flash_busy bit for command completion or wait for INT1# interrupt

T3-1.0 2016

3.1 IAP Code Examples

1. Sector-Erase example (in-line code):

```

::
::
::
MOV  A, SFCF      ; get SFCF contents
ORL  A, #0C0h    ; set IAPEN and Block 1 visibility
                    ; (#80h for SST89E/V516RDx
                    ; and SST89E/V58RDx)
MOV  SFCF, A     ; return to SFCF to enable IAP
                    ; and make 8K block visible
MOV  SFAH, (adr_h) ; put high-address byte of
                    ; sector start into SFAH
MOV  SFAL, (adr_l) ; put low-address byte of sector
                    ; start into SFAL
MOV  SFCM, #0Bh  ; issue sector erase command
cmd_done0:
MOV  A, SFST     ; get status
ANL  A, #04h    ; mask for flash_busy bit
CJNE A, #00h, cmd_done0 ; repeat loop until IAP
                    ; operation completes
::
::
::

```

2. Byte-Program example (in-line code):

```

::
::
::
MOV  A, SFCF      ; get SFCF contents
ORL  A, #0C0h    ; set IAPEN and Block 1 visibility
                    ; (#80h for SST89E/V516RDx
                    ; and SST89E/V58RDx)
MOV  SFCF, A     ; return to SFCF to enable IAP
                    ; and makes 8K block visible
MOV  SFDT, (data) ; post data to be programmed
MOV  SFAH, (adr_h) ; put high-address byte into
                    ; SFAH
MOV  SFAL, (adr_l) ; put low-address byte into SFAL
MOV  SFCM, #0Eh  ; issue program byte command
cmd_done1:
MOV  A, SFST     ; get status
JB   A.2, cmd_done1 ; repeat loop until IAP
                    ; operation completes
::
::
::

```

3. Byte-Verify (Read) example (in-line code):

```

::
::
::
MOV  A, SFCF      ; get SFCF contents
ORL  A, #0C0h    ; set IAPEN and Block 1 visibility
                    ; (#80h for SST89E/V516RDx
                    ; and SST89E/V58RDx)
MOV  SFCF, A     ; return to SFCF to enable IAP
                    ; and makes 8K block visible
MOV  SFAH, (adr_h) ; put high-address byte into
                    ; SFAH
MOV  SFAL, (adr_l) ; put low-address byte into SFAL
MOV  SFCM, #0Ch  ; issue verify byte command
                    ; the verify byte command has a
                    ; very fast execution time and
                    ; does not require the 8051
                    ; controller to poll for completion
MOV  A, SFDT     ; get data read
::
::
::

```



FlashFlex Microcontroller In-Application Programming Basics

Application Note

4. The following code segment will put the SST89E/V58RDx / SST89E/V516RDx in Security Level 3 SoftLock/SoftLock (010b) from Security Level 1 (000b) using an IAP command with interrupt (in-line code):

```

::
::
::
ORL   SFCF,#40h   ; enable IAP commands
MOV   SFDT, #0AAh ; program security bit setup
MOV   SFCM, #83h  ; issue Prog-SB2 IAP command
                           ; interrupt INT1# occurrence
                           ; indicates completion
::
::
::

```

5. Enable-Clock-Double example for the SST89E/V58RDx / SST89E/V516RDx (in-line code):

```

::
::
::
ORL   SFCF, #040h   ; enable IAP
MOV   SFAH, #055h
MOV   SFAL, #000h
MOV   SFDT, #0AAh
MOV   SFCM, #008h   ; clock-double command
WAIT:
MOV   A, SFST       ; check IAP status
JB    ACC.2, WAIT   ; wait till done
ANL   SFCF, #0BFh   ; disable IAP
::
::
::

```

TABLE 3-2: IAP Commands for SST89E/V516RDx / SST89E/V58RDx

Operation	SFAH [7:0]	SFAL [7:0]	SFDT [7:0]	SFCM [6:0] ¹
Chip-Erase ²	X	X	55H	01H
Block-Erase ³	AH	X	55H	0DH
Sector-Erase ³	AH	AL	X	0BH
Byte-Program ³	AH	AL	DI	0EH
Byte-Verify (Read) ³	AH	AL	DO	0CH
Prog-SB1 ⁴	X ⁵	X	AAH	0FH
Prog-SB2 ⁴	X	X	AAH	03H
Prog-SB3 ⁴	X	X	AAH	05H
Prog-SC0 ⁴	5AH	X	AAH	09H
Prog-SC1 ^{4,6}	AAH	X	AAH	09H
Enable-Clock-Double	55H	X	AAH	08H

T3-2.1 2016

- Interrupt/Polling enable for flash operation completion
SFCM[7] = 1: Interrupt enable for flash operation completion
0: polling enable for flash operation completion
- Chip-Erase only functions in IAP mode when EA#=0 (external memory execution) and device is not in level 4 locking.
- Refer to the SST89E/V516RDx / SST89E/V58RDx data sheet for address resolution.
- Instruction must be located in Block 1 or external code memory.
- X can be V_{IL} or V_{IH}, but no other value.
- Command exists only for the SST89E/V58RDx.

Note: X = Don't Care; AL = Address low order byte; AH = Address high order byte; DI = Data Input; DO = Data Output
All other values are in hex.
SFCF[6] = 1 enables IAP command. SFCF[6] = 0 disables IAP command.