1.0 INTRODUCTION

The CompactFlash (CF) card SST48CFxxx supports three operational modes: (1) PC card ATA using I/O mode, (2) PC card ATA using Memory mode, and (3) TrueIDE mode. Individual application notes have been generated that provide reference designs for an SST FlashFlex51 MCU to control a CF card operating in each of its three operational modes. This application note introduces the hardware and firmware reference design for the SST89C54/581 (8051-compatible microcontroller) to control the CompactFlash card SST48CFxxx2 in TrueIDE mode. For the other two modes, please reference two additional application notes at www.sst.com, namely FlashFlex51 Microcontroller Control of CompactFlash Card in Memory Mode and FlashFlex51 Microcontroller Control of CompactFlash Card in I/O Mode.

2.0 HARDWARE DESIGN

The following hardware design description refers to the detailed schematic diagram provided in Appendix A.

To enter TrueIDE mode, it is mandatory that /OE (pin 9, also called /ATASEL) be low during power-up. If /OE is high during power-up, the CF card will enter into Memory mode.

/DASP (pin 45) is connected to an LED through a resistor to VDD, which provides user visibility of CF card internal operation. When the CF card is busy, the LED will be on.

In TrueIDE mode, the CF card supports two-card operation. One card can be set as master (/CSEL=low) and the other as slave (/CSEL=high). This reference design implements only one CF card which can be set as either master or slave. The firmware checks pin P3.5 to determine whether the CF card is master or slave, but the firmware CANNOT change the setting on the fly. Any change on the /CSEL pin will take effect after the next reset.

One address latch 74HC373 (or 74HC573) is eliminated by connecting the address bits A0, A1, A2 of the CF card directly to 8051 MCU address pins A8 (P2.0), A9 (P2.1), and A10 (P2.2), instead of connecting them to A0, A1, A2 outputs of an address latch device. The same firmware code will work for either hardware connection scheme.

Connecting P1.1 to /RESET (pin 41) provides the capability to reset the CF card at any time, whether due to a system requirement or a system bug.

3.0 FIRMWARE DESIGN

It is important to understand that the ATA/IDE standard does not permit access to media such as HDD or CF card of one byte at a time. The firmware must read or write data one or more sectors at a time, where one sector equals 512 bytes of data. The system design engineer must incorporate a data buffer to support random access to CF card.

This reference design uses the secondary block (4 KByte x 8 bit) of on-chip flash memory of the SST89C54/58 as a data buffer.

In TrueIDE mode, memory or attribute registers are not accessible to the host. The default CF card operation is 16-bit. The firmware must explicitly run the Set-Features command to enable 8-bit operation before 8-bit data transfers commence.

After power-up, a reset at pin 41, or a software reset, the SST48CFxxx CF card will be ready to perform any operation after waiting 50 ms (typical) or 400 ms (maximum)3. Thus, the firmware needs either to add a 400 ms delay or poll the Busy and RDY bits of the status register until the CF card is ready. Additionally, software reset provides the host MCU another choice to reset the CF card, even if the CF card is busy on an internal operation.

The complete 8051 source code for this application is provided in Appendix B.

4.0 CONCLUSION

8051 control of the CF card in TrueIDE mode is straightforward, and it is easy to modify this reference design to support any other embedded controller as long as the designer follows the guidelines provided in this application note.

---

1. SST89C54/58 data sheet, www.sst.com
2. CompactFlash card SST48CFxxx data sheet, www.sst.com
3. See “Start Up Time” under Features on page 1 of the CF card data sheet.
APPENDIX A. SCHEMATIC DIAGRAM

OPEN = SLAVE
CLOSE = MASTER
BUSY = LED ON
READY = LED OFF
ON = Verification Success
Blink = Verification Failure
APPENDIX B. FIRMWARE 8051 SOURCE CODE

;=======================================================================
; This code is for TrueIDE mode of CompactFlash Card (CF Card).
; All commands are supported by SST48CFxxx CF Cards.
;=======================================================================
ChkPwr EQU 0E5h ; or 98h
Diagnostic EQU 90h
EraseSctr EQU 0C0h
Format EQU 50h
Identify EQU 0ECh
Idle EQU 0E3h ; or 97h
IdleImm EQU 0E1h ; or 95h
Initialize EQU 91h
ReadBuf EQU 0E4h
ReadLong EQU 22h ; or 23h
ReadMulti EQU 0C4h
ReadSctr EQU 20h ; or 21h
ReadVerify EQU 0E4h ; or 41h
Recalibrate EQU 10h ; or 1xh
ReqSense EQU 03h
Seek EQU 70h ; or 7xh
SetFeature EQU 0EFh
SetMulti EQU 0C6h
Sleep EQU 0E6h ; or 99h
Standby EQU 0E2h ; or 96h
StandbyImm EQU 0E0h ; or 94h
Translate EQU 87h
WearLevel EQU 0F5h
WriteBuf EQU 0E8h
WriteLong EQU 32h ; or 33h
WriteMulti EQU 0C5h
WriteSctr EQU 30h ; or 31h
WriteVerify EQU 0CCh
WrtMwoErase EQU 0CDh
WrtSwoErase EQU 38h
;=======================================================================
; CF Card Drive Register Set Address Allocations
Data_Reg EQU 8000h ; Data Register for read / write
Error_Reg EQU 8101h ; Error Register, read only
Features EQU 8101h ; Features Register, write only
Sectr_Cnt EQU 8202h ; Sector Count Register ( R / W )
Sectr_No EQU 8303h ; Sector Number Register, or LBA0:7 ( R / W )
Cylinder_Low EQU 8404h ; Cylinder Low Register or LBA8:15 ( R / W )
Cylinder_Hi EQU 8505h ; Cylinder High Register or LBA16:23 ( R / W )
Drv_Head EQU 8606h ; Drive Head Register ( R / W )
Status EQU 8707h ; Status Register, read only
Command EQU 8707h ; Command Register, write only
Alt_Status EQU 4606h ; Alternate Status Register, read only.
; Note: reading Alt_Status doesn't clear interrupt pending flag.
; Alt_Status register not used in this demo.
Device_Ctrl EQU 4606h ; Device Control Register, write only. Not used in this demo.
Drive_Addrs EQU 4707h ; Drive Address Register, read only. Not used in this demo.
Application Note

; SST FlashFlex51 Microcontroller-Related SFR Definitions
SFCF  DATA  0B1H ; SuperFlash Configuration
SFCM  DATA  0B2H ; SuperFlash Command
SFAL  DATA  0B3H ; SuperFlash Address Low
SFAH  DATA  0B4H ; SuperFlash Address High
SFDT  DATA  0B5H ; SuperFlash Data
SFST  DATA  0B6H ; SuperFlash Status
WDTC  DATA  0C0H ; Watchdog Timer Control
WDTD  DATA  086H ; Watchdog Timer Data/Reload

; Constant Definitions
FlashAddrs EQU 0F800h ; start address of MCU on-chip flash
Select EQU B ; bit 4 in Reg B is 0 for MASTER, B.4=1 for SLAVE

; Hardware Connection Description
RST BIT P1.1 ; RESET# = 1 (normal), 0 (Host resets CF Card)
D0-D7 to 8051 AD0-AD7
A0,A1,A2 to 8051 latched address A0,A1,A2 or direct to 8051 A8,A9,A10
/CS0 to 8051 A14
/CS1 to 8051 A15
/CSEL to 8051 P3.5
/IOWR to 8051 WR
/ORD to 8051 RD
/ATASEL to GND
DASP# is connected to a LED in series with a 1K resistor to VDD.
When the CF Card is inactive, LED is OFF; when CF Card is active, LED is ON.
All other pins of CF Card are NOT Connected.

; Code begins here

org 0000h
ljmp start ; reset vector
org 0100h

start:
clr RST ; reset CF Card
mov Select, #1110$0000b ; D4=0 for master CF Card
jnb P3.5, master ; check CF Card is set as Master or Slave
mov Select, #1111$0000b ; D4=1 for slave CF Card

master:
nop
nop
setb RST

; Initialization for erasing MCU flash memory sectors via IAP
orl SFCF, #40h ; IAPEN=1
mov SFAH, #high(FlashAddrs)
mov SFAL, #low(FlashAddrs)
mov R7, #9 ; erase 9 sectors (512 + 64 bytes)

erase:
mov SFCM, #0Bh ; sector erase!
acall Done?
mov a, SFAL
add a, #64
mov SFAL, a
mov a, SFAH
addc a, #0
mov SFAH, a
djnz R7, erase
anl SFCF, #0BFh ; disable IAP
mov r4, #5
loadr5: mov r5, #200
loadr6: mov r6, #250
 djnz r4, loadr6
 djnz r5, loadr6
 djnz r4, loadr5
; Save register reset values into MCU flash for diagnostic purposes
mov dptr, #Drv_Head ; It's very IMPORTANT for slave to
mov a, Select ; select MASTER/SLAVE data bus
movx @dptr, a
mov dptr, #Error_Reg
movx a, @dptr
mov r1, a
mov dptr, #Sectr_Cnt
movx a, @dptr
mov r2, a
mov dptr, #Sectr_No
movx a, @dptr
mov r3, a
mov dptr, #Cylinder_Low
movx a, @dptr
mov r4, a
mov dptr, #Cylinder_Hi
movx a, @dptr
mov r5, a
mov dptr, #Drv_Head
movx a, @dptr
mov r6, a
mov dptr, #Status
movx a, @dptr
mov r7, a

; Now save register values into MCU on-chip flash memory
orl SFCF, #40h ; IAPEN=1
mov r0, #1
mov SFAH, #high(FlashAddrs+200h)
mov SFAL, #low(FlashAddrs+200h)
mov more: a, @r0
mov SFDT, a
mov SFCM, #0Eh ; byte-program
lcall done?
inc SFAL
inc r0
cjne r0, #8, more

; Reset address pointer to beginning of MCU flash memory
mov SFAH, #high(FlashAddrs)
mov SFAL, #low(FlashAddrs)
alcall Enable8bit ; First of all, enable 8 bits operation!

; Perform CF Card sector operations (Write, Read, and Compare)
main: alcall Write_Sctr
alcall Read_Sctr
alcall Compare
jb F0, fail

pass: clr P1.0 ; indicates successful operations.
sjmp pass

fail: cpl P1.0 ; flags failed comparison.
mov r4, #10 ; delay 1 second
mov register5: r5, #200 ; delay 0.1 second
mov register6: r6, #250 ; delay 0.5ms for 12MHz crystal
djnz r6, $
djnz r5, register6
sjmp fail
Application Note

;=======================================================================
; SUBROUTINES
;=======================================================================
; Sets up sector count, LBA addresses and command code
;=======================================================================
Function: acall Busy
        mov dptr, #Sectr_Cnt
        mov a, R2 ; R2 is Sector Count
        movx @dptr, a
        mov dptr, #Sectr_No
        mov a, R3 ; R3 contains LBA0:7
        movx @dptr, a
        mov dptr, #Cylinder_Low
        mov a, R4 ; R4 contains LBA8:15
        movx @dptr, a
        mov dptr, #Cylinder_Hi
        mov a, R5 ; R5 contains LBA16:23
        movx @dptr, a
        mov dptr, #Drv_Head
        mov a, R6 ; R6 contains LBA24:27
       anl a, #00001111b
        orl a, Select ; CF Card as MASTER / SLAVE Drive, LBA enable.
        movx @dptr, a
        mov dptr, #command
        mov a, R7 ; R7 is command code.
        ret

;=======================================================================
; Checks status of CF Card
;=======================================================================
Busy: mov dptr, #status
        movx a, @dptr
        jb acc.7, Busy ; if BUSY=1, then Busy
        jb acc.0, errors ; if ERR=1, then read errors code and set flag C
        clr a ; acc=0 when successful
        clr C ; C=0, CF Card is not busy (BUSY=0) and no error (ERR=0)
        ; and is ready to accept commands (RDY=1)
        ret

errors: mov dptr, #Error_Reg
        movx a, @dptr
        setb C ; C=1 flags error codes contained in ACC register
        ret

;=======================================================================
; Waiting for Data Request from CF Card
;=======================================================================
WaitDRQ: mov dptr, #status
        movx a, @dptr
        jb acc.7, WaitDRQ ; if BUSY=1, then WaitDRQ
        jnb acc.3, WaitDRQ ; if DRQ=0, then WaitDRQ
        jb acc.0, errors ; if ERR=1, then read errors code and set flag C
        clr a
        clr C ; C=0, CF Card is BUSY=0, DRQ=1, ERR=0.
        ret

;=======================================================================
; Waiting for IAP Operation to complete
;=======================================================================
Done?: mov a, SFST
        jb acc.2, Done?
        ret
Application Note

FlashFlex51 Microcontroller
Control of CompactFlash Card in TrueIDE Mode

;=======================================================================
; Enables 8-bit CF Card data operation
;=======================================================================
Enable8bit: acall Busy
    mov dptr, #Features
    mov a, #01h ; enable 8 bit data transfer
    movx @dptr, a
    mov dptr, #Drv_Head
    mov a, Select ; LBA=1, CF Card as MASTER/SLAVE if bit D4=0 or 1
    movx @dptr, a
    mov dptr, #COMMAND
    mov a, #0EFh
    movx @dptr, a
    ret

;=======================================================================
; Write data to one 512-byte sector in CF Card
;=======================================================================
Write_Sctr: mov R2, #1 ; write 1 sector at a time.
    mov R3, #0Ah ; suppose LBA to be 000000Ah
    mov R4, #0
    mov R5, #0
    mov R6, #0
    mov R7, #WriteSctr
    acall Function
    acall WaitDRQ
    acall Write512
    ret

Write512: mov R0, #high(message) ; get the higher address of message
    mov R1, #low(message) ; get the lower address of message
    mov R7, #2 ; 512 bytes = 2 * 256
    mov R6, #0
    write:
    mov dph, R0 ; get the address
    mov dpl, R1
    clr a
    movc a, @a+dptr ; get the data in message
    inc dptr ; point to next byte in message
    mov R0, dph ; save the address
    mov R1, dpl
    mov DPTR, #Data_Reg ; point to CF Card data register
    movx @dptr, a ; write 1 data byte into CF Card
    djnz R6, write
    djnz R7, write ; write all 512 bytes to CF Card
    ret
Application Note

;=======================================================================
; Read data from one 512-byte sector in CF Card
;=======================================================================
Read_Sctr:  mov  R2, #1 ; read 1 sector at a time.
mov  R3, #0Ah  ; suppose LBA to be 000000Ah
mov  R4, #0
mov  R5, #0
mov  R6, #0
mov  R7, #ReadSctr
acall Function
acall WaitDRQ
acall Read512
ret

; Read 1 sector of 512 data bytes and write into on-chip flash of SST FlashFlex51 MCU
Read512: mov  R7, #2  ; 512 bytes = 2 * 256
mov  R6, #0
mov  dptr, #Data_Reg
mov  SFAH, #high(FlashAddrs)
mov  SFAL, #low(FlashAddrs)
orl  SFCF , #40h  ; set IAPEN=1 to enable IAP
read: movx a, @dptr
mov  SFDT, a
mov  SFCM, #0Eh  ; issue Byte-Program command
acall Done?  ; wait until done
mov  a, SFAL  ; adjust the address of flash
add  a, #1
mov  SFAL, a
mov  a, SFAH
addc  a, #0
mov  SFAH, a
djnz  R6, read
djnz  R7, read
anl  SFCF, #0BFh  ; disable IAP
ret
FlashFlex51 Microcontroller
Control of CompactFlash Card in TrueIDE Mode

;=======================================================================
; Compare saved data with original data
;=======================================================================
Compare:     mov dph, #high(message) ; get the higher address of message
             mov dpl, #low(message) ; get the lower address of message
             mov SFAH, #high(FlashAddrs)
             mov SFAL, #low(FlashAddrs)
             orl SFCF , #40h ; IAPEN=1
             clr F0
             mov R7, #2
             mov R6, #0
verify:      clr a
             movc a, @a+dptr
             inc dptr
             mov SFCM, #0Ch ; issue BYTE-VERIFY command
             nop
             xrl a, SFDT
             jz equal
             setb F0 ; set flag F0 (PSW.5) if any discrepancy
equal:       mov a, SFAL ; increase the address of MCU flash
             add a, #1
             mov SFAL, a
             mov a, SFAH
             addc a, #0
             mov SFAH, a
             djnz R6, verify
             djnz R7, verify
             anl SFCF , #0BFh ; disable IAP
             ret
;=======================================================================
; This message string has exactly 512 bytes of data to be written into the CF card.
message:     DB "This program demonstrates how to interface a CompactFlash card to an "
             DB "SST FlashFlex51 embedded microcontroller. The CompactFlash card is "
             DB "an ultra-small, low-cost, high-performance, removable, flash memory "
             DB "storage device. This product is well-suited for portable solid state "
             DB "mass storage applications offering new and expanded functionality while "
             DB "enabling smaller and lighter designs. The CompactFlash card can operate "
             DB "in 8-bit or 16-bit mode allowing for easy interfacing to any application system. "
             DB "www.SST.com "
             end