

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, WDT
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.154", 3.90mm Width)
Supplier Device Package	16-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f833-gsr

C8051F80x-83x

15.2. Data Memory	93
15.2.1. Internal RAM	93
15.2.1.1. General Purpose Registers	94
15.2.1.2. Bit Addressable Locations	94
15.2.1.3. Stack	94
16. In-System Device Identification	95
17. Special Function Registers	97
18. Interrupts	102
18.1. MCU Interrupt Sources and Vectors	103
18.1.1. Interrupt Priorities	103
18.1.2. Interrupt Latency	103
18.2. <u>Interrupt Register</u> Descriptions	104
18.3. INT0 and INT1 External Interrupts	111
19. Flash Memory	113
19.1. Programming The Flash Memory	113
19.1.1. Flash Lock and Key Functions	113
19.1.2. Flash Erase Procedure	113
19.1.3. Flash Write Procedure	114
19.2. Non-volatile Data Storage	114
19.3. Security Options	114
19.4. Flash Write and Erase Guidelines	115
19.4.1. VDD Maintenance and the VDD Monitor	116
19.4.2. PSWE Maintenance	116
19.4.3. System Clock	117
20. Power Management Modes	120
20.1. Idle Mode	120
20.2. Stop Mode	121
20.3. Suspend Mode	121
21. Reset Sources	123
21.1. Power-On Reset	124
21.2. Power-Fail Reset / VDD Monitor	125
21.3. External Reset	126
21.4. Missing Clock Detector Reset	126
21.5. Comparator0 Reset	127
21.6. PCA Watchdog Timer Reset	127
21.7. Flash Error Reset	127
21.8. Software Reset	127
22. Oscillators and Clock Selection	129
22.1. System Clock Selection	129
22.2. Programmable Internal High-Frequency (H-F) Oscillator	131
22.3. External Oscillator Drive Circuit	133
22.3.1. External Crystal Example	135
22.3.2. External RC Example	136
22.3.3. External Capacitor Example	137
23. Port Input/Output	138

C8051F80x-83x

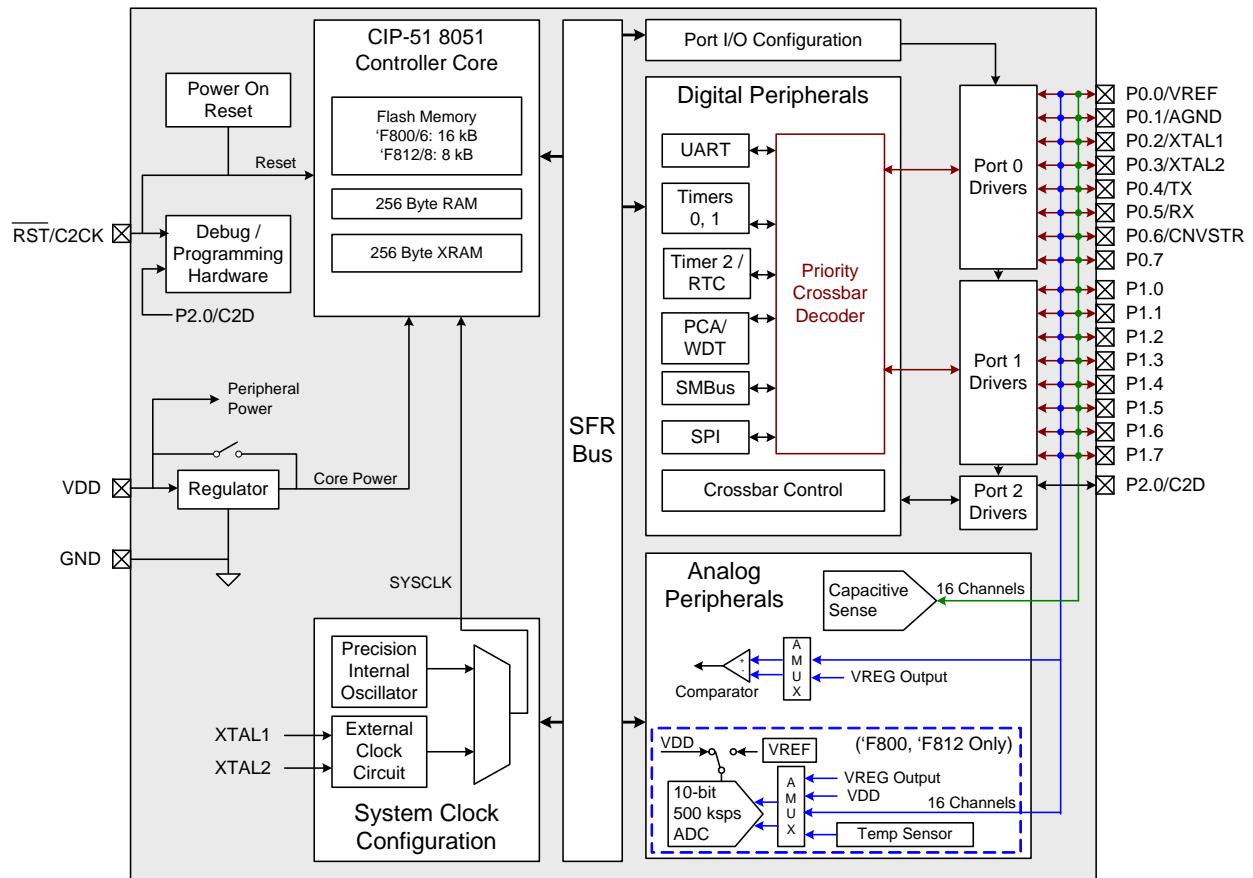


Figure 1.1. C8051F800, C8051F806, C8051F812, C8051F818 Block Diagram

C8051F80x-83x

Table 3.1. Pin Definitions for the C8051F80x-83x (Continued)

Name	Pin QSOP-24	Pin QFN-20	Pin SOIC-16	Type	Description
P0.5	21	16	14	D I/O or A In	Port 0.5.
P0.6/ CNVSTR	20	15	13	D I/O or A In D In	Port 0.6. ADC0 External Convert Start or IDA0 Update Source Input.
P0.7	19	14	12	D I/O or A In	Port 0.7.
P1.0	18	13	11	D I/O or A In	Port 1.0.
P1.1	17	12	10	D I/O or A In	Port 1.1.
P1.2	16	11	9	D I/O or A In	Port 1.2.
P1.3	15	10	8	D I/O or A In	Port 1.3.
P1.4	14	9		D I/O or A In	Port 1.4.
P1.5	11	8		D I/O or A In	Port 1.5.
P1.6	10	7		D I/O or A In	Port 1.6.
P1.7	9	6		D I/O or A In	Port 1.7.
NC	1, 12, 13, 24				No Connection.

C8051F80x-83x

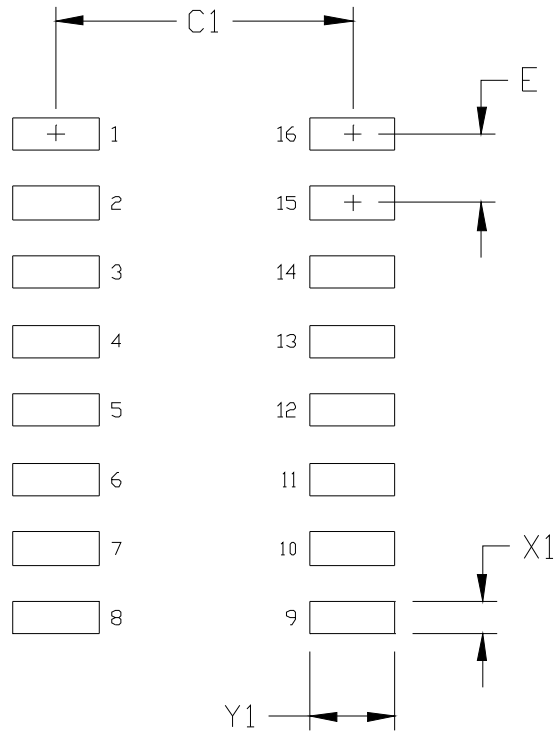


Figure 6.2. SOIC-16 PCB Land Pattern

Table 6.2. SOIC-16 PCB Land Pattern Dimensions

Dimension	Feature	(mm)
C1	Pad Column Spacing	5.40
E	Pad Row Pitch	1.27
X1	Pad Width	0.60
Y1	Pad Length	1.55

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on IPC-7351 pattern SOIC127P600X165-16N for Density Level B (Median Land Protrusion).
3. All feature sizes shown are at Maximum Material Condition (MMC) and a card fabrication tolerance of 0.05 mm is assumed.

C8051F80x-83x

8.4.1. Window Detector Example

Figure 8.4 shows two example window comparisons for right-justified data, with $ADC0LTH:ADC0LTL = 0x0080$ (128d) and $ADC0GTH:ADC0GTL = 0x0040$ (64d). The input voltage can range from 0 to $VREF \times (1023/1024)$ with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an $AD0WINT$ interrupt will be generated if the $ADC0$ conversion word ($ADC0H:ADC0L$) is within the range defined by $ADC0GTH:ADC0GTL$ and $ADC0LTH:ADC0LTL$ (if $0x0040 < ADC0H:ADC0L < 0x0080$). In the right example, and $AD0WINT$ interrupt will be generated if the $ADC0$ conversion word is outside the range defined by the $ADC0GT$ and $ADC0LT$ registers (if $ADC0H:ADC0L < 0x0040$ or $ADC0H:ADC0L > 0x0080$). Figure 8.5 shows an example using left-justified data with the same comparison values.

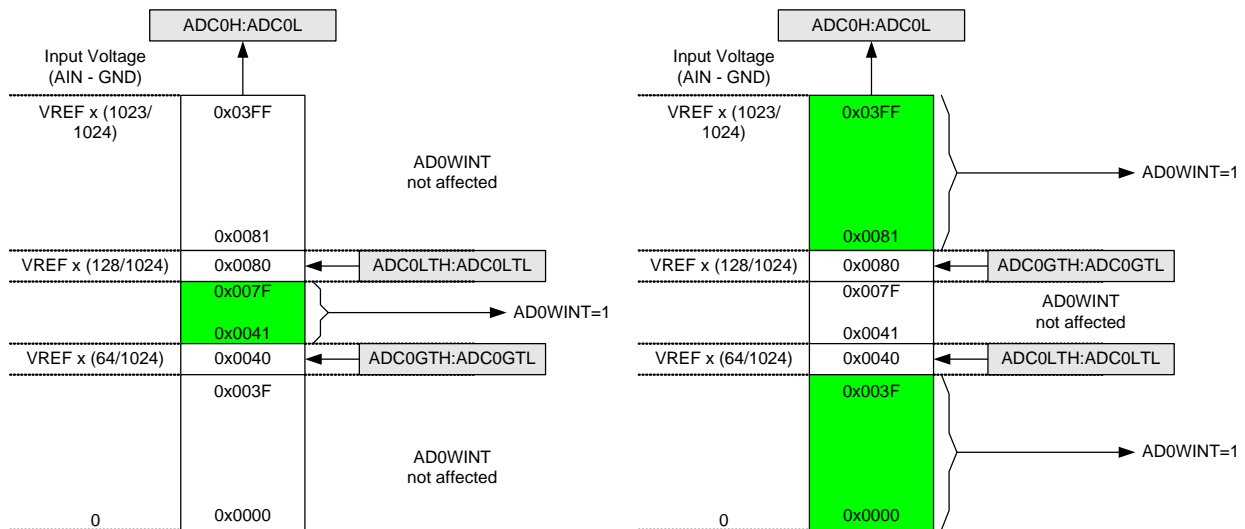


Figure 8.4. ADC Window Compare Example: Right-Justified Data

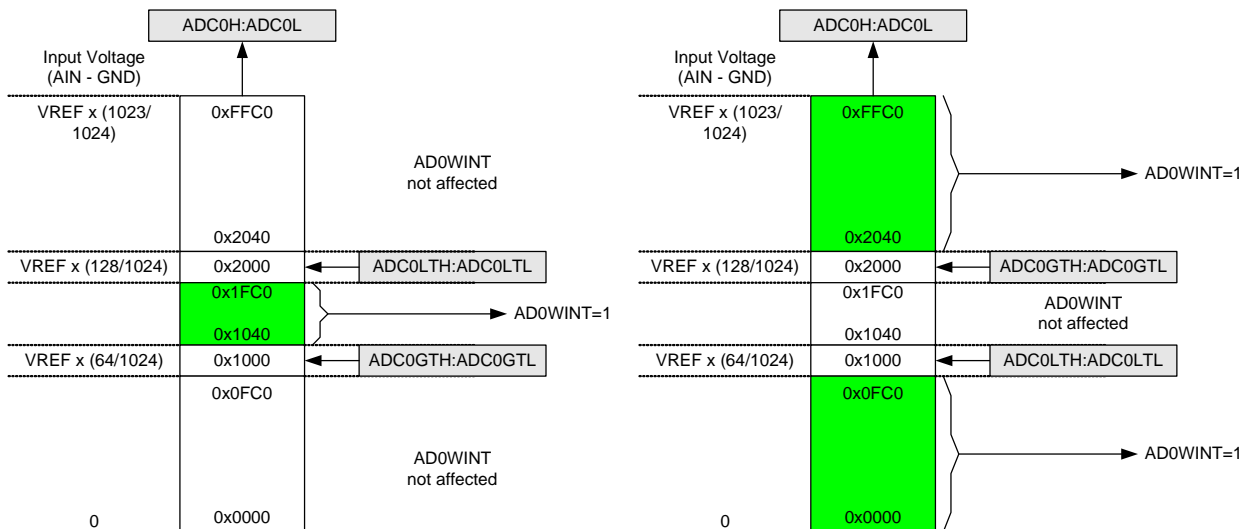


Figure 8.5. ADC Window Compare Example: Left-Justified Data

15. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory organization of the C8051F80x-83x device family is shown in Figure 15.1

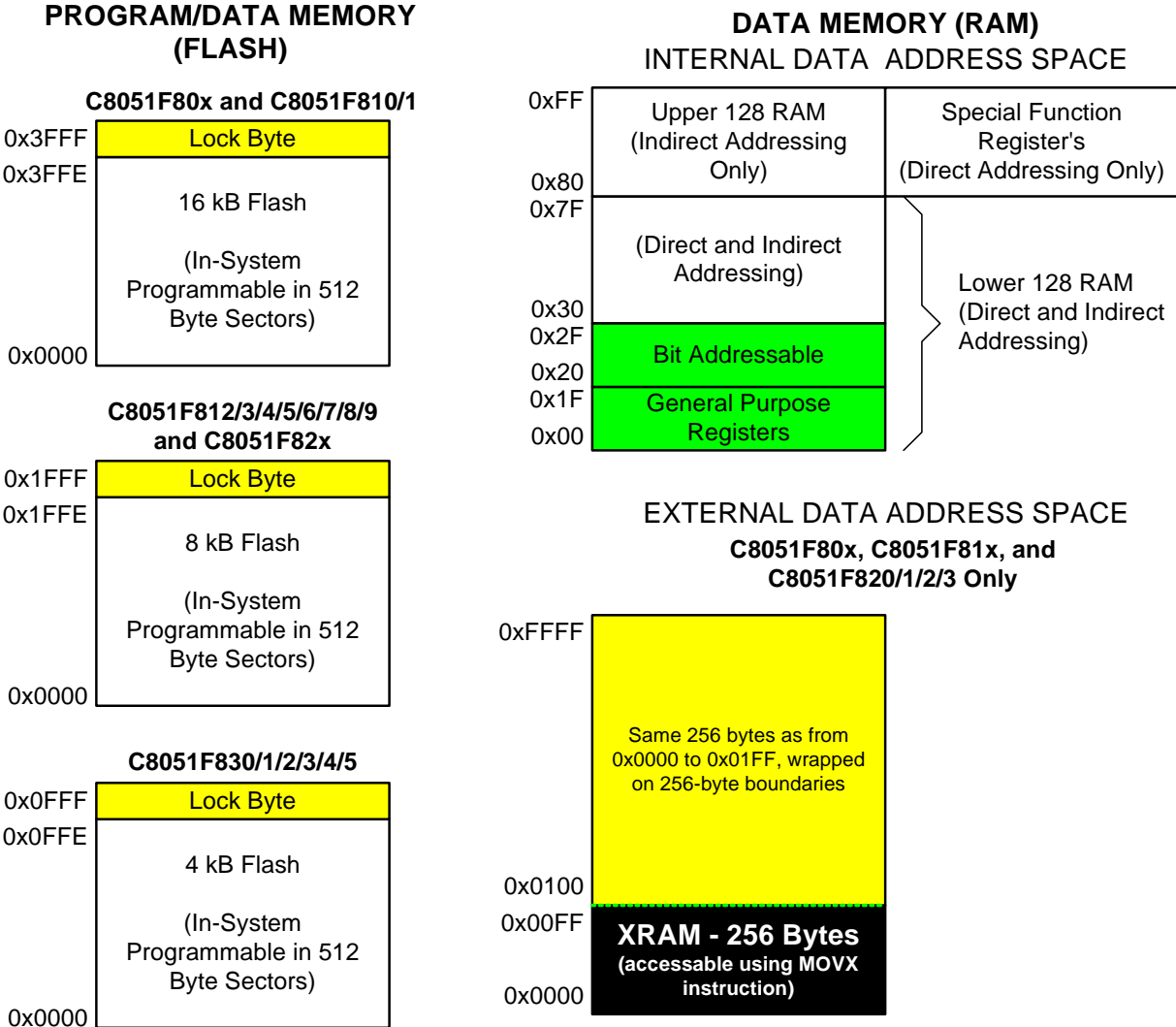


Figure 15.1. C8051F80x-83x Memory Map

17. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the C8051F80x-83x's resources and peripherals. The CIP-51 controller core duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the C8051F80x-83x. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 17.1 lists the SFRs implemented in the C8051F80x-83x device family.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g., P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 17.2, for a detailed description of each register.

Table 17.1. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	P0MAT	P0MASK	VDM0CN
F0	B	P0MDIN	P1MDIN	EIP1	EIP2			PCA0PWM
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	P1MAT	P1MASK	RSTSRC
E0	ACC	XBR0	XBR1		IT01CF		EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	CRC0IN	CRC0DATA	
D0	PSW	REF0CN	CRC0AUTO	CRC0CNT	P0SKIP	P1SKIP	SMB0ADM	SMB0ADR
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	CRC0CN	CRC0FLIP
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	
B8	IP	CS0SS	CS0SE	ADC0MX	ADC0CF	ADC0L	ADC0H	
B0	CS0CN	OSCXCN	OSCI CN	OSCI CL		HWID	REVID	FLKEY
A8	IE	CLKSEL		CS0DL	CS0DH	DERVID		
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	
98	SCON0	SBUF0		CPT0CN	CS0MX	CPT0MD	CS0CF	CPT0MX
90	P1						CS0THL	CS0THH
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH				PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Note: SFR Addresses ending in 0x0 or 0x8 are bit-addressable locations, and can be used with bitwise instructions.

Table 17.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
SBUF0	0x99	UART0 Data Buffer	207
SCON0	0x98	UART0 Control	206
SMB0ADM	0xD6	SMBus Slave Address mask	191
SMB0ADR	0xD7	SMBus Slave Address	191
SMB0CF	0xC1	SMBus Configuration	186
SMB0CN	0xC0	SMBus Control	188
SMB0DAT	0xC2	SMBus Data	192
SP	0x81	Stack Pointer	89
SPI0CFG	0xA1	SPI0 Configuration	174
SPI0CKR	0xA2	SPI0 Clock Rate Control	176
SPI0CN	0xF8	SPI0 Control	175
SPI0DAT	0xA3	SPI0 Data	176
TCON	0x88	Timer/Counter Control	215
TH0	0x8C	Timer/Counter 0 High	218
TH1	0x8D	Timer/Counter 1 High	218
TL0	0x8A	Timer/Counter 0 Low	217
TL1	0x8B	Timer/Counter 1 Low	217
TMOD	0x89	Timer/Counter Mode	216
TMR2CN	0xC8	Timer/Counter 2 Control	222
TMR2H	0xCD	Timer/Counter 2 High	224
TMR2L	0xCC	Timer/Counter 2 Low	224
TMR2RLH	0xCB	Timer/Counter 2 Reload High	223
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	223
VDM0CN	0xFF	VDD Monitor Control	126
XBR0	0xE1	Port I/O Crossbar Control 0	148
XBR1	0xE2	Port I/O Crossbar Control 1	149
All other SFR Locations		Reserved	

C8051F80x-83x

18.1. MCU Interrupt Sources and Vectors

The C8051F80x-83x MCUs support 15 interrupt sources. Software can simulate an interrupt by setting an interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 18.1. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

18.1.1. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 18.1.

18.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

C8051F80x-83x

18.3. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ External Interrupts

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “28.1. Timer 0 and Timer 1” on page 211) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 18.7). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “23.3. Priority Crossbar Decoder” on page 143 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

C8051F80x-83x

8. Restore previous interrupt state.

Steps 4–6 must be repeated for each 512-byte page to be erased.

Note: Flash security settings may prevent erasure of some Flash pages, such as the reserved area and the page containing the lock bytes. For a summary of Flash security settings and restrictions affecting Flash erase operations, please see Section “19.3. Security Options” on page 114.

19.1.3. Flash Write Procedure

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.**

The recommended procedure for writing a single byte in Flash is as follows:

1. Save current interrupt state and disable interrupts.
2. Ensure that the Flash byte has been erased (has a value of 0xFF).
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
8. Clear the PSWE bit.
9. Restore previous interrupt state.

Steps 5–7 must be repeated for each byte to be written.

Note: Flash security settings may prevent writes to some areas of Flash, such as the reserved area. For a summary of Flash security settings and restrictions affecting Flash write operations, please see Section “19.3. Security Options” on page 114.

19.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction.

Note: MOVX read instructions always target XRAM.

19.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, and erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock all Flash pages, starting at page 0, by writing a non-0xFF value to the lock byte. **Note that writing a non-0xFF value to the lock byte will lock all pages of FLASH from reads, writes, and erases, including the page containing the lock byte.**

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 19.1 summarizes the Flash security

features of the C8051F80x-83x devices.

Table 19.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	FEDR	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	FEDR	FEDR
Erase page containing Lock Byte—Unlock all pages (if any page is locked)	Only by C2DE	FEDR	FEDR
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	FEDR	FEDR
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	FEDR	FEDR
Read, Write or Erase Reserved Area	Not Permitted	FEDR	FEDR
<p>C2DE—C2 Device Erase (Erases all Flash pages including the page containing the Lock Byte) FEDR—Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is 1 after reset)</p> <ul style="list-style-type: none"> ■ All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset). ■ Locking any Flash page also locks the page containing the Lock Byte. ■ Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase. ■ If user code writes to the Lock Byte, the Lock does not take effect until the next device reset. 			

19.4. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of VDD, system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

To help prevent the accidental modification of Flash by firmware, the VDD Monitor must be enabled and enabled as a reset source on C8051F80x-83x devices for the Flash to be successfully modified. **If either the VDD Monitor or the VDD Monitor reset source is not enabled, a Flash Error Device Reset will be generated when the firmware attempts to modify the Flash.**

22.3.3. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 22.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation according to Equation 22.2, where f = the frequency of oscillation in MHz, C = the capacitor value in pF, and V_{DD} = the MCU power supply in volts.

Equation 22.2. C Mode Oscillator Frequency

$$f = (KF)/(R \times V_{DD})$$

For example: Assume $V_{DD} = 3.0$ V and $f = 150$ kHz:

$$f = KF / (C \times V_{DD})$$

$$0.150 \text{ MHz} = KF / (C \times 3.0)$$

Since the frequency of roughly 150 kHz is desired, select the K Factor from the table in SFR Definition 22.4 (OSCXCN) as $KF = 22$:

$$0.150 \text{ MHz} = 22 / (C \times 3.0)$$

$$C \times 3.0 = 22 / 0.150 \text{ MHz}$$

$$C = 146.6 / 3.0 \text{ pF} = 48.8 \text{ pF}$$

Therefore, the XFCN value to use in this example is 011b and $C = 50$ pF.

23.1. Port I/O Modes of Operation

Port pins P0.0–P1.7 use the Port I/O cell shown in Figure 23.2. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the PnMDIN and PnMDOUT registers. Port pin P2.0 can be configured by software for digital I/O using the P2MDOUT register. On reset, all Port I/O cells default to a high impedance state with weak pull-ups enabled. Until the crossbar is enabled (XBARE = 1), both the high and low port I/O drive circuits are explicitly disabled on all crossbar pins.

23.1.1. Port Pins Configured for Analog I/O

Any pins to be used as Comparator or ADC input, Capacitive Sense input, external oscillator input/output, VREF output, or AGND connection should be configured for analog I/O (PnMDIN.n = 0, Pn.n = 1). When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. To prevent the low port I/O drive circuit from pulling the pin low, a '1' should be written to the corresponding port latch (Pn.n = 1). Port pins configured for analog I/O will always read back a value of 0 regardless of the actual voltage on the pin.

Configuring pins as analog I/O saves power and isolates the Port pin from digital interference. Port pins configured as digital I/O may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

23.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SPI, SMBus, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PnMDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = 1) drive the Port pad to the VDD or GND supply rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high and low drivers turned off) when the output logic value is 1.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the Port pad to the VDD supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to GND to minimize power consumption and may be globally disabled by setting WEAKPUD to 1. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the Port pad, regardless of the output logic value of the Port pin.

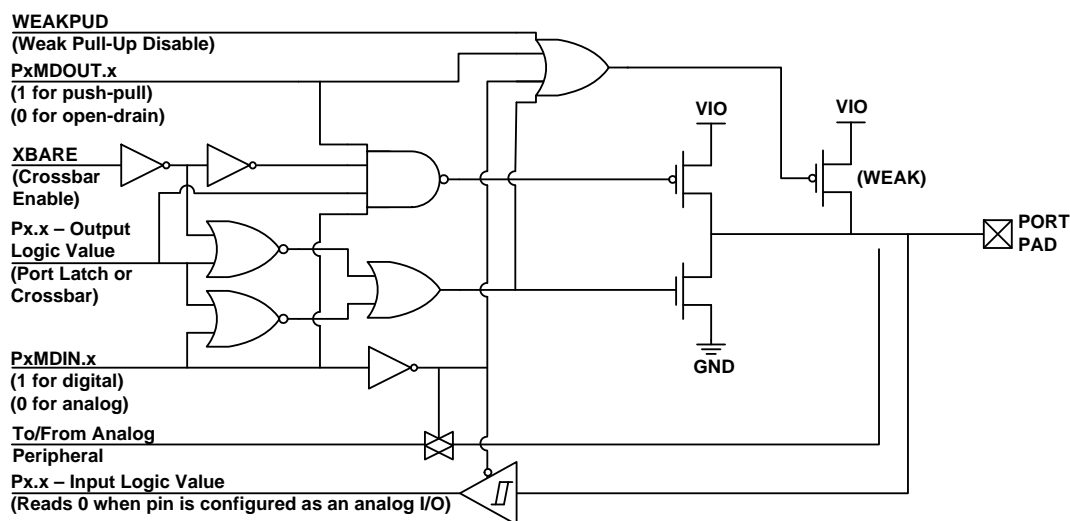


Figure 23.2. Port I/O Cell Block Diagram

C8051F80x-83x

Each Port has a corresponding PnSKIP register which allows its individual Port pins to be assigned to digital functions or skipped by the Crossbar. All Port pins used for analog functions or GPIO should have their PnSKIP bit set to 1.

The Port input mode of the I/O pins is defined using the Port Input Mode registers (PnMDIN). Each Port cell can be configured for analog or digital I/O. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is P2.0, which can only be used for digital I/O.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings.

SFR Definition 23.7. P0: Port 0

Bit	7	6	5	4	3	2	1	0
Name	P0[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x80; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P0[7:0]	Port 0 Data. Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P0.n Port pin is logic LOW. 1: P0.n Port pin is logic HIGH.

SFR Definition 26.2. SMB0CN: SMBus Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI
Type	R	R	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC0; Bit-Addressable

Bit	Name	Description	Read	Write
7	MASTER	SMBus Master/Slave Indicator. This read-only bit indicates when the SMBus is operating as a master.	0: SMBus operating in slave mode. 1: SMBus operating in master mode.	N/A
6	TXMODE	SMBus Transmit Mode Indicator. This read-only bit indicates when the SMBus is operating as a transmitter.	0: SMBus in Receiver Mode. 1: SMBus in Transmitter Mode.	N/A
5	STA	SMBus Start Flag.	0: No Start or repeated Start detected. 1: Start or repeated Start detected.	0: No Start generated. 1: When Configured as a Master, initiates a START or repeated START.
4	STO	SMBus Stop Flag.	0: No Stop condition detected. 1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).	0: No STOP condition is transmitted. 1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle. Cleared by Hardware.
3	ACKRQ	SMBus Acknowledge Request.	0: No Ack requested 1: ACK requested	N/A
2	ARBLOST	SMBus Arbitration Lost Indicator.	0: No arbitration error. 1: Arbitration Lost	N/A
1	ACK	SMBus Acknowledge.	0: NACK received. 1: ACK received.	0: Send NACK 1: Send ACK
0	SI	SMBus Interrupt Flag. This bit is set by hardware under the conditions listed in Table 15.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.	0: No interrupt pending 1: Interrupt Pending	0: Clear interrupt, and initiate next state machine event. 1: Force interrupt.

C8051F80x-83x

Table 26.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)

Mode	Values Read			Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected	
	Status Vector	ACKRQ	ARBLOST			ACK	STA	STO		ACK
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
				Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X	1000		
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						Send NACK to indicate last byte, and send STOP.	0	1	0	—
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

**Table 27.1. Timer Settings for Standard Baud Rates
Using The Internal 24.5 MHz Oscillator**

Frequency: 24.5 MHz							
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select) ¹	T1M ¹	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX ²	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in Section 28.1.
2. X = Don't care.

**Table 27.2. Timer Settings for Standard Baud Rates
Using an External 22.1184 MHz Oscillator**

Frequency: 22.1184 MHz							
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select) ¹	T1M ¹	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	96	SYSCLK	XX ²	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK / 12	00	0	0xE0
	14400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	9600	0.00%	2304	SYSCLK / 12	00	0	0xA0
	2400	0.00%	9216	SYSCLK / 48	10	0	0xA0
	1200	0.00%	18432	SYSCLK / 48	10	0	0x40
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:

1. SCA1–SCA0 and T1M bit definitions can be found in Section 28.1.
2. X = Don't care.

set is then given (in PCA clocks) by Equation 29.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL2) + (256 - PCA0L)$$

Equation 29.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH2 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF2 flag (PCA0CN.2) while the WDT is enabled.

29.4.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

1. Disable the WDT by writing a 0 to the WDTE bit.
2. Select the desired PCA clock source (with the CPS2–CPS0 bits).
3. Load PCA0CPL2 with the desired WDT update offset value.
4. Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
5. Enable the WDT by setting the WDTE bit to 1.
6. Reset the WDT timer by writing to PCA0CPH2.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL2 defaults to 0x00. Using Equation 29.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 29.3 lists some example timeout intervals for typical system clocks.

Table 29.3. Watchdog Timer Timeout Intervals¹

System Clock (Hz)	PCA0CPL2	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500 ²	255	257
3,062,500 ²	128	129.5
3,062,500 ²	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168
Notes:		
1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.		
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 8.		

29.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

30. C2 Interface

C8051F80x-83x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface operates using only two pins: a bi-directional data signal (C2D), and a clock input (C2CK). See the C2 Interface Specification for details on the C2 protocol.

30.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 30.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function																																	
7:0	C2ADD[7:0]	<p>C2 Address.</p> <p>The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.</p> <table border="1"> <thead> <tr> <th>Address</th> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>DEVICEID</td> <td>Selects the Device ID Register (read only)</td> </tr> <tr> <td>0x01</td> <td>REVID</td> <td>Selects the Revision ID Register (read only)</td> </tr> <tr> <td>0x02</td> <td>FPCTL</td> <td>Selects the C2 Flash Programming Control Register</td> </tr> <tr> <td>0xBF</td> <td>FPDAT</td> <td>Selects the C2 Flash Data Register</td> </tr> <tr> <td>0xD2</td> <td>CRC0AUTO*</td> <td>Selects the CRC0AUTO Register</td> </tr> <tr> <td>0xD3</td> <td>CRC0CNT*</td> <td>Selects the CRC0CNT Register</td> </tr> <tr> <td>0xCE</td> <td>CRC0CN*</td> <td>Selects the CRC0CN Register</td> </tr> <tr> <td>0xDE</td> <td>CRC0DATA*</td> <td>Selects the CRC0DATA Register</td> </tr> <tr> <td>0xCF</td> <td>CRC0FLIP*</td> <td>Selects the CRC0FLIP Register</td> </tr> <tr> <td>0xDD</td> <td>CRC0IN*</td> <td>Selects the CRC0IN Register</td> </tr> </tbody> </table>	Address	Name	Description	0x00	DEVICEID	Selects the Device ID Register (read only)	0x01	REVID	Selects the Revision ID Register (read only)	0x02	FPCTL	Selects the C2 Flash Programming Control Register	0xBF	FPDAT	Selects the C2 Flash Data Register	0xD2	CRC0AUTO*	Selects the CRC0AUTO Register	0xD3	CRC0CNT*	Selects the CRC0CNT Register	0xCE	CRC0CN*	Selects the CRC0CN Register	0xDE	CRC0DATA*	Selects the CRC0DATA Register	0xCF	CRC0FLIP*	Selects the CRC0FLIP Register	0xDD	CRC0IN*	Selects the CRC0IN Register
Address	Name	Description																																	
0x00	DEVICEID	Selects the Device ID Register (read only)																																	
0x01	REVID	Selects the Revision ID Register (read only)																																	
0x02	FPCTL	Selects the C2 Flash Programming Control Register																																	
0xBF	FPDAT	Selects the C2 Flash Data Register																																	
0xD2	CRC0AUTO*	Selects the CRC0AUTO Register																																	
0xD3	CRC0CNT*	Selects the CRC0CNT Register																																	
0xCE	CRC0CN*	Selects the CRC0CN Register																																	
0xDE	CRC0DATA*	Selects the CRC0DATA Register																																	
0xCF	CRC0FLIP*	Selects the CRC0FLIP Register																																	
0xDD	CRC0IN*	Selects the CRC0IN Register																																	
<p>*Note: CRC registers and functions are described in Section “24. Cyclic Redundancy Check Unit (CRC0)” on page 159.</p>																																			