

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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	8
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	10-VDFN Exposed Pad
Supplier Device Package	11-QFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f305

SFR Definition 5.2. ADC0CF: ADC0 Configuration (C8051F300/2)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	—	AMP0GN1	AMP0GN0	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

Bits7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in Table 5.1.

$$AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$$

Bit2: UNUSED. Read = 0b; Write = don't care.
Bits1–0: AMP0GN1–0: ADC0 Internal Amplifier Gain (PGA).
00: Gain = 0.5
01: Gain = 1
10: Gain = 2
11: Gain = 4

SFR Definition 5.3. ADC0: ADC0 Data Word (C8051F300/2)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBE

Bits7–0: ADC0 Data Word.
ADC0 holds the output data byte from the last ADC0 conversion. When in Single-ended mode, ADC0 holds an 8-bit unsigned integer. When in Differential mode, ADC0 holds a 2's complement signed 8-bit integer.

C8051F300/1/2/3/4/5

SFR Definition 7.2. CPT0MX: Comparator0 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	CMX0N1	CMX0N0	—	—	CMX0P1	CMX0P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

Bits7–6: UNUSED. Read = 00b, Write = don't care.

Bits6–4: CMX0N1–CMX0N0: Comparator0 Negative Input MUX Select.

These bits select which Port pin is used as the Comparator0 negative input.

CMX0N1	CMX0N0	Negative Input
0	0	P0.1
0	1	P0.3
1	0	P0.5
1	1	P0.7

Bits3–2: UNUSED. Read = 00b, Write = don't care.

Bits1–0: CMX0P1–CMX0P0: Comparator0 Positive Input MUX Select.

These bits select which Port pin is used as the Comparator0 positive input.

CMX0P1	CMX0P0	Positive Input
0	0	P0.0
0	1	P0.2
1	0	P0.4
1	1	P0.6

SFR Definition 7.3. CPT0MD: Comparator0 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	CP0MD1	CP0MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bits7–2: UNUSED. Read = 000000b, Write = don't care.

Bits1–0: CP0MD1–CP0MD0: Comparator0 Mode Select.

These bits select the response time for Comparator0.

Mode	CP0MD1	CP0MD0	CP0 Response Time (TYP)
0	0	0	Fastest Response Time
1	0	1	—
2	1	0	—
3	1	1	Lowest Power Consumption

8.2. 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 CIP-51 memory organization is shown in Figure 8.2 and Figure 8.3.

8.2.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F300/1/2/3 implements 8192 bytes of this program memory space as in-system, reprogrammable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x1FFF. Note: 512 bytes (0x1E00 - 0x1FFF) of this memory are reserved for factory use and are not available for user program storage. The C8051F304 implements 4096 bytes of reprogrammable Flash program memory space; the C8051F305 implements 2048 bytes of reprogrammable Flash program memory space. Figure 8.2 shows the program memory maps for C8051F300/1/2/3/4/5 devices.

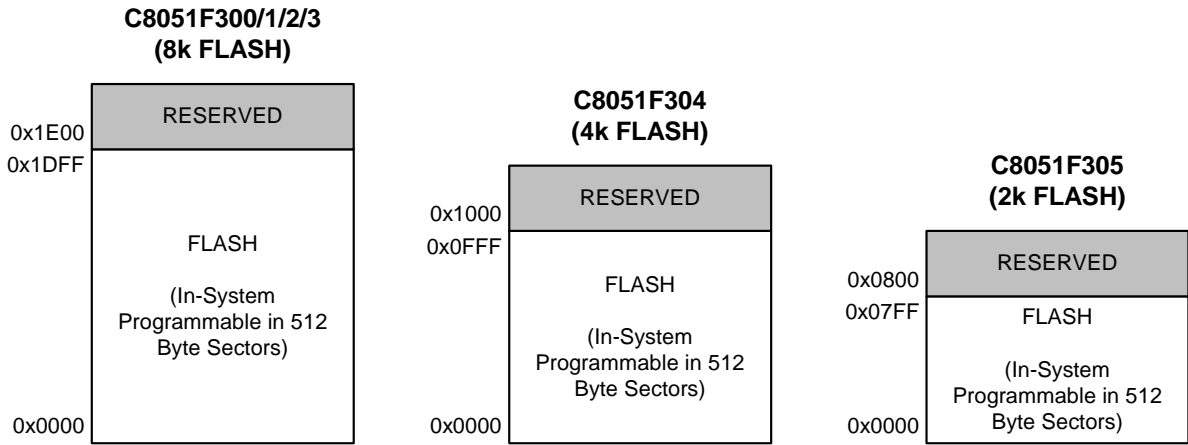


Figure 8.2. Program Memory Maps

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section “10. Flash Memory” on page 89** for further details.

8.2.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV     C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

8.2.5. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP, 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

8.2.6. 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 CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the subsystems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 8.2 lists the SFRs implemented in the CIP-51 System Controller.

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 datasheet, as indicated in Table 8.3, for a detailed description of each register.

Table 8.3. Special Function Registers* (Continued)

Register	Address	Description	Page No.
FLSCL	0xB6	Flash Scale	93
IE	0xA8	Interrupt Enable	75
IP	0xB8	Interrupt Priority	76
IT01CF	0xE4	INT0/INT1 Configuration Register	79
OSCICL	0xB3	Internal Oscillator Calibration	98
OSICN	0xB2	Internal Oscillator Control	98
OSCXCN	0xB1	External Oscillator Control	100
P0	0x80	Port 0 Latch	109
P0MDIN	0xF1	Port 0 Input Mode Configuration	109
P0MDOUT	0xA4	Port 0 Output Mode Configuration	110
PCA0CN	0xD8	PCA Control	167
PCA0MD	0xD9	PCA Mode	168
PCA0CPH0	0xFC	PCA Capture 0 High	171
PCA0CPH1	0xEA	PCA Capture 1 High	171
PCA0CPH2	0xEC	PCA Capture 2 High	171
PCA0CPL0	0xFB	PCA Capture 0 Low	171
PCA0CPL1	0xE9	PCA Capture 1 Low	171
PCA0CPL2	0xEB	PCA Capture 2 Low	171
PCA0CPM0	0xDA	PCA Module 0 Mode Register	169
PCA0CPM1	0xDB	PCA Module 1 Mode Register	169
PCA0CPM2	0xDC	PCA Module 2 Mode Register	169
PCA0H	0xFA	PCA Counter High	170
PCA0L	0xF9	PCA Counter Low	170
PCON	0x87	Power Control	81
PSCTL	0x8F	Program Store R/W Control	92
PSW	0xD0	Program Status Word	70
REF0CN	0xD1	Voltage Reference Control	49
RSTSRC	0xEF	Reset Source Configuration/Status	87
SBUF0	0x99	UART 0 Data Buffer	137
SCON0	0x98	UART 0 Control	136
SMB0CF	0xC1	SMBus Configuration	118
SMB0CN	0xC0	SMBus Control	120
SMB0DAT	0xC2	SMBus Data	122
SP	0x81	Stack Pointer	69
TMR2CN	0xC8	Timer/Counter 2 Control	154
TCON	0x88	Timer/Counter Control	147
TH0	0x8C	Timer/Counter 0 High	150

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

Table 8.3. Special Function Registers* (Continued)

Register	Address	Description	Page No.
TH1	0x8D	Timer/Counter 1 High	150
TL0	0x8A	Timer/Counter 0 Low	150
TL1	0x8B	Timer/Counter 1 Low	150
TMOD	0x89	Timer/Counter Mode	148
TMR2RLH	0xCB	Timer/Counter 2 Reload High	154
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	154
TMR2H	0xCD	Timer/Counter 2 High	154
TMR2L	0xCC	Timer/Counter 2 Low	154
XBR0	0xE1	Port I/O Crossbar Control 0	107
XBR1	0xE2	Port I/O Crossbar Control 1	107
XBR2	0xE3	Port I/O Crossbar Control 2	108
0x97, 0xAE, 0xAF, 0xB4, 0xB6, 0xBF, 0xCE, 0xD2, 0xD3, 0xD4, 0xD5, 0xD6, 0xD7, 0xDD, 0xDE, 0xDF, 0xF5		Reserved	
*Note: SFRs are listed in alphabetical order. All undefined SFR locations are reserved			

8.2.7. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic 1. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

SFR Definition 8.1. DPL: Data Pointer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x82
<p>Bits7–0: DPL: Data Pointer Low. The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory.</p>								

8.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 12 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE-EIE1). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Note: Any instruction that clears the EA bit should be immediately followed by an instruction that has two or more opcode bytes. For example:

```
// in 'C':
EA = 0;      // clear EA bit
EA = 0;      // ... followed by another 2-byte opcode

; in assembly:
CLR  EA      ; clear EA bit
CLR  EA      ; ... followed by another 2-byte opcode
```

If an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction which clears the EA bit), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. However, a read of the EA bit will return a '0' inside the interrupt service routine. When the "CLR EA" opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will reenter the ISR after the completion of the next instruction.

8.3.1. MCU Interrupt Sources and Vectors

The MCUs support 12 interrupt sources. Software can simulate an interrupt by setting any 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 8.4 on page 74. 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).

8.3.5. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described below. 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).

SFR Definition 8.7. IE: Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	IEGF0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
						(bit addressable)		0xA8
Bit7:	EA: Enable All Interrupts. This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.							
Bit6:	IEGF0: General Purpose Flag 0. This is a general purpose flag for use under software control.							
Bit5:	ET2: Enable Timer 2 Interrupt. This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.							
Bit4:	ES0: Enable UART0 Interrupt. This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.							
Bit3:	ET1: Enable Timer 1 Interrupt. This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.							
Bit2:	EX1: Enable External Interrupt 1. This bit sets the masking of external interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the /INT1 input.							
Bit1:	ET0: Enable Timer 0 Interrupt. This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.							
Bit0:	EX0: Enable External Interrupt 0. This bit sets the masking of external interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the /INT0 input.							

C8051F300/1/2/3/4/5

Accessing Flash **from user firmware** executing from an **unlocked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Locked pages cannot be read, written, or erased. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
3. **The page containing the Lock Byte cannot be erased.** It may be read or written only if it is unlocked. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing '0's to '1's in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a Flash Error device reset.

Accessing Flash **from user firmware** executing from a **locked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Any locked page except the page containing the Lock Byte may be read, written, or erased. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
3. **The page containing the Lock Byte cannot be erased.** It may only be read or written. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing '0's to '1's in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a Flash Error device reset.

SFR Definition 10.1. PSCTL: Program Store R/W Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8F

Bits7–2: UNUSED: Read = 000000b, Write = don't care.

Bit1: PSEE: Program Store Erase Enable
Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter.
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit0: PSWE: Program Store Write Enable
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX instruction. The Flash location should be erased before writing data.
0: Writes to Flash program memory disabled.
1: Writes to Flash program memory enabled; the MOVX instruction targets Flash memory.

SFR Definition 11.1. OSCICL: Internal Oscillator Calibration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—								Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB3

Bit7: UNUSED. Read = 0. Write = don't care.

Bits 6–0: OSCICL: Internal Oscillator Calibration Register.

This register calibrates the internal oscillator period. The reset value for OSCICL defines the internal oscillator base frequency. On C8051F300/1 devices, the reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

SFR Definition 11.2. OSCICN: Internal Oscillator Control

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
—	—	—	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	00010100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

Bits7–5: UNUSED. Read = 000b, Write = don't care.

Bit4: IFRDY: Internal Oscillator Frequency Ready Flag.

0: Internal Oscillator is not running at programmed frequency.

1: Internal Oscillator is running at programmed frequency.

Bit3: CLKSL: System Clock Source Select Bit.

0: SYSCLK derived from the Internal Oscillator, and scaled as per the IFCN bits.

1: SYSCLK derived from the External Oscillator circuit.

Bit2: IOSCEN: Internal Oscillator Enable Bit.

0: Internal Oscillator Disabled.

1: Internal Oscillator Enabled.

Bits1–0: IFCN1-0: Internal Oscillator Frequency Control Bits.

00: SYSCLK derived from Internal Oscillator divided by 8.

01: SYSCLK derived from Internal Oscillator divided by 4.

10: SYSCLK derived from Internal Oscillator divided by 2.

11: SYSCLK derived from Internal Oscillator divided by 1.

SFR Definition 11.3. OSCXCN: External Oscillator Control

R	R/W	R/W	R/W	R	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCND2	XOSCND1	XOSCND0	—	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB1

- Bit7: XLVLD: Crystal Oscillator Valid Flag.
(Read only when XOSCND = 11x.)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.
- Bits6–4: XOSCND2-0: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode.
011: External CMOS Clock Mode with divide by 2 stage.
100: RC Oscillator Mode with divide by 2 stage.
101: Capacitor Oscillator Mode with divide by 2 stage.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.
- Bit3: RESERVED. Read = 0, Write = don't care.
- Bits2–0: XFCN2-0: External Oscillator Frequency Control Bits.
000-111: See table below:

XFCN	Crystal (XOSCND = 11x)	RC (XOSCND = 10x)	C (XOSCND = 10x)
000	$f \leq 32 \text{ kHz}$	$f \leq 25 \text{ kHz}$	K Factor = 0.87
001	$32 \text{ kHz} < f \leq 84 \text{ kHz}$	$25 \text{ kHz} < f \leq 50 \text{ kHz}$	K Factor = 2.6
010	$84 \text{ kHz} < f \leq 225 \text{ kHz}$	$50 \text{ kHz} < f \leq 100 \text{ kHz}$	K Factor = 7.7
011	$225 \text{ kHz} < f \leq 590 \text{ kHz}$	$100 \text{ kHz} < f \leq 200 \text{ kHz}$	K Factor = 22
100	$590 \text{ kHz} < f \leq 1.5 \text{ MHz}$	$200 \text{ kHz} < f \leq 400 \text{ kHz}$	K Factor = 65
101	$1.5 \text{ MHz} < f \leq 4 \text{ MHz}$	$400 \text{ kHz} < f \leq 800 \text{ kHz}$	K Factor = 180
110	$4 \text{ MHz} < f \leq 10 \text{ MHz}$	$800 \text{ kHz} < f \leq 1.6 \text{ MHz}$	K Factor = 664
111	$10 \text{ MHz} < f \leq 30 \text{ MHz}$	$1.6 \text{ MHz} < f \leq 3.2 \text{ MHz}$	K Factor = 1590

CRYSTAL MODE (Circuit from Figure 11.1, Option 1; XOSCND = 11x)
Choose XFCN value to match crystal frequency.

RC MODE (Circuit from Figure 11.1, Option 2; XOSCND = 10x)
Choose XFCN value to match frequency range:
 $f = 1.23(10^3) / (R \times C)$, where
f = frequency of oscillation in MHz
C = capacitor value in pF
R = Pull-up resistor value in k Ω

C MODE (Circuit from Figure 11.1, Option 3; XOSCND = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
 $f = KF / (C \times V_{DD})$, where
f = frequency of oscillation in MHz
C = capacitor value the XTAL2 pin in pF
V_{DD} = Power Supply on MCU in volts

C8051F300/1/2/3/4/5

11.5. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 11.1, Option 2. 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, first select the RC network value to produce the desired frequency of oscillation. If the frequency desired is 100 kHz, let $R = 246 \text{ k}\Omega$ and $C = 50 \text{ pF}$:

$$f = 1.23 (10^3) / RC = 1.23 (10^3) / [246 \times 50] = 0.1 \text{ MHz} = 100 \text{ kHz}$$

Referring to the table in SFR Definition 11.3, the required XFCN setting is 010b.

11.6. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 11.1, 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 from the equations below. Assume $V_{DD} = 3.0 \text{ V}$ and $f = 150 \text{ kHz}$:

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

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

Since the frequency of roughly 150 kHz is desired, select the K Factor from the table in SFR Definition 11.3 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 $C = 50 \text{ pF}$.

SFR Definition 12.1. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	XSKP6	XSKP5	XSKP4	XSKP3	XSKP2	XSKP1	XSKP0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE1

Bit7: UNUSED. Read = 0b; Write = don't care.

Bits6–0: XSKP[6:0]: Crossbar Skip Enable Bits

These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (VREF input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.

0: Corresponding P0.n pin is not skipped by the Crossbar.

1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 12.2. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PCA0ME	CP0AOEN	CP0OEN	SYSCKE	SMB0OEN	URX0EN	UTX0EN		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bits7–6: PCA0ME: PCA Module I/O Enable Bits

00: All PCA I/O unavailable at Port pins.

01: CEX0 routed to Port pin.

10: CEX0, CEX1 routed to Port pins.

11: CEX0, CEX1, CEX2 routed to Port pins.

Bit5: CP0AOEN: Comparator0 Asynchronous Output Enable

0: Asynchronous CP0 unavailable at Port pin.

1: Asynchronous CP0 routed to Port pin.

Bit4: CP0OEN: Comparator0 Output Enable

0: CP0 unavailable at Port pin.

1: CP0 routed to Port pin.

Bit3: SYSCKE: /SYSCLK Output Enable

0: /SYSCLK unavailable at Port pin.

1: /SYSCLK output routed to Port pin.

Bit2: SMB0OEN: SMBus I/O Enable

0: SMBus I/O unavailable at Port pins.

1: SDA, SCL routed to Port pins.

Bit1: URX0EN: UART RX Enable

0: UART RX0 unavailable at Port pin.

1: UART RX0 routed to Port pin P0.5.

Bit0: UTX0EN: UART TX Output Enable

0: UART TX0 unavailable at Port pin.

1: UART TX0 routed to Port pin P0.4.

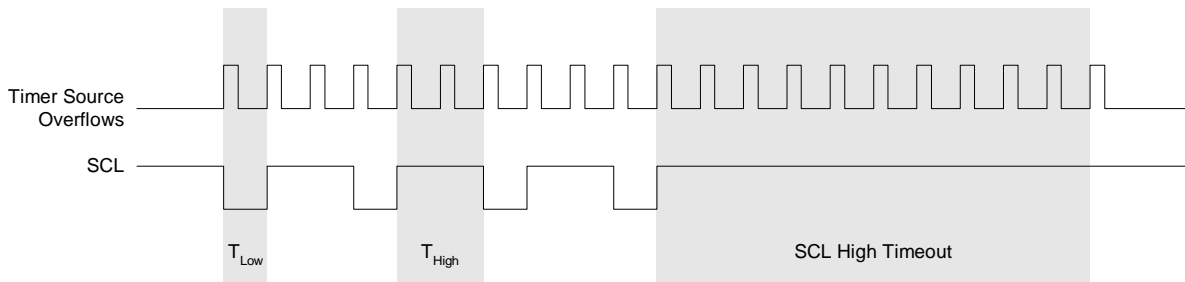


Figure 13.4. Typical SMBus SCL Generation

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 13.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCCLK is above 10 MHz.

Table 13.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	$T_{Low} - 4$ system clocks OR 1 system clock + s/w delay*	3 system clocks
1	11 system clocks	12 system clocks

***Note:** Setup Time for ACK bit transmissions and the MSB of all data transfers. The s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.

With the SMBTOE bit set, Timer 2 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see **Section “13.3.3. SCL Low Timeout” on page 114**). The SMBus interface will force Timer 2 to reload while SCL is high, and allow Timer 2 to count when SCL is low. The Timer 2 interrupt service routine should be used to reset SMBus communication by disabling and reenabling the SMBus. Timer 2 configuration is described in **Section “15.2. Timer 2” on page 151**.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 13.4). When a Free Timeout is detected, the interface will respond as if a STOP was detected (an interrupt will be generated, and STO will be set).

13.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 13.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER and TXMODE indicate the master/slave state and transmit/receive modes, respectively.

The STA bit indicates that a START has been detected or generated since the last SMBus interrupt. When set to '1', the STA bit will cause the SMBus to enter Master mode and generate a START when the bus becomes free. STA is not cleared by hardware after the START is generated; it must be cleared by software.

As a master, writing the STO bit will cause the hardware to generate a STOP condition and end the current transfer after the next ACK cycle. STO is cleared by hardware after the STOP condition is generated. As a slave, STO indicates that a STOP condition has been detected since the last SMBus interrupt. STO is also used in slave mode to manage the transition from slave receiver to slave transmitter; see **Section 13.5.4** for details on this procedure.

If STO and STA are both set to '1' (while in Master Mode), a STOP followed by a START will be generated.

As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 13.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

Table 13.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 13.4 for SMBus status decoding using the SMB0CN register.

Table 13.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	<ul style="list-style-type: none"> • A START is generated. 	<ul style="list-style-type: none"> • A STOP is generated. • Arbitration is lost.
TXMODE	<ul style="list-style-type: none"> • START is generated. • The SMBus interface enters transmitter mode (after SMB0DAT is written before the start of an SMBus frame). 	<ul style="list-style-type: none"> • A START is detected. • Arbitration is lost. • SMB0DAT is not written before the start of an SMBus frame.
STA	<ul style="list-style-type: none"> • A START followed by an address byte is received. 	<ul style="list-style-type: none"> • Must be cleared by software.
STO	<ul style="list-style-type: none"> • A STOP is detected while addressed as a slave. • Arbitration is lost due to a detected STOP. 	<ul style="list-style-type: none"> • A pending STOP is generated.
ACKRQ	<ul style="list-style-type: none"> • A byte has been received and an ACK response value is needed. 	<ul style="list-style-type: none"> • After each ACK cycle.
ARBLOST	<ul style="list-style-type: none"> • A repeated START is detected as a MASTER when STA is low (unwanted repeated START). • SCL is sensed low while attempting to generate a STOP or repeated START condition. • SDA is sensed low while transmitting a '1' (excluding ACK bits). 	<ul style="list-style-type: none"> • Each time SI is cleared.
ACK	<ul style="list-style-type: none"> • The incoming ACK value is low (ACKNOWLEDGE). 	<ul style="list-style-type: none"> • The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	<ul style="list-style-type: none"> • A START has been generated. • Lost arbitration. • A byte has been transmitted and an ACK/NACK received. • A byte has been received. • A START or repeated START followed by a slave address + R/W has been received. • A STOP has been received. 	<ul style="list-style-type: none"> • Must be cleared by software.

13.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames; however, note that the interrupt is generated before the ACK cycle when operating as a receiver, and after the ACK cycle when operating as a transmitter.

13.5.1. Master Transmitter Mode

Serial data is transmitted on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 13.5 shows a typical Master Transmitter sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that the ‘data byte transferred’ interrupts occur **after** the ACK cycle in this mode.

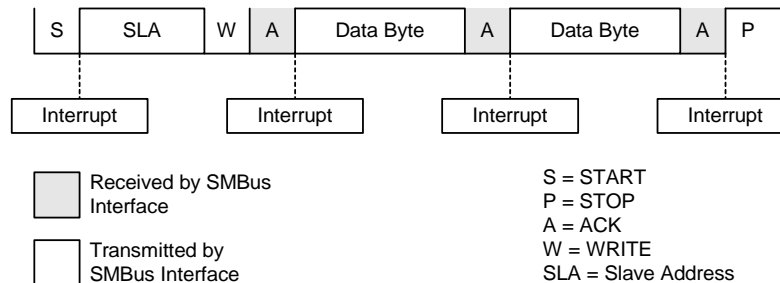


Figure 13.5. Typical Master Transmitter Sequence

14. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in **Section “14.1. Enhanced Baud Rate Generation” on page 132**). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. Reading SBUF0 accesses the buffered Receive register; writing SBUF0 accesses the Transmit register.

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

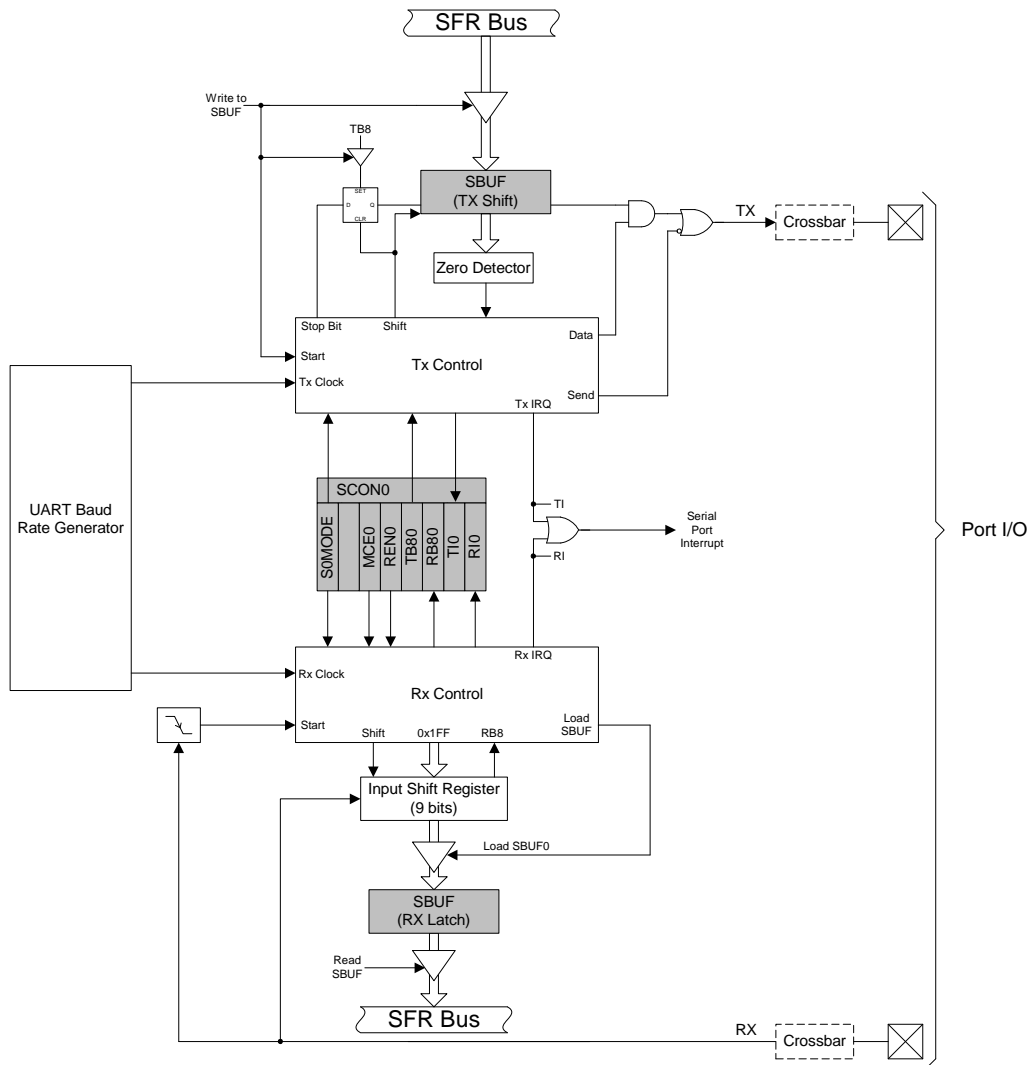


Figure 14.1. UART0 Block Diagram

16.2. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-bit Pulse Width Modulator, or 16-bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

Table 16.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1. See Figure 16.3 for details on the PCA interrupt configuration.

Table 16.2. PCA0CPM Register Settings for PCA Capture/Compare Modules

PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Operation Mode
X*	X*	1	0	0	0	0	X*	Capture triggered by positive edge on CEXn
X*	X*	0	1	0	0	0	X*	Capture triggered by negative edge on CEXn
X*	X*	1	1	0	0	0	X*	Capture triggered by transition on CEXn
X*	1	0	0	1	0	0	X*	Software Timer
X*	1	0	0	1	1	0	X*	High Speed Output
X*	1	0	0	X*	1	1	X*	Frequency Output
0	1	0	0	X*	0	1	X*	8-bit Pulse Width Modulator
1	1	0	0	X*	0	1	X*	16-bit Pulse Width Modulator

*Note: X = Don't Care

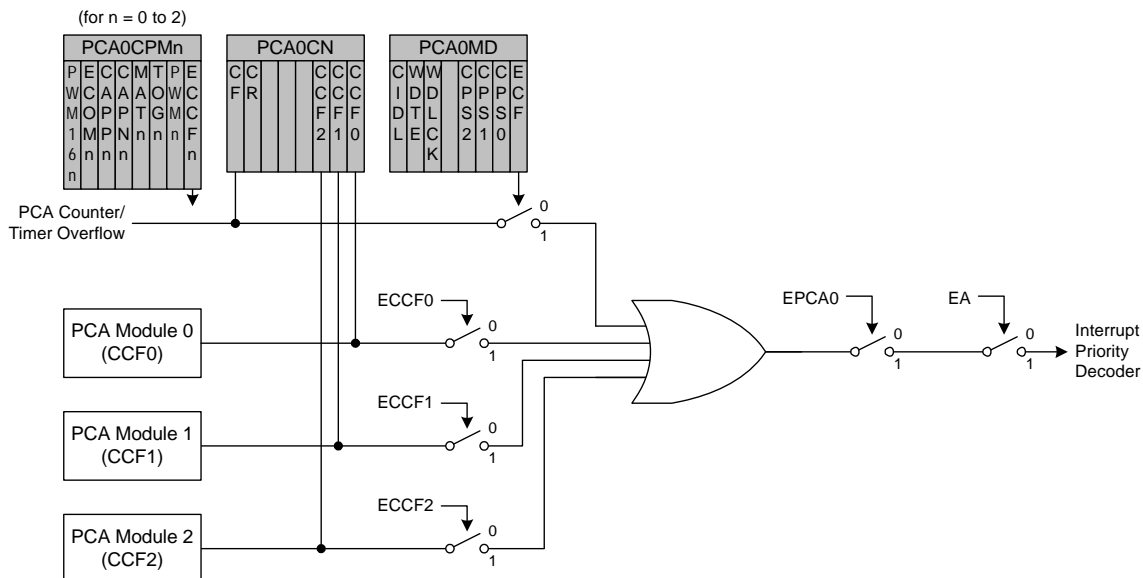


Figure 16.3. PCA Interrupt Block Diagram

17. C2 Interface

C8051F300/1/2/3/4/5 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.

17.1. 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 17.1. C2ADD: C2 Address

								Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

Address	Description
0x00	Selects the Device ID register for Data Read instructions
0x01	Selects the Revision ID register for Data Read instructions
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions
0x80	Selects the Port0 register for Data Read/Write instructions
0xF1	Selects the Port0 Input Mode register for Data Read/Write instructions
0xA4	Selects the Port0 Output Mode register for Data Read/Write instructions

C2 Register Definition 17.2. DEVICEID: C2 Device ID

								Reset Value
								00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

This read-only register returns the 8-bit device ID: 0x04 (C8051F300/1/2/3/4/5).