Welcome to 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	20MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f011

Figure 5.7. ADC0CN: ADC Control Register (C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADCEN	ADCTM	ADCINT	ADBUSY	ADSTM1	ADSTM0	ADWINT	ADLJST	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE8

Bit7: ADCEN: ADC Enable Bit
0: ADC Disabled. ADC is in low power shutdown.
1: ADC Enabled. ADC is active and ready for data conversions.

Bit6: ADCTM: ADC Track Mode Bit
0: When the ADC is enabled, tracking is always done unless a conversion is in process
1: Tracking Defined by ADSTM1-0 bits
ADSTM1-0:
00: Tracking starts with the write of 1 to ADBUSY and lasts for 3 SAR clocks
01: Tracking started by the overflow of Timer 3 and last for 3 SAR clocks
10: ADC tracks only when CNVSTR input is logic low
11: Tracking started by the overflow of Timer 2 and last for 3 SAR clocks

Bit5: ADCINT: ADC Conversion Complete Interrupt Flag
(Must be cleared by software)
0: ADC has not completed a data conversion since the last time this flag was cleared
1: ADC has completed a data conversion

Bit4: ADBUSY: ADC Busy Bit
Read
0: ADC Conversion complete or no valid data has been converted since a reset. The falling edge of ADBUSY generates an interrupt when enabled.
1: ADC Busy converting data
Write
0: No effect
1: Starts ADC Conversion if ADSTM1-0 = 00b

Bits3-2: ADSTM1-0: ADC Start of Conversion Mode Bits
00: ADC conversion started upon every write of 1 to ADBUSY
01: ADC conversions taken on every overflow of Timer 3
10: ADC conversion started upon every rising edge of CNVSTR
11: ADC conversions taken on every overflow of Timer 2

Bit1: ADWINT: ADC Window Compare Interrupt Flag
(Must be cleared by software)
0: ADC Window Comparison Data match has not occurred
1: ADC Window Comparison Data match occurred

Bit0: ADLJST: ADC Left Justify Data Bit
0: Data in ADC0H:ADC0L Registers is right justified
1: Data in ADC0H:ADC0L Registers is left justified

Table 10.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
ARITHMETIC OPERATIONS			
ADD A,Rn	Add register to A	1	1
ADD A,direct	Add direct byte to A	2	2
ADD A,@Ri	Add indirect RAM to A	1	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,direct	Add direct byte to A with carry	2	2
ADDC A,@Ri	Add indirect RAM to A with carry	1	2
ADDC A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal Adjust A	1	1
LOGICAL OPERATIONS			
ANL A,Rn	AND Register to A	1	1
ANL A,direct	AND direct byte to A	2	2
ANL A,@Ri	AND indirect RAM to A	1	2
ANL A,#data	AND immediate to A	2	2
ANL direct,A	AND A to direct byte	2	2
ANL direct,#data	AND immediate to direct byte	3	3
ORL A,Rn	OR Register to A	1	1
ORL A,direct	OR direct byte to A	2	2
ORL A,@Ri	OR indirect RAM to A	1	2
ORL A,#data	OR immediate to A	2	2
ORL direct,A	OR A to direct byte	2	2
ORL direct,#data	OR immediate to direct byte	3	3
XRL A,Rn	Exclusive-OR Register to A	1	1
XRL A,direct	Exclusive-OR direct byte to A	2	2
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL direct,A	Exclusive-OR A to direct byte	2	2
XRL direct,#data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1

Address	Register	Description	Page No.
0xC7	ADC0LTH	ADC Less-Than Data Word (High Byte)	36*, 47**
0xC6	ADC0LTL	ADC Less-Than Data Word (Low Byte)	36*, 47**
0xBA	AMX0CF	ADC MUX Configuration	31*, 42**
0xBB	AMX0SL	ADC MUX Channel Selection	32*, 43**
0xF0	B	B Register	76
0x8E	CKCON	Clock Control	144
0x9E	CPT0CN	Comparator 0 Control	56
0x9F	CPT1CN	Comparator 1 Control	58
0xD4	DAC0CN	DAC 0 Control	52
0xD3	DAC0H	DAC 0 Data Word (High Byte)	52
0xD2	DAC0L	DAC 0 Data Word (Low Byte)	52
0xD7	DAC1CN	DAC 1 Control	53
0xD6	DAC1H	DAC 1 Data Word (High Byte)	53
0xD5	DAC1L	DAC 1 Data Word (Low Byte)	53
0x83	DPH	Data Pointer (High Byte)	74
0x82	DPL	Data Pointer (Low Byte)	74
0xE6	EIE1	Extended Interrupt Enable 1	81
0xE7	EIE2	Extended Interrupt Enable 2	82
0xF6	EIP1	External Interrupt Priority 1	83
0xF7	EIP2	External Interrupt Priority 2	84
0xAF	EMI0CN	External Memory Interface Control	92***
0xB7	FLACL	Flash Access Limit	90***
0xB6	FLSCL	Flash Memory Timing Prescaler	91
0xA8	IE	Interrupt Enable	79
0xB8	IP	Interrupt Priority Control	80
0xB2	OSCICN	Internal Oscillator Control	100
0xB1	OSCXCN	External Oscillator Control	101
0x80	P0	Port 0 Latch	109
0x90	P1	Port 1 Latch	110
0xA0	P2	Port 2 Latch	111
0xB0	P3	Port 3 Latch	112
0xD8	PCA0CN	Programmable Counter Array 0 Control	160
0xFA	PCA0CPH0	PCA Capture Module 0 Data Word (High Byte)	163
0xFB	PCA0CPH1	PCA Capture Module 1 Data Word (High Byte)	163
0xFC	PCA0CPH2	PCA Capture Module 2 Data Word (High Byte)	163
0xFD	PCA0CPH3	PCA Capture Module 3 Data Word (High Byte)	163
0xFE	PCA0CPH4	PCA Capture Module 4 Data Word (High Byte)	163
0xEA	PCA0CPL0	PCA Capture Module 0 Data Word (Low Byte)	163
0xEB	PCA0CPL1	PCA Capture Module 1 Data Word (Low Byte)	163
0xEC	PCA0CPL2	PCA Capture Module 2 Data Word (Low Byte)	163
0xED	PCA0CPL3	PCA Capture Module 3 Data Word (Low Byte)	163

Address	Register	Description	Page No.
0x89	TMOD	Counter/Timer Mode	143
0x91	TMR3CN	Timer 3 Control	152
0x95	TMR3H	Timer 3 High	153
0x94	TMR3L	Timer 3 Low	153
0x93	TMR3RLH	Timer 3 Reload High	153
0x92	TMR3RLL	Timer 3 Reload Low	153
0xFF	WDTCN	Watchdog Timer Control	96
0xE1	XBR0	Port I/O Crossbar Configuration 1	105
0xE2	XBR1	Port I/O Crossbar Configuration 2	107
0xE3	XBR2	Port I/O Crossbar Configuration 3	108
0x84-86, 0x96-97, 0x9C, 0xA1-A3, 0xA9-AC, 0xAE, 0xB3-B5, 0xB9, 0xBD, 0xC9, 0xCE, 0xDF, 0xE4-E5, 0xF1-F5		Reserved	

* Refers to a register in the C8051F000/1/2/5/6/7 only.
** Refers to a register in the C8051F010/1/2/5/6/7 only.
*** Refers to a register in the C8051F005/06/07/15/16/17 only.

Figure 10.10. IP: Interrupt Priority

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PT2	PS	PT1	PX1	PT0	PX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xB8

Bits7-6: UNUSED. Read = 11b, Write = don't care.

Bit5: PT2 Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupts.
0: Timer 2 interrupts set to low priority level.
1: Timer 2 interrupts set to high priority level.

Bit4: PS: Serial Port (UART) Interrupt Priority Control.
This bit sets the priority of the Serial Port (UART) interrupts.
0: UART interrupts set to low priority level.
1: UART interrupts set to high priority level.

Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupts.
0: Timer 1 interrupts set to low priority level.
1: Timer 1 interrupts set to high priority level.

Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupts.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.

Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupts.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.

Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupts.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

11. FLASH MEMORY

These devices include 32k + 128 bytes of on-chip, reprogrammable Flash memory for program code and non-volatile data storage. The Flash memory can be programmed in-system, a single byte at a time, through the JTAG interface or by software using the MOVX instruction. Once cleared to 0, a Flash bit must be erased to set it back to 1. The bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution. Data polling to determine the end of the write/erase operation is not required. The Flash memory is designed to withstand at least 20,000 write/erase cycles. Refer to Table 11.1 for the electrical characteristics of the Flash memory.

11.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the JTAG interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the JTAG commands to program Flash memory, see Section 21.2.

The Flash memory can be programmed by software using the MOVX instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1. Writing to Flash remains enabled until the PSWE bit is cleared by software.

Writes to Flash memory can clear bits but cannot set them. Only an erase operation can set bits in Flash. Therefore, the byte location to be programmed must be erased before a new value can be written. The 32kbyte Flash memory is organized in 512-byte sectors. The erase operation applies to an entire sector (setting all bytes in the sector to 0xFF). Setting the PSEE Program Store Erase Enable bit (PSCTL.1) and PSWE (PSCTL.0) bit to logic 1 and then using the MOVX command to write a data byte to any byte location within the sector will erase an entire 512-byte sector. The data byte written can be of any value because it is not actually written to the Flash. Flash erasure remains enabled until the PSEE bit is cleared by software. The following sequence illustrates the algorithm for programming the Flash memory by software:

1. Enable Flash Memory write/erase in FLSCL Register using FLASCL bits.
2. Set PSEE (PSCTL.1) to enable Flash sector erase.
3. Set PSWE (PSCTL.0) to enable Flash writes.
4. Use MOVX to write a data byte to any location within the 512-byte sector to be erased.
5. Clear PSEE to disable Flash sector erase.
6. Use MOVX to write a data byte to the desired byte location within the erased 512-byte sector. Repeat until finished. (Any number of bytes can be written from a single byte to and entire sector.)
7. Clear the PSWE bit to disable Flash writes.

Write/Erase timing is automatically controlled by hardware based on the prescaler value held in the Flash Memory Timing Prescaler register (FLSCL). The 4-bit prescaler value FLASCL determines the time interval for write/erase operations. The FLASCL value required for a given system clock is shown in Figure 11.4, along with the formula used to derive the FLASCL values. When FLASCL is set to 1111b, the write/erase operations are disabled. Note that code execution in the 8051 is stalled while the Flash is being programmed or erased.

Table 11.1. FLASH Memory Electrical Characteristics

VDD = 2.7 to 3.6V, -40°C to +85°C unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Endurance		20k	100k		Erase/Wr
Erase Cycle Time		10			ms
Write Cycle Time		40			µs

prohibited from reading locations in the lower partition using the MOVC instruction. (Executing a MOVC instruction from the upper partition with a source address in the lower partition will always return a data value of 0x00.) Software running in the lower partition can access locations in both the upper and lower partition without restriction.

The Value-added firmware should be placed in the lower partition. On reset, control is passed to the value-added firmware via the reset vector. Once the value-added firmware completes its initial execution, it branches to a predetermined location in the upper partition. If entry points are published, software running in the upper partition may execute program code in the lower partition, but it cannot read the contents of the lower partition. Parameters may be passed to the program code running in the lower partition either through the typical method of placing them on the stack or in registers before the call or by placing them in prescribed memory locations in the upper partition.

The SRL address is specified using the contents of the Flash Access Register. The 16-bit SRL address is calculated as 0xNN00, where NN is the contents of the SRL Security Register. Thus, the SRL can be located on 256-byte boundaries anywhere in program memory space. However, the 512-byte erase sector size essentially requires that a 512 boundary be used. The contents of a non-initialized SRL security byte is 0x00, thereby setting the SRL address to 0x0000 and allowing read access to all locations in program memory space by default.

Figure 11.3. FLACL: Flash Access Limit (C8051F005/06/07/15/16/17 only)

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: 0xB7

Bits 7-0: FLACL: Flash Access Limit.
This register holds the high byte of the 16-bit program memory read/write/erase limit address. The entire 16-bit access limit address value is calculated as 0xNN00 where NN is replaced by contents of FLACL. A write to this register sets the Flash Access Limit. **This register can only be written once after any reset. Any subsequent writes are ignored until the next reset.**

Figure 13.4. RSTSRC: Reset Source Register

R	R/W	R/W	R/W	R	R	R/W	R	Reset Value
JTAGRST	CNVRSEF	C0RSEF	SWRSEF	WDTRSF	MCDRSF	PORSF	PINRSF	xxxxxxxx
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xEF

(Note: Do not use read-modify-write operations on this register.)

Bit7: JTAGRST. JTAG Reset Flag.
0: JTAG is not currently in reset state.
1: JTAG is in reset state.

Bit6: CNVRSEF: Convert Start Reset Source Enable and Flag
Write
0: CNVSTR is not a reset source
1: CNVSTR is a reset source (active low)
Read
0: Source of prior reset was not from CNVSTR
1: Source of prior reset was from CNVSTR

Bit5: C0RSEF: Comparator 0 Reset Enable and Flag
Write
0: Comparator 0 is not a reset source
1: Comparator 0 is a reset source (active low)
Read
Note: The value read from C0RSEF is not defined if Comparator 0 has not been enabled as a reset source.
0: Source of prior reset was not from Comparator 0
1: Source of prior reset was from Comparator 0

Bit4: SWRSF: Software Reset Force and Flag
Write
0: No Effect
1: Forces an internal reset. /RST pin is not effected.
Read
0: Prior reset source was not from write to the SWRSF bit.
1: Prior reset source was from write to the SWRSF bit.

Bit3: WDTRSF: Watchdog Timer Reset Flag
0: Source of prior reset was not from WDT timeout.
1: Source of prior reset was from WDT timeout.

Bit2: MCDRSF: Missing Clock Detector Flag
0: Source of prior reset was not from Missing Clock Detector timeout.
1: Source of prior reset was from Missing Clock Detector timeout.

Bit1: PORSF: Power-On Reset Force and Flag
Write
0: No effect
1: Forces a Power-On Reset. /RST is driven low.
Read
0: Source of prior reset was not from POR.
1: Source of prior reset was from POR.

Bit0: PINRSF: HW Pin Reset Flag
0: Source of prior reset was not from /RST pin.
1: Source of prior reset was from /RST pin.

Figure 14.2. OSCICN: Internal Oscillator Control Register

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
MSCLKE	-	-	IFRDY	CLKSL	IOSCEN	IFCN1	IFCN0	00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

Bit7: MSCLKE: Missing Clock Enable Bit
0: Missing Clock Detector Disabled
1: Missing Clock Detector Enabled; triggers a reset if a missing clock is detected
Bits6-5: UNUSED. Read = 00b, Write = don't care
Bit4: IFRDY: Internal Oscillator Frequency Ready Flag
0: Internal Oscillator Frequency not running at speed specified by the IFCN bits.
1: Internal Oscillator Frequency running at speed specified by the IFCN bits.
Bit3: CLKSL: System Clock Source Select Bit
0: Uses Internal Oscillator as System Clock.
1: Uses External Oscillator as System Clock.
Bit2: IOSCEN: Internal Oscillator Enable Bit
0: Internal Oscillator Disabled
1: Internal Oscillator Enabled
Bits1-0: IFCN1-0: Internal Oscillator Frequency Control Bits
00: Internal Oscillator typical frequency is 2MHz.
01: Internal Oscillator typical frequency is 4MHz.
10: Internal Oscillator typical frequency is 8MHz.
11: Internal Oscillator typical frequency is 16MHz.

Table 14.1. Internal Oscillator Electrical Characteristics

-40°C to +85°C unless otherwise specified.

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Internal Oscillator Frequency	OSCICN.[1:0] = 00 OSCICN.[1:0] = 01 OSCICN.[1:0] = 10 OSCICN.[1:0] = 11	1.5 3.1 6.2 12.3	2 4 8 16	2.4 4.8 9.6 19.2	MHz
Internal Oscillator Current Consumption (from VDD)	OSCICN.2 = 1		200		µA
Internal Oscillator Temperature Stability			4		ppm/°C
Internal Oscillator Power Supply (VDD) Stability			6.4		%/V

Figure 15.4. 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
SYSCKE	T2EXE	T2E	INT1E	T1E	INT0E	T0E	CP1OEN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bit7: SYSCKE: SYSCLK Output Enable Bit
0: SYSCLK unavailable at Port pin.
1: SYSCLK output routed to Port Pin.

Bit6: T2EXE: T2EX Enable Bit
0: T2EX unavailable at Port pin.
1: T2EX routed to Port Pin.

Bit5: T2E: T2 Enable Bit
0: T2 unavailable at Port pin.
1: T2 routed to Port Pin.

Bit4: INT1E: /INT1 Enable Bit
0: /INT1 unavailable at Port pin.
1: /INT1 routed to Port Pin.

Bit3: T1E: T1 Enable Bit
0: T1 unavailable at Port pin.
1: T1 routed to Port Pin.

Bit2: INT0E: /INT0 Enable Bit
0: /INT0 unavailable at Port pin.
1: /INT0 routed to Port Pin.

Bit1: T0E: T0 Enable Bit
0: T0 unavailable at Port pin.
1: T0 routed to Port Pin.

Bit0: CP1OEN: Comparator 1 Output Enable Bit
0: CP1 unavailable at Port pin.
1: CP1 routed to Port Pin.

15.3. General Purpose Port I/O

Each MCU has four byte-wide, bi-directional parallel ports that can be used general purpose I/O. Each port is accessed through a corresponding special function register (SFR) that is both byte addressable and bit addressable. When writing to a port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the port's input pins are returned regardless of the XBRn settings (i.e. even when the pin is assigned to another signal by the Crossbar, the Port Register can always still read its corresponding Port I/O pin). The exception to this is the execution of the *read-modify-write* instructions. The *read-modify-write* instructions when operating on a port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SET, when the destination is an individual bit in a port SFR. For these instructions, the value of the port register (not the pin) is read, modified, and written back to the SFR.

15.4. Configuring Ports Which are not Pinned Out

P2 and P3 are not pinned out on the F001/06/11/16. P1, P2, and P3 are not pinned out on the F002/07/12/17. These port registers (and corresponding interrupts, where applicable) are still available for software use in these reduced pin count MCUs. Whether used or not in software, it is recommended not to let these port drivers go to high impedance state. This is prevented after reset by having the weak pull-ups enabled as described in the XBR2 register. It is recommended that each output driver for ports not pinned out should be configured as push-pull using the corresponding PRTnCF register. This will inhibit a high impedance state even if the weak pull-up is disabled.

Figure 15.6. P0: Port0 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0x80

Bits7-0: P0.[7:0]
(Write – Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (high-impedance if corresponding PRT0CF.n bit = 0)
(Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P0.n pin is logic low.
1: P0.n pin is logic high.

Figure 15.7. PRT0CF: Port0 Configuration Register

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: 0xA4

Bits7-0: PRT0CF.[7:0]: Output Configuration Bits for P0.7-P0.0 (respectively)
0: Corresponding P0.n Output mode is Open-Drain.
1: Corresponding P0.n Output mode is Push-Pull.

(Note: When SDA, SCL, and RX appear on any of the P0 I/O, each are open-drain regardless of the value of PRT0CF).

16.2.4. Slave Receiver Mode

Serial data is received on SDA while the serial clock is received on SCL. First, a byte is received that contains an address and data direction bit. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. If the received address matches the slave's assigned address (or a general call address is received) one or more bytes of serial data are received from the master. After each byte is received, an acknowledge bit is transmitted by the slave. The master outputs START and STOP conditions to indicate the beginning and end of the serial transfer.

16.3. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remains high for a specified time. Two or more master devices may attempt to generate a START condition at the same time. Since the devices that generated the START condition may not be aware that other masters are contending for the bus, an arbitration scheme is employed. The master devices continue to transmit until one of the masters transmits a HIGH level, while the other(s) master transmits a LOW level on SDA. The first master(s) transmitting the HIGH level on SDA looses the arbitration and is required to give up the bus.

16.4. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave can hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

16.5. Timeouts

16.5.1. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25ms as a "timeout" condition. Devices that have detected the timeout condition must reset the communication no later than 10ms after detecting the timeout condition.

One of the MCU's general-purpose timers, operating in 16-bit auto-reload mode, can be used to monitor the SCL line for this timeout condition. Timer 3 is specifically designed for this purpose. (Refer to the Timer 3 Section 19.3. for detailed information on Timer 3 operation.)

16.5.2. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if a device holds the SCL and SDA lines high for more that 50usec, the bus is designated as free. The SMB0CR register is used to detect this condition when the FTE bit in SMB0CN is set.

16.6. SMBus Special Function Registers

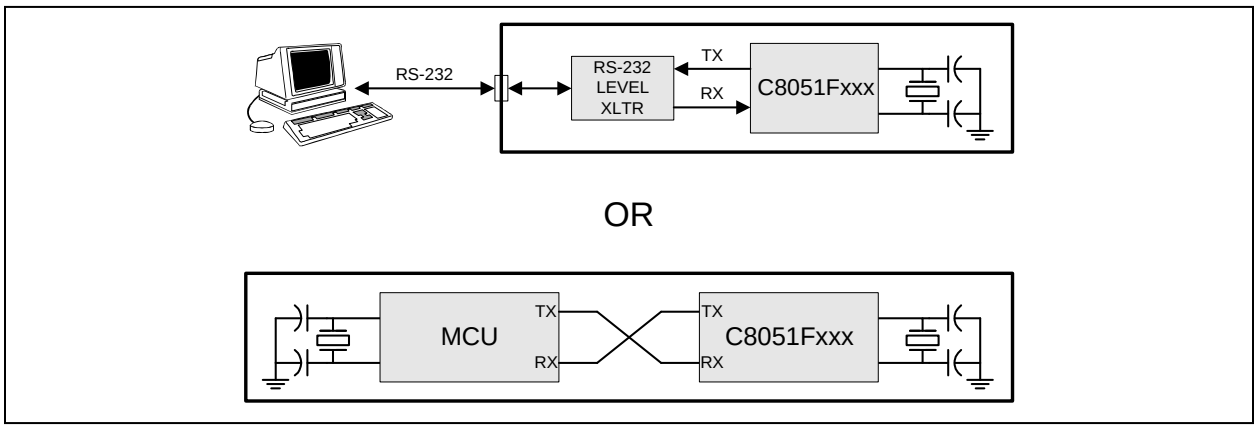
The SMBus serial interface is accessed and controlled through five SFRs: SMB0CN Control Register, SMB0CR Clock Rate Register, SMB0ADR Address Register, SMB0DAT Data Register and SMB0STA Status Register. The system device may have one or more SMBus serial interfaces implemented. The five special function registers related to the operation of the SMBus interface are described in the following section.

The Timer 2 overflow rate, when in *Baud Rate Generator Mode* and using an internal clock source, is determined solely by the Timer 2 16-bit reload value (RCAP2H:RCAP2L). The Timer 2 clock source is fixed at SYSCLK/2. The Timer 2 overflow rate can be calculated as follows:

T2_OVERFLOWRATE = (SYSCLK/2) / (65536 – [RCAP2H:RCAP2L]).

Timer 2 can be selected as the baud rate generator for RX and/or TX by setting RCLK (T2CON.5) and/or TCLK (T2CON.4), respectively. When either RCLK or TCLK is set to logic 1, Timer 2 interrupts are automatically disabled and the timer is forced into *Baud Rate Generator Mode* with SYSCLK/2 as its clock source. If a different timebase is required, setting the C/T2 bit (T2CON.1) to logic 1 will allow Timer 2 to be clocked from the external input pin T2. See the Timers section for complete timer configuration details.

Figure 18.5. UART Modes 1, 2, and 3 Interconnect Diagram



18.1.3. Mode 2: 9-Bit UART, Fixed Baud Rate

Mode 2 provides asynchronous, full-duplex communication using a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (see timing diagram in Figure 18.6). On transmit, the ninth data bit is determined by the value in TB8 (SCON.3). It can be assigned the value of the parity flag P in the PSW or used in multiprocessor communications. On receive, the ninth data bit goes into RB8 (SCON.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF register. The TI Transmit Interrupt Flag (SCON.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit (SCON.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF receive register if the following conditions are met: RI must be logic 0, and if SM2 is logic 1, the 9th bit must be logic 1.

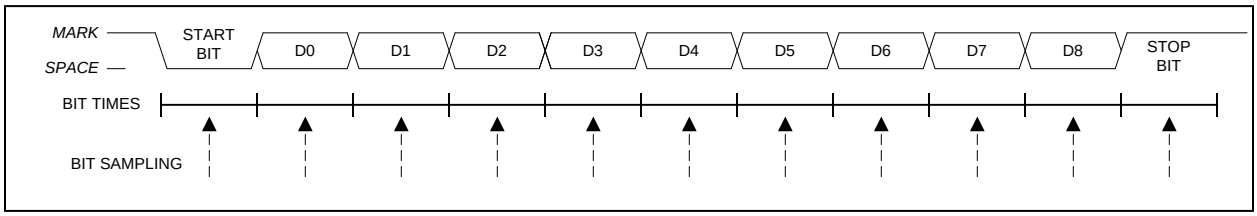
If these conditions are met, the eight bits of data are stored in SBUF, the ninth bit is stored in RB8 and the RI flag is set. If these conditions are not met, SBUF and RB8 will not be loaded and the RI flag will not be set. An interrupt will occur if enabled when either TI or RI are set.

The baud rate in Mode 2 is a direct function of the system clock frequency as follows:

$$\textit{Mode 2 Baud Rate} = 2^{SMOD} * (SYSCLK / 64).$$

The SMOD bit (PCON.7) selects whether to divide SYSCLK by 32 or 64. In the formula, 2 is raised to the power SMOD, resulting in a baud rate of either 1/32 or 1/64 of the system clock frequency. On reset, the SMOD bit is logic 0, thus selecting the lower speed baud rate by default.

Figure 18.6. UART Modes 2 and 3 Timing Diagram



18.1.4. Mode 3: 9-Bit UART, Variable Baud Rate

Mode 3 is the same as Mode 2 in all respects except the baud rate is variable. The baud rate is determined in the same manner as for Mode 1. Mode 3 operation transmits 11 bits: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. Timer 1 or Timer 2 overflows generate the baud rate just as with Mode 1. In summary, Mode 3 transmits using the same protocol as Mode 2 but with Mode 1 baud rate generation.

19. TIMERS

Each MCU implements four counter/timers: three are 16-bit counter/timers compatible with those found in the standard 8051, and one is a 16-bit timer for use with the ADC, SMBus, or for general purpose use. These can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 offers additional capabilities not available in Timers 0 and 1. Timer 3 is similar to Timer 2, but without the capture or Baud Rate Generator modes.

Timer 0 and Timer 1:	Timer 2:	Timer 3:
13-bit counter/timer	16-bit counter/timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer	16-bit counter/timer with capture	
8-bit counter/timer with auto-reload	Baud rate generator	
Two 8-bit counter/timers (Timer 0 only)		

When functioning as a timer, the counter/timer registers are incremented on each clock tick. Clock ticks are derived from the system clock divided by either one or twelve as specified by the Timer Clock Select bits (T2M-T0M) in CKCON. The twelve-clocks-per-tick option provides compatibility with the older generation of the 8051 family. Applications that require a faster timer can use the one-clock-per-tick option.

When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin for T0, T1, or T2. Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is sampled.

19.1. Timer 0 and Timer 1

Timer 0 and Timer 1 are accessed and controlled through SFRs. Each counter/timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control (TCON) register is used to enable Timer 0 and Timer 1 as well as indicate their status. Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits M1-M0 in the Counter/Timer Mode (TMOD) register. Each timer can be configured independently. Following is a detailed description of each operating mode.

19.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as a 13-bit counter/timer in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. Clearing C/T selects the system clock as the input for the timer. When C/T0 is set to logic 1, high-to-low transitions at the selected input pin increment the timer register. (Refer to Port I/O Section 15.1 for information on selecting and configuring external I/O pins.)

Figure 19.6. CKCON: Clock Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	T2M	T1M	T0M	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8E

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

Bit5: T2M: Timer 2 Clock Select.
This bit controls the division of the system clock supplied to Timer 2. This bit is ignored when the timer is in baud rate generator mode or counter mode (i.e. C/T2 = 1).
0: Timer 2 uses the system clock divided by 12.
1: Timer 2 uses the system clock.

Bit4: T1M: Timer 1 Clock Select.
This bit controls the division of the system clock supplied to Timer 1.
0: Timer 1 uses the system clock divided by 12.
1: Timer 1 uses the system clock.

Bit3: T0M: Timer 0 Clock Select.
This bit controls the division of the system clock supplied to Counter/Timer 0.
0: Counter/Timer uses the system clock divided by 12.
1: Counter/Timer uses the system clock.

Bits2-0: Reserved. Read = 000b, Must Write = 000.

Figure 19.7. TL0: Timer 0 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: 0x8A

Bits 7-0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

Figure 19.8. TL1: Timer 1 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: 0x8B

Bits 7-0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

Figure 19.9. TH0: Timer 0 High 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: 0x8C

Bits 7-0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

Figure 19.10. TH1: Timer 1 High Byte

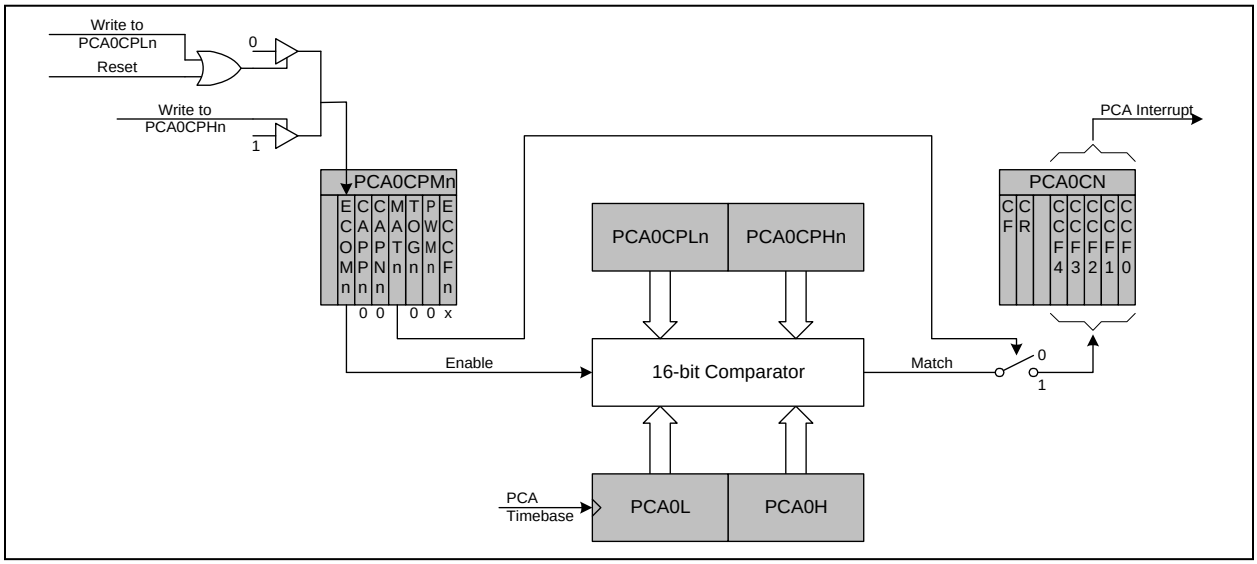
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x8D

Bits 7-0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

20.1.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Figure 20.4. PCA Software Timer Mode Diagram



20.1.3. High Speed Output Mode

In this mode, each time a match occurs between the PCA Timer Counter and a module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn) the logic level on the module's associated CEXn pin will toggle. Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

Figure 20.5. PCA High Speed Output Mode Diagram

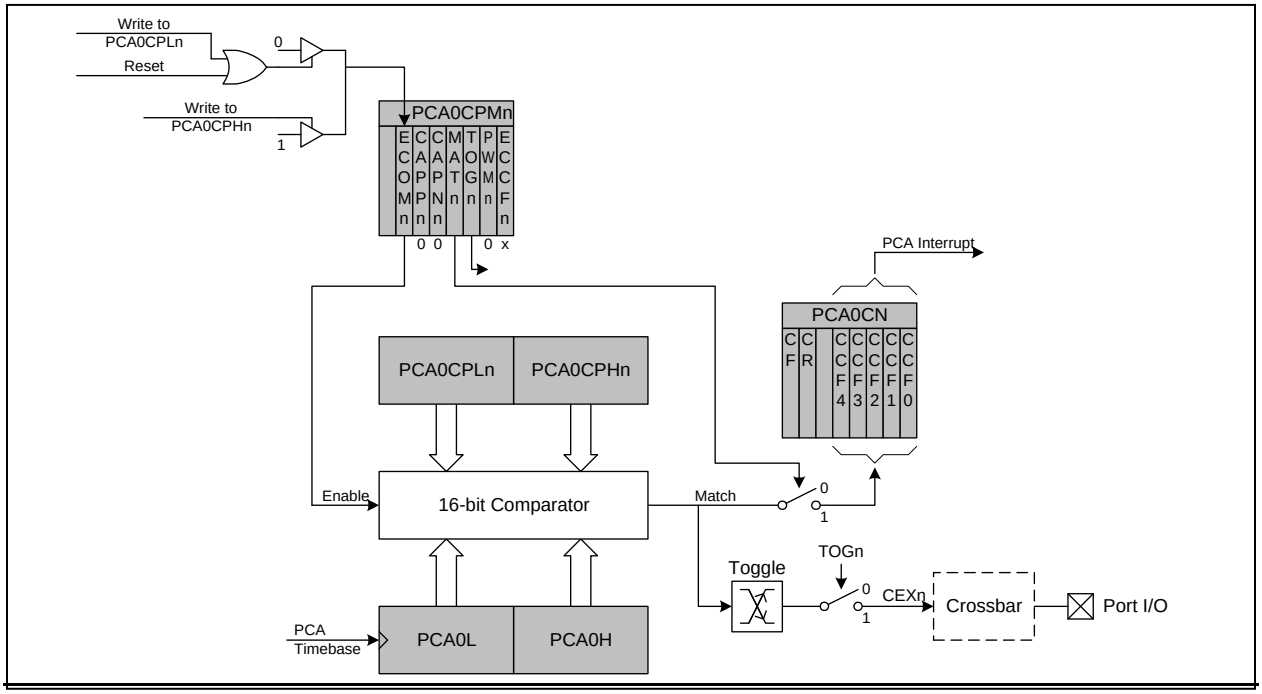


Figure 21.3. FLASHCON: JTAG Flash Control Register

WRMD3	WRMD2	WRMD1	WRMD0	RDMD3	RDMD2	RDMD1	RDMD0	Reset Value
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	00000000

This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

Bits7-4: WRMD3-0: Write Mode Select Bits.
The Write Mode Select Bits control how the interface logic responds to writes to the FLASHDAT Register per the following values:
0000: A FLASHDAT write replaces the data in the FLASHDAT register, but is otherwise ignored.
0001: A FLASHDAT write initiates a write of FLASHDAT into the memory location addressed by the FLASHADR register. FLASHADR is incremented by one when complete.
0010: A FLASHDAT write initiates an erasure (sets all bytes to 0xFF) of the Flash page containing the address in FLASHADR. FLASHDAT must be 0xA5 for the erase to occur. FLASHADR is not affected. If FLASHADR = 0x7DFE – 0x7DFF, the entire user space will be erased (i.e. entire Flash memory except for Reserved area 0x7E00 – 0x7FFF).
(All other values for WRMD3-0 are reserved.)

Bits3-0: RDMD3-0: Read Mode Select Bits.
The Read Mode Select Bits control how the interface logic responds to reads to the FLASHDAT Register per the following values:
0000: A FLASHDAT read provides the data in the FASHDAT register, but is otherwise ignored.
0001: A FLASHDAT read initiates a read of the byte addressed by the FLASHADR register if no operation is currently active. This mode is used for block reads.
0010: A FLASHDAT read initiates a read of the byte addressed by FLASHADR only if no operation is active and any data from a previous read has already been read from FLASHDAT. This mode allows single bytes to be read (or the last byte of a block) without initiating an extra read.
(All other values for RDMD3-0 are reserved.)

Figure 21.4. FLASHADR: JTAG Flash Address Register

																Reset Value
																0x0000
Bit15															Bit0	

This register holds the address for all JTAG Flash read, write, and erase operations. This register autoincrements after each read or write, regardless of whether the operation succeeded or failed.

Bits15-0: Flash Operation 16-bit Address.