Welcome to E-XFL.COM

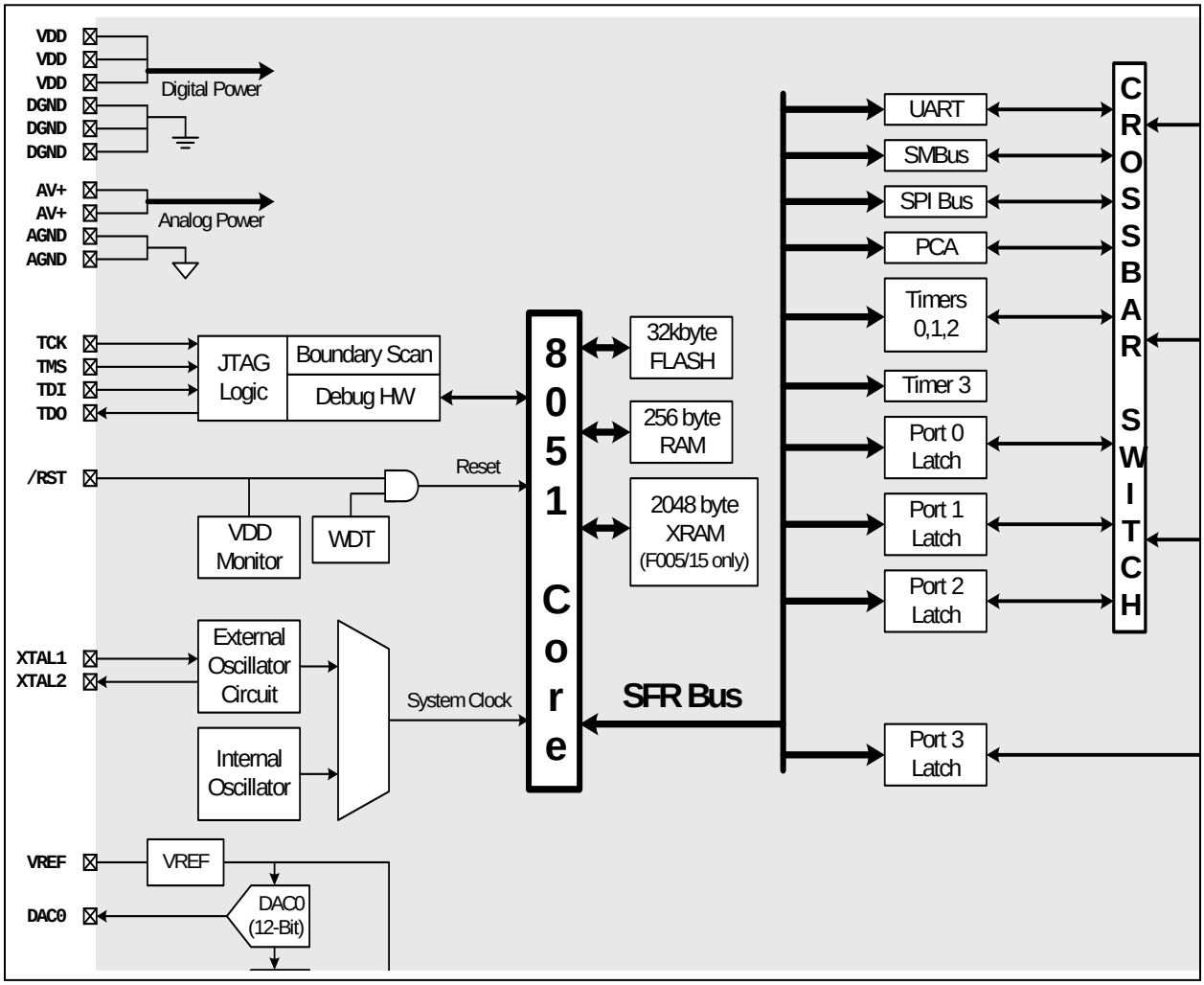
What is "Embedded - Microcontrollers"?

"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	25MHz
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	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x12b; 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/c8051f006r

Figure 1.1. C8051F000/05/10/15 Block Diagram



1.4. Programmable Digital I/O and Crossbar

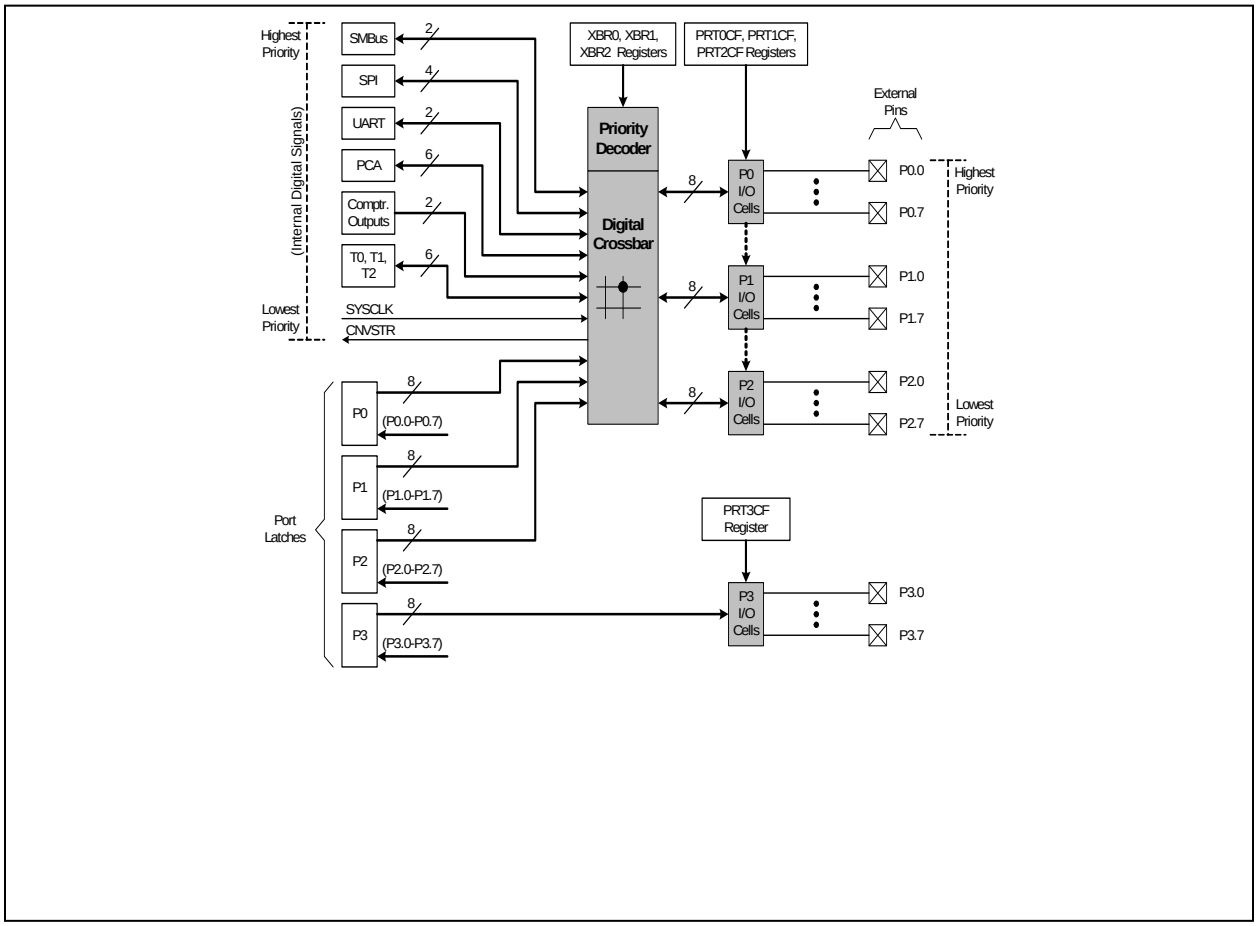
The standard 8051 Ports (0, 1, 2, and 3) are available on the MCUs. All four ports are pinned out on the F000/05/10/15. Ports 0 and 1 are pinned out on the F001/06/11/16, and only Port 0 is pinned out on the F002/07/12/17. The Ports not pinned out are still available for software use as general purpose registers. The Port I/O behave like the standard 8051 with a few enhancements.

Each Port I/O pin can be configured as either a push-pull or open-drain output. Also, the "weak pull-ups" which are normally fixed on an 8051 can be globally disabled, providing additional power saving capabilities for low power applications.

Perhaps the most unique enhancement is the Digital Crossbar. This is essentially a large digital switching network that allows mapping of internal digital system resources to Port I/O pins on P0, P1, and P2. (See Figure 1.8.) Unlike microcontrollers with standard multiplexed digital I/O, all combinations of functions are supported.

The on-board counter/timers, serial buses, HW interrupts, ADC Start of Conversion input, comparator outputs, and other digital signals in the controller can be configured to appear on the Port I/O pins specified in the Crossbar Control registers. This allows the user to select the exact mix of general purpose Port I/O and digital resources needed for his particular application.

Figure 1.8. Digital Crossbar Diagram



4. PINOUT AND PACKAGE DEFINITIONS

Table 4.1. Pin Definitions

Name	Pin Numbers			Type	Description
	F000 F005 F010 F015	F001 F006 F011 F016	F002 F007 F012 F017		
VDD	31, 40, 62	23, 32	18, 20		Digital Voltage Supply.
DGND	30, 41, 61	22, 33, 27, 19	17, 21		Digital Ground.
AV+	16, 17	13, 43	9, 29		Positive Analog Voltage Supply.
AGND	5, 15	44, 12	8, 30		Analog Ground.
TCK	22	18	14	D In	JTAG Test Clock with internal pull-up.
TMS	21	17	13	D In	JTAG Test-Mode Select with internal pull-up.
TDI	28	20	15	D In	JTAG Test Data Input with internal pull-up. TDI is latched on a rising edge of TCK.
TDO	29	21	16	D Out	JTAG Test Data Output with internal pull-up. Data is shifted out on TDO on the falling edge of TCK. TDO output is a tri-state driver.
XTAL1	18	14	10	A In	Crystal Input. This pin is the return for the internal oscillator circuit for a crystal or ceramic resonator. For a precision internal clock, connect a crystal or ceramic resonator from XTAL1 to XTAL2. If overdriven by an external CMOS clock, this becomes the system clock.
XTAL2	19	15	11	A Out	Crystal Output. This pin is the excitation driver for a crystal or ceramic resonator.
/RST	20	16	12	D I/O	Chip Reset. Open-drain output of internal Voltage Supply monitor. Is driven low when VDD is < 2.7V. An external source can force a system reset by driving this pin low.
VREF	6	3	3	A I/O	Voltage Reference. When configured as an input, this pin is the voltage reference for the MCU. Otherwise, the internal reference drives this pin.
CP0+	4	2	2	A In	Comparator 0 Non-Inverting Input.
CP0-	3	1	1	A In	Comparator 0 Inverting Input.
CP1+	2	45		A In	Comparator 1 Non-Inverting Input.
CP1-	1	46		A In	Comparator 1 Inverting Input.
DAC0	64	48	32	A Out	Digital to Analog Converter Output 0. The DAC0 voltage output. (See Section 7 DAC Specification for complete description).
DAC1	63	47	31	A Out	Digital to Analog Converter Output 1. The DAC1 voltage output. (See Section 7 DAC Specification for complete description).
AIN0	7	4	4	A In	Analog Mux Channel Input 0. (See ADC Specification for complete description).
AIN1	8	5	5	A In	Analog Mux Channel Input 1. (See ADC Specification for complete description).
AIN2	9	6	6	A In	Analog Mux Channel Input 2. (See ADC Specification for complete description).
AIN3	10	7	7	A In	Analog Mux Channel Input 3. (See ADC Specification for complete description).
AIN4	11	8		A In	Analog Mux Channel Input 4. (See ADC Specification for complete description).
AIN5	12	9		A In	Analog Mux Channel Input 5. (See ADC Specification for complete description).

6.2. ADC Modes of Operation

The ADC uses VREF to determine its full-scale voltage, thus the reference must be properly configured before performing a conversion (see Section 9). The ADC has a maximum conversion speed of 100ksps. The ADC conversion clock is derived from the system clock. Conversion clock speed can be reduced by a factor of 2, 4, 8 or 16 via the ADCSC bits in the ADC0CF Register. This is useful to adjust conversion speed to accommodate different system clock speeds.

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC Start of Conversion Mode bits (ADSTM1, ADSTM0) in ADC0CN. Conversions may be initiated by:

1. Writing a 1 to the ADBUSY bit of ADC0CN;
2. A Timer 3 overflow (i.e. timed continuous conversions);
3. A rising edge detected on the external ADC convert start signal, CNVSTR;
4. A Timer 2 overflow (i.e. timed continuous conversions).

Writing a 1 to ADBUSY provides software control of the ADC whereby conversions are performed "on-demand". During conversion, the ADBUSY bit is set to 1 and restored to 0 when conversion is complete. The falling edge of ADBUSY triggers an interrupt (when enabled) and sets the ADCINT interrupt flag. **Note: When conversions are performed "on-demand", the ADCINT flag, not ADBUSY, should be polled to determine when the conversion has completed.** Converted data is available in the ADC data word MSB and LSB registers, ADC0H, ADC0L. Converted data can be either left or right justified in the ADC0H:ADC0L register pair (see example in Figure 6.9) depending on the programmed state of the ADLJST bit in the ADC0CN register.

The ADCTM bit in register ADC0CN controls the ADC track-and-hold mode. In its default state, the ADC input is continuously tracked, except when a conversion is in progress. Setting ADCTM to 1 allows one of four different low power track-and-hold modes to be specified by states of the ADSTM1-0 bits (also in ADC0CN):

1. Tracking begins with a write of 1 to ADBUSY and lasts for 3 SAR clocks;
2. Tracking starts with an overflow of Timer 3 and lasts for 3 SAR clocks;
3. Tracking is active only when the CNVSTR input is low;
4. Tracking starts with an overflow of Timer 2 and lasts for 3 SAR clocks.

Modes 1, 2 and 4 (above) are useful when the start of conversion is triggered with a software command or when the ADC is operated continuously. Mode 3 is used when the start of conversion is triggered by external hardware. In this case, the track-and-hold is in its low power mode at times when the CNVSTR input is high. Tracking can also be disabled (shutdown) when the entire chip is in low power standby or sleep modes.

Figure 6.2. 10-Bit ADC Track and Conversion Example Timing

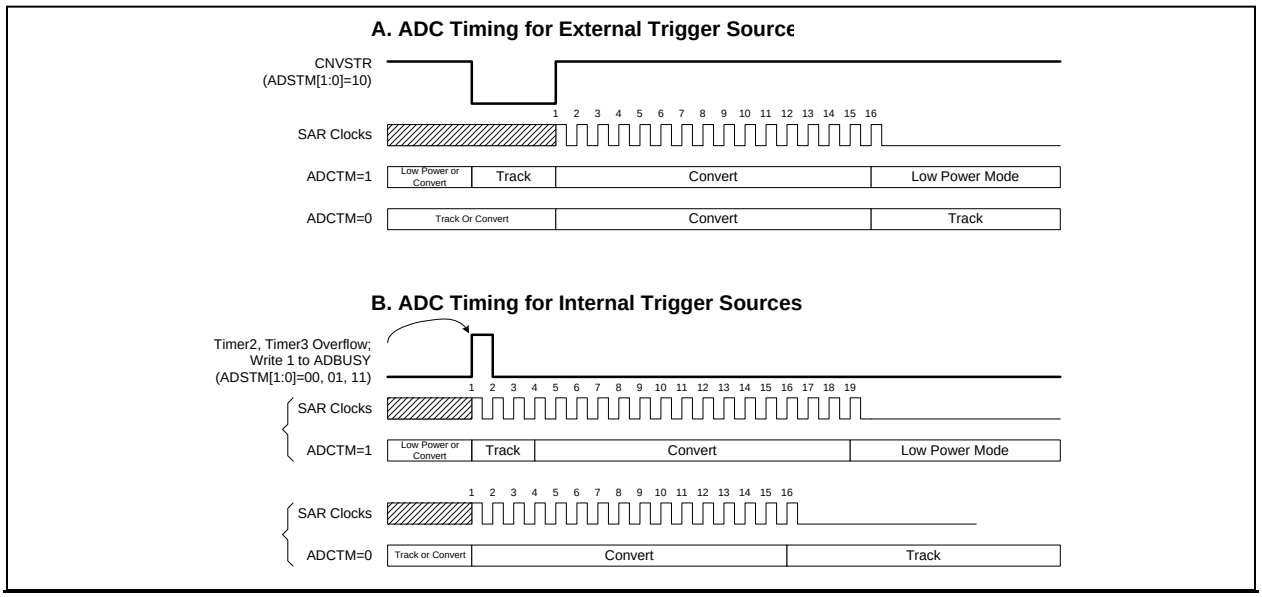


Figure 6.7. ADC0CN: ADC Control Register (C8051F01x)

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 (bit addressable)	SFR Address: 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

Figure 8.3. CPT0CN: Comparator 0 Control Register

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP1	CP0HYP0	CP0HYN1	CP0HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9E

Bit7: CP0EN: Comparator 0 Enable Bit
0: Comparator 0 Disabled.
1: Comparator 0 Enabled.

Bit6: CP0OUT: Comparator 0 Output State Flag
0: Voltage on CP0+ < CP0-
1: Voltage on CP0+ > CP0-

Bit5: CP0RIF: Comparator 0 Rising-Edge Interrupt Flag
0: No Comparator 0 Rising-Edge Interrupt has occurred since this flag was cleared
1: Comparator 0 Rising-Edge Interrupt has occurred since this flag was cleared

Bit4: CP0FIF: Comparator 0 Falling-Edge Interrupt Flag
0: No Comparator 0 Falling-Edge Interrupt has occurred since this flag was cleared
1: Comparator 0 Falling-Edge Interrupt has occurred since this flag was cleared

Bit3-2: CP0HYP1-0: Comparator 0 Positive Hysteresis Control Bits
00: Positive Hysteresis Disabled
01: Positive Hysteresis = 2mV
10: Positive Hysteresis = 4mV
11: Positive Hysteresis = 10mV

Bit1-0: CP0HYN1-0: Comparator 0 Negative Hysteresis Control Bits
00: Negative Hysteresis Disabled
01: Negative Hysteresis = 2mV
10: Negative Hysteresis = 4mV
11: Negative Hysteresis = 10mV

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

Mnemonic	Description	Bytes	Clock Cycles
PROGRAM BRANCHING			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3
JNZ rel	Jump if A does not equal zero	2	2/3
CJNE A,direct,rel	Compare direct byte to A and jump if not equal	3	3/4
CJNE A,#data,rel	Compare immediate to A and jump if not equal	3	3/4
CJNE Rn,#data,rel	Compare immediate to register and jump if not equal	3	3/4
CJNE @Ri,#data,rel	Compare immediate to indirect and jump if not equal	3	4/5
DJNZ Rn,rel	Decrement register and jump if not zero	2	2/3
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	3/4
NOP	No operation	1	1

Notes on Registers, Operands and Addressing Modes:

Rn - Register R0-R7 of the currently selected register bank.

@Ri - Data RAM location addressed indirectly through register R0-R1

rel - 8-bit, signed (two's compliment) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).

#data - 8-bit constant

#data 16 - 16-bit constant

bit - Direct-addressed bit in Data RAM or SFR.

addr 11 - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-byte page of program memory as the first byte of the following instruction.

addr 16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

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 10.7. ACC: Accumulator

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xE0

Bits 7-0: ACC: Accumulator
This register is the accumulator for arithmetic operations.

Figure 10.8. B: B Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xF0

Bits 7-0: B: B Register
This register serves as a second accumulator for certain arithmetic operations.

13. RESET SOURCES

The reset circuitry of the MCUs allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the CIP-51 halts program execution, forces the external port pins to a known state and initializes the SFRs to their defined reset values. Interrupts and timers are disabled. On exit, the program counter (PC) is reset, and program execution starts at location 0x0000.

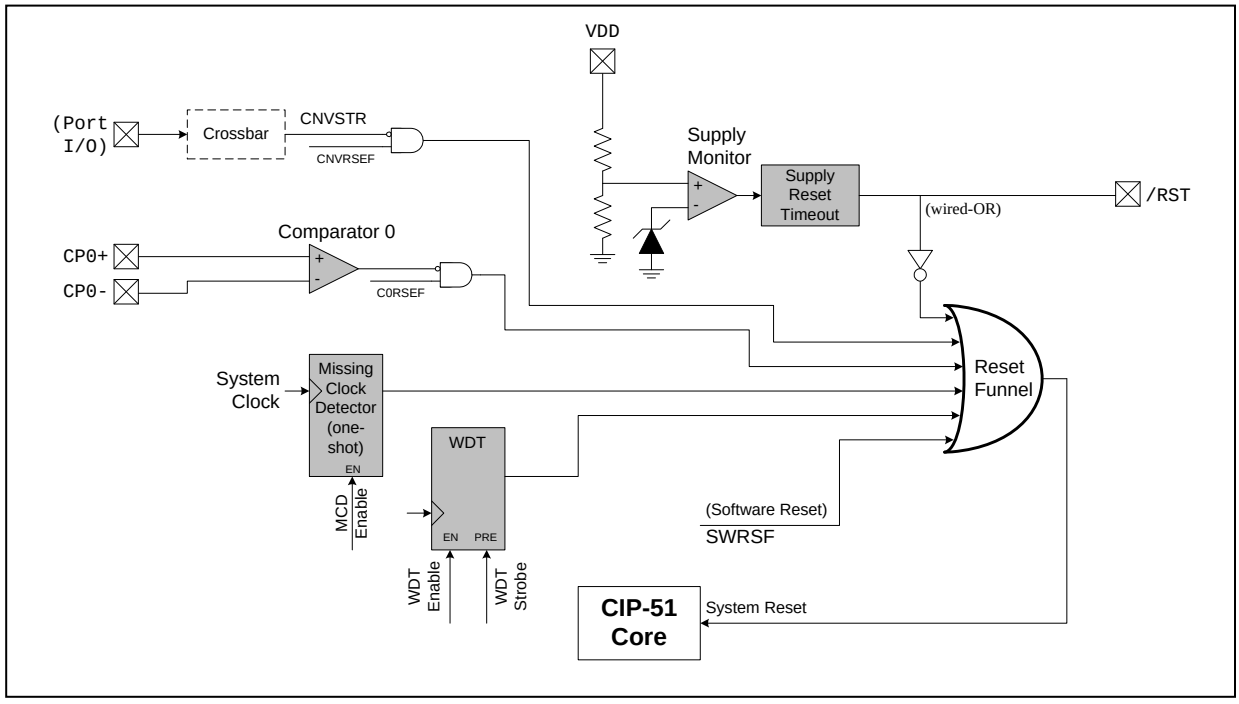
All of the SFRs are reset to predefined values. The reset values of the SFR bits are defined in the SFR detailed descriptions. The contents of internal data memory are not changed during a reset and any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost even though the data on the stack are not altered.

The I/O port latches are reset to 0xFF (all logic ones), activating internal weak pull-ups which take the external I/O pins to a high state. The weak pull-ups are enabled during and after the reset. If the source of reset is from the VDD Monitor or writing a 1 to PORSF, the /RST pin is driven low until the end of the VDD reset timeout.

On exit from the reset state, the MCU uses the internal oscillator running at 2MHz as the system clock by default. Refer to Section 14 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled using its longest timeout interval. (Section 13.8 details the use of the Watchdog Timer.)

There are seven sources for putting the MCU into the reset state: power-on/power-fail, external /RST pin, external CNVSTR signal, software commanded, Comparator 0, Missing Clock Detector, and Watchdog Timer. Each reset source is described below:

Figure 13.1. Reset Sources Diagram



14.1. External Crystal Example

If a crystal or ceramic resonator were used to generate the system clock for the MCU, the circuit would be as shown in Figure 14.1, Option 1. For an ECS-110.5-20-4 crystal, the resonate frequency is 11.0592MHz, the intrinsic capacitance is 7pF, and the ESR is 60Ω. The compensation capacitors should be 33pF each, and the PWB parasitic capacitance is estimated to be 2pF. The appropriate External Oscillator Frequency Control value (XFCN) from the Crystal column in the table in Figure 14.3 (OSCXCN Register) should be 111b.

Because the oscillator detect circuitry needs time to settle after the crystal oscillator is enabled, software should wait at least 1ms between enabling the crystal oscillator and polling the XTLVLD bit. The recommend procedure is:
1. Enable the external oscillator
2. Wait at least 1 ms
3. Poll for XTLVLD '0' ==> '1'
4. Switch to the external oscillator

Switching to the external oscillator before the crystal oscillator has stabilized could result in unpredictable behavior.

NOTE: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device, keeping the traces as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

14.2. External RC Example

If an external RC network were used to generate the system clock for the MCU, the circuit would be as shown in Figure 14.1, Option 2. The capacitor must be no greater than 100pF, but using a very small capacitor will increase the frequency drift due to the PWB parasitic capacitance. 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 100kHz, let R = 246kΩ and C = 50pF:

$$f = 1.23(10^3)/RC = 1.23(10^3) / [246 * 50] = 0.1MHz = 100kHz$$

$$XFCN \geq \log_2(f/25kHz)$$
$$XFCN \geq \log_2(100kHz/25kHz) = \log_2(4)$$
$$XFCN \geq 2, \text{ or code } 010$$

14.3. External Capacitor Example

If an external capacitor were used to generate the system clock for the MCU, the circuit would be as shown in Figure 14.1, Option 3. The capacitor must be no greater than 100pF, but using a very small capacitor will increase the frequency inaccuracy due to the PWB parasitic capacitance. 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 AV+ = 3.0V and C = 50pF:

$$f = KF / (C * VDD) = KF / (50 * 3)$$
$$f = KF / 150$$

If a frequency of roughly 90kHz is desired, select the K Factor from the table in Figure 14.3 as KF = 13:

$$f = 13 /150 = 0.087MHz, \text{ or } 87kHz$$

Therefore, the XFCN value to use in this example is 011.

Figure 15.11. P2: Port2 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xA0

Bits7-0: P2.[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 PRT2CF.n bit = 0)
(Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P2.n is logic low.
1: P2.n is logic high.

Figure 15.12. PRT2CF: Port2 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: 0xA6

Bits7-0: PRT2CF.[7:0]: Output Configuration Bits for P2.7-P2.0 (respectively)
0: Corresponding P2.n Output mode is Open-Drain.
1: Corresponding P2.n Output mode is Push-Pull.

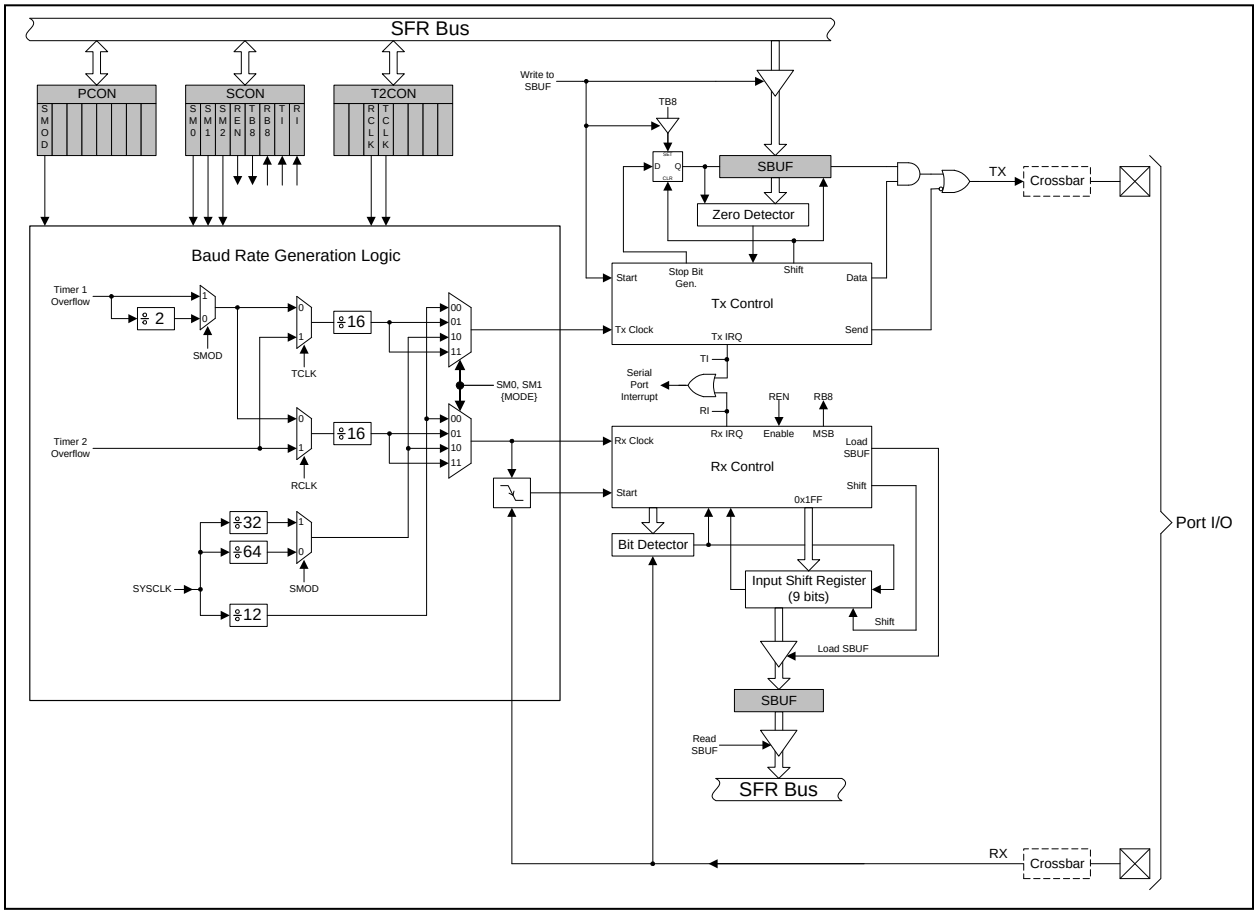
18. UART

The UART is a serial port capable of asynchronous transmission. The UART can function in full duplex mode. In all modes, receive data is buffered in a holding register. This allows the UART to start reception of a second incoming data byte before software has finished reading the previous data byte.

The UART has an associated Serial Control Register (SCON) and a Serial Data Buffer (SBUF) in the SFRs. The single SBUF location provides access to both transmit and receive registers. Reads access the Receive register and writes access the Transmit register automatically.

The UART is capable of generating interrupts if enabled. The UART has two sources of interrupts: a Transmit Interrupt flag, TI (SCON.1) set when transmission of a data byte is complete, and a Receive Interrupt flag, RI (SCON.0) set when reception of a data byte is complete. The UART interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software. This allows software to determine the cause of the UART interrupt (transmit complete or receive complete).

Figure 18.1. UART Block Diagram



18.1. UART Operational Modes

The UART provides four operating modes (one synchronous and three asynchronous) selected by setting configuration bits in the SCON register. These four modes offer different baud rates and communication protocols. The four modes are summarized in Table 18.1 below. Detailed descriptions follow.

Table 18.1. UART Modes

Mode	Synchronization	Baud Clock	Data Bits	Start/Stop Bits
0	Synchronous	SYSCLK/12	8	None
1	Asynchronous	Timer 1 or Timer 2 Overflow	8	1 Start, 1 Stop
2	Asynchronous	SYSCLK/32 or SYSCLK/64	9	1 Start, 1 Stop
3	Asynchronous	Timer 1 or Timer 2 Overflow	9	1 Start, 1 Stop

18.1.1. Mode 0: Synchronous Mode

Mode 0 provides synchronous, half-duplex communication. Serial data is transmitted and received on the RX pin. The TX pin provides the shift clock for both transmit and receive. The MCU must be the master since it generates the shift clock for transmission in both directions (see the interconnect diagram in Figure 18.2).

Eight data bits are transmitted/received, LSB first (see the timing diagram in Figure 18.3). 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 eighth bit time. Data reception begins when the REN Receive Enable bit (SCON.4) is set to logic 1 and the RI Receive Interrupt Flag (SCON.0) is cleared. One cycle after the eighth bit is shifted in, the RI flag is set and reception stops until software clears the RI bit. An interrupt will occur if enabled when either TI or RI is set.

The Mode 0 baud rate is the system clock frequency divided by twelve. RX is forced to open-drain in mode 0, and an external pull-up will typically be required.

Figure 18.2. UART Mode 0 Interconnect

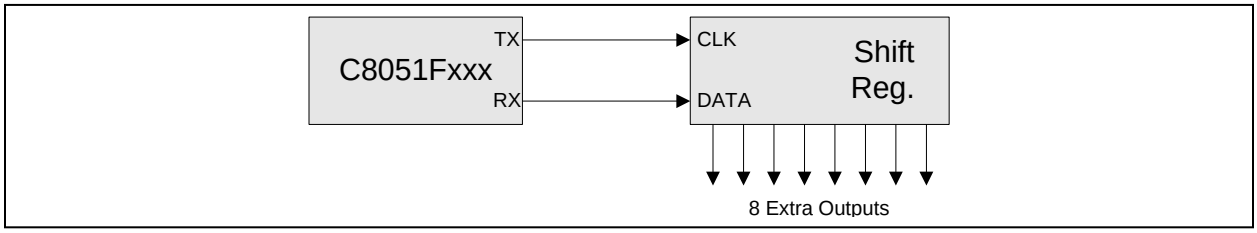
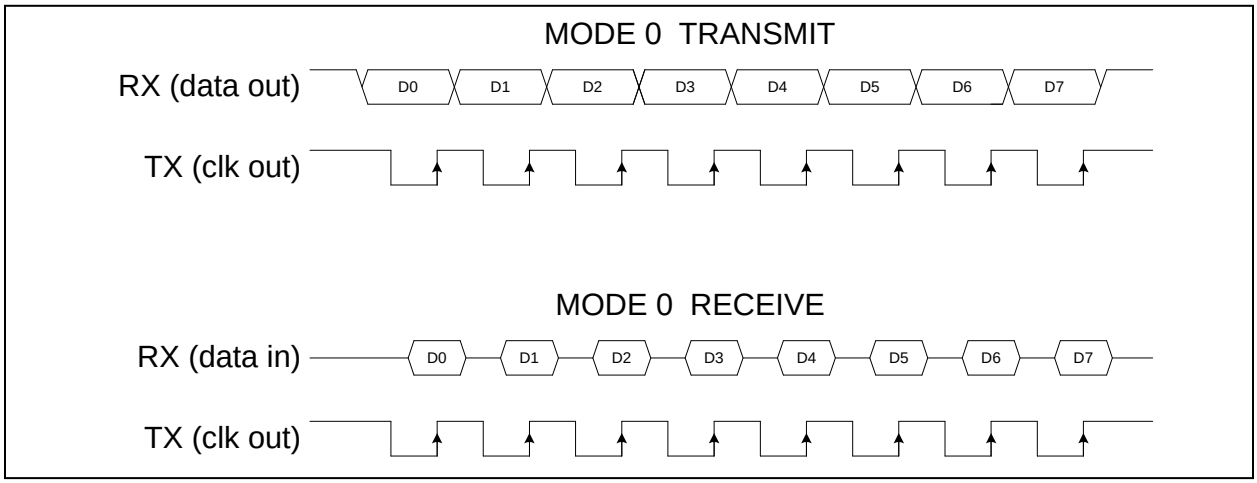


Figure 18.3. UART Mode 0 Timing Diagram



Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is 0 or the input signal /INT0 is logic-level one. Setting GATE0 to logic 1 allows the timer to be controlled by the external input signal /INT0, facilitating pulse width measurements.

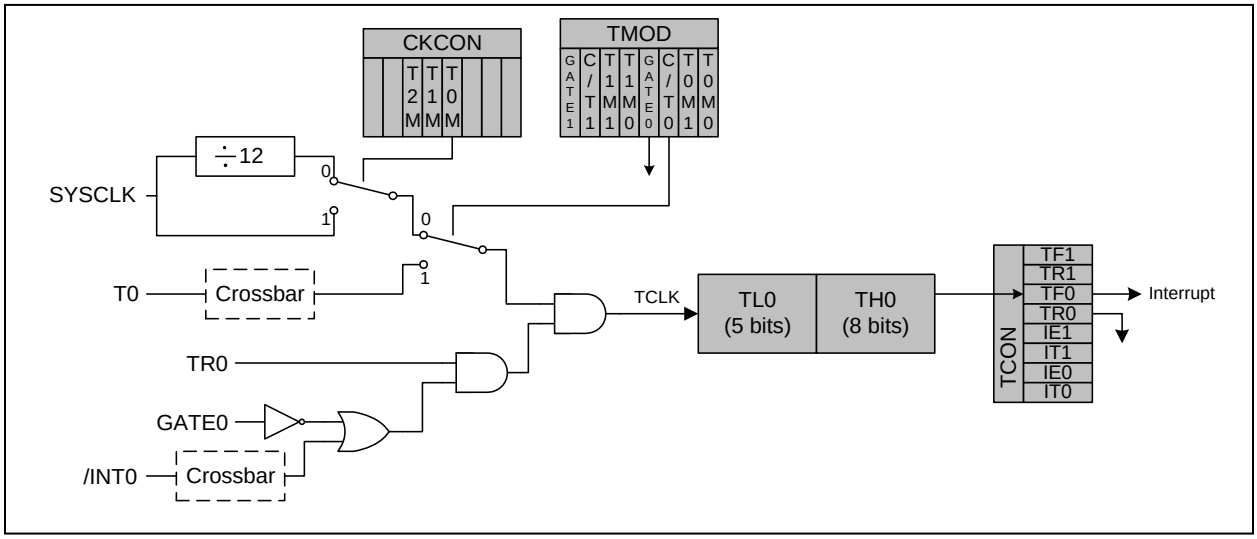
TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

X = Don't Care

Setting TR0 does not reset the timer register. The timer register should be initialized to the desired value before enabling the timer.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0.

Figure 19.1. T0 Mode 0 Block Diagram



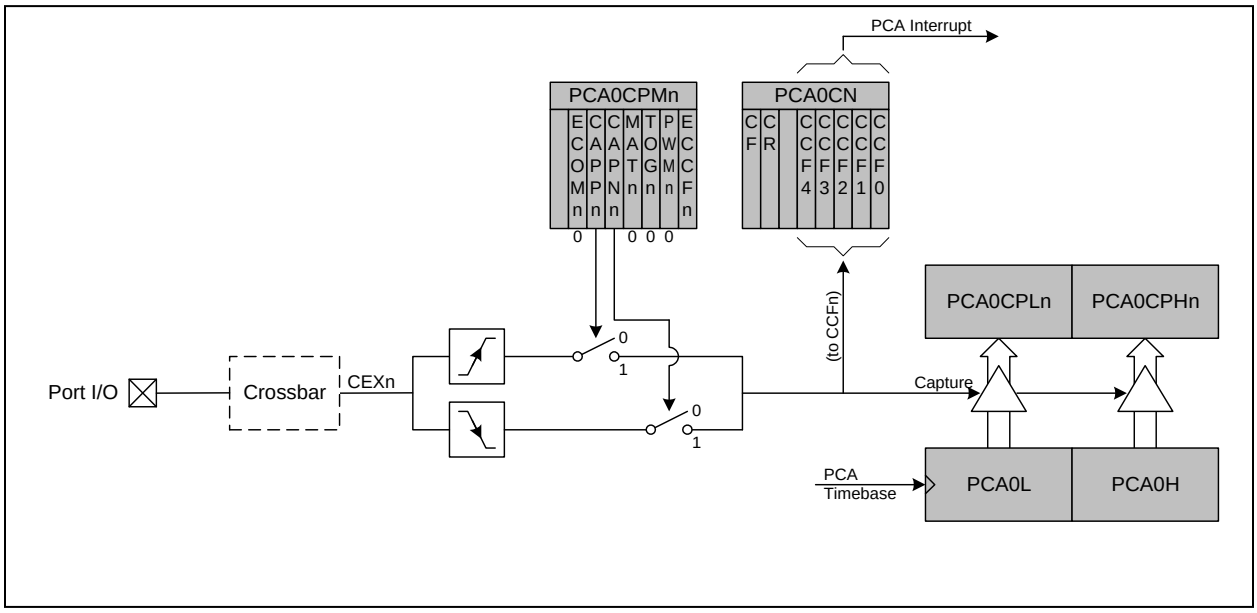
19.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

20.1.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture 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.

Figure 20.3. PCA Capture Mode Diagram



20.1.4. Pulse Width Modulator Mode

All of the modules can be used independently to generate pulse width modulated (PWM) outputs on their respective CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 20.6). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the PCA0CPHn without software intervention. It is good practice to write to PCA0CPHn instead of PCA0CPLn to avoid glitches in the digital comparator. Setting the ECOMn and PWMn bits in the PCA0CPMn register enables Pulse Width Modulator mode.

Figure 20.6. PCA PWM Mode Diagram

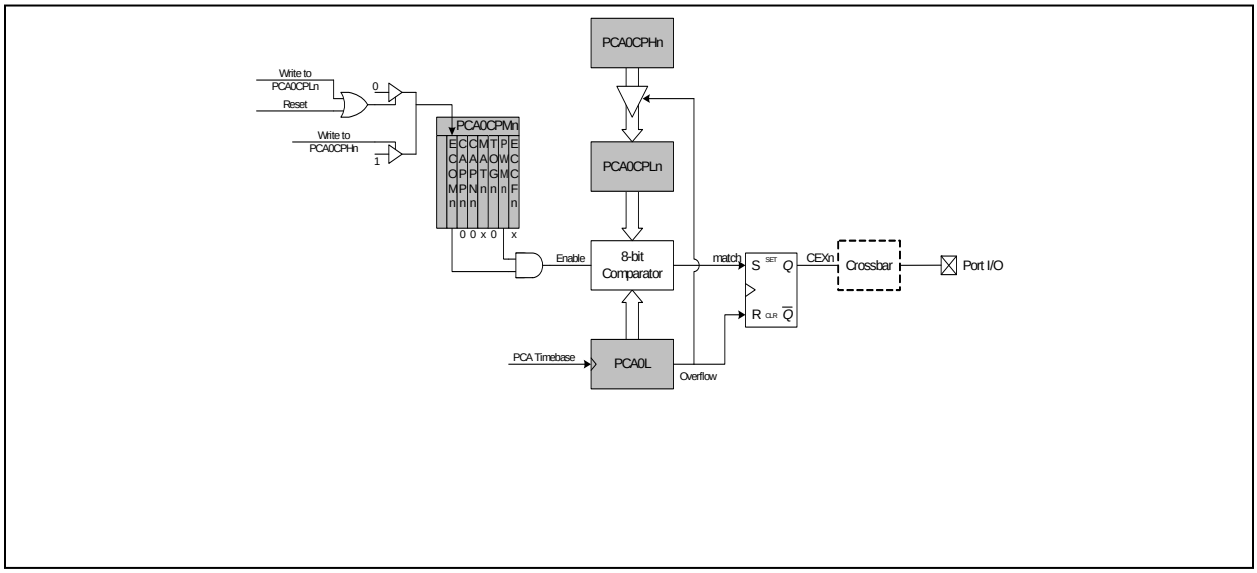


Figure 20.11. PCA0L: PCA Counter/Timer Low Byte

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: 0xE9

Bits 7-0: PCA0L: PCA Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

Figure 20.12. PCA0H: PCA Counter/Timer 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: 0xF9

Bits 7-0: PCA0H: PCA Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Note the value read is actually from the snapshot register in order to synchronize it with PCA0L.

Figure 20.13. PCA0CPLn: PCA Capture Module 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: 0xEA-0xEE

PCA0CPLn Address: PCA0CPL0 = 0xEA (n = 0)
PCA0CPL1 = 0xEB (n = 1)
PCA0CPL2 = 0xEC (n = 2)
PCA0CPL3 = 0xED (n = 3)
PCA0CPL4 = 0xEE (n = 4)

Bits7-6: PCA0CPLn: PCA Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module *n*.

Figure 20.14. PCA0CPHn: PCA Capture Module 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: 0xFA-0xFE

PCA0CPHn Address: PCA0CPH0 = 0xFA (n = 0)
PCA0CPH1 = 0xFB (n = 1)
PCA0CPH2 = 0xFC (n = 2)
PCA0CPH3 = 0xFD (n = 3)
PCA0CPH4 = 0xFE (n = 4)

Bits7-0: PCA0CPHn: PCA Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

Figure 21.5. FLASHDAT: JTAG Flash Data Register

DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	FAIL	FBUSY	Reset Value
Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	0000000000

This register is used to read or write data to the Flash memory across the JTAG interface.

Bits9-2: DATA7-0: Flash Data Byte.

Bit1: FAIL: Flash Fail Bit.
- 0: Previous Flash memory operation was successful.
- 1: Previous Flash memory operation failed. Usually indicates the associated memory location was locked.

Bit0: FBUSY: Flash Busy Bit.
- 0: Flash interface logic is not busy.
- 1: Flash interface logic is processing a request. Reads or writes while FBUSY = 1 will not initiate another operation

Figure 21.6. FLASHSCL: JTAG Flash Scale Register

FOSE	FRAE	-	-	FLSCL3	FLSCL2	FLSCL1	FLSCL0	Reset Value
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	00000000

This register controls the Flash read timing circuit and the prescaler required to generate the correct timing for Flash operations.

Bit7: FOSE: Flash One-Shot Enable Bit.
- 0: Flash read strobe is a full clock-cycle wide.
- 1: Flash read strobe is 50nsec.

Bit6: FRAE: Flash Read Always Bit.
- 0: The Flash output enable and sense amplifier enable are on only when needed to read the Flash memory.
- 1: The Flash output enable and sense amplifier enable are always on. This can be used to limit the variations in digital supply current due to switching the sense amplifiers, thereby reducing digitally induced noise.

Bits5-4: UNUSED. Read = 00b, Write = don't care.

Bits3-0: FLSCL3-0: Flash Prescaler Control Bits.
The FLSCL3-0 bits control the prescaler used to generate timing signals for Flash operations. Its value should be written before any Flash operations are initiated. The value written should be the smallest integer for which:

$$FLSCL[3:0] > \log_2(f_{SYSCLK} / 50kHz)$$

Where f_{SYSCLK} is the system clock frequency. All Flash read/write/erase operations are disallowed when FLSCL[3:0] = 1111b.