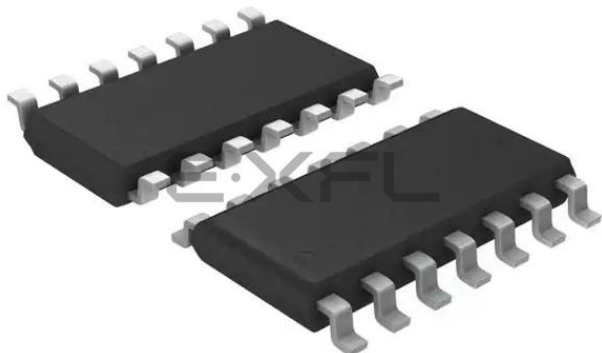




Welcome to [E-XFL.COM](https://www.e-xfl.com)

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, Temp Sensor, WDT
Number of I/O	8
Program Memory Size	2KB (2K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	14-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t604-gsr

C8051T600/1/2/3/4/5/6

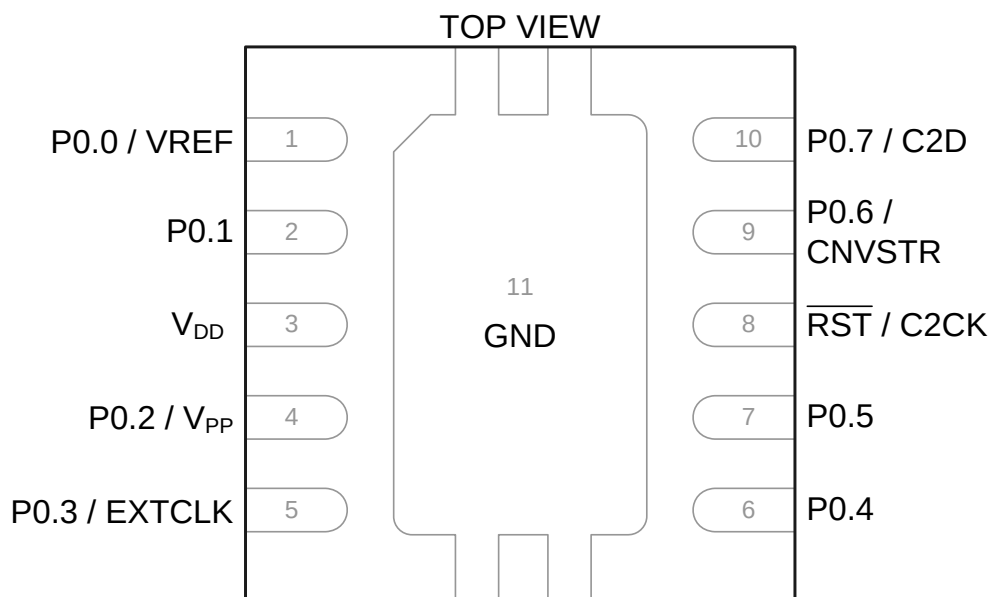


Figure 3.1. C8051T600/1/2/3/4/5-GM QFN11 Pinout Diagram (Top View)

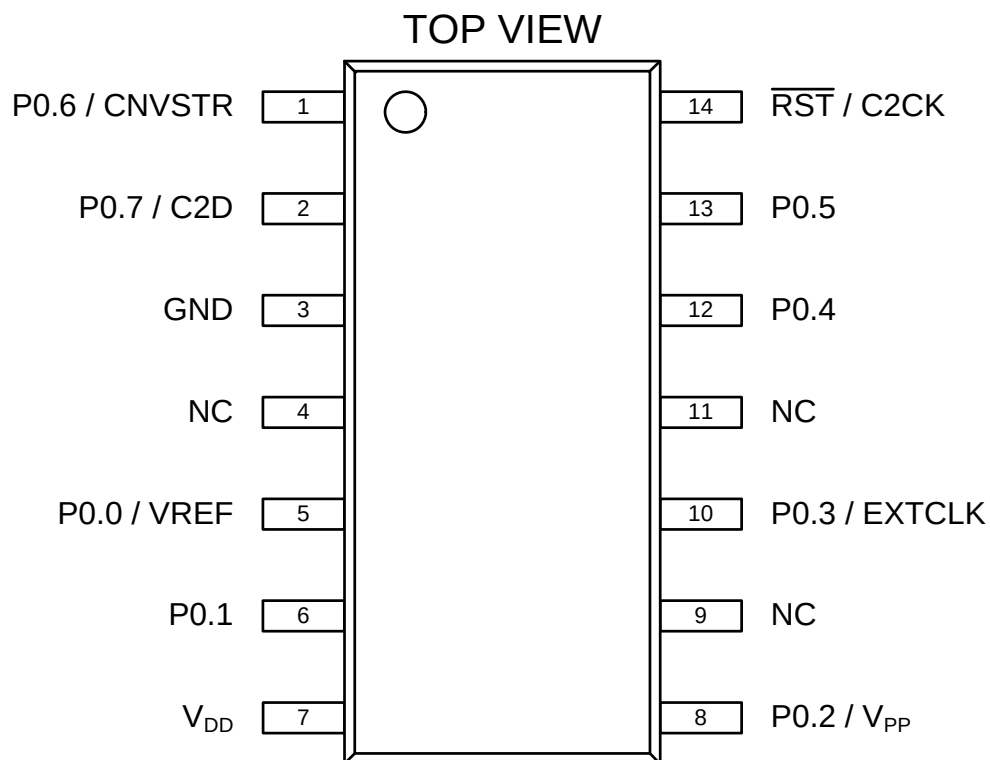


Figure 3.2. C8051T600/1/2/3/4/5-GS SOIC14 Pinout Diagram (Top View)

Table 8.7. Internal High-Frequency Oscillator Electrical Characteristics

$V_{DD} = 1.8$ to 3.6 V; $T_A = -40$ to $+85$ °C unless otherwise specified. Use factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	IFCN = 11b	24	24.5	25	MHz
Oscillator Supply Current (from V_{DD})	25 °C, $V_{DD} = 3.0$ V, OSCICN.2 = 1	—	450	700	μA
Power Supply Variance	Constant Temperature	—	±0.02	—	%/V
Temperature Variance	Constant Supply	—	±20	—	ppm/°C

Table 8.8. Temperature Sensor Electrical Characteristics

$V_{DD} = 3.0$ V, -40 to $+85$ °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Linearity		—	±0.5	—	°C
Slope		—	3.2	—	mV/°C
Slope Error*		—	±80	—	μV/°C
Offset	Temp = 0 °C	—	903	—	mV
Offset Error*	Temp = 0 °C	—	±10	—	mV
Note: Represents one standard deviation from the mean.					

Table 8.9. Voltage Reference Electrical Characteristics

$V_{DD} = 3.0$ V; -40 to $+85$ °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range		0	—	V_{DD}	V
Input Current	Sample Rate = 500 ksps; VREF = 2.5 V	—	12	—	μA

C8051T600/1/2/3/4/5/6

8.3. Typical Performance Curves

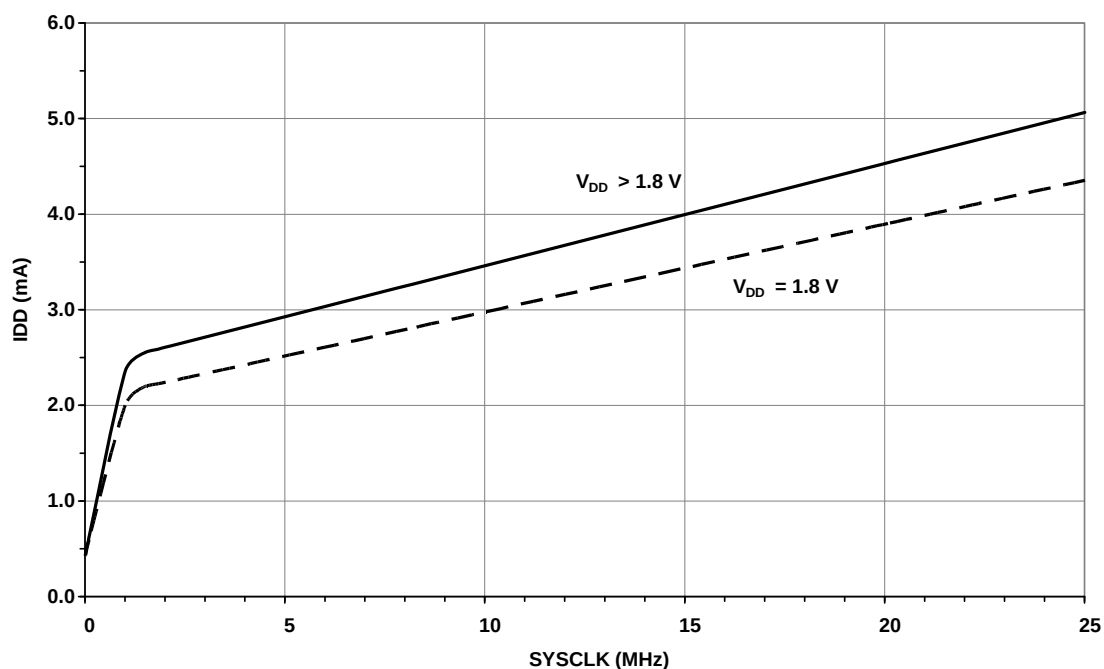


Figure 8.1. C8051T600/1/2/3/4/5 Normal Mode Supply Current vs. Frequency (MPCE = 1)

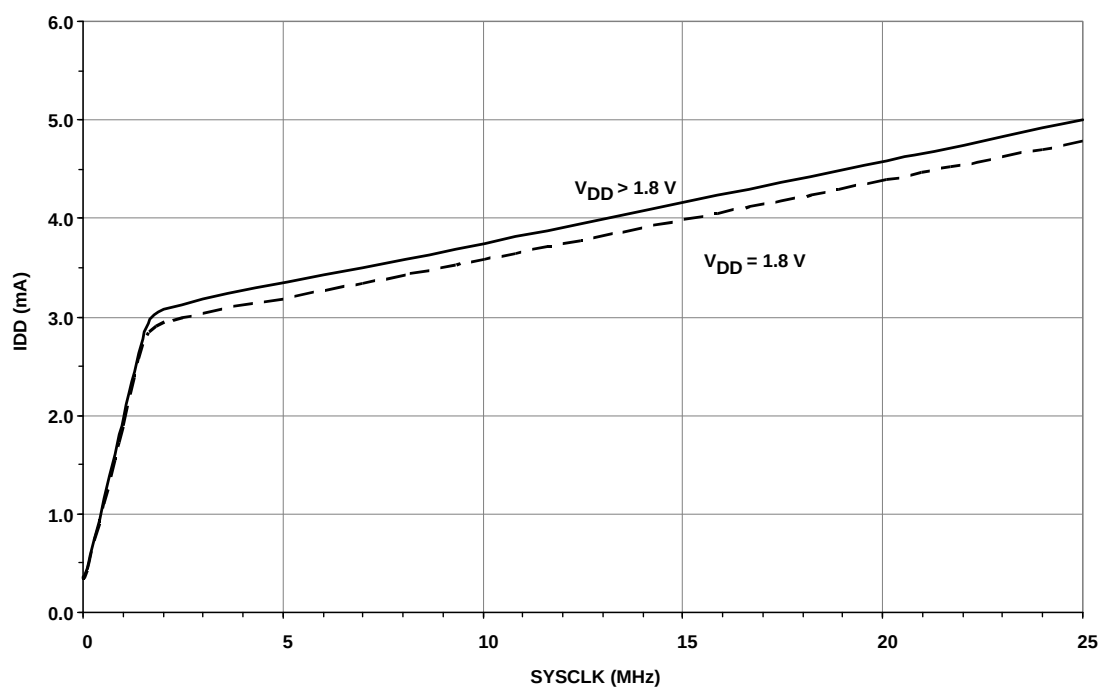


Figure 8.2. C8051T606 Normal Mode Supply Current vs. Frequency (MPCE = 1)

9.1. Output Code Formatting

The ADC measures the input voltage with reference to GND. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit. Conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to VREF x 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

9.2. 8-Bit Mode

Setting the ADC08BE bit in register ADC0CF to 1 will put the ADC in 8-bit mode. In 8-bit mode, only the 8 MSBs of data are converted, and the ADC0H register holds the results. The AD0LJST bit is ignored for 8-bit mode. 8-bit conversions take two fewer SAR clock cycles than 10-bit conversions, so the conversion is completed faster, and a 500 kps sampling rate can be achieved with a slower SAR clock.

9.3. Modes of Operation

ADC0 has a maximum conversion speed of 500 kps. The ADC0 conversion clock is a divided version of the system clock, determined by the AD0SC bits in the ADC0CF register.

9.3.1. Starting a Conversion

A conversion can be initiated in one of six ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a 1 to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 1 overflow
5. A rising edge on the CNVSTR input signal
6. A Timer 3 overflow

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. Note that when Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode. High byte overflows are used if Timer 2/3 is in 16-bit mode. See Section "25. Timers" on page 145 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as a Port I/O pin. When the CNVSTR input is used as the ADC0 conversion source, the associated pin should be skipped by the Digital Crossbar. See Section "22. Port Input/Output" on page 106 for details on Port I/O configuration.

C8051T600/1/2/3/4/5/6

SFR Definition 9.7. ADC0LTH: ADC0 Less-Than Data High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0LTH[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC6

Bit	Name	Function
7:0	ADC0LTH[7:0]	ADC0 Less-Than Data Word High-Order Bits.

SFR Definition 9.8. ADC0LTL: ADC0 Less-Than Data Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0LTL[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC5

Bit	Name	Function
7:0	ADC0LTL[7:0]	ADC0 Less-Than Data Word Low-Order Bits.

C8051T600/1/2/3/4/5/6

9.5. ADC0 Analog Multiplexer (C8051T600/2/4 only)

ADC0 on the C8051T600/2/4 uses an analog input multiplexer to select the positive input to the ADC. Any of the following may be selected as the positive input: Port 0 I/O pins, the on-chip temperature sensor, or the positive power supply (V_{DD}). The ADC0 input channel is selected in the AMX0SL register described in SFR Definition 9.9.

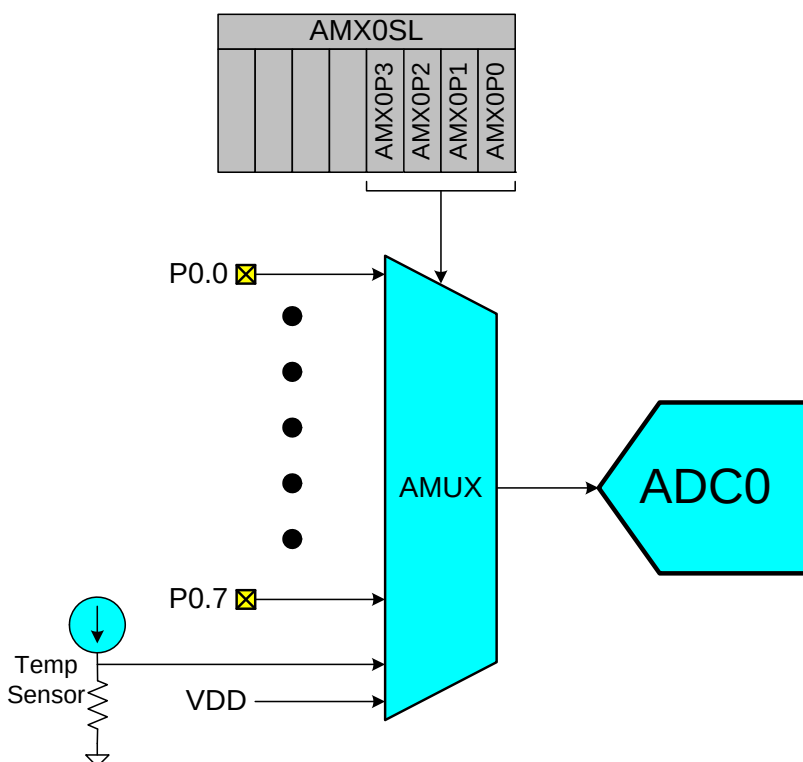


Figure 9.6. ADC0 Multiplexer Block Diagram

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set the corresponding bit in register PnMDIN to '0'. To force the Crossbar to skip a Port pin, set the corresponding bit in register XBR0 to '1'. See Section "22. Port Input/Output" on page 106 for more Port I/O configuration details.

11. Voltage Reference Options

The voltage reference multiplexer for the ADC is configurable for use with an externally connected voltage reference, the unregulated power supply voltage (V_{DD}), or the regulated 1.8 V internal supply (see Figure 11.1). The REFSL bit in the Reference Control register (REF0CN, SFR Definition 11.1) selects the reference source for the ADC. For an external source, REFSL should be set to 0 to select the VREF pin. To use V_{DD} as the reference source, REFSL should be set to 1. To override this selection and use the internal regulator as the reference source, the REGOVR bit can be set to 1. The electrical specifications for the voltage reference circuit are given in Section “8. Electrical Characteristics” on page 30.

Important Note about the VREF Pin: When using an external voltage reference, the VREF pin should be configured as an analog pin and skipped by the Digital Crossbar. Refer to Section “22. Port Input/Output” on page 106 for the location of the VREF pin, as well as details of how to configure the pin in analog mode and to be skipped by the crossbar.

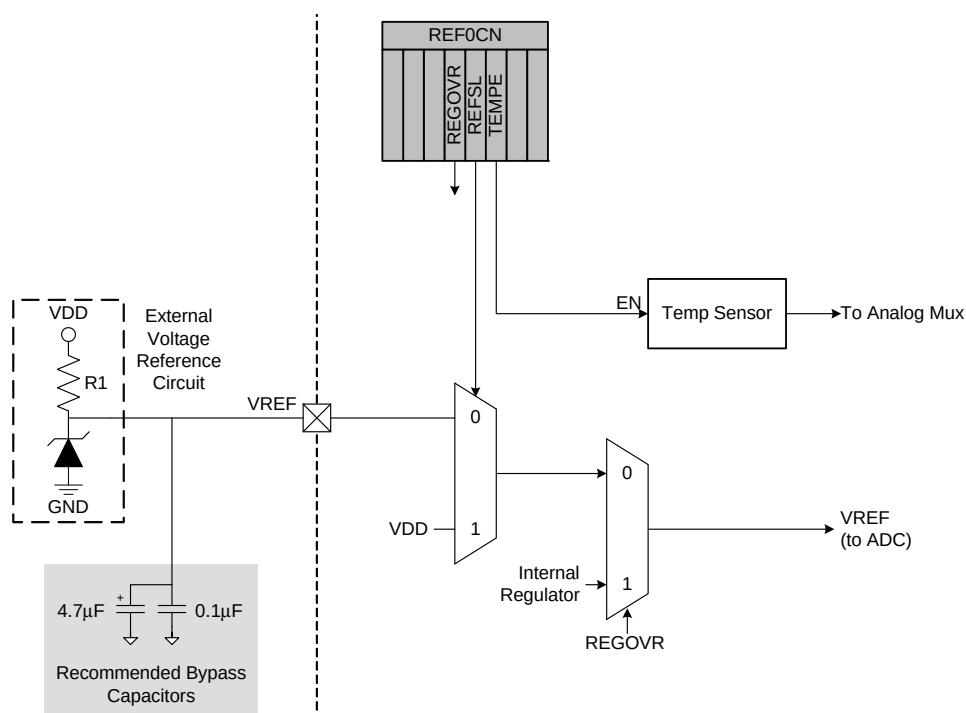


Figure 11.1. Voltage Reference Functional Block Diagram

SFR Definition 12.1. REG0CN: Voltage Regulator Control

Bit	7	6	5	4	3	2	1	0
Name	STOPCF	BYPASS						MPCE
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC7

Bit	Name	Function
7	STOPCF	Stop Mode Configuration. This bit configures the regulator's behavior when the device enters STOP mode. 0: Regulator is still active in STOP mode. Any enabled reset source will reset the device. 1: Regulator is shut down in STOP mode. Only the $\overline{\text{RST}}$ pin or power cycle can reset the device.
6	BYPASS	Bypass Internal Regulator. This bit places the regulator in bypass mode, turning off the regulator, and allowing the core to run directly from the V_{DD} supply pin. 0: Normal Mode—Regulator is on. 1: Bypass Mode—Regulator is off, and the microcontroller core operates directly from the V_{DD} supply voltage. IMPORTANT: Bypass mode is for use with an external regulator as the supply voltage only. Never place the regulator in bypass mode when the V_{DD} supply voltage is greater than the specifications given in Table 8.1 on page 30. Doing so may cause permanent damage to the device.
5:1	Reserved	Reserved. Must Write 00000b.
0	MPCE	Memory Power Controller Enable. This bit can help the system save power at slower system clock frequencies (about 2.0 MHz or less) by automatically shutting down the EPROM memory between clocks when information is not being fetched from the EPROM memory. 0: Normal Mode—Memory power controller disabled (EPROM memory is always on). 1: Low Power Mode—Memory power controller enabled (EPROM memory turns on/off as needed). Note: If an external clock source is used with the Memory Power Controller enabled, and the clock frequency changes from slow (<2.0 MHz) to fast (> 2.0 MHz), the EPROM power will turn on, and up to 20 clocks may be "skipped" to ensure that the EPROM power is stable before reading memory.

SFR Definition 13.2. CPT0MD: Comparator0 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name							CP0MD[1:0]	
Type	R	R	R	R	R	R	R/W	
Reset	0	0	0	0	0	0	1	0

SFR Address = 0x9D

Bit	Name	Function
7:2	Unused	Unused. Read = 000000b, Write = Don't Care.
1:0	CP0MD[1:0]	Comparator0 Mode Select. These bits affect the response time and power consumption for Comparator0. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

18. Power Management Modes

The C8051T600/1/2/3/4/5/6 devices have two software programmable power management modes: idle and stop. Idle mode halts the CPU while leaving the peripherals and clocks active. In stop mode, the CPU is halted, all interrupts and timers (except the missing clock detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Since clocks are running in idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering idle. Stop mode consumes the least power because the majority of the device is shut down with no clocks active. SFR Definition 18.1 describes the Power Control Register (PCON) used to control the C8051T600/1/2/3/4/5/6's stop and idle power management modes.

Although the C8051T600/1/2/3/4/5/6 has idle and stop modes available, more control over the device power can be achieved by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use.

18.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the hardware to halt the CPU and enter idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has two or more opcode bytes, for example:

```
// in 'C':
PCON |= 0x01;           // set IDLE bit
PCON = PCON;           // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h          ; set IDLE bit
MOV PCON, PCON          ; ... followed by a 3-cycle dummy instruction
```

If enabled, the watchdog timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to Section “19.6. PCA Watchdog Timer Reset” on page 94 for more information on the use and configuration of the WDT.

20.3. Program Memory CRC

A CRC engine is included on-chip, which provides a means of verifying EPROM contents once the device has been programmed. The CRC engine is available for EPROM verification even if the device is fully read and write locked, allowing for verification of code contents at any time.

The CRC engine is operated through the C2 debug and programming interface, and performs 16-bit CRCs on individual 256-byte blocks of program memory, or a 32-bit CRC the entire memory space. To prevent hacking and extrapolation of security-locked source code, the CRC engine will only allow CRCs to be performed on contiguous 256-byte blocks beginning on 256-byte boundaries (lowest 8-bits of address are 0x00). For example, the CRC engine can perform a CRC for locations 0x0400 through 0x04FF, but it cannot perform a CRC for locations 0x0401 through 0x0500, or on block sizes smaller or larger than 256 bytes.

20.3.1. Performing 32-bit CRCs on Full EPROM Content

A 32-bit CRC on the entire EPROM space is initiated by writing to the CRC1 byte over the C2 interface. The CRC calculation begins at address 0x0000 and ends at the end of user EPROM space. The EPBusy bit in register C2ADD will be set during the CRC operation, and cleared once the operation is complete. The 32-bit results will be available in the CRC3-0 registers. CRC3 is the MSB, and CRC0 is the LSB. The polynomial used for the 32-bit CRC calculation is 0x04C11DB7.

Note: If a 16-bit CRC has been performed since the last device reset, a device reset should be initiated before performing a 32-bit CRC operation.

20.3.2. Performing 16-bit CRCs on 256-Byte EPROM Blocks

A 16-bit CRC of individual 256-byte blocks of EPROM can be initiated by writing to the CRC0 byte over the C2 interface. The value written to CRC0 is the high byte of the beginning address for the CRC. For example, if CRC0 is written to 0x02, the CRC will be performed on the 256 bytes beginning at address 0x0200, and ending at address 0x2FF. The EPBusy bit in register C2ADD will be set during the CRC operation, and cleared once the operation is complete. The 16-bit results will be available in the CRC1-0 registers. CRC1 is the MSB, and CRC0 is the LSB. The polynomial for the 16-bit CRC calculation is 0x1021

21. Oscillators and Clock Selection

C8051T600/1/2/3/4/5/6 devices include a programmable internal high-frequency oscillator and an external oscillator drive circuit. The internal high-frequency oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 21.1. The system clock can be sourced by the external oscillator circuit or the internal oscillator (default). The internal oscillator offers a selectable post-scaling feature, which is initially set to divide the clock by 8.

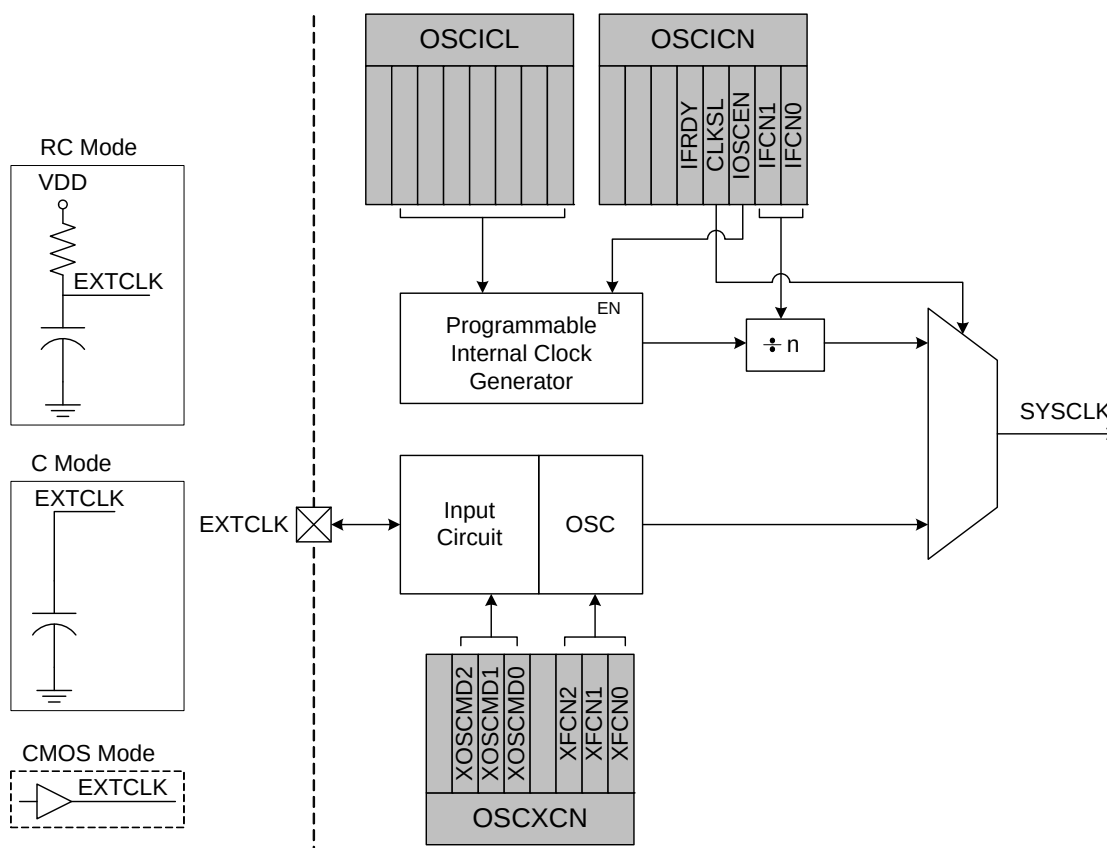


Figure 21.1. Oscillator Options

21.1. System Clock Selection

The CLKSL bit in register OSCICN selects which oscillator source is used as the system clock. CLKSL must be set to 1 for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals (timers, PCA) when the internal oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal oscillator and external oscillator, as long as the selected clock source is enabled and running.

The internal high-frequency oscillator requires little start-up time and may be selected as the system clock immediately following the register write, which enables the oscillator. The external RC and C modes also typically require no startup time.

22.4. Port I/O Initialization

Port I/O initialization consists of the following steps:

1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (P0MDIN).
2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (P0MDOUT).
3. Select any pins to be skipped by the I/O crossbar using the XBR0 register.
4. Assign Port pins to desired peripherals.
5. Enable the crossbar (XBARE = 1).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the crossbar (accomplished by setting the associated bits in XBR0). Port input mode is set in the P0MDIN register, where a 1 indicates a digital input, and a 0 indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 22.5 for the P0MDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode register (P0MDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the P0MDOUT settings. When the WEAKPUD bit in XBR2 is 0, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a 0 to avoid unnecessary power dissipation.

Registers XBR1 and XBR2 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR2 to 1 enables the crossbar. Until the crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table. An alternative is to use the Configuration Wizard utility available on the Silicon Laboratories web site to determine the Port I/O pin-assignments based on the XBRn Register settings.

The crossbar must be enabled to use Port pins as standard Port I/O in output mode. Port output drivers are disabled while the crossbar is disabled.

C8051T600/1/2/3/4/5/6

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time and waits for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data and waits for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 23.3 illustrates a typical SMBus transaction.

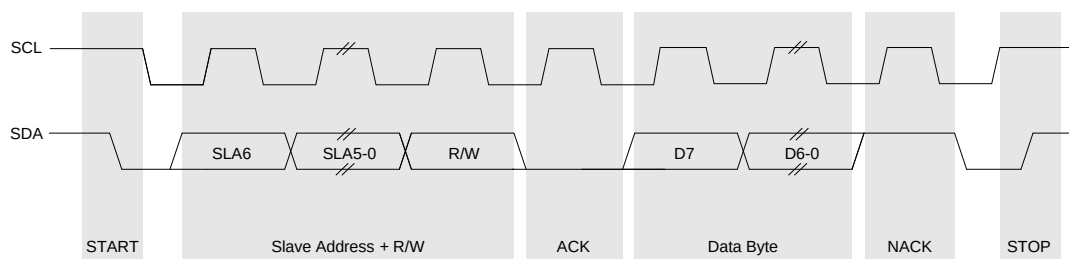


Figure 23.3. SMBus Transaction

23.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

23.3.2. 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 remain high for a specified time (see Section “23.3.5. SCL High (SMBus Free) Timeout” on page 123). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

23.3.3. 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 may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

23.3.4. 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 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

23.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 23.2). 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 indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (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 23.3 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 23.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 23.4 for SMBus status decoding using the SMB0CN register.

C8051T600/1/2/3/4/5/6

SFR Definition 23.2. SMB0CN: SMBus Control

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

SFR Address = 0xC0; Bit-Addressable

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

25.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0, and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1, or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

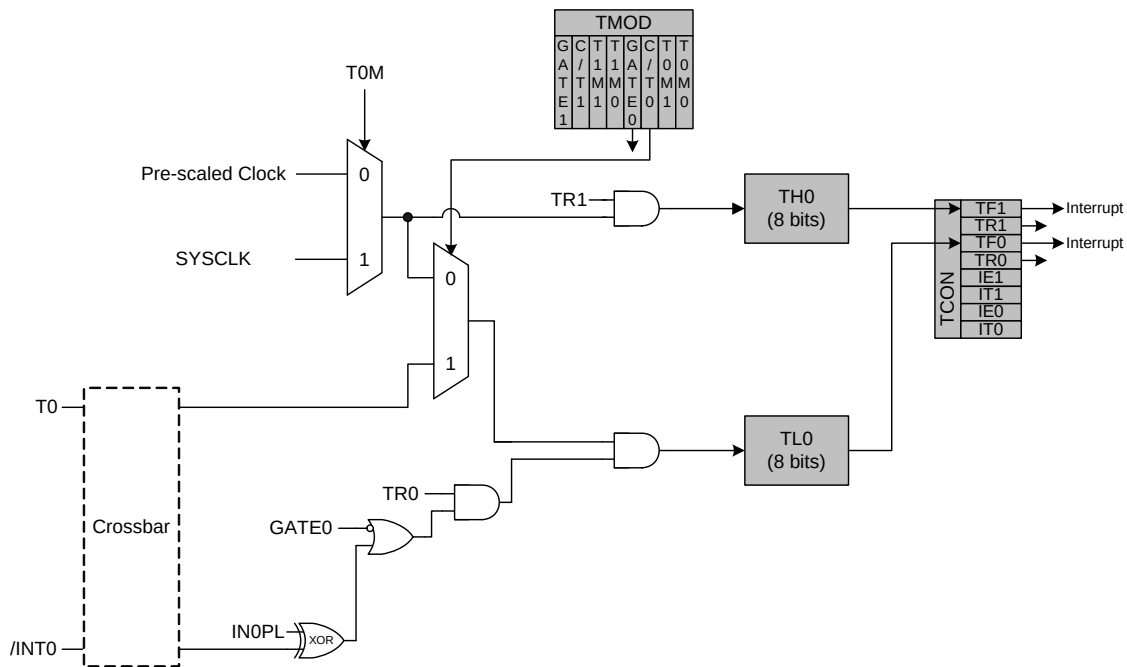


Figure 25.3. T0 Mode 3 Block Diagram

C8051T600/1/2/3/4/5/6

SFR Definition 25.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN		T2SPLIT	TR2		T2XCLK
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC8; Bit-Addressable

Bit	Name	Function
7	TF2H	Timer 2 High Byte Overflow Flag. Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit is not automatically cleared by hardware.
6	TF2L	Timer 2 Low Byte Overflow Flag. Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.
5	TF2LEN	Timer 2 Low Byte Interrupt Enable. When set to 1, this bit enables Timer 2 low byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	Unused	Unused. Read = 0b; Write = Don't Care
3	T2SPLIT	Timer 2 Split Mode Enable. When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	Timer 2 Run Control. Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.
1	Unused	Unused. Read = 0b; Write = Don't Care
0	T2XCLK	Timer 2 External Clock Select. This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. 0: Timer 2 clock is the system clock divided by 12. 1: Timer 2 clock is the external clock divided by 8 (synchronized with SYSCLK).

26.2. PCA0 Interrupt Sources

Figure 26.3 shows a diagram of the PCA interrupt tree. There are four independent event flags that can be used to generate a PCA0 interrupt. They are: the main PCA counter overflow flag (CF), which is set upon a 16-bit overflow of the PCA0 counter, and the individual flags for each PCA channel (CCF0, CCF1, and CCF2), which are set according to the operation mode of that module. These event flags are always set when the trigger condition occurs. Each of these flags can be individually selected to generate a PCA0 interrupt, using the corresponding interrupt enable flag (ECF for CF and ECCFn for each CCFn). PCA0 interrupts must be globally enabled before any individual interrupt sources are recognized by the processor. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1.

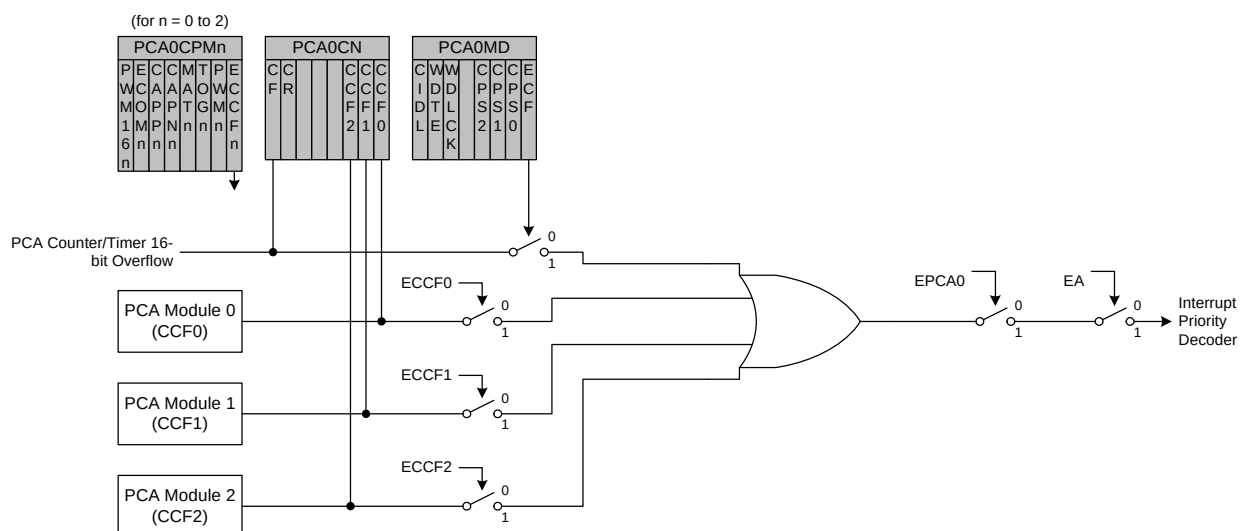


Figure 26.3. PCA Interrupt Block Diagram

C8051T600/1/2/3/4/5/6

C2 Register Definition 27.4. DEVCTL: C2 Device Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	DEVCTL[7:0]	Device Control Register. This register is used to halt the device for EPROM operations via the C2 interface. Refer to the EPROM chapter for more information.

C2 Register Definition 27.5. EPCTL: EPROM Programming Control Register

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

C2 Address: 0xDF

Bit	Name	Function
7:0	EPCTL[7:0]	EPROM Programming Control Register. This register is used to enable EPROM programming via the C2 interface. Refer to the EPROM chapter for more information.