

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

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

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

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

Details

Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	6
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	10-TFSOP, 10-MSOP (0.118", 3.00mm Width)
Supplier Device Package	10-MSOP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t606-gtr

27.2. C2 Pin Sharing	185
Document Change List.....	186
Contact Information.....	188

18. Power Management Modes	
19. Reset Sources	
20. EPROM Memory	
Table 20.1. Security Byte Decoding	98
21. Oscillators and Clock Selection	
22. Port Input/Output	
Table 22.1. Port I/O Assignment for Analog Functions	109
Table 22.2. Port I/O Assignment for Digital Functions	109
Table 22.3. Port I/O Assignment for External Digital Event Capture Functions	110
23. SMBus	
Table 23.1. SMBus Clock Source Selection	124
Table 23.2. Minimum SDA Setup and Hold Times	125
Table 23.3. Sources for Hardware Changes to SMB0CN	129
Table 23.4. SMBus Status Decoding	135
24. UART0	
Table 24.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator	144
Table 24.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator	144
25. Timers	
26. Programmable Counter Array	
Table 26.1. PCA Timebase Input Options	161
Table 26.2. PCA0CPM Bit Settings for PCA Capture/Compare Modules	163
Table 26.3. Watchdog Timer Timeout Intervals ¹	172
27. C2 Interface	

SFR Definition 25.1. CKCON: Clock Control	146
SFR Definition 25.2. TCON: Timer Control	151
SFR Definition 25.3. TMOD: Timer Mode	152
SFR Definition 25.4. TL0: Timer 0 Low Byte	153
SFR Definition 25.5. TL1: Timer 1 Low Byte	153
SFR Definition 25.6. TH0: Timer 0 High Byte	154
SFR Definition 25.7. TH1: Timer 1 High Byte	154
SFR Definition 25.8. TMR2CN: Timer 2 Control	157
SFR Definition 25.9. TMR2RLL: Timer 2 Reload Register Low Byte	158
SFR Definition 25.10. TMR2RLH: Timer 2 Reload Register High Byte	158
SFR Definition 25.11. TMR2L: Timer 2 Low Byte	158
SFR Definition 25.12. TMR2H: Timer 2 High Byte	159
SFR Definition 26.1. PCA0CN: PCA Control	173
SFR Definition 26.2. PCA0MD: PCA Mode	174
SFR Definition 26.3. PCA0CPMn: PCA Capture/Compare Mode	175
SFR Definition 26.4. PCA0L: PCA Counter/Timer Low Byte	176
SFR Definition 26.5. PCA0H: PCA Counter/Timer High Byte	176
SFR Definition 26.6. PCA0CPLn: PCA Capture Module Low Byte	177
SFR Definition 26.7. PCA0CPHn: PCA Capture Module High Byte	177
C2 Register Definition 27.1. C2ADD: C2 Address	178
C2 Register Definition 27.2. DEVICEID: C2 Device ID	179
C2 Register Definition 27.3. REVID: C2 Revision ID	179
C2 Register Definition 27.4. DEVCTL: C2 Device Control	180
C2 Register Definition 27.5. EPCTL: EPROM Programming Control Register	180
C2 Register Definition 27.6. EPDAT: C2 EPROM Data	181
C2 Register Definition 27.7. EPSTAT: C2 EPROM Status	181
C2 Register Definition 27.8. EPADDRH: C2 EPROM Address High Byte	182
C2 Register Definition 27.9. EPADDRL: C2 EPROM Address Low Byte	182
C2 Register Definition 27.10. CRC0: CRC Byte 0	183
C2 Register Definition 27.11. CRC1: CRC Byte 1	183
C2 Register Definition 27.12. CRC2: CRC Byte 2	184
C2 Register Definition 27.13. CRC3: CRC Byte 3	184

9.4.1. Window Detector Example

Figure 9.4 shows two example window comparisons for right-justified data, with $\text{ADC0LTH:ADC0LTL} = 0x0080$ (128d) and $\text{ADC0GTH:ADC0GTL} = 0x0040$ (64d). The input voltage can range from 0 to $\text{VREF} \times (1023/1024)$ with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if $0x0040 < \text{ADC0H:ADC0L} < 0x0080$). In the right example, an AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if $\text{ADC0H:ADC0L} < 0x0040$ or $\text{ADC0H:ADC0L} > 0x0080$). Figure 9.5 shows an example using left-justified data with the same comparison values.

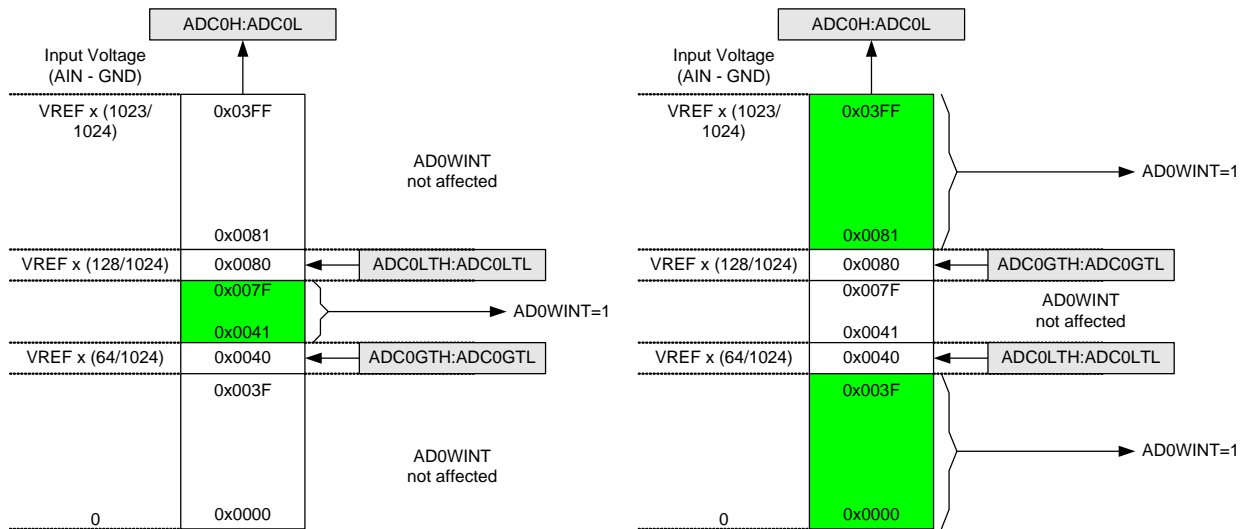


Figure 9.4. ADC Window Compare Example: Right-Justified Data

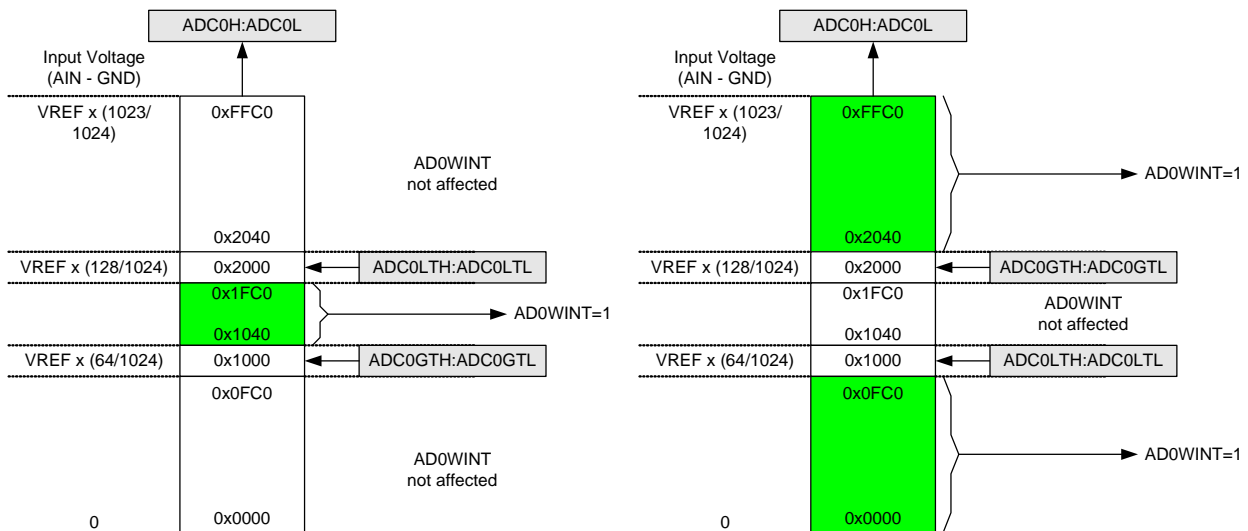


Figure 9.5. ADC Window Compare Example: Left-Justified Data

13.1. Comparator Multiplexer

C8051T600/1/2/3/4/5/6 devices include an analog input multiplexer to connect Port I/O pins to the comparator inputs. The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 13.3). The CMX0P1–CMX0P0 bits select the Comparator0 positive input; the CMX0N1–CMX0N0 bits select the Comparator0 negative input.

Important Note About Comparator Inputs: The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “22.5. Special Function Registers for Accessing and Configuring Port I/O” on page 118).

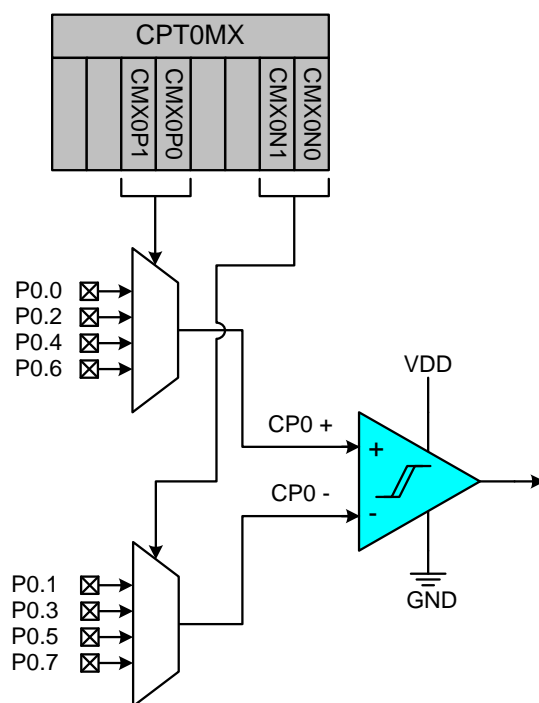


Figure 13.3. Comparator Input Multiplexer Block Diagram

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

SFR Definition 14.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	PARITY
Type	R/W	R/W	R/W	R/W		R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD0; Bit-Addressable

Bit	Name	Function
7	CY	Carry Flag. This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	Auxiliary Carry Flag. This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	User Flag 0. This is a bit-addressable, general purpose flag for use under software control.
4:3	RS[1:0]	Register Bank Select. These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	Overflow Flag. This bit is set to 1 under the following circumstances: <ul style="list-style-type: none"> ■ An ADD, ADDC, or SUBB instruction causes a sign-change overflow. ■ A MUL instruction results in an overflow (result is greater than 255). ■ A DIV instruction causes a divide-by-zero condition. The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.
1	F1	User Flag 1. This is a bit-addressable, general purpose flag for use under software control.
0	PARITY	Parity Flag. This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

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

15.2.1.1. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 14.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

15.2.1.2. Bit Addressable Locations

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

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

```
MOV    C, 22.3h
```

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

15.2.1.3. Stack

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

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

Table 16.2. Special Function Registers (Continued)

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

Register	Address	Description	Page
PCA0H	0xFA	PCA Counter High	176
PCA0L	0xF9	PCA Counter Low	176
PCA0MD	0xD9	PCA Mode	174
PCON	0x87	Power Control	91
PSW	0xD0	Program Status Word	73
REF0CN	0xD1	Voltage Reference Control	56
REG0CN	0xC7	Voltage Regulator Control	58
RSTSRC	0xEF	Reset Source Configuration/Status	96
SBUF0	0x99	UART0 Data Buffer	143
SCON0	0x98	UART0 Control	142
SMB0CF	0xC1	SMBus Configuration	126
SMB0CN	0xC0	SMBus Control	128
SMB0DAT	0xC2	SMBus Data	130
SP	0x81	Stack Pointer	72
TCON	0x88	Timer/Counter Control	151
TH0	0x8C	Timer/Counter 0 High	154
TH1	0x8D	Timer/Counter 1 High	154
TL0	0x8A	Timer/Counter 0 Low	153
TL1	0x8B	Timer/Counter 1 Low	153
TMOD	0x89	Timer/Counter Mode	152
TMR2CN	0xC8	Timer/Counter 2 Control	157
TMR2H	0xCD	Timer/Counter 2 High	159
TMR2L	0xCC	Timer/Counter 2 Low	158
TMR2RLH	0xCB	Timer/Counter 2 Reload High	158
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	158
TOFFH	0xA3	Temperature Sensor Offset Measurement High	54
TOFFL	0xA2	Temperature Sensor Offset Measurement Low	54
XBR0	0xE1	Port I/O Crossbar Control 0	115
XBR1	0xE2	Port I/O Crossbar Control 1	116
XBR2	0xE3	Port I/O Crossbar Control 2	117
All other SFR Locations		Reserved	

17. Interrupts

The C8051T600/1/2/3/4/5/6 includes an extended interrupt system supporting a total of 12 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

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

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

Note: Any instruction that clears a bit to disable an interrupt should be immediately followed by an instruction that has two or more opcode bytes. Using EA (global interrupt enable) as an example:

```
// in 'C':
EA = 0; // clear EA bit.
EA = 0; // this is a dummy instruction with two-byte opcode.

; in assembly:
CLR EA ; clear EA bit.
CLR EA ; this is a dummy instruction with two-byte opcode.
```

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

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

SFR Definition 17.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	IEGF0	ET2	ES0	ET1	EX1	ET0	EX0
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 = 0xA8; Bit-Addressable

Bit	Name	Function
7	EA	Enable All Interrupts. Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	IEGF0	General Purpose Flag 0. This is a general purpose flag for use under software control.
5	ET2	Enable Timer 2 Interrupt. This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	Enable UART0 Interrupt. This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	Enable Timer 1 Interrupt. This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	Enable External Interrupt 1. This bit sets the masking of External Interrupt 1. 0: Disable External Interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	Enable Timer 0 Interrupt. This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	Enable External Interrupt 0. This bit sets the masking of External Interrupt 0. 0: Disable External Interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

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.

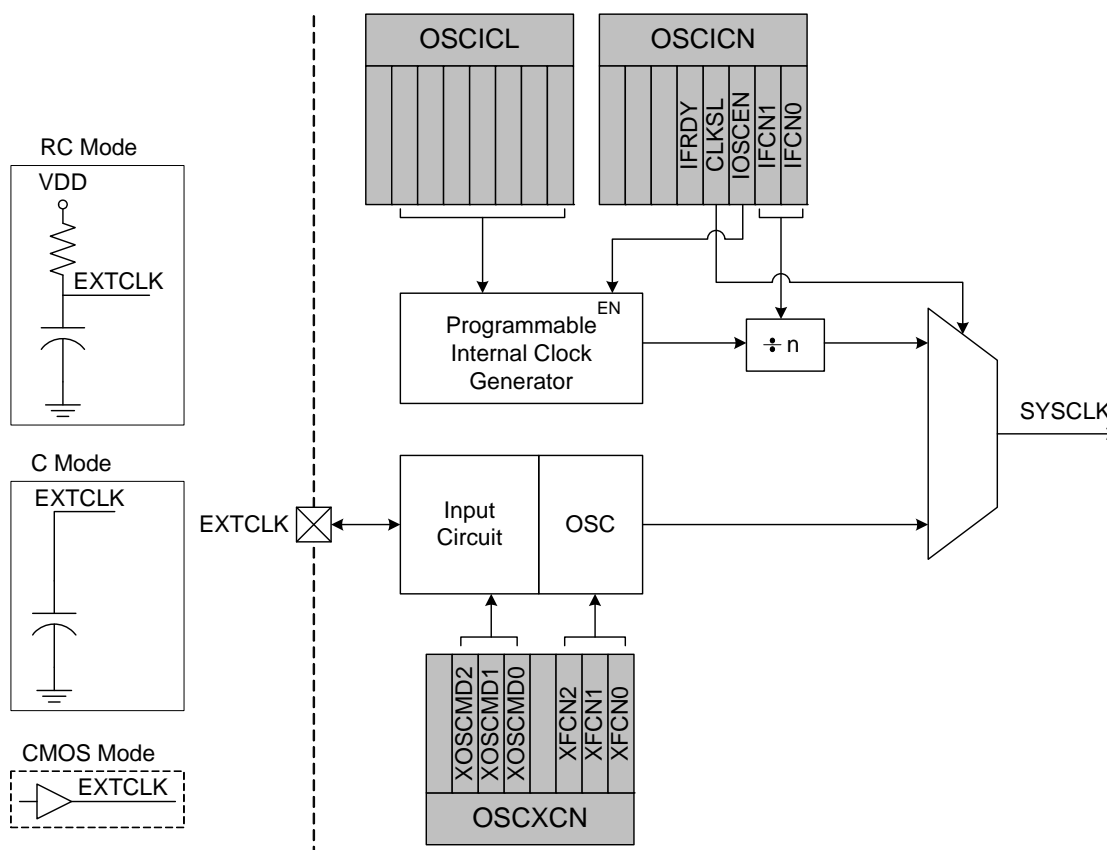


Figure 21.1. Oscillator Options

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.

22.1. Port I/O Modes of Operation

Port pins use the Port I/O cell shown in Figure 22.2. Each Port I/O cell can be configured by software for analog I/O or digital I/O using the P0MDIN registers. On reset, all Port I/O cells default to a high impedance state with weak pull-ups enabled until the crossbar is enabled (XBARE = 1).

22.1.1. Port Pins Configured for Analog I/O

Any pins to be used as inputs to the comparator, ADC, external oscillator, or VREF should be configured for analog I/O (P0MDIN.n = 0). When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. Port pins configured for analog I/O will always read back a value of 0.

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

22.1.2. Port Pins Configured For Digital I/O

Any pins to be used by digital peripherals (UART, SMBus, PCA, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (P0MDIN.n = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the P0MDOUT registers.

Push-pull outputs (P0MDOUT.n = 1) drive the Port pad to the VDD or GND supply rails based on the output logic value of the Port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the Port pad to GND when the output logic value is 0 and become high impedance inputs (both high and low drivers turned off) when the output logic value is 1.

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

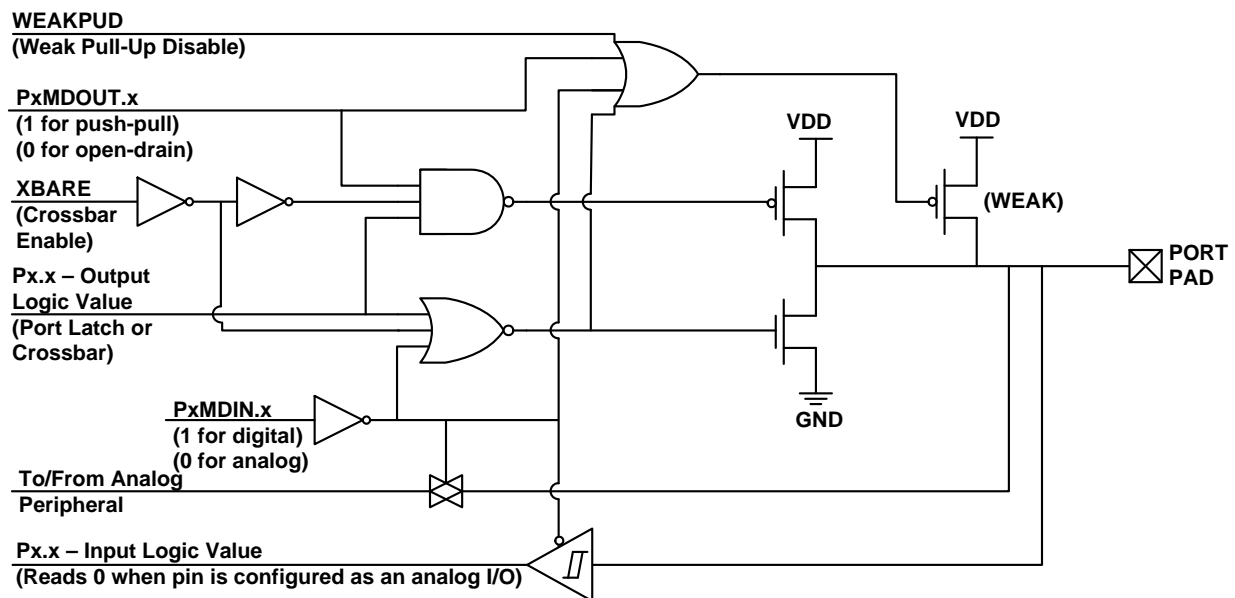


Figure 22.2. Port I/O Cell Block Diagram

23.5.4. Read Sequence (Slave)

During a read sequence, an SMBus master reads data from a slave device. The slave in this transfer will be a receiver during the address byte, and a transmitter during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode (to receive the slave address) when a START followed by a slave address and direction bit (READ in this case) is received. Upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK or ignore the received slave address with a NACK.

If the received slave address is ignored by software (by NACKing the address), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are transmitted. If the received slave address is acknowledged, data should be written to SMB0DAT to be transmitted. The interface enters slave transmitter mode and transmits one or more bytes of data. After each byte is transmitted, the master sends an acknowledge bit. If the acknowledge bit is an ACK, SMB0DAT should be written with the next data byte. If the acknowledge bit is a NACK, SMB0DAT should not be written to before SI is cleared (an error condition may be generated if SMB0DAT is written following a received NACK while in slave transmitter mode). The interface exits slave transmitter mode after receiving a STOP. Note that the interface will switch to slave receiver mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 23.8 shows a typical slave read sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode.

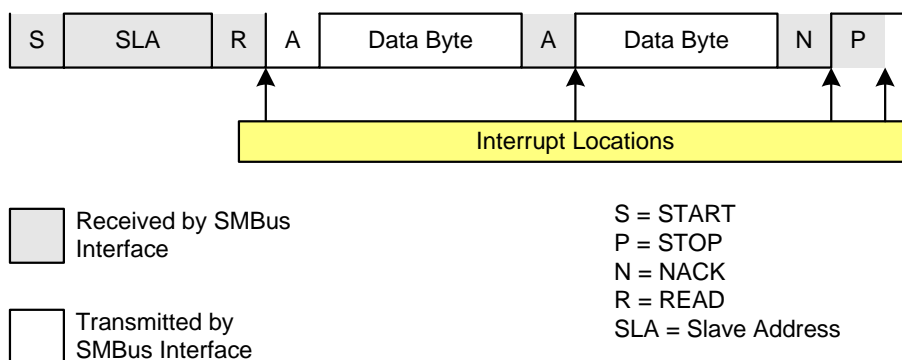


Figure 23.8. Typical Slave Read Sequence

23.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. Table 23.4 describes the typical actions taken by firmware on each condition. In the table, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed by hardware but do not conform to the SMBus specification.

24. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “24.1. Enhanced Baud Rate Generation” on page 138). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

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

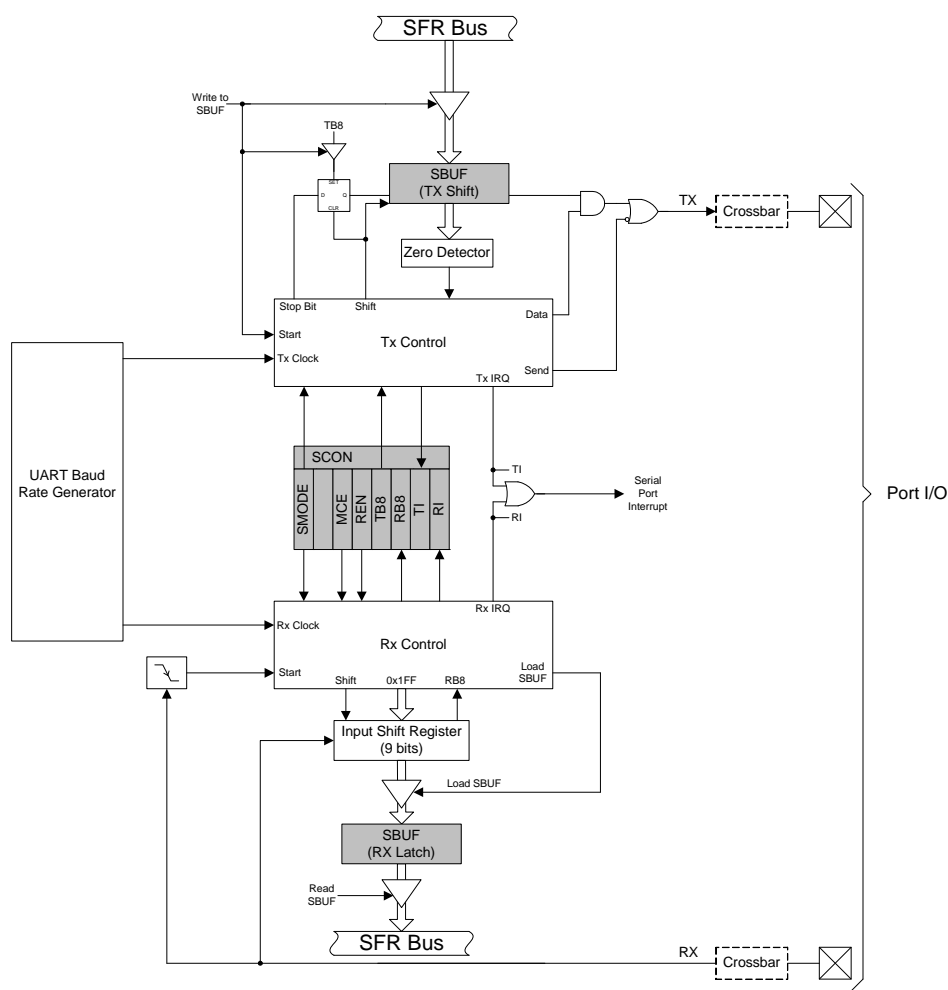


Figure 24.1. UART0 Block Diagram

25. Timers

Each MCU includes three counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and one is a 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers 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 16-bit and split 8-bit timer functionality with auto-reload.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:
13-bit counter/timer	16-bit timer with auto-reload
16-bit counter/timer	
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)	

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (see SFR Definition 25.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock 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 properly sampled.

SFR Definition 25.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x89

Bit	Name	Function
7	GATE1	Timer 1 Gate Control. 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 17.5).
6	C/T1	Counter/Timer 1 Select. 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	Timer 1 Mode Select. These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	Timer 0 Gate Control. 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 17.5).
2	C/T0	Counter/Timer 0 Select. 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	Timer 0 Mode Select. These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counters/Timers

26.3.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value 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. An interrupt request is generated if the CCFn interrupt for that module is 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.

Important Note about Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

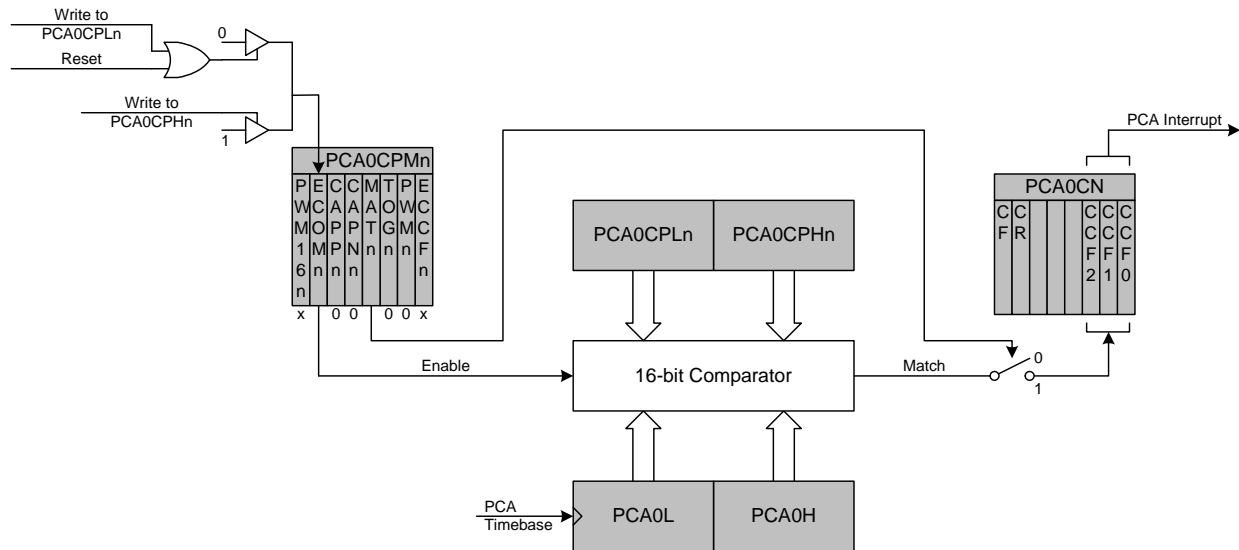


Figure 26.5. PCA Software Timer Mode Diagram

C2 Register Definition 27.2. DEVICEID: C2 Device ID

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

C2 Address: 0x00

Bit	Name	Function
7:0	DEVICEID[7:0]	Device ID. This read-only register returns the 8-bit device ID: 0x10 = C8051T600/1/2/3/4/5 0x1B = C8051T606

C2 Register Definition 27.3. REVID: C2 Revision ID

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

C2 Address: 0x01

Bit	Name	Function
7:0	REVID[7:0]	Revision ID. This read-only register returns the 8-bit revision ID. For example: 0x00 = Revision A.

C2 Register Definition 27.10. CRC0: CRC Byte 0

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

C2 Address: 0xA9

Bit	Name	Function
7:0	CRC[7:0]	CRC Byte 0. A write to this register initiates a 16-bit CRC of one 256-byte block of EPROM memory. The byte written to CRC0 is the upper byte of the 16-bit address where the CRC will begin. The lower byte of the beginning address is always 0x00. When complete, the 16-bit result will be available in CRC1 (MSB) and CRC0 (LSB). See Section "20.3. Program Memory CRC" on page 99.

C2 Register Definition 27.11. CRC1: CRC Byte 1

Bit	7	6	5	4	3	2	1	0
Name	CRC[15:8]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

C2 Address: 0xAA

Bit	Name	Function
7:0	CRC[15:8]	CRC Byte 1. A write to this register initiates a 32-bit CRC on the entire program memory space. The CRC begins at address 0x0000. When complete, the 32-bit result is stored in CRC3 (MSB), CRC2, CRC1, and CRC0 (LSB). See Section "20.3. Program Memory CRC" on page 99.

DOCUMENT CHANGE LIST

Revision 0.5 to Revision 1.0

- Updated electrical specification tables based on test, characterization, and qualification data.
- Updated with new formatting standards.
- Corrected minor typographical errors throughout document.
- Updated wording from “OTP EPROM” to “EPROM” throughout document.
- Added information on C2 EPSTAT Register.
- Updated EPROM programming sequence.
- Added Note about 100% Tin (Sn) lead finish to ordering information table.

Updated packaging information to include JEDEC-standard drawings for package and land diagram.

Revision 1.0 to Revision 1.1

Added C8051T606 device information.

Revision 1.1 to Revision 1.2

- Updated Table 8.4 on page 34.