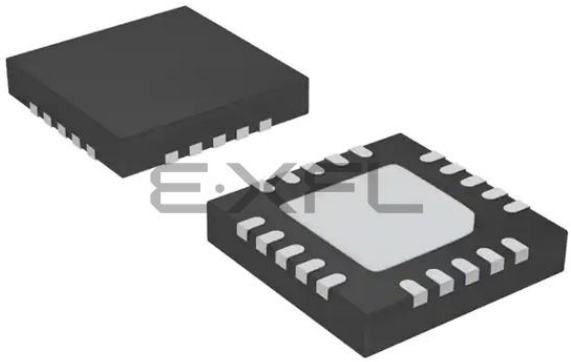


**Welcome to E-XFL.COM**

## 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 | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 25MHz |
| Connectivity | LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT |
| Number of I/O | 16 |
| Program Memory Size | 2KB (2K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 5.25V |
| Data Converters | A/D 16x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 20-VFQFN Exposed Pad |
| Supplier Device Package | 20-QFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/silicon-labs/c8051f536a-im |

## Table 2.2. Global DC Electrical Characteristics

–40 to +125 °C, 25 MHz System Clock unless otherwise specified. Typical values are given at 25 °C

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash) | | | | | |
| Idle $I_{DD}$[3,4] | **$V_{DD}$ = 2.1 V:** | | | | |
| | Clock = 32 kHz | — | 8 | — | µA |
| | Clock = 200 kHz | — | 22 | — | µA |
| | Clock = 1 MHz | — | 0.09 | — | mA |
| | Clock = 25 MHz | — | 2.2 | 5 | mA |
| | **$V_{DD}$ = 2.6 V:** | | | | |
| | Clock = 32 kHz | — | 9 | — | µA |
| | Clock = 200 kHz | — | 30 | — | µA |
| | Clock = 1 MHz | — | 0.13 | — | mA |
| | Clock = 25 MHz | — | 3 | 6.5 | mA |
| Idle $I_{DD}$ Frequency Sensitivity[3,6] | T = 25 °C: | | | | |
| | $V_{DD}$ = 2.1 V, F ≤ 1 MHz | — | 90 | — | µA/MHz |
| | $V_{DD}$ = 2.1 V, F > 1 MHz | — | 90 | — | µA/MHz |
| | $V_{DD}$ = 2.6 V, F ≤ 1 MHz | — | 118 | — | µA/MHz |
| | $V_{DD}$ = 2.6 V, F > 1 MHz | — | 118 | — | µA/MHz |
| Digital Supply Current[3] (Stop or Suspend Mode) | Oscillator not running, $V_{DD}$ Monitor Disabled. | | | | |
| | T = 25 °C | — | 2 | — | µA |
| | T = 60 °C | — | 3 | — | µA |
| | T = 125 °C | — | 50 | — | µA |

**Notes:**

1. For more information on $V_{REGIN}$ characteristics, see Table 2.6 on page 30.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Based on device characterization data; Not production tested.
4. Does not include internal oscillator or internal regulator supply current.
5. $I_{DD}$ can be estimated for frequencies <= 12 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate $I_{DD}$ > 12 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: $V_{DD}$ = 2.6 V; F= 20 MHz, $I_{DD}$ = 7.3 mA – (25 MHz – 20 MHz) x 0.184 mA/MHz = 6.38 mA.
6. Idle $I_{DD}$ can be estimated for frequencies <= 1 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate $I_{DD}$ > 1 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: $V_{DD}$ = 2.6 V; F= 5 MHz, Idle $I_{DD}$ = 3 mA – (25 MHz– 5 MHz) x 118 µA/MHz = 0.64 mA.

**Table 3.1. Pin Definitions for the C8051F52x and C8051F52xA (DFN 10)**

| Name | Pin Numbers | | Type | Description |
|---|---|---|---|---|
| | 'F52xA 'F52x-C | 'F52x | | |
| $\overline{RST}$/ | 1 | 1 | D I/O | Device Reset. Open-drain output of internal POR or $V_{DD}$ monitor. An external source can initiate a system reset by driving this pin low for at least the minimum $\overline{RST}$ low time to generate a system reset, as defined in Table 2.8 on page 32. A 1 kΩ pullup to $V_{REGIN}$ is recommended. See Reset Sources Section for a complete description. |
| C2CK | | | D I/O | Clock signal for the C2 Debug Interface. |
| P0.0/ | 2 | 2 | D I/O or A In | Port 0.0. See Port I/O Section for a complete description. |
| $V_{REF}$ | | | A O or D In | External $V_{REF}$ Input. See $V_{REF}$ Section. |
| GND | 3 | 3 | | Ground. |
| $V_{DD}$ | 4 | 4 | | Core Supply Voltage. |
| $V_{REGIN}$ | 5 | 5 | | On-Chip Voltage Regulator Input. |
| P0.5/RX*/ | 6 | — | D I/O or A In | Port 0.5. See Port I/O Section for a complete description. |
| CNVSTR | | | D In | External Converter start input for the ADC0, see Section "4. 12-Bit ADC (ADC0)" on page 52 for a complete description. |
| P0.5/ | — | 6 | D I/O or A In | Port 0.5. See Port I/O Section for a complete description. |
| CNVSTR | | | D In | External Converter start input for the ADC0, see Section "4. 12-Bit ADC (ADC0)" on page 52 for a complete description. |
| P0.4/TX* | 7 | — | D I/O or A In | Port 0.4. See Port I/O Section for a complete description. |
| P0.4/RX* | — | 7 | D I/O or A In | Port 0.4. See Port I/O Section for a complete description. |
| P0.3 | 8 | — | D I/O or A In | Port 0.3. See Port I/O Section for a complete description. |
| XTAL2 | | | D I/O | External Clock Output. For an external crystal or resonator, this pin is the excitation driver. This pin is the external clock input for CMOS, capacitor, or RC oscillator configurations. See Section "14. Oscillators" on page 135. |

**Note:** Please refer to Section "20. Device Specific Behavior" on page 210.

# 4. 12-Bit ADC (ADC0)

The ADC0 on the C8051F52x/F52xA/F53x/F53xA Family consists of an analog multiplexer (AMUX0) with 16/6 total input selections, and a 200 ksps, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold, programmable window detector, programmable gain, and hardware accumulator. The ADC0 subsystem has a special Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. The AMUX0, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 4.1. ADC0 inputs are single-ended and may be configured to measure P0.0-P1.7, the Temperature Sensor output, $V_{DD}$, or GND with respect to GND. The voltage reference for the ADC is selected as described in Section “5. Voltage Reference” on page 72. ADC0 is enabled when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1, or when performing conversions in Burst Mode. ADC0 is in low power shutdown when AD0EN is logic 0 and no Burst Mode conversions are taking place.

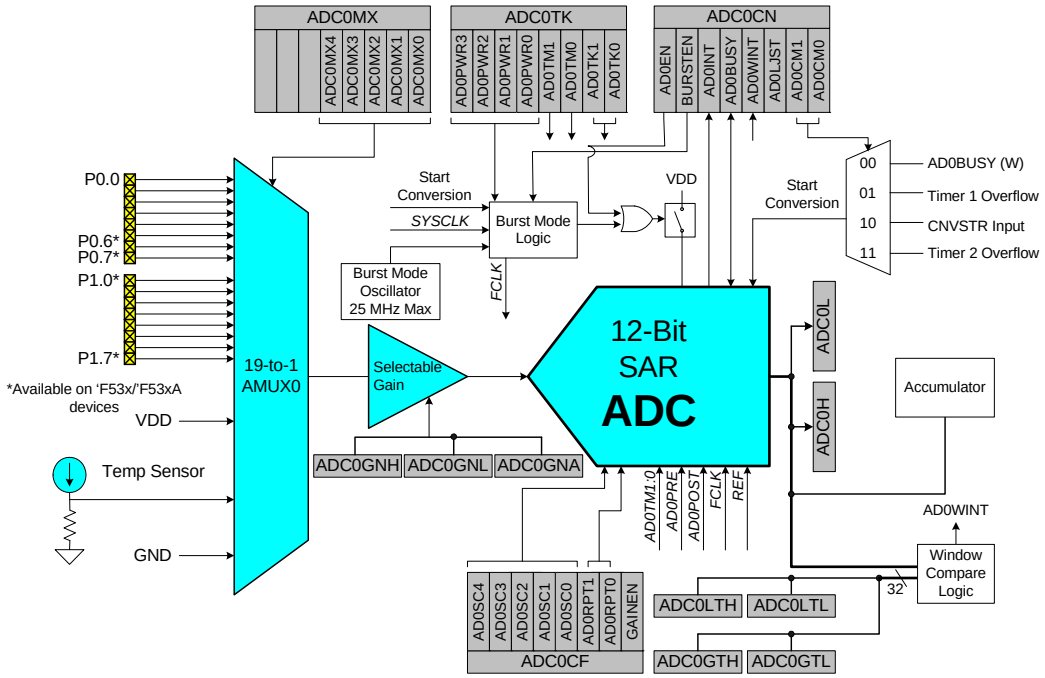



**Figure 4.1. ADC0 Functional Block Diagram**

## 4.1. Analog Multiplexer

AMUX0 selects the input channel to the ADC. Any of the following may be selected as an input: P0.0–P1.7, the on-chip temperature sensor, the core power supply ($V_{DD}$), or ground (GND). **ADC0 is single-ended and all signals measured are with respect to GND.** The ADC0 input channels are selected using the ADC0MX register as described in SFR Definition 4.4.

**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 to 0 the corresponding bit in register PnMDIN (for n = 0,1). To force the Crossbar to skip a Port pin, set to 1 the corresponding bit in register PnSKIP (for n = 0,1). See Section “13. Port Input/Output” on page 120 for more Port I/O configuration details.

### 4.3.4. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a very low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a very low power state, accumulates 1, 4, 8, or 16 samples using an internal Burst Mode Oscillator, then re-enters a very low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a very low power state within a single system clock cycle, even if the system clock is slow (e.g. 32.768 kHz), or suspended.

Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e., the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 4.5 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

**Important Note:** When Burst Mode is enabled, only Post-Tracking and Dual-Tracking modes can be used.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

**Note:** When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.

### Gain Register Definition 4.1. ADC0GNH: ADC0 Selectable Gain High Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| GAINH[7:0] | | | | | | | | 11111100 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Address: |
| | | | | | | | | 0x04 |

**Bits7–0**: High byte of Selectable Gain Word.

### Gain Register Definition 4.2. ADC0GNL: ADC0 Selectable Gain Low Byte

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| GAINL[3:0] | | | | Reserved | Reserved | Reserved | Reserved | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Address: |
| | | | | | | | | 0x07 |

**Bits7–4**: Lower 4 bits of the Selectable Gain Word.
**Bits3–0**: **Reserved**. Must Write 0000b.

### Gain Register Definition 4.3. ADC0GNA: ADC0 Additional Selectable Gain

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | GAINADD | 00000001 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Address: |
| | | | | | | | | 0x08 |

**Bits7–1**: **Reserved**. Must Write 0000000b.
**Bit0**: **GAINADD**: Additional Gain Bit.
Setting this bit adds 1/64 (0.016) gain to the gain value in the ADC0GNH and ADC0GNL registers.

# 7. Comparator

C8051F52x/F52xA/F53x/F53xA devices include one on-chip programmable voltage comparator. The Comparator is shown in Figure 7.1.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0), or an asynchronous “raw” output (CP0A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output with the device in STOP or SUSPEND mode. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see Section “13.2. Port I/O Initialization” on page 126). The Comparator may also be used as a reset source (see Section “11.5. Comparator Reset” on page 110).

The Comparator inputs are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P3–CMX0P0 bits select the Comparator0 positive input; the CMX0N3–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 “13.3. General Purpose Port I/O” on page 128).

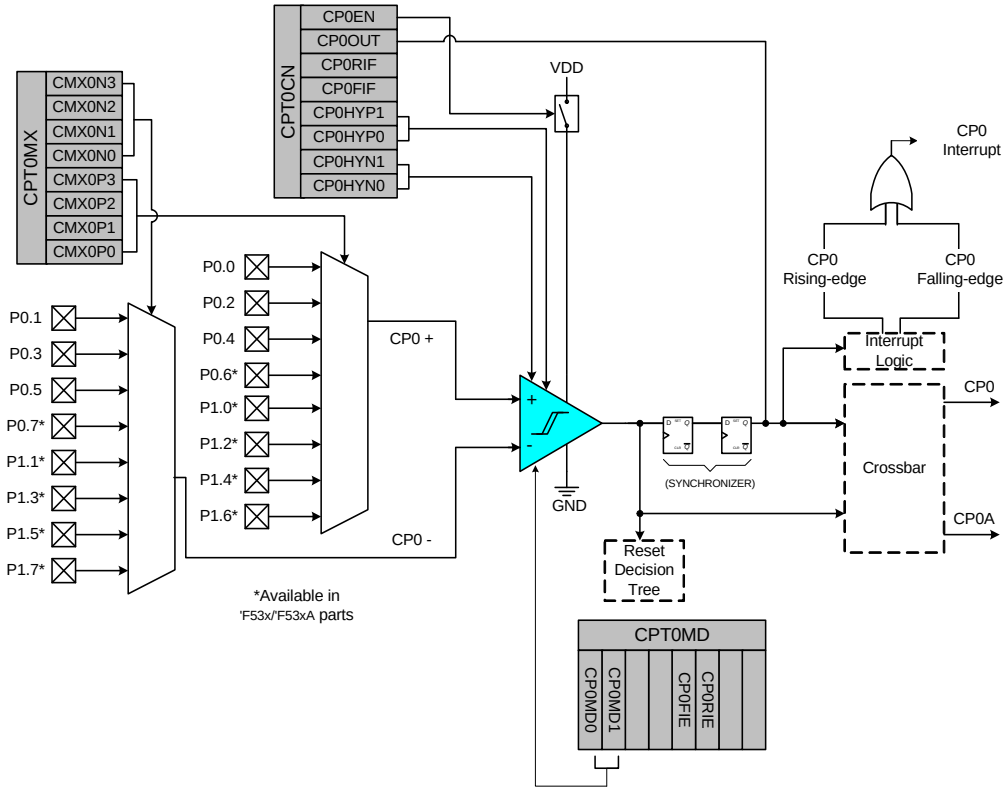



**Figure 7.1. Comparator Functional Block Diagram**

The Comparator output can be polled in software, used as an interrupt source, internal oscillator suspend awakening source and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP or SUSPEND mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to

**Table 8.1. CIP-51 Instruction Set Summary (Continued)**

| Mnemonic | Description | Bytes | Clock Cycles |
|---|---|---|---|
| **Boolean Manipulation** | | | |
| CLR C | Clear Carry | 1 | 1 |
| CLR bit | Clear direct bit | 2 | 2 |
| SETB C | Set Carry | 1 | 1 |
| SETB bit | Set direct bit | 2 | 2 |
| CPL C | Complement Carry | 1 | 1 |
| CPL bit | Complement direct bit | 2 | 2 |
| ANL C, bit | AND direct bit to Carry | 2 | 2 |
| ANL C, /bit | AND complement of direct bit to Carry | 2 | 2 |
| ORL C, bit | OR direct bit to carry | 2 | 2 |
| ORL C, /bit | OR complement of direct bit to Carry | 2 | 2 |
| MOV C, bit | Move direct bit to Carry | 2 | 2 |
| MOV bit, C | Move Carry to direct bit | 2 | 2 |
| JC rel | Jump if Carry is set | 2 | 2/3 |
| JNC rel | Jump if Carry is not set | 2 | 2/3 |
| JB bit, rel | Jump if direct bit is set | 3 | 3/4 |
| JNB bit, rel | Jump if direct bit is not set | 3 | 3/4 |
| JBC bit, rel | Jump if direct bit is set and clear bit | 3 | 3/4 |
| **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 | 4/5 |
| 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 |

## SFR Definition 10.4. EIP1: Extended Interrupt Priority 1

| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | Reset Value |
|---|---|---|---|---|---|---|---|---|
| PMAT | PREG0 | PLIN | PCPR | PCPF | PPAC0 | PREG0 | PWADC0 | 00000000 |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | |

SFR Address: 0xF6

**Bit7**: **PMAT**. Port Match Interrupt Priority Control.
This bit sets the priority of the Port Match interrupt.
0: Port Match interrupt set to low priority level.
1: Port Match interrupt set to high priority level.

**Bit6**: **PREG0**: Voltage Regulator Interrupt Priority Control.
This bit sets the priority of the Voltage Regulator interrupt.
0: Voltage Regulator interrupt set to low priority level.
1: Voltage Regulator interrupt set to high priority level.

**Bit5**: **PLIN**: LIN Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: LIN interrupt set to low priority level.
1: LIN interrupt set to high priority level.

**Bit4**: **PCPR**: Comparator Rising Edge Interrupt Priority Control.
This bit sets the priority of the Rising Edge Comparator interrupt.
0: Comparator interrupt set to low priority level.
1: Comparator interrupt set to high priority level.

**Bit3**: **PCPF**: Comparator falling Edge Interrupt Priority Control.
This bit sets the priority of the Falling Edge Comparator interrupt.
0: Comparator interrupt set to low priority level.
1: Comparator interrupt set to high priority level.

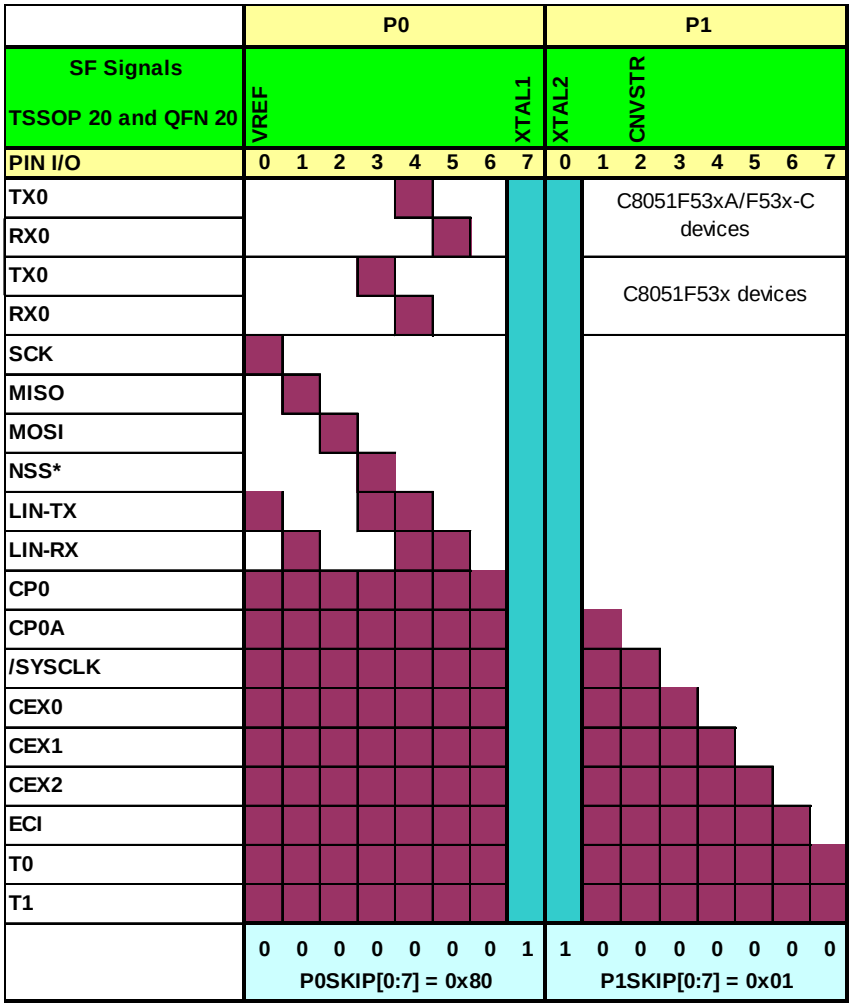
**Bit2**: **PPAC0**: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority level.
1: PCA0 interrupt set to high priority level.

**Bit1**: **PREG0**: ADC0 Conversion Complete Interrupt Priority Control.
This bit sets the priority of the ADC0 Conversion Complete interrupt.
0: ADC0 Conversion Complete interrupt set to low priority level.
1: ADC0 Conversion Complete interrupt set to high priority level.

**Bit0**: **PWADC0**: ADC0 Window Comparison Interrupt Priority Control.
This bit sets the priority of the ADC0 Window Comparison interrupt.
0: ADC0 Window Comparison interrupt set to low priority level.
1: ADC0 Window Comparison interrupt set to high priority level.

(F52x/F52xA) for the external CNVSTR signal, and any selected ADC or comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 13.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP); Figure 13.4 shows the Crossbar Decoder priority with the XTAL1 (P1.0) and XTAL2 (P1.1) pins skipped (P1SKIP = 0x03).

**Important Note on UART Pins**: On C8051F52xA/F52x-C/F53xA/F53x-C devices, the UART pins must be skipped if the UART is enabled in order for peripherals to appear on port pins beyond the UART on the crossbar. For example, with the SPI and UART enabled on the crossbar with the SPI on P1.0-P1.3, the UART pins must be skipped using P0SKIP for the SPI pins to appear correctly.




**Note:** 4-Wire SPI Only.

**Figure 13.4. Crossbar Priority Decoder with Crystal Pins Skipped (TSSOP 20 and QFN 20)**

# 14. Oscillators

C8051F52x/F52xA/F53x/F53xA devices include a programmable internal oscillator, an external oscillator drive circuit. The internal oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 14.1. The system clock (SYSCLK) can be derived from the internal oscillator, external oscillator circuit. Oscillator electrical specifications are given in Table 2.11 on page 34.

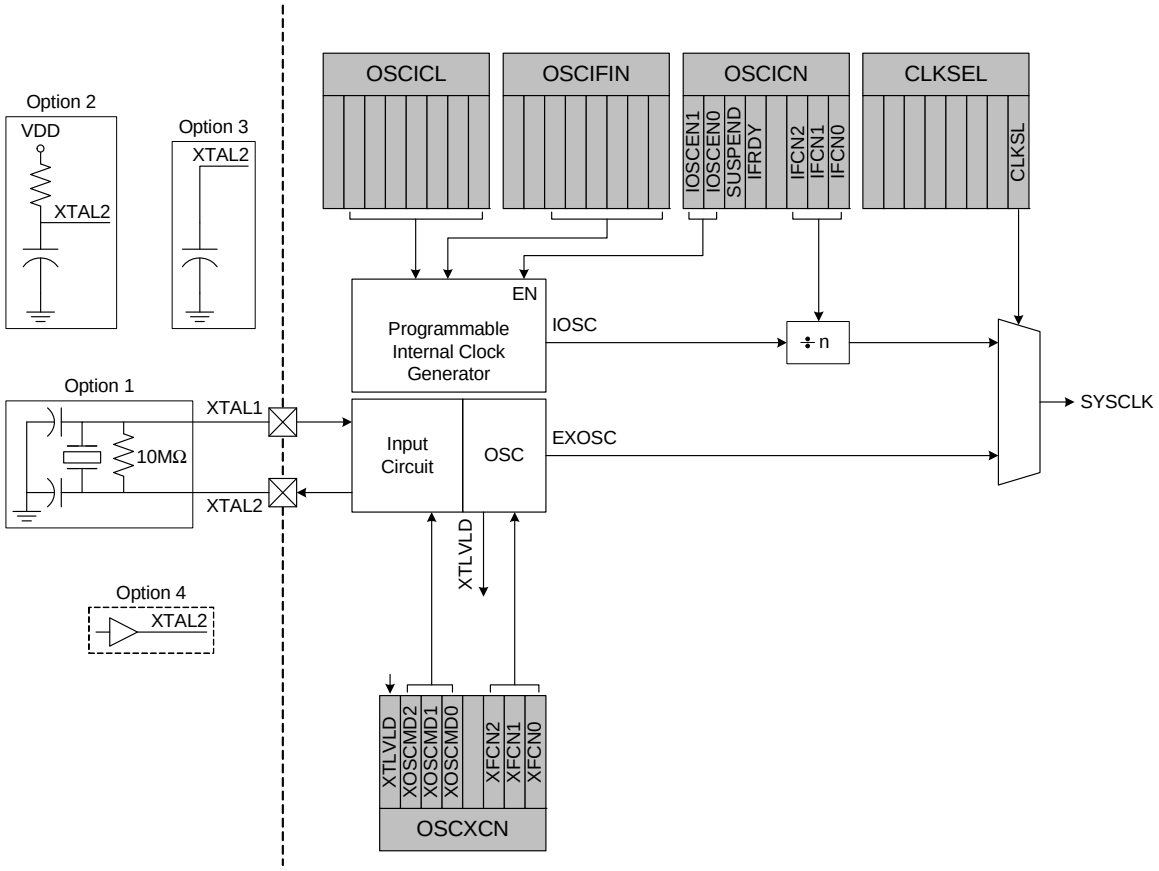



**Figure 14.1. Oscillator Diagram**

## 14.1. Programmable Internal Oscillator

All C8051F52x/53x devices include a programmable internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be programmed via the OSCICL and OSCIFIN registers, shown in SFR Definition 14.2 and SFR Definition 14.3. On C8051F52x/53x devices, OSCICL and OSCIFIN are factory calibrated to obtain a 24.5 MHz frequency.

Electrical specifications for the precision internal oscillator are given in Table 2.11 on page 34. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, 8, 16, 32, 64, or 128 as defined by the IFCN bits in register OSCICN. The divide value defaults to 128 following a reset.

**Note:** The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

The equation for determining the load capacitance for two capacitors is:

$$C_L = \frac{C_A \times C_B}{C_A + C_B} + C_S$$

Where:
$C_A$ and $C_B$ are the capacitors connected to the crystal leads.
$C_S$ is the total stray capacitance of the PCB.
The stray capacitance for a typical layout where the crystal is as close as possible to the pins is 2–5 pF per pin.

If $C_A$ and $C_B$ are the same (C), then the equation becomes:

$$C_L = \frac{C}{2} + C_S$$

For example, a tuning-fork crystal of 32 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 14.1, Option 1. With a stray capacitance of 3 pF per pin (6 pF total), the 13 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 14.2.

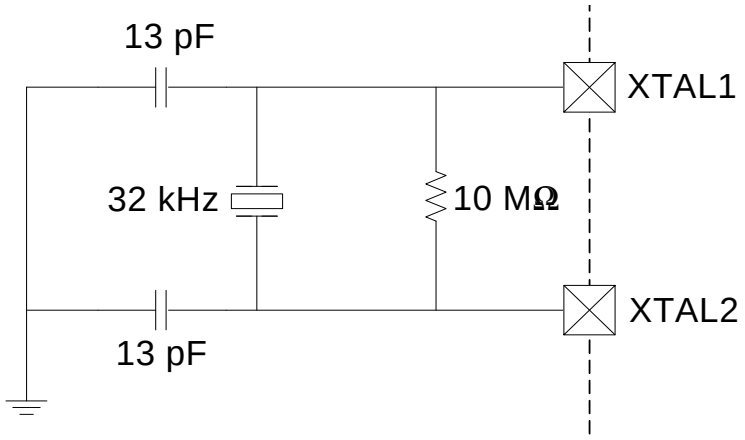



**Figure 14.2. 32 kHz External Crystal Example**

**Important Note on External Crystals:** 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. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

# 15. 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 “15.1. Enhanced Baud Rate Generation” on page 145). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte. (Please refer to Section “20. Device Specific Behavior” on page 210 for more information on the pins associated with the UART interface.)

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).

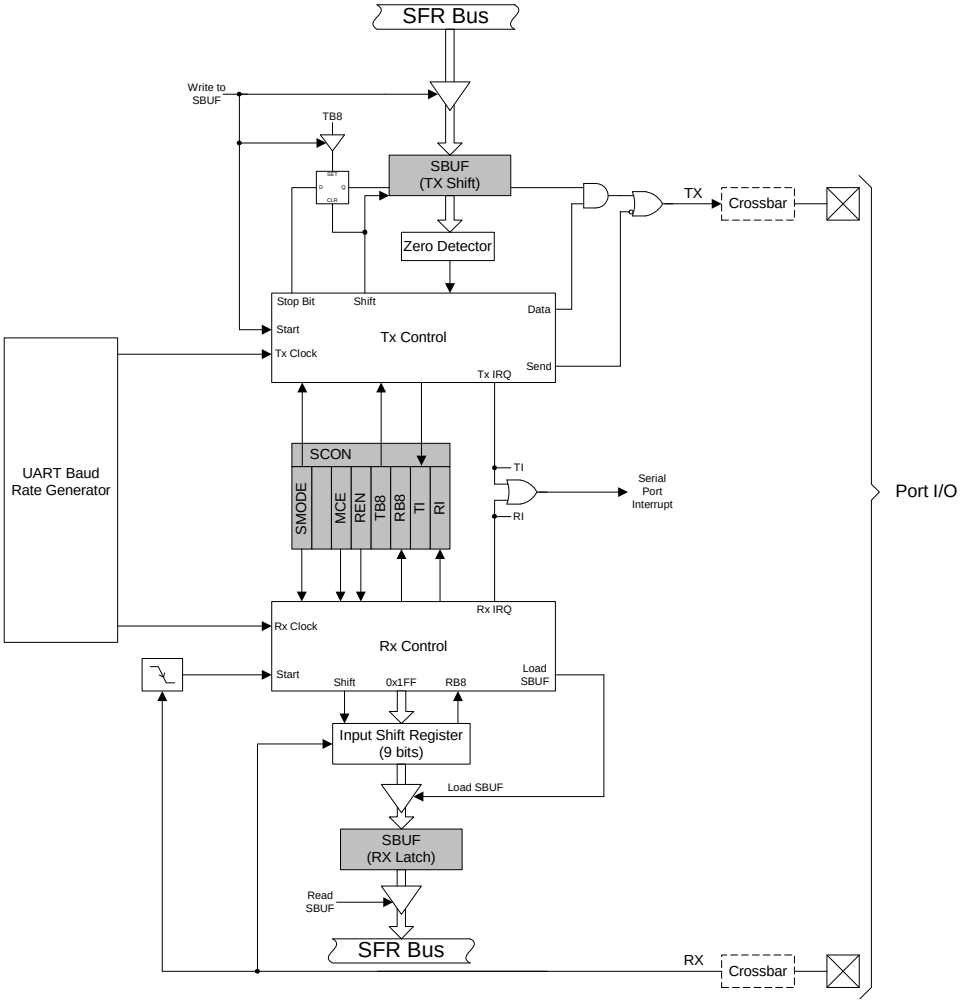



**Figure 15.1. UART0 Block Diagram**

## 16.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 16.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

### 16.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

### 16.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

### 16.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 16.2, Figure 16.3, and Figure 16.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “13. Port Input/Output” on page 120 for general purpose port I/O and crossbar information.

**Table 17.1. Baud-Rate Calculation Variable Ranges**

| Factor | Range |
|---|---|
| prescaler | 0…3 |
| multiplier | 0…31 |
| divider | 200…511 |

**Important**: The minimum system clock (SYSCLK) to operate the LIN peripheral is 8 MHz.

Use the following equations to calculate the values for the variables for the baud-rate equation:

$$multiplier = \frac{20000}{baud\_rate} - 1$$

$$prescaler = ln\left[\frac{SYSCLK}{(multiplier + 1) \times baud\_rate \times 200}\right] \times \frac{1}{ln2} - 1$$

$$divider = \frac{SYSCLK}{(2^{(\text{prescaler} + 1)} \times (multiplier + 1) \times baud\_rate)}$$

It is important to note that in all these equations, the results must be rounded down to the nearest integer.

The following example shows the steps for calculating the baud rate values for a Master node running at 24.5 MHz and communicating at 19200 bits/sec. First, calculate the multiplier:

$$multiplier = \frac{20000}{19200} - 1 = 0.0417 \cong 0$$

Next, calculate the prescaler:

$$prescaler = ln\frac{24500000}{(0 + 1) \times 19200 \times 200} \times \frac{1}{ln2} - 1 = 1.674 \cong 1$$

Finally, calculate the divider:

$$divider = \frac{24500000}{2^{(1 + 1)} \times (0 + 1) \times 19200} = 319.010 \cong 319$$

These values lead to the following baud rate:

$$baud\_rate = \frac{24500000}{2^{(1 + 1)} \times (0 + 1) \times 319} \cong 19200.63$$

# 18. 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 other device peripherals 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 offer 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 18.3 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's frequency can be counted. The input signal need not be periodic, but it must be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

## 18.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section “10.4. Interrupt Register Descriptions” on page 100); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section 10.4). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

### 18.1.1. Mode 0: 13-bit Counter/Timer

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

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

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section “13.1. Priority Crossbar Decoder” on page 122 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is

### 19.3.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2-CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 2 is forced into software timer mode.
- Writes to the Module 2 mode register (PCA0CPM2) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH2 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH2. Upon a PCA0CPH2 write, PCA0H plus the offset held in PCA0CPL2 is loaded into PCA0CPH2 (See Figure 19.10).

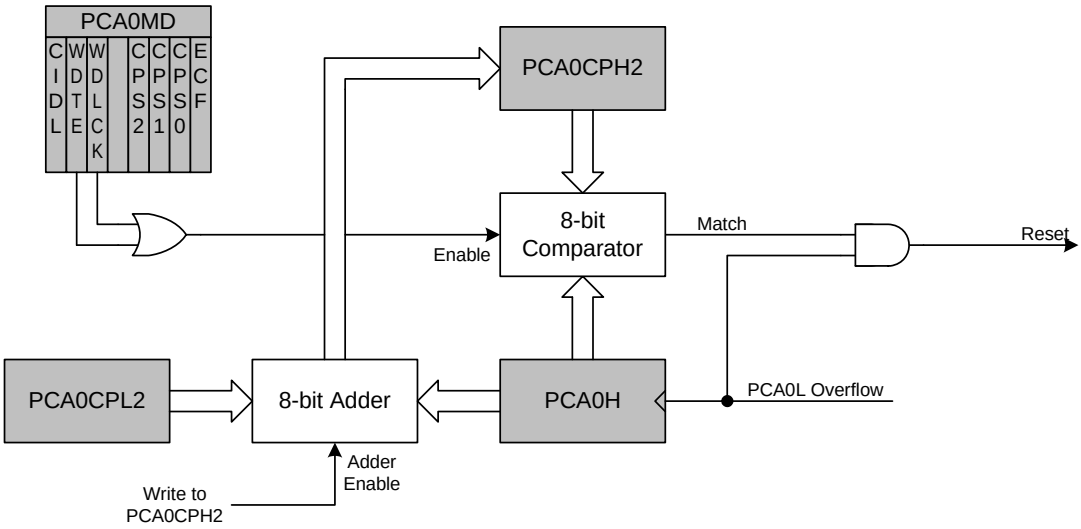



**Figure 19.10. PCA Module 2 with Watchdog Timer Enabled**

Note that the 8-bit offset held in PCA0CPH2 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 19.4, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL2) + (256 - PCA0L)$$

**Equation 19.4. Watchdog Timer Offset in PCA Clocks**

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH2 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF2 flag (PCA0CN.2) while the WDT is enabled.

# 20. Device Specific Behavior

This chapter contains behavioral differences between the silicon revisions of C8051F52x/52xA/F53x/53xA devices.

These differences do not affect the functionality or performance of most systems and are described below.

## 20.1. Device Identification

The Part Number Identifier on the top side of the device package can be used for decoding device information. The first character of the trace code identifies the silicon revision. On C8051F52x-C/53x-C devices, the trace code (second line on the TSSOP-20 and DFN-10 packages; third line on the QFN-20 package) will begin with the letter "C". The "A" suffix at the end of the part number such as "C8051F530A" is only present on Revision B devices. All other revisions do not include this suffix. Figures 20.1, 20.2, and 20.3 show how to find the part number on the top side of the device package.

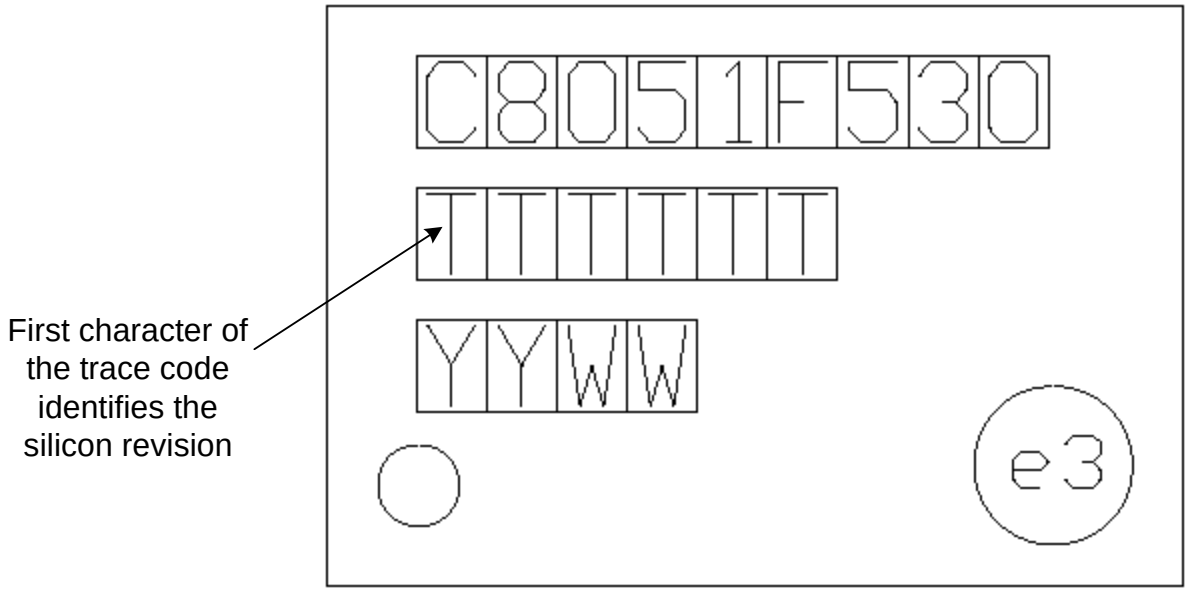



**Figure 20.1. Device Package—TSSOP 20**

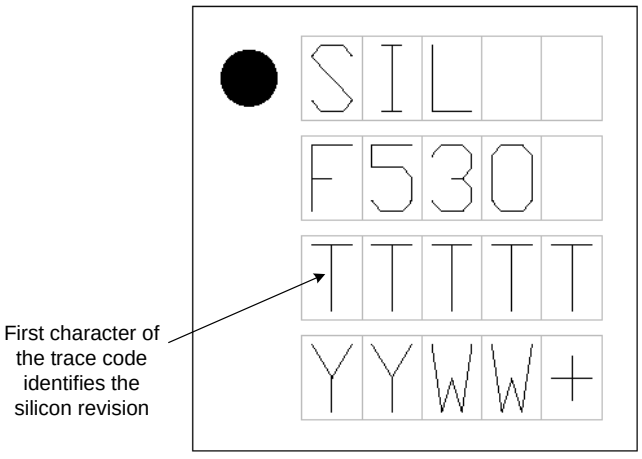



**Figure 20.2. Device Package—QFN 20**

## C2 Register Definition 21.3. REVID: C2 Revision ID

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | Varies |

This read-only register returns the 8-bit revision ID.
For example, 0x00 = Revision A.

## C2 Register Definition 21.4. FPCTL: C2 Flash Programming Control

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |

**Bits7–0** **FPCTL**: Flash Programming Control Register.
This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

## C2 Register Definition 21.5. FPDAT: C2 Flash Programming Data

| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Reset Value |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 00000000 |

**Bits7–0**: **FPDAT**: C2 Flash Programming Data Register.
This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.

| Code | Command |
|---|---|
| 0x06 | Flash Block Read |
| 0x07 | Flash Block Write |
| 0x08 | Flash Page Erase |
| 0x03 | Device Erase |