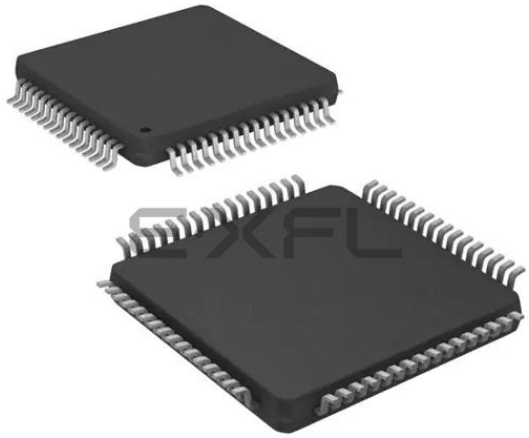

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, Temp Sensor, WDT
Number of I/O	54
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f702-gq

SFR Definition 21.5. EIP1: Extended Interrupt Priority 1 144
SFR Definition 21.6. EIP2: Extended Interrupt Priority 2 145
SFR Definition 21.7. IT01CF: INT0/INT1 Configuration 147
SFR Definition 22.1. PSCTL: Program Store R/W Control 153
SFR Definition 22.2. FLKEY: Flash Lock and Key 154
SFR Definition 23.1. EEADDR: EEPROM Byte Address 156
SFR Definition 23.2. EEDATA: EEPROM Byte Data 157
SFR Definition 23.3. EECNTL: EEPROM Control 158
SFR Definition 23.4. EEKEY: EEPROM Protect Key 159
SFR Definition 24.1. PCON: Power Control 162
SFR Definition 25.1. VDM0CN: VDD Monitor Control 166
SFR Definition 25.2. RSTSRC: Reset Source 168
SFR Definition 26.1. WDTCN: Watchdog Timer Control 170
SFR Definition 27.1. CLKSEL: Clock Select 172
SFR Definition 27.2. OSCICL: Internal H-F Oscillator Calibration 173
SFR Definition 27.3. OSCICN: Internal H-F Oscillator Control 174
SFR Definition 27.4. OSCXCN: External Oscillator Control 176
SFR Definition 28.1. XBR0: Port I/O Crossbar Register 0 190
SFR Definition 28.2. XBR1: Port I/O Crossbar Register 1 191
SFR Definition 28.3. P0MASK: Port 0 Mask Register 192
SFR Definition 28.4. P0MAT: Port 0 Match Register 193
SFR Definition 28.5. P1MASK: Port 1 Mask Register 193
SFR Definition 28.6. P1MAT: Port 1 Match Register 194
SFR Definition 28.7. P0: Port 0 195
SFR Definition 28.8. P0MDIN: Port 0 Input Mode 195
SFR Definition 28.9. P0MDOUT: Port 0 Output Mode 196
SFR Definition 28.10. P0SKIP: Port 0 Skip 196
SFR Definition 28.11. P0DRV: Port 0 Drive Strength 197
SFR Definition 28.12. P1: Port 1 197
SFR Definition 28.13. P1MDIN: Port 1 Input Mode 198
SFR Definition 28.14. P1MDOUT: Port 1 Output Mode 198
SFR Definition 28.15. P1SKIP: Port 1 Skip 199
SFR Definition 28.16. P1DRV: Port 1 Drive Strength 199
SFR Definition 28.17. P2: Port 2 200
SFR Definition 28.18. P2MDIN: Port 2 Input Mode 200
SFR Definition 28.19. P2MDOUT: Port 2 Output Mode 201
SFR Definition 28.20. P2SKIP: Port 2 Skip 201
SFR Definition 28.21. P2DRV: Port 2 Drive Strength 202
SFR Definition 28.22. P3: Port 3 202
SFR Definition 28.23. P3MDIN: Port 3 Input Mode 203
SFR Definition 28.24. P3MDOUT: Port 3 Output Mode 203
SFR Definition 28.25. P3DRV: Port 3 Drive Strength 204
SFR Definition 28.26. P4: Port 4 204
SFR Definition 28.27. P4MDIN: Port 4 Input Mode 205
SFR Definition 28.28. P4MDOUT: Port 4 Output Mode 205

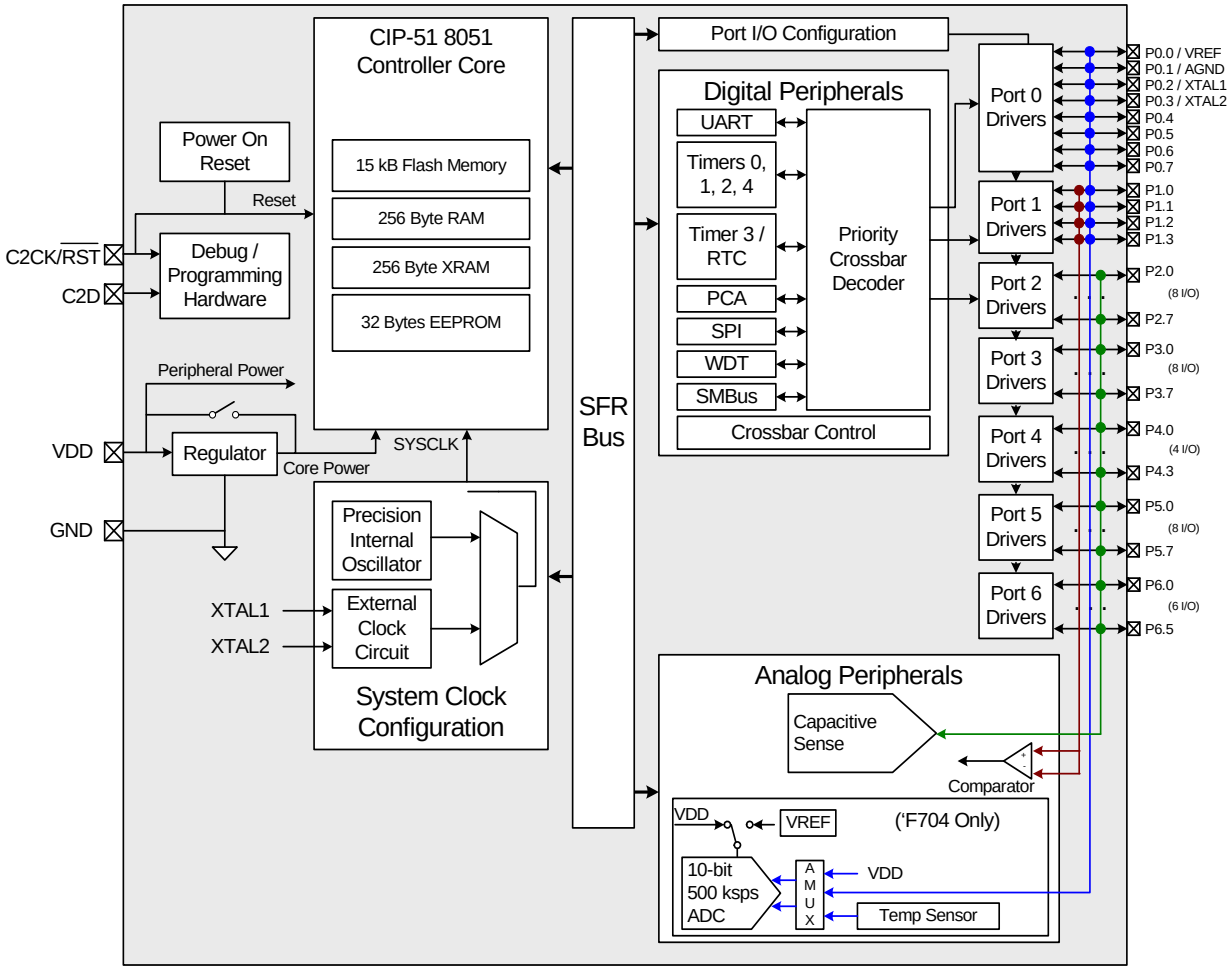

Figure 1.3. C8051F704/5 Block Diagram

11. Temperature Sensor

An on-chip temperature sensor is included on the C8051F700/2/4/6/8 and C8051F710/2/4/6 which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC mux channel should be configured to connect to the temperature sensor. The temperature sensor transfer function is shown in Figure 11.1. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 12.1. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 9.12 for the slope and offset parameters of the temperature sensor.

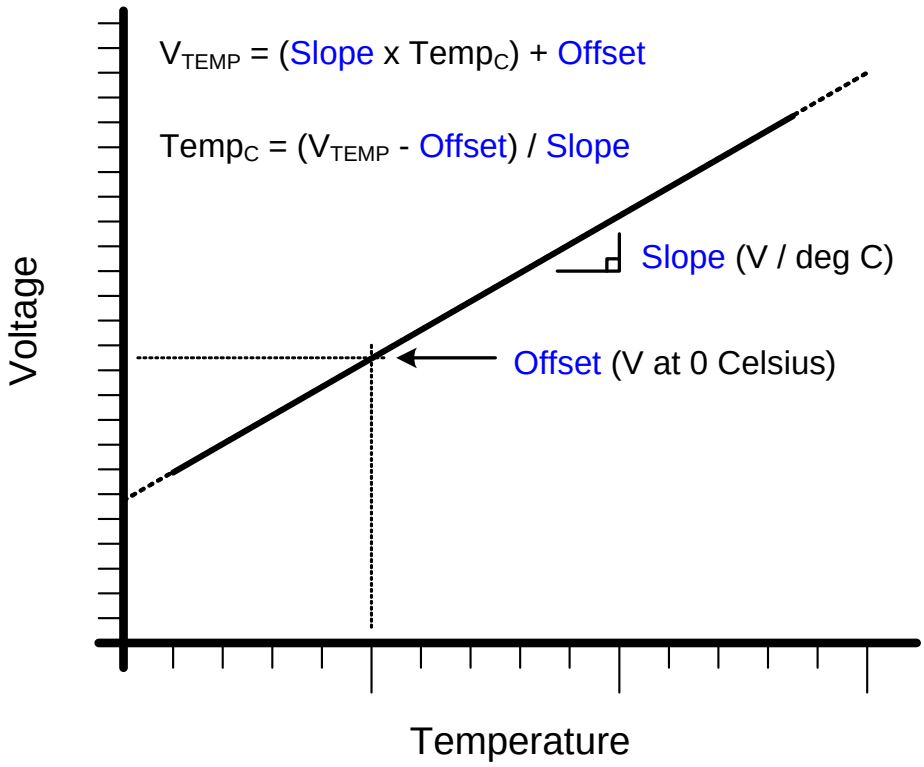

Figure 11.1. Temperature Sensor Transfer Function

11.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.1 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the ADC's input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.3 shows the typical temperature sensor error assuming a 1-point calibration at 0 °C.

12.1. External Voltage References

To use an external voltage reference, REFSL[1:0] should be set to 00. Bypass capacitors should be added as recommended by the manufacturer of the external voltage reference.

12.2. Internal Voltage Reference Options

A 1.6 V high-speed reference is included on-chip. The high speed internal reference is selected by setting REFSL[1:0] to 11. When selected, the high-speed internal reference will be automatically enabled on an as-needed basis by ADC0.

For applications with a non-varying power supply voltage, using the power supply as the voltage reference can provide ADC0 with added dynamic range at the cost of reduced power supply noise rejection. To use the 1.8 to 3.6 V power supply voltage (V_{DD}) or the 1.8 V regulated digital supply voltage as the reference source, REFSL[1:0] should be set to 01 or 10, respectively.

12.3. Analog Ground Reference

To prevent ground noise generated by switching digital logic from affecting sensitive analog measurements, a separate analog ground reference option is available. When enabled, the ground reference for ADC0 is taken from the P0.1/AGND pin. Any external sensors sampled by ADC0 should be referenced to the P0.1/AGND pin. The separate analog ground reference option is enabled by setting REFGND to 1. Note that when using this option, P0.1/AGND must be connected to the same potential as GND.

12.4. Temperature Sensor Enable

The TEMPE bit in register REF0CN enables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data.

Table 18.2. EMIF Pinout (C8051F700/1/2/3/8/9 and C8051F710/1)

Multiplexed Mode	
Signal Name	Port Pin
$\overline{RD}$	P6.1
$\overline{WR}$	P6.0
ALE	P6.2
D0/A0	P5.0
D1/A1	P5.1
D2/A2	P5.2
D3/A3	P5.3
D4/A4	P5.4
D5/A5	P5.5
D6/A6	P5.6
D7/A7	P5.7
A8	P4.0
A9	P4.1
A10	P4.2
A11	P4.3
A12	P4.4
A13	P4.5
A14	P4.6
A15	P4.7
—	—
—	—
—	—
—	—
—	—
—	—
—	—

Non Multiplexed Mode	
Signal Name	Port Pin
$\overline{RD}$	P6.1
$\overline{WR}$	P6.0
D0	P5.0
D1	P5.1
D2	P5.2
D3	P5.3
D4	P5.4
D5	P5.5
D6	P5.6
D7	P5.7
A0	P4.0
A1	P4.1
A2	P4.2
A3	P4.3
A4	P4.4
A5	P4.5
A6	P4.6
A7	P4.7
A8	P3.0
A9	P3.1
A10	P3.2
A11	P3.3
A12	P3.4
A13	P3.5
A14	P3.6
A15	P3.7

SFR Definition 19.3. REVID: Hardware Revision Identification Byte

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

SFR Address = 0xAD; SFR Page = F

Bit	Name	Description
7:0	**REVID[7:0]**	**Hardware Revision Identification Byte.** Shows the C8051F70x/71x hardware revision being used. For example, 0x00 = Revision A.

SFR Definition 21.2. IP: Interrupt Priority

Bit	7	6	5	4	3	2	1	0
Name		PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

SFR Address = 0xB8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	Unused	Read = 1b, Write = Don't Care.
6	PSPI0	**Serial Peripheral Interface (SPI0) Interrupt Priority Control.** This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	**Timer 2 Interrupt Priority Control.** This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	**UART0 Interrupt Priority Control.** This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	**Timer 1 Interrupt Priority Control.** This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	**External Interrupt 1 Priority Control.** This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	**Timer 0 Interrupt Priority Control.** This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	**External Interrupt 0 Priority Control.** This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

22.4.3. System Clock

12. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in “AN201: Writing to Flash from Firmware,″ available from the Silicon Laboratories web site.

23. EEPROM

C8051F700/1/4/5/8/9 and C8051F712/3 devices have hardware which emulates 32 bytes of non-volatile, byte-programmable EEPROM data space. The module mirrors each non-volatile byte through 32 bytes of volatile data space. This data space can be accessed indirectly through EEADDR and EEDATA. Users can copy the complete 32-byte image between EEPROM space and volatile space using controls in the EECNTL SFR.

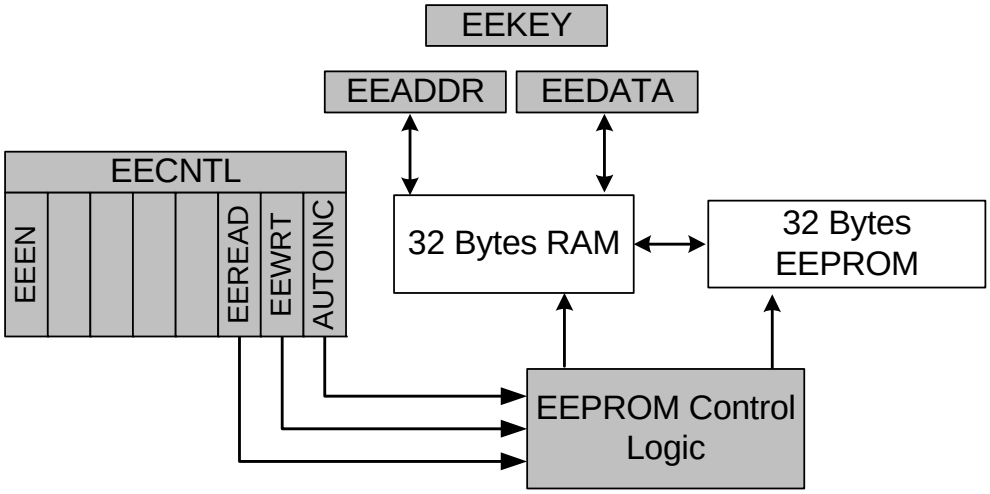

Figure 23.1. EEPROM Block Diagram

23.1. RAM Reads and Writes

In order to perform EEPROM reads and writes, the EEPROM control logic must be enabled by setting EEEN (EECNTL.7).

32 bytes of RAM can be accessed indirectly through EEADDR and EEDATA. To write to a byte of RAM, write address of byte to EEADDR and then write the value to be written to EEDATA. To read a byte from RAM, write address of byte to be read to EEADDR. The value stored at that address can then be read from EEDATA.

23.2. Auto Increment

When AUTOINC (EECNTL.0) is set, EEADDR will increment by one after each write to EEDATA and each read from EEDATA. When Auto Increment is enabled and EEADDR reaches the top address of dedicated RAM space, the next write to or read from EEDATA will cause EEADDR to wrap along the address boundary, which will set the address to 0.

23.3. Interfacing with the EEPROM

The EEPROM is accessed through the dedicated 32 bytes of RAM. Writes to EEPROM are allowed only after writes have been enabled (see “23.4. EEPROM Security”). The contents of the EEPROM can be uploaded to the RAM by setting EEREAD (EECNTL.2). Contents of RAM can be downloaded to EEPROM by setting EEWRT (EENTL.1).

Note: A minimum SYSCLK frequency is required for writing EEPROM memory, as detailed in Section “Table 9.9. EEPROM Electrical Characteristics” on page 52.

23.4. EEPROM Security

RAM can only be downloaded to EEPROM after firmware writes a sequence of two bytes to EEKEY. In order to enable EEPROM writes:

1. Write the first EEPROM key code byte to EEKEY: 0x55
2. Write the second EEPROM key code byte to EEKEY: 0xAA

After a EEPROM writes have been enabled and a single write has executed, the control logic locks EEPROM writes until the two-byte unlock sequence has been entered into EEKEY again.

The protection state of the EEPROM can be observed by reading EEPSTATE (EEKEY2:0). This state can be read at any time without affecting the EEPROM's protection state.

If the two-byte unlock sequence is entered incorrectly, or if a write is attempted without first entering the two-byte sequence, EEPROM writes will be locked until the next power-on reset.

SFR Definition 23.1. EEADDR: EEPROM Byte Address

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

SFR Address = 0xB6; SFR Page = All Pages

Bit	Name	Description
7:5	Unused	Read = 000b; Write = Don't Care
4:0	EEADDR[4:0]	**EEPROM Byte Address** Selects one of 32 EEPROM bytes to read/write.

SFR Definition 27.1. CLKSEL: Clock Select

Bit	7	6	5	4	3	2	1	0
Name	CLKRDY	CLKDIV[2:0]			Reserved	CLKSEL[2:0]		
Type	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBD; SFR Page= F

Bit	Name	Function
7	CLKRDY	**System Clock Divider Clock Ready Flag.** 0: The selected clock divide setting has not been applied to the system clock. 1: The selected clock divide setting has been applied to the system clock.
6:4	CLKDIV	**System Clock Divider Bits.** Selects the clock division to be applied to the selected source (internal or external). 000: Selected clock is divided by 1. 001: Selected clock is divided by 2. 010: Selected clock is divided by 4. 011: Selected clock is divided by 8. 100: Selected clock is divided by 16. 101: Selected clock is divided by 32. 110: Selected clock is divided by 64. 111: Selected clock is divided by 128.
3	Reserved	Read = 0b. Must write 0b.
2:0	CLKSEL[2:0]	**System Clock Select.** Selects the oscillator to be used as the undivided system clock source. 000: Internal Oscillator 001: External Oscillator All other values reserved.

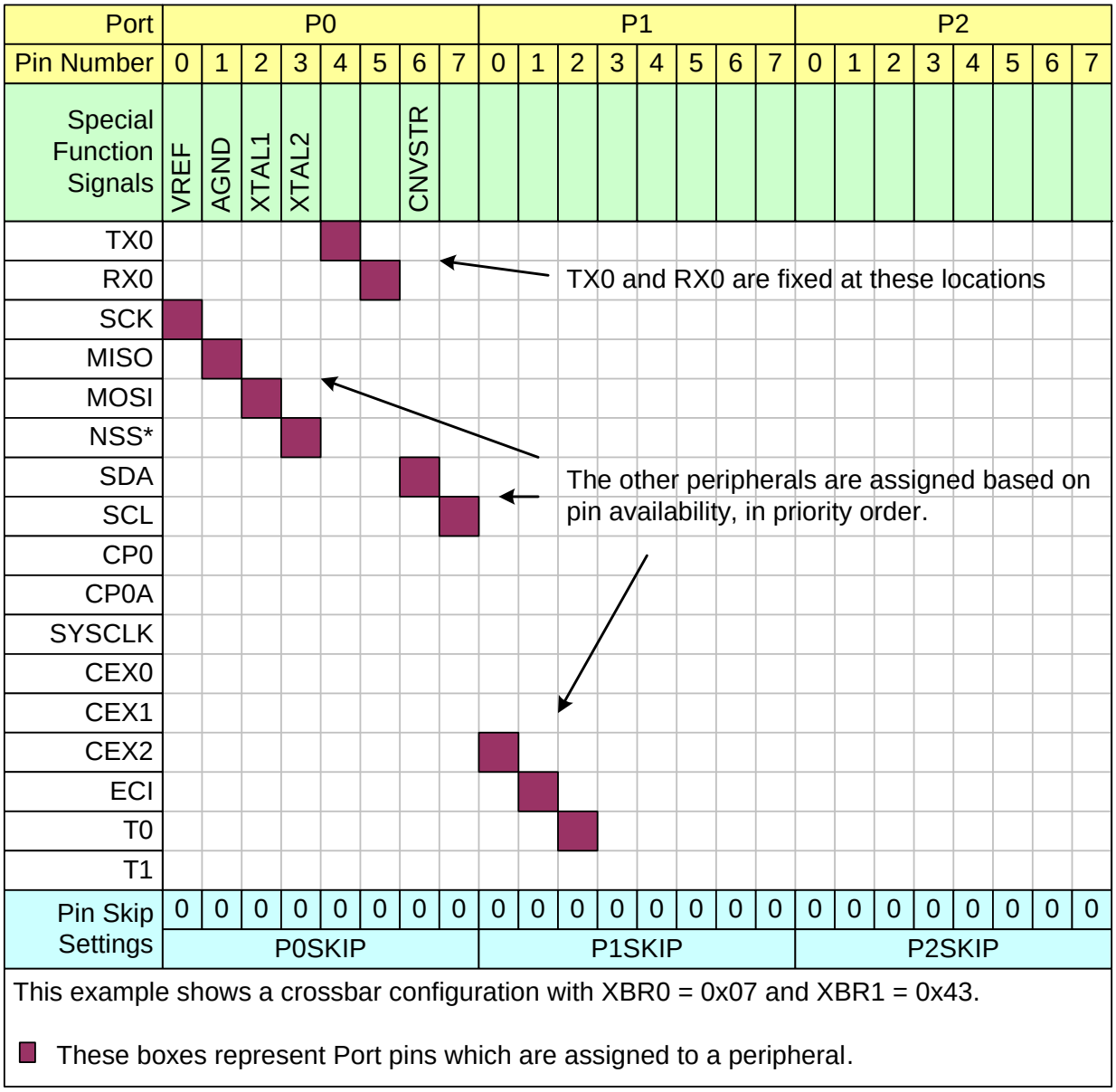

Figure 28.5. Crossbar Priority Decoder in Example Configuration—No Pins Skipped

SFR Definition 28.29. P4DRV: Port 4 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P4DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xFD; SFR Page = F

Bit	Name	Function
7:0	P4DRV[7:0]	**Drive Strength Configuration Bits for P4.7–P4.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P4.n Output has low output drive strength. 1: Corresponding P4.n Output has high output drive strength.

SFR Definition 28.30. P5: Port 5

Bit	7	6	5	4	3	2	1	0
Name	P5[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB3; SFR Page = All Pages

Bit	Name	Description	Write	Read
7:0	P5[7:0]	**Port 5 Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P5.n Port pin is logic LOW. 1: P5.n Port pin is logic HIGH.

29.2. 32-bit CRC Algorithm

The C8051F70x/71x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

1. XOR the least-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x00000000 or 0xFFFFFFFF).
2. Right-shift the CRC result.
3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit C8051F70x/71x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input){
   unsigned char i; // loop counter
   #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ CRC_input;
   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x00000001) == 0x00000001)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc >> 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc >> 1;
      }
   }
   return CRC_acc; // Return the final remainder (CRC value)
}
```

Table 29.2 lists example input values and the associated outputs using the 32-bit C8051F70x/71x CRC algorithm (an initial value of 0xFFFFFFFF is used):

Table 29.2. Example 32-bit CRC Outputs

Input	Output
0x63	0xF9462090
0xAA, 0xBB, 0xCC	0x41B207B3
0x00, 0x00, 0xAA, 0xBB, 0xCC	0x78D129BC

SFR Definition 31.3. SPI0CKR: SPI0 Clock Rate

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

SFR Address = 0xA2; SFR Page = F

Bit	Name	Function
7:0	SCR[7:0]	**SPI0 Clock Rate.** These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR[7:0] + 1)}$ for 0 <= SPI0CKR <= 255 Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 31.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA3; SFR Page = 0

Bit	Name	Function
7:0	SPI0DAT[7:0]	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

32.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 32.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

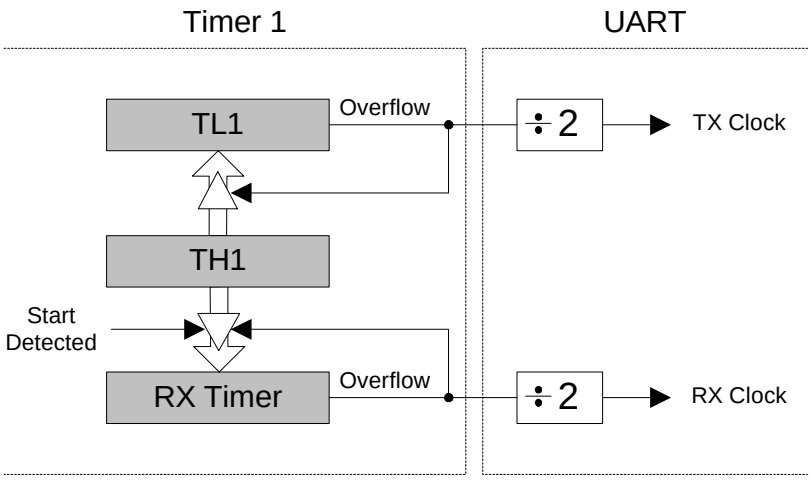

Figure 32.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “33.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 265). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK/4, SYSCLK/12, SYSCLK/48, the external oscillator clock/8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 32.1-A and Equation 32.1-B.

A) $$\text{UartBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

B) $$\text{T1\_Overflow\_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$

Equation 32.1. UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section “33. Timers” on page 262. A quick reference for typical baud rates and system clock frequencies is given in Table 32.1 through Table 32.2. The internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

32.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown in Figure 32.3.

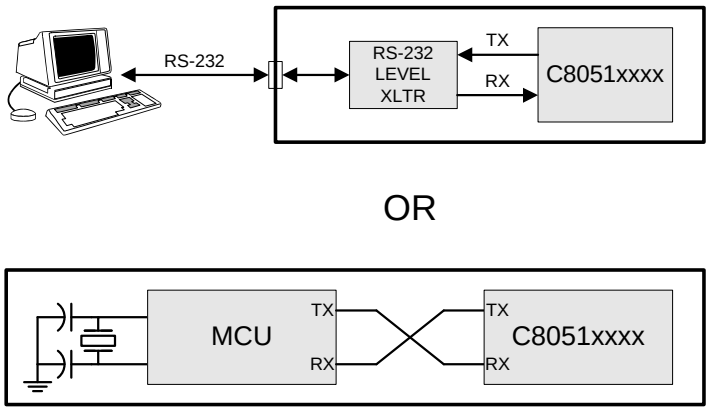

Figure 32.3. UART Interconnect Diagram

32.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

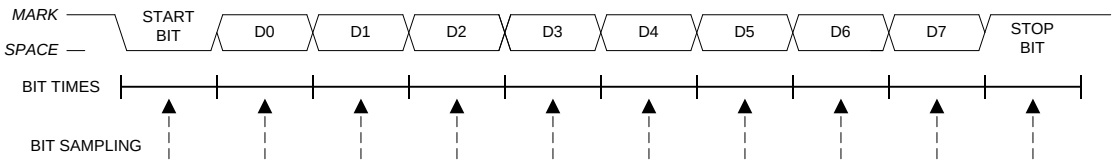

Figure 32.4. 8-Bit UART Timing Diagram

33.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode. Timer 2 can also be used in capture mode to capture rising edges of the Comparator 0 output.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. The external oscillator source divided by 8 is synchronized with the system clock.

33.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 33.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

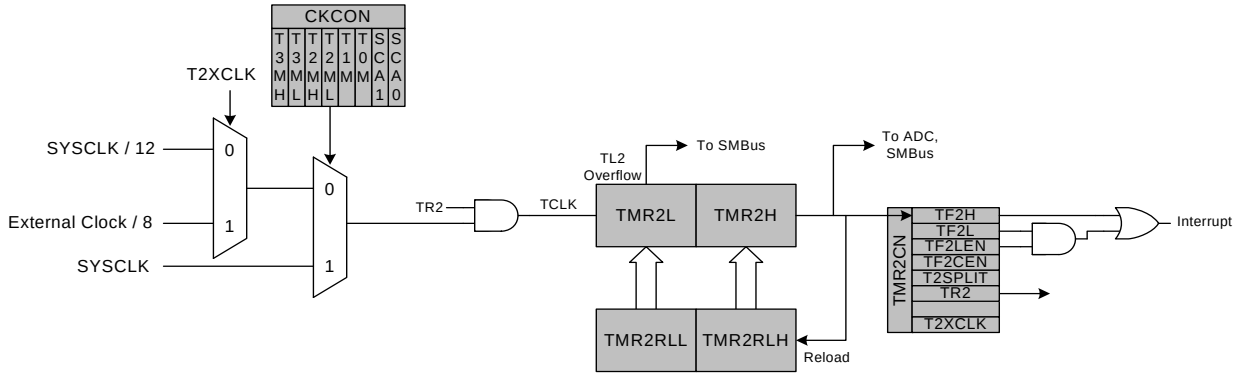

Figure 33.4. Timer 2 16-Bit Mode Block Diagram