



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), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, WDT
Number of I/O	39
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 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	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f715-gq

C8051F70x/71x

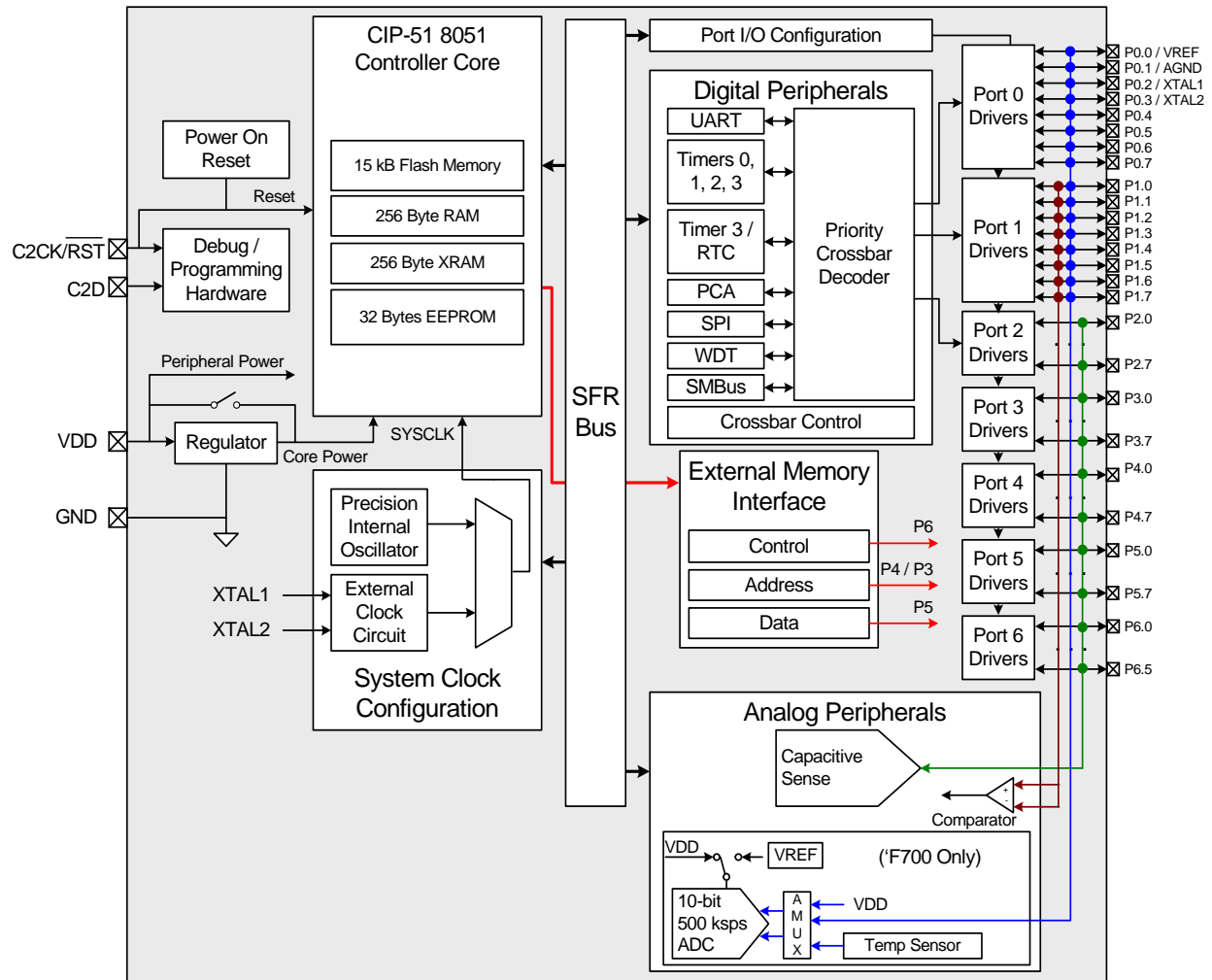


Figure 1.1. C8051F700/1 Block Diagram

C8051F70x/71x

Table 3.1. Pin Definitions for the C8051F70x/71x (Continued)

Name	TQFP64	TQFP48 QFN48	QFN32	QFN24	Type	Description
P5.1	10	10	7	—	D I/O or A In	Port 5.0. CS0 input pin 26.
P5.2	7	7	6	—	D I/O or A In	Port 5.2. CS0 input pin 27
P5.3	6	6	5	—	D I/O or A In	Port 5.3. CS0 input pin 28.
P5.4	5	5	4	—	D I/O or A In	Port 5.4. CS0 input pin 29.
P5.5	4	4	3	—	D I/O or A In	Port 5.5. CS0 input pin 30.
P5.6	3	3	2	—	D I/O or A In	Port 5.6. CS0 input pin 31.
P5.7	2	2	1	—	D I/O or A In	Port 5.7. CS0 input pin 32.
P6.0	1	—	—	—	D I/O	Port 6.0. CS0 input pin 33.
P6.1	64	—	—	—	D I/O	Port 6.1. CS0 input pin 34.
P6.2	63	—	—	—	D I/O	Port 6.2. CS0 input pin 35.
P6.3	62	1	32	—	D I/O	Port 6.3. CS0 input pin 36.
P6.4	61	48	31	1	D I/O	Port 6.4. CS0 input pin 37.
P6.5	60	47	30	24	D I/O	Port 6.5. CS0 input pin 38.

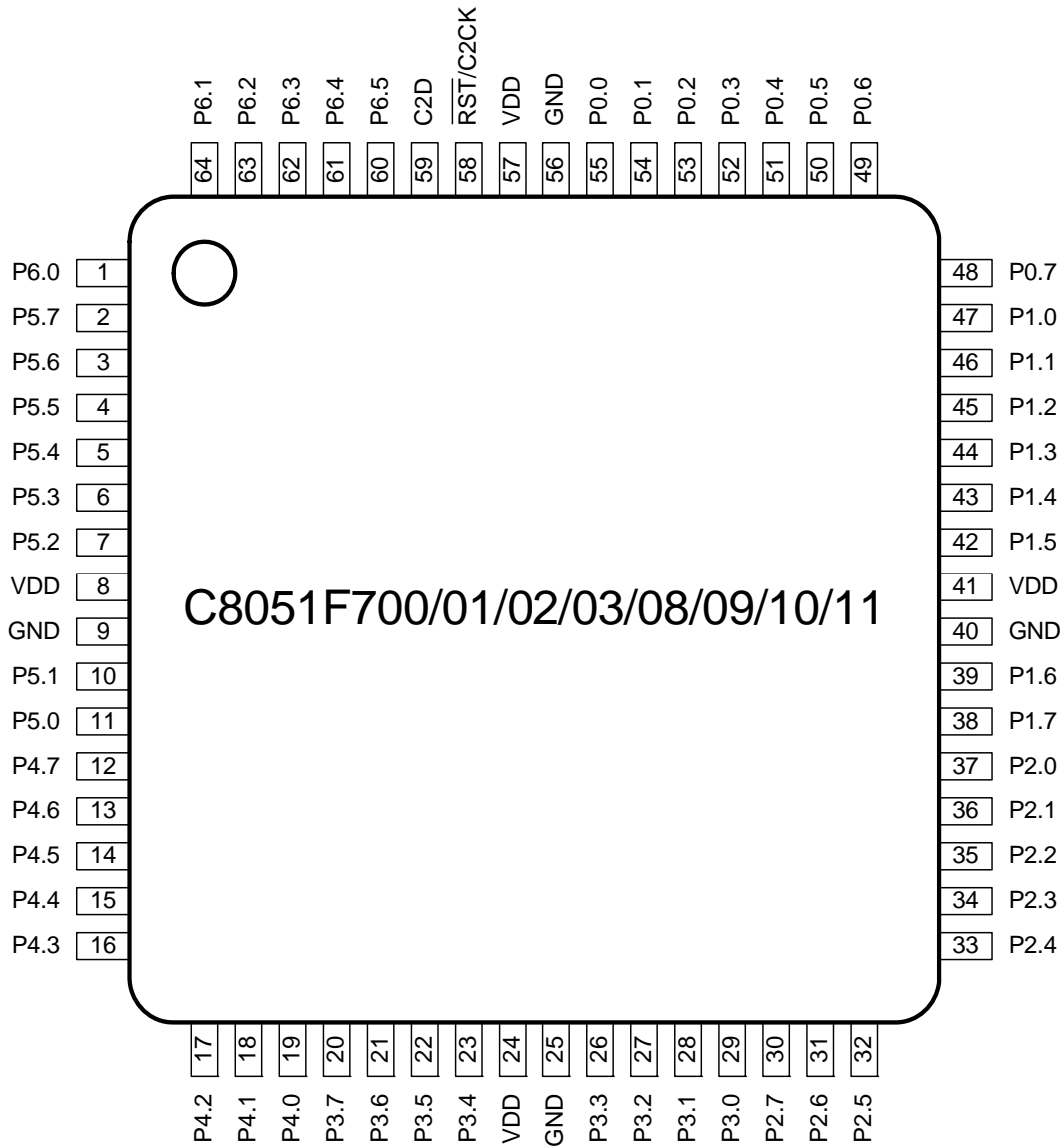


Figure 3.1. C8051F7xx-GQ TQFP64 Pinout Diagram (Top View)

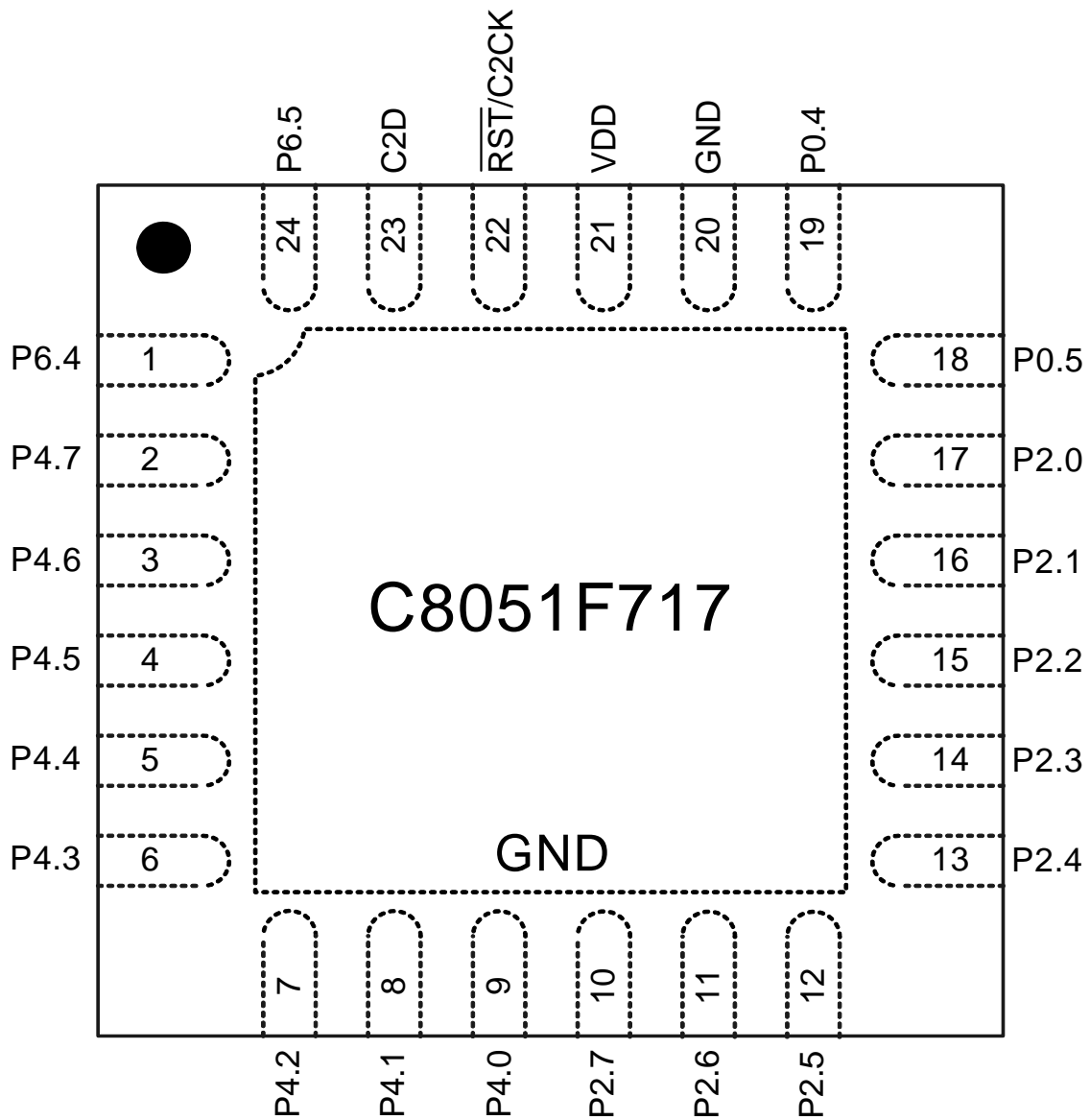


Figure 3.5. C8051F717-GM QFN24 Pinout Diagram (Top View)

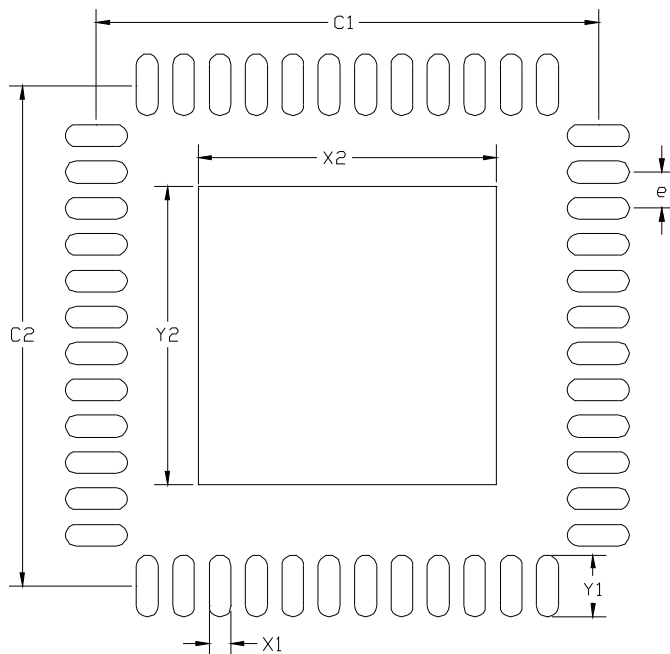


Figure 6.2. QFN-48 PCB Land Pattern

Table 6.2. QFN-48 PCB Land Pattern Dimensions

Dimension	Min	Max
e	0.50 BSC	
C1	6.80	6.90
C2	6.80	6.90
X1	0.20	0.30
X2	4.00	4.10
Y1	0.75	0.85
Y2	4.00	4.10

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on IPC-SM-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

5. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μ m minimum, all the way around the pad.

Stencil Design

6. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
7. The stencil thickness should be 0.125 mm (5 mils).
8. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
9. A 3x3 array of 1.20 mm square openings on 1.40 mm pitch should be used for the center ground pad.

Card Assembly

10. A No-Clean, Type-3 solder paste is recommended.
11. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

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.

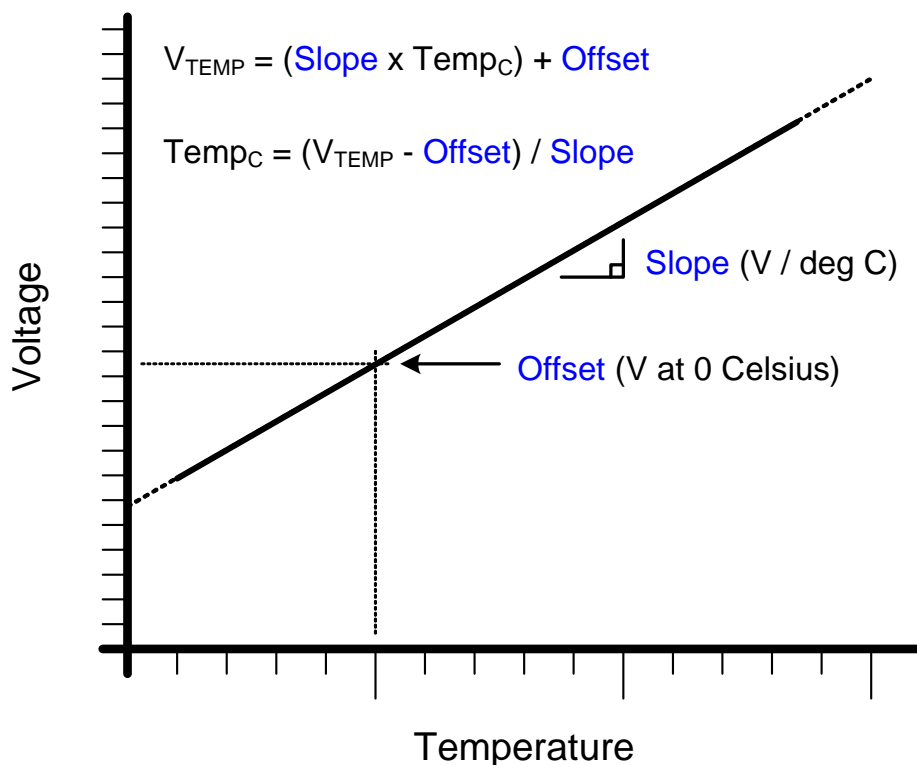


Figure 11.1. Temperature Sensor Transfer Function

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.

13. Voltage Regulator (REG0)

C8051F70x/71x devices include an internal voltage regulator (REG0) to regulate the internal core supply to 1.8 V from a V_{DD} supply of 1.8 to 3.6 V. Two power-saving modes are built into the regulator to help reduce current consumption in low-power applications. These modes are accessed through the REG0CN register (SFR Definition 13.1). Electrical characteristics for the on-chip regulator are specified in Table 9.5 on page 50

If an external regulator is used to power the device, the internal regulator may be put into bypass mode using the BYPASS bit. **The internal regulator should never be placed in bypass mode unless an external 1.8 V regulator is used to supply V_{DD} . Doing so could cause permanent damage to the device.**

Under default conditions, when the device enters STOP mode the internal regulator will remain on. This allows any enabled reset source to generate a reset for the device and bring the device out of STOP mode. For additional power savings, the STOPCF bit can be used to shut down the regulator and the internal power network of the device when the part enters STOP mode. When STOPCF is set to 1, the RST pin or a full power cycle of the device are the only methods of generating a reset.

SFR Definition 18.1. EMI0CN: External Memory Interface Control

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

SFR Address = 0xAA; SFR Page = F

Bit	Name	Function
7:0	PGSEL[7:0]	XRAM Page Select Bits. The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. 0x00: 0x0000 to 0x00FF 0x01: 0x0100 to 0x01FF ... 0xFE: 0xFE00 to 0xFEFF 0xFF: 0xFF00 to 0xFFFF

18.6.2.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 010

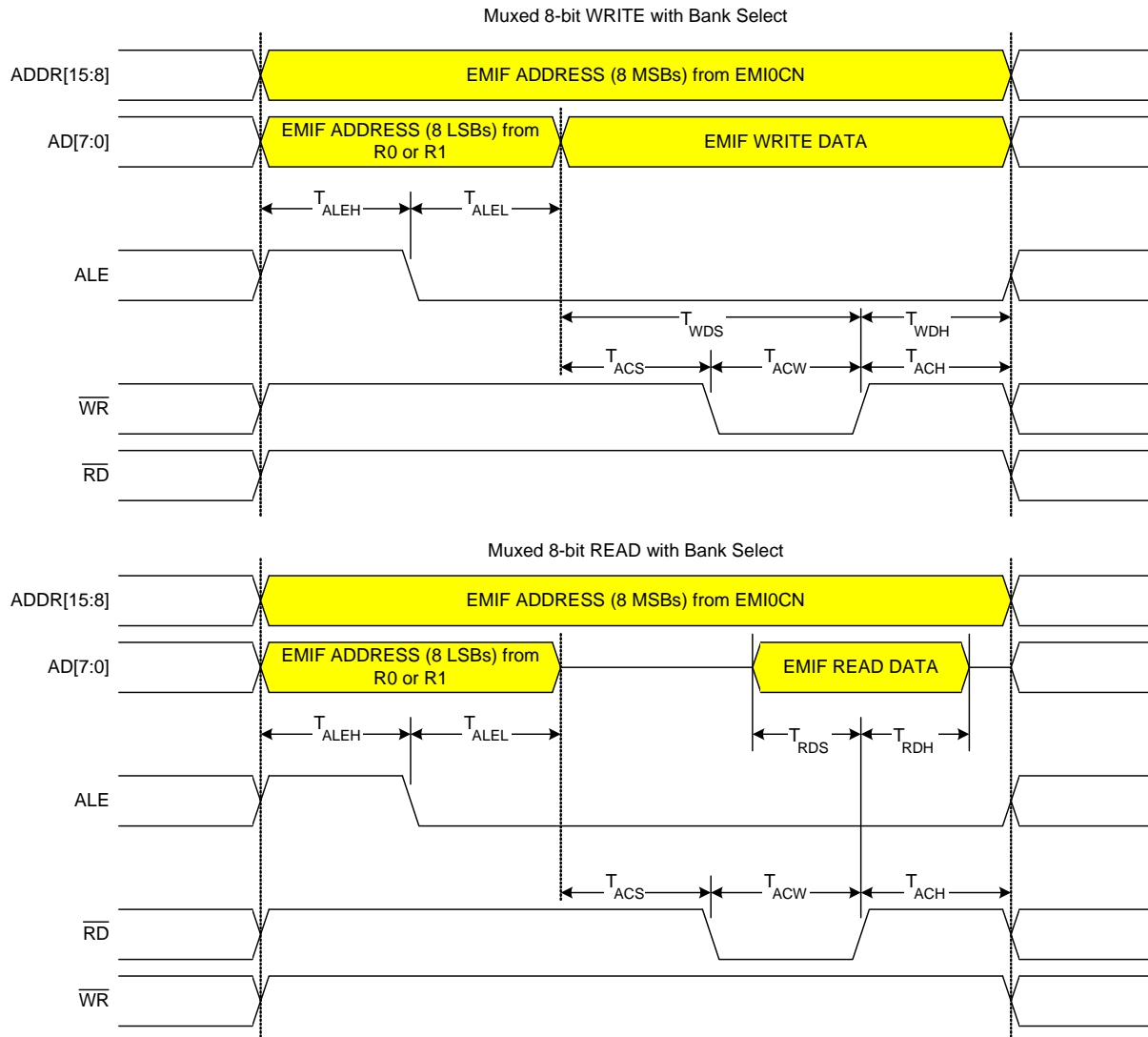


Figure 18.9. Multiplexed 8-Bit MOVX with Bank Select Timing

Table 20.2. Special Function Registers (Continued)

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

Register	Address	Page	Description	Page
CS0CN	0x9A	0	CS0 Control	88
CS0DH	0xAA	0	CS0 Data High	90
CS0DL	0xA9	0	CS0 Data Low	90
CS0CF	0x9E	0	CS0 Configuration	89
CS0MD1	0xAD	0	CS0 Mode 1	94
CS0MD2	0xBE	F	CS0 Mode 2	95
CS0MX	0x9C	0	CS0 Mux	97
CS0PM	0x9F	F	CS0 Pin Monitor	93
CS0SE	0x93	F	Auto Scan End Channel	91
CS0SS	0x92	F	Auto Scan Start Channel	91
CS0THH	0x97	0	CS0 Digital Compare Threshold High	92
CS0THL	0x96	0	CS0 Digital Compare Threshold Low	92
DERIVID	0xEC	F	Derivative Identification	128
DPH	0x83	All Pages	Data Pointer High	104
DPL	0x82	All Pages	Data Pointer Low	104
EEADDR	0xB6	All Pages	EEPROM Byte Address	156
EECNTL	0xC5	F	EEPROM Control	158
EEDATA	0xD1	All Pages	EEPROM Byte Data	157
EEKEY	0xC6	F	EEPROM Protect Key	159
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	142
EIE2	0xE7	All Pages	Extended Interrupt Enable 2	143
EIP1	0xCE	F	Extended Interrupt Priority 1	144
EIP2	0xCF	F	Extended Interrupt Priority 2	145
EMI0CF	0xC7	F	EMIF Configuration	114
EMI0CN	0xAA	F	EMIF Control	113
EMI0TC	0xEE	F	EMIF Timing Control	119
FLKEY	0xB7	All Pages	Flash Lock And Key	154
HWID	0xC4	F	Hardware Identification	128
IE	0xA8	All Pages	Interrupt Enable	140
IP	0xB8	All Pages	Interrupt Priority	141
IT01CF	0xE4	F	INT0/INT1 Configuration	147
OSCICL	0xBF	F	Internal Oscillator Calibration	173
OSICN	0xA9	F	Internal Oscillator Control	174
OSXCXN	0xB5	F	External Oscillator Control	176
P0	0x80	All Pages	Port 0 Latch	195
P0DRV	0xF9	F	Port 0 Drive Strength	197
P0MASK	0xF4	0	Port 0 Mask	192
P0MAT	0xF3	0	Port 0 Match	193

Table 21.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y		ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
Port Match	0x0043	8	None	N/A	N/A	EMAT (EIE1.1)	PMAT (EIP1.1)
ADC0 Window Compare	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 Conversion Complete	0x0053	10	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
RESERVED							
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
CS0 Conversion Complete	0x007B	15	CS0INT (CS0CN.5)	N	N	ECSCPT (EIE2.0)	PSCCPT (EIP2.0)
CS0 Greater Than Compare	0x0083	16	CS0CMPF (CS0CN.0)	N	N	ECSGRT (EIE2.1)	PSCGRT (EIP2.1)

C8051F70x/71x

SFR Definition 28.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE		PCA0ME[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2; SFR Page = F

Bit	Name	Function
7	WEAKPUD	Port I/O Weak Pullup Disable. 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	Crossbar Enable. 0: Crossbar disabled. 1: Crossbar enabled.
5	T1E	T1 Enable. 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
4	T0E	T0 Enable. 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
3	ECIE	PCA0 External Counter Input Enable. 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
2	Unused	Read = 0b; Write = Don't Care.
1:0	PCA0ME[1:0]	PCA Module I/O Enable Bits. 00: All PCA I/O unavailable at Port pins. 01: CEX0 routed to Port pin. 10: CEX0, CEX1 routed to Port pins. 11: CEX0, CEX1, CEX2 routed to Port pins.

C8051F70x/71x

SFR Definition 28.15. P1SKIP: Port 1 Skip

Bit	7	6	5	4	3	2	1	0
Name	P1SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD5; SFR Page = F

Bit	Name	Function
7:0	P1SKIP[7:0]	Port 1 Crossbar Skip Enable Bits. These bits select Port 1 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P1.n pin is not skipped by the Crossbar. 1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 28.16. P1DRV: Port 1 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P1DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xFA; SFR Page = F

Bit	Name	Function
7:0	P1DRV[7:0]	Drive Strength Configuration Bits for P1.7–P1.0 (respectively). Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P1.n Output has low output drive strength. 1: Corresponding P1.n Output has high output drive strength.

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

Table 30.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1) (Continued)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Receiver	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
	0	0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	—
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—

1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 31.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 31.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 31.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

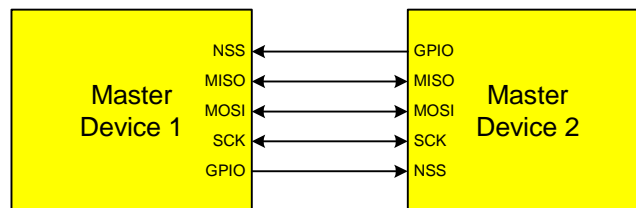


Figure 31.2. Multiple-Master Mode Connection Diagram

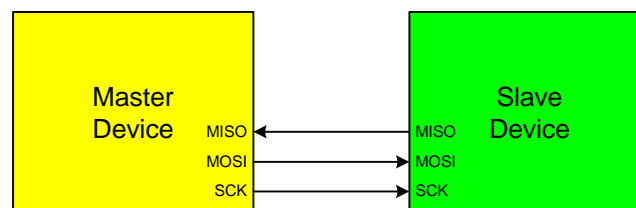


Figure 31.3. 3-Wire Single Master and Single Slave Mode Connection Diagram

32.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

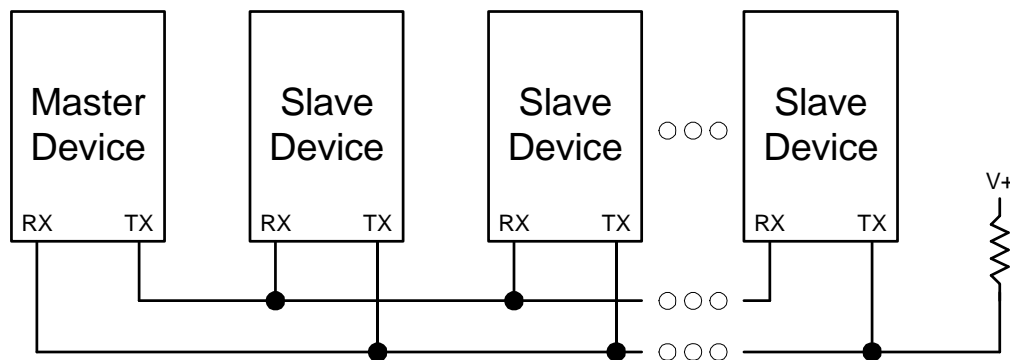


Figure 32.6. UART Multi-Processor Mode Interconnect Diagram

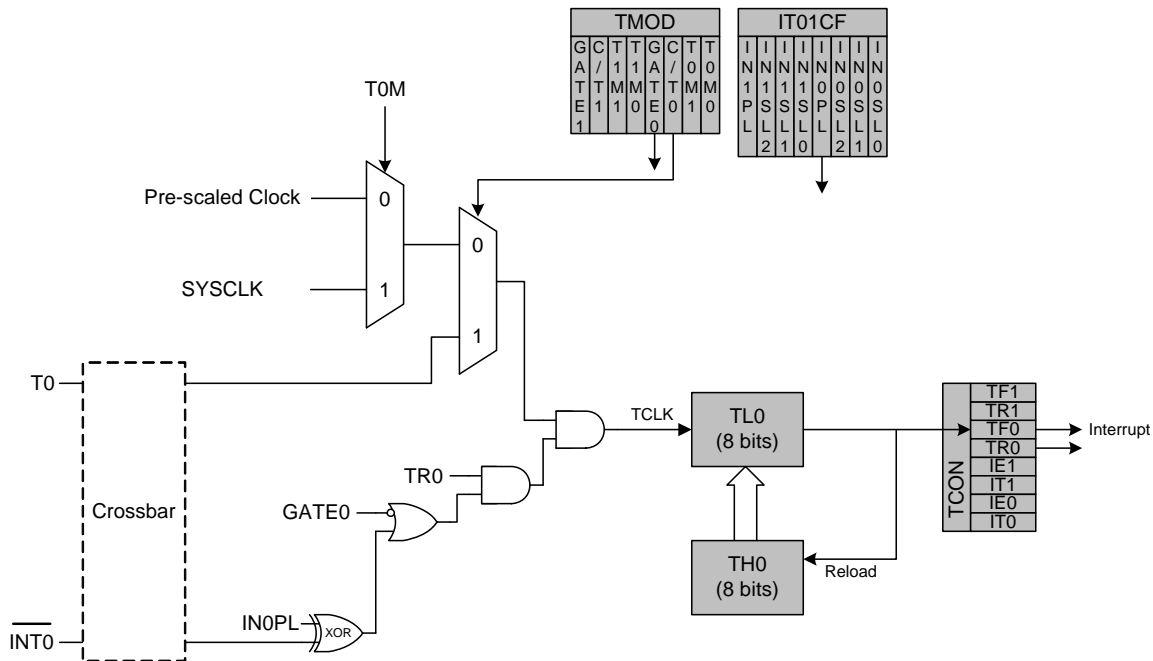


Figure 33.2. T0 Mode 2 Block Diagram

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

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

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

C8051F70x/71x

SFR Definition 34.5. PCA0L: PCA Counter/Timer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[7:0]							
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 = 0xF9; SFR Page = 0

Bit	Name	Function
7:0	PCA0[7:0]	PCA Counter/Timer Low Byte. The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

SFR Definition 34.6. PCA0H: PCA Counter/Timer High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0[15:8]							
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 = 0xFA; SFR Page = 0

Bit	Name	Function
7:0	PCA0[15:8]	PCA Counter/Timer High Byte. The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Reads of this register will read the contents of a “snapshot” register, whose contents are updated only when the contents of PCA0L are read (see Section 34.1).