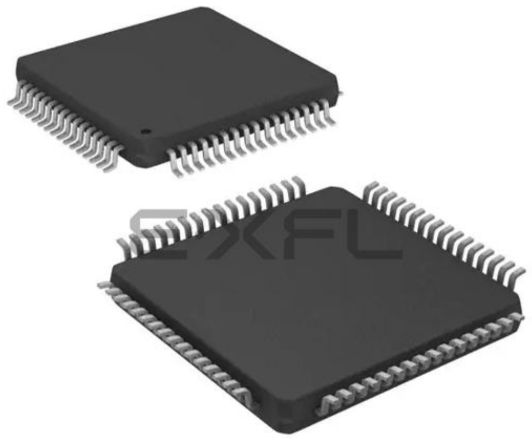

Welcome to E-XFL.COM

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, 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	-
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/c8051f703-gqr

5. TQFP-48 Package Specifications

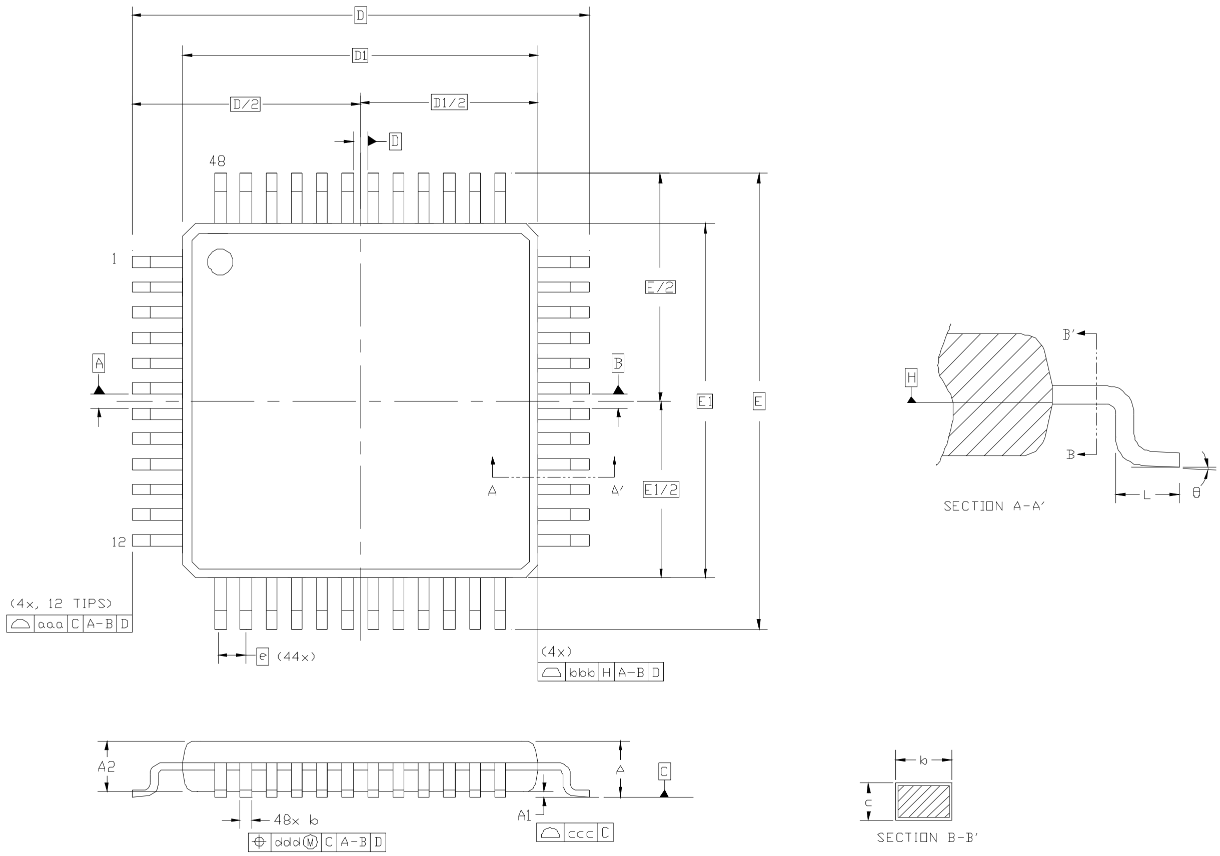

Figure 5.1. TQFP-48 Package Drawing

Table 5.1. TQFP-48 Package Dimensions

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	—	—	1.20	E	9.00 BSC.		
A1	0.05	—	0.15	E1	7.00 BSC.		
A2	0.95	1.00	1.05	L	0.45	0.60	0.75
b	0.17	0.22	0.27	aaa	0.20		
c	0.09	—	0.20	bbb	0.20		
D	9.00 BSC.			ccc	0.08		
D1	7.00 BSC.			ddd	0.08		
e	0.50 BSC.			Θ	0°	3.5°	7°

Notes:
1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MS-026, variation ABC.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

Table 9.11. Power Management Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	Units
Idle Mode Wake-Up time		2	—	3	SYSCLKs
Suspend Mode Wake-Up Time		—	250	—	ns

Table 9.12. Temperature Sensor Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	Units
Linearity		—	1	—	°C
Slope		—	3.27	—	mV/°C
Slope Error*		—	±65	—	µV/°C
Offset	Temp = 0 °C	—	868	—	mV
Offset Error*	Temp = 0 °C	—	±15.3	—	mV
***Note:** Represents one standard deviation from the mean.					

Table 9.13. Voltage Reference Electrical Characteristics

V_{DD} = 1.8 to 3.6 V; –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal High-Speed Reference (REFSL[1:0] = 11)					
Output Voltage	25 °C ambient	1.55	1.59	1.70	V
Turn-on Time		—	—	1.7	µs
Supply Current		—	200	—	µA
External Reference (REF0E = 0)					
Input Voltage Range		0	—	V_{DD}	
Input Current	Sample Rate = 500 ksps; VREF = 3.0 V	—	7	—	µA

Table 9.14. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	300	—	ns
	CP0+ – CP0– = –100 mV	—	200	—	ns
Response Time: Mode 1, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	400	—	ns
	CP0+ – CP0– = –100 mV	—	350	—	ns
Response Time: Mode 2, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	570	—	ns
	CP0+ – CP0– = –100 mV	—	870	—	ns
Response Time: Mode 3, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV	—	1500	—	ns
	CP0+ – CP0– = –100 mV	—	4500	—	ns
Common-Mode Rejection Ratio		—	1	4	mV/V
Positive Hysteresis 1	Mode 2, CP0HYP1–0 = 00	—	0	1	mV
Positive Hysteresis 2	Mode 2, CP0HYP1–0 = 01	2	5	10	mV
Positive Hysteresis 3	Mode 2, CP0HYP1–0 = 10	7	10	20	mV
Positive Hysteresis 4	Mode 2, CP0HYP1–0 = 11	10	20	30	mV
Negative Hysteresis 1	Mode 2, CP0HYN1–0 = 00	—	0	1	mV
Negative Hysteresis 2	Mode 2, CP0HYN1–0 = 01	2	5	10	mV
Negative Hysteresis 3	Mode 2, CP0HYN1–0 = 10	7	10	20	mV
Negative Hysteresis 4	Mode 2, CP0HYN1–0 = 11	10	20	30	mV
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{DD} + 0.25	V
Input Offset Voltage		–7.5	—	7.5	mV
Power Specifications					
Power Supply Rejection		—	0.1	—	mV/V
Powerup Time		—	10	—	µs
Supply Current at DC	Mode 0	—	25	—	µA
	Mode 1	—	10	—	µA
	Mode 2	—	3	—	µA
	Mode 3	—	0.5	—	µA
Note: Vcm is the common-mode voltage on CP0+ and CP0–.					

12. Voltage and Ground Reference Options

The voltage reference MUX is configurable to use an externally connected voltage reference, the on-chip voltage reference, or one of two power supply voltages (see Figure 12.1). The ground reference MUX allows the ground reference for ADC0 to be selected between the ground pin (GND) or a port pin dedicated to analog ground (P0.1/AGND).

The voltage and ground reference options are configured using the REF0CN SFR described on page 71. Electrical specifications are can be found in the Electrical Specifications Chapter.

Important Note About the V_{REF} and AGND Inputs: Port pins are used as the external V_{REF} and AGND inputs. When using an external voltage reference, P0.0/VREF should be configured as an analog input and skipped by the Digital Crossbar. When using AGND as the ground reference to ADC0, P0.1/AGND should be configured as an analog input and skipped by the Digital Crossbar. Refer to Section “28. Port Input/Output” on page 180 for complete Port I/O configuration details. The external reference voltage must be within the range $0 \leq V_{REF} \leq V_{DD}$ and the external ground reference must be at the same DC voltage potential as GND.

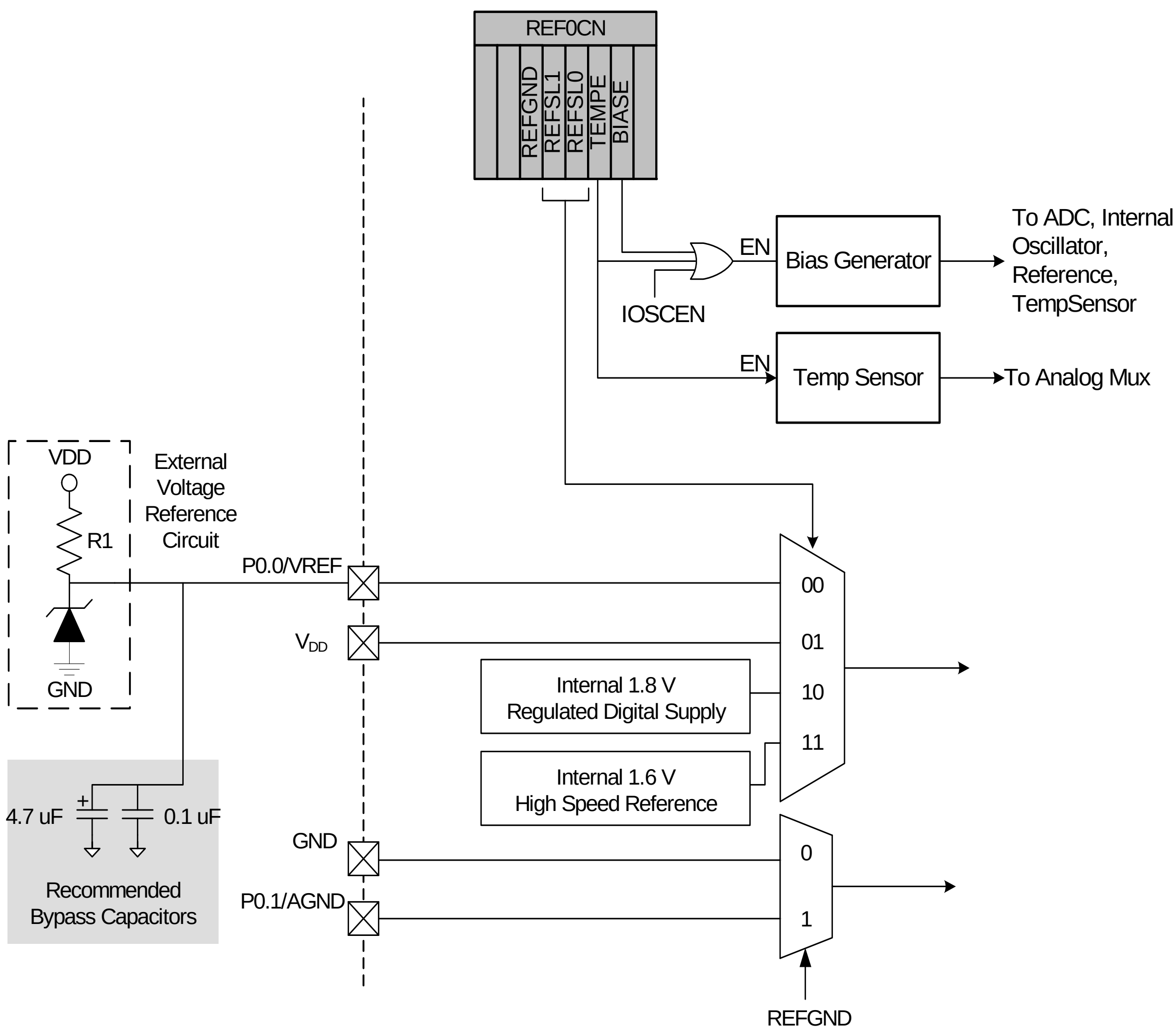

Figure 12.1. Voltage Reference Functional Block Diagram

The Comparator response time may be configured in software via the CPT0MD register (see SFR Definition 14.2). Selecting a longer response time reduces the Comparator supply current.

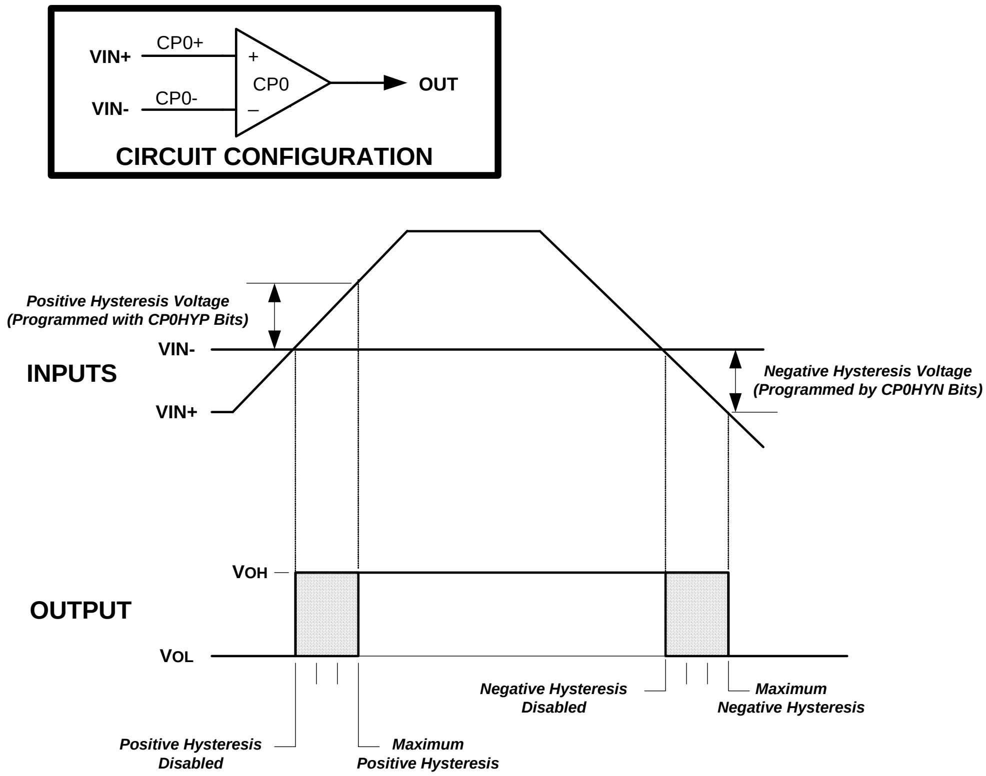

Figure 14.2. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPT0CN. The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3–0 in the Comparator Control Register CPT0CN (shown in SFR Definition 14.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 14.2, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “21.1. MCU Interrupt Sources and Vectors” on page 138). The CP0FIF flag is set to logic 1 upon a Comparator falling-edge occurrence, and the CP0RIF flag is set to logic 1 upon the Comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The Comparator rising-edge interrupt mask is enabled by setting CP0RIE to a logic 1. The Comparator0 falling-edge interrupt mask is enabled by setting CP0FIE to a logic 1.

The output state of the Comparator can be obtained at any time by reading the CP0OUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed.

Table 16.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2

17.1. Program Memory

The members of the C8051F70x/71x device family contain 16 kB (C8051F702/3/6/7 and C8051F16/7), 15 kB (C8051F700/1/4/5), or 8 kB (C8051F708/9 and C8051F710/1/2/3/4/5) of re-programmable Flash memory that can be used as non-volatile program or data storage. The last byte of user code space is used as the security lock byte (0x3FFF on 16 kB devices, 0x3BFF on 15 kB devices and 0x1FFF on 8 kB devices).

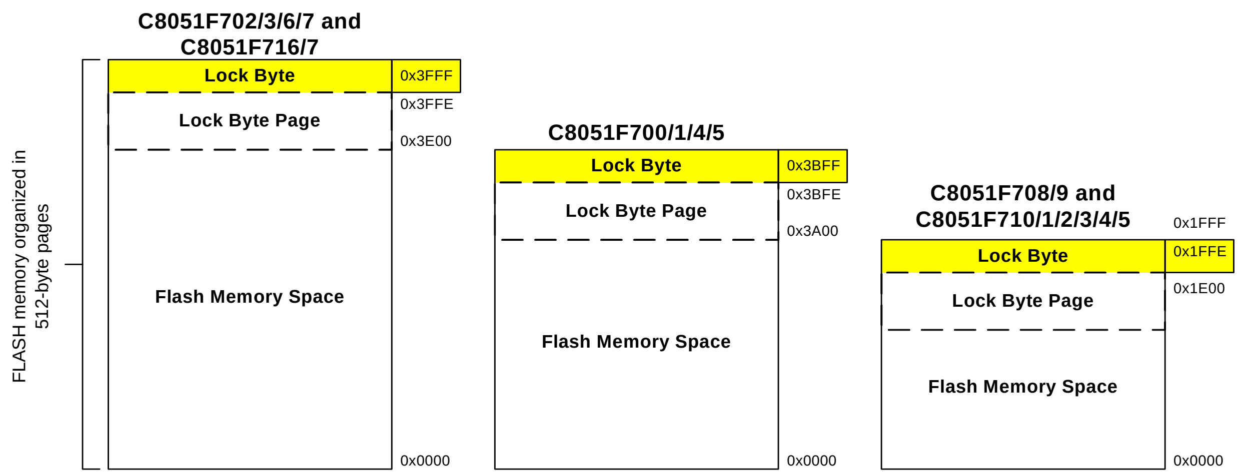

Figure 17.2. Flash Program Memory Map

17.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F70x/71x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip Flash memory space. MOVC instructions are always used to read Flash memory, while MOVX write instructions are used to erase and write Flash. This Flash access feature provides a mechanism for the C8051F70x/71x to update program code and use the program memory space for non-volatile data storage. Refer to Section “22. Flash Memory” on page 148 for further details.

17.2. EEPROM Memory

The C8051F700/1/4/5/8/9 and C8051F712/3 contain EEPROM emulation hardware, which uses Flash memory to emulate a 32-byte EEPROM memory space for non-volatile data storage. The EEPROM data is accessed through a RAM buffer for increased speed. More details about the EEPROM can be found in Section “23. EEPROM” on page 155.

17.3. Data Memory

The C8051F70x/71x device family includes 512 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. 256 bytes of this memory is on-chip “external” memory. The data memory map is shown in Figure 17.1 for reference.

17.3.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight

18.6.1.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 110

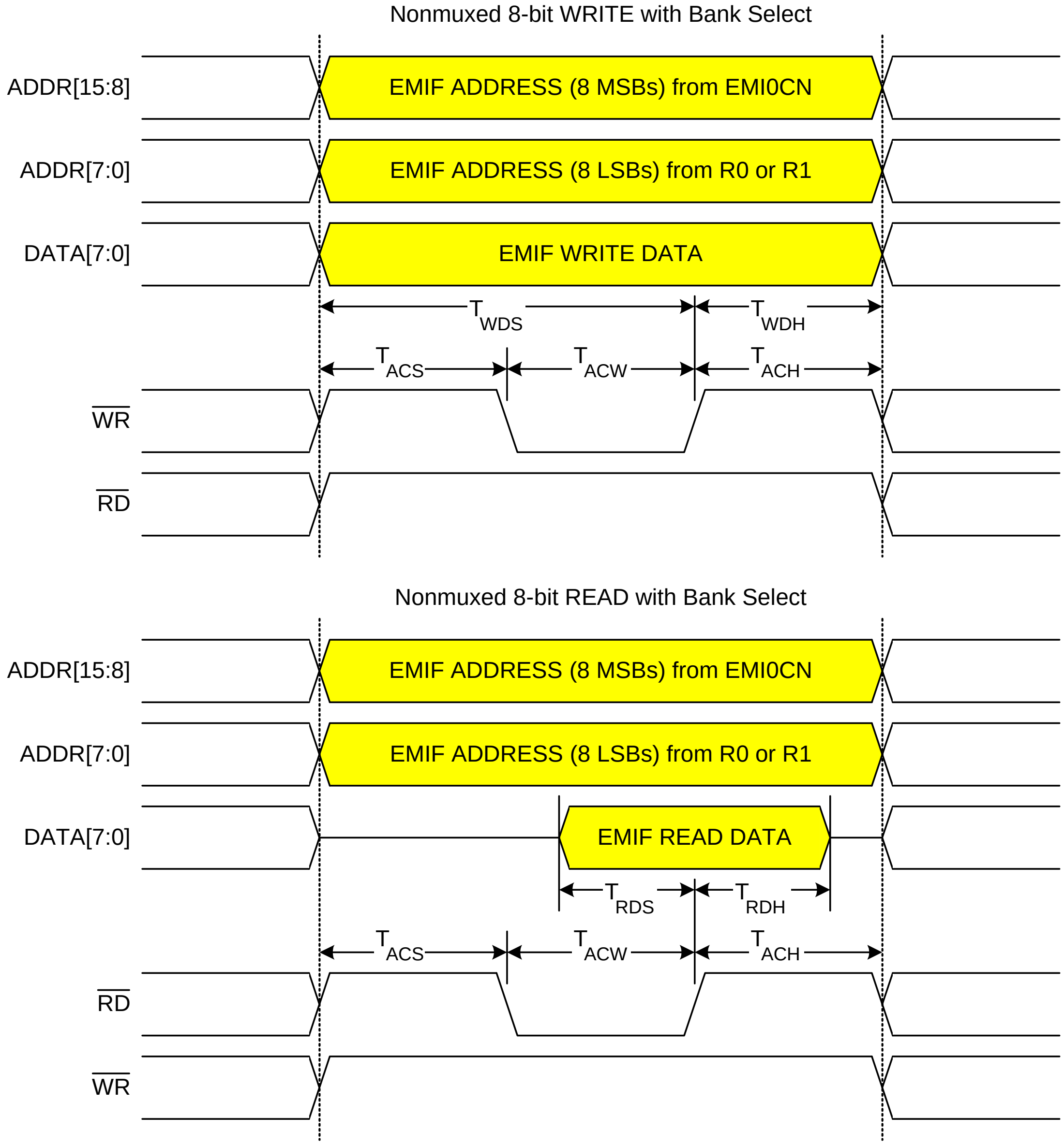

Figure 18.6. Non-Multiplexed 8-Bit MOVX with Bank Select Timing

SFR Definition 21.6. EIP2: Extended Interrupt Priority 2

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PSCGRT	PSCCPT
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCF; SFR Page = F

Bit	Name	Function
7:2	Reserved	Must Write 000000b.
1	PSCGRT	**Capacitive Sense Greater Than Comparator Priority Control.** This bit sets the priority of the Capacitive Sense Greater Than Comparator interrupt. 0: CS0 Greater Than Comparator interrupt set to low priority level. 1: CS0 Greater Than Comparator set to high priority level.
0	PSCCPT	**Capacitive Sense Conversion Complete Priority Control.** This bit sets the priority of the Capacitive Sense Conversion Complete interrupt. 0: CS0 Conversion Complete set to low priority level. 1: CS0 Conversion Complete set to high priority level.

6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
7. Clear the PSWE and PSEE bits.
8. Restore previous interrupt state.

Steps 4–6 must be repeated for each 512-byte page to be erased.

Note: Flash security settings may prevent erasure of some Flash pages, such as the reserved area and the page containing the lock bytes. For a summary of Flash security settings and restrictions affecting Flash erase operations, please see Section “22.3. Security Options” on page 149.

22.1.3. Flash Write Procedure

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.**

The recommended procedure for writing a single byte in Flash is as follows:

1. Save current interrupt state and disable interrupts.
2. Ensure that the Flash byte has been erased (has a value of 0xFF).
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
8. Clear the PSWE bit.
9. Restore previous interrupt state.

Steps 5–7 must be repeated for each byte to be written.

Note: Flash security settings may prevent writes to some areas of Flash, such as the reserved area. For a summary of Flash security settings and restrictions affecting Flash write operations, please see Section “22.3. Security Options” on page 149.

22.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction.

Note: MOVX read instructions always target XRAM.

22.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, and erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock all Flash pages, starting at page 0, by writing a non-0xFF value to the lock byte. **Note that writing a non-0xFF value to the lock byte will lock all pages of FLASH from reads, writes, and erases, including the page containing the lock byte.**

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on

SFR Definition 22.2. FLKEY: Flash Lock and Key

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

SFR Address = 0xB7; SFR Page = All Pages

Bit	Name	Function
7:0	FLKEY[7:0]	**Flash Lock and Key Register.** Write: This register provides a lock and key function for Flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a Flash write or erase operation is attempted while these operations are disabled, the Flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to Flash, it can intentionally lock the Flash by writing a non-0xA5 value to FLKEY from software. Read: When read, bits 1–0 indicate the current Flash lock state. 00: Flash is write/erase locked. 01: The first key code has been written (0xA5). 10: Flash is unlocked (writes/erases allowed). 11: Flash writes/erases disabled until the next reset.

27. Oscillators and Clock Selection

C8051F70x/71x 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 27.1. 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.

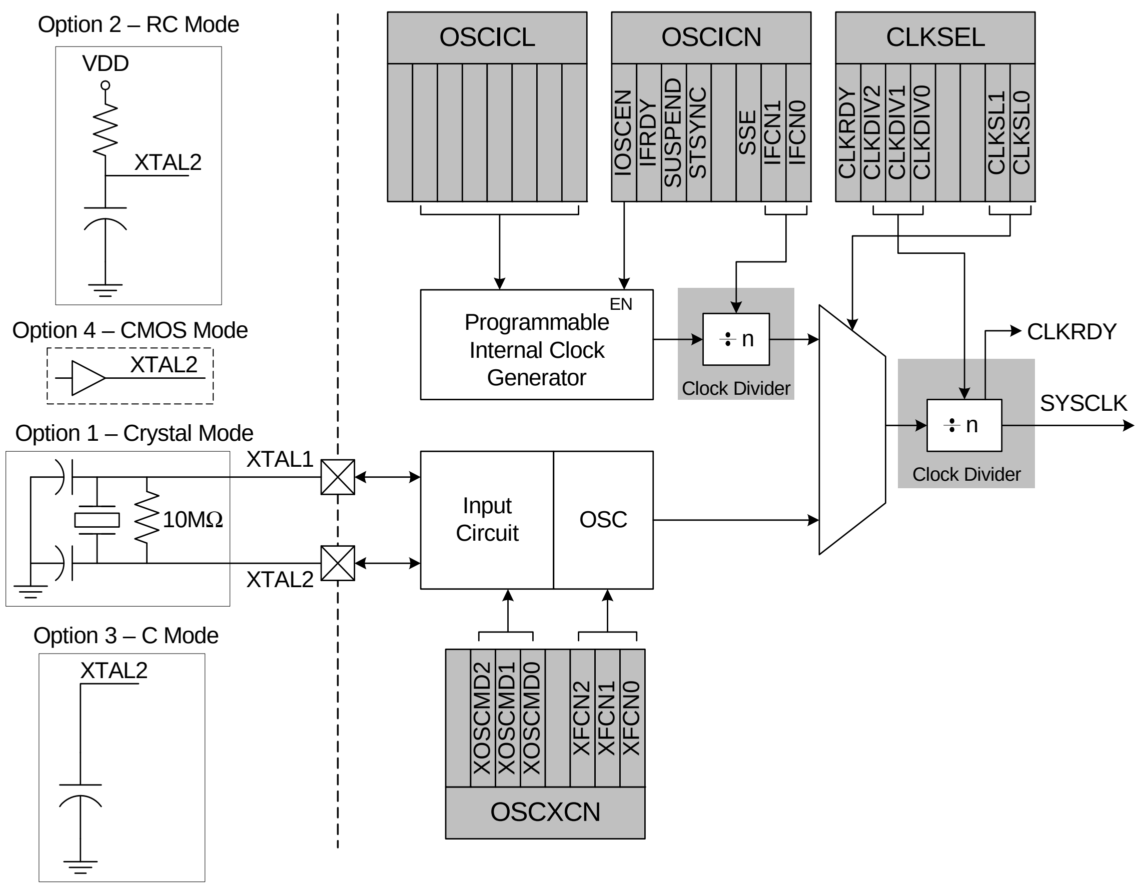

Figure 27.1. Oscillator Options

27.1. System Clock Selection

The system clock source for the MCU can be selected using the CLKSEL register. The clock selected as the system clock can be divided by 1, 2, 4, 8, 16, 32, 64, or 128. When switching between two clock divide values, the transition may take up to 128 cycles of the undivided clock source. The CLKRDY flag can be polled to determine when the new clock divide value has been applied. The clock divider must be set to "divide by 1" when entering Suspend mode. The system clock source may also be switched on-the-fly. The switchover takes effect after one clock period of the slower oscillator.

28.2.2. Assigning Port I/O Pins to Digital Functions

Any Port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the Crossbar for pin assignment; however, some digital functions bypass the Crossbar in a manner similar to the analog functions listed above. **Port pins used by these digital functions and any Port pins selected for use as GPIO should have their corresponding bit in PnSKIP set to 1.** Table 28.2 shows all available digital functions and the potential mapping of Port I/O to each digital function.

Table 28.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
UART0, SPI0, SMBus, CP0, CP0A, SYSCLK, PCA0 (CEX0-2 and ECI), T0 or T1.	Any Port pin available for assignment by the Crossbar. This includes P0.0–P2.7 pins which have their PnSKIP bit set to 0. **Note:** The Crossbar will always assign UART0 pins to P0.4 and P0.5.	XBR0, XBR1
Any pin used for GPIO	P0.0–P6.5	P0SKIP, P1SKIP, P2SKIP
External Memory Interface	P3.0–P6.2	EMI0CF

28.2.3. Assigning Port I/O Pins to External Event Trigger Functions

External event trigger functions can be used to trigger an interrupt or wake the device from a low power mode when a transition occurs on a digital I/O pin. The event trigger functions do not require dedicated pins and will function on both GPIO pins (PnSKIP = 1) and pins in use by the Crossbar (PnSKIP = 0). External event trigger functions cannot be used on pins configured for analog I/O. Table 28.3 shows all available external event trigger functions.

Table 28.3. Port I/O Assignment for External Event Trigger Functions

Event Trigger Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
External Interrupt 0	P0.0–P0.7	IT01CF
External Interrupt 1	P0.0–P0.7	IT01CF
Port Match	P0.0–P1.7	P0MASK, P0MAT P1MASK, P1MAT

SFR Definition 28.21. P2DRV: Port 2 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	P2DRV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xFB; SFR Page = F

Bit	Name	Function
7:0	P2DRV[7:0]	**Drive Strength Configuration Bits for P2.7–P2.0 (respectively).** Configures digital I/O Port cells to high or low output drive strength. 0: Corresponding P2.n Output has low output drive strength. 1: Corresponding P2.n Output has high output drive strength.

SFR Definition 28.22. P3: Port 3

Bit	7	6	5	4	3	2	1	0
Name	P3[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xB0; SFR Page = All Pages; Bit Addressable

Bit	Name	Description	Write	Read
7:0	P3[7:0]	**Port 3 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: P3.n Port pin is logic LOW. 1: P3.n Port pin is logic HIGH.

29.3. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should select the desired polynomial and set the initial value of the result. Two polynomials are available: 0x1021 (16-bit) and 0x04C11DB7 (32-bit). The CRC0 result may be initialized to one of two values: 0x00000000 or 0xFFFFFFFF. The following steps can be used to initialize CRC0.

1. Select a polynomial (Set CRC0SEL to 0 for 32-bit or 1 for 16-bit).
2. Select the initial result value (Set CRC0VAL to 0 for 0x00000000 or 1 for 0xFFFFFFFF).
3. Set the result to its initial value (Write 1 to CRC0INIT).

29.4. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more Flash sectors. The following steps can be used to automatically perform a CRC on Flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit in CRC0AUTO.
4. Write the number of Flash sectors to perform in the CRC calculation to CRC0CNT.

Note: Each Flash sector is 512 bytes.

5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes.
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result using the procedure below.

29.5. Accessing the CRC0 Result

The internal CRC0 result is 32-bits (CRC0SEL = 0b) or 16-bits (CRC0SEL = 1b). The CRC0PNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

Table 30.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0) (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	
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0	—
						Reschedule failed transfer.	1	0	0	1110

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

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 Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). Set ACK for initial data byte.	0	0	1	1000

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

<table>
<tr><th rowspan="2">Mode</th><th colspan="4">Values Read</th><th rowspan="2">Current SMbus State</th><th rowspan="2">Typical Response Options</th><th colspan="3">Values to Write</th><th rowspan="2">Next Status Vector Expected</th></tr>
<tr><th>Status Vector</th><th>ACKRQ</th><th>ARBLOST</th><th>ACK</th><th>STA</th><th>STO</th><th>ACK</th></tr>
<tr><td rowspan="8">Master Receiver</td><td rowspan="8">1000</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">1</td><td rowspan="4">A master data byte was received; ACK sent.</td><td>Set ACK for next data byte;
Read SMB0DAT.</td><td>0</td><td>0</td><td>1</td><td>1000</td></tr>
<tr><td>Set NACK to indicate next data byte as the last data byte;
Read SMB0DAT.</td><td>0</td><td>0</td><td>0</td><td>1000</td></tr>
<tr><td>Initiate repeated START.</td><td>1</td><td>0</td><td>0</td><td>1110</td></tr>
<tr><td>Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).</td><td>0</td><td>0</td><td>X</td><td>1100</td></tr>
<tr><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">0</td><td rowspan="4">A master data byte was received; NACK sent (last byte).</td><td>Read SMB0DAT; send STOP.</td><td>0</td><td>1</td><td>0</td><td>—</td></tr>
<tr><td>Read SMB0DAT; Send STOP followed by START.</td><td>1</td><td>1</td><td>0</td><td>1110</td></tr>
<tr><td>Initiate repeated START.</td><td>1</td><td>0</td><td>0</td><td>1110</td></tr>
<tr><td>Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).</td><td>0</td><td>0</td><td>X</td><td>1100</td></tr>
<tr><td rowspan="4">Slave Transmitter</td><td rowspan="3">0100</td><td>0</td><td>0</td><td>0</td><td>A slave byte was transmitted; NACK received.</td><td>No action required (expecting STOP condition).</td><td>0</td><td>0</td><td>X</td><td>0001</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>A slave byte was transmitted; ACK received.</td><td>Load SMB0DAT with next data byte to transmit.</td><td>0</td><td>0</td><td>X</td><td>0100</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>A Slave byte was transmitted; error detected.</td><td>No action required (expecting Master to end transfer).</td><td>0</td><td>0</td><td>X</td><td>0001</td></tr>
<tr><td>0101</td><td>0</td><td>X</td><td>X</td><td>An illegal STOP or bus error was detected while a Slave Transmission was in progress.</td><td>Clear STO.</td><td>0</td><td>0</td><td>X</td><td>—</td></tr>
</table>

SFR Definition 33.16. TMR3L: Timer 3 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x94; SFR Page = 0

Bit	Name	Function
7:0	TMR3L[7:0]	**Timer 3 Low Byte.** In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value.

SFR Definition 33.17. TMR3H Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x95; SFR Page = 0

Bit	Name	Function
7:0	TMR3H[7:0]	**Timer 3 High Byte.** In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

SFR Definition 34.4. PCA0CPMn: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
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 Addresses: PCA0CPM0 = 0xDA, PCA0CPM1 = 0xDB, PCA0CPM2 = 0xDC

SFR Pages: PCA0CPM0 = F, PCA0CPM1 = F, PCA0CPM2 = F

Bit	Name	Function
7	PWM16n	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOMn	**Comparator Function Enable.** This bit enables the comparator function for PCA module n when set to 1.
5	CAPPn	**Capture Positive Function Enable.** This bit enables the positive edge capture for PCA module n when set to 1.
4	CAPNn	**Capture Negative Function Enable.** This bit enables the negative edge capture for PCA module n when set to 1.
3	MATn	**Match Function Enable.** This bit enables the match function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	**Toggle Function Enable.** This bit enables the toggle function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function for PCA module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	ECCFn	**Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.