



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	Active
Core Processor	8051
Core Size	8-Bit
Speed	48 MIPS
Connectivity	I ² C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.25V
Data Converters	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f38c-gmr

C8051F380/1/2/3/4/5/6/7/C

SFR Definition 19.2. OSCICL: Internal H-F Oscillator Calibration	145
SFR Definition 19.3. OSCICN: Internal H-F Oscillator Control	146
SFR Definition 19.4. CLKMUL: Clock Multiplier Control	147
SFR Definition 19.5. OSCLCN: Internal L-F Oscillator Control	148
SFR Definition 19.6. OSCXCN: External Oscillator Control	152
SFR Definition 20.1. XBR0: Port I/O Crossbar Register 0	159
SFR Definition 20.2. XBR1: Port I/O Crossbar Register 1	160
SFR Definition 20.3. XBR2: Port I/O Crossbar Register 2	161
SFR Definition 20.4. P0: Port 0	162
SFR Definition 20.5. P0MDIN: Port 0 Input Mode	162
SFR Definition 20.6. P0MDOUT: Port 0 Output Mode	163
SFR Definition 20.7. P0SKIP: Port 0 Skip	163
SFR Definition 20.8. P1: Port 1	164
SFR Definition 20.9. P1MDIN: Port 1 Input Mode	164
SFR Definition 20.10. P1MDOUT: Port 1 Output Mode	165
SFR Definition 20.11. P1SKIP: Port 1 Skip	165
SFR Definition 20.12. P2: Port 2	166
SFR Definition 20.13. P2MDIN: Port 2 Input Mode	166
SFR Definition 20.14. P2MDOUT: Port 2 Output Mode	167
SFR Definition 20.15. P2SKIP: Port 2 Skip	167
SFR Definition 20.16. P3: Port 3	168
SFR Definition 20.17. P3MDIN: Port 3 Input Mode	168
SFR Definition 20.18. P3MDOUT: Port 3 Output Mode	169
SFR Definition 20.19. P3SKIP: Port 3 Skip	169
SFR Definition 20.20. P4: Port 4	170
SFR Definition 20.21. P4MDIN: Port 4 Input Mode	170
SFR Definition 20.22. P4MDOUT: Port 4 Output Mode	171
SFR Definition 21.1. USB0XCN: USB0 Transceiver Control	174
SFR Definition 21.2. USB0ADR: USB0 Indirect Address	176
SFR Definition 21.3. USB0DAT: USB0 Data	177
USB Register Definition 21.4. INDEX: USB0 Endpoint Index	179
USB Register Definition 21.5. CLKREC: Clock Recovery Control	180
USB Register Definition 21.6. FIFOn: USB0 Endpoint FIFO Access	182
USB Register Definition 21.7. FADDR: USB0 Function Address	183
USB Register Definition 21.8. POWER: USB0 Power	185
USB Register Definition 21.9. FRAMEL: USB0 Frame Number Low	186
USB Register Definition 21.10. FRAMEH: USB0 Frame Number High	186
USB Register Definition 21.11. IN1INT: USB0 IN Endpoint Interrupt	187
USB Register Definition 21.12. OUT1INT: USB0 OUT Endpoint Interrupt	188
USB Register Definition 21.13. CMINT: USB0 Common Interrupt	189
USB Register Definition 21.14. IN1IE: USB0 IN Endpoint Interrupt Enable	190
USB Register Definition 21.15. OUT1IE: USB0 OUT Endpoint Interrupt Enable	191
USB Register Definition 21.16. CMIE: USB0 Common Interrupt Enable	192
USB Register Definition 21.17. E0CSR: USB0 Endpoint0 Control	195
USB Register Definition 21.18. E0CNT: USB0 Endpoint0 Data Count	196

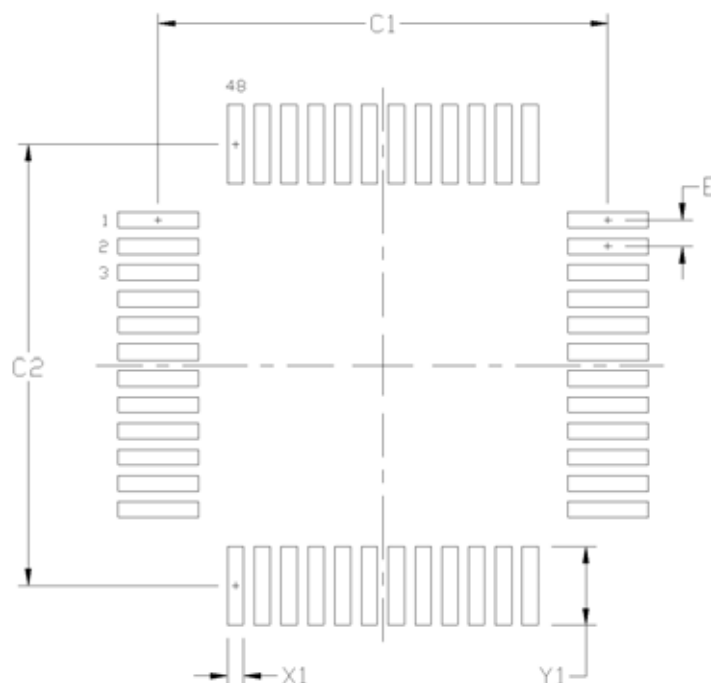


Figure 3.3. TQFP-48 Recommended PCB Land Pattern

Table 3.3. TQFP-48 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	8.30	8.40
C2	8.30	8.40
E	0.50 BSC	
X1	0.20	0.30
Y1	1.40	1.50

Notes:

General:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design:

3. 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:

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all pads.

Card Assembly:

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

C8051F380/1/2/3/4/5/6/7/C

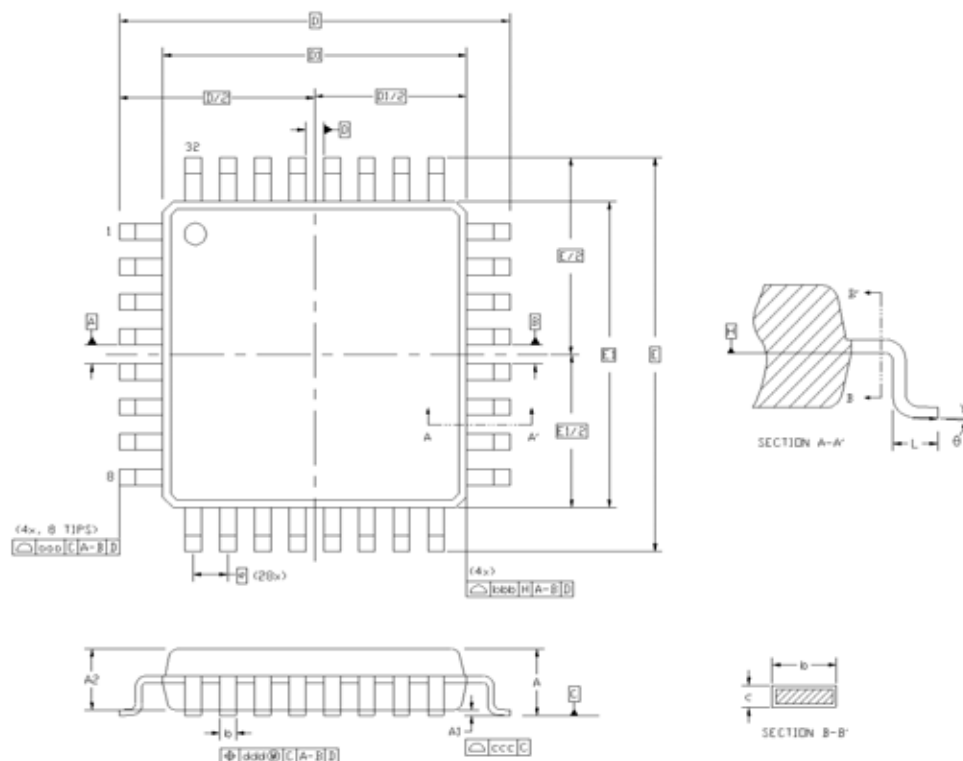


Figure 3.5. LQFP-32 Package Diagram

Table 3.4. LQFP-32 Package Dimensions

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	—	—	1.60	E	9.00 BSC		
A1	0.05	—	0.15	E1	7.00 BSC		
A2	1.35	1.40	1.45	L	0.45	0.60	0.75
b	0.30	0.37	0.45	aaa	0.20		
c	0.09	—	0.20	bbb	0.20		
D	9.00 BSC			ccc	0.10		
D1	7.00 BSC			ddd	0.20		
e	0.80 BSC			q	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 BBA.
4. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

C8051F380/1/2/3/4/5/6/7/C

6.1. Output Code Formatting

The conversion code format differs between Single-ended and Differential modes. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit (ADC0CN.0). When in Single-ended Mode, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to $V_{REF} \times 1023/1024$. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Input Voltage (Single-Ended)	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
$V_{REF} \times 1023/1024$	0x03FF	0xFFC0
$V_{REF} \times 512/1024$	0x0200	0x8000
$V_{REF} \times 256/1024$	0x0100	0x4000
0	0x0000	0x0000

When in Differential Mode, conversion codes are represented as 10-bit signed 2s complement numbers. Inputs are measured from $-V_{REF}$ to $V_{REF} \times 511/512$. Example codes are shown below for both right-justified and left-justified data. For right-justified data, the unused MSBs of ADC0H are a sign-extension of the data word. For left-justified data, the unused LSBs in the ADC0L register are set to 0.

Input Voltage (Differential)	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
$V_{REF} \times 511/512$	0x01FF	0x7FC0
$V_{REF} \times 256/512$	0x0100	0x4000
0	0x0000	0x0000
$-V_{REF} \times 256/512$	0xFF00	0xC000
$-V_{REF}$	0xFE00	0x8000

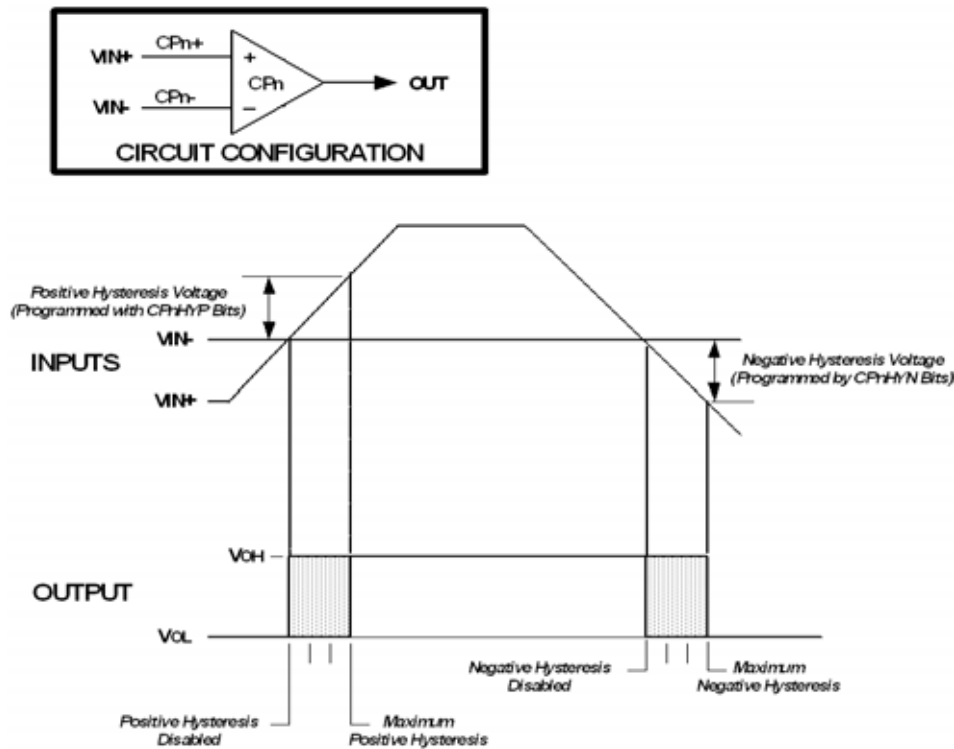


Figure 8.3. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPTnCN (for $n = 0$ or 1). 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 Bits 3–0 in the Comparator Control Register CPTnCN (shown in SFR Definition 8.1). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. Settings of 20, 10 or 5 mV of nominal 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 CPnHYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “16.1. MCU Interrupt Sources and Vectors” on page 119). The CPnFIF flag is set to logic 1 upon a Comparator falling-edge occurrence, and the CPnRIF 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 CPnRIE to a logic 1. The Comparator falling-edge interrupt mask is enabled by setting CPnFIE to a logic 1.

The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN 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.

SFR Definition 10.1. PCON: Power Control

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

SFR Address = 0x87; SFR Page = All Pages

Bit	Name	Function
7:2	GF[5:0]	General Purpose Flags 5–0. These are general purpose flags for use under software control.
1	STOP	Stop Mode Select. Setting this bit will place the CIP-51 in stop mode. This bit will always be read as 0. 1: CPU goes into stop mode (internal oscillator stopped).
0	IDLE	IDLE: Idle Mode Select. Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0. 1: CPU goes into Idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active.)

C8051F380/1/2/3/4/5/6/7/C

Table 11.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
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

11.2. CIP-51 Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should always be written to the value indicated in the SFR description. Future product versions may use these bits to implement new features in which case the reset value of the bit will be the indicated value, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

SFR Definition 11.1. DPL: Data Pointer Low Byte

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

SFR Address = 0x82; SFR Page = All Pages

Bit	Name	Function
7:0	DPL[7:0]	Data Pointer Low. The DPL register is the low byte of the 16-bit DPTR.

SFR Definition 11.2. DPH: Data Pointer High Byte

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

SFR Address = 0x83; SFR Page = All Pages

Bit	Name	Function
7:0	DPH[7:0]	Data Pointer High. The DPH register is the high byte of the 16-bit DPTR.

12. Prefetch Engine

The C8051F380/1/2/3/4/5/6/7/C family of devices incorporate a 2-byte prefetch engine. Because the access time of the Flash memory is 40 ns, and the minimum instruction time is roughly 20 ns, the prefetch engine is necessary for full-speed code execution. Instructions are read from Flash memory two bytes at a time by the prefetch engine and given to the CIP-51 processor core to execute. When running linear code (code without any jumps or branches), the prefetch engine allows instructions to be executed at full speed. When a code branch occurs, the processor may be stalled for up to two clock cycles while the next set of code bytes is retrieved from Flash memory. It is recommended that the prefetch be used for optimal code execution timing.

Note: The prefetch engine can be disabled when the device is in suspend mode to save power.

SFR Definition 12.1. PFE0CN: Prefetch Engine Control

Bit	7	6	5	4	3	2	1	0
Name			PFEN					FLBWE
Type	R	R	R/W	R	R	R	R	R/W
Reset	0	0	1	0	0	0	0	0

SFR Address = 0xAF; SFR Page = All Pages

Bit	Name	Function
7:6	Unused	Read = 00b, Write = don't care.
5	PFEN	Prefetch Enable. This bit enables the prefetch engine. 0: Prefetch engine is disabled. 1: Prefetch engine is enabled.
4:1	Unused	Read = 0000b. Write = don't care.
0	FLBWE	Flash Block Write Enable. This bit allows block writes to Flash memory from software. 0: Each byte of a software Flash write is written individually. 1: Flash bytes are written in groups of two.

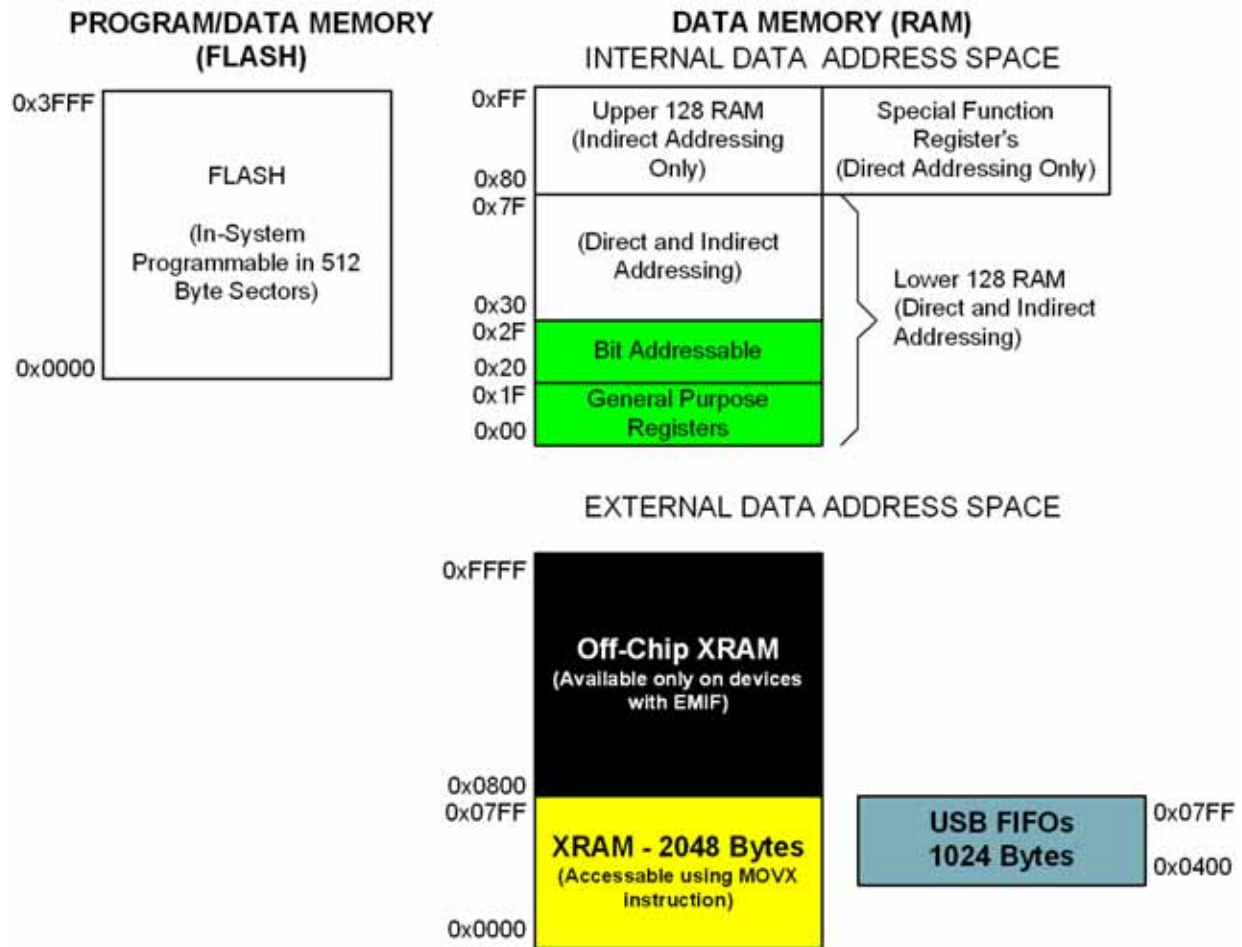


Figure 13.3. On-Chip Memory Map for 16 kB Devices (C8051F38C)

13.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F380/1/2/3/4/5/6/7/C implements 64 kB, 32 kB, or 16 kB of this program memory space as in-system, re-programmable Flash memory. Note that on the C8051F380/1/4/5 (64 kB version), addresses above 0xFBFF are reserved.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section "18. Flash Memory" on page 135 for further details.

13.2. Data Memory

The CIP-51 includes 256 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 byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

19.6. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. Figure 19.1 shows a block diagram of the four external oscillator options. The external oscillator is enabled and configured using the OSCXCN register (see SFR Definition 19.6).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are used as XTAL1 and XTAL2, respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pin used by the oscillator circuit; see Section “20.1. Priority Crossbar Decoder” on page 154 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “20.2. Port I/O Initialization” on page 158 for details on Port input mode selection.

The external oscillator output may be selected as the system clock or used to clock some of the digital peripherals (e.g. Timers, PCA, etc.). See the data sheet chapters for each digital peripheral for details. See Section “5. Electrical Characteristics” on page 37 for complete oscillator specifications.

19.6.1. External Crystal Mode

If a crystal or ceramic resonator is used as the external oscillator, the crystal/resonator and a 10 MΩ resistor must be wired across the XTAL1 and XTAL2 pins as shown in Figure 19.1, “Crystal Mode”. Appropriate loading capacitors should be added to XTAL1 and XTAL2, and both pins should be configured for analog I/O with the digital output drivers disabled.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are “in series” as seen by the crystal and “in parallel” with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The recommended load capacitance depends upon the crystal and the manufacturer. 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 19.1, 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 19.2.

21.5. FIFO Management

1024 bytes of on-chip XRAM are used as FIFO space for USB0. This FIFO space is split between Endpoints0-3 as shown in Figure 21.3. FIFO space allocated for Endpoints1-3 is configurable as IN, OUT, or both (Split Mode: half IN, half OUT).

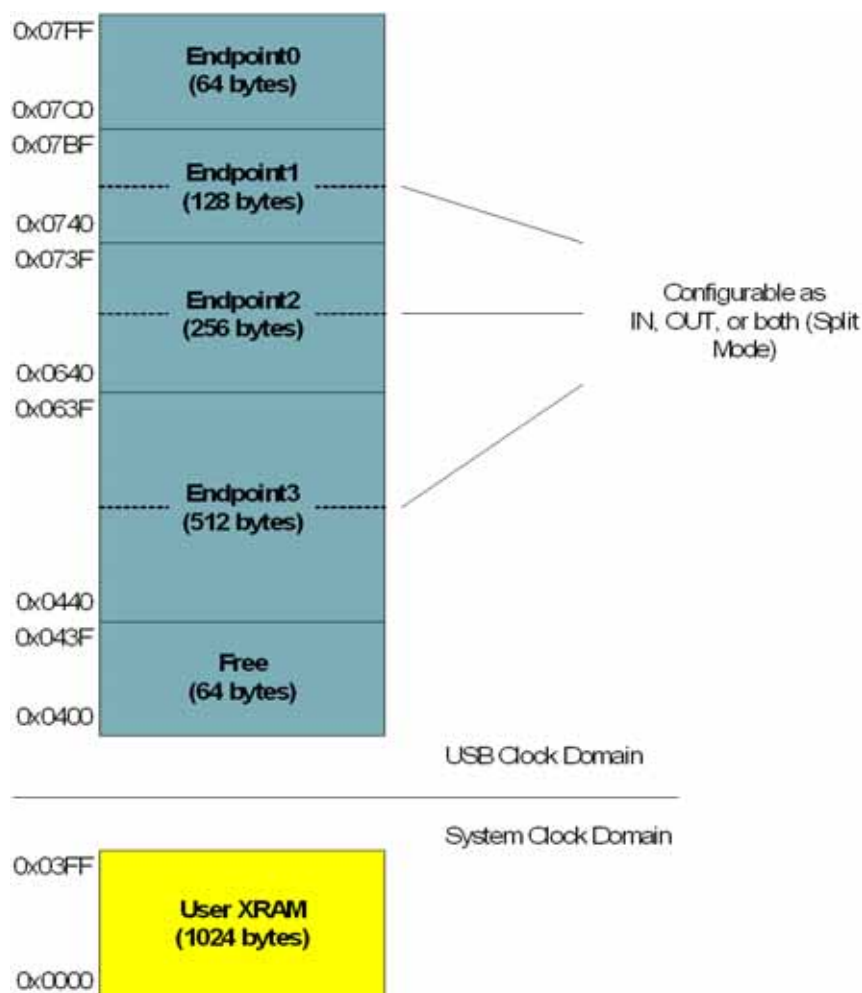


Figure 21.3. USB FIFO Allocation

21.5.1. FIFO Split Mode

The FIFO space for Endpoints1-3 can be split such that the upper half of the FIFO space is used by the IN endpoint, and the lower half is used by the OUT endpoint. For example: if the Endpoint3 FIFO is configured for Split Mode, the upper 256 bytes (0x0540 to 0x063F) are used by Endpoint3 IN and the lower 256 bytes (0x0440 to 0x053F) are used by Endpoint3 OUT.

If an endpoint FIFO is not configured for Split Mode, that endpoint IN/OUT pair's FIFOs are combined to form a single IN or OUT FIFO. In this case only one direction of the endpoint IN/OUT pair may be used at a time. The endpoint direction (IN/OUT) is determined by the DIRSEL bit in the corresponding endpoint's EINCSRH register (see SFR Definition 21.13).

USB Register Definition 21.18. E0CNT: USB0 Endpoint0 Data Count

Bit	7	6	5	4	3	2	1	0
Name	E0CNT[6:0]							
Type	R							
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x16

Bit	Name	Function
7	Unused	Read = 0b. Write = don't care.
6:0	E0CNT[6:0]	Endpoint 0 Data Count. This 7-bit number indicates the number of received data bytes in the Endpoint 0 FIFO. This number is only valid while bit OPRDY is a 1.

21.11. Configuring Endpoints1-3

Endpoints1-3 are configured and controlled through their own sets of the following control/status registers: IN registers EINCSSL and EINCSRH, and OUT registers EOUTCSRL and EOUTCSRH. Only one set of endpoint control/status registers is mapped into the USB register address space at a time, defined by the contents of the INDEX register (USB Register Definition 21.4).

Endpoints1-3 can be configured as IN, OUT, or both IN/OUT (Split Mode) as described in Section 21.5.1. The endpoint mode (Split/Normal) is selected via the SPLIT bit in register EINCSRH.

When SPLIT = 1, the corresponding endpoint FIFO is split, and both IN and OUT pipes are available.

When SPLIT = 0, the corresponding endpoint functions as either IN or OUT; the endpoint direction is selected by the DIRSEL bit in register EINCSRH.

Endpoints1-3 can be disabled individually by the corresponding bits in the ENABLE register. When an Endpoint is disabled, it will not respond to bus traffic or stall the bus. All Endpoints are enabled by default.

C8051F380/1/2/3/4/5/6/7/C

USB Register Definition 21.23. EOUTCSRH: USB0 OUT Endpoint Control High Byte

Bit	7	6	5	4	3	2	1	0
Name	DBOEN	ISO						
Type	R/W	R/W	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x15

Bit	Name	Function
7	DBOEN	Double-buffer Enable. 0: Double-buffering disabled for the selected OUT endpoint. 1: Double-buffering enabled for the selected OUT endpoint.
6	ISO	Isochronous Transfer Enable. This bit enables/disables isochronous transfers on the current endpoint. 0: Endpoint configured for bulk/interrupt transfers. 1: Endpoint configured for isochronous transfers.
5:0	Unused	Read = 000000b. Write = don't care.

USB Register Definition 21.24. EOUTCNTL: USB0 OUT Endpoint Count Low

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

USB Register Address = 0x16

Bit	Name	Function
7:0	EOCL[7:0]	OUT Endpoint Count Low Byte. EOCL holds the lower 8-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = 1.

SFR Definition 22.7. SMB0ADM: SMBus0 Slave Address Mask

Bit	7	6	5	4	3	2	1	0
Name	SLVM0[6:0]							EHACK0
Type	R/W							R/W
Reset	1	1	1	1	1	1	1	0

SFR Address = 0xCE; SFR Page = 0

Bit	Name	Function
7:1	SLVM0[6:0]	SMBus0 Slave Address Mask. Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM0[6:0] enables comparisons with the corresponding bit in SLV0[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK0	Hardware Acknowledge Enable. Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled.

SFR Definition 22.8. SMB1ADR: SMBus1 Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV1[6:0]							GC1
Type	R/W							R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCF; SFR Page = F

Bit	Name	Function
7:1	SLV1[6:0]	SMBus1 Hardware Slave Address. Defines the SMBus1 Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM1[6:0] are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC1	General Call Address Enable. When hardware address recognition is enabled (EHACK1 = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

23.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction 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 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

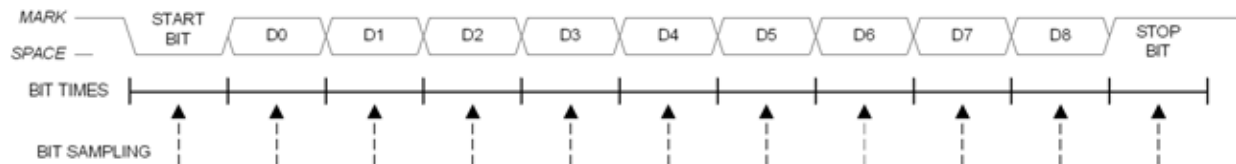


Figure 23.5. 9-Bit UART Timing Diagram

C8051F380/1/2/3/4/5/6/7/C

24.1. Baud Rate Generator

The UART1 baud rate is generated by a dedicated 16-bit timer which runs from the controller's core clock (SYSCLK), and has prescaler options of 1, 4, 12, or 48. The timer and prescaler options combined allow for a wide selection of baud rates over many SYSCLK frequencies.

The baud rate generator is configured using three registers: SBCON1, SBRLH1, and SBRL1. The UART1 Baud Rate Generator Control Register (SBCON1, SFR Definition) enables or disables the baud rate generator, and selects the prescaler value for the timer. The baud rate generator must be enabled for UART1 to function. Registers SBRLH1 and SBRL1 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16. The baud rate for UART1 is defined in Equation 24.1.

$$\text{Baud Rate} = \frac{\text{SYSCLK}}{(65536 - (\text{SBRLH1}:\text{SBRL1}))} \times \frac{1}{2} \times \frac{1}{\text{Prescaler}}$$

Equation 24.1. UART1 Baud Rate

A quick reference for typical baud rates and system clock frequencies is given in Table 24.1.

Table 24.1. Baud Rate Generator Settings for Standard Baud Rates

	Target Baud Rate (bps)	Actual Baud Rate (bps)	Baud Rate Error	Oscillator Divide Factor	SB1PS[1:0] (Prescaler Bits)	Reload Value in SBRLH1:SBRL1
SYSCLK = 12 MHz	230400	230769	0.16%	52	11	0xFFE6
	115200	115385	0.16%	104	11	0xFFCC
	57600	57692	0.16%	208	11	0xFF98
	28800	28846	0.16%	416	11	0xFF30
	14400	14388	0.08%	834	11	0xFE5F
	9600	9600	0.0%	1250	11	0xFD8F
	2400	2400	0.0%	5000	11	0xF63C
	1200	1200	0.0%	10000	11	0xEC78
SYSCLK = 24 MHz	230400	230769	0.16%	104	11	0xFFCC
	115200	115385	0.16%	208	11	0xFF98
	57600	57692	0.16%	416	11	0xFF30
	28800	28777	0.08%	834	11	0xFE5F
	14400	14406	0.04%	1666	11	0xFCBF
	9600	9600	0.0%	2500	11	0xFB1E
	2400	2400	0.0%	10000	11	0xEC78
	1200	1200	0.0%	20000	11	0xD8F0
SYSCLK = 48 MHz	230400	230769	0.16%	208	11	0xFF98
	115200	115385	0.16%	416	11	0xFF30
	57600	57554	0.08%	834	11	0xFE5F
	28800	28812	0.04%	1666	11	0xFCBF
	14400	14397	0.02%	3334	11	0xF97D
	9600	9600	0.0%	5000	11	0xF63C
	2400	2400	0.0%	20000	11	0xD8F0
	1200	1200	0.0%	40000	11	0xB1E0

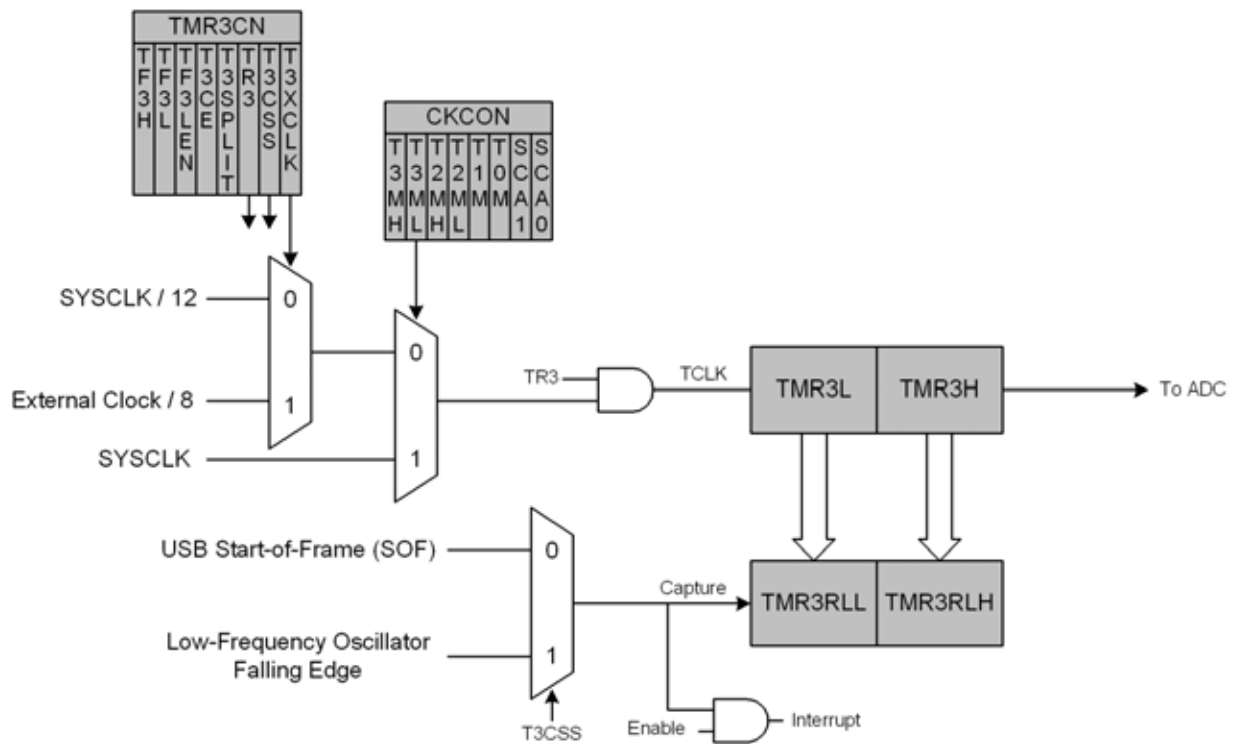


Figure 26.10. Timer 3 Capture Mode (T3SPLIT = 0)

When T3SPLIT = 1, the Timer 3 registers (TMR3H and TMR3L) act as two 8-bit counters. Each counter counts up independently and overflows from 0xFF to 0x00. Each time a capture event is received, the contents of the Timer 3 registers are latched into the Timer 3 Reload registers (TMR3RLH and TMR3RLL). A Timer 3 interrupt is generated if enabled.

26.5. Timer 5

Timer 5 is a 16-bit timer formed by two 8-bit SFRs: TMR5L (low byte) and TMR5H (high byte). Timer 5 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T5SPLIT bit (TMR5CN.3) defines

Timer 5 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

26.5.1. 16-bit Timer with Auto-Reload

When T5SPLIT (TMR5CN.3) is zero, Timer 5 operates as a 16-bit timer with auto-reload. Timer 5 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 5 reload registers (TMR5RLH and TMR5RLL) is loaded into the Timer 5 register as shown in Figure 26.14, and the Timer 5 High Byte Overflow Flag (TMR5CN.7) is set. If Timer 5 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 5 overflow. Additionally, if Timer 5 interrupts are enabled and the TF5LEN bit is set (TMR5CN.5), an interrupt will be generated each time the lower 8 bits (TMR5L) overflow from 0xFF to 0x00.

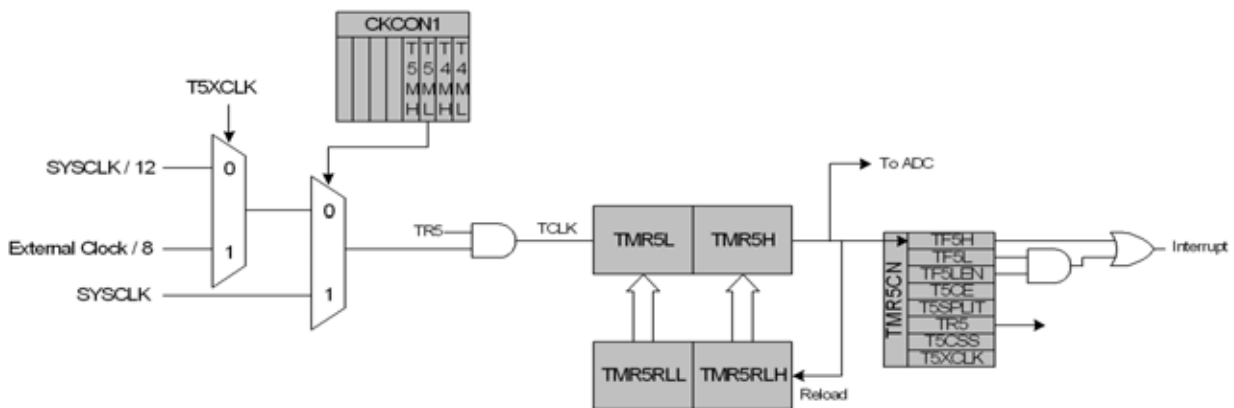


Figure 26.14. Timer 5 16-Bit Mode Block Diagram

SFR Definition 27.4. 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 = All Pages

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.
Note: When the WDTE bit is set to 1, the PCA0L register cannot be modified by software. To change the contents of the PCA0L register, the Watchdog Timer must first be disabled.		

SFR Definition 27.5. 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 = All Pages

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 27.1).
Note: When the WDTE bit is set to 1, the PCA0H register cannot be modified by software. To change the contents of the PCA0H register, the Watchdog Timer must first be disabled.		