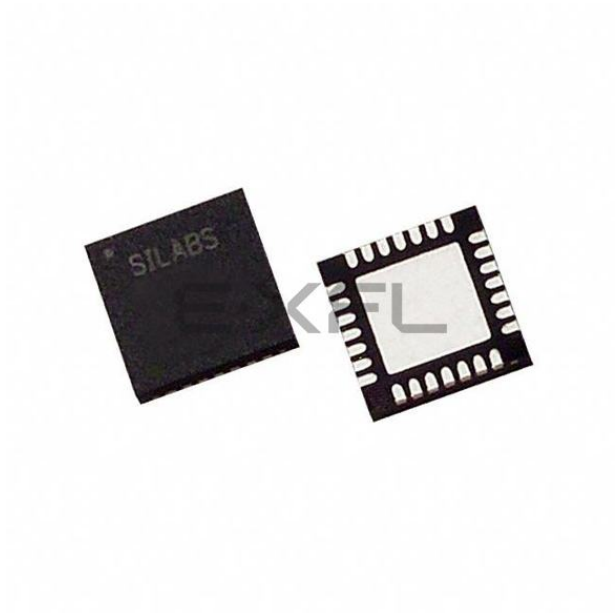

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	100MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 17x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f362-c-gmr

List of Figures

List of Registers

nels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the ADC and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized.

The C8051F360DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F36x MCUs. The kit includes software with a developer's studio and debugger, an integrated 8051 assembler, and a debug adapter. It also has a target application board with the associated MCU installed and prototyping area, plus the required cables, and wall-mount power supply. The Development Kit requires a PC running Windows98SE or later.

The Silicon Labs IDE interface is a vastly superior developing and debugging configuration, compared to standard MCU emulators that use on-board "ICE Chips" and require the MCU in the application board to be socketed. Silicon Labs' debug paradigm increases ease of use and preserves the performance of the precision analog peripherals.

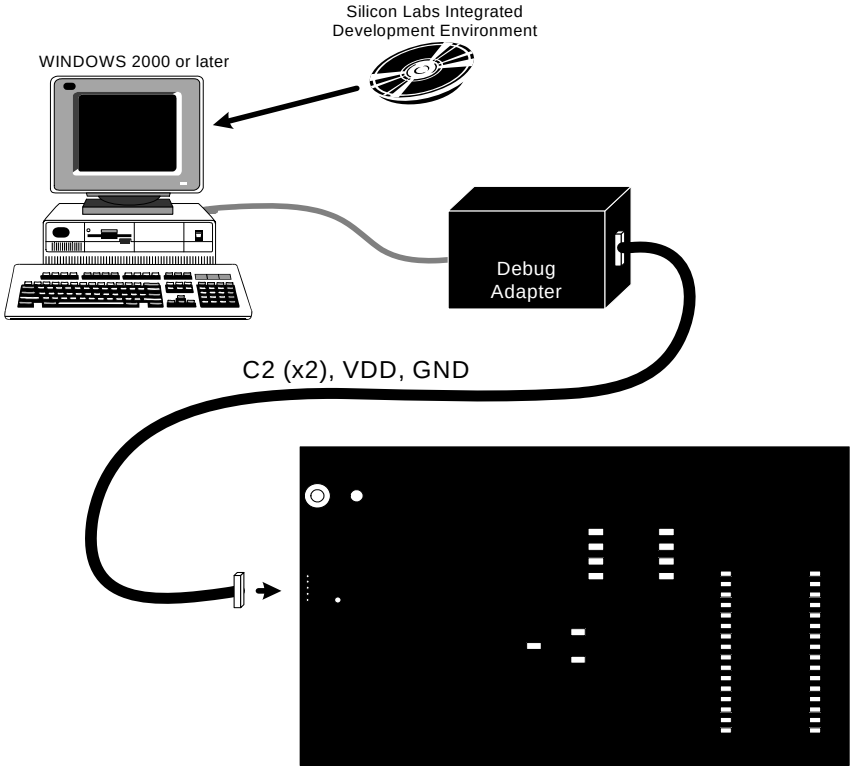

Figure 1.7. Development/In-System Debug Diagram

1.4. Programmable Digital I/O and Crossbar

C8051F36x devices include up to 39 I/O pins (four byte-wide Ports and one 7-bit-wide Port). The C8051F36x Ports behave like typical 8051 Ports with a few enhancements. Each Port pin may be configured as an analog input or a digital I/O pin. Pins selected as digital I/Os may additionally be configured for push-pull or open-drain output. The “weak pullups” that are fixed on typical 8051 devices may be globally disabled, providing power savings capabilities.

SFR Definition 9.1. SFR0CN: SFR Page Control

SFR Page: F
SFR Address: 0xE5

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SFRPGEN	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–1: RESERVED. Read = 0000000b. Must Write 0000000b.
Bit 0: SFRPGEN: SFR Automatic Page Control Enable.
Upon interrupt, the C8051 Core will vector to the specified interrupt service routine and automatically switch to SFR page 0. This bit is used to control this autopaging function.
0: SFR Automatic Paging disabled. C8051 core will not automatically change to SFR page 0.
1: SFR Automatic Paging enabled. Upon interrupt, the C8051 will automatically switch to SFR page 0.

SFR Definition 9.2. SFRPAGE: SFR Page

SFR Page: all pages
SFR Address: 0xA7

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SFR Page Bits: Byte Represents the SFR Page the C8051 MCU uses when reading or modifying SFR's.
Write: Sets the SFR Page.
Read: Byte is the SFR page the C8051 MCU is using.

When enabled in the SFR Page Control Register (SFR0CN), the C8051 will automatically switch to SFR Page 0x00 and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt).
SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and **not** by reading/writing to the SFRPAGE register)

Table 9.3. Special Function Registers

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

Register	Address	SFR Page	Description	Page No.
ACC	0xE0	All Pages	Accumulator	page 104
ADC0CF	0xBC	All Pages	ADC0 Configuration	page 56[1]
ADC0CN	0xE8	All Pages	ADC0 Control	page 57[1]
ADC0GTH	0xC4	All Pages	ADC0 Greater-Than High Byte	page 58[1]
ADC0GTL	0xC3	All Pages	ADC0 Greater-Than Low Byte	page 58[1]
ADC0H	0xBE	All Pages	ADC0 Data Word High Byte	page 56[1]
ADC0L	0xBD	All Pages	ADC0 Data Word Low Byte	page 56[1]
ADC0LTH	0xC6	All Pages	ADC0 Less-Than High Byte	page 59[1]
ADC0LTL	0xC5	All Pages	ADC0 Less-Than Low Byte	page 59[1]
AMX0N	0xBA	All Pages	AMUX0 Negative Channel Select	page 55[1]
AMX0P	0xBB	All Pages	AMUX0 Positive Channel Select	page 54[1]
B	0xF0	All Pages	B Register	page 104
CCH0CN	0x84	F	Cache Control	page 149
CCH0LC	0xD2	F	Cache Lock	page 151
CCH0MA	0xD3	F	Cache Miss Accumulator	page 152
CCH0TN	0xC9	F	Cache Tuning	page 150
CKCON	0x8E	All Pages	Clock Control	page 252
CLKSEL	0x8F	F	System Clock Select	page 173
CPT0CN	0x9B	All Pages	Comparator0 Control	page 73
CPT0MD	0x9D	All Pages	Comparator0 Configuration	page 75
CPT0MX	0x9F	All Pages	Comparator0 MUX Selection	page 74
CPT1CN	0x9A	All Pages	Comparator1 Control	page 76
CPT1MD	0x9C	All Pages	Comparator1 Configuration	page 78
CPT1MX	0x9E	All Pages	Comparator1 MUX Selection	page 77
DPH	0x83	All Pages	Data Pointer High Byte	page 102
DPL	0x82	All Pages	Data Pointer Low Byte	page 102
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	page 112
EIE2	0xE7	All Pages	Extended Interrupt Enable 2	page 114
EIP1	0xCE	F	Extended Interrupt Priority 1	page 113
EIP2	0xCF	F	Extended Interrupt Priority 2	page 114

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

9.4.7. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic ‘1’. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic ‘0’, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the data sheet associated with their corresponding system function.

SFR Definition 9.5. SP: Stack Pointer

SFR Page: all pages
SFR Address: 0x81

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SP: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 9.6. DPL: Data Pointer Low Byte

SFR Page all pages
SFR Address: 0x82

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

SFR Definition 9.7. DPH: Data Pointer High Byte

SFR Page: all pages
SFR Address: 0x83

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: DPH: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

11.7.2. Multiply Only Example

The example below implements the equation:

$$4660 \times -292 = -1360720$$

```
MOV   MAC0CF, #01h    ; Use integer numbers, and multiply only mode (add to zero)
MOV   MAC0AH, #12h    ; Load MAC0A register with 1234 hex = 4660 decimal
MOV   MAC0AL, #34h
MOV   MAC0BH, #FEh    ; Load MAC0B register with FEDC hex = -292 decimal
MOV   MAC0BL, #DCh    ; This line initiates the Multiply operation
NOP
NOP                   ; After this instruction, the Accumulator should be equal to
                      ; FFFFEB3CB0 hex = -1360720 decimal. The MAC0STA register should
                      ; be 0x01, indicating a negative result.
NOP                   ; After this instruction, the Rounding register is updated
```

11.7.3. MAC0 Accumulator Shift Example

The example below shifts the MAC0 accumulator left one bit, and then right two bits:

```
MOV   MAC0OVR, #40h   ; The next few instructions load the accumulator with the value
MOV   MAC0ACC3, #88h  ; 4088442211 Hex.
MOV   MAC0ACC2, #44h
MOV   MAC0ACC1, #22h
MOV   MAC0ACC0, #11h
MOV   MAC0CF, #20h    ; Initiate a Left-shift
NOP                   ; After this instruction, the accumulator should be 0x8110884422
NOP                   ; The rounding register is updated after this instruction
MOV   MAC0CF, #30h    ; Initiate a Right-shift
MOV   MAC0CF, #30h    ; Initiate a second Right-shift
NOP                   ; After this instruction, the accumulator should be 0xE044221108
NOP                   ; The rounding register is updated after this instruction
```

13.4. Flash Read Timing

On reset, the C8051F36x Flash read timing is configured for operation with system clocks up to 25 MHz. If the system clock will not be increased above 25 MHz, then the Flash timing registers may be left at their reset value.

For every Flash read or fetch, the system provides an internal Flash read strobe to the Flash memory. The Flash read strobe lasts for one or two system clock cycles, based on the FLRT bits (FLSCL.4 and FLSCL.5). **If the system clock is greater than 25 MHz, the FLRT bit must be changed to the appropriate setting.** Otherwise, data read or fetched from Flash may not represent the actual contents of Flash. When the Flash read strobe is asserted, Flash memory is active. When it is de-asserted, Flash memory is in a low power state.

The recommended procedure for updating FLRT is:

Step 1. Select SYSCLK to 25 MHz or less.
Step 2. Disable the prefetch engine (CHPFEN = ‘0’ in CCH0CN register).
Step 3. Set the FLRT bits to the appropriate setting for the SYSCLK.
Step 4. Enable the prefetch engine (CHPFEN = ‘1’ in CCH0CN register).

SFR Definition 13.3. FLSCL: Flash Memory Control

SFR Page: 0
SFR Address: 0xB6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	FLRT		Reserved	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 00b. Write = don’t care.
Bits 5–4: FLRT: Flash Read Time.
These bits should be programmed to the smallest allowed value, according to the system clock speed.
00: SYSCLK ≤ 25 MHz.
01: SYSCLK ≤ 50 MHz.
10: SYSCLK ≤ 75 MHz.
11: SYSCLK ≤ 100 MHz.
Bits 3–0: RESERVED. Read = 0000b. Must Write 0000b.

Important Note: When changing the FLRT bits to a lower setting (e.g. when changing from a value of 11b to 00b), cache reads, cache writes, and the prefetch engine should be disabled using the CCH0CN register (see SFR Definition 14.1).

14. Branch Target Cache

The C8051F36x device families incorporate a 32x4 byte branch target cache with a 4-byte prefetch engine. Because the access time of the Flash memory is 40 ns, and the minimum instruction time is 10 ns (C8051F360/1/2/3/4/5/6/7) or 20 ns (C8051F368/9), the branch target cache and prefetch engine are necessary for full-speed code execution. Instructions are read from Flash memory four 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 alone allows instructions to be executed at full speed. When a code branch occurs, a search is performed for the branch target (destination address) in the cache. If the branch target information is found in the cache (called a “cache hit”), the instruction data is read from the cache and immediately returned to the CIP-51 with no delay in code execution. If the branch target is not found in the cache (called a “cache miss”), the processor may be stalled for up to four clock cycles while the next set of four instructions is retrieved from Flash memory. Each time a cache miss occurs, the requested instruction data is written to the cache if allowed by the current cache settings. A data flow diagram of the interaction between the CIP-51 and the Branch Target Cache and Prefetch Engine is shown in Figure 14.1.

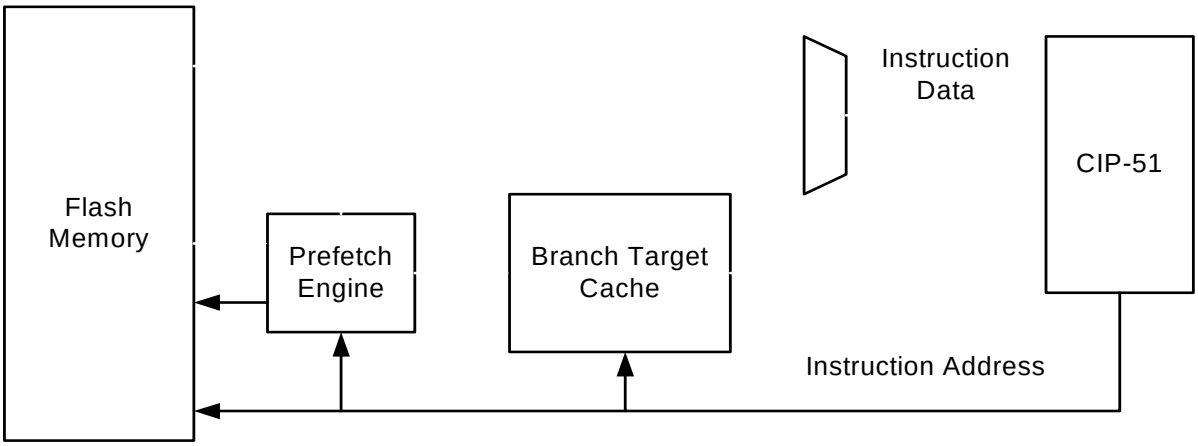

Figure 14.1. Branch Target Cache Data Flow

14.1. Cache and Prefetch Operation

The branch target cache maintains two sets of memory locations: “slots” and “tags”. A slot is where the cached instruction data from Flash is stored. Each slot holds four consecutive code bytes. A tag contains the 13 most significant bits of the corresponding Flash address for each four-byte slot. Thus, instruction data is always cached along four-byte boundaries in code space. A tag also contains a “valid bit”, which indicates whether a cache location contains valid instruction data. A special cache location (called the linear tag and slot), is reserved for use by the prefetch engine. The cache organization is shown in Figure 14.2. Each time a Flash read is requested, the address is compared with all valid cache tag locations (including the linear tag). If any of the tag locations match the requested address, the data from that slot is immediately provided to the CIP-51. If the requested address matches a location that is currently being read by the prefetch engine, the CIP-51 will be stalled until the read is complete. If a match is not found, the current prefetch operation is abandoned, and a new prefetch operation is initiated for the requested instruction data. When the prefetch operation is finished, the CIP-51 begins executing the instructions that were retrieved, and the prefetch engine begins reading the next four-byte word from Flash memory. If the newly-fetched data also meets the criteria necessary to be cached, it will be written to the cache in the slot indicated by the current replacement algorithm.

The replacement algorithm is selected with the Cache Algorithm bit, CHALGM (CCH0TN.3). When CHALGM is cleared to ‘0’, the cache will use the rebound algorithm to replace cache locations. The rebound algorithm replaces locations in order from the beginning of cache memory to the end, and then from the end of cache memory to the beginning. When CHALGM is set to ‘1’, the cache will use the pseudo-random algorithm to replace cache locations. The pseudo-random algorithm uses a pseudo-random number to determine which cache location to replace. The cache can be manually emptied by writing a ‘1’ to the CHFLUSH bit (CCH0CN.4).

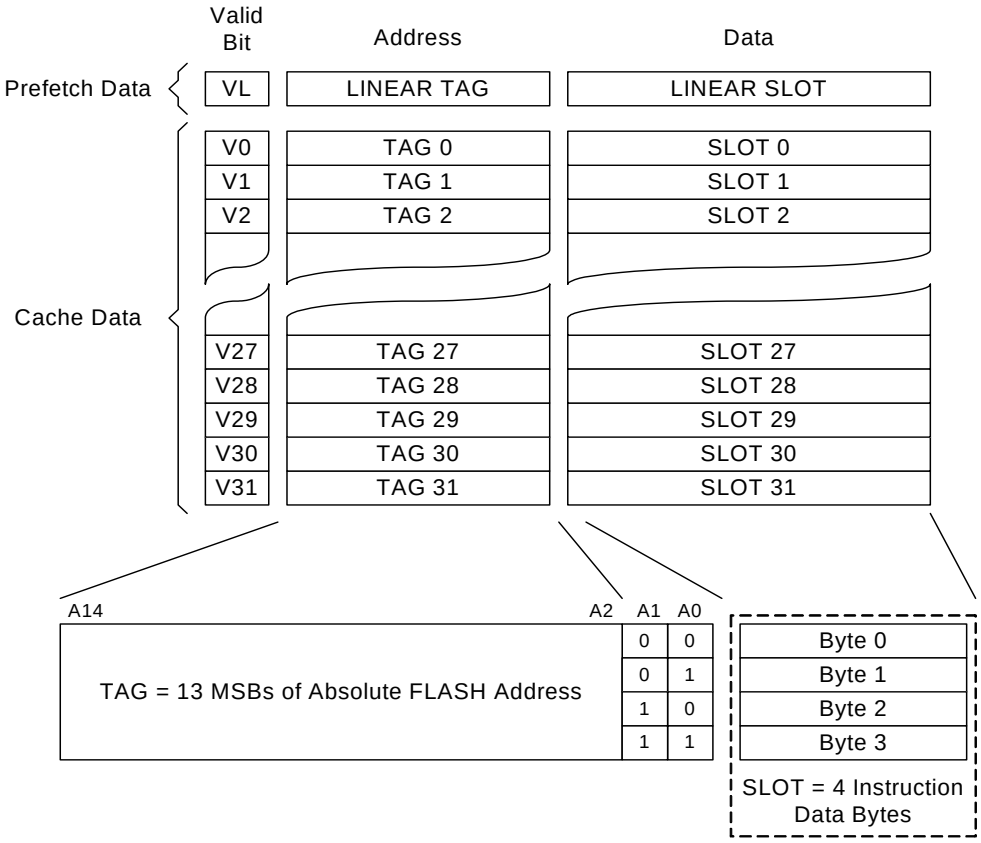

Figure 14.2. Branch Target Cache Organization

14.2. Cache and Prefetch Optimization

By default, the branch target cache is configured to provide code speed improvements for a broad range of circumstances. **In most applications, the cache control registers should be left in their reset states.** Sometimes it is desirable to optimize the execution time of a specific routine or critical timing loop. The branch target cache includes options to exclude caching of certain types of data, as well as the ability to pre-load and lock time-critical branch locations to optimize execution speed.

The most basic level of cache control is implemented with the Cache Miss Penalty Threshold bits, CHMSTH (CCH0TN.1–0). If the processor is stalled during a prefetch operation for more clock cycles than the number stored in CHMSTH, the requested data will be cached when it becomes available. The CHMSTH bits are set to zero by default, meaning that any time the processor is stalled, the new data will be cached. If, for example, CHMSTH is equal to 2, any cache miss causing a delay of 3 or 4 clock cycles will be cached, while a cache miss causing a delay of 1–2 clock cycles will not be cached.

16.6. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 16.1, Option 2. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation. If the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

$f = 1.23(10^3)/RC = 1.23\ (10^3)/[246 \times 50] = 0.1\ MHz = 100\ kHz$

Referring to the table in SFR Definition 16.5, the required XFCN setting is 010b. Programming XFCN to a higher setting in RC mode will improve frequency accuracy at a slightly increased external oscillator supply current.

16.7. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 16.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation from the equations below. Assume V_{DD} = 3.0 V and f = 75 kHz:

$f = KF / (C \times V_{DD})$
$0.075\ MHz = KF / (C \times 3.0)$

Since the frequency of roughly 75 kHz is desired, select the K Factor from the table in SFR Definition 16.5 as KF = 7.7:

$0.075\ MHz = 7.7 / (C \times 3.0)$

$C \times 3.0 = 7.7 / 0.075\ MHz$

$C = 102.6 / 3.0\ pF = 34.2\ pF$

Therefore, the XFCN value to use in this example is 010b.

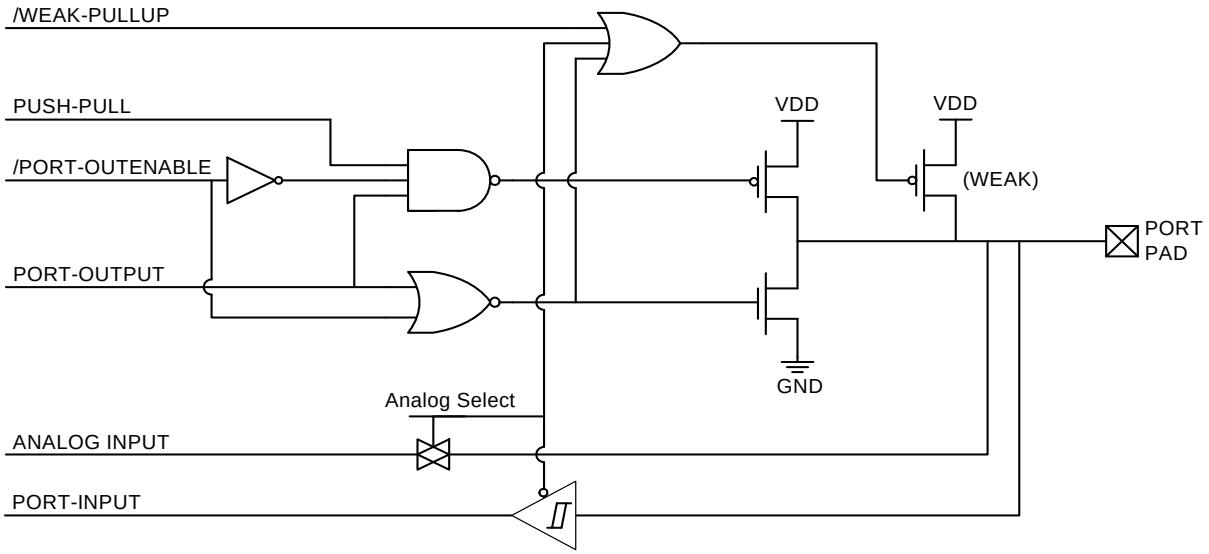

Figure 17.2. Port I/O Cell Block Diagram

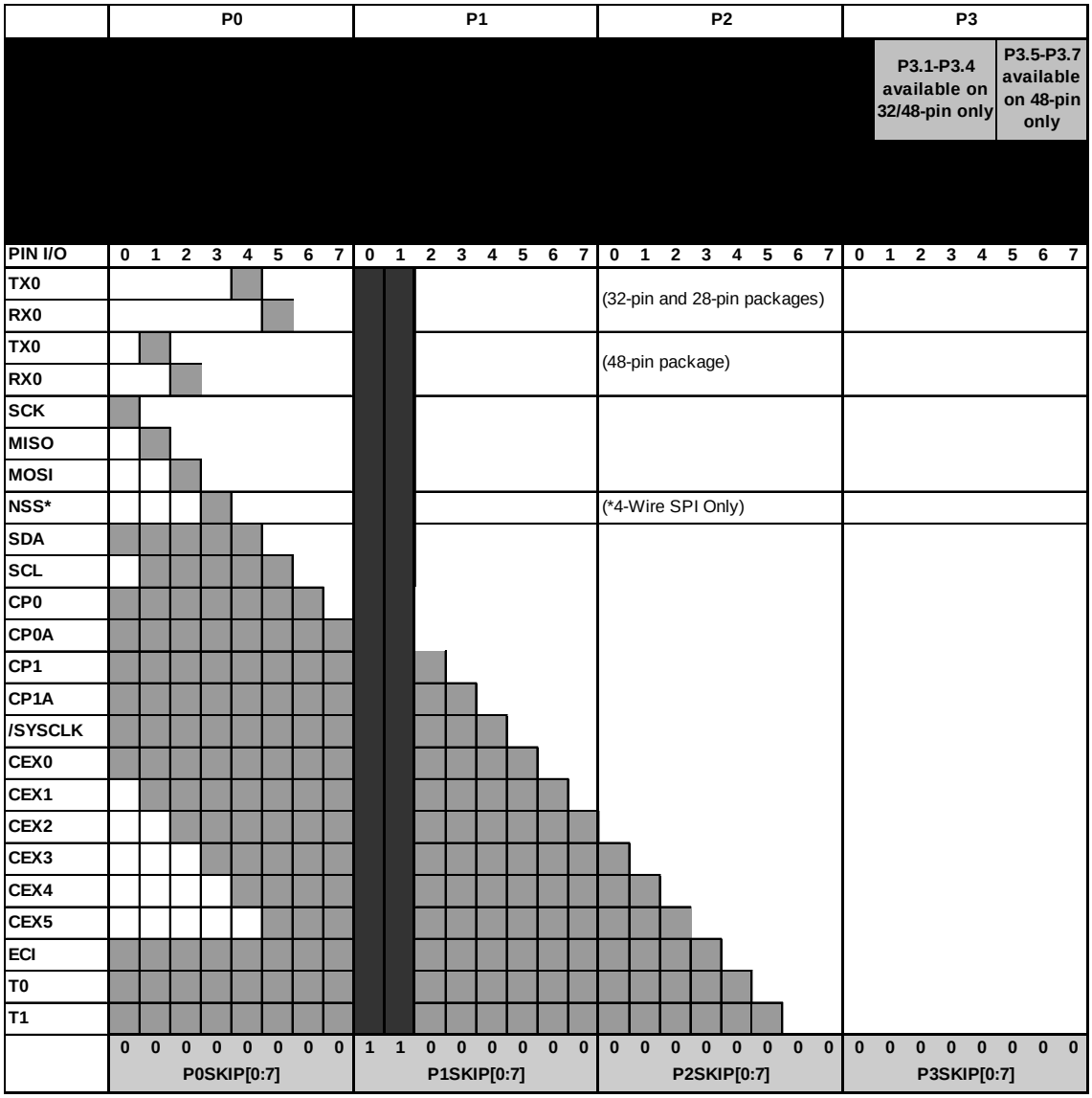

Figure 17.4. Crossbar Priority Decoder with Port Pins Skipped

Registers XBR0 and XBR1 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL); when the UART is selected, the Crossbar assigns both pins associated with the UART (TX and RX). UART0 pin assignments are fixed for bootloading purposes: UART TX0 is always assigned to P0.1 (C8051F360/3) or P0.4 (C8051F361/2/4/5/6/7/8/9); UART RX0 is always assigned to P0.2 (C8051F360/3) or P0.5 (C8051F361/2/4/5/6/7/8/9). Standard Port I/Os appear contiguously starting at P0.0 after prioritized functions and skipped pins are assigned.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1-NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

SFR Definition 17.6. P0SKIP: Port0 Skip

SFR Page: F
SFR Address: 0xD4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P0SKIP[7:0]: Port0 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P0.n pin is not skipped by the Crossbar.
1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 17.7. P0MAT: Port0 Match

SFR Page: 0
SFR Address: 0xF3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P0MAT[7:0]: Port0 Match Value.
These bits control the value that unmasked P0 Port pins are compared against. A Port Match event is generated if (P0 & P0MASK) does not equal (P0MAT & P0MASK).

SFR Definition 17.8. P0MASK: Port0 Mask

SFR Page: 0
SFR Address: 0xF4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P0MASK[7:0]: Port0 Mask Value.
These bits select which Port pins will be compared to the value stored in P0MAT.
0: Corresponding P0.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P0.n pin is compared to the corresponding bit in P0MAT.

SFR Definition 17.16. P2MDIN: Port2 Input Mode

SFR Page: F
SFR Address: 0xF3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Analog Input Configuration Bits for P2.7-P2.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P2.n pin is configured as an analog input.
1: Corresponding P2.n pin is not configured as an analog input.

SFR Definition 17.17. P2MDOUT: Port2 Output Mode

SFR Page: F
SFR Address: 0xA6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Output Configuration Bits for P2.7-P2.0 (respectively): ignored if corresponding bit in register P2MDIN is logic ‘0’.
0: Corresponding P2.n Output is open-drain.
1: Corresponding P2.n Output is push-pull.

SFR Definition 17.25. P4: Port4

SFR Page: all pages
SFR Address: 0xB5

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	01111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: UNUSED. Read = 0b. Write = don't care.
Bits 6–0: P4.[6:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P4MDOUT.n bit = 0).
Read - Directly reads Port pin.
0: P4.n pin is logic low.
1: P4.n pin is logic high.

SFR Definition 17.26. P4MDOUT: Port4 Output Mode

SFR Page: F
SFR Address: 0xAE

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: UNUSED. Read = 0b. Write = don't care.
Bits 6–0: Output Configuration Bits for P4.6-P4.0 (respectively).
0: Corresponding P4.n Output is open-drain.
1: Corresponding P4.n Output is push-pull.

21.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR3CN.3) defines the Timer 3 operation mode.

Timer 3 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 3 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

21.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 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 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 21.6, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

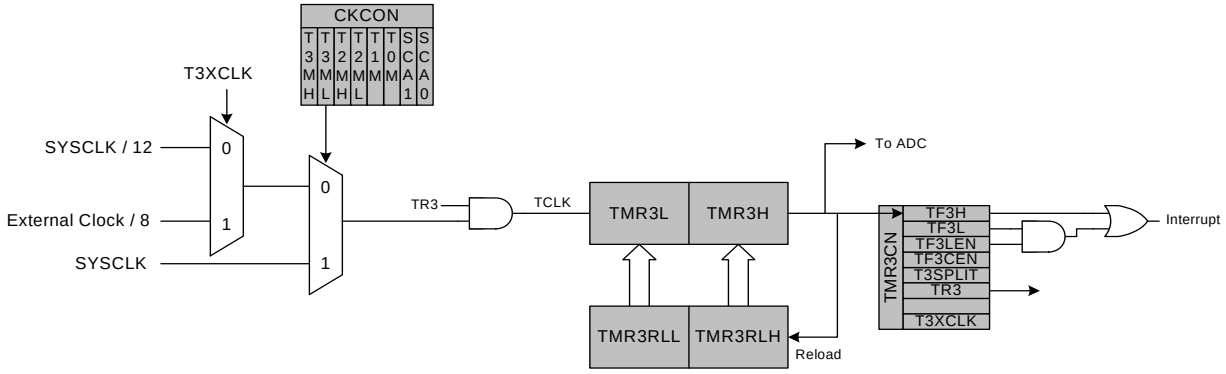

Figure 21.6. Timer 3 16-Bit Mode Block Diagram

21.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 21.7. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bit (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

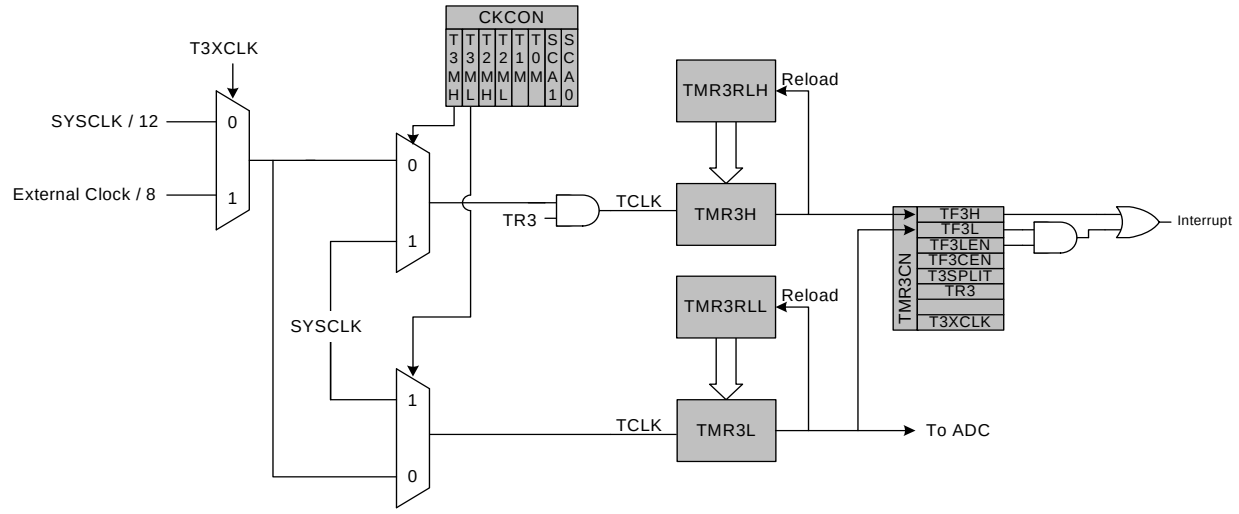

Figure 21.7. Timer 3 8-Bit Mode Block Diagram

22.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA0 counter/timer is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic ‘1’ and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to ‘0’; writing to PCA0CPHn sets ECOMn to ‘1’.

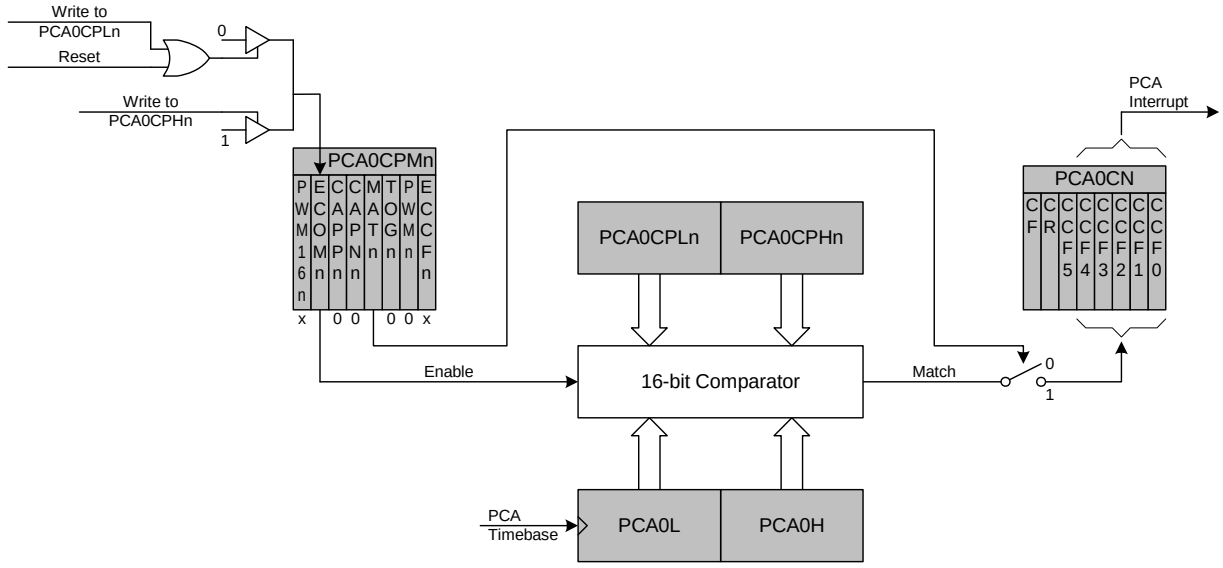

Figure 22.5. PCA Software Timer Mode Diagram

SFR Definition 22.7. PCA0CPHn: PCA0 Capture Module High Byte

SFR Page: PCA0CPH0: all pages, PCA0CPH1: all pages, PCA0CPH2: all pages, PCA0CPH3: all pages, PCA0CPH4: all pages, PCA0CPH5: all pages

SFR Address: PCA0CPH0: 0xFC, PCA0CPH1: 0xEA, PCA0CPH2: 0xEC, PCA0CPH3: 0xEE, PCA0CPH4: 0xFE, PCA0CPH5: 0xF6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: PCA0CPHn: PCA0 Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.