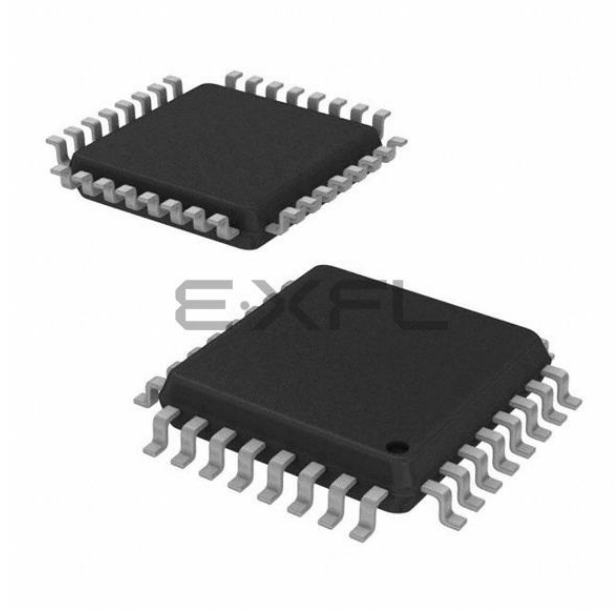

Welcome to E-XFL.COM

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	Discontinued at Digi-Key
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	29
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 21x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f368-gq

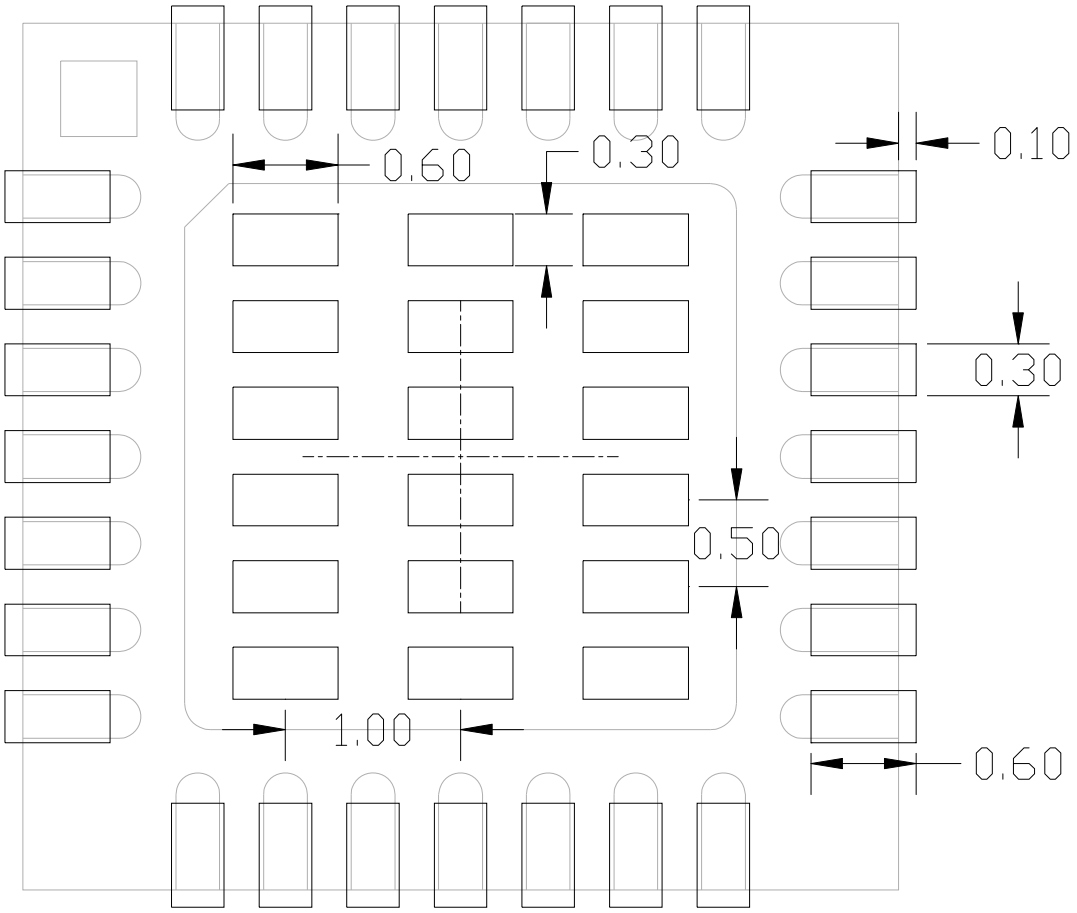

Figure 4.8. QFN-28 Solder Paste Recommendation

Table 5.1. ADC0 Electrical Characteristics

V_{DD} = 3.0 V, VREF = 2.40 V (REFSL=0), –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			10		bits
Integral Nonlinearity		—	±0.5	±1	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	±0.5	±1	LSB
Offset Error		–12	3	12	LSB
Full Scale Error	Differential mode	–5	1	5	LSB
Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion		53	58	—	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	–75	—	dB
Spurious-Free Dynamic Range		—	75	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	3	MHz
Conversion Time in SAR Clocks		13	—	—	clocks
Track/Hold Acquisition Time		300	—	—	ns
Throughput Rate		—	—	200	ksps
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ – GND) Differential (AIN+ – AIN–)	0 –VREF	— —	VREF VREF	V V
Absolute Pin Voltage with respect to GND	Single Ended or Differential	0	—	V_{DD}	V
Input Capacitance		—	5	—	pF
Temperature Sensor					
Linearity*		—	±0.2	—	°C
Slope		—	2.18	—	mV/ºC
Slope Error*		—	±172	—	µV/ºC
Offset	(Temp = 0 °C)	—	802	—	mV
Offset Error*		—	±18.5	—	mV
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Operating Mode, 200 ksps	—	450	900	µA
Power Supply Rejection		—	3	—	mV/V

***Note:** Represents one standard deviation from the mean. Includes ADC offset, gain, and linearity variations.

complete Port I/O configuration details. The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data.

SFR Definition 7.1. REF0CN: Reference Control

SFR Page: all pages
SFR Address: 0xD1

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	REFSL	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–4: UNUSED. Read = 0000b; Write = don't care.
Bit 3: REFSL: Voltage Reference Select.
This bit selects the source for the internal voltage reference.
0: VREF pin used as voltage reference.
1: V_{DD} used as voltage reference.
Bit 2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.
Bit 1: BIASE: Internal Analog Bias Generator Enable Bit.
0: Internal Bias Generator off.
1: Internal Bias Generator on.
Bit 0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer disabled.
1: Internal Reference Buffer enabled. Internal voltage reference driven on the VREF pin.

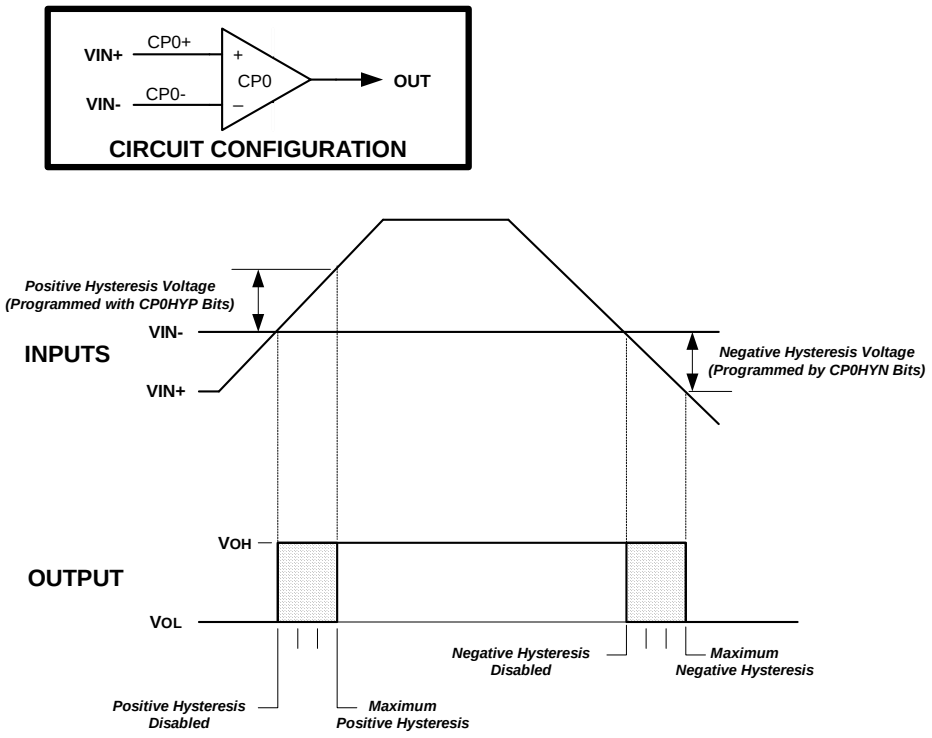

Figure 8.3. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via the Comparator Control registers CPT0CN and CPT1CN. 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 registers CPT0CN and CPT1CN (shown in SFR Definition 8.1 and SFR Definition 8.4). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN and CP1HYN bits. As shown in Figure 8.3, 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 and CP1HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “10. Interrupt Handler” on page 107). The CP0FIF or CP1FIF flag is set to logic ‘1’ upon a Comparator falling-edge occurrence, and the CP0RIF or CP1RIF 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 or CP1RIE to a logic ‘1’. The Comparator falling-edge interrupt mask is enabled by setting CP0FIE or CP1FIE to a logic ‘1’.

The output state of the Comparator can be obtained at any time by reading the CP0OUT or CP1OUT bit. The Comparator is enabled by setting the CP0EN or CP1EN 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. This Power Up Time is specified in Table 8.1 on page 79.

Table 9.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
JZ rel	Jump if A equals zero	2	2/3*
JNZ rel	Jump if A does not equal zero	2	2/3*
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/4*
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4*
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4*
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5*
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3*
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4*
NOP	No operation	1	1
* Branch instructions will incur a cache-miss penalty if the branch target location is not already stored in the Branch Target Cache. See Section “14. Branch Target Cache” on page 145 for more details.			

Notes on Registers, Operands and Addressing Modes:

Rn - Register R0-R7 of the currently selected register bank.

@Ri - Data RAM location addressed indirectly through R0 or R1.

rel - 8-bit, signed (2s complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct - 8-bit internal data location’s address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).

#data - 8-bit constant

#data16 - 16-bit constant

bit - Direct-accessed bit in Data RAM or SFR

addr11 - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-byte page of program memory as the first byte of the following instruction.

addr16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

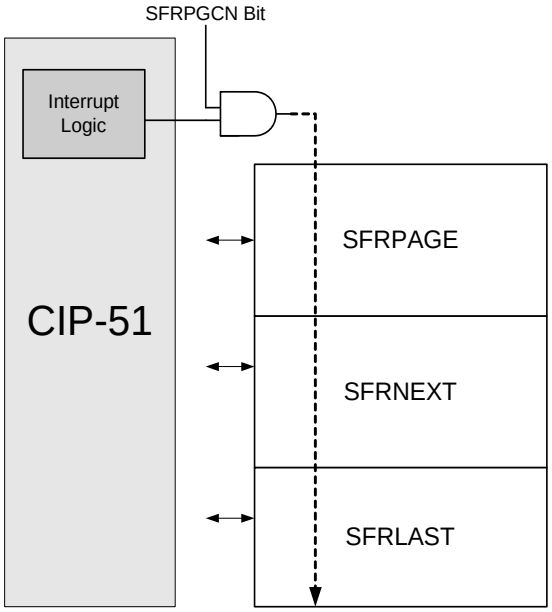

Figure 9.3. SFR Page Stack

Automatic hardware switching of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFR0CN). This function defaults to ‘enabled’ upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) is provided in Table 9.2. in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Note that certain SFR’s are accessible from ALL SFR pages, and are denoted by the background shading in the table. For example, the Port I/O registers P0, P1, P2, and P3 all have a shaded background, indicating these SFR’s are accessible from all SFR pages regardless of the SFRPAGE register value.

SFR Definition 10.1. IE: Interrupt Enable

SFR Page: all pages
SFR Address: 0xA8 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: EA: Global Interrupt Enable.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit 6: ESPI0: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of the SPI0 interrupts.
0: Disable all SPI0 interrupts.
1: Enable interrupt requests generated by SPI0.

Bit 5: ET2: Enable Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable interrupt requests generated by the TF2L or TF2H flags.

Bit 4: ES0: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit 3: ET1: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit 2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the /INT1 input.

Bit 1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit 0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the /INT0 input.

SFR Definition 11.7. MAC0ACC3: MAC0 Accumulator Byte 3

SFR Page: 0
SFR Address: 0xD5

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

Bits 7–0: Byte 3 (bits 31–24) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.8. MAC0ACC2: MAC0 Accumulator Byte 2

SFR Page: 0
SFR Address: 0xD4

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

Bits 7–0: Byte 2 (bits 23–16) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.9. MAC0ACC1: MAC0 Accumulator Byte 1

SFR Page: 0
SFR Address: 0xD3

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

Bits 7–0: Byte 1 (bits 15–8) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

13.2.1. Summary of Flash Security Options

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 unlocked pages, and user firmware executing on locked pages. Table 13.1 summarizes the Flash security features of the C8051F36x devices.

Table 13.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	FEDR	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	FEDR	FEDR
Erase page containing Lock Byte - Unlock all pages (if any page is locked)	Only C2DE	FEDR	FEDR
Lock additional pages (change '1's to '0's in the Lock Byte)	Not Permitted	FEDR	FEDR
Unlock individual pages (change '0's to '1's in the Lock Byte)	Not Permitted	FEDR	FEDR
Read, Write or Erase Reserved Area	Not Permitted	FEDR	FEDR

C2DE - C2 Device Erase (Erases all Flash pages including the page containing the Lock Byte)
FEDR - Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is '1' after reset)

- All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).
- Locking any Flash page also locks the page containing the Lock Byte.
- Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.
- If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.

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.

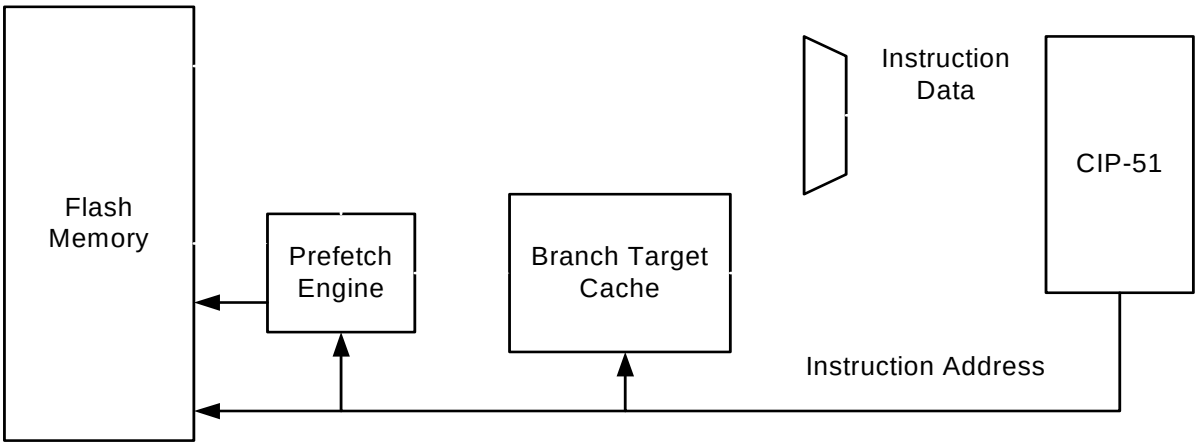

Figure 14.1. Branch Target Cache Data Flow

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.

Certain types of instruction data or certain blocks of code can also be excluded from caching. The destinations of RETI instructions are, by default, excluded from caching. To enable caching of RETI destinations, the CHRETI bit (CCH0CN.3) can be set to ‘1’. It is generally not beneficial to cache RETI destinations unless the same instruction is likely to be interrupted repeatedly (such as a code loop that is waiting for an interrupt to happen). Instructions that are part of an interrupt service routine (ISR) can also be excluded from caching. By default, ISR instructions are cached, but this can be disabled by clearing the CHISR bit (CCH0CN.2) to ‘0’. The other information that can be explicitly excluded from caching are the data returned by MOVC instructions. Clearing the CHMOV bit (CCH0CN.1) to ‘0’ will disable caching of MOVC data. If MOVC caching is allowed, it can be restricted to only use slot 0 for the MOVC information (excluding cache push operations). The CHFIXM bit (CCH0TN.2) controls this behavior.

Further cache control can be implemented by disabling all cache writes. Cache writes can be disabled by clearing the CHWREN bit (CCH0CN.7) to ‘0’. Although normal cache writes (such as those after a cache miss) are disabled, data can still be written to the cache with a cache push operation. Disabling cache writes can be used to prevent a non-critical section of code from changing the cache contents. Note that regardless of the value of CHWREN, a Flash write or erase operation automatically removes the affected bytes from the cache. Cache reads and the prefetch engine can also be individually disabled. Disabling cache reads forces all instructions data to execute from Flash memory or from the prefetch engine. To disable cache reads, the CHRDEN bit (CCH0CN.6) can be cleared to ‘0’. Note that when cache reads are disabled, cache writes will still occur (if CHWREN is set to ‘1’). Disabling the prefetch engine is accomplished using the CHPFEN bit (CCH0CN.5). When this bit is cleared to ‘0’, the prefetch engine will be disabled. If both CHPFEN and CHRDEN are ‘0’, code will execute at a fixed rate, as instructions become available from the Flash memory.

Cache locations can also be pre-loaded and locked with time-critical branch destinations. For example, in a system with an ISR that must respond as fast as possible, the entry point for the ISR can be locked into a cache location to minimize the response latency of the ISR. Up to 30 locations can be locked into the cache at one time. Instructions are locked into cache by enabling cache push operations with the CHPUSH bit (CCH0LC.7). When CHPUSH is set to ‘1’, a MOVC instruction will cause the four-byte segment containing the data byte to be written to the cache slot location indicated by CHSLOT (CCH0LC.4-0). CHSLOT is them decremented to point to the next lockable cache location. This process is called a cache push operation. Cache locations that are above CHSLOT are “locked”, and cannot be changed by the processor core, as shown in Figure 14.3. Cache locations can be unlocked by using a cache pop operation. A cache pop is performed by writing a ‘1’ to the CHPOP bit (CCH0LC.6). When a cache pop is initiated, the value of CHSLOT is incremented. This unlocks the most recently locked cache location, but does not remove the information from the cache. Note that a cache pop should not be initiated if CHSLOT is equal to 11110b. Doing so may have an adverse effect on cache performance. **Important: Although locking cache location 1 is not explicitly disabled by hardware, the entire cache will be unlocked when CHSLOT is equal to 00000b. Therefore, cache locations 1 and 0 must remain unlocked at all times.**

16.2.1. Calibrating the Internal L-F Oscillator

Timers 2 and 3 include capture functions that can be used to capture the oscillator frequency, when running from a known time base. When either Timer 2 or Timer 3 is configured for L-F Oscillator Capture Mode, a falling edge (Timer 2) or rising edge (Timer 3) of the low-frequency oscillator's output will cause a capture event on the corresponding timer. As a capture event occurs, the current timer value (TMRnH:TMRnL) is copied into the timer reload registers (TMRnRLH:TMRnRLL). By recording the difference between two successive timer capture values, the low-frequency oscillator's period can be calculated. The OSCLF bits can then be adjusted to produce the desired oscillator frequency.

SFR Definition 16.3. OSCLCN: Internal L-F Oscillator Control

SFR Page: F
SFR Address: 0xAD

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
OSCLEN	OSCLRDY	OSCLF3	OSCLF2	OSCLF1	OSCLF0	OSCLD1	OSCLD0	00vvvv00
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: OSCLEN: Internal L-F Oscillator Enable.
0: Internal L-F Oscillator Disabled.
1: Internal L-F Oscillator Enabled.

Bit 6: OSCLRDY: Internal L-F Oscillator Ready.
0: Internal L-F Oscillator frequency not stabilized.
1: Internal L-F Oscillator frequency stabilized.

Bits 5–2: OSCLF[3:0]: Internal L-F Oscillator Frequency Control bits.
Fine-tune control bits for the Internal L-F oscillator frequency. When set to 0000b, the L-F oscillator operates at its fastest setting. When set to 1111b, the L-F oscillator operates at its slowest setting.

Bits 1–0: OSCLD[1:0]: Internal L-F Oscillator Divider Select.
00: Divide by 8 selected.
01: Divide by 4 selected.
10: Divide by 2 selected.
11: Divide by 1 selected.

Table 16.2. Internal Low Frequency Oscillator Electrical Characteristics

–40°C to +85°C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	OSCLD = 11b	72	80	88	kHz
Oscillator Supply Current (from V_{DD})	25 °C, V_{DD} = 3.0 V, OSCLCN.7 = 1	—	5.5	10	µA
Power Supply Sensitivity	Constant Temperature	—	2.4	—	%/V
Temperature Sensitivity	Constant Supply	—	30	—	ppm/°C

16.3. 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. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 16.1. A 10 MΩ resistor also must be wired across the XTAL1 and XTAL2 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 16.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 16.5).

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 P0.5 and P0.6 (C8051F360/3) or P0.2 and P0.3 (C8051F361/2/4/5/6/7/8/9) are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.6 (C8051F360/3) or P0.3 (C8051F361/2/4/5/6/7/8/9) is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “17.1. Priority Crossbar Decoder” on page 184 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 “17.2. Port I/O Initialization” on page 186 for details on Port input mode selection.

16.4. System Clock Selection

The internal oscillator requires little start-up time, and may be enabled and selected as the system clock in the same write to OSCICN. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use as the system clock. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to ‘1’ by hardware when the external oscillator is settled. To avoid reading a false XTLVLD, in crystal mode software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD. RC and C modes typically require no startup time. The PLL also requires time to lock onto the desired frequency, and the PLL Lock Flag (PLLLCK in register PLL0CN) is set to ‘1’ by hardware once the PLL is locked on the correct frequency.

The CLKSL1-0 bits in register CLKSEL select which oscillator source generates the system clock. CLKSL1-0 must be set to ‘01’ for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals, such as the timers and PCA, when the internal oscillator or the PLL is selected as the system clock. The system clock may be switched on-the-fly between the internal and external oscillators or the PLL, so long as the selected oscillator source is enabled and settled.

18.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

18.2. SMBus Configuration

Figure 18.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

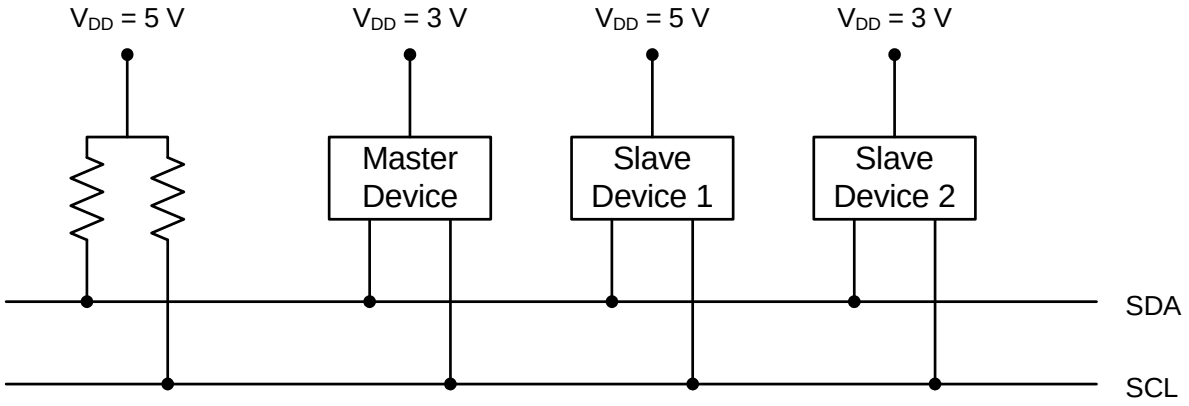

Figure 18.2. Typical SMBus Configuration

18.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. Note that it is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Each byte that is received (by a master or slave) must be acknowledged (ACK) with a low SDA during a high SCL (see Figure 18.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

Table 18.4. SMBus Status Decoding (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1
						Send NACK to indicate last byte, and send STOP.	0	1	0
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0
						Send ACK followed by repeated START.	1	0	1
						Send NACK to indicate last byte, and send repeated START.	1	0	0
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X
	0101	0	X	X	A STOP was detected while an addressed Slave Transmitter.	No action required (transfer complete).	0	0	X

19.3. Multiprocessor Communications

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

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

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

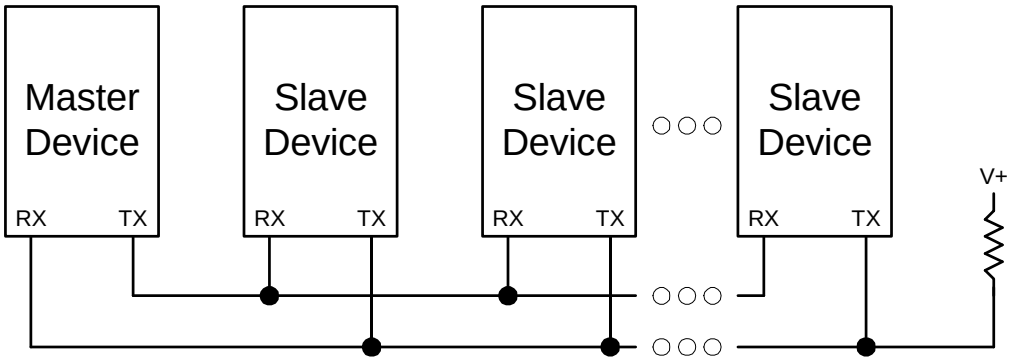

Figure 19.6. UART Multi-Processor Mode Interconnect Diagram

Table 19.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	**Baud Rate % Error**	**Oscillator Divide Factor**	**Timer Clock Source**	**SCA1–SCA0 (pre-scale select)[1]**	**T1M[1]**	**Timer 1 Reload Value (hex)**
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX[2]	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

20.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic ‘1’, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer’s contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic ‘0’, and disabled when NSS is logic ‘1’. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 20.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 20.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

20.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic ‘1’:

All of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic ‘1’ at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic ‘1’ if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic ‘1’ when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic ‘0’ to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic ‘1’ when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

SFR Definition 21.3. CKCON: Clock Control

SFR Page: all pages
SFR Address: 0x8E

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: T3MH: Timer 3 High Byte Clock Select.
This bit selects the clock supplied to the Timer 3 high byte if Timer 3 is configured in split 8-bit timer mode. T3MH is ignored if Time 3 is in any other mode.
0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 high byte uses the system clock.

Bit 6: T3ML: Timer 3 Low Byte Clock Select.
This bit selects the clock supplied to Timer 3. If Timer 3 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 low byte uses the system clock.

Bit 5: T2MH: Timer 2 High Byte Clock Select.
This bit selects the clock supplied to the Timer 2 high byte if Timer 2 is configured in split 8-bit timer mode. T2MH is ignored if Timer 2 is in any other mode.
0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 high byte uses the system clock.

Bit 4: T2ML: Timer 2 Low Byte Clock Select.
This bit selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 low byte uses the system clock.

Bit 3: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic ‘1’.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.

Bit 2: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic ‘1’.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Counter/Timer 0 uses the system clock.

Bits 1–0: SCA1–SCA0: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48
1	1	External clock divided by 8

Note: External clock divided by 8 is synchronized with the system clock.

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.

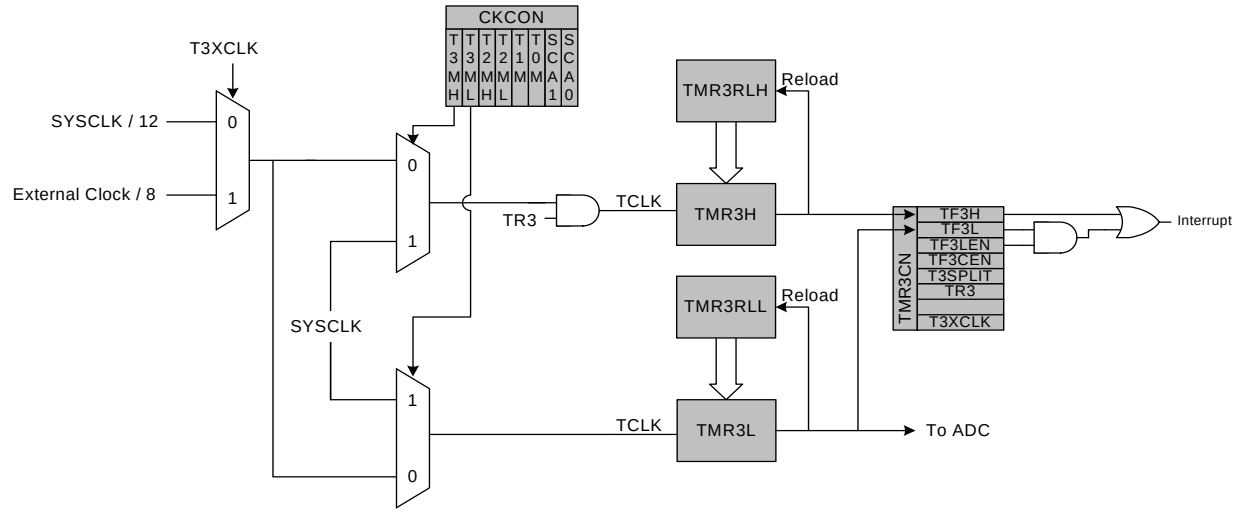

Figure 21.7. Timer 3 8-Bit Mode Block Diagram