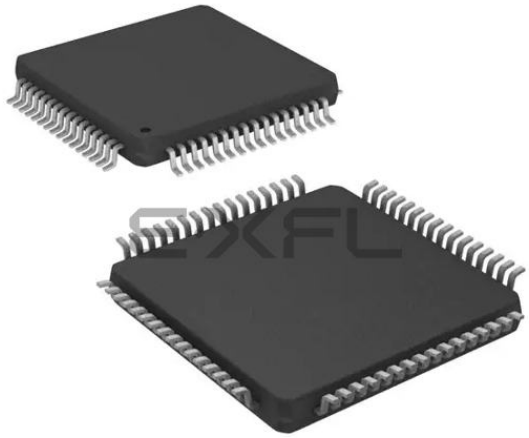

Welcome to E-XFL.COM

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	EBI/EMI, SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x8b, 8x10b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f123-gq

List of Figures

Table 4.1. Pin Definitions (Continued)

Name	Pin Numbers				Type	Description
	‘F120 ‘F122 ‘F124 ‘F126	‘F121 ‘F123 ‘F125 ‘F127	‘F130 ‘F132	‘F131 ‘F133		
AD0/D0/P3.0	54	47	54	47	D I/O	Bit 0 External Memory Address/Data bus (Multiplexed mode) Bit 0 External Memory Data bus (Non-multiplexed mode) Port 3.0 See Port Input/Output section for complete description.
AD1/D1/P3.1	53	46	53	46	D I/O	Port 3.1. See Port Input/Output section for complete description.
AD2/D2/P3.2	52	45	52	45	D I/O	Port 3.2. See Port Input/Output section for complete description.
AD3/D3/P3.3	51	44	51	44	D I/O	Port 3.3. See Port Input/Output section for complete description.
AD4/D4/P3.4	50	43	50	43	D I/O	Port 3.4. See Port Input/Output section for complete description.
AD5/D5/P3.5	49	42	49	42	D I/O	Port 3.5. See Port Input/Output section for complete description.
AD6/D6/P3.6	48	39	48	39	D I/O	Port 3.6. See Port Input/Output section for complete description.
AD7/D7/P3.7	47	38	47	38	D I/O	Port 3.7. See Port Input/Output section for complete description.
P4.0	98		98		D I/O	Port 4.0. See Port Input/Output section for complete description.
P4.1	97		97		D I/O	Port 4.1. See Port Input/Output section for complete description.
P4.2	96		96		D I/O	Port 4.2. See Port Input/Output section for complete description.
P4.3	95		95		D I/O	Port 4.3. See Port Input/Output section for complete description.
P4.4	94		94		D I/O	Port 4.4. See Port Input/Output section for complete description.

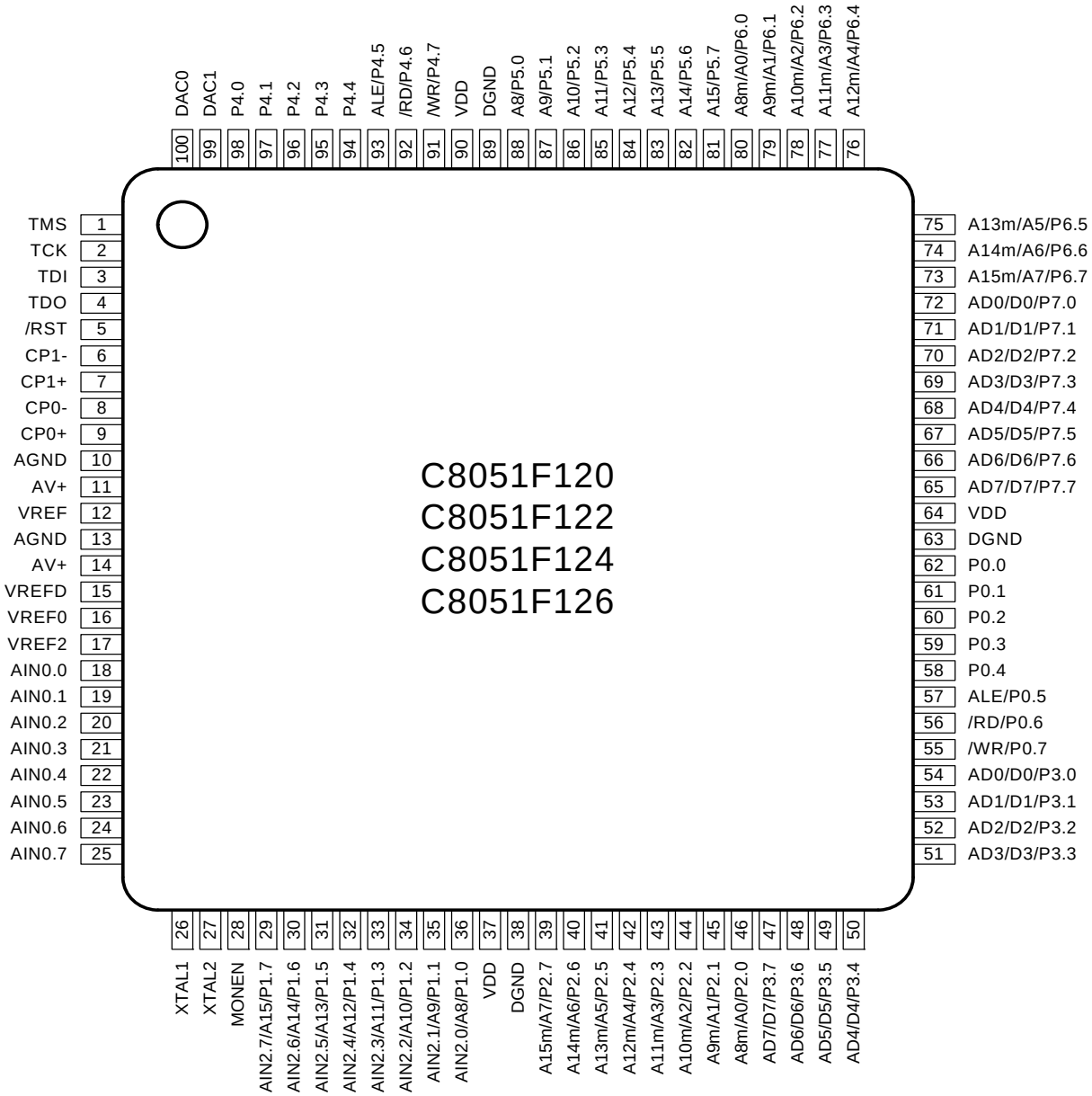

Figure 4.1. C8051F120/2/4/6 Pinout Diagram (TQFP-100)

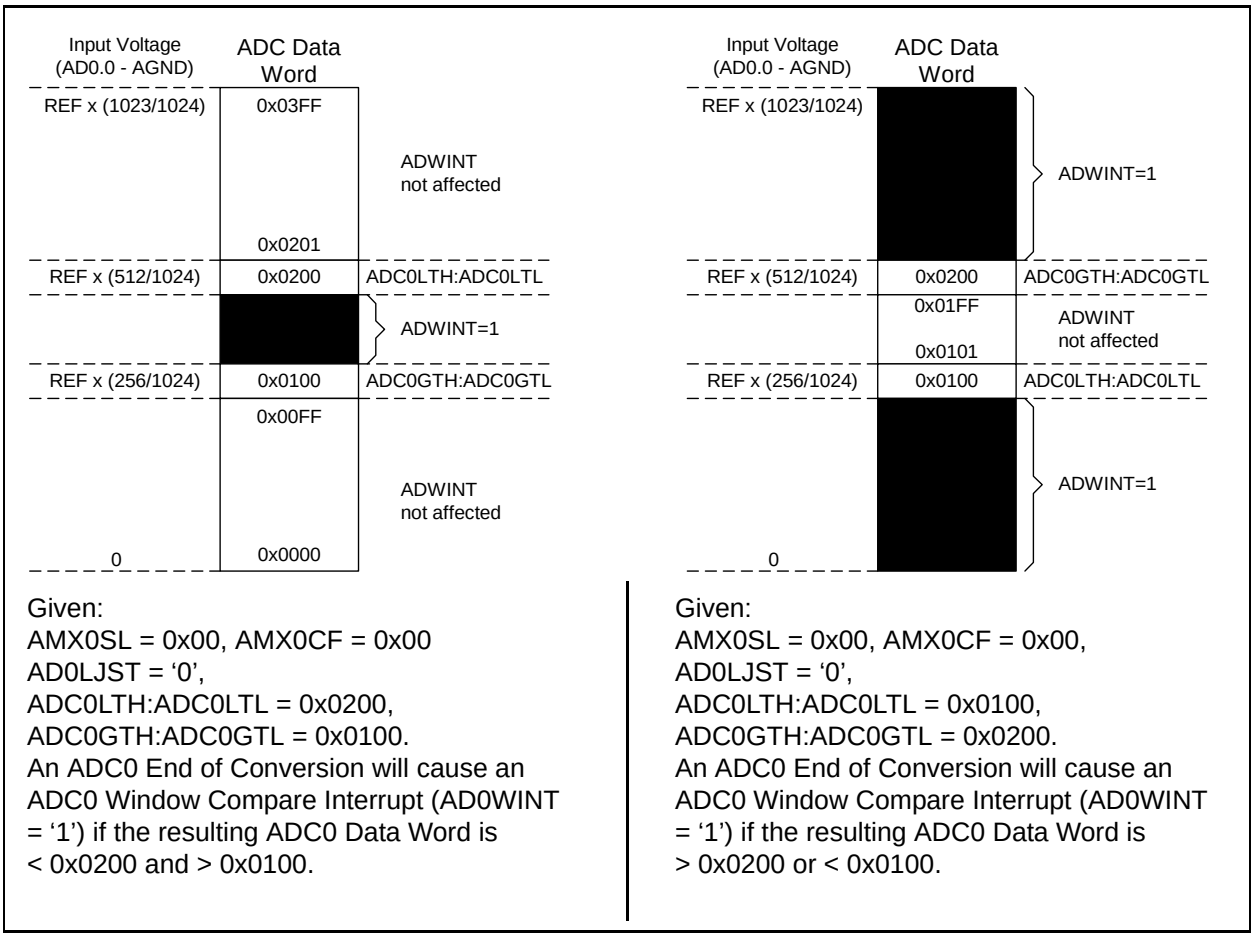

Figure 6.6. 10-Bit ADC0 Window Interrupt Example: Right Justified Single-Ended Data

SFR Definition 7.1. AMX2CF: AMUX2 Configuration

SFR Page: 2
SFR Address: 0xBA

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	PIN67IC	PIN45IC	PIN23IC	PIN01IC	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–4: UNUSED. Read = 0000b; Write = don't care.
Bit3: PIN67IC: AIN2.6, AIN2.7 Input Pair Configuration Bit.
0: AIN2.6 and AIN2.7 are independent single-ended inputs.
1: AIN2.6 and AIN2.7 are (respectively) +, – differential input pair.
Bit2: PIN45IC: AIN2.4, AIN2.5 Input Pair Configuration Bit.
0: AIN2.4 and AIN2.5 are independent single-ended inputs.
1: AIN2.4 and AIN2.5 are (respectively) +, – differential input pair.
Bit1: PIN23IC: AIN2.2, AIN2.3 Input Pair Configuration Bit.
0: AIN2.2 and AIN2.3 are independent single-ended inputs.
1: AIN2.2 and AIN2.3 are (respectively) +, – differential input pair.
Bit0: PIN01IC: AIN2.0, AIN2.1 Input Pair Configuration Bit.
0: AIN2.0 and AIN2.1 are independent single-ended inputs.
1: AIN2.0 and AIN2.1 are (respectively) +, – differential input pair.

Note: The ADC2 Data Word is in 2's complement format for channels configured as differential.

8.1.1. Update Output On-Demand

In its default mode (DAC0CN.[4:3] = ‘00’) the DAC0 output is updated “on-demand” on a write to the high-byte of the DAC0 data register (DAC0H). It is important to note that writes to DAC0L are held, and have no effect on the DAC0 output until a write to DAC0H takes place. If writing a full 12-bit word to the DAC data registers, the 12-bit data word is written to the low byte (DAC0L) and high byte (DAC0H) data registers. Data is latched into DAC0 after a write to the corresponding DAC0H register, **so the write sequence should be DAC0L followed by DAC0H** if the full 12-bit resolution is required. The DAC can be used in 8-bit mode by initializing DAC0L to the desired value (typically 0x00), and writing data to only DAC0H (also see **Section 8.2** for information on formatting the 12-bit DAC data word within the 16-bit SFR space).

8.1.2. Update Output Based on Timer Overflow

Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the DAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the DAC is used to generate a waveform of a defined sampling rate by eliminating the effects of variable interrupt latency and instruction execution on the timing of the DAC output. When the DAC0MD bits (DAC0CN.[4:3]) are set to ‘01’, ‘10’, or ‘11’, writes to both DAC data registers (DAC0L and DAC0H) are held until an associated Timer overflow event (Timer 3, Timer 4, or Timer 2, respectively) occurs, at which time the DAC0H:DAC0L contents are copied to the DAC input latches allowing the DAC output to change to the new value.

8.2. DAC Output Scaling/Justification

In some instances, input data should be shifted prior to a DAC0 write operation to properly justify data within the DAC input registers. This action would typically require one or more load and shift operations, adding software overhead and slowing DAC throughput. To alleviate this problem, the data-formatting feature provides a means for the user to program the orientation of the DAC0 data word within data registers DAC0H and DAC0L. The three DAC0DF bits (DAC0CN.[2:0]) allow the user to specify one of five data word orientations as shown in the DAC0CN register definition.

DAC1 is functionally the same as DAC0 described above. The electrical specifications for both DAC0 and DAC1 are given in Table 8.1.

NOTES:

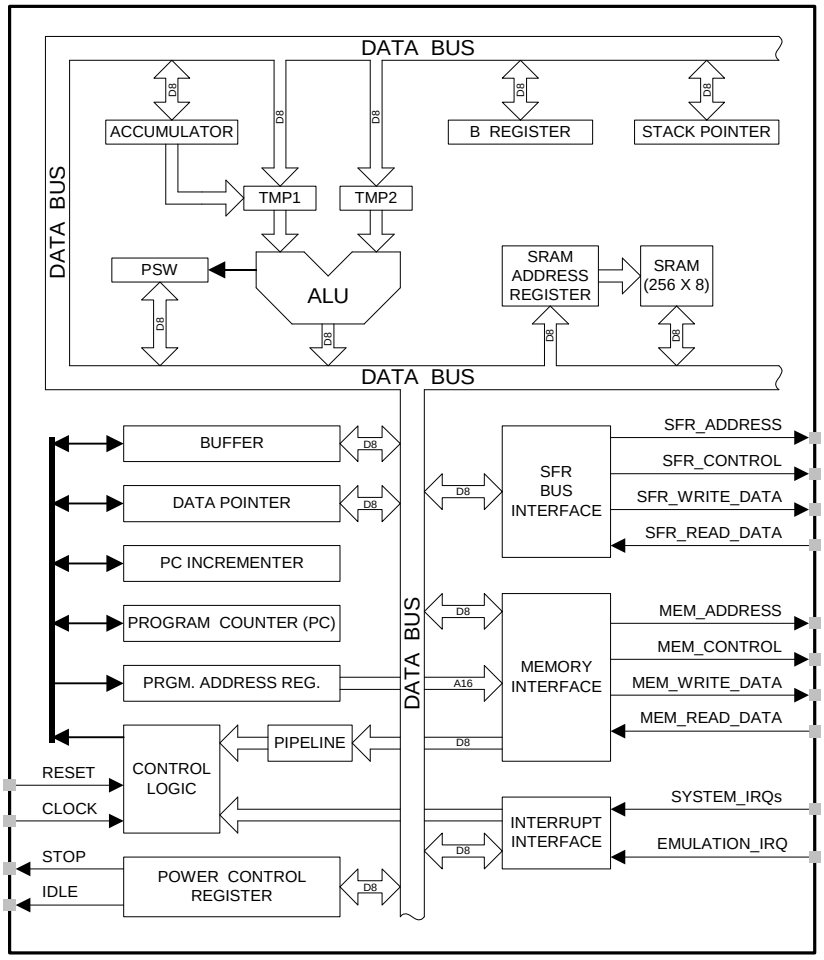

Figure 11.1. CIP-51 Block Diagram

Programming and Debugging Support

A JTAG-based serial interface is provided for in-system programming of the Flash program memory and communication with on-chip debug support logic. The re-programmable Flash can also be read and changed by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints and watch points, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debug is completely non-intrusive and non-invasive, requiring no RAM, Stack, timers, or other on-chip resources.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, macro assembler, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via its JTAG interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

Table 11.4. Interrupt Summary (Continued)

Interrupt Source	Interrupt Vector	Priority Order	Pending Flags	Bit addressable?	Cleared by HW?	SFRPAGE (SFRPGEN = 1)	Enable Flag	Priority Control
Comparator 1 Rising Edge	0x006B	13	CP1RIF (CPT1CN.5)	Y		2	ECP1R (EIE1.7)	PCP1F (EIP1.7)
Timer 3	0x0073	14	TF3 (TMR3CN.7) EXF3 (TMR3CN.6)	Y		1	ET3 (EIE2.0)	PT3 (EIP2.0)
ADC0 End of Conversion	0x007B	15	AD0INT (ADC0CN.5)	Y		0	EADC0 (EIE2.1)	PADC0 (EIP2.1)
Timer 4	0x0083	16	TF4 (TMR4CN.7) EXF4 (TMR4CN.7)	Y		2	ET4 (EIE2.2)	PT4 (EIP2.2)
ADC2 Window Comparator	0x008B	17	AD2WINT (ADC2CN.0)	Y		2	EWADC2 (EIE2.3)	PWADC2 (EIP2.3)
ADC2 End of Conversion	0x0093	18	AD2INT (ADC2CN.5)	Y		2	EADC2 (EIE2.4)	PADC2 (EIP2.4)
RESERVED	0x009B	19	N/A	N/A	N/A	N/A	N/A	N/A
UART1	0x00A3	20	RI1 (SCON1.0) TI1 (SCON1.1)	Y		1	ES1 (EIE2.6)	PS1 (EIP2.6)

11.3.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP-EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 11.4.

11.3.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. Additional clock cycles will be required if a cache miss occurs (see **Section “16. Branch Target Cache” on page 211** for more details). If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) is when the CPU is performing an RETI instruction followed by a DIV as the next instruction, and a cache miss event also occurs. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

SFR Definition 14.8. PLL0FLT: PLL Filter

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	PLLICO1	PLLICO0	PLLLP3	PLLLP2	PLLLP1	PLLLP0	00110001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8F
SFR Page: F

Bits 7–6: UNUSED: Read = 00b; Write = don't care.
Bits 5–4: PLLICO1-0: PLL Current-Controlled Oscillator Control Bits.
Selection is based on the desired output frequency, according to the following table:

PLL Output Clock	PLLICO1-0
65–100 MHz	00
45–80 MHz	01
30–60 MHz	10
25–50 MHz	11

Bits 3–0: PLLLP3-0: PLL Loop Filter Control Bits.
Selection is based on the divided PLL reference clock, according to the following table:

Divided PLL Reference Clock	PLLLP3-0
19–30 MHz	0001
12.2–19.5 MHz	0011
7.8–12.5 MHz	0111
5–8 MHz	1111

Table 14.2. PLL Frequency Characteristics

–40 to +85 °C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Input Frequency (Divided Reference Frequency)		5		30	MHz
PLL Output Frequency		25		100*	MHz
***Note:** The maximum operating frequency of the C8051F124/5/6/7 is 50 MHz					

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 61 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.5-0). 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 16.3. 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 111110b. 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 000000b. Therefore, cache locations 1 and 0 must remain unlocked at all times.**

SFR Definition 16.4. CCH0MA: Cache Miss Accumulator

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

SFR Address: 0x9A
SFR Page: F

Bit 7: CHMSOV: Cache Miss Penalty Overflow.
This bit indicates when the Cache Miss Penalty Accumulator has overflowed since it was last written.
0: The Cache Miss Penalty Accumulator has not overflowed since it was last written.
1: An overflow of the Cache Miss Penalty Accumulator has occurred since it was last written.

Bits 6–0: CHMSCTH: Cache Miss Penalty Accumulator (bits 11–5)
These are bits 11-5 of the Cache Miss Penalty Accumulator. The next four bits (bits 4-1) are stored in CHMSCTL in the CCH0TN register.
The Cache Miss Penalty Accumulator is incremented every clock cycle that the processor is delayed due to a cache miss. This is primarily used as a diagnostic feature, when optimizing code for execution speed.
Writing to CHMSCTH clears the lower 5 bits of the Cache Miss Penalty Accumulator.
Reading from CHMSCTH returns the current value of CHMSTCH, and latches bits 4-1 into CHMSTCL so that they can be read. Because bit 0 of the Cache Miss Penalty Accumulator is not available, the Cumulative Miss Penalty is equal to 2 * (CCHMSTCH:CCHMSTCL).

SFR Definition 16.5. FLSTAT: Flash Status

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	-	FLBUSY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x88
SFR Page: F

Bit 7–1: Reserved.

Bit 0: FLBUSY: Flash Busy
This bit indicates when a Flash write or erase operation is in progress.
0: Flash is idle or reading.
1: Flash write/erase operation is currently in progress.

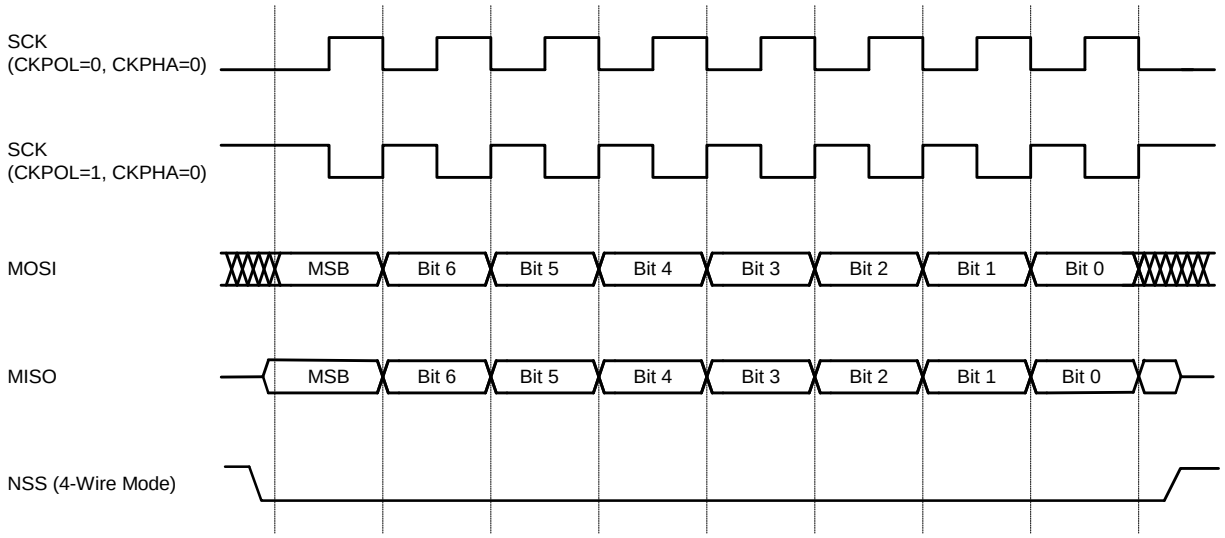

Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

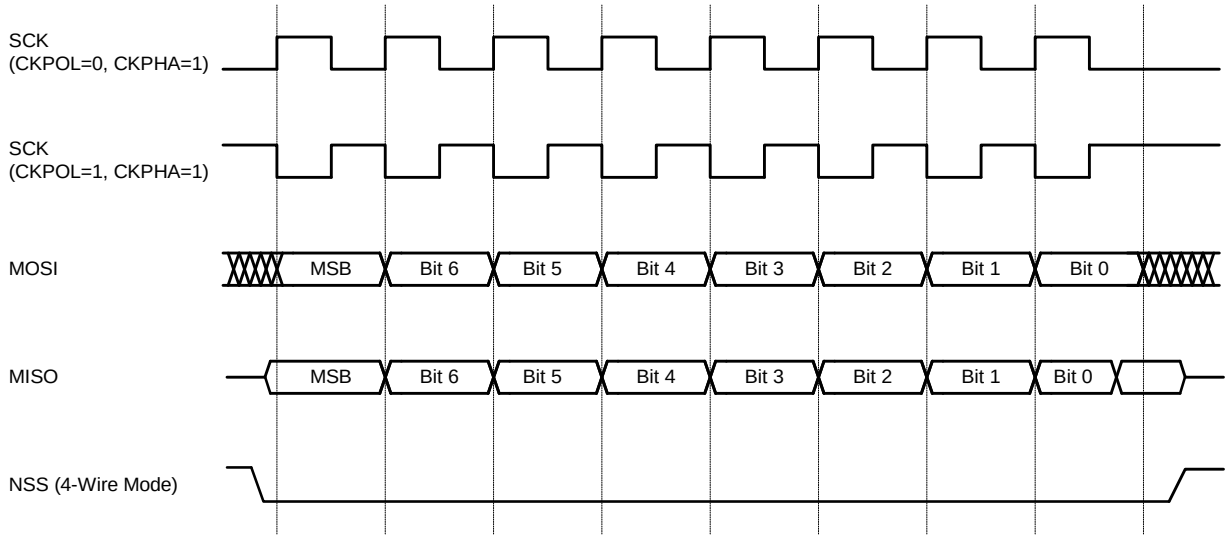

Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)

NOTES:

23.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

23.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 or Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from 0xFF to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal /INT0 is low

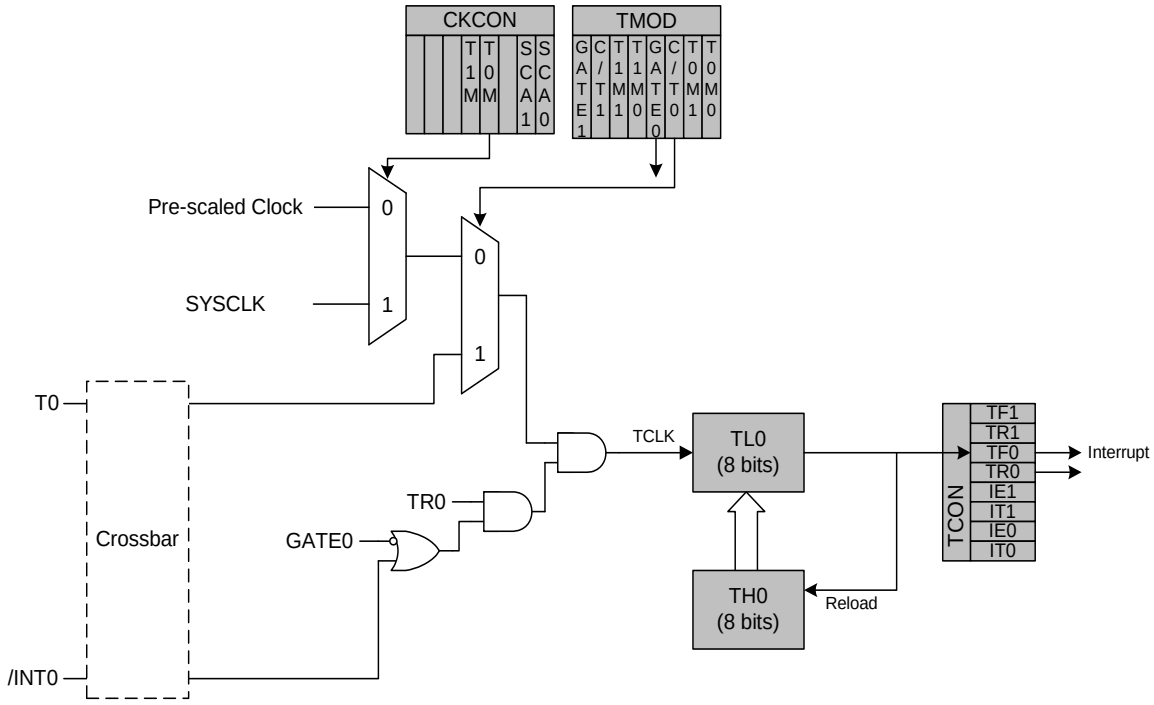

Figure 23.2. T0 Mode 2 Block Diagram

23.2.4. Toggle Output Mode (Timer 2 and Timer 4 Only)

Timers 2 and 4 have the capability to toggle the state of their respective output port pins (T2 or T4) to produce a 50% duty cycle waveform output. The port pin state will change upon the overflow or underflow of the respective timer (depending on whether the timer is counting *up* or *down*). The toggle frequency is determined by the clock source of the timer and the values loaded into RCAPnH and RCAPnL. When counting DOWN, the auto-reload value for the timer is 0xFFFF, and underflow will occur when the value in the timer matches the value stored in RCAPnH:RCAPnL. When counting UP, the auto-reload value for the timer is RCAPnH:RCAPnL, and overflow will occur when the value in the timer transitions from 0xFFFF to the reload value.

To output a square wave, the timer is placed in reload mode (the Capture/Reload Select Bit in TMRnCN and the Timer/Counter Select Bit in TMRnCN are cleared to '0'). The timer output is enabled by setting the Timer Output Enable Bit in TMRnCF to '1'. The timer should be configured via the timer clock source and reload/underflow values such that the timer overflow/underflows at 1/2 the desired output frequency. The port pin assigned by the crossbar as the timer's output pin should be configured as a digital output (see **Section "18. Port Input/Output" on page 235**). Setting the timer's Run Bit (TRn) to '1' will start the toggle of the pin. A Read/Write of the Timer's Toggle Output State Bit (TMRnCF.2) is used to read the state of the toggle output, or to force a value of the output. This is useful when it is desired to start the toggle of a pin in a known state, or to force the pin into a desired state when the toggle mode is halted.

Equation 23.1. Square Wave Frequency (Timer 2 and Timer 4 Only)

$$F_{sq} = \frac{F_{TCLK}}{2 \times (65536 - RCAPn)}$$

SFR Definition 24.7. PCA0CPHn: PCA0 Capture Module High Byte

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	

SFR Address: PCA0CPH0: 0xFC, PCA0CPH1: 0xFD, PCA0CPH2: 0xEA, PCA0CPH3: 0xEC, PCA0CPH4: 0xEE, PCA0CPH5: 0xE2

SFR Page: PCA0CPH0: page 0, PCA0CPH1: page 0, PCA0CPH2: page 0, PCA0CPH3: page 0, PCA0CPH4: page 0, PCA0CPH5: page 0

Bits7–0: PCA0CPHn: PCA0 Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

25.1. Boundary Scan

The DR in the Boundary Scan path is an 134-bit shift register. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature via the EXTEST and SAMPLE commands.

Table 25.1. Boundary Data Register Bit Definitions

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

Bit	Action	Target
0	Capture	Reset Enable from MCU (64-pin TQFP devices)
	Update	Reset Enable to $\overline{\text{RST}}$ pin (64-pin TQFP devices)
1	Capture	Reset input from $\overline{\text{RST}}$ pin (64-pin TQFP devices)
	Update	Reset output to $\overline{\text{RST}}$ pin (64-pin TQFP devices)
2	Capture	Reset Enable from MCU (100-pin TQFP devices)
	Update	Reset Enable to $\overline{\text{RST}}$ pin (100-pin TQFP devices)
3	Capture	Reset input from $\overline{\text{RST}}$ pin (100-pin TQFP devices)
	Update	Reset output to $\overline{\text{RST}}$ pin (100-pin TQFP devices)
4	Capture	External Clock from XTAL1 pin
	Update	Not used
5	Capture	Weak pullup enable from MCU
	Update	Weak pullup enable to Port Pins
6, 8, 10, 12, 14, 16, 18, 20	Capture	P0.n output enable from MCU (e.g. Bit6=P0.0, Bit8=P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit6=P0.0oe, Bit8=P0.1oe, etc.)
7, 9, 11, 13, 15, 17, 19, 21	Capture	P0.n input from pin (e.g. Bit7=P0.0, Bit9=P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit7=P0.0, Bit9=P0.1, etc.)
22, 24, 26, 28, 30, 32, 34, 36	Capture	P1.n output enable from MCU
	Update	P1.n output enable to pin
23, 25, 27, 29, 31, 33, 35, 37	Capture	P1.n input from pin
	Update	P1.n output to pin
38, 40, 42, 44, 46, 48, 50, 52	Capture	P2.n output enable from MCU
	Update	P2.n output enable to pin
39, 41, 43, 45, 47, 49, 51, 53	Capture	P2.n input from pin
	Update	P2.n output to pin
54, 56, 58, 60, 62, 64, 66, 68	Capture	P3.n output enable from MCU
	Update	P3.n output enable to pin
55, 57, 59, 61, 63, 65, 67, 69	Capture	P3.n input from pin
	Update	P3.n output to pin
70, 72, 74, 76, 78, 80, 82, 84	Capture	P4.n output enable from MCU
	Update	P4.n output enable to pin
71, 73, 75, 77, 79, 81, 83, 85	Capture	P4.n input from pin
	Update	P4.n output to pin
86, 88, 90, 92, 94, 96, 98, 100	Capture	P5.n output enable from MCU
	Update	P5.n output enable to pin
87, 89, 91, 93, 95, 97, 99, 101	Capture	P5.n input from pin
	Update	P5.n output to pin
102, 104, 106, 108, 110, 112, 114, 116	Capture	P6.n output enable from MCU
	Update	P6.n output enable to pin

JTAG Register Definition 25.3. FLASHCON: JTAG Flash Control

								Reset Value
SFLE	WRMD2	WRMD1	WRMD0	RDMD3	RDMD2	RDMD1	RDMD0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

This register determines how the Flash interface logic will respond to reads and writes to the FLASHDAT Register.

Bit7: SFLE: Scratchpad Flash Memory Access Enable
When this bit is set, Flash reads and writes are directed to the two 128-byte Scratchpad Flash sectors. When SFLE is set to logic 1, Flash accesses out of the address range 0x00-0xFF should not be attempted (with the exception of address 0x400, which can be used to simultaneously erase both Scratchpad areas). Reads/Writes out of this range will yield undefined results.
0: Flash access directed to the Program/Data Flash sector.
1: Flash access directed to the two 128 byte Scratchpad sectors.

Bits6–4: WRMD2–0: Write Mode Select Bits.
The Write Mode Select Bits control how the interface logic responds to writes to the FLASHDAT Register per the following values:
000: A FLASHDAT write replaces the data in the FLASHDAT register, but is otherwise ignored.
001: A FLASHDAT write initiates a write of FLASHDAT into the memory address by the FLASHADR register. FLASHADR is incremented by one when complete.
010: A FLASHDAT write initiates an erasure (sets all bytes to 0xFF) of the Flash page containing the address in FLASHADR. The data written must be 0xA5 for the erase to occur. FLASHADR is not affected. If FLASHADR = 0x1FBFE – 0x1FBFF, the entire user space will be erased (i.e. entire Flash memory except for Reserved area 0x1FC00 – 0x1FFFF).
(All other values for WRMD2-0 are reserved.)

Bits3–0: RDMD3–0: Read Mode Select Bits.
The Read Mode Select Bits control how the interface logic responds to reads from the FLASHDAT Register per the following values:
0000: A FLASHDAT read provides the data in the FLASHDAT register, but is otherwise ignored.
0001: A FLASHDAT read initiates a read of the byte addressed by the FLASHADR register if no operation is currently active. This mode is used for block reads.
0010: A FLASHDAT read initiates a read of the byte addressed by FLASHADR only if no operation is active and any data from a previous read has already been read from FLASHDAT. This mode allows single bytes to be read (or the last byte of a block) without initiating an extra read.
(All other values for RDMD3–0 are reserved.)