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	POR, PWM, WDT
Number of I/O	39
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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f363-c-gqr

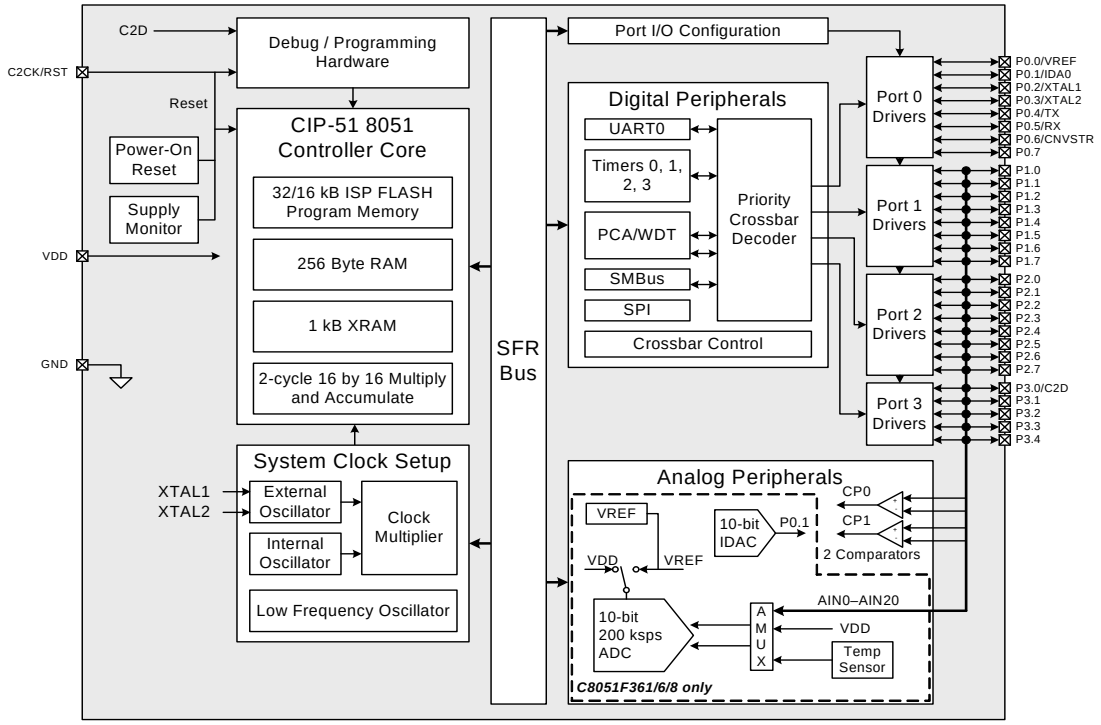

Figure 1.2. C8051F361/4/6/8 Block Diagram

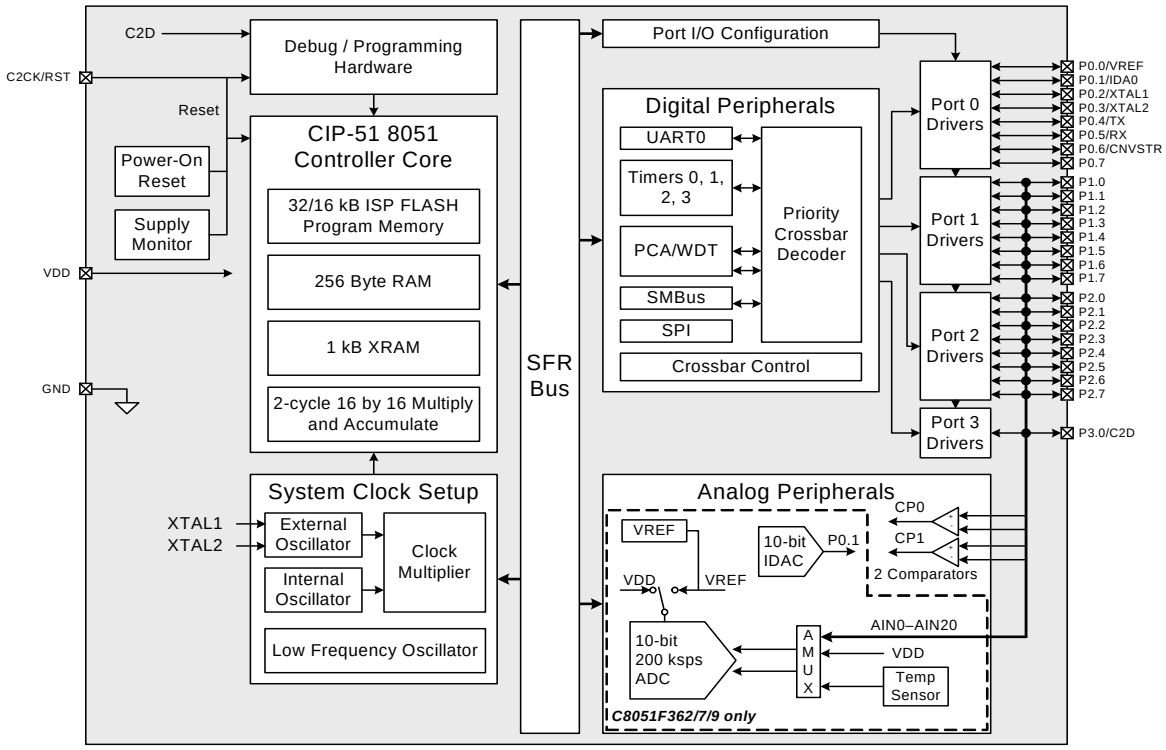

Figure 1.3. C8051F362/5/7/9 Block Diagram

cated by a status bit and an interrupt (if enabled). The resulting 10-bit data word is latched into the ADC data SFRs upon completion of a conversion.

Window compare registers for the ADC data can be configured to interrupt the controller when ADC data is either within or outside of a specified range. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within/outside the specified range.

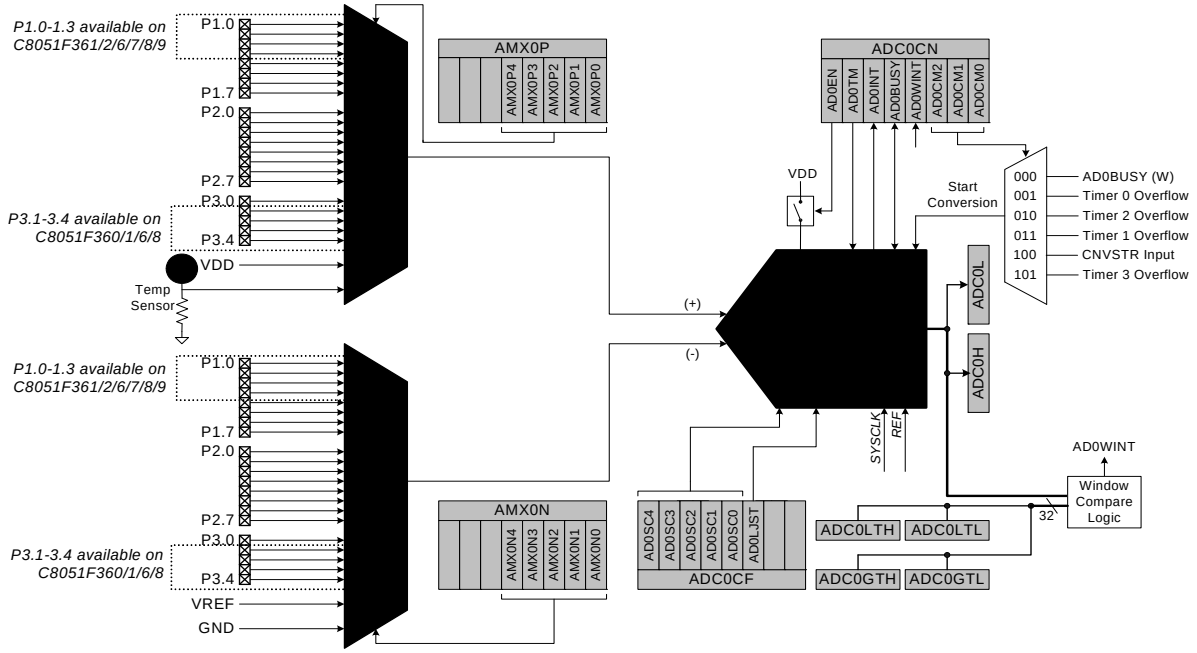

Figure 1.11. 10-Bit ADC Block Diagram

1.8. Comparators

C8051F36x devices include two on-chip voltage comparators that are enabled/disabled and configured via user software. Port I/O pins may be configured as comparator inputs via a selection mux. Two comparator outputs may be routed to a Port pin if desired: a latched output and/or an unlatched (asynchronous) output. Comparator response time is programmable, allowing the user to select between high-speed and low-power modes. Positive and negative hysteresis are also configurable.

Comparator interrupts may be generated on rising, falling, or both edges. When in IDLE mode, these interrupts may be used as a “wake-up” source. Comparator0 may also be configured as a reset source. Figure 1.12 shows the Comparator0 block diagram, and Figure 1.13 shows the Comparator1 block diagram.

Note: The first Port I/O pins shown in Figure 1.12 and Figure 1.13 are for the 48-pin (C8051F360/3) devices. The second set of Port I/O pins are for the 32-pin and 28-pin (C8051F361/2/4/5/6/7/8/9) devices. Please refer to the CPTnMX registers (SFR Definition 8.2 and SFR Definition 8.5) for more information.

4. Pinout and Package Definitions

Table 4.1. Pin Definitions for the C8051F36x

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
V_{DD}	19, 31, 43	4	4		Power Supply Voltage.
GND	18, 30, 42	3	3		Ground.
AGND	6	—	—		Analog Ground.
AV+	7	—	—		Analog Supply Voltage. Must be tied to +2.7 to +3.6 V.
$\overline{RST}$/	8	5	5	D I/O	Device Reset. Open-drain output of internal POR or V_{DD} Monitor. An external source can initiate a system reset by driving this pin low for at least 10 µs.
C2CK				D I/O	Clock signal for the C2 Debug Interface.
P4.6/	9	—	—	D I/O or A In	Port 4.6. See Section 17 for a complete description.
C2D				D I/O	Bi-directional data signal for the C2 Debug Interface.
P3.0/	—	6	6	D I/O or A In	Port 3.0. See Section 17 for a complete description.
C2D				D I/O	Bi-directional data signal for the C2 Debug Interface.
P0.0	5	2	2	D I/O or A In	Port 0.0. See Section 17 for a complete description.
P0.1	4	1	1	D I/O or A In	Port 0.1. See Section 17 for a complete description.
P0.2	3	32	28	D I/O or A In	Port 0.2. See Section 17 for a complete description.
P0.3	2	31	27	D I/O or A In	Port 0.3. See Section 17 for a complete description.
P0.4	1	30	26	D I/O or A In	Port 0.4. See Section 17 for a complete description.
P0.5	48	29	25	D I/O or A In	Port 0.5. See Section 17 for a complete description.
P0.6	47	28	24	D I/O or A In	Port 0.6. See Section 17 for a complete description.
P0.7	46	27	23	D I/O or A In	Port 0.7. See Section 17 for a complete description.

5.1. Analog Multiplexer

AMUX0 selects the positive and negative inputs to the ADC. Any of the following may be selected as the positive input: the AMUX0 Port I/O inputs, the on-chip temperature sensor, or the positive power supply (V_{DD}). Any of the following may be selected as the negative input: the AMUX0 Port I/O inputs, VREF, or GND. **When GND is selected as the negative input, ADC0 operates in Single-ended Mode; all other times, ADC0 operates in Differential Mode.** The ADC0 input channels are selected in the AMX0P and AMX0N registers as described in SFR Definition 5.1 and SFR Definition 5.2.

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

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

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

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 511/512	0x01FF	0x7FC0
VREF x 256/512	0x0100	0x4000
0	0x0000	0x0000
–VREF x 256/512	0xFF00	0xC000
–VREF	0xFE00	0x8000

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to ‘0’ the corresponding bit in register PnMDIN (for n = 0,1,2,3). To force the Crossbar to skip a Port pin, set to ‘1’ the corresponding bit in register PnSKIP (for n = 0,1,2,3). See Section “17. Port Input/Output” on page 182 for more Port I/O configuration details.

SFR Definition 6.3. IDA0L: IDA0 Data Word LSB

SFR Page: all pages
SFR Address: 0x96

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

Bits 7–6: IDA0 Data Word Low-Order Bits.
Lower 2 bits of the 10-bit Data Word.
Bits 5–0: UNUSED. Read = 000000b, Write = don't care.

Table 6.1. IDAC Electrical Characteristics

–40 to +85 °C, V_{DD} = 3.0 V Full-scale output current set to 2 mA unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Static Performance					
Resolution			10		bits
Integral Nonlinearity		—	±0.5	±2	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	±0.5	±1	LSB
Output Compliance Range		—	—	V_{DD} – 1.2	V
Offset Error		—	0	—	LSB
Full Scale Error	2 mA Full Scale Output Current	–15	0	15	LSB
Full Scale Error Tempco		—	30	—	ppm/°C
V_{DD} Power Supply Rejection Ratio		—	6.5	—	µA/V
Dynamic Performance					
Output Settling Time to 1/2 LSB	IDA0H:L = 0x3FF to 0x000	—	5	—	µs
Startup Time		—	5	—	µs
Gain Variation	1 mA Full Scale Output Current 0.5 mA Full Scale Output Current	— —	±1 ±1	— —	% %
Power Consumption					
Power Supply Current (V_{DD} supplied to IDAC)	2 mA Full Scale Output Current 1 mA Full Scale Output Current 0.5 mA Full Scale Output Current	— — —	2140 1140 640	— — —	µA µA µA

9.4.2. Data Memory

The CIP-51 implements 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFR's. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 9.2 illustrates the data memory organization of the CIP-51.

9.4.3. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 9.8). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

9.4.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination). The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte.

For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

9.4.5. Stack

A programmer's stack can be located anywhere in the 256 byte data memory. The stack area is designated using the Stack Pointer (SP, address 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07; therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

The MCUs also have built-in hardware for a stack record which is accessed by the debug logic. The stack record is a 32-bit shift register, where each PUSH or increment SP pushes one record bit onto the register,

SFR Definition 9.3. SFRNEXT: SFR Next Register

SFR Page: all pages
SFR Address: 0x85

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 Stack Bits: SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to 'push' or 'pop'. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt.

Read: Returns the value of the SFR page contained in the second byte of the SFR stack. This is the value that will go to the SFR Page register upon a return from interrupt.

SFR Definition 9.4. SFRLAST: SFR Last Register

SFR Page: all pages
SFR Address: 0x86

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 Stack Bits: SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to 'push' or 'pop'. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Write: Sets the SFR Page in the last entry of the SFR Stack. This will cause the SFRNEXT SFR to have this SFR page value upon a return from interrupt.

Read: Returns the value of the SFR page contained in the last entry of the SFR stack.

SFR Definition 11.2. MAC0STA: MAC0 Status

SFR Page: 0
SFR Address: 0xCF

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	MAC0HO	MAC0Z	MAC0SO	MAC0N	00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

Bits 7–4: UNUSED: Read = 0000b, Write = don't care.

Bit 3: MAC0HO: Hard Overflow Flag.
This bit is set to '1' whenever an overflow out of the MAC0OVR register occurs during a MAC operation (i.e. when MAC0OVR changes from 0x7F to 0x80 or from 0x80 to 0x7F). The hard overflow flag must be cleared in software by directly writing it to '0', or by resetting the MAC logic using the MAC0CA bit in register MAC0CF.

Bit 2: MAC0Z: Zero Flag.
This bit is set to '1' if a MAC0 operation results in an Accumulator value of zero. If the result is non-zero, this bit will be cleared to '0'.

Bit 1: MAC0SO: Soft Overflow Flag.
This bit is set to '1' when a MAC operation causes an overflow into the sign bit (bit 31) of the MAC0 Accumulator. If the overflow condition is corrected after a subsequent MAC operation, this bit is cleared to '0'.

Bit 0: MAC0N: Negative Flag.
If the MAC Accumulator result is negative, this bit will be set to '1'. If the result is positive or zero, this flag will be cleared to '0'.

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

SFR Definition 11.3. MAC0AH: MAC0 A High Byte

SFR Page: 0
SFR Address: 0xA5

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

Bits 7–0: High Byte (bits 15–8) of MAC0 A Register.

Steps 3–8 must be repeated for each byte to be written

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. When writing to addresses located in any of the four code banks, a Flash write block is four bytes long, from addresses ending in 00b to addresses ending in 11b. Writes must be performed sequentially (i.e. addresses ending in 00b, 01b, 10b, and 11b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 11b. The Flash write will be performed following the MOVX write that targets the address ending in 1b. If any bytes in the block do not need to be updated in Flash, they should be written to 0xFF. The recommended procedure for writing Flash in blocks is as follows:

Step 1. Disable interrupts.
Step 2. Set CHBLKW (register CCH0CN) to select block write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Using the MOVX instruction, write the first data byte to the first block location (ending in 00b).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Write the first key code to FLKEY: 0xA5.
Step 10. Write the second key code to FLKEY: 0xF1.
Step 11. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 12. Clear the PSEE bit (register PSCTL).
Step 13. Using the MOVX instruction, write the second data byte to the second block location (ending in 01b).
Step 14. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 15. Write the first key code to FLKEY: 0xA5.
Step 16. Write the second key code to FLKEY: 0xF1.
Step 17. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 18. Clear the PSEE bit (register PSCTL).
Step 19. Using the MOVX instruction, write the third data byte to the third block location (ending in 10b).
Step 20. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 21. Write the first key code to FLKEY: 0xA5.
Step 22. Write the second key code to FLKEY: 0xF1.
Step 23. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 24. Clear the PSEE bit (register PSCTL).
Step 25. Using the MOVX instruction, write the fourth data byte to the last block location (ending in 11b).
Step 26. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 27. Re-enable interrupts.

Steps 3-26 must be repeated for each block to be written.

13.1.4. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written and erased using the MOVX write instruction (as described in Section 13.1.2 and Section 13.1.3) and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

SFR Definition 13.1. PSCTL: Program Store Read/Write Control

SFR Page: 0
SFR Address: 0x8F

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

Bits 7–2: UNUSED. Read = 000000b, Write = don't care.

Bit 1: PSEE: Program Store Erase Enable.
Setting this bit allows an entire page of the Flash program memory to be erased provided the PSWE bit is also set. After setting this bit, a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter. **Note: The Flash page containing the Read Lock Byte and Write/Erase Lock Byte cannot be erased by software.**
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit 0: PSWE: Program Store Write Enable.
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The location must be erased prior to writing data.
0: Write to Flash program memory disabled. MOVX write operations target External RAM.
1: Write to Flash program memory enabled. MOVX write operations target Flash memory.

SFR Definition 13.2. FLKEY: Flash Lock and Key

SFR Page: 0
SFR Address: 0xB7

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

Table 15.2. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20		ns
T_{RDH}	Read Data Hold Time	0		ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

18.5.3. Slave Receiver Mode

Serial data is received on SDA and the clock is received on SCL. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. Upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. Software responds to the received slave address with an ACK, or ignores the received slave address with a NACK. If the received slave address is ignored, slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received. Software must write the ACK bit after each received byte to ACK or NACK the received byte. The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 18.7 shows a typical Slave Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur **before** the ACK cycle in this mode.

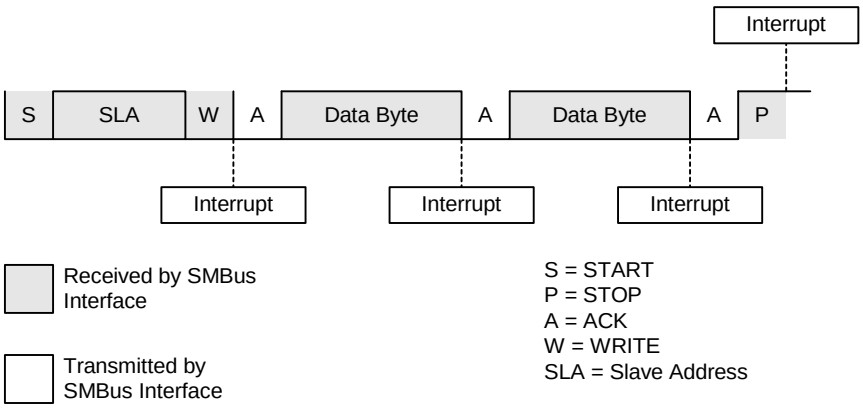

Figure 18.7. Typical Slave Receiver Sequence

18.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. In the table below, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed but do not conform to the SMBus specification.

Table 18.4. SMBus Status Decoding

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X
						Abort transfer.	0	1	X
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X
						End transfer with STOP.	0	1	X
						End transfer with STOP and start another transfer.	1	1	X
						Send repeated START.	1	0	X
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X

Table 19.2. Timer Settings for Standard Baud Rates Using an External 25.0 MHz Oscillator

	Frequency: 25.0 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 and Timer Clock from External Osc.	230400	–0.47%	108	SYSCLK	XX[2]	1	0xCA
	115200	0.45%	218	SYSCLK	XX	1	0x93
	57600	–0.01%	434	SYSCLK	XX	1	0x27
	28800	0.45%	872	SYSCLK/4	01	0	0x93
	14400	–0.01%	1736	SYSCLK/4	01	0	0x27
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D
	2400	0.45%	10464	SYSCLK/48	10	0	0x93
	1200	–0.01%	20832	SYSCLK/48	10	0	0x27
SYSCLK from Internal Osc., Timer Clock from External Osc.	57600	–0.47%	432	EXTCLK/8	11	0	0xE5
	28800	–0.47%	864	EXTCLK/8	11	0	0xCA
	14400	0.45%	1744	EXTCLK/8	11	0	0x93
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D

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

20.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 20.5. For slave mode, the clock and data relationships are shown in Figure 20.6 and Figure 20.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, C8051F33x, and C8051F36x.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 20.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

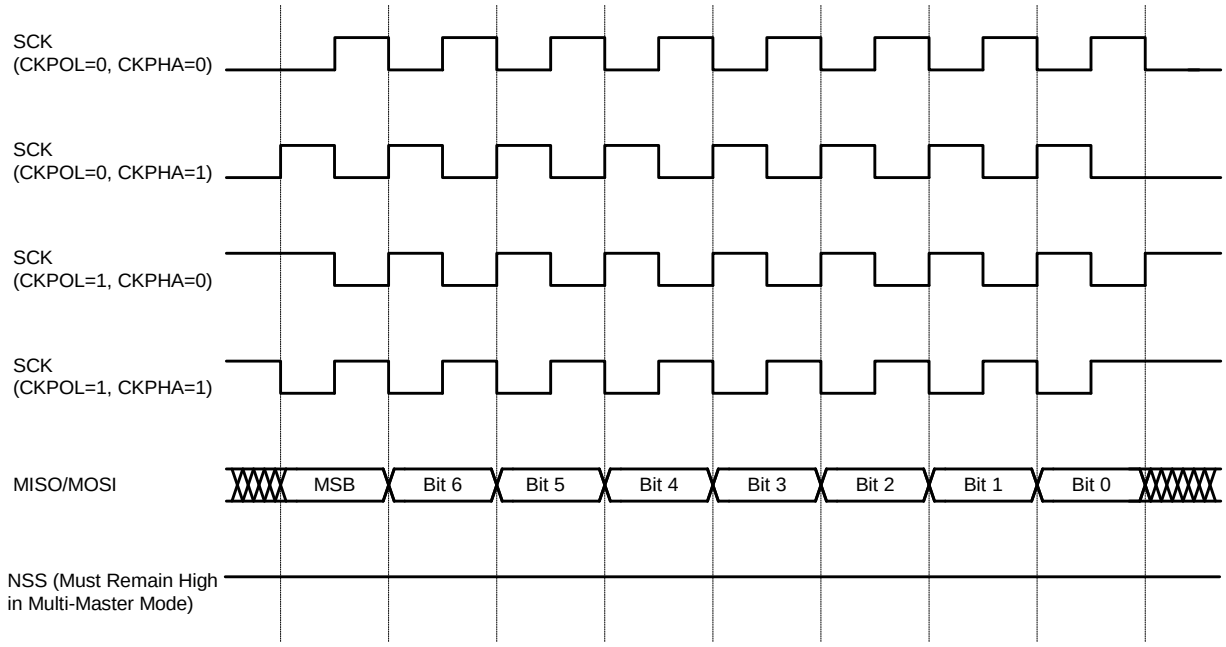

Figure 20.5. Master Mode Data/Clock Timing

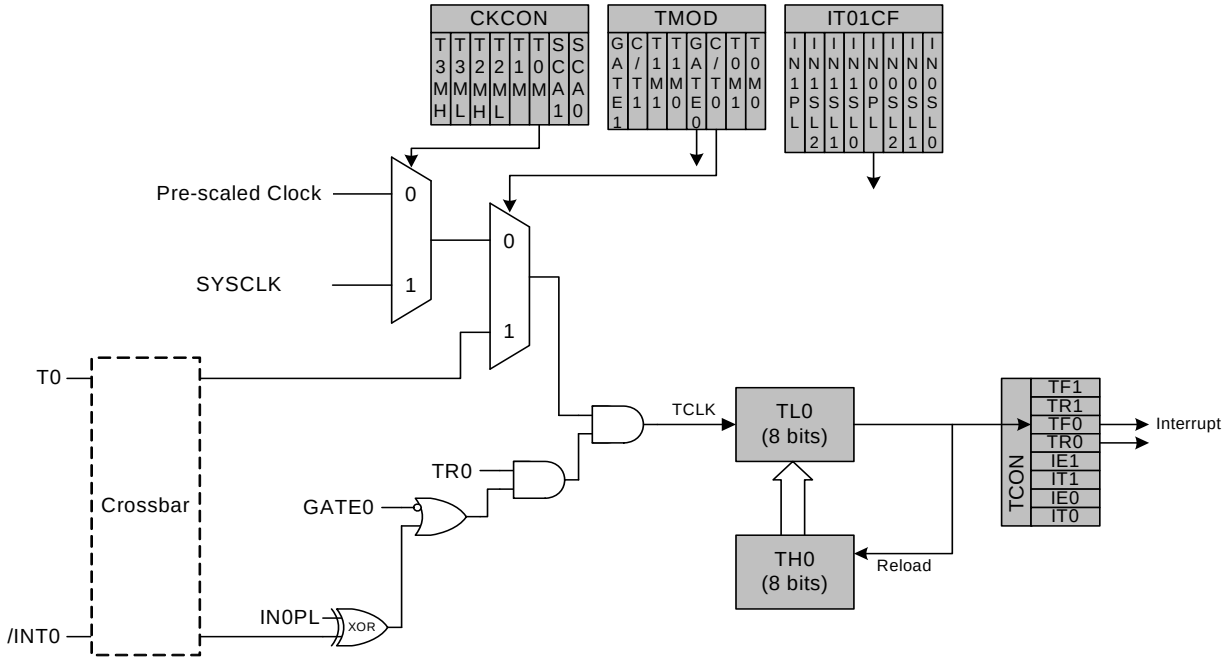

Figure 21.2. T0 Mode 2 Block Diagram

SFR Definition 21.8. TMR2CN: Timer 2 Control

SFR Page: all pages
SFR Address: 0xC8 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	–	T2XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: TF2H: Timer 2 High Byte Overflow Flag.
Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. TF2H is not automatically cleared by hardware and must be cleared by software.

Bit 6: TF2L: Timer 2 Low Byte Overflow Flag.
Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF2LEN is set and Timer 2 interrupts are enabled. TF2L will set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.

Bit 5: TF2LEN: Timer 2 Low Byte Interrupt Enable.
This bit enables/disables Timer 2 Low Byte interrupts. If TF2LEN is set and Timer 2 interrupts are enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
0: Timer 2 Low Byte interrupts disabled.
1: Timer 2 Low Byte interrupts enabled.

Bit 4: TF2CEN: Timer 2 Low-Frequency Oscillator Capture Enable.
This bit enables/disables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL. See Section “16. Oscillators” on page 168 for more details.
0: Timer 2 Low-Frequency Oscillator Capture disabled.
1: Timer 2 Low-Frequency Oscillator Capture enabled.

Bit 3: T2SPLIT: Timer 2 Split Mode Enable.
When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload.
0: Timer 2 operates in 16-bit auto-reload mode.
1: Timer 2 operates as two 8-bit auto-reload timers.

Bit 2: TR2: Timer 2 Run Control.
This bit enables/disables Timer 2. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in this mode.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: T2XCLK: Timer 2 External Clock Select.
This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.
0: Timer 2 external clock selection is the system clock divided by 12.
1: Timer 2 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

Table 22.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
18,432,000	255	42.7
18,432,000	128	21.5
18,432,000	32	5.5
11,059,200	255	71.1
11,059,200	128	35.8
11,059,200	32	9.2
3,062,500[2]	255	257
3,062,500[2]	128	129.5
3,062,500[2]	32	33.1
191,406	255	4109
191,406	128	2070
191,406	32	530
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:

1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal oscillator reset frequency.