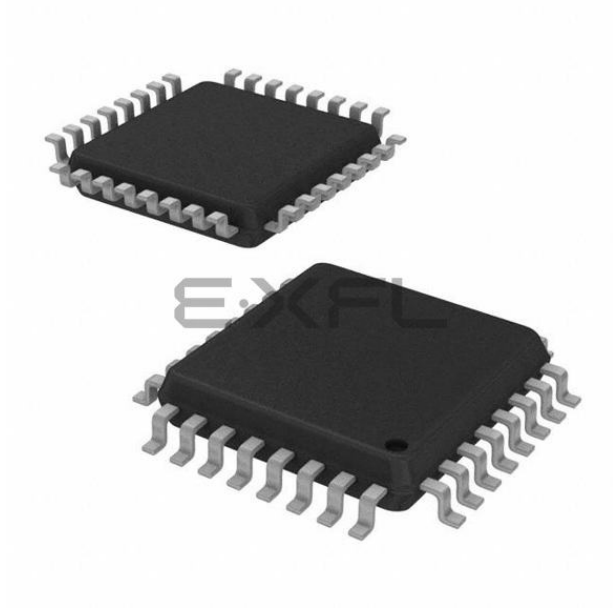

Welcome to E-XFL.COM

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	25MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	22
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
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/c8051f231-gq

NOTES:

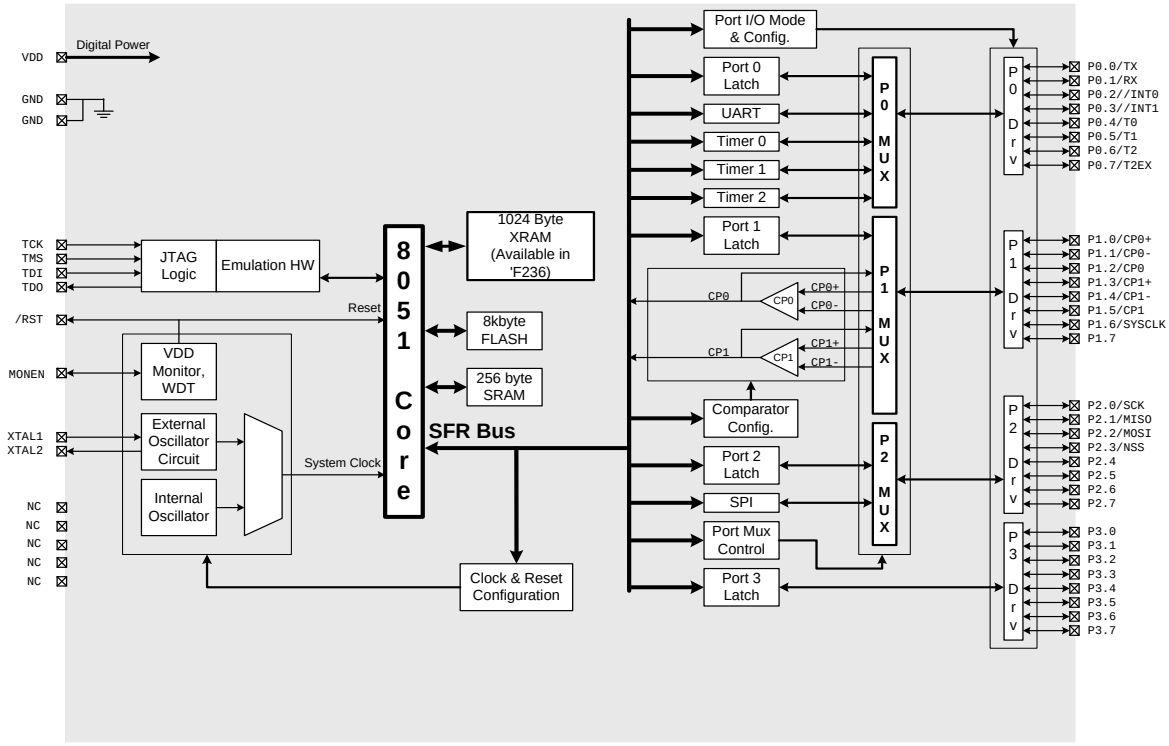

Figure 1.3. C8051F230 and C8051F236 Block Diagram (48 TQFP)

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. Figure 1.5 shows a comparison of peak throughputs of various 8-bit microcontroller cores with their maximum system clocks.

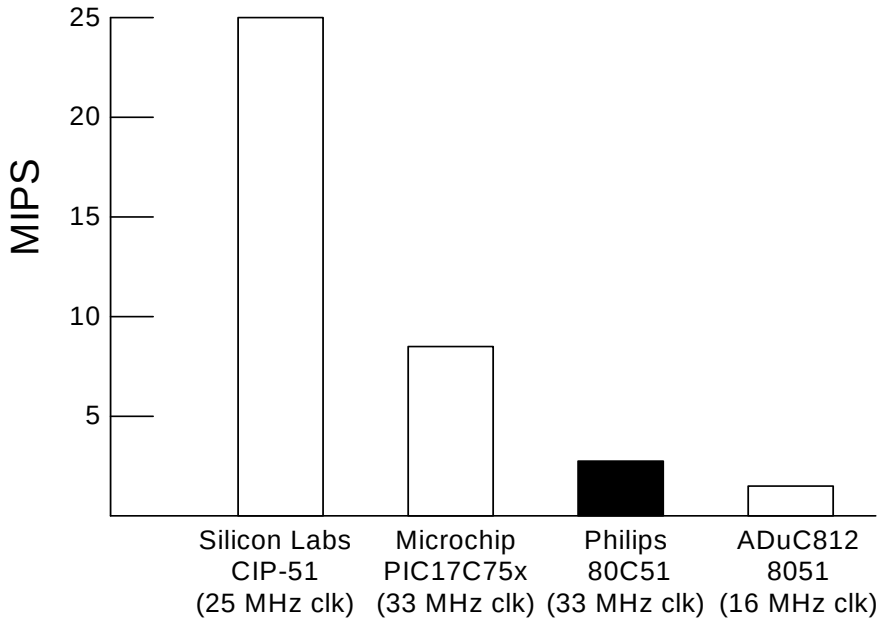

Figure 1.5. Comparison of Peak MCU Throughputs

1.1.3. Additional Features

The C8051F206, C8051F220/1/6 and C8051F230/1/6 have several key enhancements both inside and outside the CIP-51 core to improve overall performance and ease of use in end applications.

The extended interrupt handler provides 22 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing the numerous analog and digital peripherals to interrupt the controller. (An interrupt driven system requires less intervention by the MCU, giving it more effective throughput.) The extra interrupt sources are very useful when building multi-tasking, real-time systems.

There are up to six reset sources for the MCU: an on-board V_{DD} monitor, a Watchdog Timer, a missing clock detector, a voltage level detection from Comparator 0, a forced software reset, and an external reset pin. The $\overline{\text{RST}}$ pin is bi-directional, accommodating an external reset, or allowing the internally generated reset to be output on the $\overline{\text{RST}}$ pin. The on-board V_{DD} monitor is enabled by pulling the MONEN pin high (digital 1). The user may disable each reset source except for the V_{DD} monitor and Reset Input Pin from software. The watchdog timer may be permanently enabled in software after a power-on reset during MCU initialization.

The MCU has an internal, stand-alone clock generator that is used by default as the system clock after reset. If desired, the clock source may be switched "on the fly" to the external oscillator, which can use a crystal, ceramic resonator, capacitor, RC, or external clock source to generate the system clock. This can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) external crystal source, while periodically switching to the fast (up to 16MHz) internal oscillator as needed.

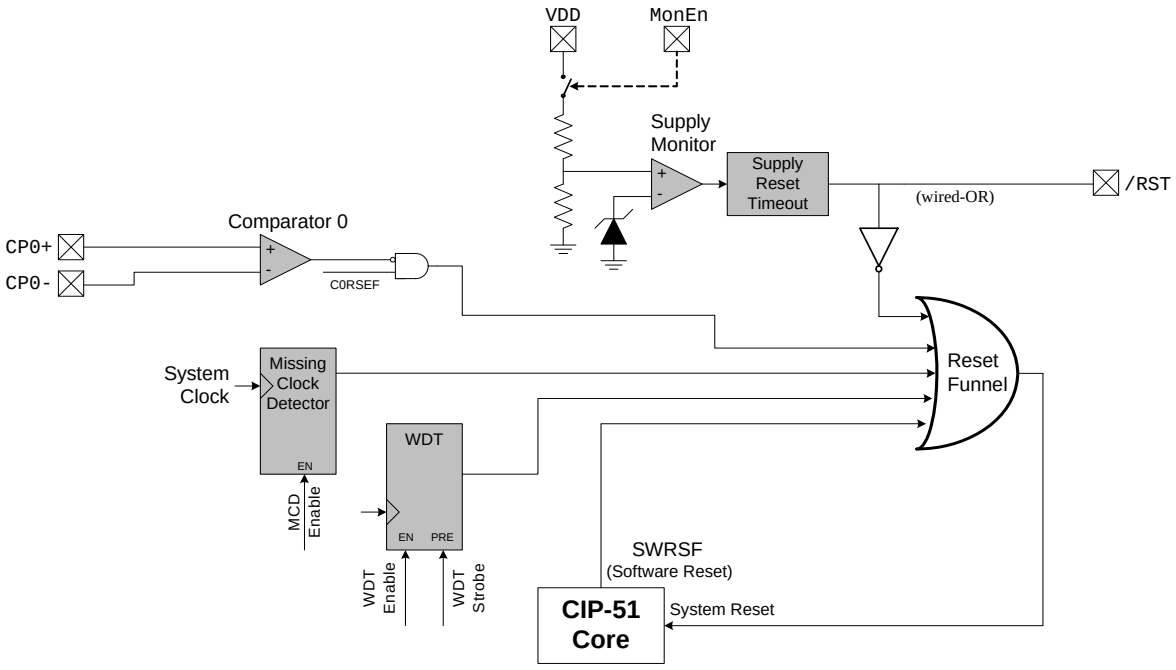

Figure 1.6. Comparison of Peak MCU Throughputs

1.2. On-Board Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. An optional 1024 bytes of XRAM is available on the 'F206, 'F226 and 'F236. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128-byte SFR address space. The lower 128 bytes of RAM are accessible via direct or indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

The MCU's program memory consists of 8 k + 128 bytes of Flash. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. The 512 bytes from addresses 0x1E00 to 0x1FFF are reserved for factory use. There is also a user programmable 128-byte sector at address 0x2000 to 0x207F, which may be useful as a table for storing software constants, nonvolatile configuration information, or as additional program space. See Figure 1.7 for the MCU system memory map.

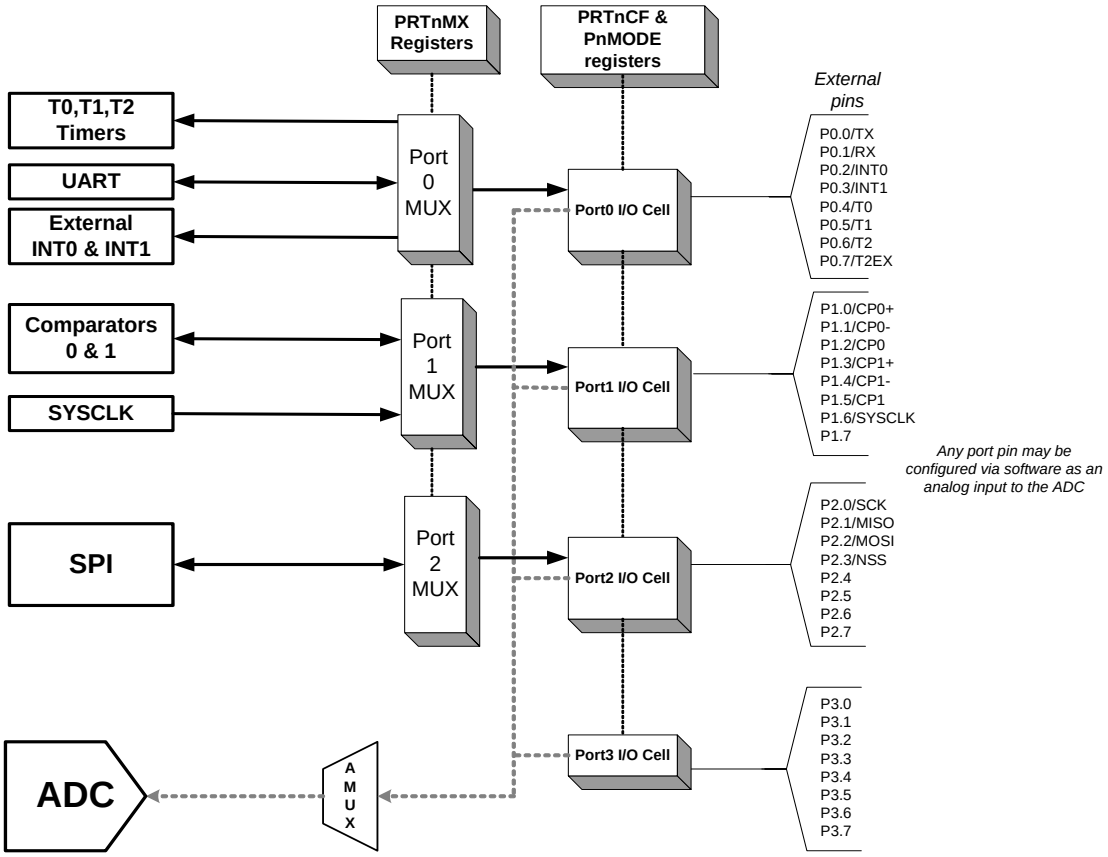

Figure 1.9. Port I/O Functional Block Diagram

1.5. Serial Ports

The C8051F206, C8051F220/1/6 and C8051F230/1/6 include a Full-Duplex UART and SPI Bus. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little intervention by the CPU. The serial buses do not have to "share" resources such as timers, interrupts, or Port I/O, so both of the serial buses may be used simultaneously. (You may use Timer1, Timer 2, or SYSCLK to generate baud rates for UART).

1.6. Analog to Digital Converter

The C8051F220/1/6 has an on-chip 8-bit SAR ADC and the C8051F206 has a 12-bit SAR ADC with a programmable gain amplifier. With a maximum throughput of 100ksps, the ADC offers true 8-bit with an INL of ±1/4 LSB, and or 12-bit accuracy with ±2 LSB. The voltage reference can be the power supply (V_{DD}), or an external reference voltage (VREF). Also, the system controller can place the ADC into a power-saving shutdown mode when not in use. A programmable gain amplifier follows the analog multiplexer. The gain can be set in software from 0.5 to 16 in powers of 2.

Conversions can be initiated in two ways; a software command or an overflow on Timer 2. This flexibility allows the start of conversion to be triggered by software events, or convert continuously. A completed conversion causes an interrupt, or a status bit can be polled in software to determine the end of conversion. The resulting 8-bit data word is latched into an SFR upon completion of a conversion.

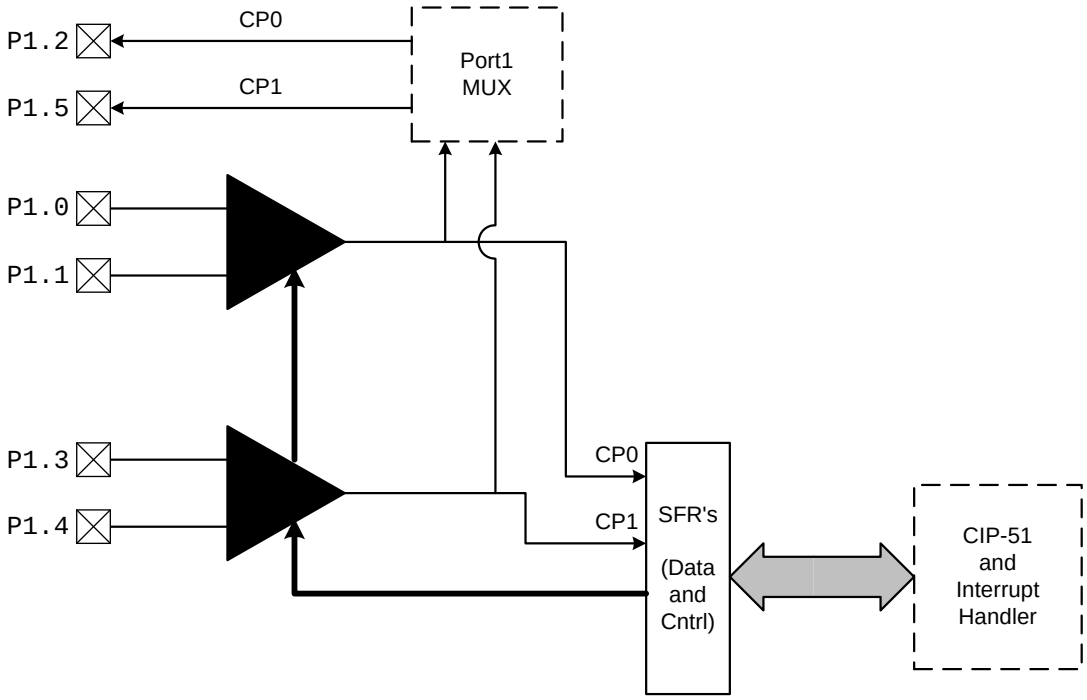

Figure 1.11. Comparator Diagram

3. Global DC Electrical Characteristics

Table 3.1. Global DC Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Analog Supply Voltage[1]		2.7	3.0	3.6	V
V_{DD} supply current with ADC and comparators active, and CPU active	Clock = 25 MHz Clock = 1 MHz Clock = 32 kHz	—	13 1.5 300	—	mA mA µA
V_{DD} supply current with ADC and comparators active, and CPU inactive (Idle Mode)	Clock = 25 MHz Clock = 1 MHz Clock = 32 kHz	—	9 1.8 275	—	mA mA µA
V_{DD} supply current with ADC and comparators inactive, and CPU active	Clock = 25 MHz Clock = 1 MHz Clock = 32 kHz	—	12.5 1.0 25	—	mA mA µA
Digital Supply Current with CPU inactive (Idle Mode)	Clock = 25 MHz Clock = 1 MHz Clock = 32 kHz	—	8.5 1.4 25	—	mA mA µA
Digital Supply Current (Stop Mode), V_{DD} monitor enabled	Oscillator not running	—	10	—	µA
Digital Supply Current (Stop Mode), V_{DD} monitor disabled	Oscillator not running	—	0.1	—	µA
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
Specified Operating Temperature Range		–40	—	+85	°C
SYSCLK (system clock frequency)[2]		0	—	25	MHz
Tsysl (SYSCLK low time)		18	—	—	ns
Tsysh (SYSCLK high time)		18	—	—	ns

Notes:
1. Analog Supply AV+ must be greater than 1 V for V_{DD} monitor to operate.
2. SYSCLK must be at least 32 kHz to enable debugging.

Table 4.1. Pin Definitions (Continued)

Name	'F206, F220, 226, 230, 236	'F221, 231	Type	Description
	48-Pin	32-Pin		
P0.3/INT1	37	25	D I/O A In	Port0 Bit3. (See the Port I/O Sub-System section for complete description).
P0.4/T0	36	24	D I/O A In	Port0 Bit4. (See the Port I/O Sub-System section for complete description).
P0.5/T1	35	23	D I/O A In	Port0 Bit5. (See the Port I/O Sub-System section for complete description).
P0.6/T2	34	22	D I/O A In	Port0 Bit6. (See the Port I/O Sub-System section for complete description).
P0.7/T2EX	33	21	D I/O A In	Port0 Bit7. (See the Port I/O Sub-System section for complete description).
P1.0/CP0+	4	4	D I/O A In	Port1 Bit0. (See the Port I/O Sub-System section for complete description).
P1.1/CP0-	3	3	D I/O A In	Port1 Bit1. (See the Port I/O Sub-System section for complete description).
P1.2/CP0	2	2	D I/O A In	Port1 Bit2. (See the Port I/O Sub-System section for complete description).
P1.3/CP1+	1	1	D I/O A In	Port1 Bit3. (See the Port I/O Sub-System section for complete description).
P1.4/CP1-	48	32	D I/O A In	Port1 Bit4. (See the Port I/O Sub-System section for complete description).
P1.5/CP1	47	31	D I/O A In	Port1 Bit5. (See the Port I/O Sub-System section for complete description).
P1.6/SYSCLK	46	30	D I/O A In	Port1 Bit6. (See the Port I/O Sub-System section for complete description).
P1.7	45	29	D I/O A In	Port1 Bit7. (See the Port I/O Sub-System section for complete description).
P2.0/SCK	24	16	D I/O A In	Port2 Bit0. (See the Port I/O Sub-System section for complete description).
P2.1/MISO	23	15	D I/O A In	Port2 Bit1. (See the Port I/O Sub-System section for complete description).
P2.2/MOSI	22	14	D I/O A In	Port2 Bit2. (See the Port I/O Sub-System section for complete description).
P2.3/NSS	21	13	D I/O A In	Port2 Bit3. (See the Port I/O Sub-System section for complete description).
P2.4	15	11	D I/O A In	Port2 Bit4. (See the Port I/O Sub-System section for complete description).
P2.5	16	12	D I/O A In	Port2 Bit5. (See the Port I/O Sub-System section for complete description).
P2.6	17		D I/O A In	Port2 Bit6. (See the Port I/O Sub-System section for complete description).

Table 4.1. Pin Definitions (Continued)

Name	'F206, F220, 226, 230, 236	'F221, 231	Type	Description
	48-Pin	32-Pin		
P2.7	18		D I/O A In	Port2 Bit7. (See the Port I/O Sub-System section for complete description).
P3.0	44		D I/O A In	Port3 Bit0. (See the Port I/O Sub-System section for complete description).
P3.1	43		D I/O A In	Port3 Bit1. (See the Port I/O Sub-System section for complete description).
P3.2	42		D I/O A In	Port3 Bit2. (See the Port I/O Sub-System section for complete description).
P3.3	41		D I/O A In	Port3 Bit3. (See the Port I/O Sub-System section for complete description).
P3.4	30		D I/O A In	Port3 Bit4. (See the Port I/O Sub-System section for complete description).
P3.5	29		D I/O A In	Port3 Bit5. (See the Port I/O Sub-System section for complete description).
P3.6	20		D I/O A In	Port3 Bit6. (See the Port I/O Sub-System section for complete description).
P3.7	19		D I/O A In	Port3 Bit7. (See the Port I/O Sub-System section for complete description).

SFR Definition 5.4. ADC0H: ADC Data Word (‘F220/1/6 and ‘F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
MSB							LSB	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBF

Bits7–0: ADC Data Word Bits

EXAMPLE: ADC Data Word Conversion Map

AIN – GND(Volts)	ADC0H
REF x (255/256)	0xFF
REF x ½	0x80
REF x (127/256)	0x7F
0	0x00

5.3. ADC Programmable Window Detector

The ADC programmable window detector is very useful in many applications. It continuously compares the ADC output to user-programmed limits and notifies the system when an out-of-band condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (ADWINT in ADC0CN) can also be used in polled mode. The high and low bytes of the reference words are loaded into the ADC Greater-Than and ADC Less-Than registers (ADC0GTH and ADC0LTH).

SFR Definition 5.5. ADC0GTH: ADC Greater-Than Data (‘F220/1/6 and ‘F206)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC5

Bits7–0:
The high byte of the ADC Greater-Than Data Word.

SFR Definition 5.6. ADC0LTH: ADC Less-Than Data Byte (‘F220/1/6 and ‘F206)

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: 0xC7

Bits7–0:
The high byte of the ADC Less-Than Data Word.

9.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

9.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 9.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

9.1.2. MOVX Instruction and Program Memory

The MOVX instruction is typically used to access external data memory. The CIP-51 does not support external data or program memory. In the CIP-51, the MOVX instruction accesses the on-chip program memory space implemented as re-programmable Flash memory and the 1024 bytes of XRAM (optionally available on ‘F226/236 and ‘F206). This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section 10 (Flash Memory) and Section 11 (External RAM) for further details.

Table 9.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A,Rn	Add register to A	1	1
ADD A,direct	Add direct byte to A	2	2
ADD A,@Ri	Add indirect RAM to A	1	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,direct	Add direct byte to A with carry	2	2
ADDC A,@Ri	Add indirect RAM to A with carry	1	2
ADDC A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2

SFR Definition 9.4. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0xD0

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation results in a carry (addition) or a borrow (subtraction). It is cleared to 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag.
This bit is set when the last arithmetic operation results in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general-purpose flag for use under software control.

Bits4–3: RS1–RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00–0x07
0	1	1	0x08–0x0F
1	0	2	0x10–0x17
1	1	3	0x18–0x1F

Note: Any instruction which changes the RS1–RS0 bits must not be immediately followed by the “MOV Rn, A” instruction.

Bit2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
•An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
•A MUL instruction results in an overflow (result is greater than 255).
•A DIV instruction causes a divide-by-zero condition.
The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

9.4. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting up to 22 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE–EIE2). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

9.4.1. MCU Interrupt Sources and Vectors

The MCU allocates 9 interrupt sources to on-chip peripherals. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. The MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 9.4. Refer to the datasheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

9.4.2. External Interrupts

The two external interrupt sources (/INT0 and /INT1) are configurable as active-low level-sensitive or active-low edge-sensitive inputs depending on the setting of IT0 (TCON.0) and IT1 (TCON.2). IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flag for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag follows the state of the external interrupt's input pin. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

9.4.3. Software Controlled Interrupts

The C8051F2xx family of devices features four Software Controlled Interrupts controlled by flags located in the Software Controlled Interrupt Flag Register (SWCINT). See SFR Definition 9.7. When a logic '1' is written to a Software-Controlled Interrupt Flag, the CIP-51 will jump to an associated interrupt service vector (see Table 9.4, “Interrupt Summary,” on page 75). These interrupt flags must be cleared by software.

The Flash Access Limit security feature protects proprietary program code and data from being read by software running on the CIP-51. This feature provides support for OEMs that wish to program the MCU with proprietary value-added firmware before distribution. The value-added firmware can be protected while allowing additional code to be programmed in remaining program memory space later.

The Software Read Limit (SRL) is a 16-bit address that establishes two logical partitions in the program memory space. The first is an upper partition consisting of all the program memory locations at or above the SRL address, and the second is a lower partition consisting of all the program memory locations starting at 0x0000 up to (but excluding) the SRL address. Software in the upper partition can execute code in the lower partition, but is prohibited from reading locations in the lower partition using the MOVC instruction. (Executing a MOVC instruction from the upper partition with a source address in the lower partition will always return a data value of 0x00.) Software running in the lower partition can access locations in both the upper and lower partition without restriction.

The Value-added firmware should be placed in the lower partition. On reset, control is passed to the value-added firmware via the reset vector. Once the value-added firmware completes its initial execution, it branches to a predetermined location in the upper partition. If entry points are published, software running in the upper partition may execute program code in the lower partition, but it cannot read the contents of the lower partition. Parameters may be passed to the program code running in the lower partition either through the typical method of placing them on the stack or in registers before the call or by placing them in prescribed memory locations in the upper partition.

The SRL address is specified using the contents of the Flash Access Register. The 16-bit SRL address is calculated as 0xNN00, where NN is the contents of the SRL Security Register. Thus, the SRL can be located on 256-byte boundaries anywhere in program memory space. However, the 512-byte erase sector size essentially requires that a 512 boundary be used. The contents of a non-initialized SRL security byte is 0x00, thereby setting the SRL address to 0x0000 and allowing read access to all locations in program memory space by default.

SFR Definition 10.1. PSCTL: Program Store RW Control

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	SFR Address: 0x8F

Bits7–2: UNUSED. Read = 000000b, Write = don't care.

Bit1: 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 set to '1'). 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.
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit0: PSWE: Program Store Write Enable.
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX instruction. The location must be erased before writing data.
0: Write to Flash program memory disabled.
1: Write to Flash program memory enabled.

SFR Definition 10.2. FLSCL: Flash Memory Timing Prescaler

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FOSE	FRAE	-	-	FLASCL				10001111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB6

Bit7: FOSE: Flash One-Shot Timer Enable
0: Flash One-shot timer disabled.
1: Flash One-shot timer enabled

Bit6: FRAE: Flash Read Always Enable
0: Flash reads per one-shot timer
1: Flash always in read mode

Bits5–4: UNUSED. Read = 00b, Write = don't care.

Bits3–0: FLASCL: Flash Memory Timing Prescaler.
This register specifies the prescaler value for a given system clock required to generate the correct timing for Flash write/erase operations. If the prescaler is set to 1111b, Flash write/erase operations are disabled.
0000: System Clock < 50 kHz
0001: 50 kHz ≤ System Clock < 100 kHz
0010: 100 kHz ≤ System Clock < 200 kHz
0011: 200 kHz ≤ System Clock < 400 kHz
0100: 400 kHz ≤ System Clock < 800 kHz
0101: 800 kHz ≤ System Clock < 1.6 MHz
0110: 1.6 MHz ≤ System Clock < 3.2 MHz
0111: 3.2 MHz ≤ System Clock < 6.4 MHz
1000: 6.4 MHz ≤ System Clock < 12.8 MHz
1001: 12.8 MHz ≤ System Clock < 25.6 MHz
1010: 25.6 MHz ≤ System Clock < 51.2 MHz*
1011, 1100, 1101, 1110: Reserved Values
1111: Flash Memory Write/Erase Disabled

The prescaler value is the smallest value satisfying the following equation:
FLASCL > log2(System Clock / 50kHz)

*For test purposes. The C8051F2xx is not guaranteed to operate over 25 MHz.

SFR Definition 10.3. FLACL: Flash Access Limit

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: 0xB7

Bits 7–0: FLACL: Flash Memory Access Limit.
This register holds the high byte of the 16-bit program memory read/write/erase limit address. The entire 16-bit access limit address value is calculated as 0xNN00 where NN is replaced by contents of FLACL. A write to this register sets the Flash Access Limit. Any subsequent writes are ignored until the next reset.

12.4. External Reset

The external RST pin provides a means for external circuitry to force the CIP-51 into a reset state. Asserting an active-low signal on the RST pin will cause the CIP-51 to enter the reset state. Although there is a weak pull-up, it may be desirable to provide an external pull-up and/or decoupling of the RST pin to avoid erroneous noise-induced resets. The CIP-51 will remain in reset until at least 12 clock cycles after the active-low RST signal is removed. The PINRSF flag (RSTSRC.0) is set on exit from an external reset. The RST pin is 5 V tolerant.

12.5. Missing Clock Detector Reset

The Missing Clock Detector is essentially a one-shot circuit that is triggered by the MCU system clock. If the system clock goes away for more than 100msec, the one-shot will time out and generate a reset. After a Missing Clock Detector reset, the MCDRSF flag (RSTSRC.2) will be set, signifying the MSD as the reset source; otherwise, this bit reads 0. The state of the RST pin is unaffected by this reset. Setting the MSCLKE bit in the OSCICN register (see SFR Definition 13.1) enables the Missing Clock Detector.

12.6. Comparator 0 Reset

Comparator 0 can be configured as a reset input by writing a 1 to the C0RSEF flag (RSTSRC.5). Comparator 0 should be enabled using CPT0CN.7 (see SFR Definition 8.1) prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. When configured as a reset, if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0–), the MCU is put into the reset state. After a Comparator 0 Reset, the C0RSEF flag (RSTSRC.5) will read 1 signifying Comparator 0 as the reset source; otherwise, this bit reads 0. The state of the RST pin is unaffected by this reset.

12.7. Watchdog Timer Reset

The MCU includes a programmable Watchdog Timer (WDT) running off the system clock. The WDT will force the MCU into the reset state when the watchdog timer overflows. To prevent the reset, the WDT must be restarted by application software before the overflow occurs. If the system experiences a software/hardware malfunction preventing the software from restarting the WDT, the WDT will overflow and cause a reset. This should prevent the system from running out of control.

The WDT is automatically enabled and started with the default maximum time interval on exit from all resets. If desired, the WDT can be disabled by system software or locked 'on' to prevent accidental disabling. Once locked, the WDT cannot be disabled until the next system reset. The state of the RST pin is unaffected by this reset.

12.7.1. Watchdog Usage

The WDT consists of a 21-bit timer running from the programmed system clock. The timer measures the period between specific writes to its control register. If this period exceeds the programmed limit, a WDT reset is generated. The WDT can be enabled and disabled as needed in software, or can be permanently enabled if desired. Watchdog features are controlled via the Watchdog Timer Control Register (WDTCN) shown in SFR Definition 12.1.

Enable/Reset WDT

The watchdog timer is both enabled and reset by writing 0xA5 to the WDTCN register. The user's application software should include periodic writes of 0xA5 to WDTCN as needed to prevent a watchdog timer overflow. The WDT is enabled and reset as a result of any system reset.

15.2. Operation

Only a SPI master device can initiate a data transfer. The SPI is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.1). Writing a byte of data to the SPI data register (SPI0DAT) when in Master Mode starts a data transfer. The SPI master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. The SPI master can be configured to shift in/out from one to eight bits in a transfer operation in order to accommodate slave devices with different word lengths. The SPIFRS bits in the SPI Configuration Register (SPI0CFG.[2:0]) are used to select the number of bits to shift in/out in a transfer operation.

While the SPI master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. The data byte received from the slave replaces the data in the master's data register. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data transfer in both directions is synchronized with the serial clock generated by the master. Figure 15.3 illustrates the full-duplex operation of an SPI master and an addressed slave.

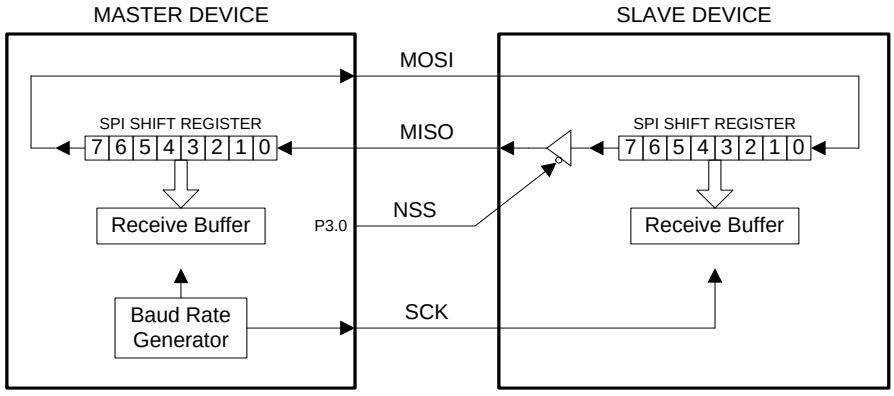

Figure 15.3. Full Duplex Operation

The SPI data register is double buffered on reads, but not on a write. If a write to SPI0DAT is attempted during a data transfer, the WCOL flag (SPI0CN.6) will be set to logic 1 and the write is ignored. The current data transfer will continue uninterrupted. A read of the SPI data register by the system controller actually reads the receive buffer. If the receive buffer still holds unread data from a previous transfer when the last bit of the current transfer is shifted into the SPI shift register, a receive overrun occurs and the RXOVRN flag (SPI0CN.4) is set to logic 1. The new data is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte causing the overrun is lost.

When the SPI is enabled and not configured as a master, it will operate as an SPI slave. Another SPI device acting as a master will initiate a transfer by driving the NSS signal low. The master then shifts data out of the shift register on the MOSI pin using the its serial clock. The SPIF flag is set to logic 1 at the end of a data transfer (when the NSS signal goes high). The slave can load its shift register for the next data transfer by writing to the SPI data register. The slave must make the write to the data register at least one SPI serial clock cycle before the master starts the next transmission. Otherwise, the byte of data already in the slave's shift register will be transferred.

If T1M (CKCON.4) is logic 1, then the above equation becomes:

T1_OVERFLOWRATE = (SYSCLK) / (256 – TH1).

If T1M (CKCON.4) is logic 0, then the above equation becomes:

T1_OVERFLOWRATE = (SYSCLK/12) / (256 – TH1).

The Timer 2 overflow rate, when in Baud Rate Generator Mode and using an internal clock source, is determined solely by the Timer 2 16-bit reload value (RCAP2H:RCAP2L). The Timer 2 clock source is fixed at SYSCLK/2. The Timer 2 overflow rate can be calculated as follows:

T2_OVERFLOWRATE = (SYSCLK/2) / (65536 – [RCAP2H:RCAP2L]).

Timer 2 can be selected as the baud rate generator for RX and/or TX by setting RCLK (T2CON.5) and/or TCLK (T2CON.4), respectively. When either RCLK or TCLK is set to logic 1, Timer 2 interrupts are automatically disabled and the timer is forced into Baud Rate Generator Mode with SYSCLK/2 as its clock source. If a different timebase is required, setting the C/T2 bit (T2CON.1) to logic 1 will allow Timer 2 to be clocked from the external input pin T2. See the Timers section for complete timer configuration details.

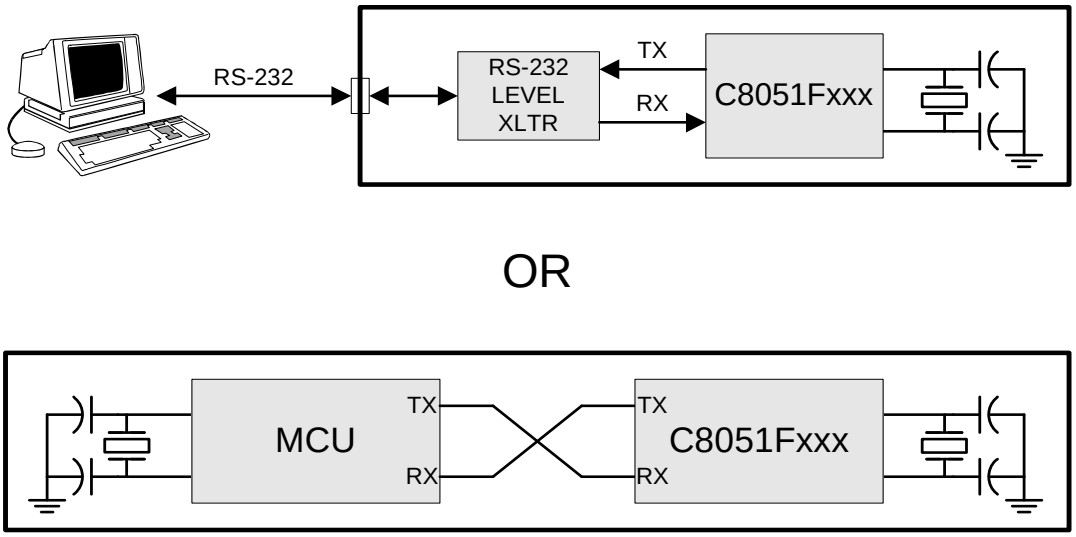

Figure 16.5. UART Modes 1, 2, and 3 Interconnect Diagram

17.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

Timer 0 and Timer 1 behave differently in Mode 3. Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. It can use either the system clock or an external input signal as its time base. The TH0 register is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3, so with Timer 0 in Mode 3, Timer 1 can be turned off and on by switching it into and out of its Mode 3. When Timer 0 is in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used for baud rate generation. Refer to Section 16 (UART) for information on configuring Timer 1 for baud rate generation.

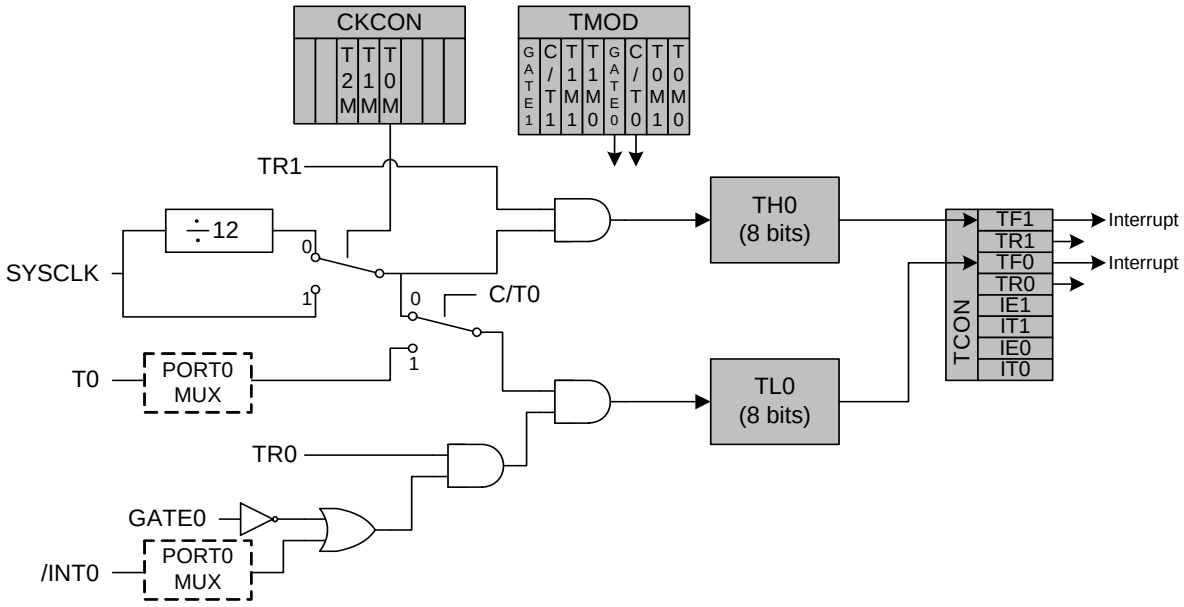

Figure 17.3. T0 Mode 3 Block Diagram

SFR Definition 17.4. TL0: Timer 0 Low 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: 0x8A

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 17.5. TL1: Timer 1 Low 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: 0x8B

Bits 7–0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

SFR Definition 17.6. TH0: Timer 0 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: 0x8C

Bits 7–0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 17.7. TH1: Timer 1 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: 0x8D

Bits 7–0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.