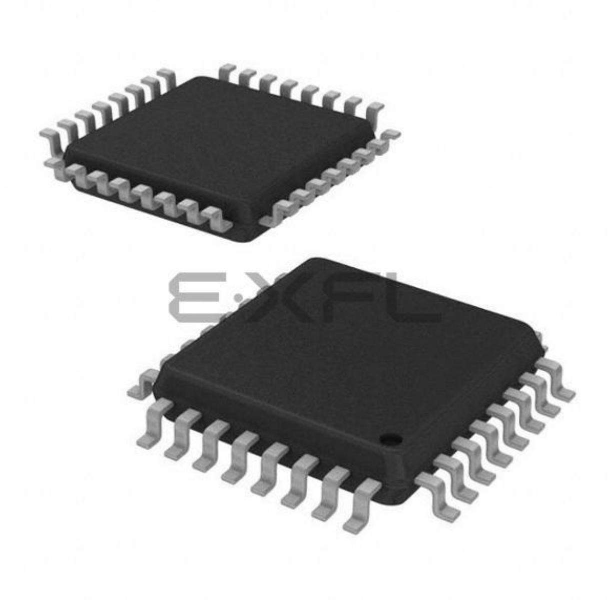

Welcome to E-XFL.COM

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	Obsolete
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	A/D 22x8b
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/c8051f221r

Table of Contents

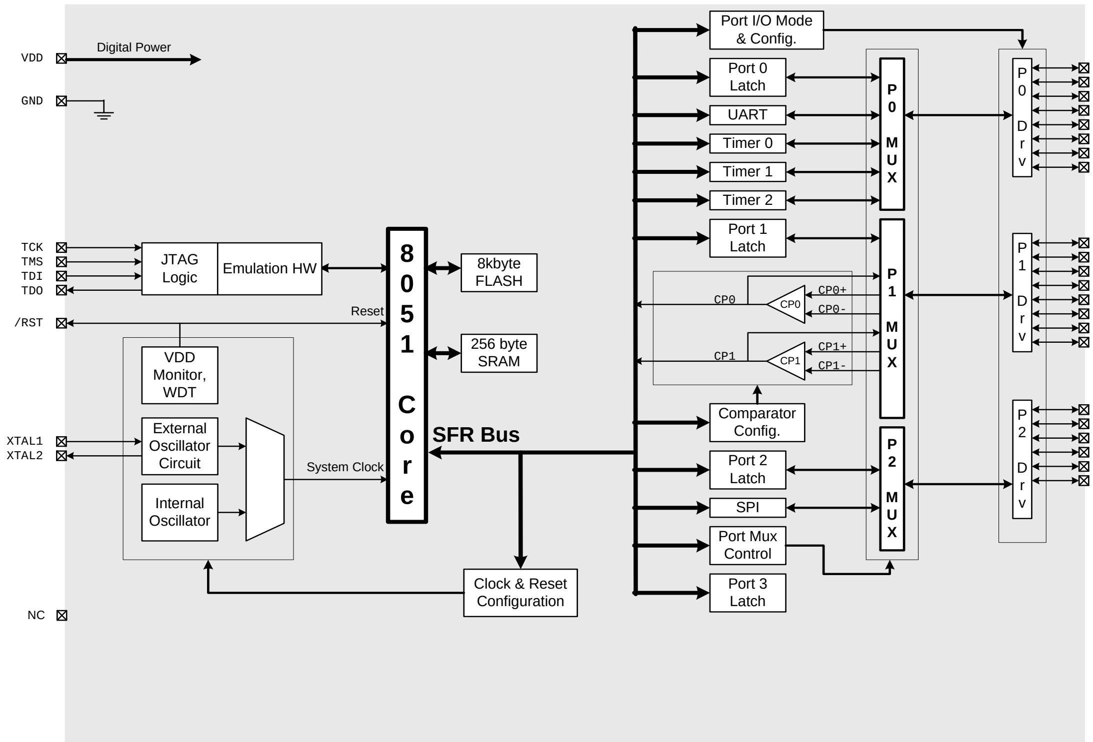

Figure 1.4. C8051F231 Block Diagram (32 LQFP)

1.1. CIP-51™ Microcontroller Core

1.1.1. Fully 8051 Compatible

The C8051F206, C8051F220/1/6 and C8051F230/1/6 utilize Silcon Labs' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set. Standard 803x/805x assemblers and compilers can be used to develop software. The core contains the peripherals included with a standard 8052, including three 16-bit counter/timers, a full-duplex UART, 256 bytes of internal RAM, an optional 1024 bytes of XRAM, 128 byte Special Function Register (SFR) address space, and four byte-wide I/O Ports.

1.1.2. Improved Throughput

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute with a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with only four instructions taking more than four system clock cycles.

The CIP-51 has a total of 109 instructions. The number of instructions versus the system clock cycles to execute them is as follows:

Instructions	26	50	5	14	7	3	1	2	1
Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8

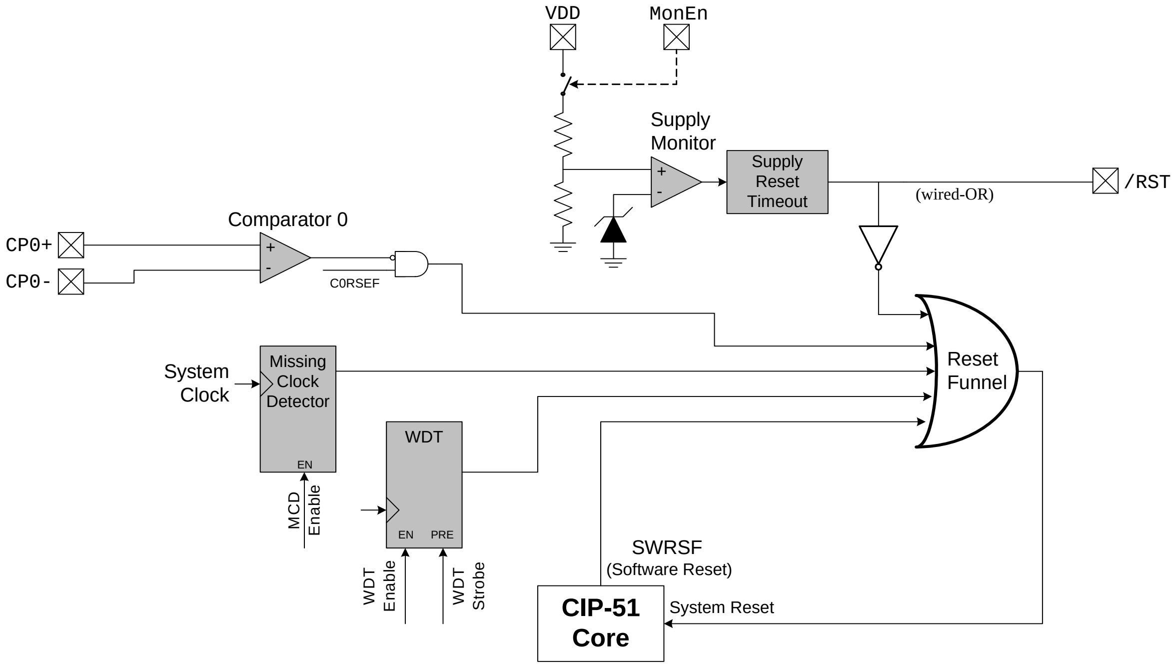

Figure 1.6. Comparison of Peak MCU Throughputs

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.

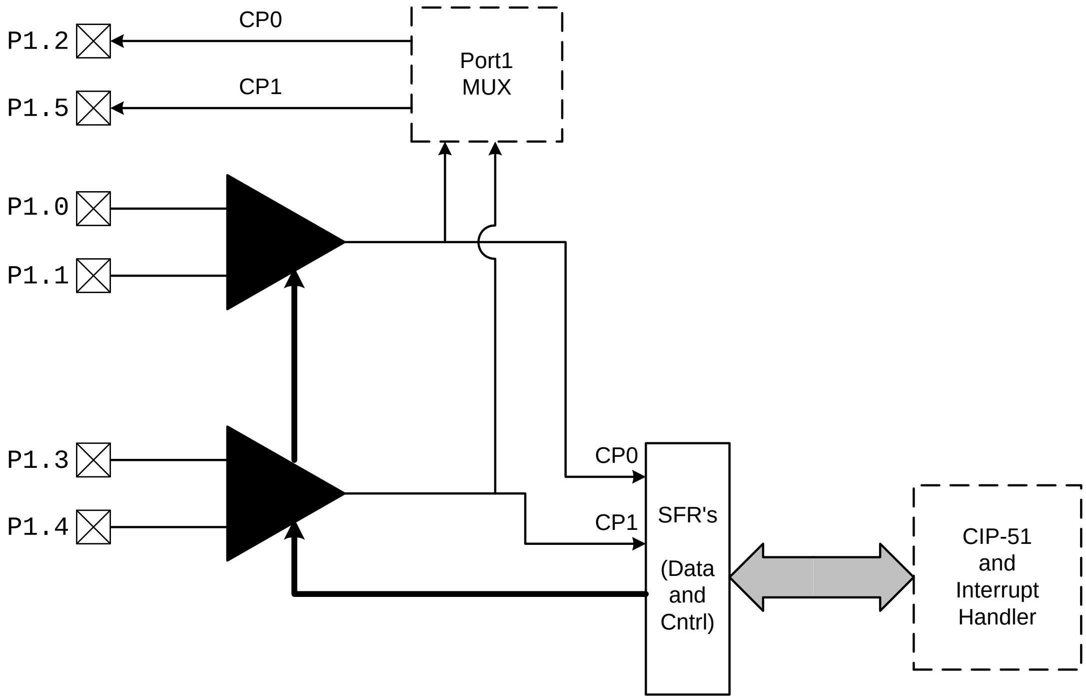

Figure 1.11. Comparator Diagram

2. Absolute Maximum Ratings

Table 2.1. Absolute Maximum Ratings*

Parameter	Conditions	Min	Typ	Max	Units
Ambient Temperature under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on any Pin (except V_{DD} and Port I/O) with respect to DGND		–0.3	—	V_{DD} + 0.3	V
Voltage on any Port I/O Pin or $\overline{\text{RST}}$ pins with respect to DGND		–0.3	—	5.8	V
Voltage on V_{DD} with respect to DGND		–0.3		4.2	V
Total Power Dissipation		—	1.0	800	W
Maximum Output Current Sunk by any Port pin		—	—	200	mA
Maximum Output Current Sunk by any other I/O pin		—	—	25	mA
Maximum Output Current Sourced by any Port pin		—	—	200	mA
Maximum Output Current Sourced by any other I/O pin		—	—	25	mA

***Note:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

SFR Definition 8.1. CPT0CN: Comparator 0 Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP1	CP0HYP0	CP0HYN1	CP0HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9E

Bit7: CP0EN: Comparator 0 Enable Bit
0: Comparator 0 Disabled.
1: Comparator 0 Enabled.

Bit6: CP0OUT: Comparator 0 Output State Flag
0: Voltage on CP0+ < CP0–
1: Voltage on CP0+ > CP0–

Bit5: CP0RIF: Comparator 0 Rising-Edge Interrupt Flag
0: No Comparator 0 Rising-Edge Interrupt has occurred since this flag was cleared
1: Comparator 0 Rising-Edge Interrupt has occurred since this flag was cleared

Bit4: CP0FIF: Comparator 0 Falling-Edge Interrupt Flag
0: No Comparator 0 Falling-Edge Interrupt has occurred since this flag was cleared
1: Comparator 0 Falling-Edge Interrupt has occurred since this flag was cleared

Bit3–2: CP0HYP1–0: Comparator 0 Positive Hysteresis Control Bits
00: Positive Hysteresis Disabled
01: Positive Hysteresis = 2 mV
10: Positive Hysteresis = 4 mV
11: Positive Hysteresis = 10 mV

Bit1–0: CP0HYN1–0: Comparator 0 Negative Hysteresis Control Bits
00: Negative Hysteresis Disabled
01: Negative Hysteresis = 2 mV
10: Negative Hysteresis = 4 mV
11: Negative Hysteresis = 10 mV

Table 8.1. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +85 ×C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Response Time1*	(CP+) – (CP–) = 100 mV	—	4	—	µs
Response Time2*	(CP+) – (CP–) = 10 mV	—	12	—	µs
Common Mode Rejection Ratio		—	1.5	4	mV/V
Positive Hysteresis1	CPnHYP1-0 = 00	—	0	1	mV
Positive Hysteresis2	CPnHYP1-0 = 01	2	4.5	7	mV
Positive Hysteresis3	CPnHYP1-0 = 10	4	9	15	mV
Positive Hysteresis4	CPnHYP1-0 = 11	10	17	25	mV
Negative Hysteresis1	CPnHYN1-0 = 00	—	0	1	mV
Negative Hysteresis2	CPnHYN1-0 = 01	2	4.5	7	mV
Negative Hysteresis3	CPnHYN1-0 = 10	4	9	15	mV
Negative Hysteresis4	CPnHYN1-0 = 11	10	17	25	mV
Inverting or Non-inverting Input Voltage Range		–0.25	—	(V_{DD}) + 0.25	V
Input Capacitance		—	7	—	pF
Input Bias Current		–5	0.001	+5	nA
Input Offset Voltage		–10	—	+10	mV
POWER SUPPLY					
Power-up Time	CPnEN from 0 to 1	—	20	—	µs
Power Supply Rejection		—	0.1	1	mV/V
Supply Current	Operating Mode (each comparator) at DC	—	1.5	4	µA
***Note:** CPnHYP1-0 = CPnHYN1-0 = 00.					

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

9.2. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. There are 256 bytes of internal data memory and 8 kB of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 9.2.

9.2.1. Program Memory

The CIP-51 has a 8 kB program memory space. The MCU implements 8320 bytes of this program memory space as in-system, reprogrammable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x207F. Note: 512 bytes (0x1E00 – 0x1FFF) of this memory are reserved for factory use and are not available for user program storage.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. 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 for further details.

9.2.2. 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 bit 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 SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F will access the upper 128 bytes of data memory. Figure 9.2 illustrates the data memory organization of the CIP-51.

Additionally, the C8051F206/226/236 feature 1024 Bytes of RAM mapped in the external data memory space. All address locations may be accessed using the MOVX instruction. (Please see Section 11).

Table 9.4. Interrupt Summary (Continued)

Interrupt Source	Interrupt Vector	Priority Order	Interrupt-Pending Flag	Enable
ADC0 End of Conversion	0x007B	15	ADCINT (ADC0CN.5)	EADC0 (EIE2.1)
Software Controlled Interrupt 0	0x0083	16	SCI0 (SWCINT.4)	ESCI0 (EIE2.2)
Software Controlled Interrupt 1	0x008B	17	SCI1 (SWCINT.5)	ESCI1 (EIE2.3)
Software Controlled Interrupt 2	0x0093	18	SCI2 (SWCINT.6)	ESCI2 (EIE2.4)
Software Controlled Interrupt 3	0x009B	19	SCI3 (SWCINT.7)	ESCI3 (EIE2.5)
Unused Interrupt Location	0x00A3	20	None	Reserved (EIE2.6)
External Crystal OSC Ready	0x00AB	21	XTLVLD (OSCXCN.7)	EXVLD (EIE2.7)

9.4.4. 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.

9.4.5. 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. 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) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. NOTE: If a Flash write or erase is performed, the MCU is stalled during the operation and interrupts will not be serviced until the operation is complete. 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 9.12. EIP1: Extended Interrupt Priority 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PCP1R	PCP1F	PCP0R	PCP0F	-	PWADC0	-	PSPI0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF6

Bit7: PCP1R: Comparator 1 (CP1) Rising Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 rising interrupt set to low priority level.
1: CP1 rising interrupt set to high priority level.

Bit6: PCP1F: Comparator 1 (CP1) Falling Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 falling interrupt set to low priority level.
1: CP1 falling interrupt set to high priority level.

Bit5: PCP0R: Comparator 0 (CP0) Rising Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 rising interrupt set to low priority level.
1: CP0 rising interrupt set to high priority level.

Bit4: PCP0F: Comparator 0 (CP0) Falling Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 falling interrupt set to low priority level.
1: CP0 falling interrupt set to high priority level.

Bit3: Reserved. Read = 0, Write = don't care.

Bit2: PWADC0: Analog-to-Digital Converter 0 window compare (ADC0) Interrupt Priority Control.
This bit sets the priority of the ADC0 window compare interrupt.
0: ADC0 window compare interrupt set to low priority level.
1: ADC0 window compare interrupt set to high priority level.

Bit1: UNUSED. Read = 0, Write = don't care.

Bit0: PSPI0: Serial Peripheral Interface 0 Interrupt Priority Control.
This bit sets the priority of the SPI0 interrupt.
0: SPI0 interrupt set to low priority level.
1: SPI0 interrupt set to high priority level.

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.

12.1. Power-on Reset

The CIP-51 incorporates a power supply monitor that holds the MCU in the reset state until V_{DD} rises above the VRST level during power-up. (See Figure 12.2 for timing diagram, and refer to Table 12.1 for the Electrical Characteristics of the power supply monitor circuit.) The $\overline{\text{RST}}$ pin is asserted (low) until the end of the 100msec V_{DD} Monitor timeout in order to allow the V_{DD} supply to become stable. On 48-pin packages, the V_{DD} monitor is enabled by pulling the MONEN pin high and is disabled by pulling the MONEN pin low. The MONEN pin should never be left floating. On 32-pin packages, the V_{DD} monitor is always enabled and cannot be disabled.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. All of the other reset flags in the RSTSRC Register are indeterminate. PORSF is cleared by all other resets. Since all resets cause program execution to begin at the same location (0x0000), software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset.

12.2. Software Forced Reset

Writing a 1 to the PORSF bit forces a Power-On Reset as described in Section 12.1.

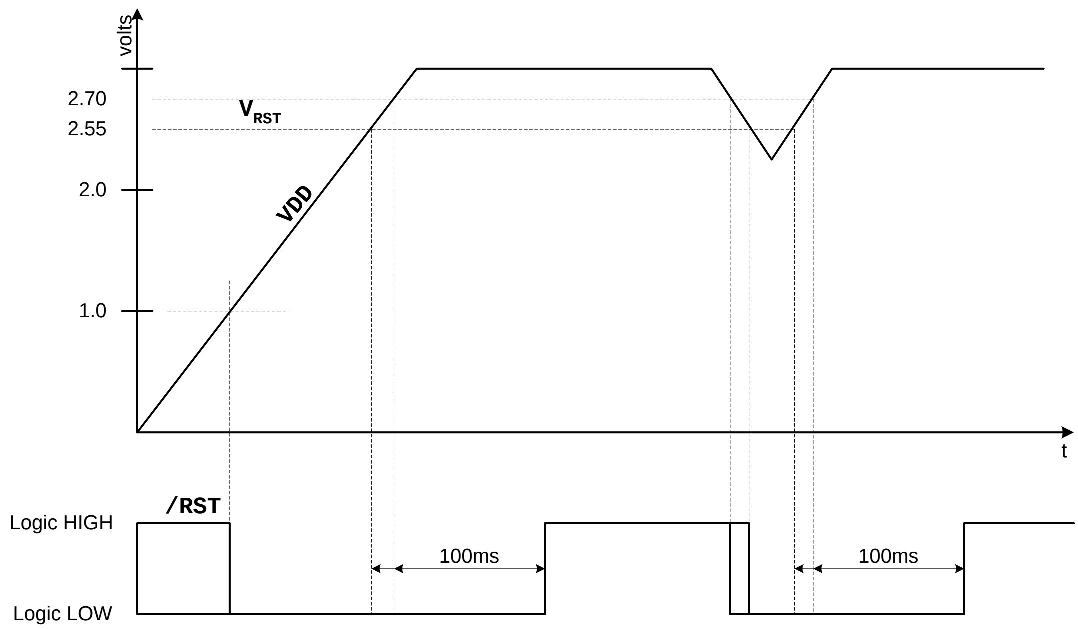

Figure 12.2. V_{DD} Monitor Timing Diagram

12.3. Power-fail Reset

When the V_{DD} monitor is enabled, the MONEN pin (not on C8051F221/F231 32 pin parts) is "pulled high", and power-down transition or power irregularity causes V_{DD} to drop below VRST, the power supply monitor will drive the $\overline{\text{RST}}$ pin low and return the CIP-51 to the reset state (see Figure 12.2). When V_{DD} returns to a level above VRST, the CIP-51 will leave the reset state in the same manner as that for the power-on reset. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if V_{DD} dropped below the level required for data retention. If the PORSF flag is set, the data may no longer be valid.

14. Port Input/Output

Description

The C8051F221/231 have three I/O Ports: Port0, Port1, and Port2. The C8051F206, C8051F220/6 and C8051F230/6 have four I/O Ports: Port0, Port1, Port2, and Port3. A wide array of digital resources can be assigned to these ports by the simple configuration of the port's corresponding multiplexer (MUX). Please see Figure 8.1. Additionally, all external port pins are available as analog input.

14.1. Port I/O Initialization

Port I/O initialization is straightforward. Registers PRT0MX, PRT1MX and PRT2MX must be loaded with the appropriate values to select the digital I/O functions required by the design. The output driver characteristics of the I/O pins are defined using the Port Configuration Registers PRT0CF, PRT1CF, PRT2CF and PRT3CF. Each Port Output driver can be configured as either Open Drain or Push-Pull. This is required even for the digital resources selected in the PRTnMX registers, and is not automatic.

Any or all pins may be configured as digital I/O or as analog input. The default mode is digital I/O. The P0MODE, P1MODE, P2MODE, and P3MODE special function registers are used to configure the port pins as digital or analog as defined in this section.

The final step is initializing the individual resources selected using the appropriate setup registers. Initialization procedures for the various digital resources may be found in the detailed explanation of each available function. The reset state of each register is shown in the figures that describe each individual register.

NOTE: The input mode of pins configured for use with Timer 0, 1, or 2 must be manually configured.

1. The output mode of all ports pins must be configured regardless of whether the port pin is either standard general-purpose I/O or controlled by a digital peripheral.
2. For all pins used as Timer inputs (P0.4/T0, P0.5/T1, P0.6/T2, and P0.7/T2EX), the output mode must be "open-drain" (which is the reset state), and "1" must be written to the associated port pin to prevent possible contention for the port pin that could result in an overcurrent condition. For example, to configure a Timer0, set PRT0MX's T0E Timer0 enable bit to '1' to route Timer0 to Port Pin P0.4. Then place P0.4/T0 in open-drain configuration (which is set in PRT0CF by default), and write a '1' to P0.4 to set its output state to high impedance for use as a digital peripheral input (port pins also default to logic high state upon reset). Lastly, ensure P0MODE.4 is '1' for digital input mode. (All pins default to digital input mode upon reset.)

15. Serial Peripheral Interface Bus

The Serial Peripheral Interface (SPI) provides access to a four-wire, full-duplex, serial bus. SPI supports the connection of multiple slave devices to a master device on the same bus. A separate slave-select signal (NSS) is used to select a slave device and enable a data transfer between the master and the selected slave. Multiple masters on the same bus are also supported. Collision detection is provided when two or more masters attempt a data transfer at the same time. The SPI can operate as either a master or a slave. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency.

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, and the serial input data synchronously with the system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less that 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 ¼ the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the system clock.

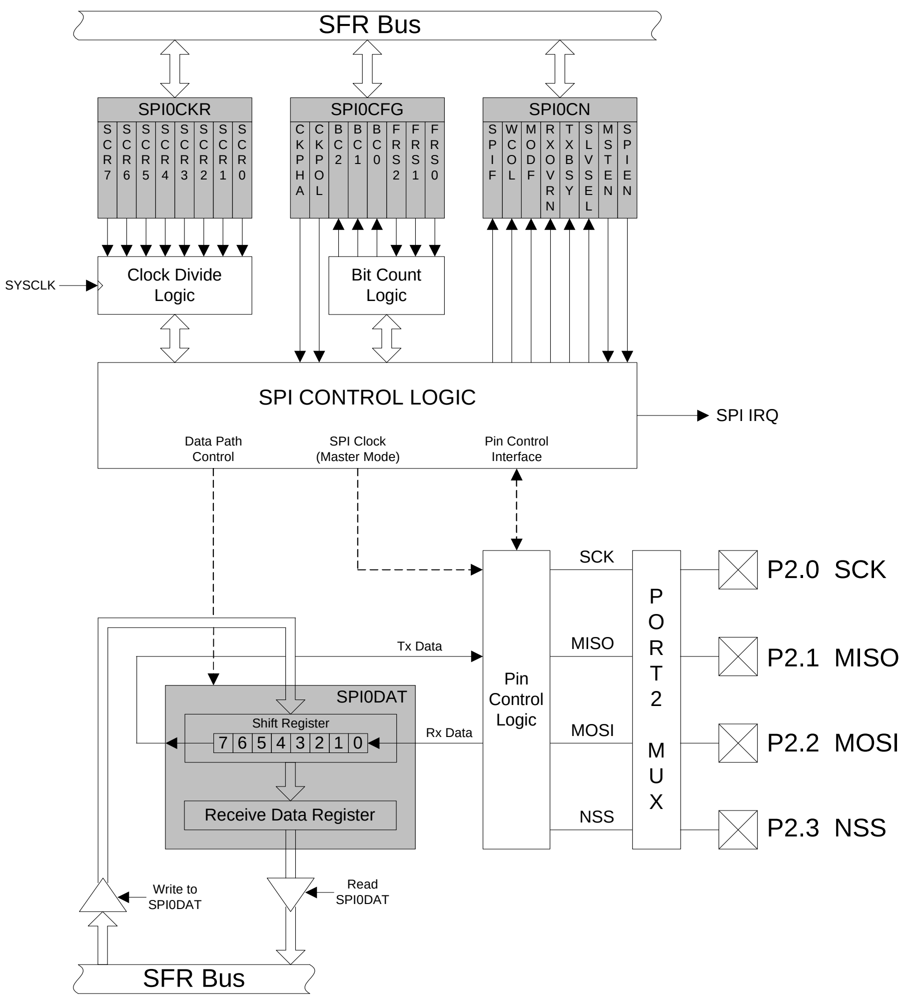

Figure 15.1. SPI Block Diagram

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.

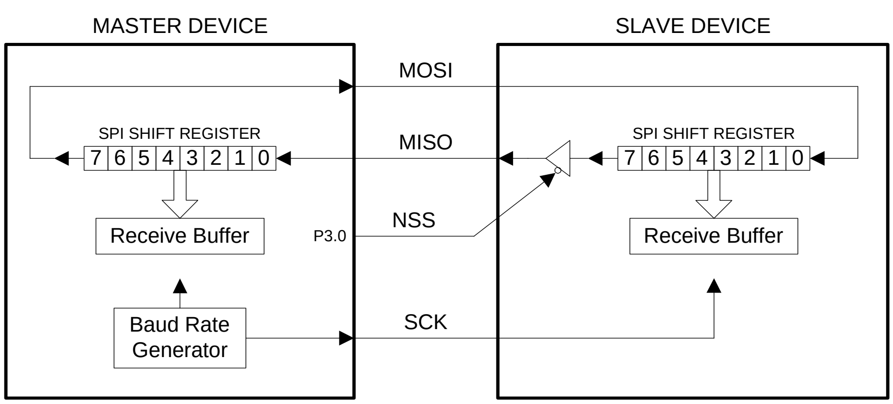

Figure 15.3. Full Duplex Operation

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.

SFR Definition 15.2. SPI0CN: SPI Control

R/W	R/W	R/W	R/W	R	R	R/W	R/W	Reset Value
SPIF	WCOL	MODF	RXOVRN	TXBSY	SLVSEL	MSTEN	SPIEN	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF8

Bit7: SPIF: SPI Interrupt Flag.
This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit6: WCOL: Write Collision Flag.
This bit is set to logic 1 by hardware (and generates a SPI interrupt) to indicate a write to the SPI data register was attempted while a data transfer was in progress. It is cleared by software.

Bit5: MODF: Mode Fault Flag.
This bit is set to logic 1 by hardware (and generates a SPI interrupt) when a master mode collision is detected (NSS is low and MSTEN = 1). This bit is not automatically cleared by hardware. It must be cleared by software.

Bit4: RXOVRN: Receive Overrun Flag.
This bit is set to logic 1 by hardware (and generates a SPI interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI shift register. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit3: TXBSY: Transmit Busy Flag.
This bit is set to logic 1 by hardware while a master mode transfer is in progress. It is cleared by hardware at the end of the transfer.

Bit2: SLVSEL: Slave Selected Flag.
This bit is set to logic 1 whenever the NSS pin is low indicating it is enabled as a slave. It is cleared to logic 0 when NSS is high (slave disabled).

Bit1: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit0: SPIEN: SPI Enable.
This bit enables/disables the SPI.
0: SPI disabled.
1: SPI enabled.

Table 16.2. Oscillator Frequencies for Standard Baud Rates (Continued)

Oscillator Frequency (MHz)	Divide Factor	Timer 1 Load Value*	Resulting Baud Rate**
7.3728	64	0xFC	115200
5.5296	48	0xFD	115200
3.6864	32	0xFE	115200
1.8432	16	0xFF	115200
24.576	320	0xEC	76800
25.0	434	0xE5	57600 (57870)
25.0	868	0xCA	28800
24.576	848	0xCB	28800 (28921)
24.0	833	0xCC	28800 (28846)
23.592	819	0xCD	28800 (28911)
22.1184	768	0xD0	28800
18.432	640	0xD8	28800
16.5888	576	0xDC	28800
14.7456	512	0xE0	28800
12.9024	448	0xE4	28800
11.0592	348	0xE8	28800
9.216	320	0xEC	28800
7.3728	256	0xF0	28800
5.5296	192	0xF4	28800
3.6864	128	0xF8	28800
1.8432	64	0xFC	28800

SFR Definition 16.1. SBUF: Serial (UART) Data Buffer

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: 0x99

Bits7–0: SBUF.[7:0]: Serial Data Buffer Bits 7–0 (MSB-LSB)
This is actually two registers; a transmit and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. Moving a byte to SBUF is what initiates the transmission. When data is moved from SBUF, it comes from the receive buffer.

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

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. The TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If 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.

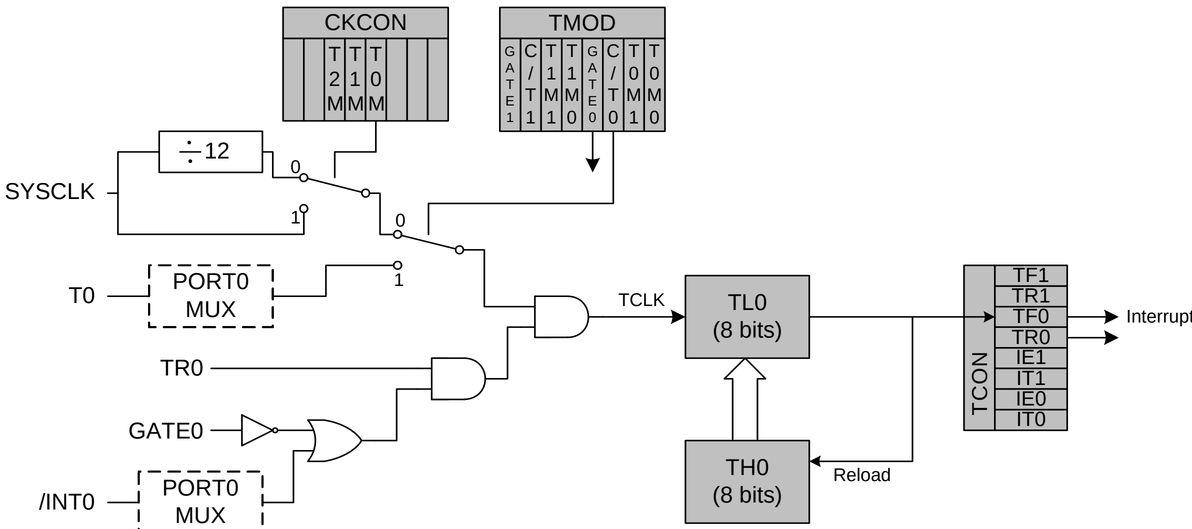

Figure 17.2. T0 Mode 2 Block Diagram