



Welcome to [E-XFL.COM](https://www.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	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I <sup>2</sup> C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/silicon-labs/c8051f505-iqr">https://www.e-xfl.com/product-detail/silicon-labs/c8051f505-iqr</a>

## List of Figures

Figure 1.1. C8051F500/1/4/5 Block Diagram .....	17
Figure 1.2. C8051F508/9-F510/1 Block Diagram .....	18
Figure 1.3. C8051F502/3/6/7 Block Diagram .....	19
Figure 3.1. QFP-48 Pinout Diagram (Top View) .....	25
Figure 3.2. QFN-48 Pinout Diagram (Top View) .....	26
Figure 3.3. QFN-40 Pinout Diagram (Top View) .....	27
Figure 3.4. QFP-32 Pinout Diagram (Top View) .....	28
Figure 3.5. QFN-32 Pinout Diagram (Top View) .....	29
Figure 4.1. QFP-48 Package Drawing .....	30
Figure 4.2. QFP-48 Landing Diagram .....	31
Figure 4.3. QFN-48 Package Drawing .....	32
Figure 4.4. QFN-48 Landing Diagram .....	33
Figure 4.5. Typical QFN-40 Package Drawing .....	34
Figure 4.6. QFN-40 Landing Diagram .....	35
Figure 4.7. QFP-32 Package Drawing .....	36
Figure 4.8. QFP-32 Package Drawing .....	37
Figure 4.9. QFN-32 Package Drawing .....	38
Figure 4.10. QFN-32 Package Drawing .....	39
Figure 5.1. Minimum VDD Monitor Threshold vs. System Clock Frequency .....	44
Figure 6.1. ADC0 Functional Block Diagram .....	52
Figure 6.2. ADC0 Tracking Modes .....	54
Figure 6.3. 12-Bit ADC Tracking Mode Example .....	55
Figure 6.4. 12-Bit ADC Burst Mode Example With Repeat Count Set to 4 .....	56
Figure 6.5. ADC0 Equivalent Input Circuit .....	58
Figure 6.6. ADC Window Compare Example: Right-Justified Data .....	69
Figure 6.7. ADC Window Compare Example: Left-Justified Data .....	69
Figure 6.8. ADC0 Multiplexer Block Diagram .....	70
Figure 7.1. Temperature Sensor Transfer Function .....	72
Figure 8.1. Voltage Reference Functional Block Diagram .....	73
Figure 9.1. Comparator Functional Block Diagram .....	75
Figure 9.2. Comparator Hysteresis Plot .....	76
Figure 9.3. Comparator Input Multiplexer Block Diagram .....	81
Figure 10.1. External Capacitors for Voltage Regulator Input/Output— Regulator Enabled .....	84
Figure 10.2. External Capacitors for Voltage Regulator Input/Output— Regulator Disabled .....	85
Figure 11.1. CIP-51 Block Diagram .....	87
Figure 12.1. C8051F50x-F51x Memory Map .....	97
Figure 12.2. Flash Program Memory Map .....	98
Figure 13.1. SFR Page Stack .....	101
Figure 13.2. SFR Page Stack While Using SFR Page 0x0 To Access SPI0DAT ..	102
Figure 13.3. SFR Page Stack After CAN0 Interrupt Occurs .....	103
Figure 13.4. SFR Page Stack Upon PCA Interrupt Occurring During a CAN0 ISR	104

# C8051F50x/F51x

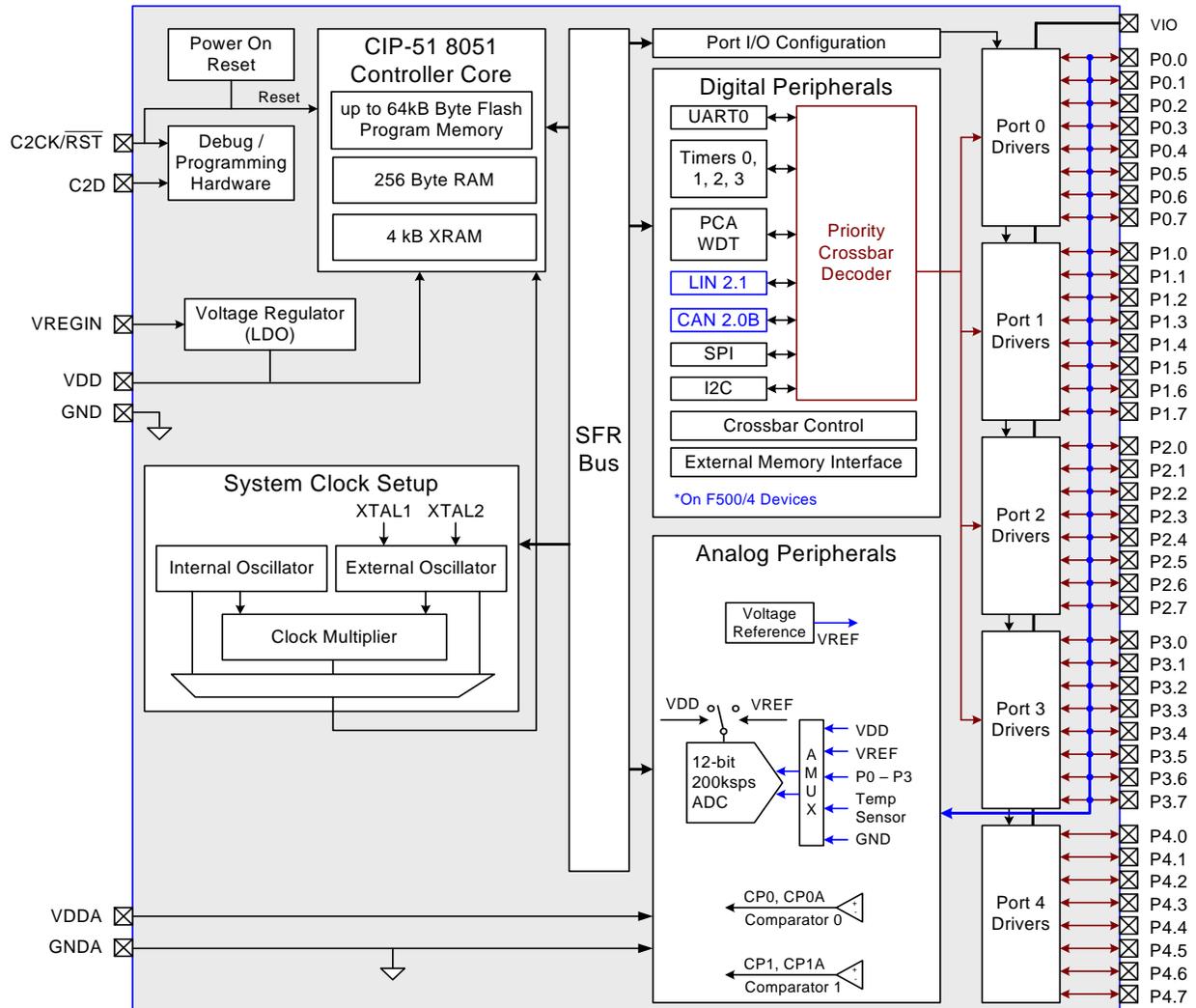


Figure 1.1. C8051F500/1/4/5 Block Diagram

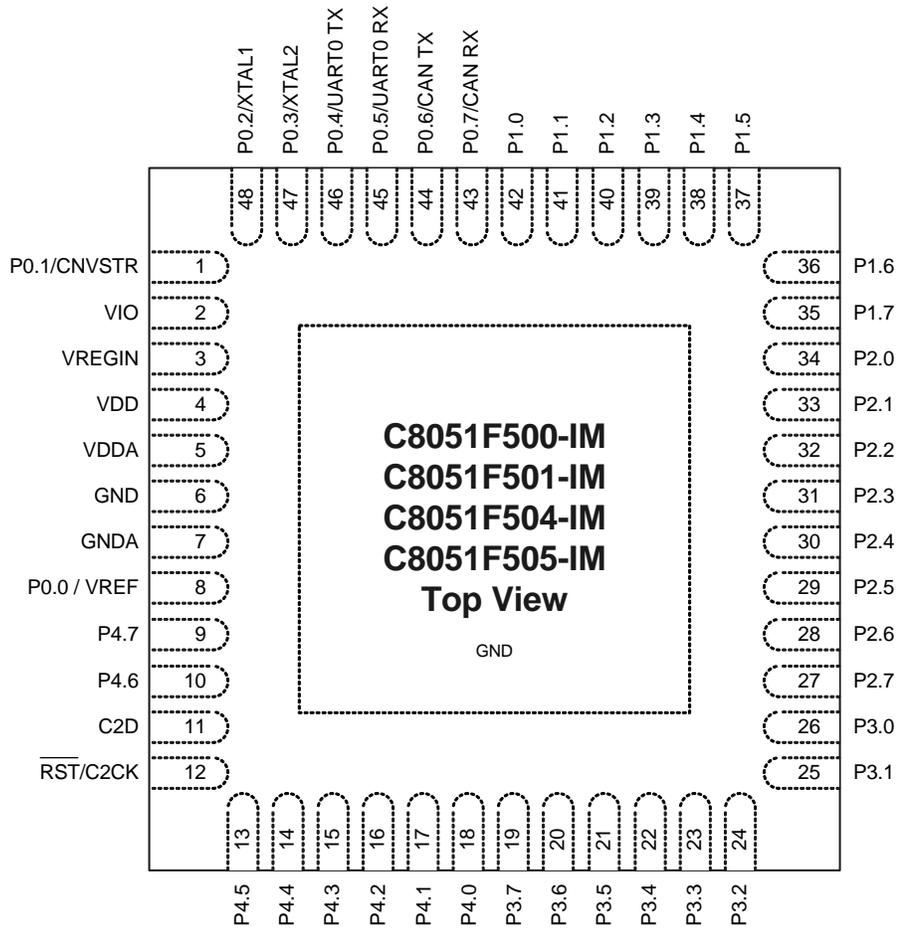
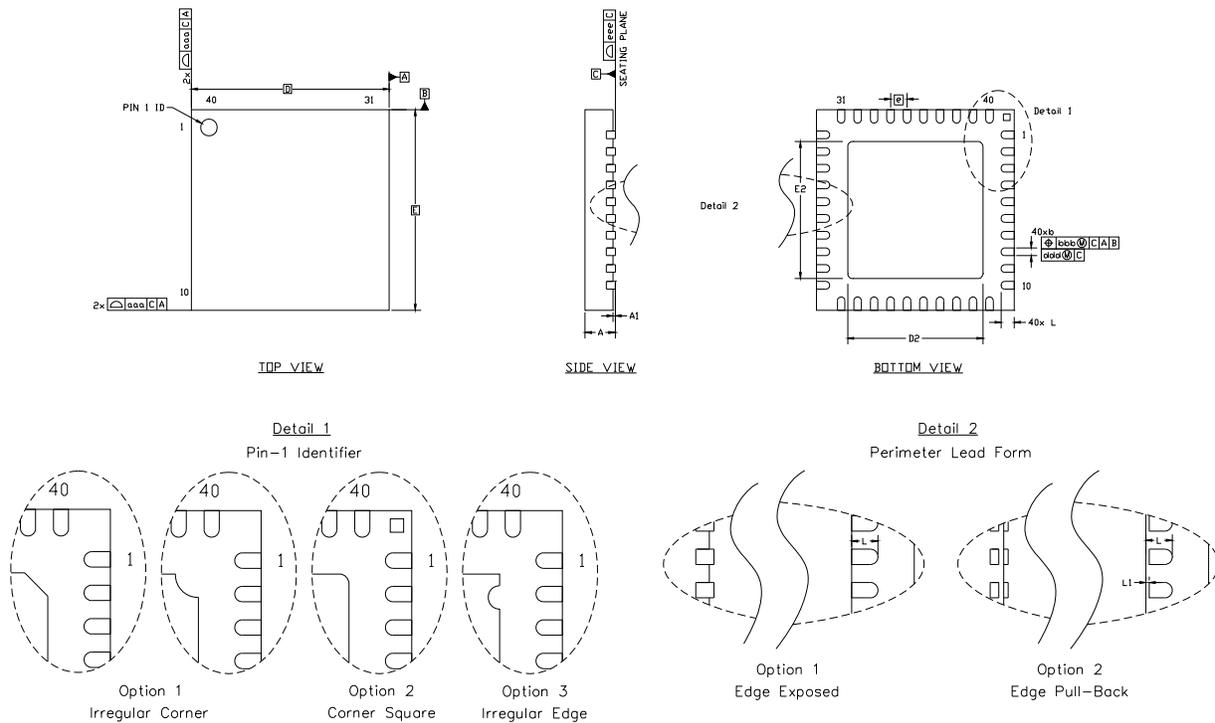


Figure 3.2. QFN-48 Pinout Diagram (Top View)

## 4.3. QFN-40 Package Specifications



**Figure 4.5. Typical QFN-40 Package Drawing**

**Table 4.5. QFN-40 Package Dimensions**

Dimension	Min	Typ	Max	Dimension	Min	Typ	Max
A	0.80	0.85	0.90	E2	4.00	4.10	4.20
A1	0.00		0.05	L	0.35	0.40	0.45
b	0.18	0.23	0.28	L1			0.10
D	6.00 BSC			aaa			0.10
D2	4.00	4.10	4.20	bbb			0.10
e	0.50 BSC			ddd			0.05
E	6.00 BSC			eee			0.08

**Notes:**

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC Solid State Outline MO-220, variation VJJD-5, except for features A, D2, and E2 which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

# C8051F50x/F51x

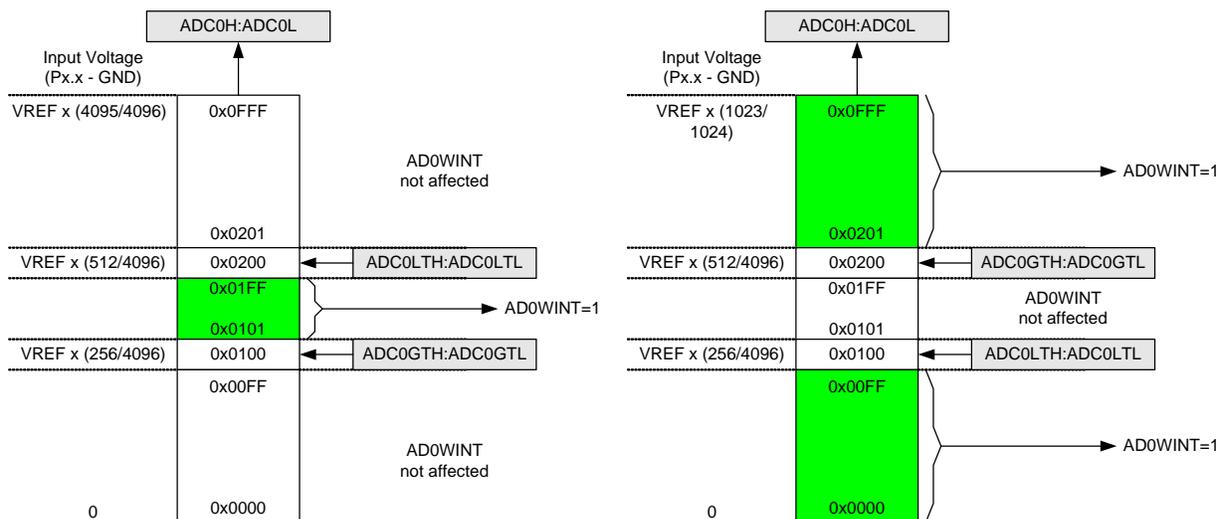


Figure 6.6. ADC Window Compare Example: Right-Justified Data

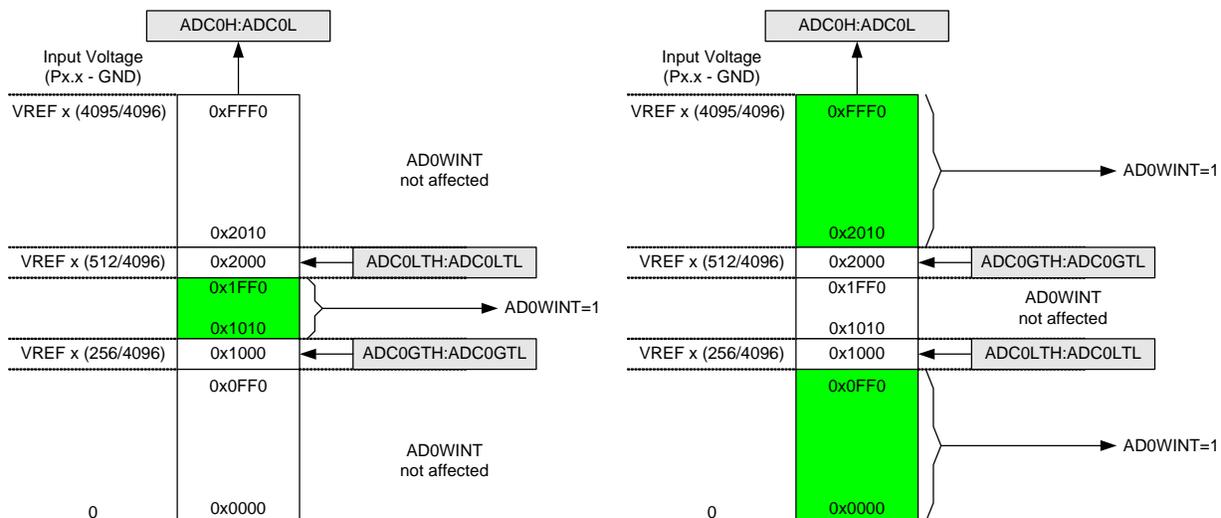


Figure 6.7. ADC Window Compare Example: Left-Justified Data

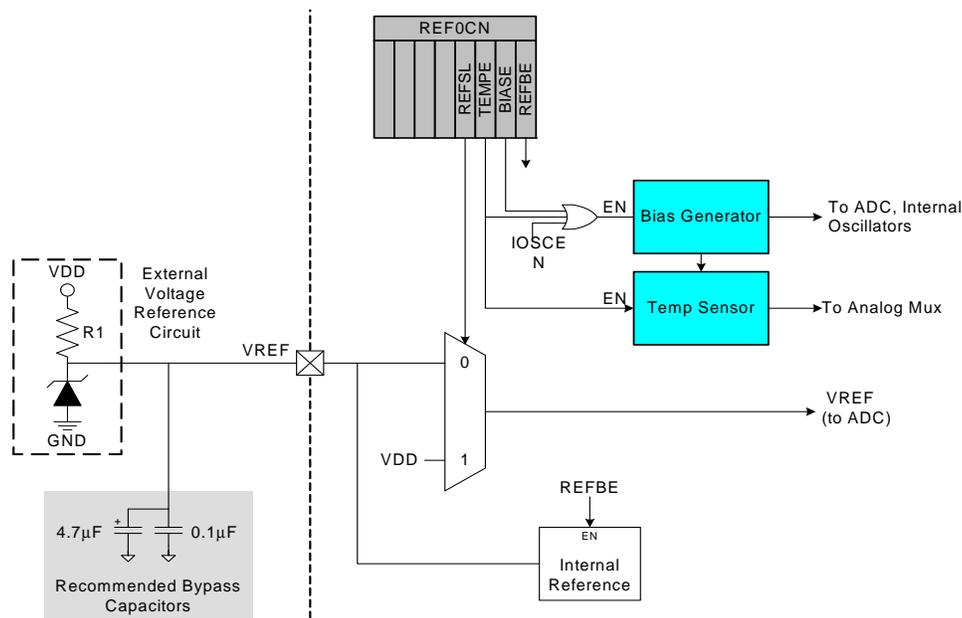
## 8. Voltage Reference

The Voltage reference multiplexer on the C8051F50x/F51x devices is configurable to use an externally connected voltage reference, the on-chip reference voltage generator routed to the VREF pin, or the  $V_{DD}$  power supply voltage (see Figure 8.1). The REFSL bit in the Reference Control register (REF0CN, SFR Definition 8.1) selects the reference source for the ADC. For an external source or the on-chip reference, REFSL should be set to 0 to select the VREF pin. To use  $V_{DD}$  as the reference source, REFSL should be set to 1.

The BIASE bit enables the internal voltage bias generator, which is used by the ADC, Temperature Sensor, and internal oscillator. This bias is automatically enabled when any peripheral which requires it is enabled, and it does not need to be enabled manually. The bias generator may be enabled manually by writing a 1 to the BIASE bit in register REF0CN. The electrical specifications for the voltage reference circuit are given in Table 5.11.

The on-chip voltage reference circuit consists of a temperature stable bandgap voltage reference generator and a gain-of-two output buffer amplifier. The output voltage is selectable between 1.5 V and 2.25 V. The on-chip voltage reference can be driven on the VREF pin by setting the REFBE bit in register REF0CN to a 1. The maximum load seen by the VREF pin must be less than 200  $\mu\text{A}$  to GND. Bypass capacitors of 0.1  $\mu\text{F}$  and 4.7  $\mu\text{F}$  are recommended from the VREF pin to GND. If the on-chip reference is not used, the REFBE bit should be cleared to 0. Electrical specifications for the on-chip voltage reference are given in Table 5.11.

**Important Note about the VREF Pin:** When using either an external voltage reference or the on-chip reference circuitry, the VREF pin should be configured as an analog pin and skipped by the Digital Crossbar. Refer to Section “20. Port Input/Output” on page 177 for the location of the VREF pin, as well as details of how to configure the pin in analog mode and to be skipped by the crossbar.



**Figure 8.1. Voltage Reference Functional Block Diagram**

# C8051F50x/F51x

## 12.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F50x/F51x devices implement 64 kB or 32 kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0xFFFF in 64 kB devices and addresses 0x0000 to 0x7FFF in 32 kB devices. The address 0xFBFF in 64 kB devices and 0x7FFF in 32 kB devices serves as the security lock byte for the device. Addresses above 0xFDFE are reserved in the 64 kB devices.

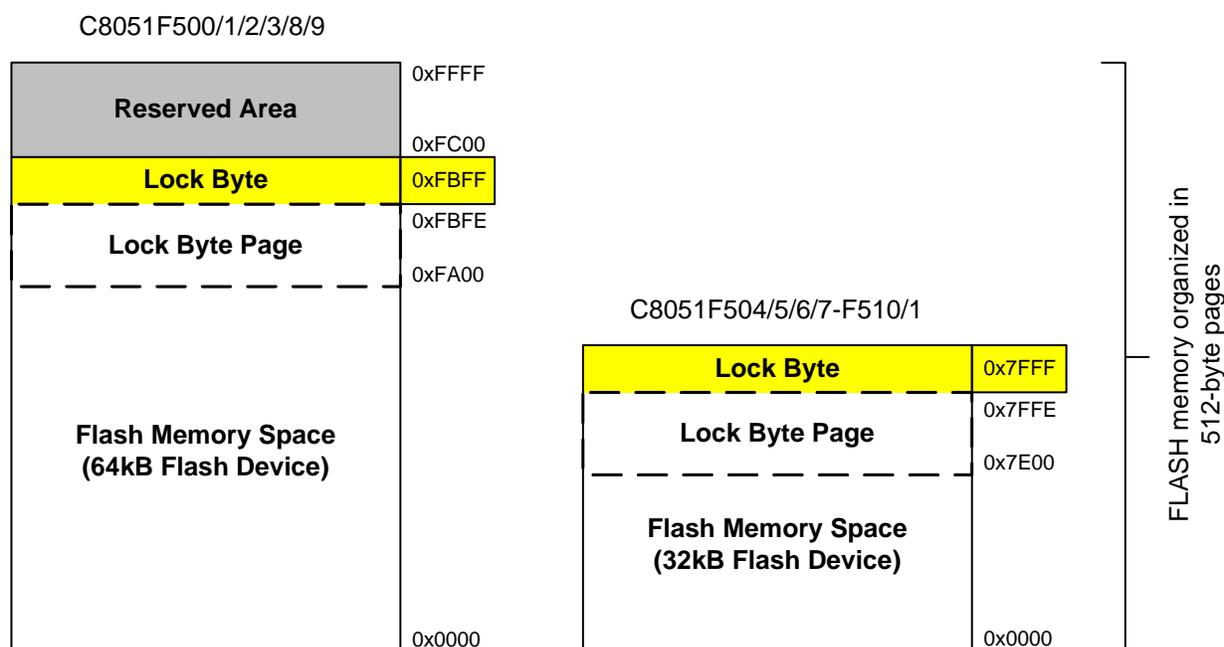


Figure 12.2. Flash Program Memory Map

### 12.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F50x/F51x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip Flash memory space. MOVC instructions are always used to read Flash memory, while MOVX write instructions are used to erase and write Flash. This Flash access feature provides a mechanism for the C8051F50x/F51x to update program code and use the program memory space for non-volatile data storage. Refer to Section “15. Flash Memory” on page 129 for further details.

## 12.2. Data Memory

The C8051F50x/F51x devices include 4352 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. The other 4096 bytes of this memory is on-chip “external” memory. The data memory map is shown in Figure 12.1 for reference.

### 12.2.1. Internal RAM

There are 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 scratch pad 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 SFRs. 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 12.1 illustrates the data memory organization of the C8051F50x/F51x.

### 12.2.1.1. 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 11.6). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

### 12.2.1.2. 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 bit7 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.

### 12.2.1.3. 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) 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.

## SFR Definition 14.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
<b>Name</b>	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
<b>Type</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Reset</b>	0	0	0	0	0	0	0	0

SFR Address = 0xA8; Bit-Addressable; SFR Page = All Pages

Bit	Name	Function
7	EA	<p><b>Enable All Interrupts.</b>                      Globally enables/disables all interrupts. It overrides individual interrupt mask settings.                      0: Disable all interrupt sources.                      1: Enable each interrupt according to its individual mask setting.</p>
6	ESPI0	<p><b>Enable Serial Peripheral Interface (SPI0) Interrupt.</b>                      This bit sets the masking of the SPI0 interrupts.                      0: Disable all SPI0 interrupts.                      1: Enable interrupt requests generated by SPI0.</p>
5	ET2	<p><b>Enable Timer 2 Interrupt.</b>                      This bit sets the masking of the Timer 2 interrupt.                      0: Disable Timer 2 interrupt.                      1: Enable interrupt requests generated by the TF2L or TF2H flags.</p>
4	ES0	<p><b>Enable UART0 Interrupt.</b>                      This bit sets the masking of the UART0 interrupt.                      0: Disable UART0 interrupt.                      1: Enable UART0 interrupt.</p>
3	ET1	<p><b>Enable Timer 1 Interrupt.</b>                      This bit sets the masking of the Timer 1 interrupt.                      0: Disable all Timer 1 interrupt.                      1: Enable interrupt requests generated by the TF1 flag.</p>
2	EX1	<p><b>Enable External Interrupt 1.</b>                      This bit sets the masking of External Interrupt 1.                      0: Disable external interrupt 1.                      1: Enable interrupt requests generated by the <math>\overline{\text{INT1}}</math> input.</p>
1	ET0	<p><b>Enable Timer 0 Interrupt.</b>                      This bit sets the masking of the Timer 0 interrupt.                      0: Disable all Timer 0 interrupt.                      1: Enable interrupt requests generated by the TF0 flag.</p>
0	EX0	<p><b>Enable External Interrupt 0.</b>                      This bit sets the masking of External Interrupt 0.                      0: Disable external interrupt 0.                      1: Enable interrupt requests generated by the <math>\overline{\text{INT0}}</math> input.</p>

# C8051F50x/F51x

## SFR Definition 18.2. EMI0CF: External Memory Configuration

Bit	7	6	5	4	3	2	1	0
Name				EMD2	EMD[1:0]		EALE[1:0]	
Type	R/W							
Reset	0	0	0	0	0	0	1	1

SFR Address = 0xB2; SFR Page = 0x0F

Bit	Name	Function
7:5	Unused	Read = 000b; Write = Don't Care.
4	EMD2	<b>EMIF Multiplex Mode Select Bit.</b> 0: EMIF operates in multiplexed address/data mode 1: EMIF operates in non-multiplexed mode (separate address and data pins)
3:2	EMD[1:0]	<b>EMIF Operating Mode Select Bits.</b> 00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space 01: Split Mode without Bank Select: Accesses below the 4 kB boundary are directed on-chip. Accesses above the 4 kB boundary are directed off-chip. 8-bit off-chip MOVX operations use current contents of the Address high port latches to resolve the upper address byte. To access off chip space, EMI0CN must be set to a page that is not contained in the on-chip address space. 10: Split Mode with Bank Select: Accesses below the 4 kB boundary are directed on-chip. Accesses above the 4 kB boundary are directed off-chip. 8-bit off-chip MOVX operations uses the contents of EMI0CN to determine the high-byte of the address. 11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.
1:0	EALE[1:0]	<b>ALE Pulse-Width Select Bits.</b> These bits only have an effect when EMD2 = 0. 00: ALE high and ALE low pulse width = 1 SYCLK cycle. 01: ALE high and ALE low pulse width = 2 SYCLK cycles. 10: ALE high and ALE low pulse width = 3 SYCLK cycles. 11: ALE high and ALE low pulse width = 4 SYCLK cycles.

# C8051F50x/F51x

## 18.6.1. Non-Multiplexed Mode

### 18.6.1.1. 16-bit MOVX: EMI0CF[4:2] = 101, 110, or 111

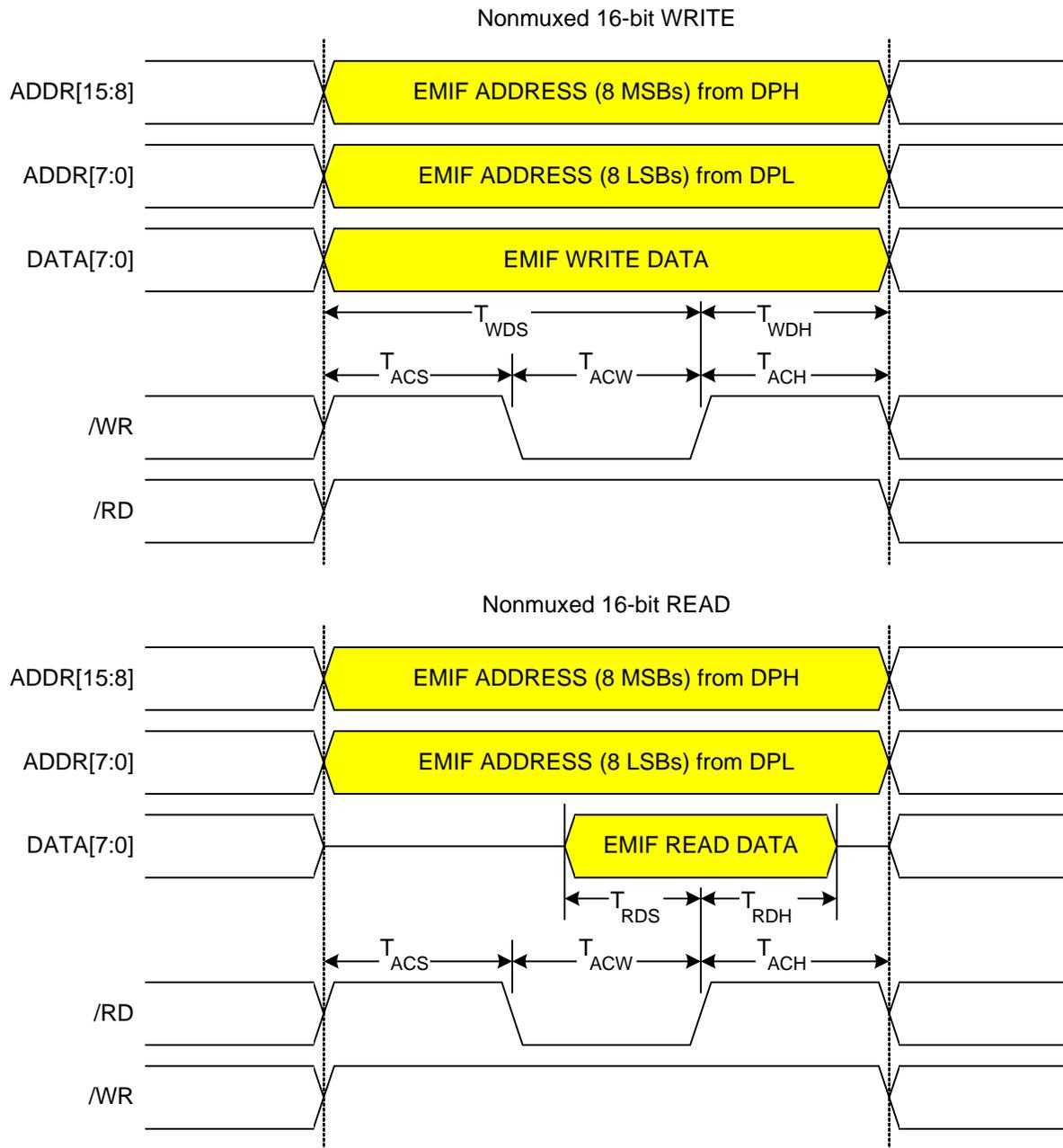


Figure 18.4. Non-multiplexed 16-bit MOVX Timing

---

20.13 for the PnMDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR2 is 0, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a 0 to avoid unnecessary power dissipation.

Registers XBR0, XBR1, and XBR2 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR2 to 1 enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility of the Silicon Labs IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. Port output drivers are disabled while the Crossbar is disabled.

## LIN Register Definition 21.6. LIN0ST: LIN0 Status Register

Bit	7	6	5	4	3	2	1	0
<b>Name</b>	ACTIVE	IDLTOUT	ABORT	DTREQ	LININT	ERROR	WAKEUP	DONE
<b>Type</b>	R	R	R	R	R	R	R	R
<b>Reset</b>	0	0	0	0	0	0	0	0

Indirect Address = 0x09

Bit	Name	Function
7	ACTIVE	<b>LIN Active Indicator Bit.</b> 0: No transmission activity detected on the LIN bus. 1: Transmission activity detected on the LIN bus.
6	IDLTOUT	<b>Bus Idle Timeout Bit. (slave mode only)</b> 0: The bus has not been idle for four seconds. 1: No bus activity has been detected for four seconds, but the bus is not yet in Sleep mode.
5	ABORT	<b>Aborted Transmission Bit. (slave mode only)</b> 0: The current transmission has not been interrupted or stopped. This bit is reset to 0 after receiving a SYNCH BREAK that does not interrupt a pending transmission. 1: New SYNCH BREAK detected before the end of the last transmission or the STOP bit (LINOCTRL.7) has been set.
4	DTREQ	<b>Data Request Bit. (slave mode only)</b> 0: Data identifier has not been received. 1: Data identifier has been received.
3	LININT	<b>Interrupt Request Bit.</b> 0: An interrupt is not pending. This bit is cleared by setting RSTINT (LINOCTRL.3) 1: There is a pending LIN0 interrupt.
2	ERROR	<b>Communication Error Bit.</b> 0: No error has been detected. This bit is cleared by setting RSTERR (LINOCTRL.2) 1: An error has been detected.
1	WAKEUP	<b>Wakeup Bit.</b> 0: A wakeup signal is not being transmitted and has not been received. 1: A wakeup signal is being transmitted or has been received
0	DONE	<b>Transmission Complete Bit.</b> 0: A transmission is not in progress or has not been started. This bit is cleared at the start of a transmission. 1: The current transmission is complete.

# C8051F50x/F51x

## 22.1. Bosch CAN Controller Operation

The CAN Controller featured in the C8051F500/2/4/6/8-F510 devices is a full implementation of Bosch's full CAN module and fully complies with CAN specification 2.0B. A block diagram of the CAN controller is shown in Figure 22.2. The CAN Core provides shifting (CANTX and CANRX), serial/parallel conversion of messages, and other protocol related tasks such as transmission of data and acceptance filtering. The message RAM stores 32 message objects which can be received or transmitted on a CAN network. The CAN registers and message handler provide an interface for data transfer and notification between the CAN controller and the CIP-51.

The function and use of the CAN Controller is detailed in the Bosch CAN User's Guide. The User's Guide should be used as a reference to configure and use the CAN controller. This data sheet describes how to access the CAN controller.

All of the CAN controller registers are located on SFR Page 0x0C. Before accessing any of the CAN registers, the SFRPAGE register must be set to 0x0C.

The CAN Controller is typically initialized using the following steps:

1. Set the SFRPAGE register to the CAN registers page (page 0x0C).
2. Set the INIT and the CCE bits to 1 in CAN0CN. See the CAN User's Guide for bit definitions.
3. Set timing parameters in the Bit Timing Register and the BRP Extension Register.
4. Initialize each message object or set its MsgVal bit to NOT VALID.
5. Reset the INIT bit to 0.

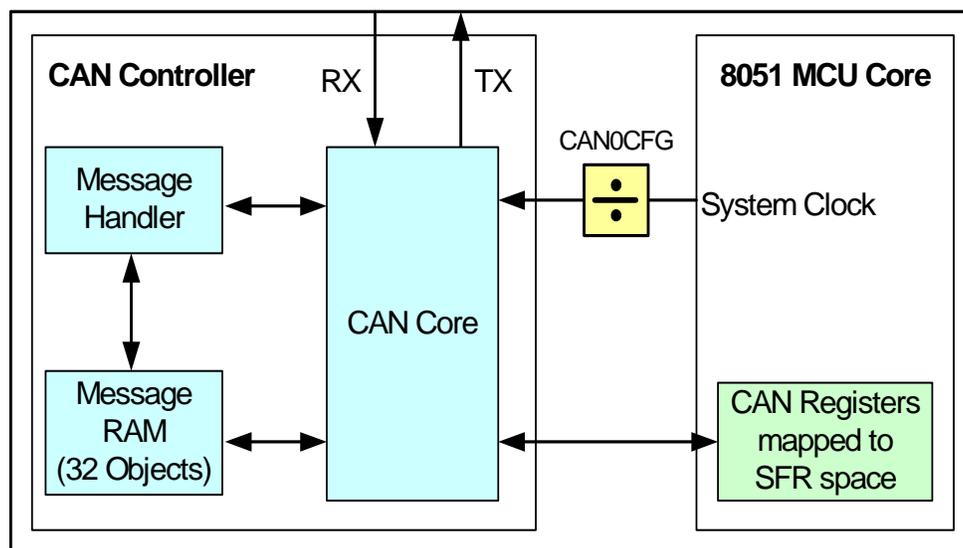


Figure 22.2. CAN Controller Diagram

### 22.1.1. CAN Controller Timing

The CAN controller's clock ( $f_{sys}$ ) is derived from the CIP-51 system clock (SYSCLK). The internal oscillator is accurate to within 0.5% of 24 MHz across the entire temperature range and for VDD voltages greater than or equal to the minimum output of the on-chip voltage regulator, and so an external oscillator is not required for CAN communication for most systems. Refer to Section "4.10.4 Oscillator Tolerance Range" in the Bosch CAN User's Guide for further information regarding this topic.

# C8051F50x/F51x

## SFR Definition 24.3. SBUF0: Serial (UART0) Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x99; SFR Page = 0x00

Bit	Name	Function
7:0	SBUF0[7:0]	<p><b>Serial Data Buffer Bits 7–0 (MSB–LSB).</b></p> <p>This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.</p>

## SFR Definition 24.4. SB0CON0: UART0 Baud Rate Generator Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved	SB0RUN	Reserved	Reserved	Reserved	Reserved	SB0PS[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAB; SFR Page = 0x0F

Bit	Name	Function
7	Reserved	Read = 0b; Must Write 0b;
6	SB0RUN	<p><b>Baud Rate Generator Enable.</b></p> <p>0: Baud Rate Generator disabled. UART0 will not function. 1: Baud Rate Generator enabled.</p>
5:2	Reserved	Read = 0000b; Must Write = 0000b;
1:0	SB0PS[1:0]	<p><b>Baud Rate Prescaler Select.</b></p> <p>00: Prescaler = 12. 01: Prescaler = 4. 10: Prescaler = 48. 11: Prescaler = 1.</p>

# C8051F50x/F51x

---

## 25.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

### 25.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

### 25.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

### 25.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

### 25.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 25.2, Figure 25.3, and Figure 25.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “20. Port Input/Output” on page 177 for general purpose port I/O and crossbar information.

# C8051F50x/F51x

## SFR Definition 25.3. SPI0CKR: SPI0 Clock Rate

Bit	7	6	5	4	3	2	1	0
Name	SCR[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA2; SFR Page = 0x00

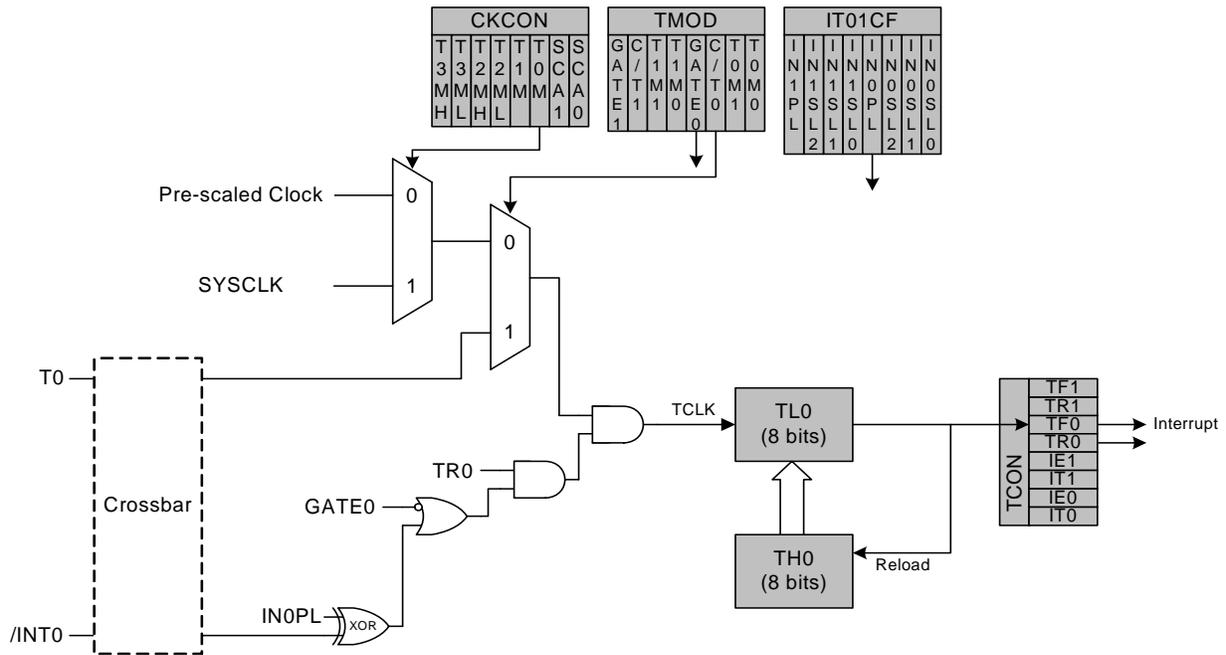
Bit	Name	Function
7:0	SCR[7:0]	<p><b>SPI0 Clock Rate.</b></p> <p>These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where SYSCLK is the system clock frequency and SPI0CKR is the 8-bit value held in the SPI0CKR register.</p> $f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR[7:0] + 1)}$ <p>for <math>0 \leq SPI0CKR \leq 255</math></p> <p>Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,</p> $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200 \text{ kHz}$

## SFR Definition 25.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA3; SFR Page = 0x00

Bit	Name	Function
7:0	SPI0DAT[7:0]	<p><b>SPI0 Transmit and Receive Data.</b></p> <p>The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.</p>



**Figure 26.2. T0 Mode 2 Block Diagram**

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

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. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled 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. When Timer 0 is operating 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 to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

**Table 27.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules**

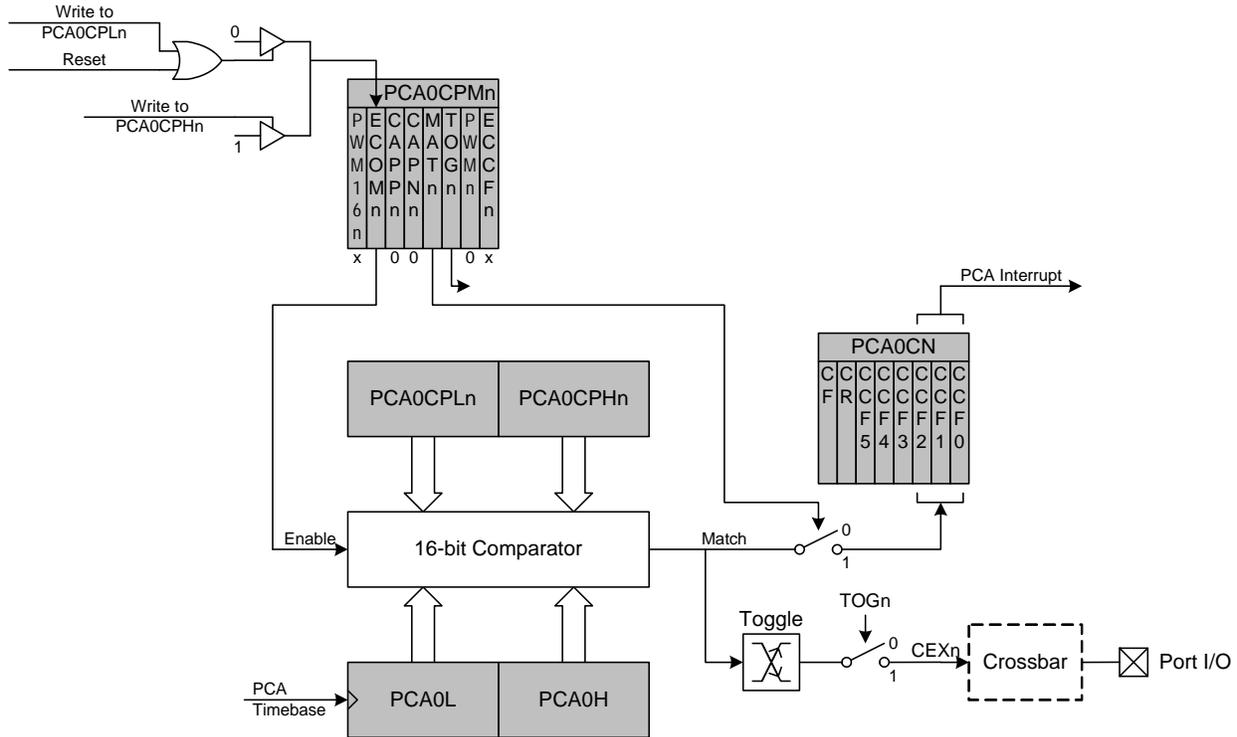
Operational Mode Bit Number	PCA0CPMn								PCA0PWM				
	7	6	5	4	3	2	1	0	7	6	5	4-2	1-0
Capture triggered by positive edge on CEXn	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
Capture triggered by negative edge on CEXn	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
Capture triggered by any transition on CEXn	X	X	1	1	0	0	0	A	0	X	B	XXX	XX
Software Timer	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
High Speed Output	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
Frequency Output	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16-Bit Pulse Width Modulator	1	C	0	0	E	0	1	A	0	X	B	XXX	XX

**Notes:**

1. X = Don't Care (no functional difference for individual module if 1 or 0).
2. A = 1 to enable interrupts for this module (PCA interrupt triggered on CCFn set to 1).
3. B = 1 to enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL[1:0]).
4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0).
5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA0CPHn and PCA0CPLn.
6. E = When set to 1, a match event will cause the CCFn flag for the associated channel to be set.
7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting.

### 27.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.



**Figure 27.6. PCA High-Speed Output Mode Diagram**

### 27.3.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 27.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

#### Equation 27.1. Square Wave Frequency Output

Where  $F_{PCA}$  is the frequency of the clock selected by the CPS[2:0] bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.