



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	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, WDT
Number of I/O	54
Program Memory Size	15KB (15K x 8)
Program Memory Type	FLASH
EEPROM Size	32 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f701-gqr

Table of Contents

1. System Overview	17
2. Ordering Information	26
3. Pin Definitions.....	28
4. TQFP-64 Package Specifications	37
5. TQFP-48 Package Specifications	39
6. QFN-48 Package Specifications	41
7. QFN-32 Package Specifications	43
8. QFN-24 Package Specifications	45
9. Electrical Characteristics	47
9.1. Absolute Maximum Specifications.....	47
9.2. Electrical Characteristics	48
10. 10-Bit ADC (ADC0)	55
10.1. Output Code Formatting	56
10.2. 8-Bit Mode	56
10.3. Modes of Operation	56
10.3.1. Starting a Conversion.....	56
10.3.2. Tracking Modes.....	57
10.3.3. Settling Time Requirements.....	58
10.4. Programmable Window Detector.....	62
10.4.1. Window Detector Example.....	64
10.5. ADC0 Analog Multiplexer	65
11. Temperature Sensor	67
11.1. Calibration	67
12. Voltage and Ground Reference Options.....	69
12.1. External Voltage References.....	70
12.2. Internal Voltage Reference Options	70
12.3. Analog Ground Reference.....	70
12.4. Temperature Sensor Enable	70
13. Voltage Regulator (REG0)	72
14. Comparator0.....	74
14.1. Comparator Multiplexer	78
15. Capacitive Sense (CS0).....	80
15.1. Configuring Port Pins as Capacitive Sense Inputs	81
15.2. CS0 Gain Adjustment	81
15.3. Capacitive Sense Start-Of-Conversion Sources	81
15.4. Automatic Scanning.....	83
15.5. CS0 Comparator.....	84
15.6. CS0 Conversion Accumulator	85
15.7. CS0 Pin Monitor	86
15.8. Adjusting CS0 For Special Situations.....	87
15.9. Capacitive Sense Multiplexer	96
16. CIP-51 Microcontroller.....	98
16.1. Instruction Set.....	99

List of Registers

SFR Definition 10.1. ADC0CF: ADC0 Configuration	59
SFR Definition 10.2. ADC0H: ADC0 Data Word MSB	60
SFR Definition 10.3. ADC0L: ADC0 Data Word LSB	60
SFR Definition 10.4. ADC0CN: ADC0 Control	61
SFR Definition 10.5. ADC0GTH: ADC0 Greater-Than Data High Byte	62
SFR Definition 10.6. ADC0GTL: ADC0 Greater-Than Data Low Byte	62
SFR Definition 10.7. ADC0LTH: ADC0 Less-Than Data High Byte	63
SFR Definition 10.8. ADC0LTL: ADC0 Less-Than Data Low Byte	63
SFR Definition 10.9. ADC0MX: AMUX0 Channel Select	66
SFR Definition 12.1. REF0CN: Voltage Reference Control	71
SFR Definition 13.1. REG0CN: Voltage Regulator Control	73
SFR Definition 14.1. CPT0CN: Comparator0 Control	76
SFR Definition 14.2. CPT0MD: Comparator0 Mode Selection	77
SFR Definition 14.3. CPT0MX: Comparator0 MUX Selection	79
SFR Definition 15.1. CS0CN: Capacitive Sense Control	88
SFR Definition 15.2. CS0CF: Capacitive Sense Configuration	89
SFR Definition 15.3. CS0DH: Capacitive Sense Data High Byte	90
SFR Definition 15.4. CS0DL: Capacitive Sense Data Low Byte	90
SFR Definition 15.5. CS0SS: Capacitive Sense Auto-Scan Start Channel	91
SFR Definition 15.6. CS0SE: Capacitive Sense Auto-Scan End Channel	91
SFR Definition 15.7. CS0THH: Capacitive Sense Comparator Threshold High Byte ...	92
SFR Definition 15.8. CS0THL: Capacitive Sense Comparator Threshold Low Byte	92
SFR Definition 15.9. CS0PM: Capacitive Sense Pin Monitor	93
SFR Definition 15.10. CS0MD1: Capacitive Sense Mode 1	94
SFR Definition 15.11. CS0MD2: Capacitive Sense Mode 2	95
SFR Definition 15.12. CS0MX: Capacitive Sense Mux Channel Select	97
SFR Definition 16.1. DPL: Data Pointer Low Byte	104
SFR Definition 16.2. DPH: Data Pointer High Byte	104
SFR Definition 16.3. SP: Stack Pointer	105
SFR Definition 16.4. ACC: Accumulator	105
SFR Definition 16.5. B: B Register	106
SFR Definition 16.6. PSW: Program Status Word	107
SFR Definition 18.1. EMI0CN: External Memory Interface Control	113
SFR Definition 18.2. EMI0CF: External Memory Configuration	114
SFR Definition 18.3. EMI0TC: External Memory Timing Control	119
SFR Definition 19.1. HWID: Hardware Identification Byte	128
SFR Definition 19.2. DERIVID: Derivative Identification Byte	128
SFR Definition 19.3. REVID: Hardware Revision Identification Byte	129
SFR Definition 20.1. SFRPAGE: SFR Page	132
SFR Definition 21.1. IE: Interrupt Enable	140
SFR Definition 21.2. IP: Interrupt Priority	141
SFR Definition 21.3. EIE1: Extended Interrupt Enable 1	142
SFR Definition 21.4. EIE2: Extended Interrupt Enable 2	143

C8051F70x/71x

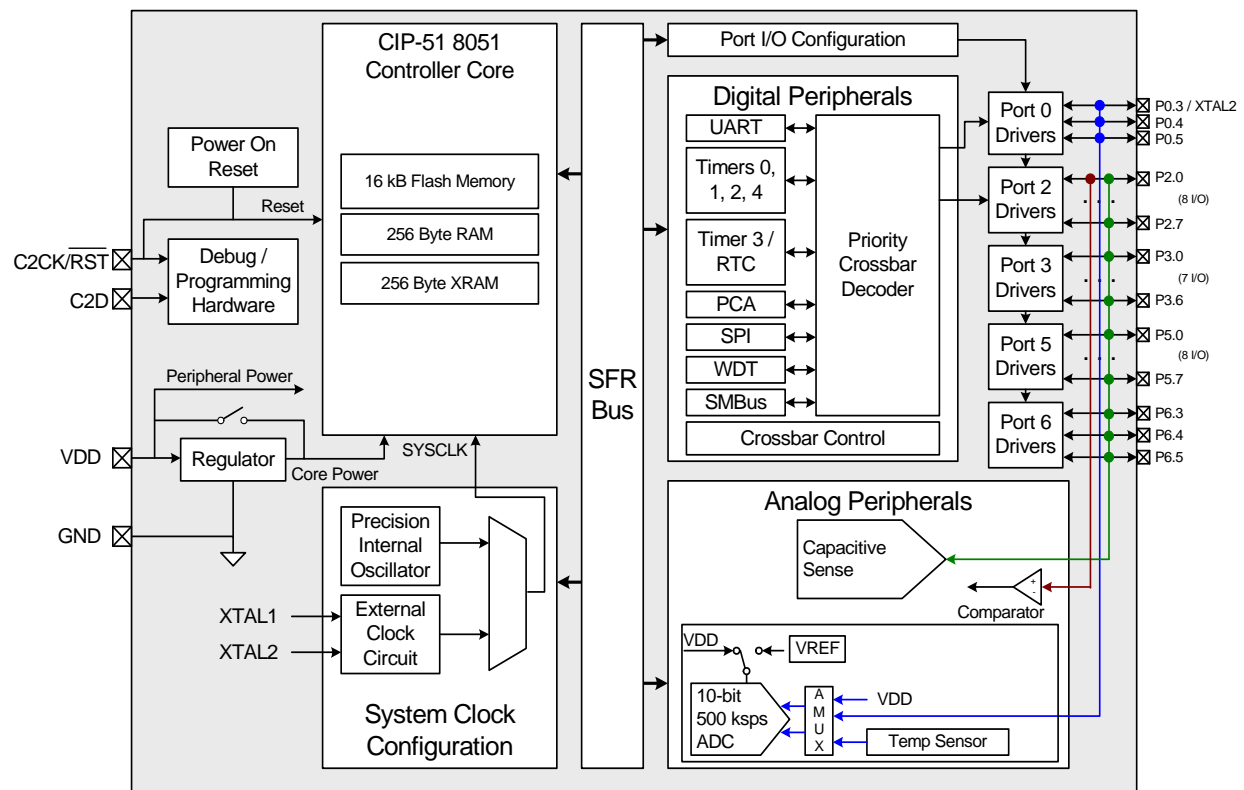


Figure 1.7. C8051F716 Block Diagram

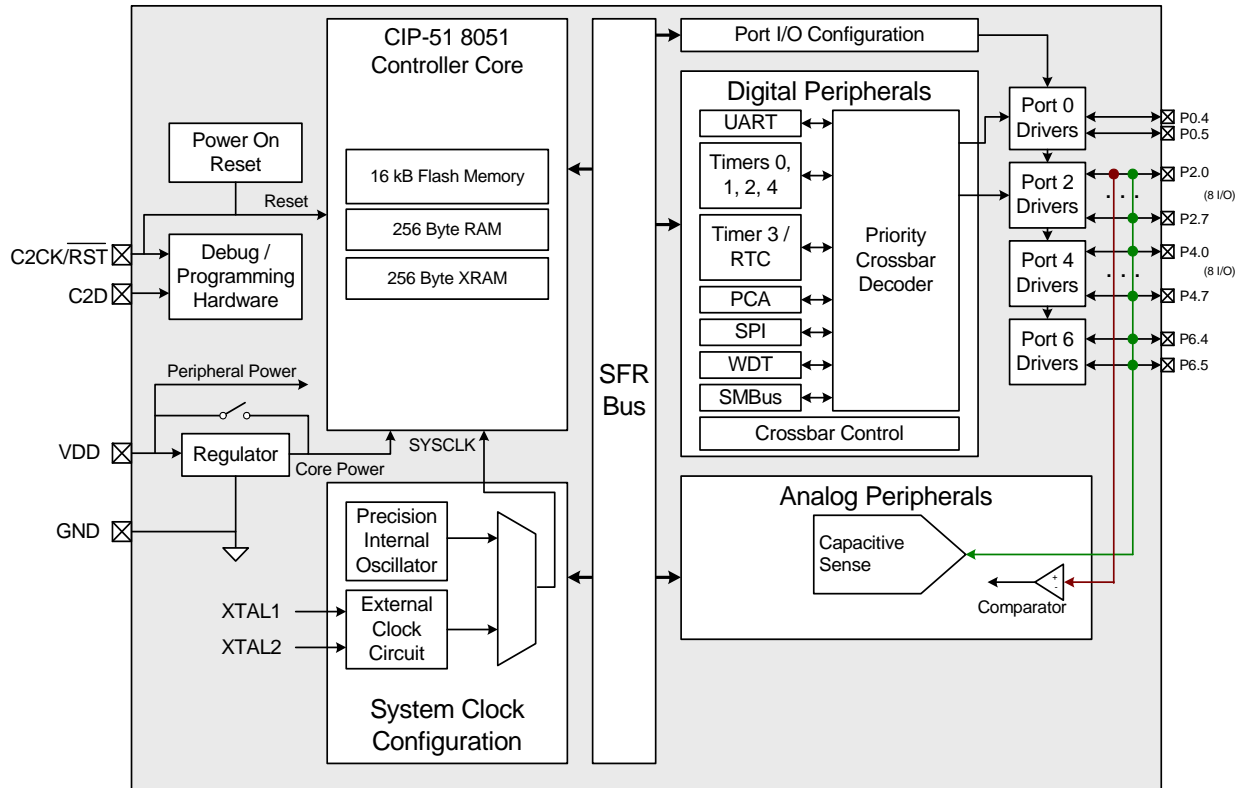


Figure 1.8. C8051F717 Block Diagram

6. QFN-48 Package Specifications

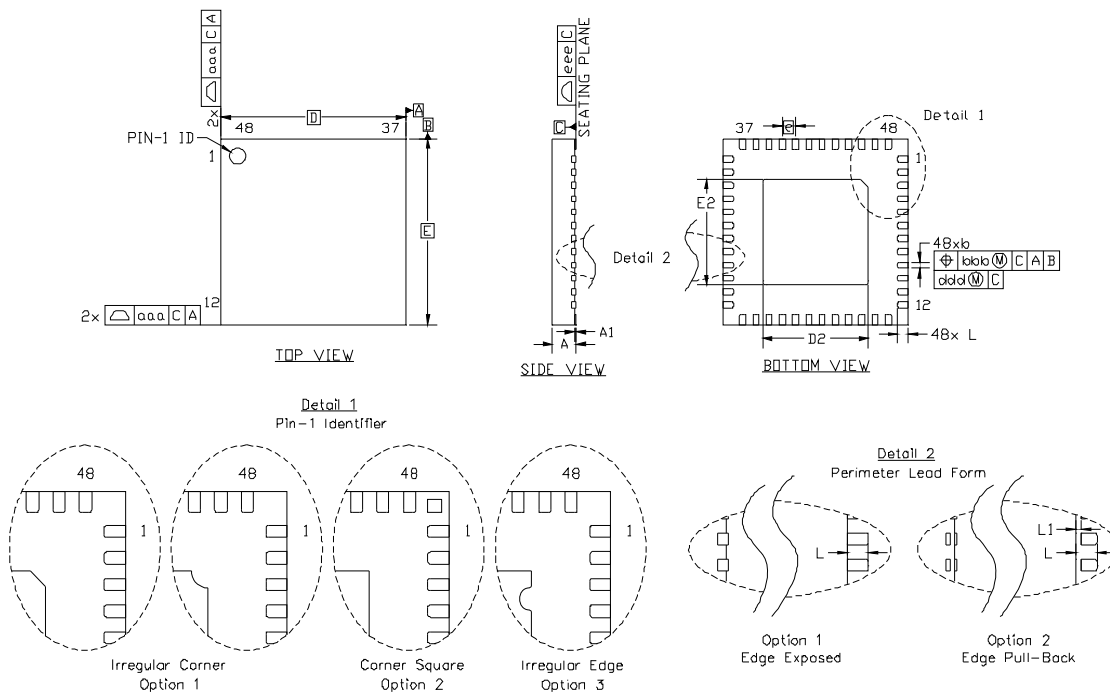


Figure 6.1. QFN-48 Package Drawing

Table 6.1. QFN-48 Package Dimensions

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	0.80	0.90	1.00	E2	3.90	4.00	4.10
A1	0.00	—	0.05	L	0.30	0.40	0.50
b	0.18	0.23	0.30	L1	0.00	—	0.10
D	7.00 BSC.			aaa	—	—	0.10
D2	3.90	4.00	4.10	bbb	—	—	0.10
e	0.50 BSC.			ccc	—	—	0.05
E	7.00 BSC.			ddd	—	—	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 outline MO-220, variation VKKD-4 except for features D2 and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

13. Voltage Regulator (REG0)

C8051F70x/71x devices include an internal voltage regulator (REG0) to regulate the internal core supply to 1.8 V from a V_{DD} supply of 1.8 to 3.6 V. Two power-saving modes are built into the regulator to help reduce current consumption in low-power applications. These modes are accessed through the REG0CN register (SFR Definition 13.1). Electrical characteristics for the on-chip regulator are specified in Table 9.5 on page 50

If an external regulator is used to power the device, the internal regulator may be put into bypass mode using the BYPASS bit. **The internal regulator should never be placed in bypass mode unless an external 1.8 V regulator is used to supply V_{DD} . Doing so could cause permanent damage to the device.**

Under default conditions, when the device enters STOP mode the internal regulator will remain on. This allows any enabled reset source to generate a reset for the device and bring the device out of STOP mode. For additional power savings, the STOPCF bit can be used to shut down the regulator and the internal power network of the device when the part enters STOP mode. When STOPCF is set to 1, the RST pin or a full power cycle of the device are the only methods of generating a reset.

14. Comparator0

C8051F70x/71x devices include an on-chip programmable voltage comparator, Comparator0, shown in Figure 14.1.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0), or an asynchronous “raw” output (CP0A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see Section “28.4. Port I/O Initialization” on page 189). Comparator0 may also be used as a reset source (see Section “25.5. Comparator0 Reset” on page 167).

The Comparator0 inputs are selected by the comparator input multiplexer, as detailed in Section “14.1. Comparator Multiplexer” on page 78.

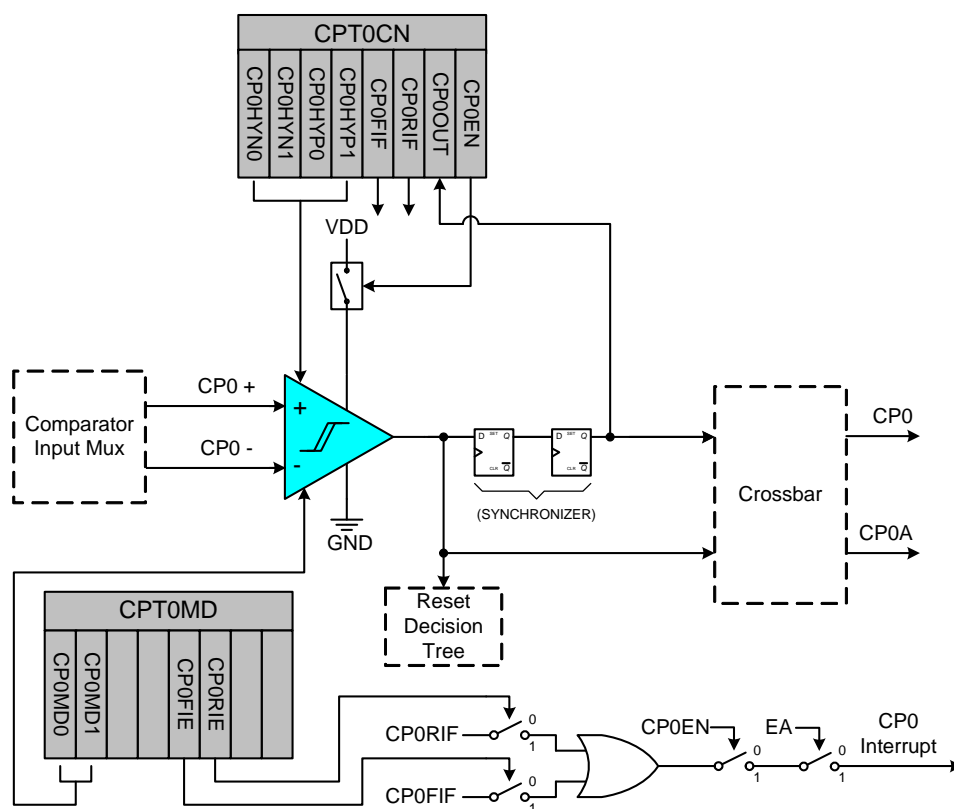


Figure 14.1. Comparator0 Functional Block Diagram

The Comparator output can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the digital Crossbar) defaults to the logic low state, and the power supply to the comparator is turned off. See Section “28.3. Priority Crossbar Decoder” on page 185 for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from -0.25 V to $(V_{DD}) + 0.25\text{ V}$ without damage or upset. The complete Comparator electrical specifications are given in Section “9. Electrical Characteristics” on page 47.

15.4. Automatic Scanning

CS0 can be configured to automatically scan a sequence of contiguous CS0 input channels by configuring and enabling auto-scan. Using auto-scan with the CS0 comparator interrupt enabled allows a system to detect a change in measured capacitance without requiring any additional dedicated MCU resources.

Auto-scan is enabled by setting the CS0 start-of-conversion bits (CS0CF6:4) to 111b. After enabling auto-scan, the starting and ending channels should be set to appropriate values in CS0SS and CS0SE, respectively. Writing to CS0SS when auto-scan is enabled will cause the value written to CS0SS to be copied into CS0MX. After being enabled, writing a 1 to CS0BUSY will start auto-scan conversions. When auto-scan completes the number of conversions defined in the CS0 accumulator bits (CS0CF2:0), auto-scan configures CS0MX to the next sequential port pin configured as an analog input and begins a conversion on that channel. The scan sequence continues until CS0MX reaches the ending input channel value defined in CS0SE.

Note: All other CS0 pins configured for analog input with a 0 in the port latch are grounded during the conversion.

After the final channel conversion, auto-scan configures CS0MX back to the starting input channel. For an example system configured to use auto-scan, please see Figure “15.2 Auto-Scan Example” on page 83.

Note: Auto-scan attempts one conversion on a CS0MX channel regardless of whether that channel's port pin has been configured as an analog input. Auto-scan will also complete the current rotation when the device is halted for debugging.

If auto-scan is enabled when the device enters suspend mode, auto-scan will remain enabled and running. This feature allows the device to wake from suspend through CS0 greater-than comparator event on any configured capacitive sense input included in the auto-scan sequence of inputs.

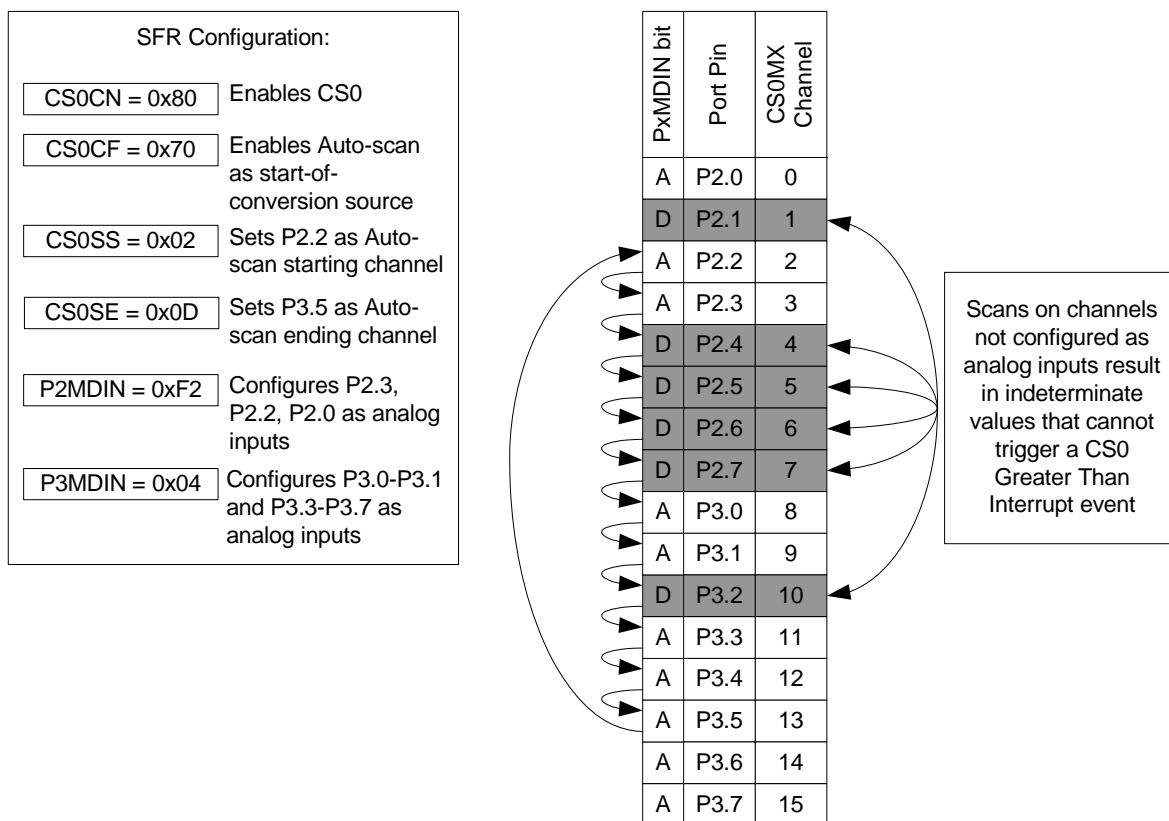


Figure 15.2. Auto-Scan Example

17. 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. The memory organization of the C8051F70x/71x device family is shown in Figure 17.1

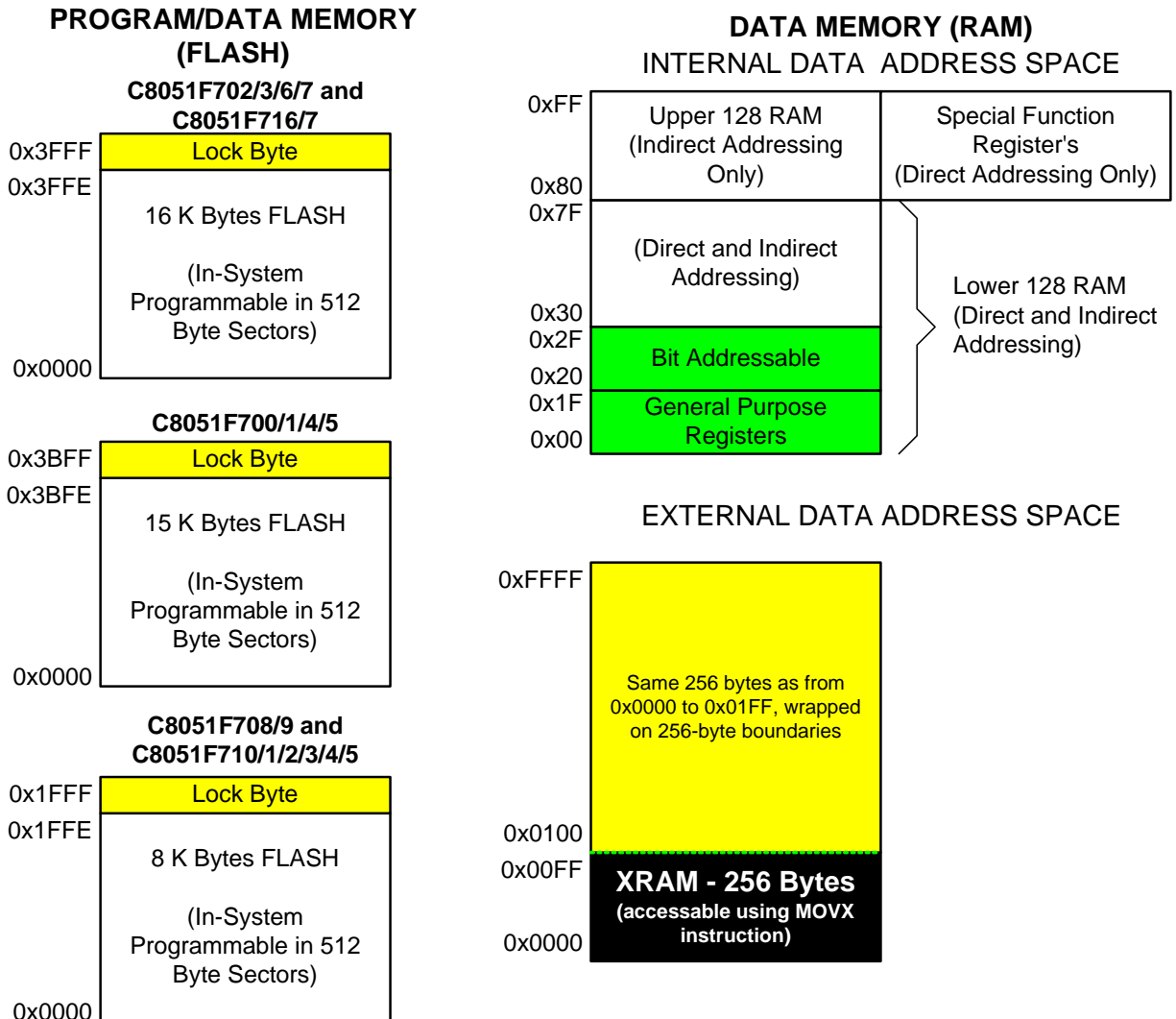


Figure 17.1. C8051F70x/71x Memory Map

18.6.1.3. 8-bit MOVX with Bank Select: EMI0CF[4:2] = 110

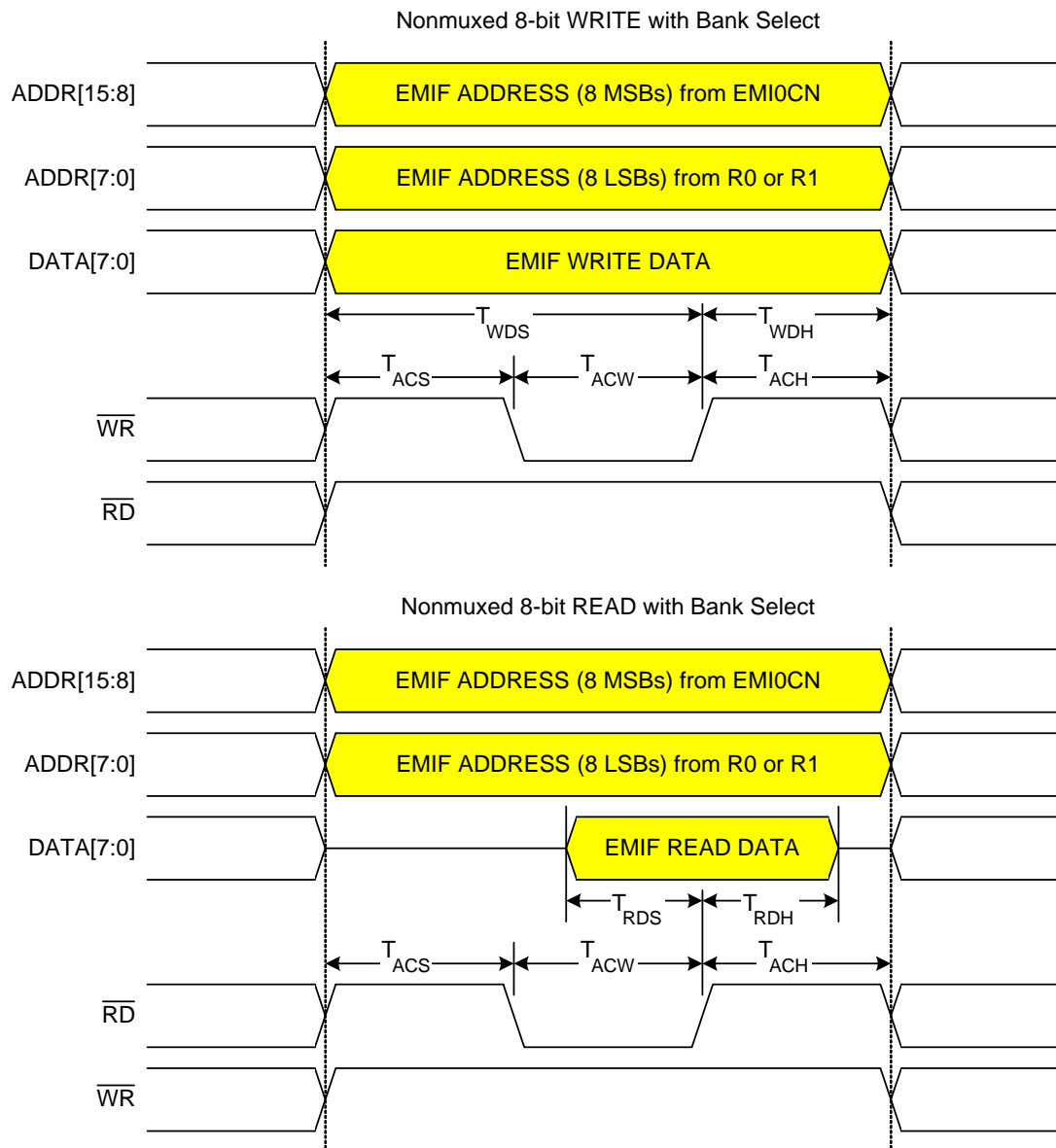


Figure 18.6. Non-Multiplexed 8-Bit MOVX with Bank Select Timing

Table 20.1. Special Function Register (SFR) Memory Map

Addr	SFR Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0 F	SPI0CN	PCA0L P0DRV	PCA0H P1DRV	PCA0CPL0 P2DRV	PCA0CPH0 P3DRV	P4DRV	P5DRV	VDM0CN
F0	0 F	B	P0MDIN	P1MDIN	P0MAT P2MDIN	P0MASK P3MDIN	P4MDIN	P5MDIN	P6MDIN
E8	0 F	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2 DERIVID	PCA0MD	EMI0TC	RSTSRC
E0	0 F	ACC	P1MAT XBR0	P1MASK XBR1	WDTCN	IT01CF		EIE1	EIE2
D8	0 F	PCA0CN	CRC0DATA	PCA0CPM0	PCA0CPM1	PCA0CPM2			
D0	0 F	PSW	EEDATA	REF0CN		P0SKIP	P1SKIP	P2SKIP	
C8	0 F	TMR2CN		TMR2RLL	TMR2RLH	TMR2L	TMR2H	EIP1	EIP2
C0	0 F	SMB0CN	SMB0CF P6DRV	SMB0DAT	ADC0GTL	ADC0GTH HWID	ADC0LTL EECNTL	ADC0LTH EEKEY	EMI0CF
B8	0 F	IP	REG0CN	SMB0ADR	ADC0MX SMB0ADM	ADC0CF	ADC0L CLKSEL	ADC0H CS0MD2	OSCICL
B0	0 F	P3		P6	P5		OSCXCN	EEADDR	FLKEY
A8	0 F	IE	CS0DL OSCICN	CS0DH EMI0CN		P4	CS0MD1 REVID		P3MDOUT
A0	0 F	P2	SPI0CFG PCA0PWM	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	SFRPAGE
98	0 F	SCON0	SBUF0	CS0CN P4MDOUT	CPT0CN P5MDOUT	CS0MX P6MDOUT	CPT0MD	CS0CF	CPT0MX CS0PM
90	0 F	P1	TMR3CN CRC0CN	TMR3RLL CS0SS	TMR3RLH CS0SE	TMR3L CRC0IN	TMR3H CRC0FLIP	CS0THL CRC0AUTO	CS0THH CRC0CNT
88	0 F	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	0 F	P0	SP	DPL	DPH				PCON
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Notes:

1. SFR addresses ending in 0x0 or 0x8 (leftmost column) are bit-addressable.
2. SFRs indicated with bold lettering and shaded cells are available on both SFR Page 0 and F.

25.2. Power-Fail Reset / V_{DD} Monitor

When a power-down transition or power irregularity causes V_{DD} to drop below V_{RST} , the power supply monitor will drive the \overline{RST} pin low and hold the CIP-51 in a reset state (see Figure 25.2). When V_{DD} returns to a level above V_{RST} , the CIP-51 will be released from the reset state. 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 reads 1, the data may no longer be valid. The V_{DD} monitor is enabled after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} monitor is disabled by code and a software reset is performed, the V_{DD} monitor will still be disabled after the reset.

Important Note: If the V_{DD} monitor is being turned on from a disabled state, it should be enabled before it is selected as a reset source. Selecting the V_{DD} monitor as a reset source before it is enabled and stabilized may cause a system reset. In some applications, this reset may be undesirable. If this is not desirable in the application, a delay should be introduced between enabling the monitor and selecting it as a reset source. The procedure for enabling the V_{DD} monitor and configuring it as a reset source from a disabled state is shown below:

1. Enable the V_{DD} monitor (VDMEN bit in VDM0CN = 1).
2. If necessary, wait for the V_{DD} monitor to stabilize.
3. Select the V_{DD} monitor as a reset source (PORSF bit in RSTSRC = 1).

See Figure 25.2 for V_{DD} monitor timing; note that the power-on-reset delay is not incurred after a V_{DD} monitor reset. See Section “9. Electrical Characteristics” on page 47 for complete electrical characteristics of the V_{DD} monitor.

SFR Definition 25.2. RSTSRC: Reset Source

Bit	7	6	5	4	3	2	1	0
Name		FERROR	CORSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF
Type	R	R	R/W	R/W	R	R/W	R/W	R
Reset	0	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xEF; SFR Page = All Pages

Bit	Name	Description	Write	Read
7	Unused	Unused.	Don't care.	0
6	FERROR	Flash Error Reset Flag.	N/A	Set to 1 if Flash read/write/erase error caused the last reset.
5	CORSEF	Comparator0 Reset Enable and Flag.	Writing a 1 enables Comparator0 as a reset source (active-low).	Set to 1 if Comparator0 caused the last reset.
4	SWRSF	Software Reset Force and Flag.	Writing a 1 forces a system reset.	Set to 1 if last reset was caused by a write to SWRSF.
3	WDTRSF	Watchdog Timer Reset Flag.	N/A	Set to 1 if Watchdog Timer overflow caused the last reset.
2	MCDRSF	Missing Clock Detector Enable and Flag.	Writing a 1 enables the Missing Clock Detector. The MCD triggers a reset if a missing clock condition is detected.	Set to 1 if Missing Clock Detector timeout caused the last reset.
1	PORSF	Power-On / V_{DD} Monitor Reset Flag, and V_{DD} monitor Reset Enable.	Writing a 1 enables the V _{DD} monitor as a reset source. Writing 1 to this bit before the V_{DD} monitor is enabled and stabilized may cause a system reset.	Set to 1 anytime a power-on or V _{DD} monitor reset occurs. When set to 1 all other RSTSRC flags are indeterminate.
0	PINRSF	HW Pin Reset Flag.	N/A	Set to 1 if RST pin caused the last reset.

Note: Do not use read-modify-write operations on this register

SFR Definition 28.13. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDIN[7:0]							
Type	R/W							
Reset	1*	1	1	1	1	1	1	1

SFR Address = 0xF2; SFR Page = F

Bit	Name	Function
7:0	P1MDIN[7:0]	Analog Configuration Bits for P1.7–P1.0 (respectively). Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.
Note: On C8051F716 and C8051F717 devices, P1.7 will default to analog mode. If the P1MDIN register is written on the C8051F716 and C8051F717 devices, P1.7 should always be configured as analog.		

SFR Definition 28.14. P1MDOUT: Port 1 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA5; SFR Page = F

Bit	Name	Function
7:0	P1MDOUT[7:0]	Output Configuration Bits for P1.7–P1.0 (respectively). These bits are ignored if the corresponding bit in register P1MDIN is logic 0. 0: Corresponding P1.n Output is open-drain. 1: Corresponding P1.n Output is push-pull.

overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

30.3.5. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more than 50 μ s, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. A clock source is required for free timeout detection, even in a slave-only implementation.

30.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information
- Optional hardware recognition of slave address and automatic acknowledgement of address/data

SMBus interrupts are generated for each data byte or slave address that is transferred. When hardware acknowledgement is disabled, the point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e., sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e., receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. If hardware acknowledgement is enabled, these interrupts are always generated after the ACK cycle. See Section 30.5 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in Section 30.4.2; Table 30.5 provides a quick SMB0CN decoding reference.

30.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

30.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0-DAT is written while an active Master Receiver. Figure 30.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.

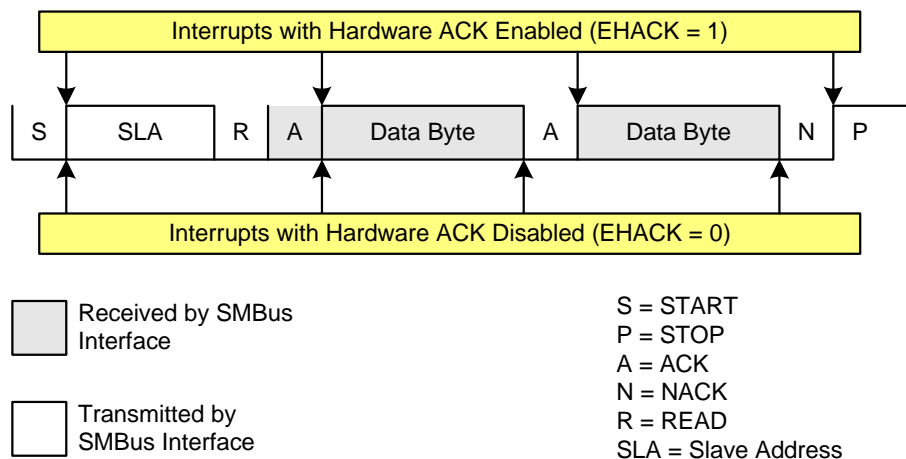


Figure 30.6. Typical Master Read Sequence

30.5.4. Read Sequence (Slave)

During a read sequence, an SMBus master reads data from a slave device. The slave in this transfer will be a receiver during the address byte, and a transmitter during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode (to receive the slave address) when a START followed by a slave address and direction bit (READ in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are transmitted. If the received slave address is acknowledged, data should be written to SMB0DAT to be transmitted. The interface enters slave transmitter mode, and transmits one or more bytes of data. After each byte is transmitted, the master sends an acknowledge bit; if the acknowledge bit is an ACK, SMB0DAT should be written with the next data byte. If the acknowledge bit is a NACK, SMB0DAT should not be written to before SI is cleared (an error condition may be generated if SMB0DAT is written following a received NACK while in slave transmitter mode). The interface exits slave transmitter mode after receiving a STOP. The interface will switch to slave receiver mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 30.8 shows a typical slave read sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

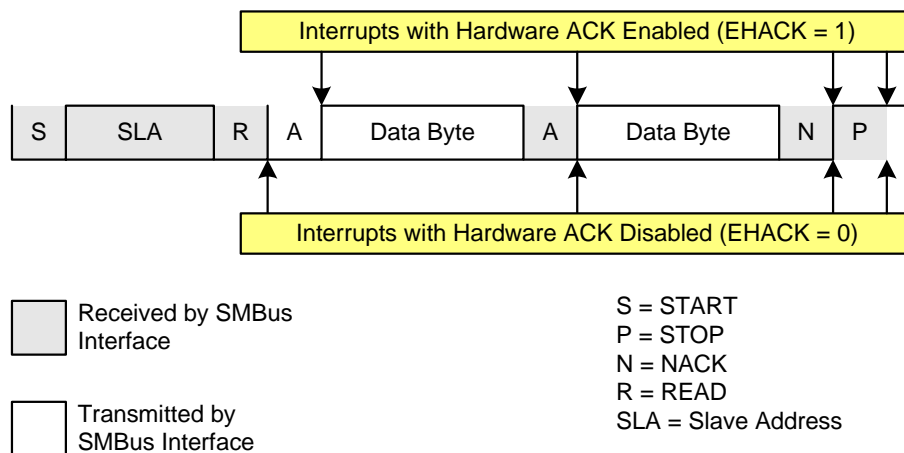


Figure 30.8. Typical Slave Read Sequence

30.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. The appropriate actions to take in response to an SMBus event depend on whether hardware slave address recognition and ACK generation is enabled or disabled. Table 30.5 describes the typical actions when hardware slave address recognition and ACK generation is disabled. Table 30.6 describes the typical actions when hardware slave address recognition and ACK generation is enabled. In the tables, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed by hardware but do not conform to the SMBus specification.

32.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

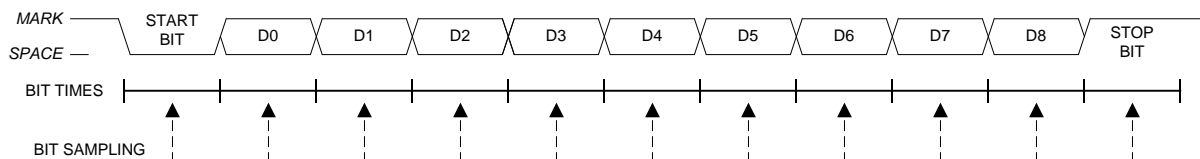


Figure 32.5. 9-Bit UART Timing Diagram

33.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 33.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode. Timer 3 can also be used in capture mode to capture rising edges of the Comparator 0 output.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 clock select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bits (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

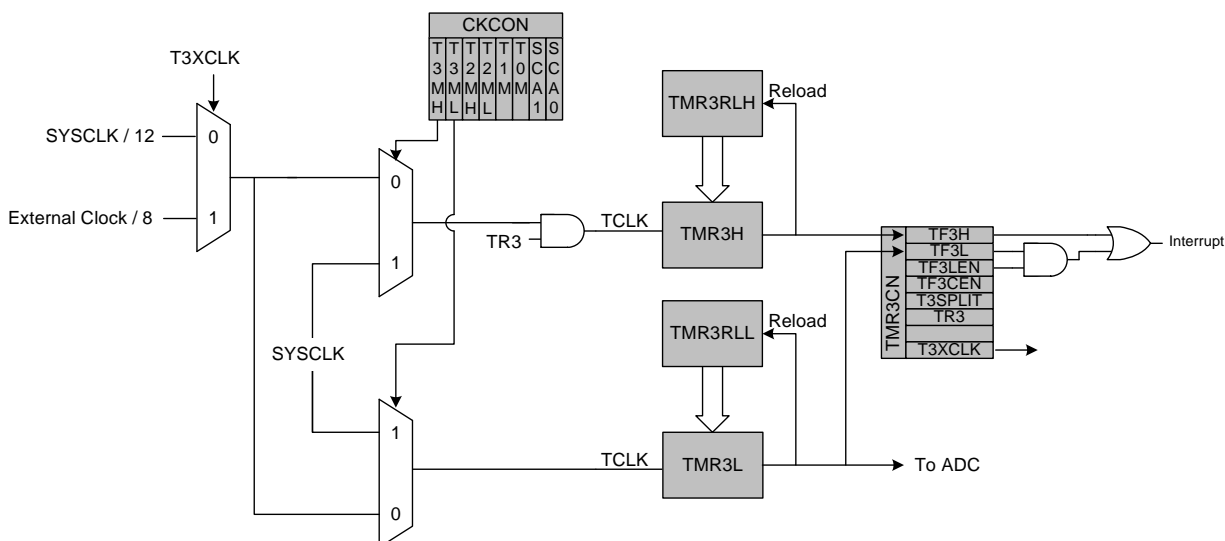


Figure 33.8. Timer 3 8-Bit Mode Block Diagram

C8051F70x/71x

34.4. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 34.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR				CCF2	CCF1	CCF0
Type	R/W	R/W	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD8; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	CF	PCA Counter/Timer Overflow Flag. Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.
6	CR	PCA Counter/Timer Run Control. This bit enables/disables the PCA Counter/Timer. 0: PCA Counter/Timer disabled. 1: PCA Counter/Timer enabled.
5:3	Unused	Read = 000b; Write = Don't care
2:0	CCF[2:0]	PCA Module n Capture/Compare Flag. These bits are set by hardware when a match or capture occurs in the associated PCA Module n. When the CCFn interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.