



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	48 MIPS
Connectivity	I ² C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f383-gmr

3. Pinout and Package Definitions

Table 3.1. Pin Definitions for the C8051F380/1/2/3/4/5/6/7/C

Name	Pin Numbers		Type	Description
	48-pin	32-pin		
V _{DD}	10	6	Power In Power Out	2.7–3.6 V Power Supply Voltage Input. 3.3 V Voltage Regulator Output.
GND	7	3		Ground.
RST/ C2CK	13	9	D I/O D I/O	Device Reset. Open-drain output of internal POR or V _{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 μ s. Clock signal for the C2 Debug Interface.
C2D	14	—	D I/O	Bi-directional data signal for the C2 Debug Interface.
P3.0 / C2D	—	10	D I/O D I/O	Port 3.0. See Section 20 for a complete description of Port 3. Bi-directional data signal for the C2 Debug Interface.
REGIN	11	7	Power In	5 V Regulator Input. This pin is the input to the on-chip voltage regulator.
VBUS	12	8	D In	VBUS Sense Input. This pin should be connected to the VBUS signal of a USB network. A 5 V signal on this pin indicates a USB network connection.
D+	8	4	D I/O	USB D+.
D–	9	5	D I/O	USB D–.
P0.0	6	2	D I/O or A In	Port 0.0. See Section 20 for a complete description of Port 0.
P0.1	5	1	D I/O or A In	Port 0.1.
P0.2	4	32	D I/O or A In	Port 0.2.
P0.3	3	31	D I/O or A In	Port 0.3.
P0.4	2	30	D I/O or A In	Port 0.4.
P0.5	1	29	D I/O or A In	Port 0.5.
P0.6	48	28	D I/O or A In	Port 0.6.

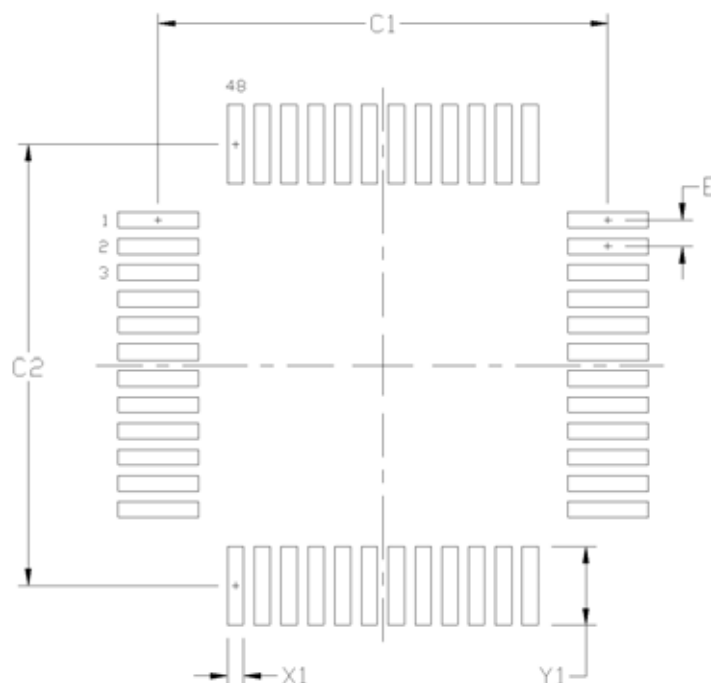


Figure 3.3. TQFP-48 Recommended PCB Land Pattern

Table 3.3. TQFP-48 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	8.30	8.40
C2	8.30	8.40
E	0.50 BSC	
X1	0.20	0.30
Y1	1.40	1.50

Notes:

General:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design:

3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μ m minimum, all the way around the pad.

Stencil Design:

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all pads.

Card Assembly:

7. A No-Clean, Type-3 solder paste is recommended.
8. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

C8051F380/1/2/3/4/5/6/7/C

Table 5.10. ADC0 Electrical Characteristics

$V_{DD} = 3.0\text{ V}$, $V_{REF} = 2.40\text{ V}$ (REFSL=0), PGA Gain = 1, -40 to $+85\text{ }^{\circ}\text{C}$ unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
DC Accuracy					
Resolution		10			bits
Integral Nonlinearity		—	± 0.5	± 1	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	± 0.5	± 1	LSB
Offset Error		-2	0	2	LSB
Full Scale Error		-5	-2	0	LSB
Offset Temperature Coefficient		—	0.005	—	LSB/ $^{\circ}\text{C}$
Dynamic performance (10 kHz sine-wave single-ended input, 1 dB below Full Scale, 500 kps)					
Signal-to-Noise Plus Distortion		55	58	—	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	-73	—	dB
Spurious-Free Dynamic Range		—	78	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	8.33	MHz
Conversion Time in SAR Clocks	10-bit Mode	13	—	—	clocks
	8-bit Mode	11	—	—	clocks
Track/Hold Acquisition Time		300	—	—	ns
Throughput Rate		—	—	500	kps
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ – GND)	0	—	V_{REF}	V
	Differential (AIN+ – AIN–)	$-V_{REF}$	—	V_{REF}	V
Absolute Pin Voltage with respect to GND	Single Ended or Differential	0	—	V_{DD}	V
Sampling Capacitance		—	30	—	pF
Input Multiplexer Impedance		—	5	—	k Ω
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Operating Mode, 500 kps	—	750	1000	μA
Power Supply Rejection		—	1	—	mV/V
Note: Represents one standard deviation from the mean.					

8.1. Comparator Multiplexers

C8051F380/1/2/3/4/5/6/7/C devices include an analog input multiplexer to connect Port I/O pins to the comparator inputs. The Comparator inputs are selected in the CPTnMX registers (SFR Definition 8.5 and SFR Definition 8.6). The CMXnP2–CMXnP0 bits select the Comparator positive input; the CMXnN2–CMXnN0 bits select the Comparator negative input.

Important Note About Comparator Inputs: The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “20.3. General Purpose Port I/O” on page 161).

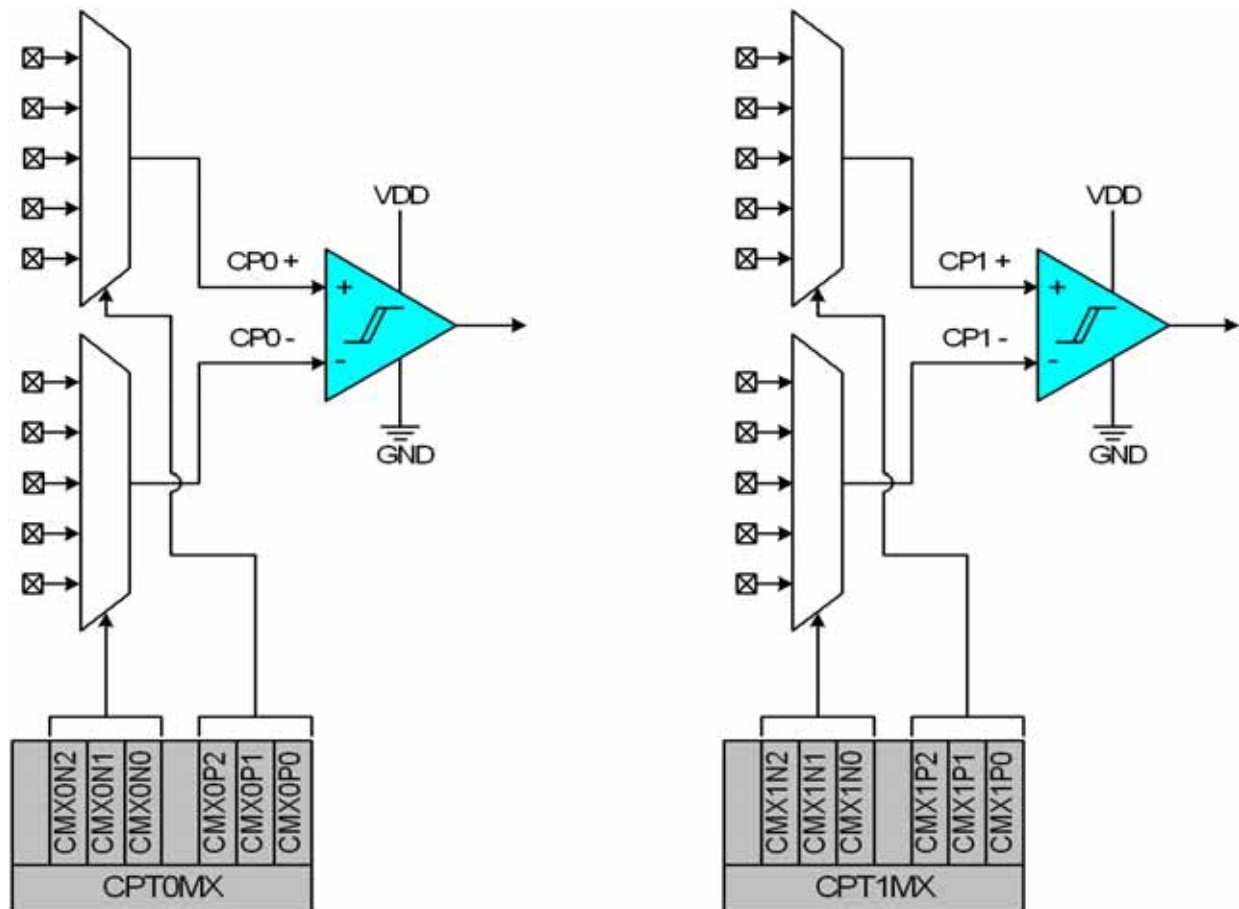


Figure 8.4. Comparator Input Multiplexer Block Diagram

SFR Definition 8.6. CPT1MX: Comparator1 MUX Selection

Bit	7	6	5	4	3	2	1	0
Name		CMX1N[2:0]				CMX1P[2:0]		
Type	R	R/W			R	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9E; SFR Page = All Pages

Bit	Name	Function		
7	Unused	Read = 0b; Write = don't care.		
6:4	CMX1N[2:0]	Comparator1 Negative Input MUX Selection.		
		Selection	32-pin Package	48-pin Package
		000:	P1.3	P2.3
		001:	P1.7	P3.1
		010:	P2.3	P4.0
		011:	Reserved	P4.6
		100:	P0.5	P1.2
		101-111:	Reserved	Reserved
3	Unused	Read = 0b; Write = don't care.		
2:0	CMX1P[2:0]	Comparator1 Positive Input MUX Selection.		
		Selection	32-pin Package	48-pin Package
		000:	P1.2	P2.2
		001:	P1.6	P3.0
		010:	P2.2	P3.7
		011:	Reserved	P4.5
		100:	P0.4	P1.1
		101-111:	Reserved	Reserved

nitely, waiting for an external stimulus to wake up the system. Refer to Section “17.6. PCA Watchdog Timer Reset” on page 133 for more information on the use and configuration of the WDT.

10.2. Stop Mode

Setting the stop mode Select bit (PCON.1) causes the controller core to enter stop mode as soon as the instruction that sets the bit completes execution. In stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering stop mode. Stop mode can only be terminated by an internal or external reset. On reset, the device performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout.

By default, when in stop mode the internal regulator is still active. However, the regulator can be configured to shut down while in stop mode to save power. To shut down the regulator in stop mode, the STOPCF bit in register REG01CN should be set to 1 prior to setting the STOP bit (see SFR Definition 9.1). If the regulator is shut down using the STOPCF bit, only the $\overline{\text{RST}}$ pin or a full power cycle are capable of resetting the device.

10.3. Suspend Mode

Setting the SUSPEND bit (OSCICN.5) causes the hardware to halt the high-frequency internal oscillator and go into suspend mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. The CPU is not halted in Suspend, so code can still be executed using an oscillator other than the internal high-frequency oscillator.

Suspend mode can be terminated by resume signalling on the USB data pins, or a device reset event. When suspend mode is terminated, if the oscillator source is the internal high-frequency oscillator, the device will continue execution on the instruction following the one that set the SUSPEND bit. If the wake event was configured to generate an interrupt, the interrupt will be serviced upon waking the device. If suspend mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

C8051F380/1/2/3/4/5/6/7/C

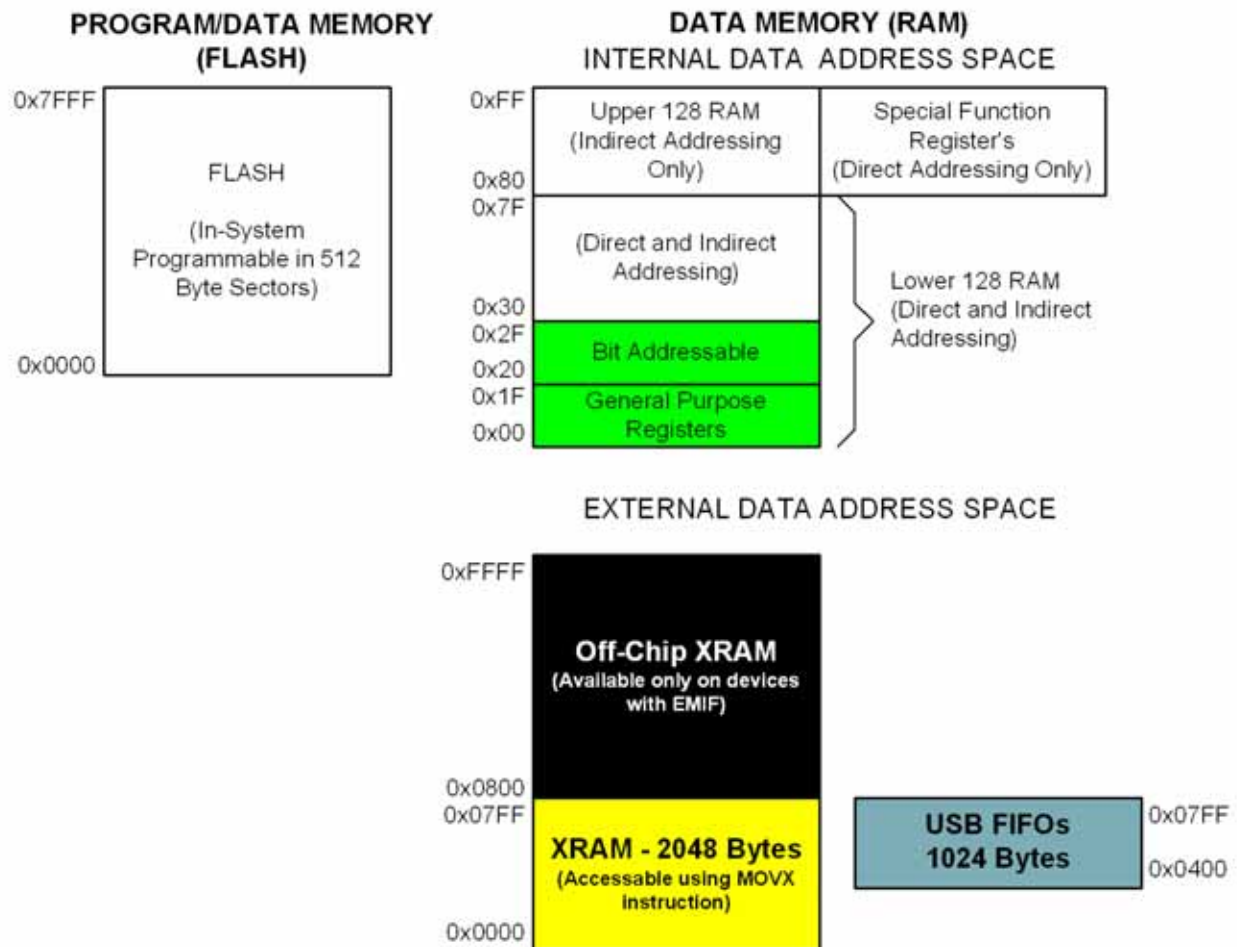


Figure 13.2. On-Chip Memory Map for 32 kB Devices (C8051F382/3/6/7)

15. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the C8051F380/1/2/3/4/5/6/7/C's resources and peripherals. The CIP-51 controller core duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the C8051F380/1/2/3/4/5/6/7/C. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 15.1 lists the SFRs implemented in the C8051F380/1/2/3/4/5/6/7/C device family.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 15.2, for a detailed description of each register.

15.1. SFR Paging

The CIP-51 features SFR paging, allowing the device to map many SFRs into the 0x80 to 0xFF memory address space. The SFR memory space has 256 pages. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFRs. The C8051F380/1/2/3/4/5/6/7/C devices utilize two SFR pages: 0x0, and 0xF. Most SFRs are available on both pages. SFR pages are selected using the Special Function Register Page Selection register, SFRPAGE. The procedure for reading and writing an SFR is as follows:

1. Select the appropriate SFR page number using the SFRPAGE register.
2. Use direct accessing mode to read or write the special function register (MOV instruction).

Important Note: When reading or writing SFRs that are not available on all pages within an ISR, it is recommended to save the state of the SFRPAGE register on ISR entry, and restore state on exit.

SFR Definition 15.1. SFRPAGE: SFR Page

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

SFR Address = 0xBF; SFR Page = All Pages

Bit	Name	Function
7:0	SFRPAGE[7:0]	SFR Page Bits. Represents the SFR Page the C8051 core uses when reading or modifying SFRs. Write: Sets the SFR Page. Read: Byte is the SFR page the C8051 core is using.

C8051F380/1/2/3/4/5/6/7/C

Accessing FLASH from the C2 debug interface:

1. Any unlocked page may be read, written, or erased.
2. Locked pages cannot be read, written, or erased.
3. The page containing the Lock Byte may be read, written, or erased if it is unlocked.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing 1s to 0s in the Lock Byte) is not permitted.
6. Unlocking FLASH pages (changing 0s to 1s in the Lock Byte) requires the C2 Device Erase command, which erases all FLASH pages including the page containing the Lock Byte and the Lock Byte itself.
7. The Reserved Area cannot be read, written, or erased.

Accessing FLASH from user firmware executing on an **unlocked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Locked pages cannot be read, written, or erased.
3. The page containing the Lock Byte cannot be erased. It may be read or written only if it is unlocked.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing 1s to 0s in the Lock Byte) is not permitted.
6. Unlocking FLASH pages (changing 0s to 1s in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a FLASH Error device reset.

Accessing FLASH from user firmware executing on a **locked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Any locked page except the page containing the Lock Byte may be read, written, or erased.
3. The page containing the Lock Byte cannot be erased. It may only be read or written.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing 1s to 0s in the Lock Byte) is not permitted.
6. Unlocking FLASH pages (changing 0s to 1s in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a FLASH Error device reset.

SFR Definition 20.3. XBR2: Port I/O Crossbar Register 2

Bit	7	6	5	4	3	2	1	0
Name							SMB1E	URT1E
Type	R/W	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 = 0xE3; SFR Page = All Pages

Bit	Name	Function
7:2	Reserved	Must write 000000b
1	SMB1E	SMBus1 I/O Enable. 0: SMBus1 I/O unavailable at Port pins. 1: SMBus1 I/O routed to Port pins.
0	URT1E	UART1 I/OEnable. 0: UART1 I/O unavailable at Port pins. 1: UART1 TX1, RX1 routed to Port pins.

20.3. General Purpose Port I/O

Port pins that remain unassigned by the Crossbar and are not used by analog peripherals can be used for general purpose I/O. Ports 3-0 are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable. Port 4 (C8051F380/2/4/6 only) uses an SFR which is byte-addressable. When writing to a Port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the Port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the Crossbar, the Port register can always read its corresponding Port I/O pin). The exception to this is the execution of the read-modify-write instructions. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a Port SFR. For these instructions, the value of the register (not the pin) is read, modified, and written back to the SFR.

C8051F380/1/2/3/4/5/6/7/C

USB Register Definition 21.4. INDEX: USB0 Endpoint Index

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

USB Register Address = 0x0E

Bit	Name	Function
7:4	Unused	Read = 0000b. Write = don't care.
3:0	EPSEL[3:0]	Endpoint Select Bits. These bits select which endpoint is targeted when indexed USB0 registers are accessed. 0000: Endpoint 0 0001: Endpoint 1 0010: Endpoint 2 0011: Endpoint 3 0100-1111: Reserved.

21.4. USB Clock Configuration

USB0 is capable of communication as a Full or Low Speed USB function. Communication speed is selected via the SPEED bit in SFR USB0XCN. When operating as a Low Speed function, the USB0 clock must be 6 MHz. When operating as a Full Speed function, the USB0 clock must be 48 MHz. Clock options are described in Section “19. Oscillators and Clock Selection” on page 142. The USB0 clock is selected via SFR CLKSEL (see SFR Definition 19.1).

Clock Recovery circuitry uses the incoming USB data stream to adjust the internal oscillator; this allows the internal oscillator to meet the requirements for USB clock tolerance. Clock Recovery should be used in the following configurations:

Communication Speed	USB Clock
Full Speed	Internal Oscillator
Low Speed	Internal Oscillator / 8

When operating USB0 as a Low Speed function with Clock Recovery, software must write 1 to the CRLOW bit to enable Low Speed Clock Recovery. Clock Recovery is typically not necessary in Low Speed mode.

Single Step Mode can be used to help the Clock Recovery circuitry to lock when high noise levels are present on the USB network. This mode is not required (or recommended) in typical USB environments.

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 22.3 illustrates a typical SMBus transaction.

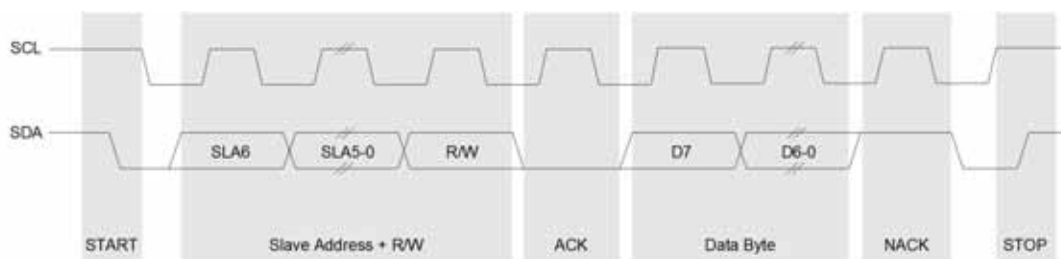


Figure 22.3. SMBus Transaction

22.3.1. Transmitter Vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

22.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “22.3.5. SCL High (SMBus Free) Timeout” on page 208). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

22.3.3. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

22.3.4. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

For the SMBus0 interface, Timer 3 is used to implement SCL low timeouts. Timer 4 is used on the SMBus1 interface for SCL low timeouts. The SCL low timeout feature is enabled by setting the SMBnTOE bit in SMBnCF. The associated timer is forced to reload when SCL is high, and allowed to count when SCL is

22.4.3. SMBnCN Control Register

SMBnCN is used to control the interface and to provide status information (see SFR Definition 22.4). The higher four bits of SMBnCN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a 1 to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a 1 to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 22.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

22.4.3.1. Software ACK Generation

When the EHACK bit in register SMBnADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

22.4.3.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 22.4.4. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 22.3 lists all sources for hardware changes to the SMBnCN bits. Refer to Table 22.5 for SMBus status decoding using the SMBnCN register.

Table 22.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0) (Continued)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—
Slave Receiver	0010	1	0	X	A slave address + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
	0010				Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	—

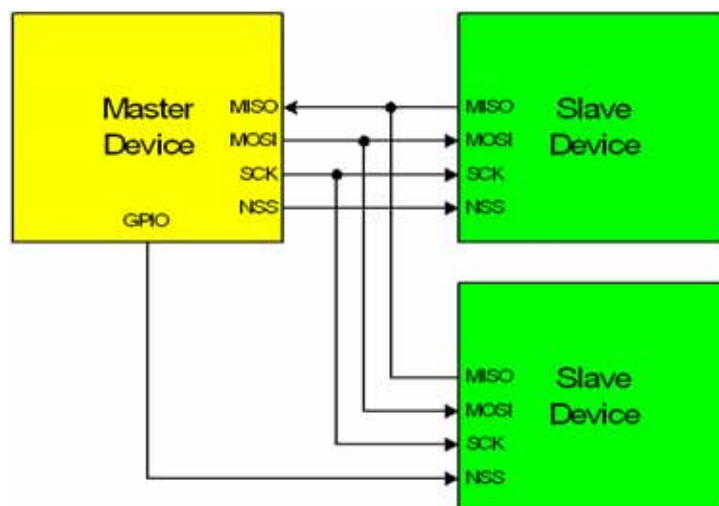


Figure 25.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

25.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 25.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 25.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

C8051F380/1/2/3/4/5/6/7/C

SFR Definition 26.1. CKCON: Clock Control

Bit	7	6	5	4	3	2	1	0
Name	T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA[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 = 0x8E; SFR Page = All Pages

Bit	Name	Function
7	T3MH	Timer 3 High Byte Clock Select. Selects the clock supplied to the Timer 3 high byte (split 8-bit timer mode only). 0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 high byte uses the system clock.
6	T3ML	Timer 3 Low Byte Clock Select. Selects the clock supplied to Timer 3. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 low byte uses the system clock.
5	T2MH	Timer 2 High Byte Clock Select. Selects the clock supplied to the Timer 2 high byte (split 8-bit timer mode only). 0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 high byte uses the system clock.
4	T2ML	Timer 2 Low Byte Clock Select. Selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer. 0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 low byte uses the system clock.
3	T1	Timer 1 Clock Select. Selects the clock source supplied to Timer 1. Ignored when C/T1 is set to 1. 0: Timer 1 uses the clock defined by the prescale bits SCA[1:0]. 1: Timer 1 uses the system clock.
2	T0	Timer 0 Clock Select. Selects the clock source supplied to Timer 0. Ignored when C/T0 is set to 1. 0: Counter/Timer 0 uses the clock defined by the prescale bits SCA[1:0]. 1: Counter/Timer 0 uses the system clock.
1:0	SCA[1:0]	Timer 0/1 Prescale Bits. These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12 01: System clock divided by 4 10: System clock divided by 48 11: External clock divided by 8 (synchronized with the system clock)

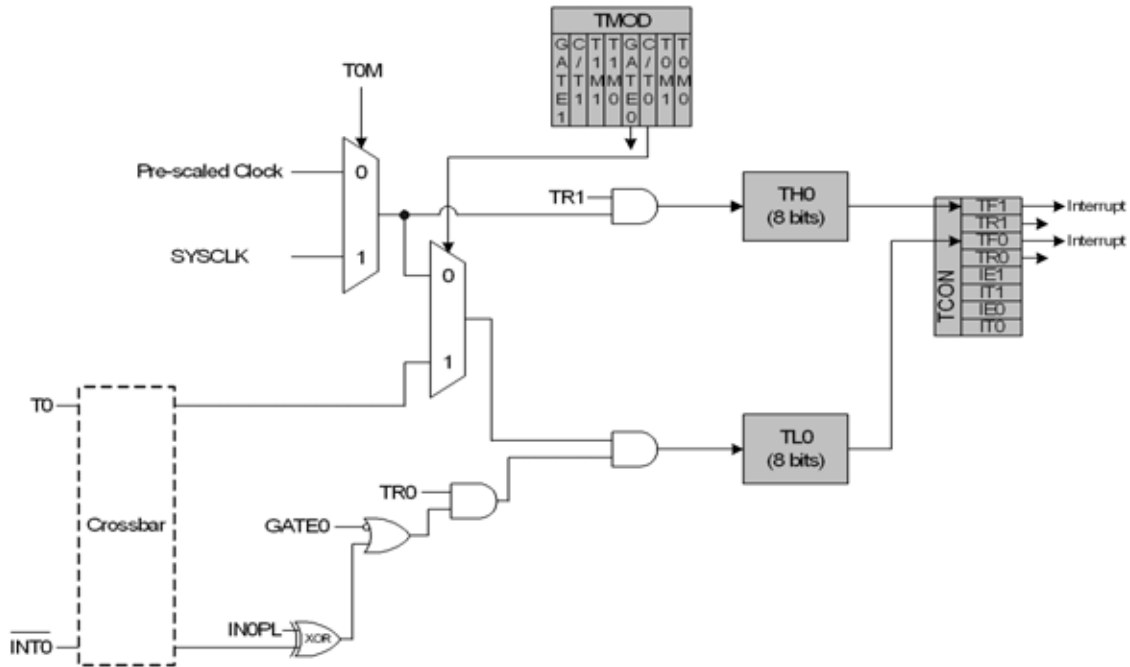


Figure 26.3. T0 Mode 3 Block Diagram

SFR Definition 26.18. TMR3H Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x95; SFR Page = 0

Bit	Name	Function
7:0	TMR3H[7:0]	Timer 3 High Byte. In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

SFR Definition 26.28. TMR5H Timer 5 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR5H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD; SFR Page = F

Bit	Name	Function
7:0	TMR5H[7:0]	Timer 5 High Byte. In 16-bit mode, the TMR5H register contains the high byte of the 16-bit Timer 5. In 8-bit mode, TMR5H contains the 8-bit high byte timer value.