

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART
Peripherals	Cap Sense, POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f812-gmr

C8051F80x-83x

26.4.4. Data Register	192
26.5. SMBus Transfer Modes	193
26.5.1. Write Sequence (Master)	193
26.5.2. Read Sequence (Master)	194
26.5.3. Write Sequence (Slave)	195
26.5.4. Read Sequence (Slave)	196
26.6. SMBus Status Decoding	196
27. UART0	201
27.1. Enhanced Baud Rate Generation	202
27.2. Operational Modes	203
27.2.1. 8-Bit UART	203
27.2.2. 9-Bit UART	204
27.3. Multiprocessor Communications	205
28. Timers	209
28.1. Timer 0 and Timer 1	211
28.1.1. Mode 0: 13-bit Counter/Timer	211
28.1.2. Mode 1: 16-bit Counter/Timer	212
28.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload	212
28.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)	213
28.2. Timer 2	219
28.2.1. 16-bit Timer with Auto-Reload	219
28.2.2. 8-bit Timers with Auto-Reload	220
29. Programmable Counter Array	225
29.1. PCA Counter/Timer	226
29.2. PCA0 Interrupt Sources	227
29.3. Capture/Compare Modules	228
29.3.1. Edge-Triggered Capture Mode	229
29.3.2. Software Timer (Compare) Mode	230
29.3.3. High-Speed Output Mode	231
29.3.4. Frequency Output Mode	232
29.3.5. 8-bit through 15-bit Pulse Width Modulator Modes	232
29.3.5.1. 8-bit Pulse Width Modulator Mode	233
29.3.5.2. 9-bit through 15-bit Pulse Width Modulator Mode	234
29.3.6. 16-Bit Pulse Width Modulator Mode	235
29.4. Watchdog Timer Mode	236
29.4.1. Watchdog Timer Operation	236
29.4.2. Watchdog Timer Usage	237
29.5. Register Descriptions for PCA0	237
30. C2 Interface	244
30.1. C2 Interface Registers	244
30.2. C2CK Pin Sharing	247
Document Change List	248
Contact Information	250

List of Tables

1. System Overview	
2. Ordering Information	
Table 2.1. Product Selection Guide	26
3. Pin Definitions	
Table 3.1. Pin Definitions for the C8051F80x-83x	28
4. QFN-20 Package Specifications	
Table 4.1. QFN-20 Package Dimensions	33
Table 4.2. QFN-20 PCB Land Pattern Dimensions	34
5. QSOP-24 Package Specifications	
Table 5.1. QSOP-24 Package Dimensions	35
Table 5.2. QSOP-24 PCB Land Pattern Dimensions	36
6. SOIC-16 Package Specifications	
Table 6.1. SOIC-16 Package Dimensions	37
Table 6.2. SOIC-16 PCB Land Pattern Dimensions	38
7. Electrical Characteristics	
Table 7.1. Absolute Maximum Ratings	39
Table 7.2. Global Electrical Characteristics	40
Table 7.3. Port I/O DC Electrical Characteristics	41
Table 7.4. Reset Electrical Characteristics	41
Table 7.5. Internal Voltage Regulator Electrical Characteristics	41
Table 7.6. Flash Electrical Characteristics	42
Table 7.7. Internal High-Frequency Oscillator Electrical Characteristics	42
Table 7.8. Capacitive Sense Electrical Characteristics	42
Table 7.9. ADC0 Electrical Characteristics	43
Table 7.10. Power Management Electrical Characteristics	44
Table 7.11. Temperature Sensor Electrical Characteristics	44
Table 7.12. Voltage Reference Electrical Characteristics	44
Table 7.13. Comparator Electrical Characteristics	45
8. 10-Bit ADC (ADC0)	
9. Temperature Sensor	
10. Voltage and Ground Reference Options	
11. Voltage Regulator (REG0)	
12. Comparator0	
13. Capacitive Sense (CS0)	
Table 13.1. Operation with Auto-scan and Accumulate	74
14. CIP-51 Microcontroller	
Table 14.1. CIP-51 Instruction Set Summary	84
15. Memory Organization	
16. In-System Device Identification	
17. Special Function Registers	
Table 17.1. Special Function Register (SFR) Memory Map	97
Table 17.2. Special Function Registers	98
18. Interrupts	

SFR Definition 28.8. TMR2CN: Timer 2 Control	222
SFR Definition 28.9. TMR2RLL: Timer 2 Reload Register Low Byte	223
SFR Definition 28.10. TMR2RLH: Timer 2 Reload Register High Byte	223
SFR Definition 28.11. TMR2L: Timer 2 Low Byte	224
SFR Definition 28.12. TMR2H: Timer 2 High Byte	224
SFR Definition 29.1. PCA0CN: PCA0 Control	238
SFR Definition 29.2. PCA0MD: PCA0 Mode	239
SFR Definition 29.3. PCA0PWM: PCA0 PWM Configuration	240
SFR Definition 29.4. PCA0CPMn: PCA0 Capture/Compare Mode	241
SFR Definition 29.5. PCA0L: PCA0 Counter/Timer Low Byte	242
SFR Definition 29.6. PCA0H: PCA0 Counter/Timer High Byte	242
SFR Definition 29.7. PCA0CPLn: PCA0 Capture Module Low Byte	243
SFR Definition 29.8. PCA0CPHn: PCA0 Capture Module High Byte	243
C2 Register Definition 30.1. C2ADD: C2 Address	244
C2 Register Definition 30.3. REVID: C2 Revision ID	245
C2 Register Definition 30.2. DEVICEID: C2 Device ID	245
C2 Register Definition 30.4. FPCTL: C2 Flash Programming Control	246
C2 Register Definition 30.5. FPDAT: C2 Flash Programming Data	246

C8051F80x-83x

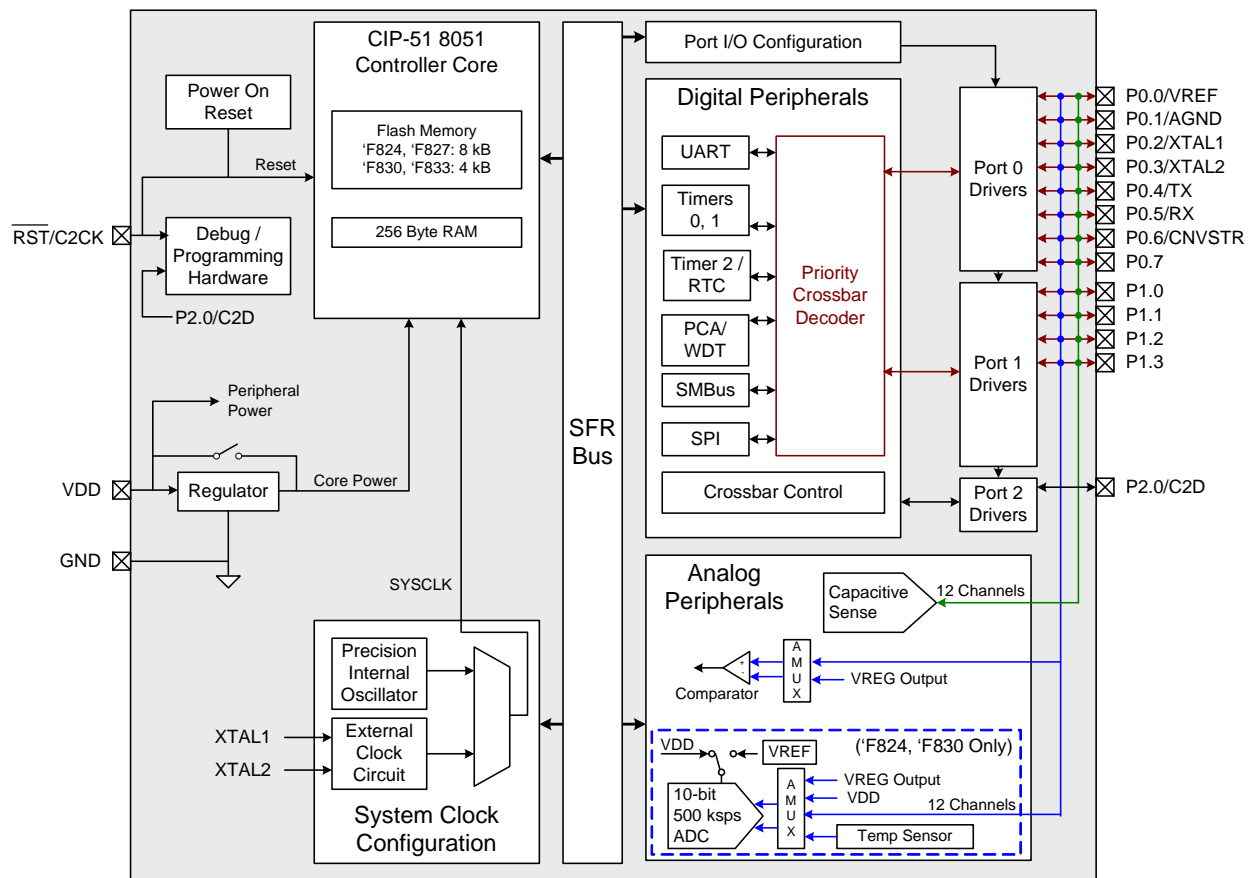


Figure 1.7. C8051F824, C8051F827, C8051F830, C8051F833 Block Diagram

C8051F80x-83x

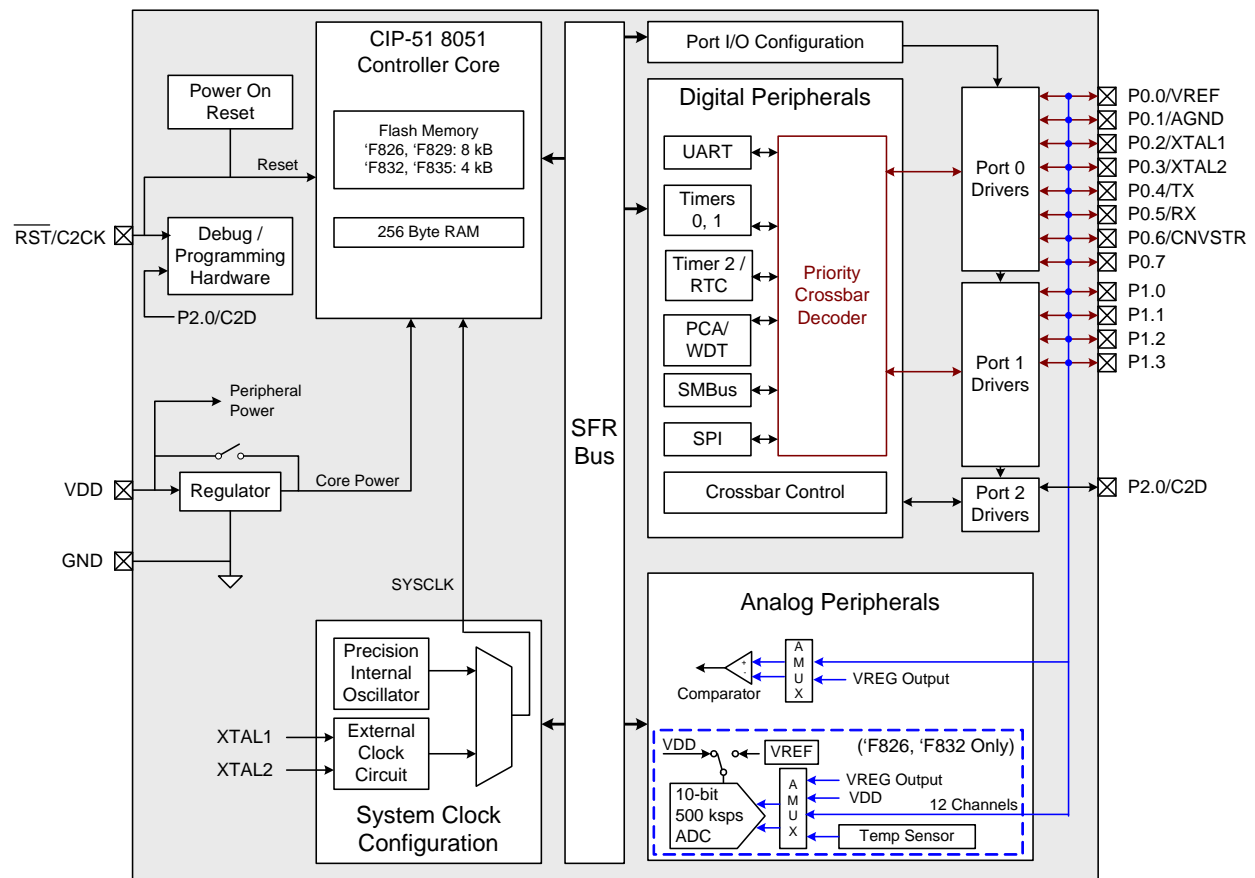


Figure 1.9. C8051F826, C8051F829, C8051F832, C8051F835 Block Diagram

C8051F80x-83x

Table 2.1. Product Selection Guide

Part Number	Digital Port I/Os	Capacitive Sense Channels	Flash Memory (kB)	RAM (Bytes)	10-bit 500 ksps ADC	ADC Channels	Temperature Sensor	Package (RoHS)
C8051F800-GU	17	16	16	512	✓	16	✓	QSOP-24
C8051F801-GU	17	8	16	512	✓	16	✓	QSOP-24
C8051F802-GU	17	—	16	512	✓	16	✓	QSOP-24
C8051F800-GM	17	16	16	512	✓	16	✓	QFN-20
C8051F801-GM	17	8	16	512	✓	16	✓	QFN-20
C8051F802-GM	17	—	16	512	✓	16	✓	QFN-20
C8051F803-GS	13	12	16	512	✓	12	✓	SOIC-16
C8051F804-GS	13	8	16	512	✓	12	✓	SOIC-16
C8051F805-GS	13	—	16	512	✓	12	✓	SOIC-16
C8051F806-GU	17	16	16	512	—	—	—	QSOP-24
C8051F807-GU	17	8	16	512	—	—	—	QSOP-24
C8051F808-GU	17	—	16	512	—	—	—	QSOP-24
C8051F806-GM	17	16	16	512	—	—	—	QFN-20
C8051F807-GM	17	8	16	512	—	—	—	QFN-20
C8051F808-GM	17	—	16	512	—	—	—	QFN-20
C8051F809-GS	13	12	16	512	—	—	—	SOIC-16
C8051F810-GS	13	8	16	512	—	—	—	SOIC-16
C8051F811-GS	13	—	16	512	—	—	—	SOIC-16
C8051F812-GU	17	16	8	512	✓	16	✓	QSOP-24
C8051F813-GU	17	8	8	512	✓	16	✓	QSOP-24
C8051F814-GU	17	—	8	512	✓	16	✓	QSOP-24
C8051F812-GM	17	16	8	512	✓	16	✓	QFN-20
C8051F813-GM	17	8	8	512	✓	16	✓	QFN-20
C8051F814-GM	17	—	8	512	✓	16	✓	QFN-20
C8051F815-GS	13	12	8	512	✓	12	✓	SOIC-16
C8051F816-GS	13	8	8	512	✓	12	✓	SOIC-16
C8051F817-GS	13	—	8	512	✓	12	✓	SOIC-16
C8051F818-GU	17	16	8	512	—	—	—	QSOP-24
C8051F819-GU	17	8	8	512	—	—	—	QSOP-24
C8051F820-GU	17	—	8	512	—	—	—	QSOP-24
C8051F818-GM	17	16	8	512	—	—	—	QFN-20
C8051F819-GM	17	8	8	512	—	—	—	QFN-20
C8051F820-GM	17	—	8	512	—	—	—	QFN-20

C8051F80x-83x

SFR Definition 8.2. ADC0H: ADC0 Data Word MSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBE

Bit	Name	Function
7:0	ADC0H[7:0]	ADC0 Data Word High-Order Bits. For AD0LJST = 0: Bits 7–2 will read 000000b. Bits 1–0 are the upper 2 bits of the 10-bit ADC0 Data Word. For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 10-bit ADC0 Data Word. Note: In 8-bit mode AD0LJST is ignored, and ADC0H holds the 8-bit data word.

SFR Definition 8.3. ADC0L: ADC0 Data Word LSB

Bit	7	6	5	4	3	2	1	0
Name	ADC0L[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBD

Bit	Name	Function
7:0	ADC0L[7:0]	ADC0 Data Word Low-Order Bits. For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 10-bit Data Word. For AD0LJST = 1: Bits 7–6 are the lower 2 bits of the 10-bit Data Word. Bits 5–0 will always read 0. Note: In 8-bit mode AD0LJST is ignored, and ADC0L will read back 00000000b.

10.1. External Voltage References

To use an external voltage reference, REFSL[1:0] should be set to 00. Bypass capacitors should be added as recommended by the manufacturer of the external voltage reference.

10.2. Internal Voltage Reference Options

A 1.65 V high-speed reference is included on-chip. The high speed internal reference is selected by setting REFSL[1:0] to 11. When selected, the high speed internal reference will be automatically enabled on an as-needed basis by ADC0.

For applications with a non-varying power supply voltage, using the power supply as the voltage reference can provide ADC0 with added dynamic range at the cost of reduced power supply noise rejection. To use the 1.8 to 3.6 V power supply voltage (V_{DD}) or the 1.8 V regulated digital supply voltage as the reference source, REFSL[1:0] should be set to 01 or 10, respectively.

10.3. Analog Ground Reference

To prevent ground noise generated by switching digital logic from affecting sensitive analog measurements, a separate analog ground reference option is available. When enabled, the ground reference for ADC0 is taken from the P0.1/AGND pin. Any external sensors sampled by ADC0 should be referenced to the P0.1/AGND pin. The separate analog ground reference option is enabled by setting REFGND to 1. Note that when using this option, P0.1/AGND must be connected to the same potential as GND.

10.4. Temperature Sensor Enable

The TEMPE bit in register REF0CN enables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data.

Table 14.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2

Table 18.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y		ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
Port Match	0x0043	8	None	N/A	N/A	EMAT (EIE1.1)	PMAT (EIP1.1)
ADC0 Window Compare	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 Conversion Complete	0x0053	10	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
RESERVED							
RESERVED							
CS0 Conversion Complete	0x007B	15	CS0INT (CS0CN.5)	N	N	ECSCPT (EIE2.0)	PSCCPT (EIP2.0)
CS0 Greater Than	0x0083	16	CS0CMPF (CS0CN.0)	N	N	ECSGRT (EIE2.1)	PSCGRT (EIP2.1)

18.2. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described in this section. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

23.1.3. Interfacing Port I/O to 5 V Logic

All Port I/O configured for digital, open-drain operation are capable of interfacing to digital logic operating at a supply voltage up to 2 V higher than VDD and less than 5.25 V. An external pull-up resistor to the higher supply voltage is typically required for most systems.

Important Note: In a multi-voltage interface, the external pull-up resistor should be sized to allow a current of at least 150 μA to flow into the Port pin when the supply voltage is between (VDD + 0.6V) and (VDD + 1.0V). Once the Port pin voltage increases beyond this range, the current flowing into the Port pin is minimal. Figure 23.3 shows the input current characteristics of port pins driven above VDD. The port pin requires 150 μA peak overdrive current when its voltage reaches approximately (VDD + 0.7 V).

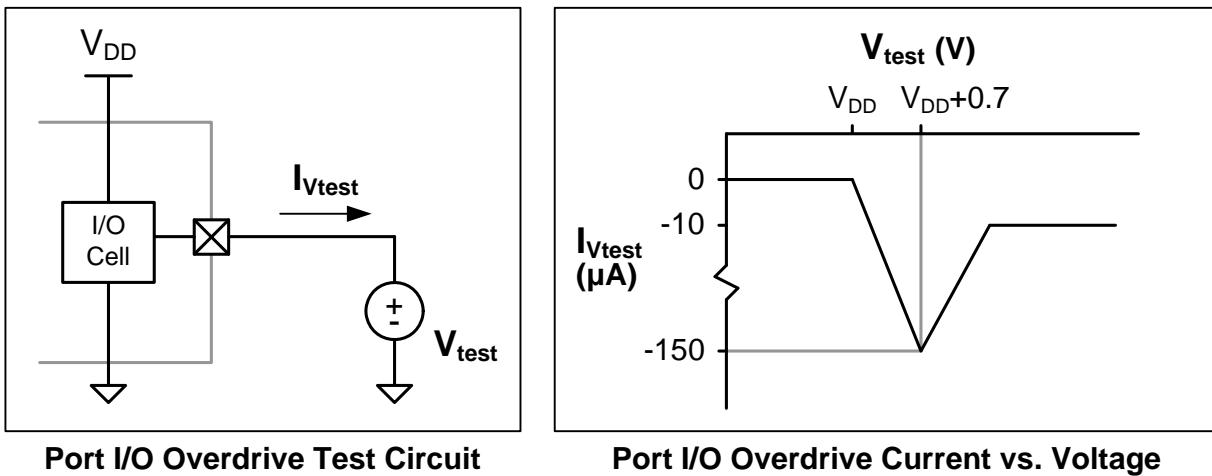


Figure 23.3. Port I/O Overdrive Current

23.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins P0.0–P1.7 can be assigned to various analog, digital, and external interrupt functions. The Port pins assigned to analog functions should be configured for analog I/O, and Port pins assigned to digital or external interrupt functions should be configured for digital I/O.

23.2.1. Assigning Port I/O Pins to Analog Functions

Table 23.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. **Any selected pins should also have their corresponding bit in the Port Latch set to 1 (Pn.n = 1).** This prevents the low port I/O drive circuit from pulling the pin low. Table 23.1 shows the potential mapping of Port I/O to each analog function.

23.3. Priority Crossbar Decoder

The Priority Crossbar Decoder assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which is always at pins 4 and 5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Because of the nature of the Priority Crossbar Decoder, not all peripherals can be located on all port pins. Figure 23.4 maps peripherals to the potential port pins on which the peripheral I/O can appear.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.1 if AGND is used, P0.3 and/or P0.2 if the external oscillator circuit is enabled, P0.6 if the ADC is configured to use the external conversion start signal (CNVSTR), and any selected ADC, Comparator, or Capacitive Sense inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin.

Registers XBR0, XBR1, and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL); when a UART is selected, the Crossbar assigns both pins associated with the UART (TX and RX). UART0 pin assignments are fixed for bootloading purposes: UART TX0 is always assigned to P0.4; UART RX0 is always assigned to P0.5. Standard Port I/Os appear contiguously after the prioritized functions have been assigned.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 25.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 25.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 25.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

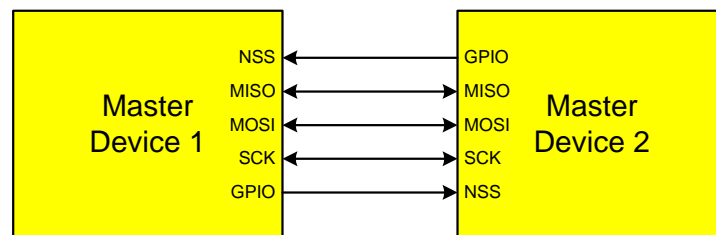


Figure 25.2. Multiple-Master Mode Connection Diagram

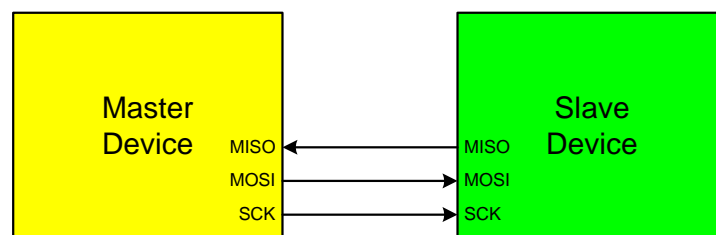


Figure 25.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

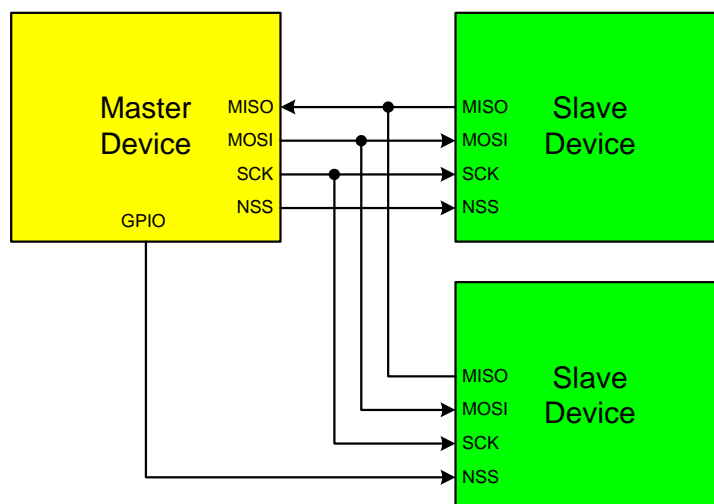


Figure 25.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

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.

SFR Definition 26.2. SMB0CN: SMBus Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI
Type	R	R	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC0; Bit-Addressable

Bit	Name	Description	Read	Write
7	MASTER	SMBus Master/Slave Indicator. This read-only bit indicates when the SMBus is operating as a master.	0: SMBus operating in slave mode. 1: SMBus operating in master mode.	N/A
6	TXMODE	SMBus Transmit Mode Indicator. This read-only bit indicates when the SMBus is operating as a transmitter.	0: SMBus in Receiver Mode. 1: SMBus in Transmitter Mode.	N/A
5	STA	SMBus Start Flag.	0: No Start or repeated Start detected. 1: Start or repeated Start detected.	0: No Start generated. 1: When Configured as a Master, initiates a START or repeated START.
4	STO	SMBus Stop Flag.	0: No Stop condition detected. 1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).	0: No STOP condition is transmitted. 1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle. Cleared by Hardware.
3	ACKRQ	SMBus Acknowledge Request.	0: No Ack requested 1: ACK requested	N/A
2	ARBLOST	SMBus Arbitration Lost Indicator.	0: No arbitration error. 1: Arbitration Lost	N/A
1	ACK	SMBus Acknowledge.	0: NACK received. 1: ACK received.	0: Send NACK 1: Send ACK
0	SI	SMBus Interrupt Flag. This bit is set by hardware under the conditions listed in Table 15.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.	0: No interrupt pending 1: Interrupt Pending	0: Clear interrupt, and initiate next state machine event. 1: Force interrupt.

26.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE 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 received.

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.**

The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 26.7 shows a typical slave write 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.

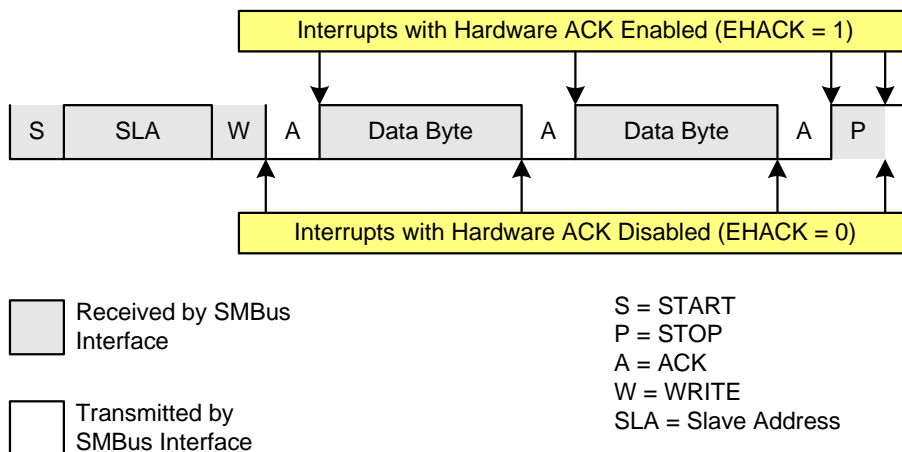


Figure 26.7. Typical Slave Write Sequence

Table 26.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1)
(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	0	0	X	A slave address + R/W was received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
		0	1	X	Lost arbitration as master; slave address + R/W received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
	0001					Reschedule failed transfer	1	0	X	1110
		0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
	0000	0	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
		0	0	X	A slave byte was received.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	0000
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Set NACK for next data byte; Read SMB0DAT.	0	0	0	0000
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	0	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110

27.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown in Figure 27.3.

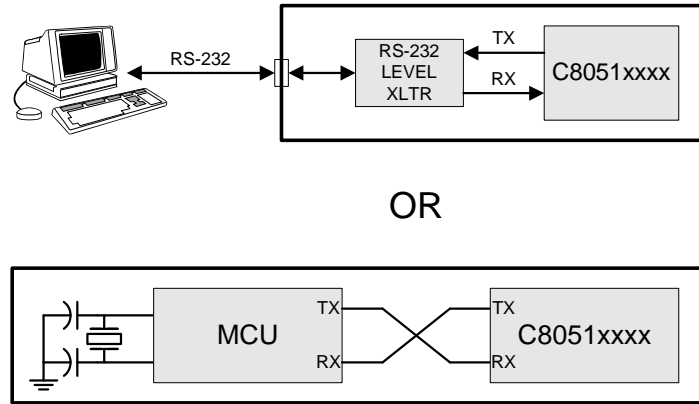


Figure 27.3. UART Interconnect Diagram

27.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software 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 logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

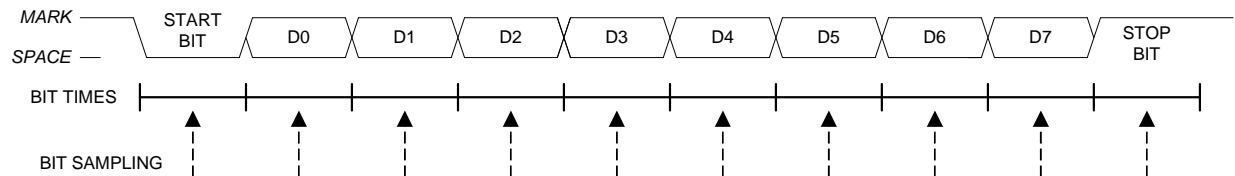


Figure 27.4. 8-Bit UART Timing Diagram

28.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 28.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode. Timer 2 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 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

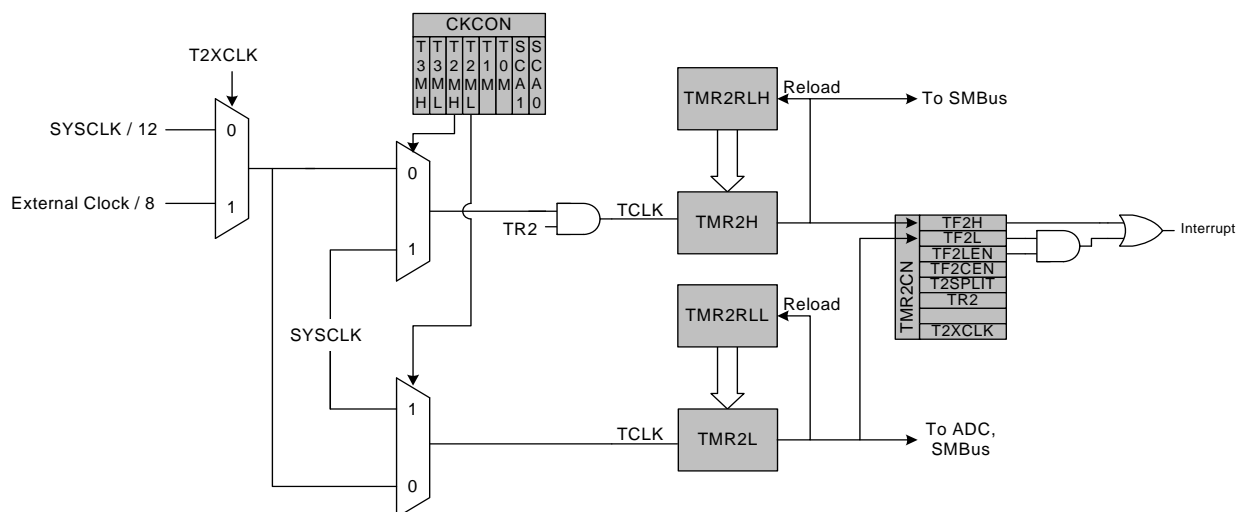


Figure 28.5. Timer 2 8-Bit Mode Block Diagram

29.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. **Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter.** Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2–CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 29.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 29.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8 (Note)
1	1	x	Reserved

Note: External oscillator source divided by 8 is synchronized with the system clock.

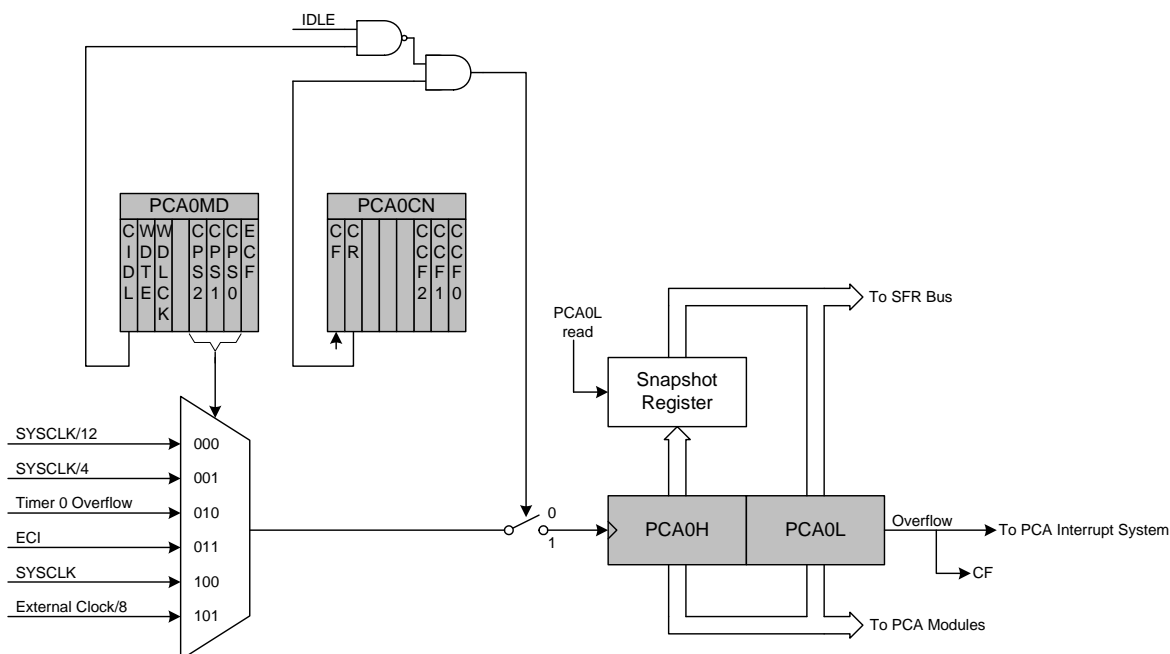


Figure 29.2. PCA Counter/Timer Block Diagram