Silicon Labs - C8051F005R Datasheet

Welcome to E-XFL.COM

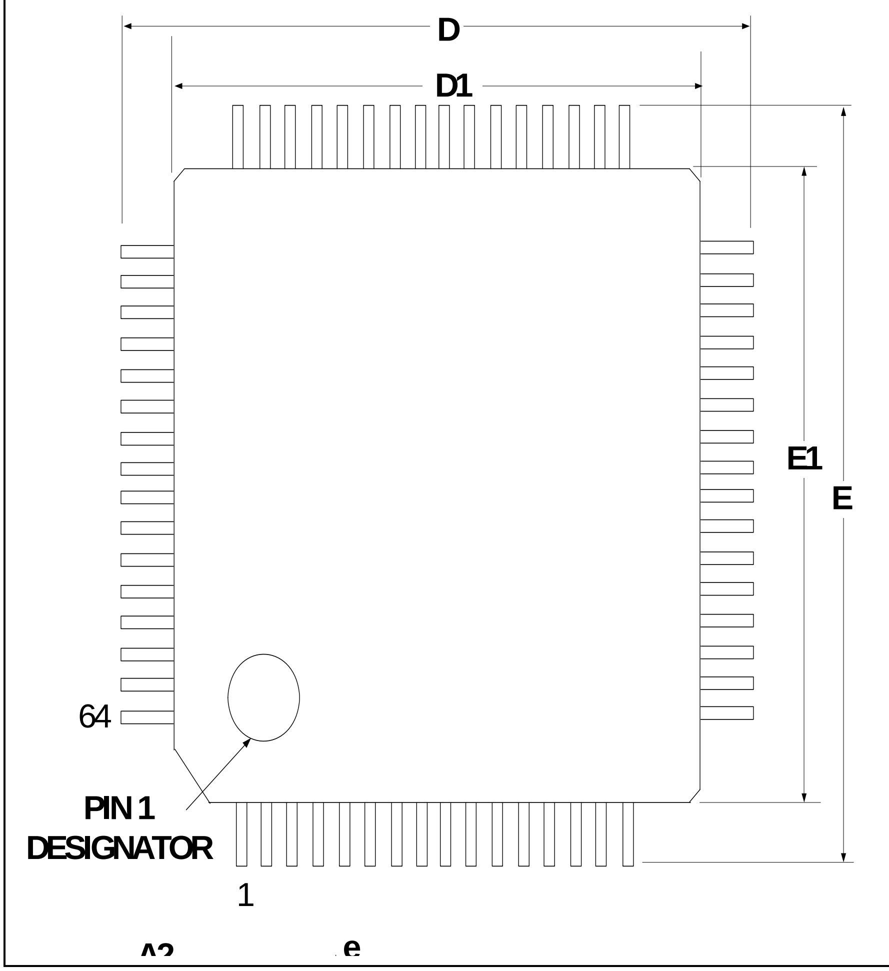
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	32
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 ~ 3.6V
Data Converters	A/D 8x12b; D/A 2x12b
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/c8051f005r

Figure 4.2. TQFP-64 Package Drawing



	MIN (mm)	NOM (mm)
A	-	-
A1	0.05	-
A2	0.95	-
b	0.17	0.22
D	-	12.00
D1	-	10.00
e	-	0.50

Figure 5.8. ADC0H: ADC Data Word MSB Register (C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xBF

Bits7-0: ADC Data Word Bits
For ADLJST = 1: Upper 8-bits of the 12-bit ADC Data Word.
For ADLJST = 0: Bits7-4 are the sign extension of Bit3. Bits 3-0 are the upper 4-bits of the 12-bit ADC Data Word.

Figure 5.9. ADC0L: ADC Data Word LSB Register (C8051F00x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xBE

Bits7-0: ADC Data Word Bits
For ADLJST = 1: Bits7-4 are the lower 4-bits of the 12-bit ADC Data Word. Bits3-0 will always read 0.
For ADLJST = 0: Bits7-0 are the lower 8-bits of the 12-bit ADC Data Word.

NOTE: Resulting 12-bit ADC Data Word appears in the ADC Data Word Registers as follows:
ADC0H[3:0]:ADC0L[7:0], if ADLJST = 0
(ADC0H[7:4] will be sign extension of ADC0H.3 if a differential reading, otherwise = 0000b)

ADC0H[7:0]:ADC0L[7:4], if ADLJST = 1
(ADC0L[3:0] = 0000b)

EXAMPLE: ADC Data Word Conversion Map, AIN0 Input in Single-Ended Mode
(AMX0CF=0x00, AMX0SL=0x00)

AIN0 – AGND (Volts)	**ADC0H:ADC0L** (ADLJST = 0)	**ADC0H:ADC0L** (ADLJST = 1)
REF x (4095/4096)	0x0FFF	0xFFF0
REF x ½	0x0800	0x8000
REF x (2047/4096)	0x07FF	0x7FF0
0	0x0000	0x0000

EXAMPLE: ADC Data Word Conversion Map, AIN0-AIN1 Differential Input Pair
(AMX0CF=0x01, AMX0SL=0x00)

AIN0 – AIN1 (Volts)	**ADC0H:ADC0L** (ADLJST = 0)	**ADC0H:ADC0L** (ADLJST = 1)
REF x (2047/2048)	0x07FF	0x7FF0
0	0x0000	0x0000
-REF x (1/2048)	0xFFFF	0xFFF0
-REF	0xF800	0x8000

Figure 6.6. ADC0CF: ADC Configuration Register (C8051F01x)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADCSC2	ADCSC1	ADCSC0	-	-	AMPGN2	AMPGN1	AMPGN0	01100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

Bits7-5: ADCSC2-0: ADC SAR Conversion Clock Period Bits
000: SAR Conversion Clock = 1 System Clock
001: SAR Conversion Clock = 2 System Clocks
010: SAR Conversion Clock = 4 System Clocks
011: SAR Conversion Clock = 8 System Clocks
1xx: SAR Conversion Clock = 16 Systems Clocks
(Note: Conversion clock should be ≤ 2MHz.)
Bits4-3: UNUSED. Read = 00b; Write = don’t care
Bits2-0: AMPGN2-0: ADC Internal Amplifier Gain
000: Gain = 1
001: Gain = 2
010: Gain = 4
011: Gain = 8
10x: Gain = 16
11x: Gain = 0.5

Figure 7.2. DAC0H: DAC0 High Byte Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD3

Bits7-0: DAC0 Data Word Most Significant Byte.

Figure 7.3. DAC0L: DAC0 Low Byte Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD2

Bits7-0: DAC0 Data Word Least Significant Byte.

Figure 7.4. DAC0CN: DAC0 Control Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
DAC0EN	-	-	-	-	DAC0DF2	DAC0DF1	DAC0DF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD4

Bit7: DAC0EN: DAC0 Enable Bit
0: DAC0 Disabled. DAC0 Output pin is disabled; DAC0 is in low power shutdown mode.
1: DAC0 Enabled. DAC0 Output is pin active; DAC0 is operational.
Bits6-3: UNUSED. Read = 0000b; Write = don’t care
Bits2-0: DAC0DF2-0: DAC0 Data Format Bits
000: The most significant nybble of the DAC0 Data Word is in DAC0H[3:0], while the least significant byte is in DAC0L.

DAC0H								DAC0L							
▨	▨	▨	▨	MSB											LSB

001: The most significant 5-bits of the DAC0 Data Word is in DAC0H[4:0], while the least significant 7-bits is in DAC0L[7:1].

DAC0H								DAC0L							
▨	▨	▨	MSB											LSB	▨

010: The most significant 6-bits of the DAC0 Data Word is in DAC0H[5:0], while the least significant 6-bits is in DAC0L[7:2].

DAC0H								DAC0L							
▨	▨	MSB											LSB	▨	▨

011: The most significant 7-bits of the DAC0 Data Word is in DAC0H[6:0], while the least significant 5-bits is in DAC0L[7:3].

DAC0H								DAC0L							
▨	MSB											LSB	▨	▨	▨

1xx: The most significant byte of the DAC0 Data Word is in DAC0H, while the least significant nybble is in DAC0L[7:4].

DAC0H								DAC0L							
MSB											LSB	▨	▨	▨	▨

9. VOLTAGE REFERENCE

The voltage reference circuit consists of a 1.2V, 15ppm/°C (typical) bandgap voltage reference generator and a gain-of-two output buffer amplifier. The reference voltage on VREF can be connected to external devices in the system, as long as the maximum load seen by the VREF pin is less than 200µA to AGND (see Figure 9.1).

If a different reference voltage is required, an external reference can be connected to the VREF pin and the internal bandgap and buffer amplifier disabled in software. The external reference voltage must still be less than AV+ - 0.3V. The Reference Control Register, REF0CN (defined in Figure 9.2), provides the means to enable or disable the bandgap and buffer amplifier. The BIASE bit in REF0CN enables the bias circuitry for the ADC and DACs while the REFBE bit enables the bandgap reference and buffer amplifier which drive the VREF pin. When disabled, the supply current drawn by the bandgap and buffer amplifier falls to less than 1uA (typical) and the output of the buffer amplifier enters a high impedance state. If the internal bandgap is used as the reference voltage generator, BIASE and REFBE must both be set to 1. If an external reference is used, REFBE must be set to 0 and BIASE must be set to 1. If neither the ADC nor the DAC are being used, both of these bits can be set to 0 to conserve power. The electrical specifications for the Voltage Reference are given in Table 9.1.

The temperature sensor connects to the highest order input of the A/D converter’s input multiplexer (see Figure 5.1 and Figure 5.5 for details). The TEMPE bit within REF0CN enables and disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any A/D measurements performed on the sensor while disabled result in meaningless data.

Figure 9.1. Voltage Reference Functional Block Diagram

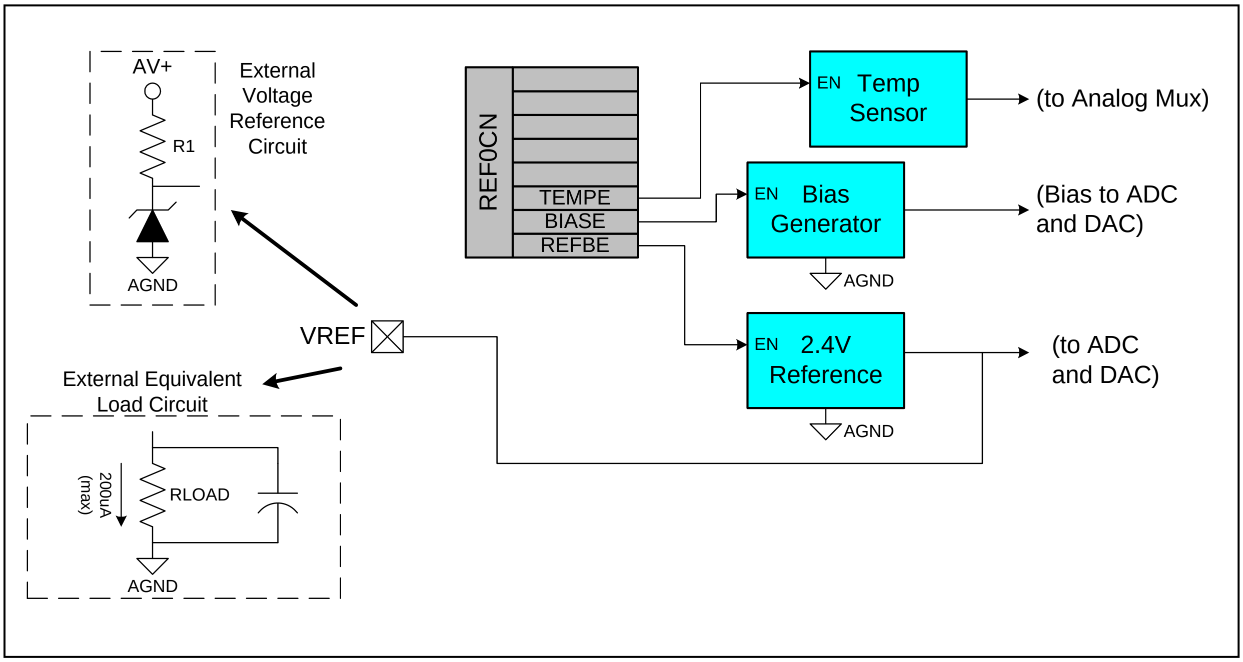


Table 10.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
ARITHMETIC OPERATIONS			
ADD A,Rn	Add register to A	1	1
ADD A,direct	Add direct byte to A	2	2
ADD A,@Ri	Add indirect RAM to A	1	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,direct	Add direct byte to A with carry	2	2
ADDC A,@Ri	Add indirect RAM to A with carry	1	2
ADDC A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal Adjust A	1	1
LOGICAL OPERATIONS			
ANL A,Rn	AND Register to A	1	1
ANL A,direct	AND direct byte to A	2	2
ANL A,@Ri	AND indirect RAM to A	1	2
ANL A,#data	AND immediate to A	2	2
ANL direct,A	AND A to direct byte	2	2
ANL direct,#data	AND immediate to direct byte	3	3
ORL A,Rn	OR Register to A	1	1
ORL A,direct	OR direct byte to A	2	2
ORL A,@Ri	OR indirect RAM to A	1	2
ORL A,#data	OR immediate to A	2	2
ORL direct,A	OR A to direct byte	2	2
ORL direct,#data	OR immediate to direct byte	3	3
XRL A,Rn	Exclusive-OR Register to A	1	1
XRL A,direct	Exclusive-OR direct byte to A	2	2
XRL A,@Ri	Exclusive-OR indirect RAM to A	1	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL direct,A	Exclusive-OR A to direct byte	2	2
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

Address	Register	Description	Page No.
0xEE	PCA0CPL4	PCA Capture Module 4 Data Word (Low Byte)	163
0xDA	PCA0CPM0	Programmable Counter Array 0 Capture/Compare 0	162
0xDB	PCA0CPM1	Programmable Counter Array 0 Capture/Compare 1	162
0xDC	PCA0CPM2	Programmable Counter Array 0 Capture/Compare 2	162
0xDD	PCA0CPM3	Programmable Counter Array 0 Capture/Compare 3	162
0xDE	PCA0CPM4	Programmable Counter Array 0 Capture/Compare 4	162
0xF9	PCA0H	PCA Counter/Timer Data Word (High Byte)	163
0xE9	PCA0L	PCA Counter/Timer Data Word (Low Byte)	163
0xD9	PCA0MD	Programmable Counter Array 0 Mode	161
0x87	PCON	Power Control	86
0xA4	PRT0CF	Port 0 Configuration	109
0xA5	PRT1CF	Port 1 Configuration	110
0xAD	PRT1IF	Port 1 Interrupt Flags	110
0xA6	PRT2CF	Port 2 Configuration	111
0xA7	PRT3CF	Port 3 Configuration	112
0x8F	PSCTL	Program Store RW Control	88
0xD0	PSW	Program Status Word	75
0xCB	RCAP2H	Counter/Timer 2 Capture (High Byte)	151
0xCA	RCAP2L	Counter/Timer 2 Capture (Low Byte)	151
0xD1	REF0CN	Voltage Reference Control Register	61
0xEF	RSTSRC	Reset Source Register	97
0x99	SBUF	Serial Data Buffer (UART)	136
0x98	SCON	Serial Port Control (UART)	137
0xC3	SMB0ADR	SMBus 0 Address	120
0xC0	SMB0CN	SMBus 0 Control	118
0xCF	SMB0CR	SMBus 0 Clock Rate	119
0xC2	SMB0DAT	SMBus 0 Data	120
0xC1	SMB0STA	SMBus 0 Status	121
0x81	SP	Stack Pointer	74
0x9A	SPI0CFG	Serial Peripheral Interface Configuration	127
0x9D	SPI0CKR	SPI Clock Rate	129
0xF8	SPI0CN	SPI Bus Control	128
0x9B	SPI0DAT	SPI Port 1Data	129
0xC8	T2CON	Counter/Timer 2 Control	150
0x88	TCON	Counter/Timer Control	142
0x8C	TH0	Counter/Timer 0 Data Word (High Byte)	145
0x8D	TH1	Counter/Timer 1 Data Word (High Byte)	145
0xCD	TH2	Counter/Timer 2 Data Word (High Byte)	151
0x8A	TL0	Counter/Timer 0 Data Word (Low Byte)	145
0x8B	TL1	Counter/Timer 1 Data Word (Low Byte)	145
0xCC	TL2	Counter/Timer 2 Data Word (Low Byte)	151

Address	**Register**	Description	Page No.
0x89	TMOD	Counter/Timer Mode	143
0x91	TMR3CN	Timer 3 Control	152
0x95	TMR3H	Timer 3 High	153
0x94	TMR3L	Timer 3 Low	153
0x93	TMR3RLH	Timer 3 Reload High	153
0x92	TMR3RLL	Timer 3 Reload Low	153
0xFF	WDTCN	Watchdog Timer Control	96
0xE1	XBR0	Port I/O Crossbar Configuration 1	105
0xE2	XBR1	Port I/O Crossbar Configuration 2	107
0xE3	XBR2	Port I/O Crossbar Configuration 3	108
0x84-86, 0x96-97, 0x9C, 0xA1-A3, 0xA9-AC, 0xAE, 0xB3-B5, 0xB9, 0xBD, 0xC9, 0xCE, 0xDF, 0xE4-E5, 0xF1-F5		Reserved	

* Refers to a register in the C8051F000/1/2/5/6/7 only.

** Refers to a register in the C8051F010/1/2/5/6/7 only.

*** Refers to a register in the C8051F005/06/07/15/16/17 only.

prohibited from reading locations in the lower partition using the MOVC instruction. (Executing a MOVC instruction from the upper partition with a source address in the lower partition will always return a data value of 0x00.) Software running in the lower partition can access locations in both the upper and lower partition without restriction.

The Value-added firmware should be placed in the lower partition. On reset, control is passed to the value-added firmware via the reset vector. Once the value-added firmware completes its initial execution, it branches to a predetermined location in the upper partition. If entry points are published, software running in the upper partition may execute program code in the lower partition, but it cannot read the contents of the lower partition. Parameters may be passed to the program code running in the lower partition either through the typical method of placing them on the stack or in registers before the call or by placing them in prescribed memory locations in the upper partition.

The SRL address is specified using the contents of the Flash Access Register. The 16-bit SRL address is calculated as 0xNN00, where NN is the contents of the SRL Security Register. Thus, the SRL can be located on 256-byte boundaries anywhere in program memory space. However, the 512-byte erase sector size essentially requires that a 512 boundary be used. The contents of a non-initialized SRL security byte is 0x00, thereby setting the SRL address to 0x0000 and allowing read access to all locations in program memory space by default.

Figure 11.3. FLACL: Flash Access Limit (C8051F005/06/07/15/16/17 only)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xB7

Bits 7-0: FLACL: Flash Access Limit.
This register holds the high byte of the 16-bit program memory read/write/erase limit address. The entire 16-bit access limit address value is calculated as 0xNN00 where NN is replaced by contents of FLACL. A write to this register sets the Flash Access Limit. **This register can only be written once after any reset. Any subsequent writes are ignored until the next reset.**

13.1. Power-on Reset

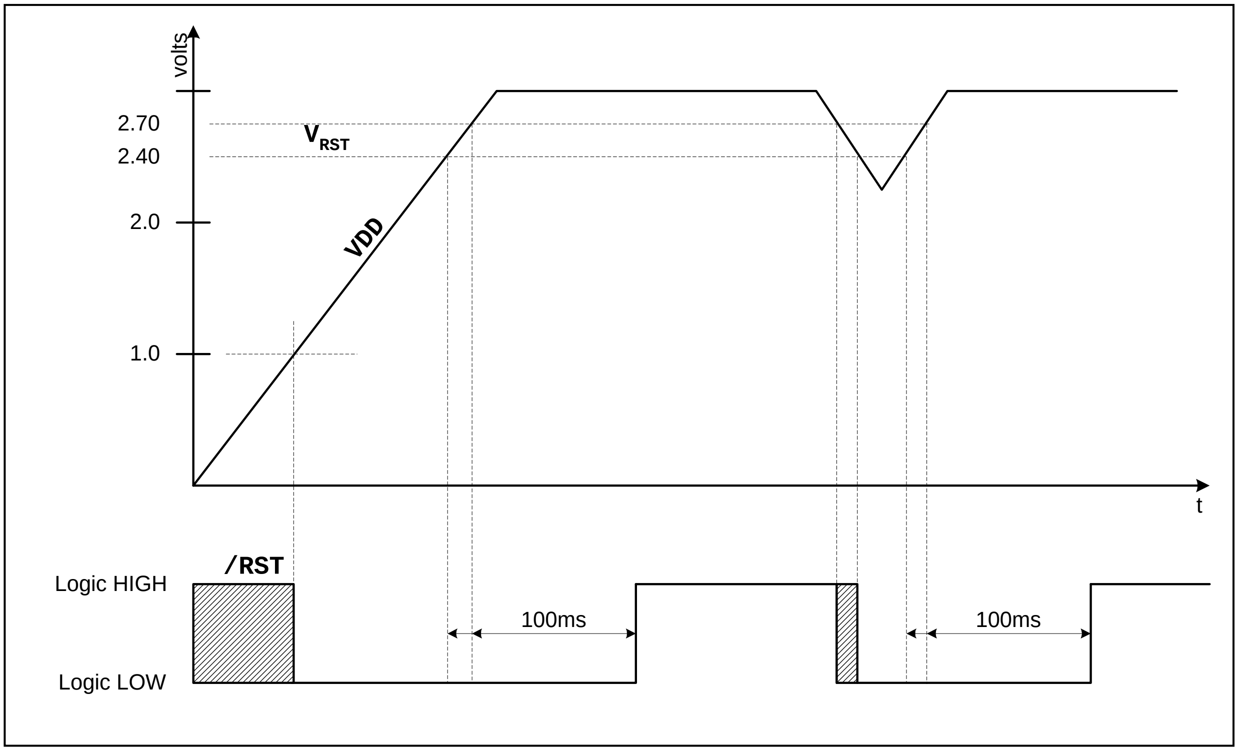
The C8051F000 family incorporates a power supply monitor that holds the MCU in the reset state until VDD rises above the V_{RST} level during power-up. (See Figure 13.2 for timing diagram, and refer to Table 13.1 for the Electrical Characteristics of the power supply monitor circuit.) The /RST pin is asserted (low) until the end of the 100ms VDD Monitor timeout in order to allow the VDD supply to become stable.

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

13.2. Software Forced Reset

Writing a 1 to the PORSF bit forces a Power-On Reset as described in Section 13.1.

Figure 13.2. VDD Monitor Timing Diagram



13.3. Power-fail Reset

When a power-down transition or power irregularity causes VDD to drop below V_{RST}, the power supply monitor will drive the /RST pin low and return the CIP-51 to the reset state (see Figure 13.2). When VDD returns to a level above V_{RST}, the CIP-51 will leave the reset state in the same manner as that for the power-on reset. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if VDD dropped below the level required for data retention. If the PORSF flag is set, the data may no longer be valid.

15. PORT INPUT/OUTPUT

The MCUs have a wide array of digital resources, which are available through four digital I/O ports, P0, P1, P2 and P3. Each of the pins on Ports 0, 1, and 2 can be defined as either its corresponding port I/O or one of the internal digital resources assigned as shown in Figure 15.1. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins available on the selected package (the C8051F000/05/10/15 have all four ports pinned out, the F001/06/11/16 have P0 and P1, and the F002/07/12/17 have P0). This resource assignment flexibility is achieved through the use of a Priority CrossBar Decoder. (Note that the state of a Port I/O pin can always be read in the corresponding Port latch regardless of the Crossbar settings).

The CrossBar assigns the selected internal digital resources to the I/O pins based on the Priority Decode Table 15.1. The registers XBR0, XBR1, and XBR2, defined in Figure 15.3, Figure 15.4, and Figure 15.5 are used to select an internal digital function or let an I/O pin default to being a Port I/O. The crossbar functions identically for each MCU, with the caveat that P2 is not pinned out on the F001/06/11/16, and both P1 and P2 are not pinned out on the F002/07/12/17. Digital resources assigned to port pins that are not pinned out cannot be accessed.

All Port I/Os are 5V tolerant (Refer to Figure 15.2 for the port cell circuit.) The Port I/O cells are configured as either push-pull or open-drain in the Port Configuration Registers (PRT0CF, PRT1CF, PRT2CF, PRT3CF). Complete Electrical Specifications for Port I/O are given in Table 15.2.

15.1. Priority Cross Bar Decoder

One of the design goals of this MCU family was to make the entire palette of digital resources available to the designer even on reduced pin count packages. The Priority CrossBar Decoder provides an elegant solution to the problem of connecting the internal digital resources to the physical I/O pins.

The Priority CrossBar Decode (Table 15.1) assigns a priority to each I/O function, starting at the top with the SMBus. As the table illustrates, when selected, its two signals will be assigned to Pin 0 and 1 of I/O Port 0. The decoder always fills I/O bits from LSB to MSB starting with Port 0, then Port 1, finishing if necessary with Port 2. If you choose not to use a resource, the next function down on the table will fill the priority slot. In this way it is possible to choose only the functions required by the design, making full use of the available I/O pins. Also, any extra Port I/O are grouped together for more convenient use in application code.

Registers XBR0, XBR1 and XBR2 are used to assign the digital I/O resources to the physical I/O Port pins. It is important to understand that when the SMBus, SPI Bus, or UART is selected, the crossbar assigns all pins associated with the selected bus. It would be impossible for instance to assign the RX pin from the UART function without also assigning the TX function. Standard Port I/Os appear contiguously after the prioritized functions have been assigned. For example, if you choose functions that take the first 14 Port I/O (P0.[7:0], P1.[5:0]), you would have 18 Port I/O left unused by the crossbar (P1.[7:6], P2 and P3).

15.2. Port I/O Initialization

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

The output driver characteristics of the I/O pins are defined using the Port Configuration Registers PRT0CF, PRT1CF, PRT2CF and PRT3CF (see Figure 15.7, Figure 15.9, Figure 15.12, and Figure 15.14). Each Port Output driver can be configured as either Open Drain or Push-Pull. This is required even for the digital resources selected in the XBRn registers and is not automatic. The only exception to this is the SMBus (SDA, SCL) and UART Receive (RX, when in mode 0) pins which are Open-drain regardless of the PRTnCF settings. When the WEAKPUD bit in XBR2 is 0, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does

Figure 15.5. XBR2: Port I/O CrossBar Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
WEAKPUD	XBARE	-	-	-	-	-	CNVSTE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3

Bit7: WEAKPUD: Port I/O Weak Pull-up Disable Bit
0: Weak Pull-ups Enabled (except for Ports whose I/O are configured as push-pull)
1: Weak Pull-ups Disabled
Bit6: XBARE: Crossbar Enable Bit
0: Crossbar Disabled
1: Crossbar Enabled
Bits5-1: UNUSED. Read = 00000b, Write = don't care.
Bit0: CNVSTE: ADC Convert Start Input Enable Bit
0: CNVSTR unavailable at Port pin.
1: CNVSTR routed to Port Pin.

Example Usage of XBR0, XBR1, XBR2:
When selected, the digital resources fill the Port I/O pins in order (top to bottom as shown in Table 15.1) starting with P0.0 through P0.7, and then P1.0 through P1.7, and finally P2.0 through P2.7. If the digital resources are not mapped to the Port I/O pins, they default to their matching internal Port Register bits.

Example1: If XBR0 = 0x11, XBR1 = 0x00, and XBR2 = 0x40:
P0.0=SDA, P0.1=SCL, P0.2=CEX0, P0.3=CEX1, P0.4 … P2.7 map to corresponding Port I/O.

Example2: If XBR0 = 0x80, XBR1 = 0x04, and XBR2 = 0x41:
P0.0=CP0, P0.1=/INT0, P0.2 = CNVSTR, P0.3 … P2.7 map to corresponding Port I/O.

15.3. General Purpose Port I/O

Each MCU has four byte-wide, bi-directional parallel ports that can be used general purpose I/O. Each port is accessed through a corresponding special function register (SFR) that is both byte addressable and bit 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 still 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 SET, when the destination is an individual bit in a port SFR. For these instructions, the value of the port register (not the pin) is read, modified, and written back to the SFR.

15.4. Configuring Ports Which are not Pinned Out

P2 and P3 are not pinned out on the F001/06/11/16. P1, P2, and P3 are not pinned out on the F002/07/12/17. These port registers (and corresponding interrupts, where applicable) are still available for software use in these reduced pin count MCUs. Whether used or not in software, it is recommended not to let these port drivers go to high impedance state. This is prevented after reset by having the weak pull-ups enabled as described in the XBR2 register. It is recommended that each output driver for ports not pinned out should be configured as push-pull using the corresponding PRTnCF register. This will inhibit a high impedance state even if the weak pull-up is disabled.

Figure 15.6. P0: Port0 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (bit addressable)	SFR Address: 0x80

Bits7-0: P0.[7:0]
(Write – Output appears on I/O pins per XBR0, XBR1, and XBR2 Registers)
0: Logic Low Output.
1: Logic High Output (high-impedance if corresponding PRT0CF.n bit = 0)
(Read – Regardless of XBR0, XBR1, and XBR2 Register settings).
0: P0.n pin is logic low.
1: P0.n pin is logic high.

Figure 15.7. PRT0CF: Port0 Configuration Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xA4

Bits7-0: PRT0CF.[7:0]: Output Configuration Bits for P0.7-P0.0 (respectively)
0: Corresponding P0.n Output mode is Open-Drain.
1: Corresponding P0.n Output mode is Push-Pull.

(Note: When SDA, SCL, and RX appear on any of the P0 I/O, each are open-drain regardless of the value of PRT0CF).

16.2.4. Slave Receiver Mode

Serial data is received on SDA while the serial clock is received on SCL. First, a byte is received that contains an address and data direction bit. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. If the received address matches the slave's assigned address (or a general call address is received) one or more bytes of serial data are received from the master. After each byte is received, an acknowledge bit is transmitted by the slave. The master outputs START and STOP conditions to indicate the beginning and end of the serial transfer.

16.3. 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 remains high for a specified time. Two or more master devices may attempt to generate a START condition at the same time. Since the devices that generated the START condition may not be aware that other masters are contending for the bus, an arbitration scheme is employed. The master devices continue to transmit until one of the masters transmits a HIGH level, while the other(s) master transmits a LOW level on SDA. The first master(s) transmitting the HIGH level on SDA looses the arbitration and is required to give up the bus.

16.4. 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 can hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

16.5. Timeouts

16.5.1. 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 25ms as a "timeout" condition. Devices that have detected the timeout condition must reset the communication no later than 10ms after detecting the timeout condition.

One of the MCU's general-purpose timers, operating in 16-bit auto-reload mode, can be used to monitor the SCL line for this timeout condition. Timer 3 is specifically designed for this purpose. (Refer to the Timer 3 Section 19.3. for detailed information on Timer 3 operation.)

16.5.2. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if a device holds the SCL and SDA lines high for more that 50usec, the bus is designated as free. The SMB0CR register is used to detect this condition when the FTE bit in SMB0CN is set.

16.6. SMBus Special Function Registers

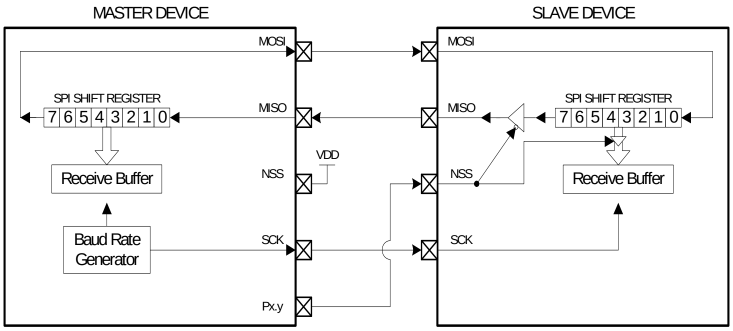
The SMBus serial interface is accessed and controlled through five SFRs: SMB0CN Control Register, SMB0CR Clock Rate Register, SMB0ADR Address Register, SMB0DAT Data Register and SMB0STA Status Register. The system device may have one or more SMBus serial interfaces implemented. The five special function registers related to the operation of the SMBus interface are described in the following section.

17.2. Operation

Only a SPI master device can initiate a data transfer. The SPI is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.1). Writing a byte of data to the SPI data register (SPI0DAT) when in Master Mode starts a data transfer. The SPI master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. The SPI master can be configured to shift in/out from one to eight bits in a transfer operation in order to accommodate slave devices with different word lengths. The SPIFRS bits in the SPI Configuration Register (SPI0CFG.[2:0]) are used to select the number of bits to shift in/out in a transfer operation.

While the SPI master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. The data byte received from the slave replaces the data in the master's data register. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data transfer in both directions is synchronized with the serial clock generated by the master. Figure 17.3 illustrates the full-duplex operation of an SPI master and an addressed slave.

Figure 17.3. Full Duplex Operation



The SPI data register is double buffered on reads, but not on a write. If a write to SPI0DAT is attempted during a data transfer, the WCOL flag (SPI0CN.6) will be set to logic 1 and the write is ignored. The current data transfer will continue uninterrupted. A read of the SPI data register by the system controller actually reads the receive buffer. If the receive buffer still holds unread data from a previous transfer when the last bit of the current transfer is shifted into the SPI shift register, a receive overrun occurs and the RXOVRN flag (SPI0CN.4) is set to logic 1. The new data is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte causing the overrun is lost.

When the SPI is enabled and not configured as a master, it will operate as an SPI slave. Another SPI device acting as a master will initiate a transfer by driving the NSS signal low. The master then shifts data out of the shift register on the MOSI pin using the its serial clock. The SPIF flag is set to logic 1 at the end of a data transfer (when the NSS signal goes high). The slave can load its shift register for the next data transfer by writing to the SPI data register. The slave must make the write to the data register at least one SPI serial clock cycle before the master starts the next transmission. Otherwise, the byte of data already in the slave's shift register will be transferred.

Multiple masters may reside on the same bus. A Mode Fault flag (MODF, SPI0CN.5) is set to logic 1 when the SPI is configured as a master (MSTEN = 1) and its slave select signal NSS is pulled low. When the Mode Fault flag is set, the MSTEN and SPIEN bits of the SPI control register are cleared by hardware, thereby placing the SPI module

17.4. SPI Special Function Registers

The SPI is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI Bus are described in the following section.

Figure 17.5. SPI0CFG: SPI Configuration Register

R/W	R/W	R	R	R	R/W	R/W	R/W	Reset Value
CKPHA	CKPOL	BC2	BC1	BC0	SPIFRS2	SPIFRS1	SPIFRS0	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9A

Bit7: CKPHA: SPI Clock Phase.
This bit controls the SPI clock phase.
0: Data sampled on first edge of SCK period.
1: Data sampled on second edge of SCK period.

Bit6: CKPOL: SPI Clock Polarity.
This bit controls the SPI clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bits5-3: BC2-BC0: SPI Bit Count.
Indicates which of the up to 8 bits of the SPI word have been transmitted.

BC2-BC0			**Bit Transmitted**
0	0	0	Bit 0 (LSB)
0	0	1	Bit 1
0	1	0	Bit 2
0	1	1	Bit 3
1	0	0	Bit 4
1	0	1	Bit 5
1	1	0	Bit 6
1	1	1	Bit 7 (MSB)

Bits2-0: SPIFRS2-SPIFRS0: SPI Frame Size.
These three bits determine the number of bits to shift in/out of the SPI shift register during a data transfer in master mode. They are ignored in slave mode.

SPIFRS			**Bits Shifted**
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

.

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is 0 or the input signal /INT0 is logic-level one. Setting GATE0 to logic 1 allows the timer to be controlled by the external input signal /INT0, facilitating pulse width measurements.

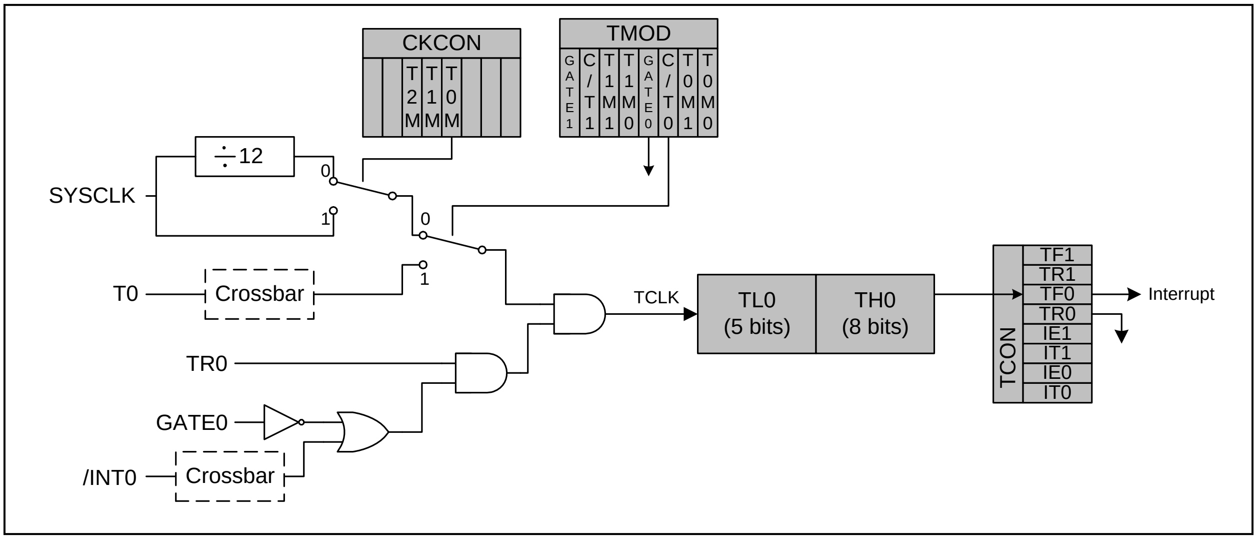
TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled

X = Don’t Care

Setting TR0 does not reset the timer register. The timer register should be initialized to the desired value before enabling the timer.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0.

Figure 19.1. T0 Mode 0 Block Diagram



19.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

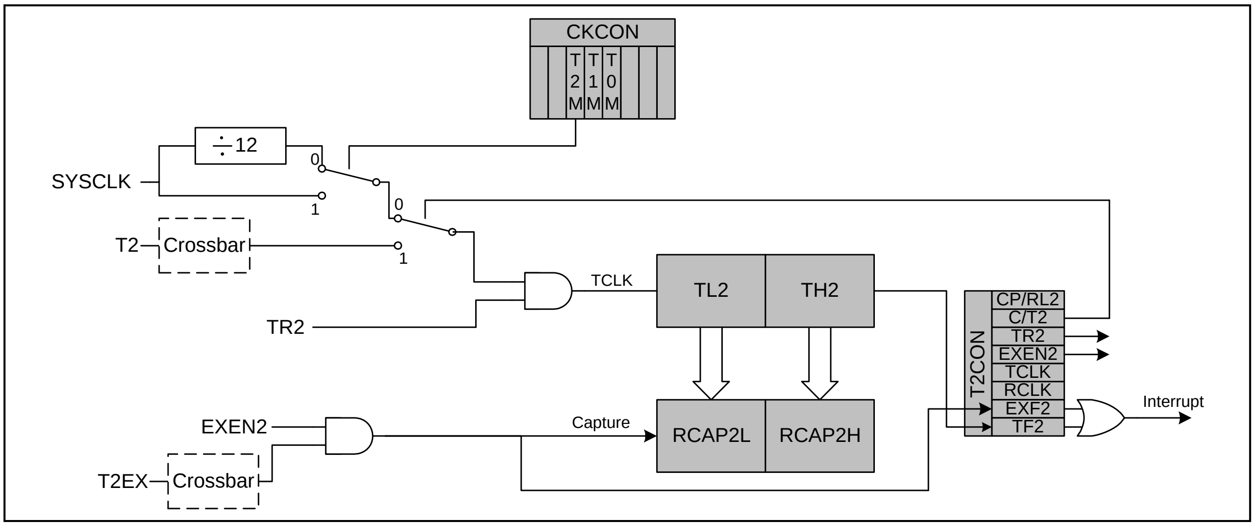
19.2.1. Mode 0: 16-bit Counter/Timer with Capture

In this mode, Timer 2 operates as a 16-bit counter/timer with capture facility. A high-to-low transition on the T2EX input pin causes the 16-bit value in Timer 2 (TH2, TL2) to be loaded into the capture registers (RCAP2H, RCAP2L).

Timer 2 can use either SYSCLK, SYSCLK divided by 12, or high-to-low transitions on the external T2 pin as its clock source when operating in Counter/Timer with Capture mode. Clearing the C/T2 bit (T2CON.1) selects the system clock as the input for the timer (divided by one or twelve as specified by the Timer Clock Select bit T2M in CKCON). When C/T2 is set to logic 1, a high-to-low transition at the T2 input pin increments the counter/timer register. As the 16-bit counter/timer register increments and overflows from 0xFFFF to 0x0000, the TF2 timer overflow flag (T2CON.7) is set and an interrupt will occur if the interrupt is enabled.

Counter/Timer with Capture mode is selected by setting the Capture/Reload Select bit CP/RL2 (T2CON.0) and the Timer 2 Run Control bit TR2 (T2CON.2) to logic 1. The Timer 2 External Enable EXEN2 (T2CON.3) must also be set to logic 1 to enable a capture. If EXEN2 is cleared, transitions on T2EX will be ignored.

Figure 19.11. T2 Mode 0 Block Diagram



21.1. Boundary Scan

The Data Register in the Boundary Scan path is an 87-bit shift register. The Boundary DR provides control and observability of all the device pins as well as the SFR bus and Weak Pullup feature via the EXTEST and SAMPLE commands.

Table 21.1. Boundary Data Register Bit Definitions

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

Bit	Action	Target
0	Capture	Reset Enable from MCU
	Update	Reset Enable to /RST pin
1	Capture	Reset input from /RST pin
	Update	Reset output to /RST pin
2	Capture	External Clock from XTAL1 pin
	Update	Not used
3	Capture	Weak pullup enable from MCU
	Update	Weak pullup enable to Port Pins
4-11	Capture	SFR Address Bus bit from CIP-51 (e.g. Bit4=SFRA0, Bit5=SFRA1…)
	Update	SFR Address Bus bit to SFR Address Bus (e.g. Bit4=XSFRA0, Bit5=XSFRA1)
12-19	Capture	SFR Data Bus bit read from SFR (e.g. Bit12=SFRD0, Bit13=SFRD1…)
	Update	SFR Data Bus bit written to SFR (e.g. Bit12=SFRD0, Bit13=SFRD1…)
20	Capture	SFR Write Strobe from CIP-51
	Update	SFR Write Strobe to SFR Bus
21	Capture	SFR Read Strobe from CIP-51
	Update	SFR Read Strobe to SFR Bus
22	Capture	SFR Read/Modify/Write Strobe from CIP-51
	Update	SFR Read/Modify/Write Strobe to SFR Bus
23,25,27,29, 31,33,35,37	Capture	P0.n output enable from MCU (e.g. Bit23=P0.0, Bit25=P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit23=P0.0oe, Bit25=P0.1oe, etc.)
24,26,28,30, 32,34,36,38	Capture	P0.n input from pin (e.g. Bit24=P0.0, Bit26=P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit24=P0.0, Bit26=P0.1, etc.)
39,41,43,45, 47,49,51,53	Capture	P1.n output enable from MCU (e.g. Bit39=P1.0, Bit41=P1.1, etc.)
	Update	P1.n output enable to pin (e.g. Bit39=P1.0oe, Bit41=P1.1oe, etc.)
40,42,44,46, 48,50,52,54	Capture	P1.n input from pin (e.g. Bit40=P1.0, Bit42=P1.1, etc.)
	Update	P1.n output to pin (e.g. Bit40=P1.0, Bit42=P1.1, etc.)
55,57,59,61, 63,65,67,69	Capture	P2.n output enable from MCU (e.g. Bit55=P2.0, Bit57=P2.1, etc.)
	Update	P2.n output enable to pin (e.g. Bit55=P2.0oe, Bit57=P2.1oe, etc.)
56,58,60,62, 64,66,68,70	Capture	P2.n input from pin (e.g. Bit56=P2.0, Bit58=P2.1, etc.)
	Update	P2.n output to pin (e.g. Bit56=P2.0, Bit58=P2.1, etc.)
71,73,75,77, 79,81,83,85	Capture	P3.n output enable from MCU (e.g. Bit71=P3.0, Bit73=P3.1, etc.)
	Update	P3.n output enable to pin (e.g. Bit71=P3.0oe, Bit73=P3.1oe, etc.)
72,74,76,78, 80,82,84,86	Capture	P3.n input from pin (e.g. Bit72=P3.0, Bit74=P3.1, etc.)
	Update	P3.n output to pin (e.g. Bit72=P3.0, Bit74=P3.1, etc.)