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	Active
Core Processor	8051
Core Size	8-Bit
Speed	48 MIPS
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	40
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.25V
Data Converters	A/D 32x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f38a-b-gq

List of Figures

5. Electrical Characteristics

5.1. Absolute Maximum Specifications

Table 5.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Junction Temperature Under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on $\overline{RST}$, INT2, or any Port I/O Pin with Respect to GND	$V_{DD} \geq 2.2$ V $V_{DD} < 2.2$ V	–0.3 –0.3	— —	5.8 V_{DD} + 3.6	V V
Voltage on V_{DD} with Respect to GND	Regulator1 in Normal Mode Regulator1 in Bypass Mode	–0.3 –0.3	— —	4.2 1.98	V V
Maximum Total Current through V_{DD} or GND		—	—	500	mA
Maximum Output Current sunk by $\overline{RST}$ or any Port Pin		—	—	100	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Table 5.5. Internal Voltage Regulator Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Test Condition	Min	Typ	Max	Unit
Voltage Regulator (REG0)					
Input Voltage Range[1]		2.7	—	5.25	V
Output Voltage (V_{DD})[2]	Output Current = 1 to 100 mA	3.0	3.3	3.6	V
Output Current[2]		—	—	100	mA
Dropout Voltage (V_{DO})[3]		—	1	—	mV/mA
Voltage Regulator (REG1)					
Input Voltage Range		1.8	—	3.6	V

Notes:
1. Input range specified for regulation. When an external regulator is used, should be tied to V_{DD}.
2. Output current is total regulator output, including any current required by the C8051F388/9/A/B.
3. The minimum input voltage is 2.70 V or V_{DD} + V_{DO} (max load), whichever is greater.

Table 5.6. Flash Electrical Characteristics

Parameter	Test Condition	Min	Typ	Max	Unit
Flash Size	C8051F388/9* C8051F38A/B	65536* 32768	—	—	Bytes Bytes
Endurance		10k	100k	—	Erase/Write
Erase Cycle Time	25 MHz System Clock	10	15	22.5	ms
Write Cycle Time	25 MHz System Clock	10	15	20	µs

Notes:
1. 1024 bytes at location 0xFC00 to 0xFFFF are not available for program storage.
2. Data Retention Information is published in the Quarterly Quality and Reliability Report.

SFR Definition 6.9. AMX0P: AMUX0 Positive Channel Select

Bit	7	6	5	4	3	2	1	0
Name			AMX0P[5:0]					
Type	R	R	R/W					
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xBB; SFR Page = All Pages

Bit	Name	Function					
7:6	Unused	Read = 00b; Write = don't care.					
5:0	AMX0P[5:0]	**AMUX0 Positive Input Selection.**					
		AMX0P	32-pin Packages	48-pin Packages	AMX0P	32-pin Packages	48-pin Packages
		000000:	P1.0	P2.0	010010:	P0.1	P0.4
		000001:	P1.1	P2.1	010011:	P0.4	P1.1
		000010:	P1.2	P2.2	010100:	P0.5	P1.2
		000011:	P1.3	P2.3	010101:	Reserved	P1.0
		000100:	P1.4	P2.5	010110:	Reserved	P1.3
		000101:	P1.5	P2.6	010111:	Reserved	P1.6
		000110:	P1.6	P3.0	011000:	Reserved	P1.7
		000111:	P1.7	P3.1	011001:	Reserved	P2.4
		001000:	P2.0	P3.4	011010:	Reserved	P2.7
		001001:	P2.1	P3.5	011011:	Reserved	P3.2
		001010:	P2.2	P3.7	011100:	Reserved	P3.3
		001011:	P2.3	P4.0	011101:	Reserved	P3.6
		001100:	P2.4	P4.3	011110:	Temp Sensor	Temp Sensor
		001101:	P2.5	P4.4	011111:	V_{DD}	V_{DD}
		001110:	P2.6	P4.5	100000:	Reserved	P4.1
		001111:	P2.7	P4.6	100001:	Reserved	P4.2
		010000:	P3.0	Reserved	100010:	Reserved	P4.7
		010001:	P0.0	P0.3	100011 - 111111:	Reserved	Reserved

SFR Definition 8.4. CPT1MD: Comparator1 Mode Selection

Bit	7	6	5	4	3	2	1	0
Name			CP1RIE	CP1FIE			CP1MD[1:0]	
Type	R	R	R/W	R/W	R	R	R/W	
Reset	0	0	0	0	0	0	1	0

SFR Address = 0x9C; SFR Page = All Pages

Bit	Name	Function
7:6	Unused	Read = 00b, Write = don't care.
5	CP1RIE	**Comparator1 Rising-Edge Interrupt Enable.** 0: Comparator1 Rising-edge interrupt disabled. 1: Comparator1 Rising-edge interrupt enabled.
4	CP1FIE	**Comparator1 Falling-Edge Interrupt Enable.** 0: Comparator1 Falling-edge interrupt disabled. 1: Comparator1 Falling-edge interrupt enabled.
3:2	Unused	Read = 00b, Write = don't care.
1:0	CP1MD[1:0]	**Comparator1 Mode Select.** These bits affect the response time and power consumption for Comparator1. 00: Mode 0 (Fastest Response Time, Highest Power Consumption) 01: Mode 1 10: Mode 2 11: Mode 3 (Slowest Response Time, Lowest Power Consumption)

SFR Definition 11.3. SP: Stack Pointer

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

SFR Address = 0x81; SFR Page = All Pages

Bit	Name	Function
7:0	SP[7:0]	**Stack Pointer.** The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 11.4. ACC: Accumulator

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

SFR Address = 0xE0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7:0	ACC[7:0]	**Accumulator.** This register is the accumulator for arithmetic operations.

SFR Definition 11.5. B: B Register

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

SFR Address = 0xF0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7:0	B[7:0]	**B Register.** This register serves as a second accumulator for certain arithmetic operations.

14.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 101 or 111

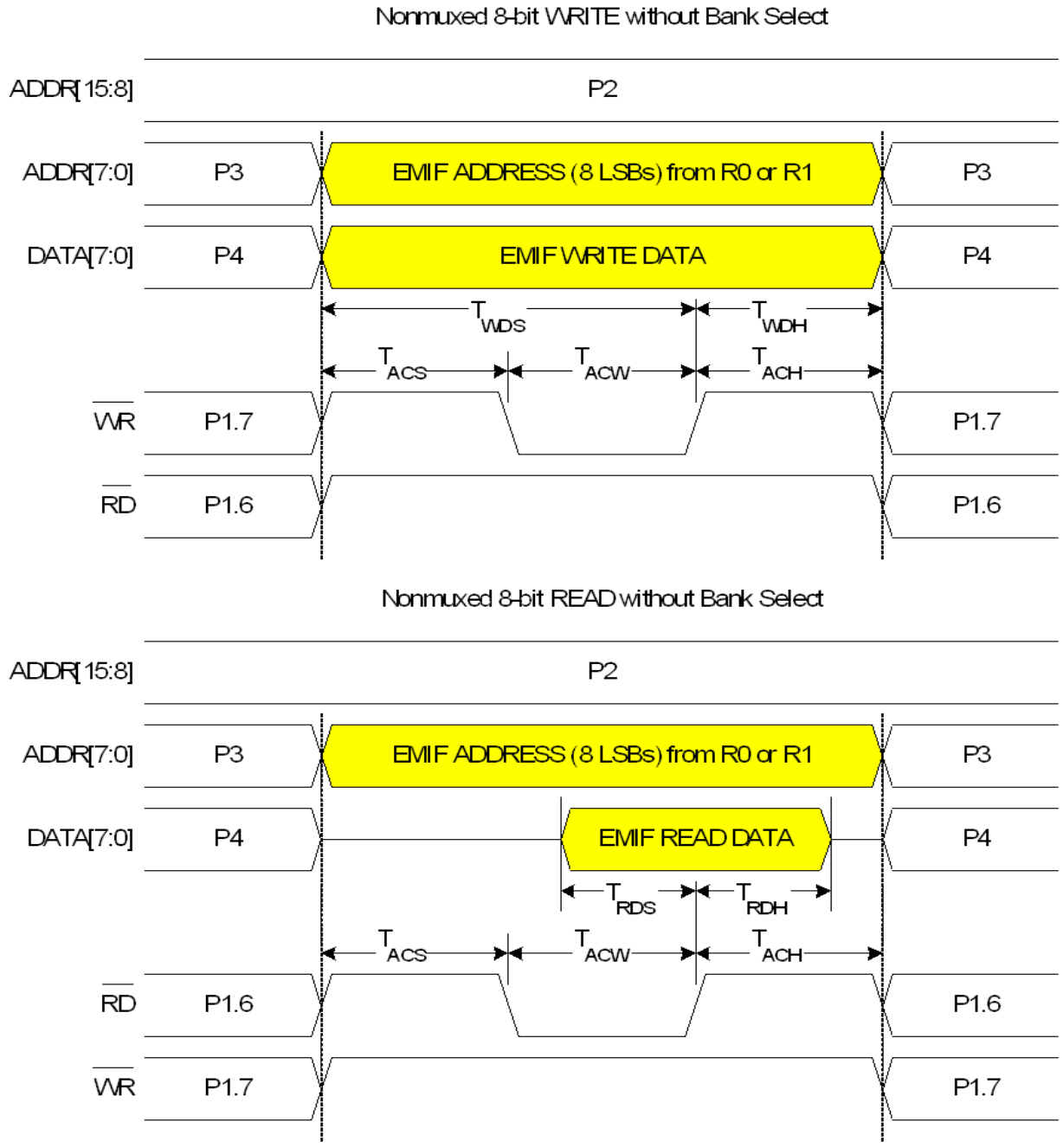

Figure 14.5. Non-Multiplexed 8-bit MOVX without Bank Select Timing

Table 15.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Page	Description	Page
SMB1CN	0xC0	F	SMBus1 Control	177
SMB1DAT	0xC2	F	SMBus1 Data	183
SMBTC	0xB9	F	SMBus0/1 Timing Control	174
SMOD1	0xE5	All Pages	UART1 Mode	207
SP	0x81	All Pages	Stack Pointer	82
SPI0CFG	0xA1	All Pages	SPI Configuration	218
SPI0CKR	0xA2	All Pages	SPI Clock Rate Control	220
SPI0CN	0xF8	All Pages	SPI Control	219
SPI0DAT	0xA3	All Pages	SPI Data	220
TCON	0x88	All Pages	Timer/Counter Control	231
TH0	0x8C	All Pages	Timer/Counter 0 High	234
TH1	0x8D	All Pages	Timer/Counter 1 High	234
TL0	0x8A	All Pages	Timer/Counter 0 Low	233
TL1	0x8B	All Pages	Timer/Counter 1 Low	233
TMOD	0x89	All Pages	Timer/Counter Mode	232
TMR2CN	0xC8	0	Timer/Counter 2 Control	239
TMR2H	0xCD	0	Timer/Counter 2 High	241
TMR2L	0xCC	0	Timer/Counter 2 Low	240
TMR2RLH	0xCB	0	Timer/Counter 2 Reload High	240
TMR2RLL	0xCA	0	Timer/Counter 2 Reload Low	240
TMR3CN	0x91	0	Timer/Counter 3 Control	246
TMR3H	0x95	0	Timer/Counter 3 High	248
TMR3L	0x94	0	Timer/Counter 3 Low	247
TMR3RLH	0x93	0	Timer/Counter 3 Reload High	247
TMR3RLL	0x92	0	Timer/Counter 3 Reload Low	247
TMR4CN	0x91	F	Timer/Counter 4 Control	251
TMR4H	0x95	F	Timer/Counter 4 High	253
TMR4L	0x94	F	Timer/Counter 4 Low	252
TMR4RLH	0x93	F	Timer/Counter 4 Reload High	252
TMR4RLL	0x92	F	Timer/Counter 4 Reload Low	252
TMR5CN	0xC8	F	Timer/Counter 5 Control	256
TMR5H	0xCD	F	Timer/Counter 5 High	258
TMR5L	0xCC	F	Timer/Counter 5 Low	257
TMR5RLH	0xCB	F	Timer/Counter 5 Reload High	257

16.3. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ External Interrupt Sources

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “25.1. Timer 0 and Timer 1” on page 227) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{\text{INT0}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	$\overline{\text{INT1}}$ Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 16.7). Note that $\overline{\text{INT0}}$ and $\overline{\text{INT0}}$ Port pin assignments are independent of any Crossbar assignments. $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to $\overline{\text{INT0}}$ and/or $\overline{\text{INT1}}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register PnSKIP (see Section “20.1. Priority Crossbar Decoder” on page 148 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupts, respectively. If an $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$ external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

18. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation.

18.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “27. C2 Interface” on page 277.

To ensure the integrity of Flash contents, it is strongly recommended that the V_{DD} monitor be left enabled in any system which writes or erases Flash memory from code. It is also crucial to ensure that the FLRT bit in register FLSCL be set to '1' if a clock speed higher than 25 MHz is being used for the device.

18.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 18.2.

18.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY); and (2) Setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed must be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

1. Disable interrupts (recommended).
2. Write the first key code to FLKEY: 0xA5.
3. Write the second key code to FLKEY: 0xF1.
4. Set the PSEE bit (register PSCTL).
5. Set the PSWE bit (register PSCTL).
6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
7. Clear the PSWE bit (register PSCTL).
8. Clear the PSEE bit (register PSCTI).

20. Port Input/Output

Digital and analog resources are available through 40 I/O pins (C8051F388/A) or 25 I/O pins (C8051F389/B). Port pins are organized as shown in Figure 20.1. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input; Port pins P0.0-P3.7 can be assigned to one of the internal digital resources as shown in Figure 20.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. 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 Decoder (Figure 20.3 and Figure 20.4). The registers XBR0, XBR1, and XBR2 defined in SFR Definition 20.1, SFR Definition 20.2, and SFR Definition 20.3, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 20.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1,2,3,4).

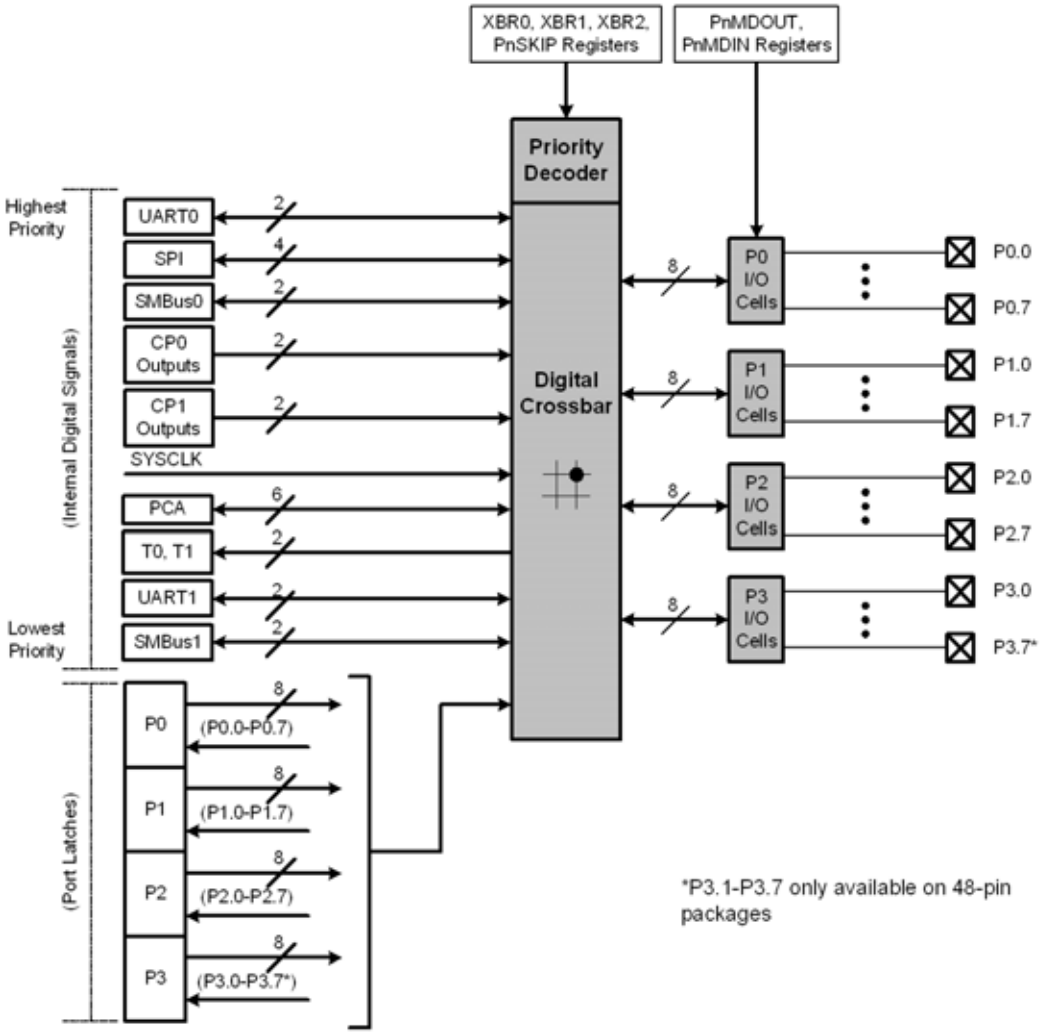

Figure 20.1. Port I/O Functional Block Diagram (Port 0 through Port 3)

SFR Definition 20.10. P1MDOUT: Port 1 Output Mode

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

SFR Address = 0xA5; SFR Page = All Pages

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

SFR Definition 20.11. P1SKIP: Port 1 Skip

Bit	7	6	5	4	3	2	1	0
Name	P1SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD5; SFR Page = All Pages

Bit	Name	Function
7:0	P1SKIP[7:0]	**Port 1 Crossbar Skip Enable Bits.** These bits select Port 1 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P1.n pin is not skipped by the Crossbar. 1: Corresponding P1.n pin is skipped by the Crossbar.

21.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

21.2. SMBus Configuration

Figure 21.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

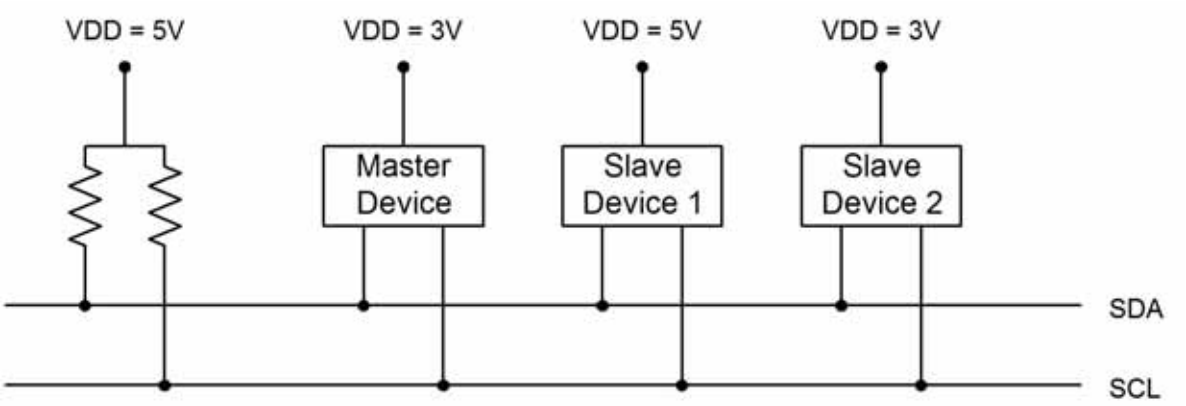

Figure 21.2. Typical SMBus Configuration

21.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. It is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 21.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

Table 21.3. Sources for Hardware Changes to SMBnCN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTERn	■ A START is generated.	■ A STOP is generated. ■ Arbitration is lost.
TXMODEn	■ START is generated. ■ SMBnDAT is written before the start of an SMBus frame.	■ A START is detected. ■ Arbitration is lost. ■ SMBnDAT is not written before the start of an SMBus frame.
STAn	■ A START followed by an address byte is received.	■ Must be cleared by software.
STOn	■ A STOP is detected while addressed as a slave. ■ Arbitration is lost due to a detected STOP.	■ A pending STOP is generated.
ACKRQn	■ A byte has been received and an ACK response value is needed (only when hardware ACK is not enabled).	■ After each ACK cycle.
ARBLOSTn	■ A repeated START is detected as a MASTER when STAn is low (unwanted repeated START). ■ SCLn is sensed low while attempting to generate a STOP or repeated START condition. ■ SDAn is sensed low while transmitting a 1 (excluding ACK bits).	■ Each time SIn is cleared.
ACKn	■ The incoming ACK value is low (ACKNOWLEDGE).	■ The incoming ACK value is high (NOT ACKNOWLEDGE).
SIn	■ A START has been generated. ■ Lost arbitration. ■ A byte has been transmitted and an ACK/NACK received. ■ A byte has been received. ■ A START or repeated START followed by a slave address + R/W has been received. ■ A STOP has been received.	■ Must be cleared by software.

21.4.4. Hardware Slave Address Recognition

The SMBus hardware has the capability to automatically recognize incoming slave addresses and send an ACK without software intervention. Automatic slave address recognition is enabled by setting the EHACK bit in register SMB0ADM to 1. This will enable both automatic slave address recognition and automatic hardware ACK generation for received bytes (as a master or slave). More detail on automatic hardware ACK generation can be found in Section 21.4.3.2.

The registers used to define which address(es) are recognized by the hardware are the SMBus Slave Address register and the SMBus Slave Address Mask register. A single address or range of addresses (including the General Call Address 0x00) can be specified using these two registers. The most-significant seven bits of the two registers are used to define which addresses will be ACKed. A 1 in bit positions of the slave address mask SLVM[6:0] enable a comparison between the received slave address and the hardware's slave address SLV[6:0] for those bits. A 0 in a bit of the slave address mask means that bit will be treated as a "don't care" for comparison purposes. In this case, either a 1 or a 0 value are acceptable on

21.5.3. 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. The interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 21.7 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.

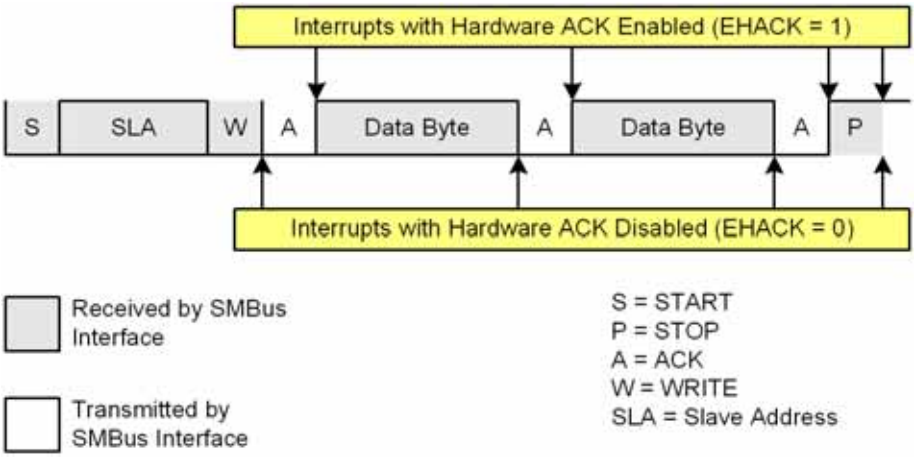

Figure 21.7. Typical Slave Write Sequence

25. Timers

Each MCU includes six counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and four are 16-bit auto-reload timer for use with the SMBus or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2, 3, 4, and 5 offer 16-bit and split 8-bit timer functionality with auto-reload.

Timer 0 and Timer 1 Modes:	Timer 2, 3, 4, and 5 Modes:
13-bit counter/timer	16-bit timer with auto-reload
16-bit counter/timer	
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)	

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 25.1 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2, 3, 4, and 5 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

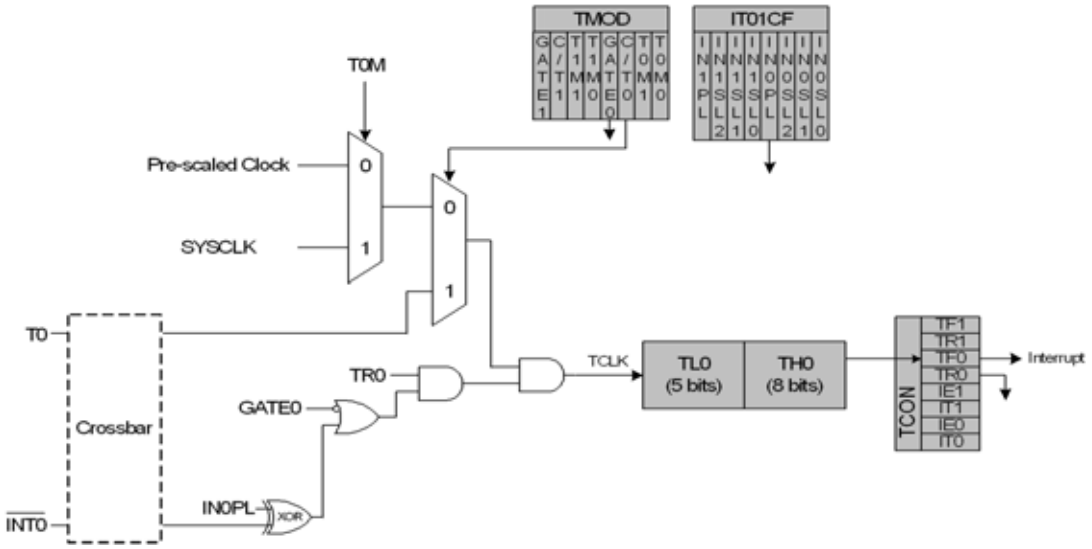

Figure 25.1. T0 Mode 0 Block Diagram

25.1.2. 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.

25.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 16.7 for details on the external input signals $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$).

SFR Definition 25.3. TCON: Timer Control

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
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 = 0x88; SFR Page = All Pages; Bit-Addressable

Bit	Name	Function
7	TF1	**Timer 1 Overflow Flag.** Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
6	TR1	**Timer 1 Run Control.** Timer 1 is enabled by setting this bit to 1.
5	TF0	**Timer 0 Overflow Flag.** Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
4	TR0	**Timer 0 Run Control.** Timer 0 is enabled by setting this bit to 1.
3	IE1	**External Interrupt 1.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine in edge-triggered mode.
2	IT1	**Interrupt 1 Type Select.** This bit selects whether the configured $\overline{\text{INT1}}$ interrupt will be edge or level sensitive. $\overline{\text{INT1}}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 16.7). 0: $\overline{\text{INT1}}$ is level triggered. 1: $\overline{\text{INT1}}$ is edge triggered.
1	IE0	**External Interrupt 0.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine in edge-triggered mode.
0	IT0	**Interrupt 0 Type Select.** This bit selects whether the configured $\overline{\text{INT0}}$ interrupt will be edge or level sensitive. $\overline{\text{INT0}}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 16.7). 0: $\overline{\text{INT0}}$ is level triggered. 1: $\overline{\text{INT0}}$ is edge triggered.

25.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode, (split) 8-bit auto-reload mode, or Low-Frequency Oscillator (LFO) Falling Edge capture mode. The Timer 2 operation mode is defined by the T2SPLIT (TMR2CN.3), T2CE (TMR2CN.4) bits, and T2CSS (TMR2CN.1) bits.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

25.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 25.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

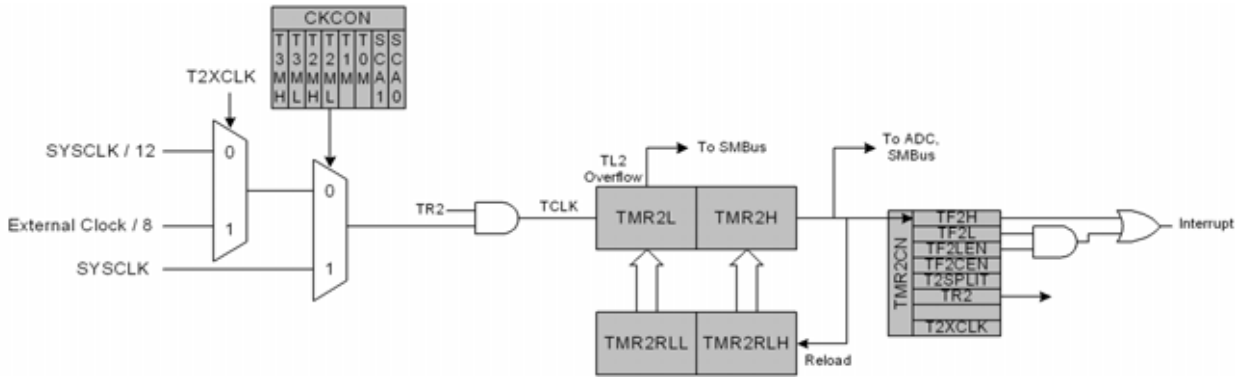

Figure 25.4. Timer 2 16-Bit Mode Block Diagram

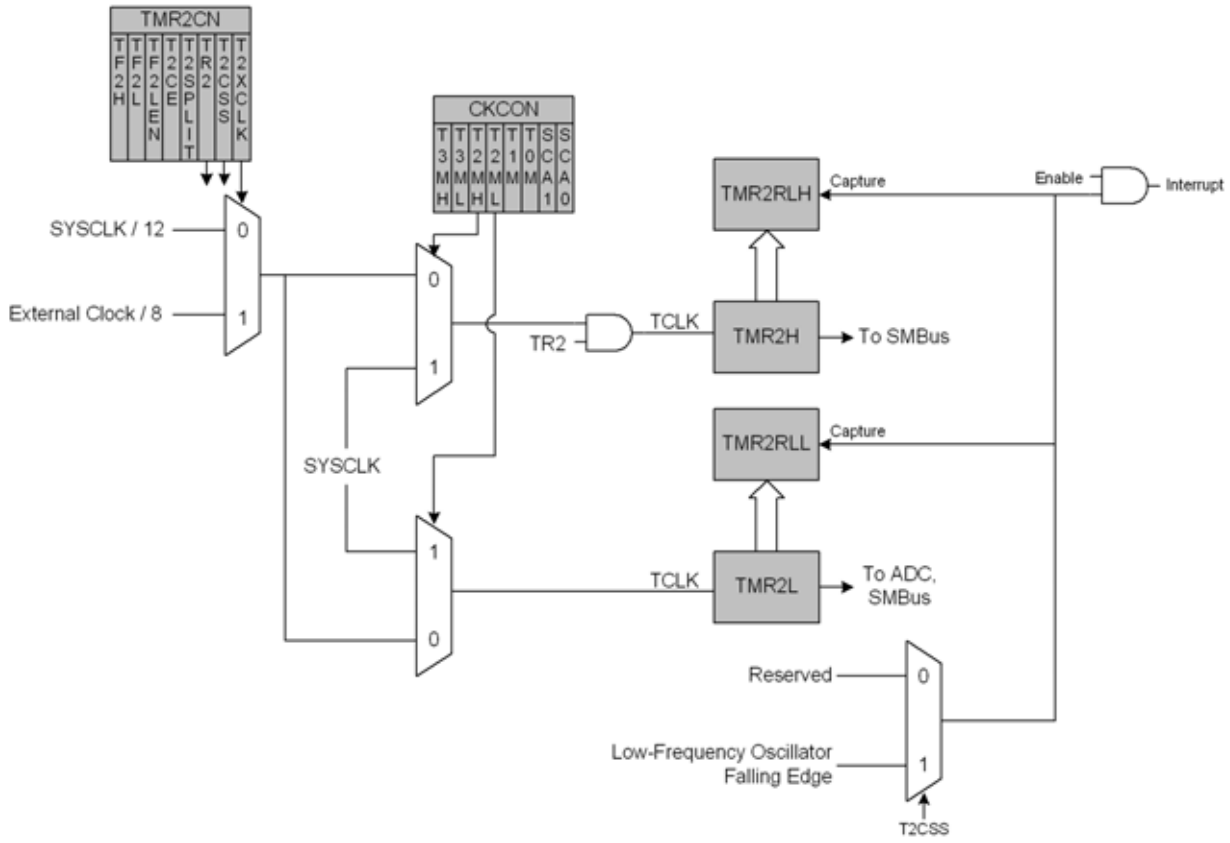

Figure 25.7. Timer 2 Capture Mode (T2SPLIT = 0)