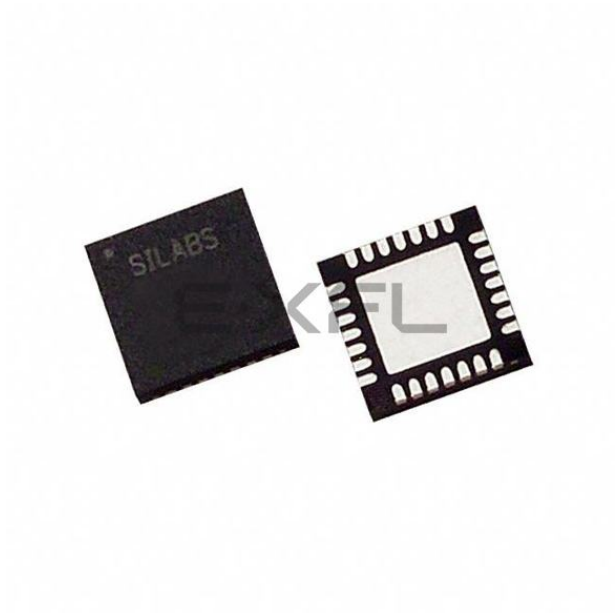

Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	48 MIPS
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	POR, PWM, WDT
Number of I/O	15
Program Memory Size	16KB (16K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t327-gmr

List of Tables

Table 2.1. Product Selection Guide 18
Table 3.1. Pin Definitions for the C8051T622/3 and C8051T326/7 19
Table 4.1. QFN-24 Package Dimensions 24
Table 4.2. QFN-24 PCB Land Pattern Dimesions 25
Table 5.1. QFN-28 Package Dimensions 26
Table 5.2. QFN-28 PCB Land Pattern Dimensions 27
Table 6.1. Absolute Maximum Ratings 28
Table 6.2. Global Electrical Characteristics 29
Table 6.3. Port I/O DC Electrical Characteristics 30
Table 6.4. Reset Electrical Characteristics 30
Table 6.5. Internal Voltage Regulator Electrical Characteristics 31
Table 6.6. EPROM Electrical Characteristics 31
Table 6.7. Internal High-Frequency Oscillator Electrical Characteristics 32
Table 6.8. Internal Low-Frequency Oscillator Electrical Characteristics 32
Table 6.9. External Oscillator Electrical Characteristics 32
Table 6.10. USB Transceiver Electrical Characteristic 33
Table 8.1. CIP-51 Instruction Set Summary 42
Table 11.1. Special Function Register (SFR) Memory Map 56
Table 11.2. Special Function Registers 57
Table 12.1. Interrupt Summary 62
Table 13.1. Security Byte Decoding 73
Table 17.1. Port I/O Assignment for Analog Functions 99
Table 17.2. Port I/O Assignment for Digital Functions 99
Table 17.3. Port I/O Assignment for External Digital Event Capture Functions 100
Table 18.1. Endpoint Addressing Scheme 117
Table 18.2. USB0 Controller Registers 122
Table 18.3. FIFO Configurations 126
Table 19.1. SMBus Clock Source Selection 153
Table 19.2. Minimum SDA Setup and Hold Times 154
Table 19.3. Sources for Hardware Changes to SMB0CN 158
Table 19.4. Hardware Address Recognition Examples (EHACK = 1) 159
Table 19.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0) 166
Table 19.6. SMBus Status Decoding With Hardware ACK Generation Enabled (EHACK = 1) 168
Table 20.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator 178
Table 20.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator 178
Table 21.1. Baud Rate Generator Settings for Standard Baud Rates 180
Table 22.1. SPI Slave Timing Parameters 201
Table 24.1. PCA Timebase Input Options 225

List of Registers

SFR Definition 7.1. REG01CN: Voltage Regulator Control 39
SFR Definition 8.1. DPL: Data Pointer Low Byte 46
SFR Definition 8.2. DPH: Data Pointer High Byte 46
SFR Definition 8.3. SP: Stack Pointer 47
SFR Definition 8.4. ACC: Accumulator 47
SFR Definition 8.5. B: B Register 47
SFR Definition 8.6. PSW: Program Status Word 48
SFR Definition 9.1. PFE0CN: Prefetch Engine Control 49
SFR Definition 10.1. EMI0CN: External Memory Interface Control 53
SFR Definition 10.2. EMI0CF: External Memory Configuration 55
SFR Definition 12.1. IE: Interrupt Enable 63
SFR Definition 12.2. IP: Interrupt Priority 64
SFR Definition 12.3. EIE1: Extended Interrupt Enable 1 65
SFR Definition 12.4. EIP1: Extended Interrupt Priority 1 66
SFR Definition 12.5. EIE2: Extended Interrupt Enable 2 67
SFR Definition 12.6. EIP2: Extended Interrupt Priority 2 68
SFR Definition 12.7. IT01CF: INT0/INT1 ConfigurationO 70
SFR Definition 13.1. PSCTL: Program Store R/W Control 75
SFR Definition 13.2. MEMKEY: EPROM Memory Lock and Key 75
SFR Definition 13.3. IAPCN: In-Application Programming Control 76
SFR Definition 14.1. PCON: Power Control 79
SFR Definition 15.1. VDM0CN: VDD Monitor Control 83
SFR Definition 15.2. RSTSRC: Reset Source 85
SFR Definition 16.1. CLKSEL: Clock Select 88
SFR Definition 16.2. OSCICL: Internal H-F Oscillator Calibration 89
SFR Definition 16.3. OSCICN: Internal H-F Oscillator Control 90
SFR Definition 16.4. CLKMUL: Clock Multiplier Control 91
SFR Definition 16.5. OSCLCN: Internal L-F Oscillator Control 92
SFR Definition 16.6. OSCXCN: External Oscillator Control 96
SFR Definition 17.1. XBR0: Port I/O Crossbar Register 0 105
SFR Definition 17.2. XBR1: Port I/O Crossbar Register 1 106
SFR Definition 17.3. XBR2: Port I/O Crossbar Register 2 107
SFR Definition 17.4. P0MASK: Port 0 Mask Register 108
SFR Definition 17.5. P0MAT: Port 0 Match Register 108
SFR Definition 17.6. P1MASK: Port 1 Mask Register 109
SFR Definition 17.7. P1MAT: Port 1 Match Register 109
SFR Definition 17.8. P0: Port 0 110
SFR Definition 17.9. P0MDIN: Port 0 Input Mode 111
SFR Definition 17.10. P0MDOUT: Port 0 Output Mode 111
SFR Definition 17.11. P0SKIP: Port 0 Skip 112
SFR Definition 17.12. P1: Port 1 112
SFR Definition 17.13. P1MDIN: Port 1 Input Mode 113
SFR Definition 17.14. P1MDOUT: Port 1 Output Mode 113

)

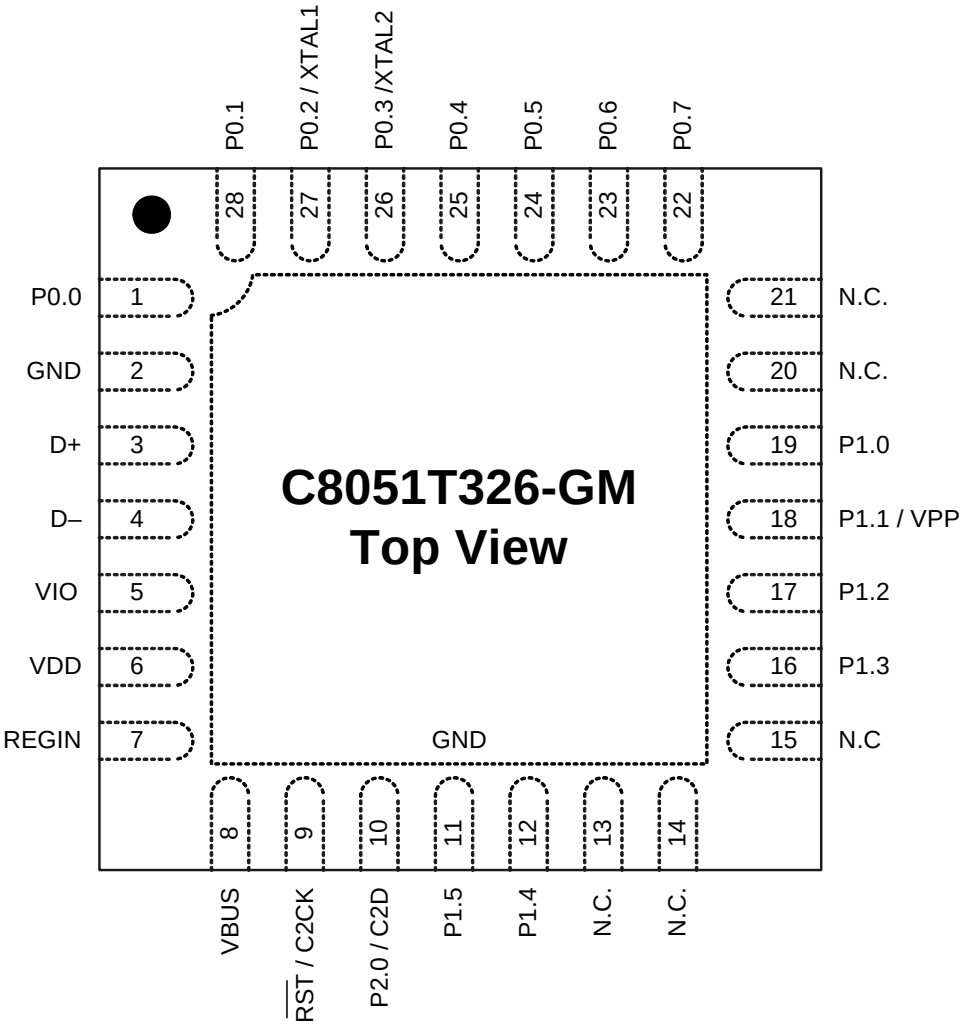

Figure 3.2. C8051T326 (QFN-28) Pinout Diagram (Top View)

Table 6.5. Internal Voltage Regulator Electrical Characteristics
–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Voltage Regulator (REG0)					
Input Voltage Range[1, 3]		2.7	—	5.25	V
Output Voltage (V_{DD})[2]	Output Current = 1 to 100 mA	3.3	3.45	3.6	V
Output Current[2]		—	—	100	mA
VBUS Detection Input Threshold		2.5	—	—	V
Bias Current	Normal Mode (REG0MD = 0) Low Power Mode (REG0MD = 1)	— —	83 43	99 55	µA
Dropout Voltage (V_{DO})[3]	I_{DD} = 1 mA I_{DD} = 100 mA	— —	1 100	— —	mV/mA
Voltage Regulator (REG1)					
Input Voltage Range		1.8	—	3.6	V
Bias Current	Normal Mode (REG1MD = 0) Low Power Mode (REG1MD = 1)	— —	340 —	425 185	µA

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 C8051T622/3 and C8051T326/7.
3. The minimum input voltage is 2.7 V or V_{DD} + V_{DO} (max load), whichever is greater.

Table 6.6. EPROM Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
EPROM Size	C8051T622/326/327 (Note 1) C8051T623	16384 8192	— —	— —	Bytes
Write Cycle Time (per Byte) (Note 2)		105	155	205	µs
In-Application Programming Write Cycle Time (per Byte) (Note 3)	Capacitor on V_{PP} = 4.7 µF and fully discharged Capacitor on V_{PP} = 4.7 µF and initially charged to 3.3 V	— —	37 26	— —	ms ms
Programming Voltage (V_{PP})		5.75	6.0	6.25	V
Capacitor on V_{PP} for In-application Programming		—	4.7	—	µF

Notes:
1. 512 bytes at location 0x3E00 to 0x3FFF are not available for program storage
2. For devices with a Date Code prior to 1111, the programming time over the C2 interface is twice as long.
3. Duration of write time is largely dependent on VIO voltage, supply voltage, and residual charge on the VPP capacitor. The majority of the write time consists of charging the voltage on VPP to 6.0 V. These measurements include the VPP ramp time and VDD = VIO = 3.3 V

7.2. Voltage Regulator (REG1)

Under default conditions, the internal REG1 regulator will remain on when the device enters STOP mode. This allows any enabled reset source to generate a reset for the device and bring the device out of STOP mode. For additional power savings, the STOPCF bit can be used to shut down the regulator and the internal power network of the device when the part enters STOP mode. When STOPCF is set to 1, the $\overline{\text{RST}}$ pin and a full power cycle of the device are the only methods of generating a reset.

REG1 offers an additional low power mode intended for use when the device is in suspend mode. This low power mode should not be used during normal operation or if the REG0 Voltage Regulator is disabled. See Table 6.5 for normal and low power mode supply current specifications. The REG1 mode selection is controlled via the REG1MD bit in register REG01CN.

Important Note: At least 12 clock instructions must occur after placing REG1 in low power mode before the Internal High Frequency Oscillator is Suspended (OSCICN.5 = 1b).

Table 8.1. CIP-51 Instruction Set Summary(Continued)

Mnemonic	Description	Bytes	Clock Cycles
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/4
JNC rel	Jump if Carry is not set	2	2/4
JB bit, rel	Jump if direct bit is set	3	3/5
JNB bit, rel	Jump if direct bit is not set	3	3/5
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/5
Program Branching			
ACALL addr11	Absolute subroutine call	2	4
LCALL addr16	Long subroutine call	3	5
RET	Return from subroutine	1	6
RETI	Return from interrupt	1	6
AJMP addr11	Absolute jump	2	4
LJMP addr16	Long jump	3	5
SJMP rel	Short jump (relative address)	2	4
JMP @A+DPTR	Jump indirect relative to DPTR	1	4
JZ rel	Jump if A equals zero	2	2/4
JNZ rel	Jump if A does not equal zero	2	2/4
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/5
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/5
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/5
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/6
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/4
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/5
NOP	No operation	1	1

SFR Definition 16.3. OSCICN: Internal H-F Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY	SUSPEND				IFCN[1:0]	
Type	R/W	R	R/W	R	R	R	R/W	
Reset	1	1	0	0	0	0	0	0

SFR Address = 0xB2

Bit	Name	Function
7	IOSCEN	**Internal H-F Oscillator Enable Bit.** 0: Internal H-F Oscillator Disabled. 1: Internal H-F Oscillator Enabled.
6	IFRDY	**Internal H-F Oscillator Frequency Ready Flag.** 0: Internal H-F Oscillator is not running at programmed frequency. 1: Internal H-F Oscillator is running at programmed frequency.
5	SUSPEND	**Internal Oscillator Suspend Enable Bit.** Setting this bit to logic 1 places the internal oscillator in SUSPEND mode. The internal oscillator resumes operation when one of the SUSPEND mode awakening events occurs.
4:2	Unused	Unused. Read = 000b; Write = Don't Care
1:0	IFCN[1:0]	**Internal H-F Oscillator Frequency Divider Control Bits.** The Internal H-F Oscillator is divided by the IFCN bit setting after a divide-by-4 stage. 00: SYSCLK can be derived from Internal H-F Oscillator divided by 8 (1.5 MHz). 01: SYSCLK can be derived from Internal H-F Oscillator divided by 4 (3 MHz). 10: SYSCLK can be derived from Internal H-F Oscillator divided by 2 (6 MHz). 11: SYSCLK can be derived from Internal H-F Oscillator divided by 1 (12 MHz).

SFR Definition 17.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME[2:0]		
Type	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
4	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
3	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
2:0	PCA0ME[2:0]	**PCA Module I/O Enable Bits.** 000: All PCA I/O unavailable at Port pins. 001: CEX0 routed to Port pin. 010: CEX0, CEX1 routed to Port pins. 011: CEX0, CEX1, CEX2 routed to Port pins. 100-111: Reserved.

USB Register Definition 18.11. IN1INT: USB0 IN Endpoint Interrupt

Bit	7	6	5	4	3	2	1	0
Name						IN2	IN1	EP0
Type	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

USB Register Address = 0x02

Bit	Name	Function
7:3	Unused	Unused. Read = 00000b. Write = don't care.
2	IN2	**IN Endpoint 2 Interrupt-Pending Flag.** This bit is cleared when software reads the IN1INT register. 0: IN Endpoint 2 interrupt inactive. 1: IN Endpoint 2 interrupt active.
1	IN1	**IN Endpoint 1 Interrupt-Pending Flag.** This bit is cleared when software reads the IN1INT register. 0: IN Endpoint 1 interrupt inactive. 1: IN Endpoint 1 interrupt active.
0	EP0	**Endpoint 0 Interrupt-Pending Flag.** This bit is cleared when software reads the IN1INT register. 0: Endpoint 0 interrupt inactive. 1: Endpoint 0 interrupt active.

5. Hardware sets the SUEND bit (E0CSR.4) because a control transfer ended before firmware sets the DATAEND bit (E0CSR.3).

The E0CNT register (USB Register Definition 18.11) holds the number of received data bytes in the Endpoint0 FIFO.

Hardware will automatically detect protocol errors and send a STALL condition in response. Firmware may force a STALL condition to abort the current transfer. When a STALL condition is generated, the STSTL bit will be set to 1 and an interrupt generated. The following conditions will cause hardware to generate a STALL condition:

1. The host sends an OUT token during a OUT data phase after the DATAEND bit has been set to 1.
2. The host sends an IN token during an IN data phase after the DATAEND bit has been set to 1.
3. The host sends a packet that exceeds the maximum packet size for Endpoint0.
4. The host sends a non-zero length DATA1 packet during the status phase of an IN transaction.

Firmware sets the SDSTL bit (E0CSR.5) to 1.

18.10.1. Endpoint0 SETUP Transactions

All control transfers must begin with a SETUP packet. SETUP packets are similar to OUT packets, containing an 8-byte data field sent by the host. Any SETUP packet containing a command field of anything other than 8 bytes will be automatically rejected by USB0. An Endpoint0 interrupt is generated when the data from a SETUP packet is loaded into the Endpoint0 FIFO. Software should unload the command from the Endpoint0 FIFO, decode the command, perform any necessary tasks, and set the SOPRDY bit to indicate that it has serviced the OUT packet.

18.10.2. Endpoint0 IN Transactions

When a SETUP request is received that requires USB0 to transmit data to the host, one or more IN requests will be sent by the host. For the first IN transaction, firmware should load an IN packet into the Endpoint0 FIFO, and set the INPRDY bit (E0CSR.1). An interrupt will be generated when an IN packet is transmitted successfully. Note that no interrupt will be generated if an IN request is received before firmware has loaded a packet into the Endpoint0 FIFO. If the requested data exceeds the maximum packet size for Endpoint0 (as reported to the host), the data should be split into multiple packets; each packet should be of the maximum packet size excluding the last (residual) packet. If the requested data is an integer multiple of the maximum packet size for Endpoint0, the last data packet should be a zero-length packet signaling the end of the transfer. Firmware should set the DATAEND bit to 1 after loading into the Endpoint0 FIFO the last data packet for a transfer.

Upon reception of the first IN token for a particular control transfer, Endpoint0 is said to be in Transmit Mode. In this mode, only IN tokens should be sent by the host to Endpoint0. The SUEND bit (E0CSR.4) is set to 1 if a SETUP or OUT token is received while Endpoint0 is in Transmit Mode.

Endpoint0 will remain in Transmit Mode until any of the following occur:

1. USB0 receives an Endpoint0 SETUP or OUT token.
2. Firmware sends a packet less than the maximum Endpoint0 packet size.
3. Firmware sends a zero-length packet.

Firmware should set the DATAEND bit (E0CSR.3) to 1 when performing (2) and (3) above.

The SIE will transmit a NAK in response to an IN token if there is no packet ready in the IN FIFO (INPRDY = 0).

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 19.3 illustrates a typical SMBus transaction.

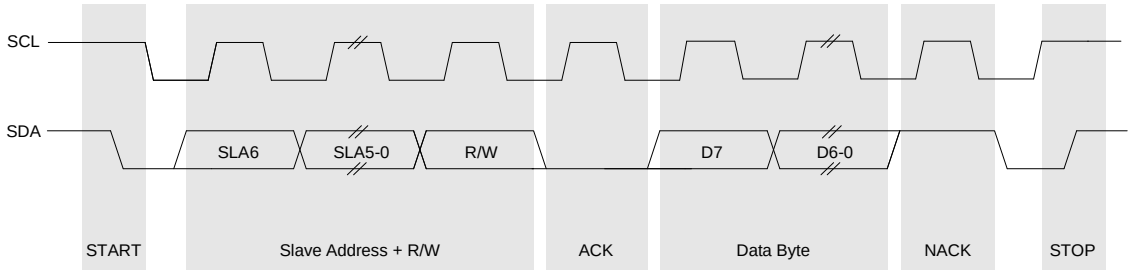

Figure 19.3. SMBus Transaction

19.3.1. Transmitter Vs. Receiver

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

19.3.2. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “19.3.5. SCL High (SMBus Free) Timeout” on page 152). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

19.3.3. Clock Low Extension

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

19.3.4. SCL Low Timeout

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

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to

SFR Definition 19.4. SMB0ADM: SMBus Slave Address Mask

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

SFR Address = 0xCF

Bit	Name	Function
7:1	SLVM[6:0]	**SMBus Slave Address Mask.** Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM[6:0] enables comparisons with the corresponding bit in SLV[6:0]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK	**Hardware Acknowledge Enable.** Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic Slave Address Recognition and Hardware Acknowledge is Enabled.

19.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames. Note that the position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur **after** the ACK, regardless of whether hardware ACK generation is enabled or not.

19.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 0 (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 19.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

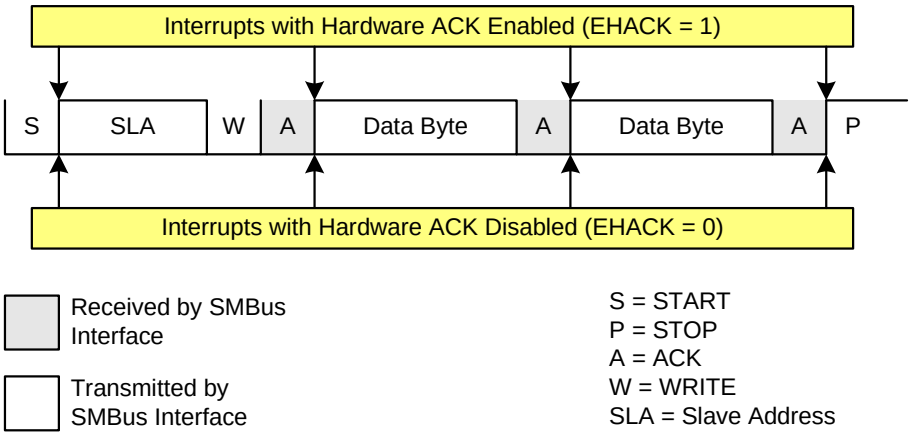

Figure 19.5. Typical Master Write Sequence

20.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 20.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

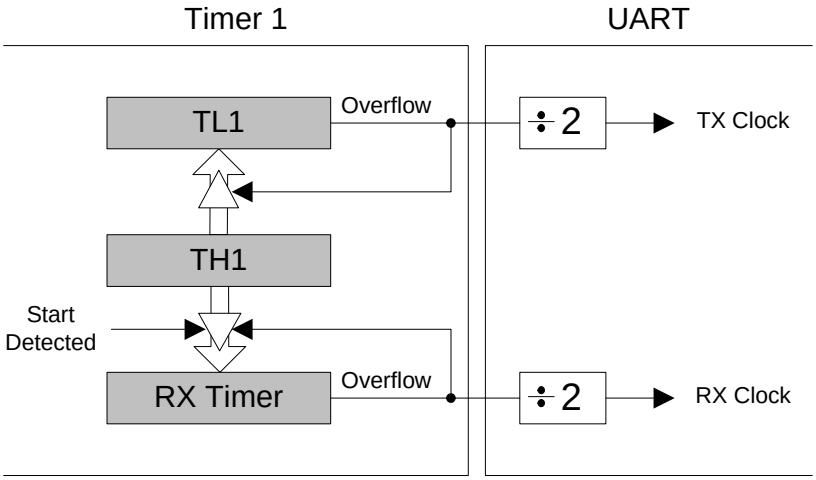

Figure 20.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “23.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 205). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK/4, SYSCLK/12, SYSCLK/48, the external oscillator clock/8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 20.1-A and Equation 20.1-B.

A) $$\text{UARTBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

B) $$\text{T1\_Overflow\_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$

Equation 20.1. UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section “23. Timers” on page 202. A quick reference for typical baud rates and system clock frequencies is given in Table 20.1 through Table 20.2. The internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

SFR Definition 22.3. SPI0CKR: SPI0 Clock Rate

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

SFR Address = 0xA2

Bit	Name	Function
7:0	SCR[7:0]	**SPI0 Clock Rate.** These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register. $f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR[7:0] + 1)}$ for 0 <= SPI0CKR <= 255 Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04, $f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$ $f_{SCK} = 200kHz$

SFR Definition 22.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA3

Bit	Name	Function
7:0	SPI0DAT[7:0]	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.

Table 22.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing (See Figure 22.8 and Figure 22.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing (See Figure 22.10 and Figure 22.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns
Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

SFR Definition 23.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8A

Bit	Name	Function
7:0	TL0[7:0]	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 23.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8B

Bit	Name	Function
7:0	TL1[7:0]	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

23.2. 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 or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode.

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.

23.2.1. 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 23.4, 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.

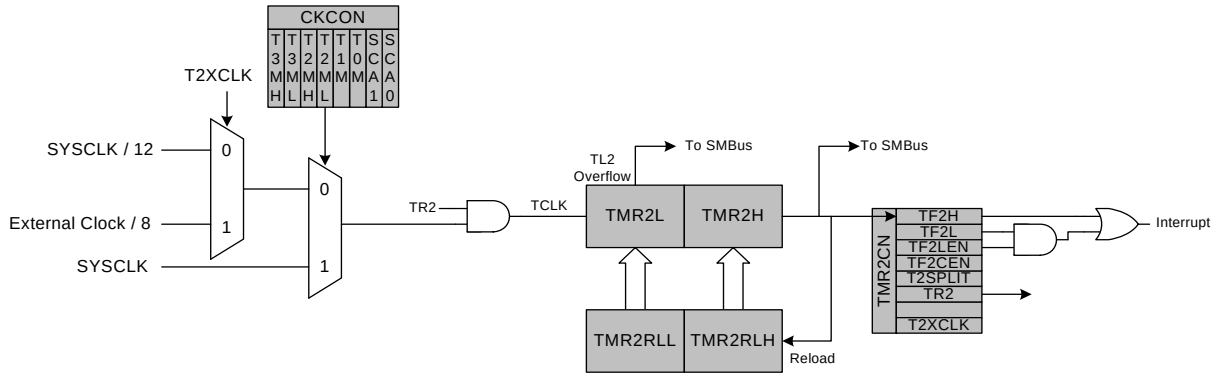

Figure 23.4. Timer 2 16-Bit Mode Block Diagram

SFR Definition 23.12. TMR2H Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xCD

Bit	Name	Function
7:0	TMR2H[7:0]	**Timer 2 Low Byte.** In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

SFR Definition 24.4. PCA0CPMn: PCA Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
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 Addresses: 0xDA (n = 0), 0xDB (n = 1), 0xDC (n = 2),

Bit	Name	Function
7	PWM16n	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOMn	**Comparator Function Enable.** This bit enables the comparator function for PCA module n when set to 1.
5	CAPPn	**Capture Positive Function Enable.** This bit enables the positive edge capture for PCA module n when set to 1.
4	CAPNn	**Capture Negative Function Enable.** This bit enables the negative edge capture for PCA module n when set to 1.
3	MATn	**Match Function Enable.** This bit enables the match function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	**Toggle Function Enable.** This bit enables the toggle function for PCA module n when set to 1. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function for PCA module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	ECCFn	**Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.

Note: When the WDTE bit is set to 1, the PCA0CPM2 register cannot be modified, and module 2 acts as the watchdog timer. To change the contents of the PCA0CPM2 register or the function of module 2, the Watchdog Timer must be disabled.