

Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	57
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.8V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	76-VFQFN Dual Rows, Exposed Pad
Supplier Device Package	76-DQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f968-a-gm

C8051F96x

SFR Definition 25.14. PC0CMP0M: PC0 Comparator 0 Middle	334
SFR Definition 25.15. PC0CMP0L: PC0 Comparator 0 Low (LSB)	334
SFR Definition 25.16. PC0CMP1H: PC0 Comparator 1 High (MSB)	335
SFR Definition 25.17. PC0CMP1M: PC0 Comparator 1 Middle	335
SFR Definition 25.18. PC0CMP1L: PC0 Comparator 1 Low (LSB)	335
SFR Definition 25.19. PC0HIST: PC0 History	336
SFR Definition 25.20. PC0INT0: PC0 Interrupt 0	337
SFR Definition 25.21. PC0INT1: PC0 Interrupt 1	338
SFR Definition 26.1. LCD0Dn: LCD0 Data	340
SFR Definition 26.2. LCD0CN: LCD0 Control Register	342
SFR Definition 26.3. LCD0CNTRST: LCD0 Contrast Adjustment	346
SFR Definition 26.4. LCD0MSCN: LCD0 Master Control	347
SFR Definition 26.5. LCD0MSCF: LCD0 Master Configuration	348
SFR Definition 26.6. LCD0PWR: LCD0 Power	348
SFR Definition 26.7. LCD0VBMCN: LCD0 VBAT Monitor Control	349
SFR Definition 26.8. LCD0CLKDIVH: LCD0 Refresh Rate Prescaler High Byte	350
SFR Definition 26.9. LCD0CLKDIVL: LCD Refresh Rate Prescaler Low Byte	350
SFR Definition 26.10. LCD0BLINK: LCD0 Blink Mask	351
SFR Definition 26.11. LCD0TOGR: LCD0 Toggle Rate	352
SFR Definition 26.12. LCD0CF: LCD0 Configuration	353
SFR Definition 26.13. LCD0CHPCN: LCD0 Charge Pump Control	353
SFR Definition 26.14. LCD0CHPCF: LCD0 Charge Pump Configuration	354
SFR Definition 26.15. LCD0CHPMD: LCD0 Charge Pump Mode	354
SFR Definition 26.16. LCD0BUFCN: LCD0 Buffer Control	354
SFR Definition 26.17. LCD0BUFCF: LCD0 Buffer Configuration	355
SFR Definition 26.18. LCD0BUFMD: LCD0 Buffer Mode	355
SFR Definition 26.19. LCD0VBMCF: LCD0 VBAT Monitor Configuration	355
SFR Definition 27.1. XBR0: Port I/O Crossbar Register 0	363
SFR Definition 27.2. XBR1: Port I/O Crossbar Register 1	364
SFR Definition 27.3. XBR2: Port I/O Crossbar Register 2	365
SFR Definition 27.4. P0MASK: Port0 Mask Register	366
SFR Definition 27.5. P0MAT: Port0 Match Register	366
SFR Definition 27.6. P1MASK: Port1 Mask Register	367
SFR Definition 27.7. P1MAT: Port1 Match Register	367
SFR Definition 27.8. P0: Port0	369
SFR Definition 27.9. P0SKIP: Port0 Skip	369
SFR Definition 27.10. P0MDIN: Port0 Input Mode	370
SFR Definition 27.11. P0MDOUT: Port0 Output Mode	370
SFR Definition 27.12. P0DRV: Port0 Drive Strength	371
SFR Definition 27.13. P1: Port1	371
SFR Definition 27.14. P1SKIP: Port1 Skip	372
SFR Definition 27.15. P1MDIN: Port1 Input Mode	372
SFR Definition 27.16. P1MDOUT: Port1 Output Mode	373
SFR Definition 27.17. P1DRV: Port1 Drive Strength	373
SFR Definition 27.18. P2: Port2	374

5. SAR ADC with 16-bit Auto-Averaging Accumulator and Autonomous Low Power Burst Mode

The ADC0 on C8051F96x devices is a 300 ksp/s, 10-bit or 75 ksp/s, 12-bit successive-approximation-register (SAR) ADC with integrated track-and-hold and programmable window detector. ADC0 also has an autonomous low power Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. It also has a 16-bit accumulator that can automatically oversample and average the ADC results. See Section 5.4 for more details on using the ADC in 12-bit mode.

The ADC is fully configurable under software control via Special Function Registers. The ADC0 operates in Single-ended mode and may be configured to measure various different signals using the analog multiplexer described in “5.7. ADC0 Analog Multiplexer” on page 95. The voltage reference for the ADC is selected as described in “5.9. Voltage and Ground Reference Options” on page 100.

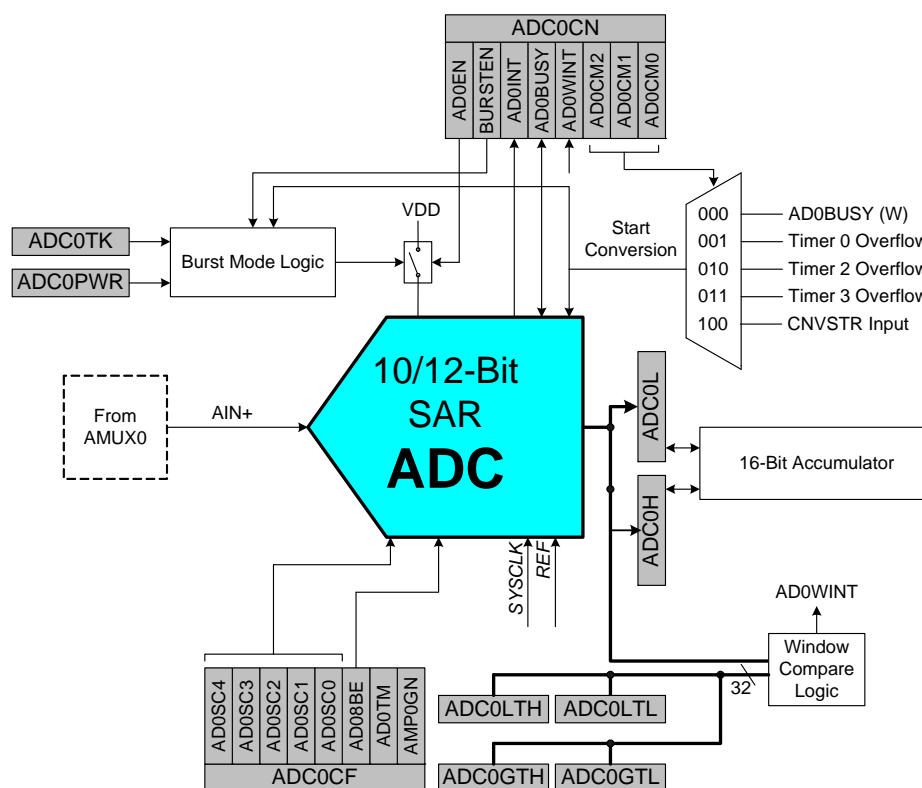


Figure 5.1. ADC0 Functional Block Diagram

5.1. Output Code Formatting

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0SJUST[2:0]. When the repeat count is set to 1, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from 0 to $V_{REF} \times 1023/1024$. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to 0.

Table 8.1. 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

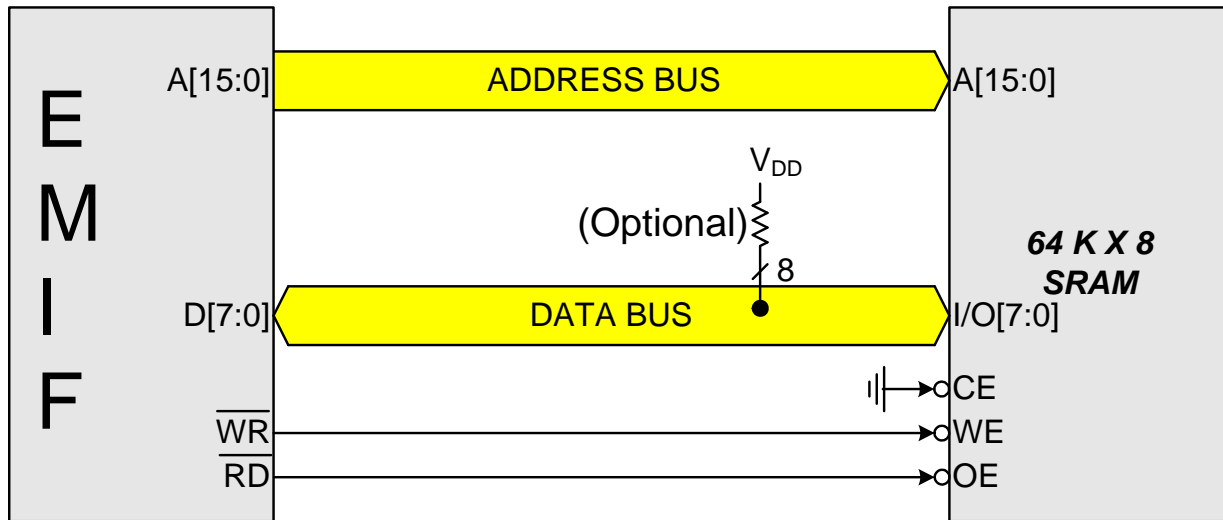


Figure 10.2. Non-multiplexed Configuration Example

10.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 10.3, based on the EMIOCF Mode bits in the EMI0CF register (SFR Definition 10.2). These modes are summarized below. More information about the different modes can be found in Section “10.6. Timing” on page 137.

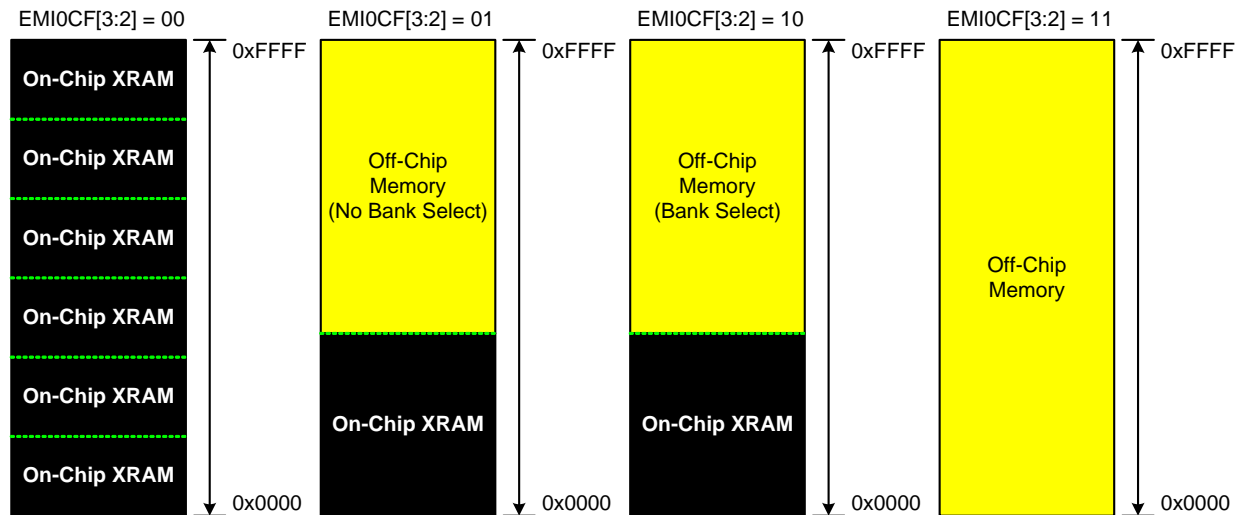


Figure 10.3. EMIF Operating Modes

10.6.1.2. 8-bit MOVX without Bank Select: EMI0CF[4:2] = 101 or 111

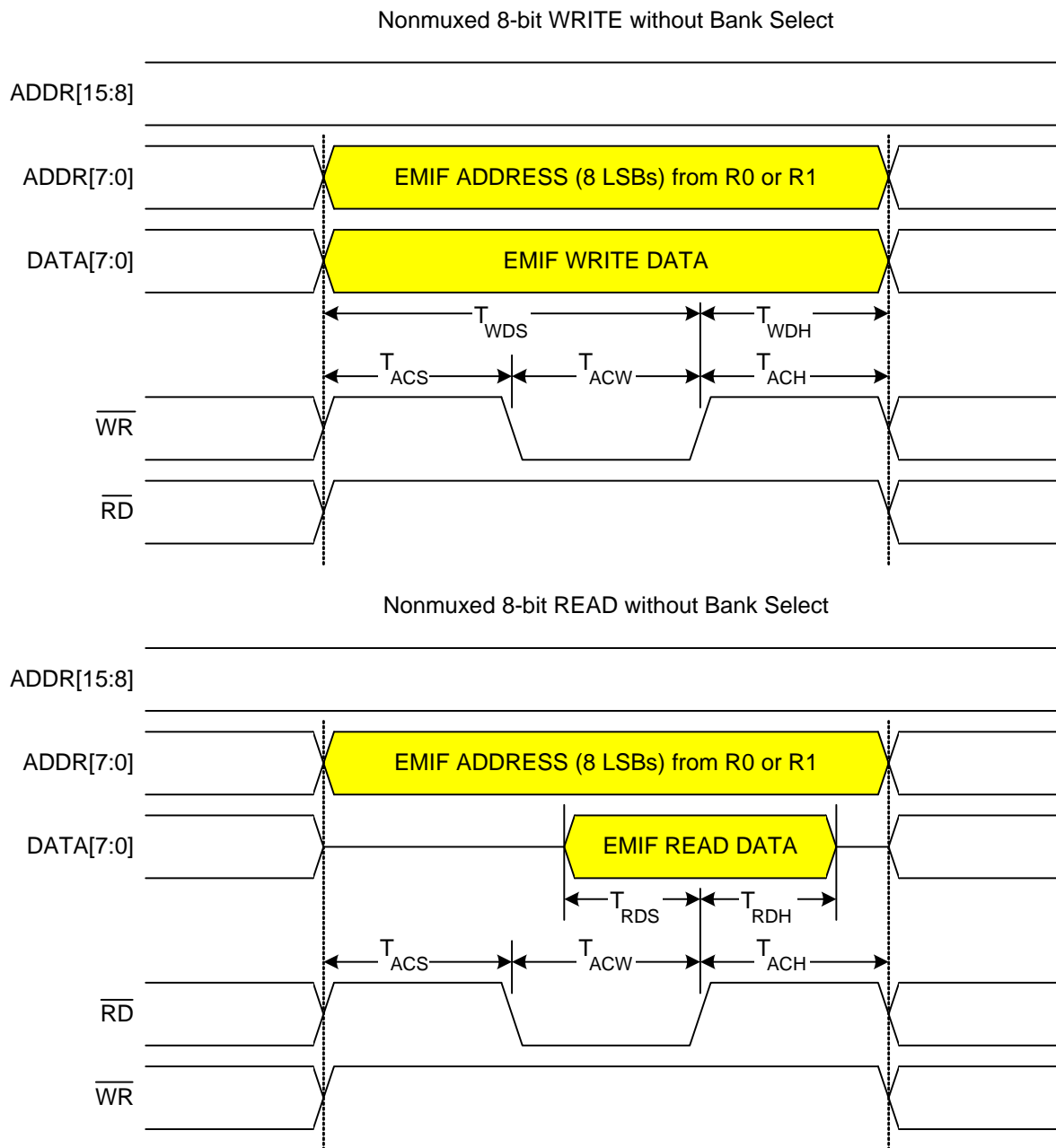


Figure 10.5. Non-multiplexed 8-bit MOVX without Bank Select Timing

SFR Definition 11.1. DMA0EN: DMA0 Channel Enable

Bit	7	6	5	4	3	2	1	0
Name		CH6_EN	CH5_EN	CH4_EN	CH3_EN	CH2_EN	CH1_EN	CH0_EN
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD2

Bit	Name	Function
7	Unused	Read = 0b, Write = Don't Care
6	CH6_EN	Channel 6 Enable. 0: Disable DMA0 channel 6. 1: Enable DMA0 channel 6.
5	CH5_EN	Channel 5 Enable. 0: Disable DMA0 channel 5. 1: Enable DMA0 channel 5.
4	CH4_EN	Channel 4 Enable. 0: Disable DMA0 channel 4. 1: Enable DMA0 channel 4.
3	CH3_EN	Channel 3 Enable. 0: Disable DMA0 channel 3. 1: Enable DMA0 channel 3.
2	CH2_EN	Channel 2 Enable. 0: Disable DMA0 channel 2. 1: Enable DMA0 channel 2.
1	CH1_EN	Channel 1 Enable. 0: Disable DMA0 channel 1. 1: Enable DMA0 channel 1.
0	CH0_EN	Channel 0 Enable. 0: Disable DMA0 channel 0. 1: Enable DMA0 channel 0.

SFR Definition 11.2. DMA0INT: DMA0 Full-Length Interrupt

Bit	7	6	5	4	3	2	1	0
Name		CH6_INT	CH5_INT	CH4_INT	CH3_INT	CH2_INT	CH1_INT	CH0_INT
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xD3

Bit	Name	Function
7	Unused	Read = 0b, Write = Don't Care
6	CH6_INT	Channel 6 Full-Length Interrupt Flag.¹ 0: Full-length interrupt has not occurred on channel 6. 1: Full-length interrupt has not occurred on channel 6.
5	CH5_INT	0: Full-length interrupt has not occurred on channel 5. 1: Full-length interrupt has not occurred on channel 5.
4	CH4_INT	Channel 4 Full-Length Interrupt Flag.¹ 0: Full-length interrupt has not occurred on channel 4. 1: Full-length interrupt has not occurred on channel 4.
3	CH3_INT	Channel 3 Full-Length Interrupt Flag.¹ 0: Full-length interrupt has not occurred on channel 3. 1: Full-length interrupt has not occurred on channel 3.
2	CH2_INT	Channel 2 Full-Length Interrupt Flag.¹ 0: Full-length interrupt has not occurred on channel 2. 1: Full-length interrupt has not occurred on channel 2.
1	CH1_INT	Channel 1 Full-Length Interrupt Flag.¹ 0: Full-length interrupt has not occurred on channel 1. 1: Full-length interrupt has not occurred on channel 1.
0	CH0_INT	Channel 0 Full-Length Interrupt Flag.¹ 0: Full-length interrupt has not occurred on channel 0. 1: Full-length interrupt has not occurred on channel 0.

Note: 1.Full-length interrupt flag is set when the offset address DMA0NAOH/L is equals to data transfer size DMA0NSZH/L minus 1. This flag must be cleared by software or system reset. The full-length interrupt is enabled by setting bit 7 of DMA0NCF with DMA0SEL configured for the corresponding channel.

C8051F96x

12.2. 32-bit CRC Algorithm

The C8051F41x CRC unit calculates the 32-bit CRC using a poly of 0x04C11DB7. The CRC-32 algorithm is "reflected", meaning that all of the input bytes and the final 32-bit output are bit-reversed in the processing engine. The following is a description of a simplified CRC algorithm that produces results identical to the hardware:

- Step 1. XOR the least-significant byte of the current CRC result with the input byte. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x00000000 or 0xFFFFFFFF).
- Step 2. Right-shift the CRC result.
- Step 3. If the LSB of the CRC result is set, XOR the CRC result with the reflected polynomial (0xEDB88320).
- Step 4. Repeat at Step 2 for the number of input bits (8).

For example, the 32-bit 'F41x CRC algorithm can be described by the following code:

```
unsigned long UpdateCRC (unsigned long CRC_acc, unsigned char CRC_input)
{
    unsigned char i; // loop counter
    #define POLY 0xEDB88320 // bit-reversed version of the poly 0x04C11DB7
    // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
    // with no carries)

    CRC_acc = CRC_acc ^ CRC_input;

    // "Divide" the poly into the dividend using CRC XOR subtraction
    // CRC_acc holds the "remainder" of each divide
    //
    // Only complete this division for 8 bits since input is 1 byte
    for (i = 0; i < 8; i++)
    {
        // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
        // into the "dividend")
        if ((CRC_acc & 0x00000001) == 0x00000001)
        {
            // if so, shift the CRC value, and XOR "subtract" the poly
            CRC_acc = CRC_acc >> 1;
            CRC_acc ^= POLY;
        }
        else
        {
            // if not, just shift the CRC value
            CRC_acc = CRC_acc >> 1;
        }
    }
    // Return the final remainder (CRC value)
    return CRC_acc;
}
```

The following table lists several input values and the associated outputs using the 32-bit 'F41x CRC algorithm (an initial value of 0xFFFFFFFF is used):

SFR Definition 15.2. ENC0L: ENC0 Data Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ENC0L[7:0]							
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 Page = 0x2; SFR Address = 0xC2; Bit-Addressable

Bit	Name	Function
7:0	ENC0L[7:0]	ENC0 Data Low Byte.

SFR Definition 15.3. ENC0M: ENC0 Data Middle Byte

Bit	7	6	5	4	3	2	1	0
Name	ENC0M[7:0]							
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 Page = 0x2; SFR Address = 0xC3; Bit-Addressable

Bit	Name	Function
7:0	ENC0M[7:0]	ENC0 Data Middle Byte.

SFR Definition 15.4. ENC0H: ENC0 Data High Byte

Bit	7	6	5	4	3	2	1	0
Name	ENC0H[7:0]							
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 Page = 0x2; SFR Address = 0xC4; Bit-Addressable

Bit	Name	Function
7:0	ENC0H[7:0]	ENC0 Data High Byte.

17.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. If a high priority interrupt preempts a low priority interrupt, the low priority interrupt will finish execution after the high priority interrupt completes. Each interrupt has an associated interrupt priority bit in the Interrupt Priority and Extended Interrupt Priority registers used to configure its priority level. Low priority is the default.

If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate. See Table 17.1 on page 239 to determine the fixed priority order used to arbitrate between simultaneously recognized interrupts.

17.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 7 system clock cycles: 1 clock cycle to detect the interrupt, 1 clock cycle to execute a single instruction, and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 19 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

Table 17.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
SmaRTClock Alarm	0x0043	8	ALRM (RTC0CN.2)*	N	N	EARTC0 (EIE1.1)	PARTC0 (EIP1.1)
ADC0 Window Compara- tor	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 End of Conversion	0x0053	10	AD0INT (ADC0STA.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
Comparator1	0x006B	13	CP1FIF (CPT1CN.4) CP1RIF (CPT1CN.5)	N	N	ECP1 (EIE1.6)	PCP1 (EIP1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
VDD/VBAT Supply Monitor Early Warning	0x007B	15	VDDOK (VDM0CN.5) ¹ VBOK (VDM0CN.2) ¹			EWARN (EIE2.0)	PWARN (EIP2.0)
Port Match	0x0083	16	None			EMAT (EIE2.1)	PMAT (EIP2.1)

SFR Definition 19.6. PMU0MD: Power Management Unit Mode

Bit	7	6	5	4	3	2	1	0
Name	RTCOE	WAKEOE	MONDIS					
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 Page = 0x0; SFR Address = 0xB6

Bit	Name	Function
7	RTCOE	Buffered SmaRTClock Output Enable. Enables the buffered SmaRTClock oscillator output on P0.2. 0: Buffered SmaRTClock output not enabled. 1: Buffered SmaRTClock output not enabled.
6	WAKEOE	Wakeup Request Output Enable. Enables the Sleep Mode wake-up request signal on P0.3. 0: Wake-up request signal is not enabled. 1: Wake-up request signal is enabled.
5	MONDIS	POR Supply Monitor Disable. Writing a 1 to this bit disables the POR supply monitor.
4:0	Unused	Read = 00000b. Write = Don't Care.

23. Clocking Sources

C8051F96x devices include a programmable precision internal oscillator, an external oscillator drive circuit, a low power internal oscillator, and a SmartClock real time clock oscillator. The precision internal oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 23.1. The external oscillator can be configured using the OSCXCN register. The low power internal oscillator is automatically enabled and disabled when selected and deselected as a clock source. SmartClock operation is described in the SmartClock oscillator chapter.

The system clock (SYSCLK) can be derived from the precision internal oscillator, external oscillator, low power internal oscillator, low power internal oscillator divided by 8, or SmartClock oscillator. The global clock divider can generate a system clock that is 1, 2, 4, 8, 16, 32, 64, or 128 times slower than the selected input clock source. Oscillator electrical specifications can be found in the Electrical Specifications Chapter.

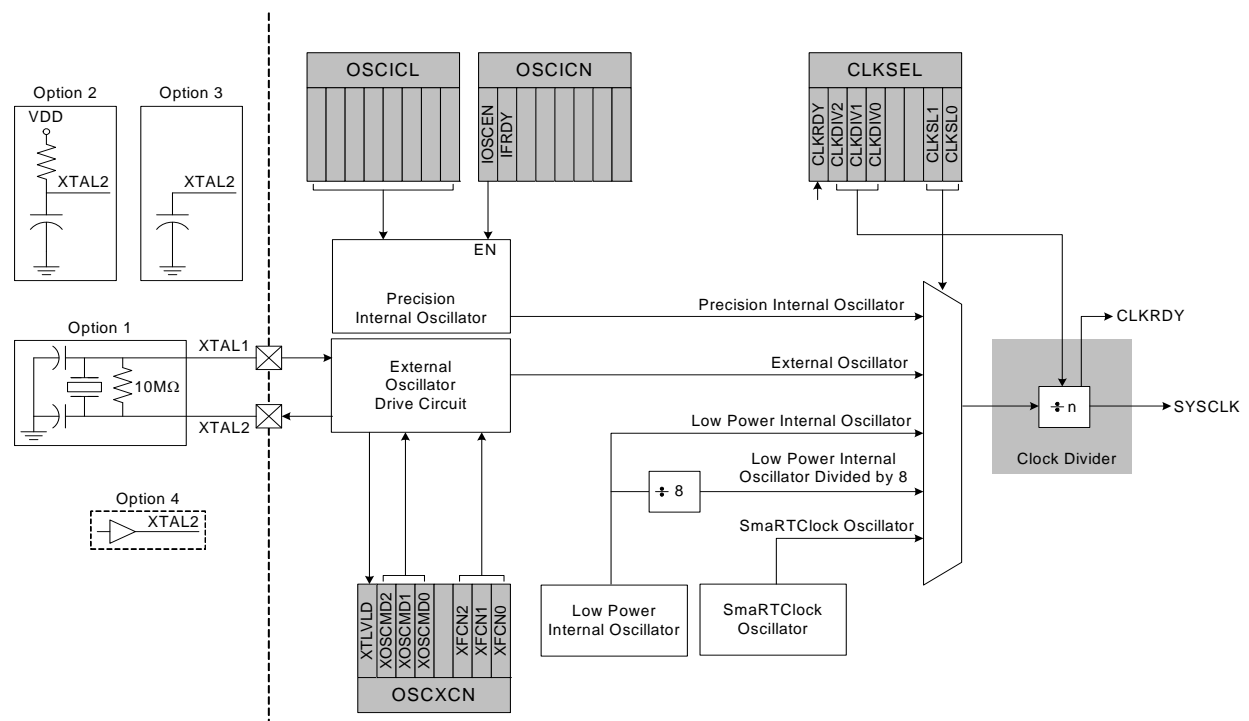


Figure 23.1. Clocking Sources Block Diagram

The proper way of changing the system clock when both the clock source and the clock divide value are being changed is as follows:

If switching from a fast “undivided” clock to a slower “undivided” clock:

1. Change the clock divide value.
2. Poll for CLKRDY > 1.
3. Change the clock source.

If switching from a slow “undivided” clock to a faster “undivided” clock:

1. Change the clock source.
2. Change the clock divide value.
3. Poll for CLKRDY > 1.

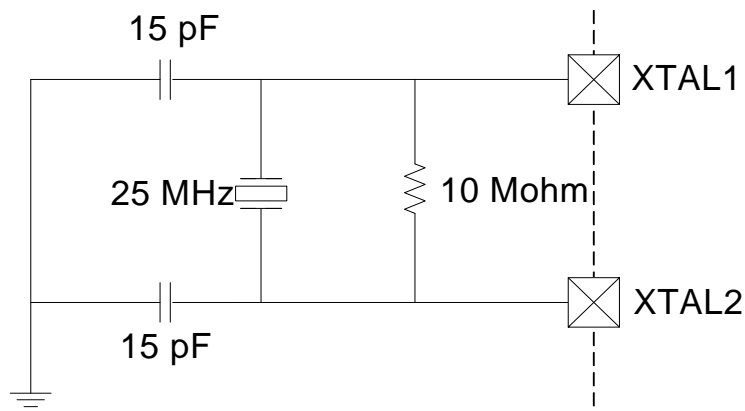


Figure 23.2. 25 MHz External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency. The selection should be based on Table 23.1. For example, a 25 MHz crystal requires an XFCN setting of 111b.

Table 23.1. Recommended XFCN Settings for Crystal Mode

XFCN	Crystal Frequency	Bias Current	Typical Supply Current (VDD = 2.4 V)
000	$f \leq 20 \text{ kHz}$	0.5 μA	3.0 μA , $f = 32.768 \text{ kHz}$
001	$20 \text{ kHz} < f \leq 58 \text{ kHz}$	1.5 μA	4.8 μA , $f = 32.768 \text{ kHz}$
010	$58 \text{ kHz} < f \leq 155 \text{ kHz}$	4.8 μA	9.6 μA , $f = 32.768 \text{ kHz}$
011	$155 \text{ kHz} < f \leq 415 \text{ kHz}$	14 μA	28 μA , $f = 400 \text{ kHz}$
100	$415 \text{ kHz} < f \leq 1.1 \text{ MHz}$	40 μA	71 μA , $f = 400 \text{ kHz}$
101	$1.1 \text{ MHz} < f \leq 3.1 \text{ MHz}$	120 μA	193 μA , $f = 400 \text{ kHz}$
110	$3.1 \text{ MHz} < f \leq 8.2 \text{ MHz}$	550 μA	940 μA , $f = 8 \text{ MHz}$
111	$8.2 \text{ MHz} < f \leq 25 \text{ MHz}$	2.6 mA	3.9 mA, $f = 25 \text{ MHz}$

When the crystal oscillator is first enabled, the external oscillator valid detector allows software to determine when the external system clock has stabilized. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure for starting the crystal is as follows:

1. Configure XTAL1 and XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD => 1.
4. Switch the system clock to the external oscillator.

SFR Definition 23.2. OSCICN: Internal Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY						
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	Varies	Varies	Varies	Varies	Varies	Varies

SFR Page = 0x0; SFR Address = 0xB2

Bit	Name	Function
7	IOSCEN	Internal Oscillator Enable. 0: Internal oscillator disabled. 1: Internal oscillator enabled.
6	IFRDY	Internal Oscillator Frequency Ready Flag. 0: Internal oscillator is not running at its programmed frequency. 1: Internal oscillator is running at its programmed frequency.
5:0	Reserved	Must perform read-modify-write.

Notes:

1. Read-modify-write operations such as ORL and ANL must be used to set or clear the enable bit of this register.
2. OSCBIAS (REG0CN.4) must be set to 1 before enabling the precision internal oscillator.

26.3.2. Contrast Control Mode 2 (Minimum Contrast Mode)

In Contrast Control Mode 2, a minimum contrast voltage is maintained, as shown in Figure 26.4. The VLCD supply is powered directly from VBAT as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to achieve the desired minimum contrast voltage on VLCD. Minimum Contrast Mode is selected using the following procedure:

1. Clear Bit 2 of the LCD0MSCN register to 0b (LCD0MSCN &= ~0x04)
2. Set Bit 0 of the LCD0MSCF register to 1b (LCD0MSCF |= 0x01)
3. Set Bit 3 of the LCD0PWR register to 1b (LCD0PWR |= 0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

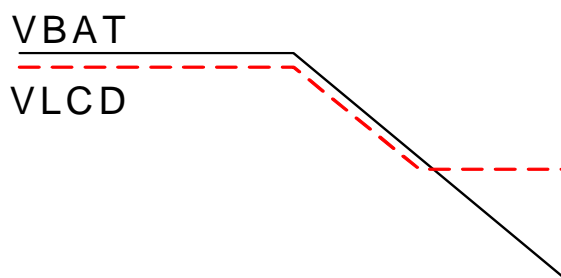


Figure 26.4. Contrast Control Mode 2

26.3.3. Contrast Control Mode 3 (Constant Contrast Mode)

In Contrast Control Mode 3, a constant contrast voltage is maintained. The VLCD supply is regulated to the programmed contrast voltage using a variable resistor between VBAT and VLCD as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to achieve the desired contrast voltage on VLCD. Constant Contrast Mode is selected using the following procedure:

1. Set Bit 2 of the LCD0MSCN register to 1b (LCD0MSCN |= 0x04)
2. Clear Bit 0 of the LCD0MSCF register to 0b (LCD0MSCF &= ~0x01)
3. Set Bit 3 of the LCD0PWR register to 1b (LCD0PWR |= 0x08)
4. Set Bit 7 of the LCD0VBMCN register to 1b (LCD0VBMCN |= 0x80)

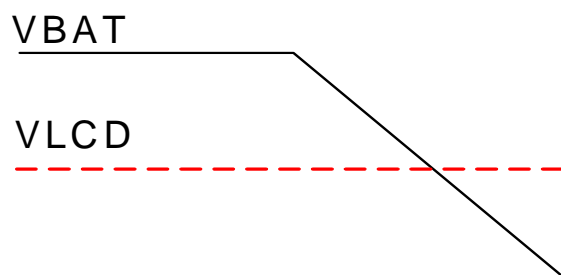


Figure 26.5. Contrast Control Mode 3

SFR Definition 26.11. LCD0TOGR: LCD0 Toggle Rate

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

SFR Page = 0x2; SFR Address = 0x9F

Bit	Name	Function
7:4	Unused	Read = 0000. Write = Don't Care.
3:0	TOGR[3:0]	<p>LCD Toggle Rate Divider.</p> <p>Sets the LCD Toggle Rate according to the following equation:</p> $\text{LCD Toggle Rate} = \frac{\text{Refresh Rate} \times \text{mux_mode} \times 2}{\text{Toggle Rate Divider}}$ <p>0000: Reserved. 0001: Reserved. 0010: Toggle Rate Divider is set to divide by 2. 0011: Toggle Rate Divider is set to divide by 4. 0100: Toggle Rate Divider is set to divide by 8. 0101: Toggle Rate Divider is set to divide by 16. 0110: Toggle Rate Divider is set to divide by 32. 0111: Toggle Rate Divider is set to divide by 64. 1000: Toggle Rate Divider is set to divide by 128. 1001: Toggle Rate Divider is set to divide by 256. 1010: Toggle Rate Divider is set to divide by 512. 1011: Toggle Rate Divider is set to divide by 1024. 1100: Toggle Rate Divider is set to divide by 2048. 1101: Toggle Rate Divider is set to divide by 4096. All other values reserved.</p>

28.1. Supporting Documents

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

1. The I²C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I²C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

28.2. SMBus Configuration

Figure 28.2 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.

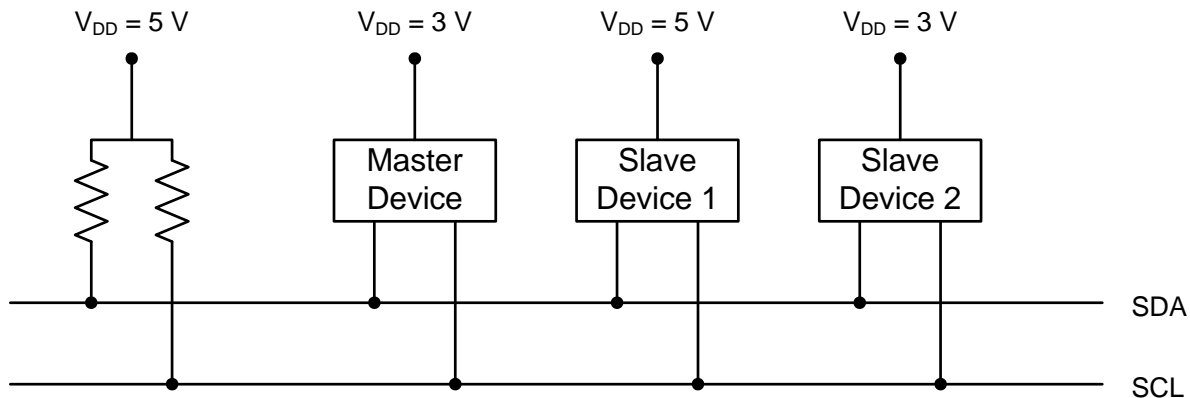


Figure 28.2. Typical SMBus Configuration

28.3. 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. Note that 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 28.3). 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 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)

Mode	Values Read				Current SMBus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X	1100
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	-
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	-
						End transfer with STOP and start another transfer.	1	1	X	-
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X	1000
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						Send NACK to indicate last byte, and send STOP.	0	1	0	-
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

Table 28.5. SMBus Status Decoding With Hardware ACK Generation Disabled (EHACK = 0)

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