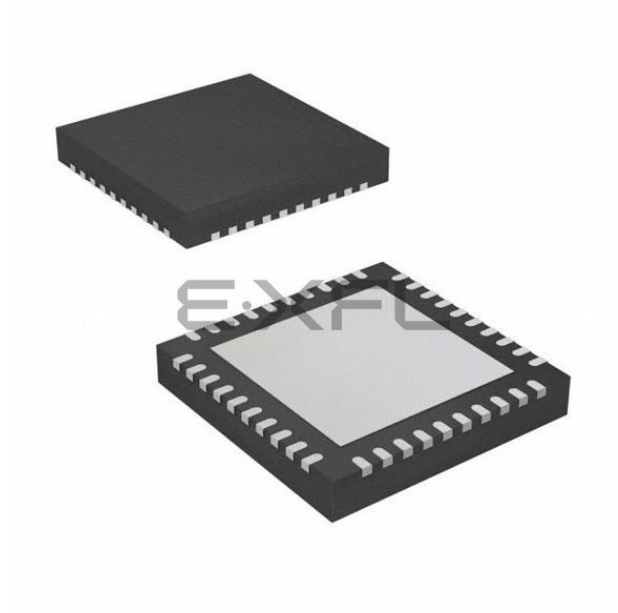


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, LCD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K 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	40-VFQFN Exposed Pad
Supplier Device Package	40-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f961-a-gm

Table 4.4. Digital Supply Current with DC-DC Converter Disabled (Continued)
–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current (Sleep Mode, SmaRTClock running, 32.768 kHz Crystal, LCD Contrast Mode 3 (2.7 V), charge pump enabled, 60 Hz refresh rate, driving 32 segment pins w/ no load)	1.8 V, T = 25 °C, static LCD 1.8 V, T = 25 °C, 2-Mux LCD 1.8 V, T = 25 °C, 3-Mux LCD 1.8 V, T = 25 °C, 4-Mux LCD	— — — —	1.3 1.8 1.8 2.0	— — — —	µA

Notes:

1. Active Current measure using typical code loop - Digital Supply Current depends upon the particular code being executed. Digital Supply Current depends on the particular code being executed. The values in this table are obtained with the CPU executing a mix of instructions in two loops: djnz R1, $, followed by a loop that accesses an SFR, and moves data around using the CPU (between accumulator and b-register). The supply current will vary slightly based on the physical location of this code in flash. As described in the Flash Memory chapter, it is best to align the jump addresses with a flash word address (byte location /4), to minimize flash accesses and power consumption.
2. Includes oscillator and regulator supply current.
3. Based on device characterization data; Not production tested.
4. Measured with one-shot enabled.
5. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x0F.
6. Using SmaRTClock osillator with external 32.768 kHz CMOS clock. Does not include crystal bias current.
7. Low-Power Idle mode current measured with CLKMODE = 0x04, PCON = 0x01, and PCLKEN = 0x00.

5.2.3. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a low power state, accumulates 1, 4, 8, 16, 32, or 64 using an internal Burst Mode clock (approximately 20 MHz), then re-enters a low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a low power state within a single system clock cycle, even if the system clock is slow (e.g. 32.768 kHz), or suspended.

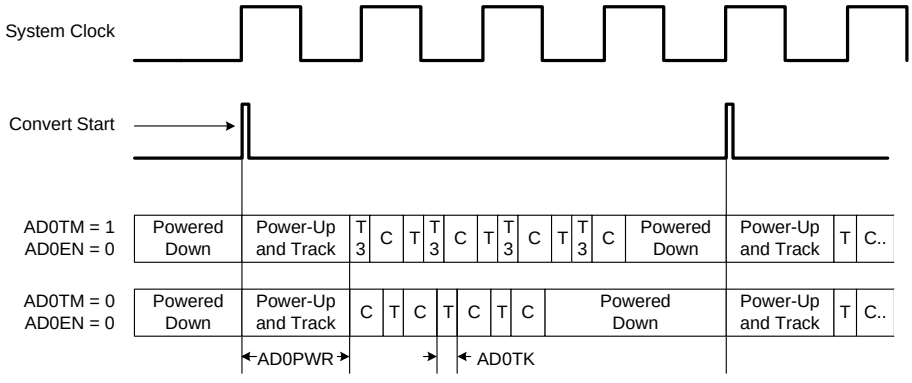
Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e. the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 5.3 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

In Burst Mode, tracking is determined by the settings in AD0PWR and AD0TK. The default settings for these registers will work in most applications without modification; however, settling time requirements may need adjustment in some applications. Refer to “5.2.4. Settling Time Requirements” on page 83 for more details.

Notes:

- Setting AD0TM to 1 will insert an additional 3 SAR clocks of tracking before each conversion, regardless of the settings of AD0PWR and AD0TK.
- When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.



T = Tracking set by AD0TK
T3 = Tracking set by AD0TM (3 SAR clocks)
C = Converting

Figure 5.3. Burst Mode Tracking Example with Repeat Count Set to 4

5.8. Temperature Sensor

An on-chip temperature sensor is included on the C8051F96x which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC mux channel should select the temperature sensor. The temperature sensor transfer function is shown in Figure 5.8. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 5.15. REF0CN: Voltage Reference Control. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to Table 4.12 for the slope and offset parameters of the temperature sensor.

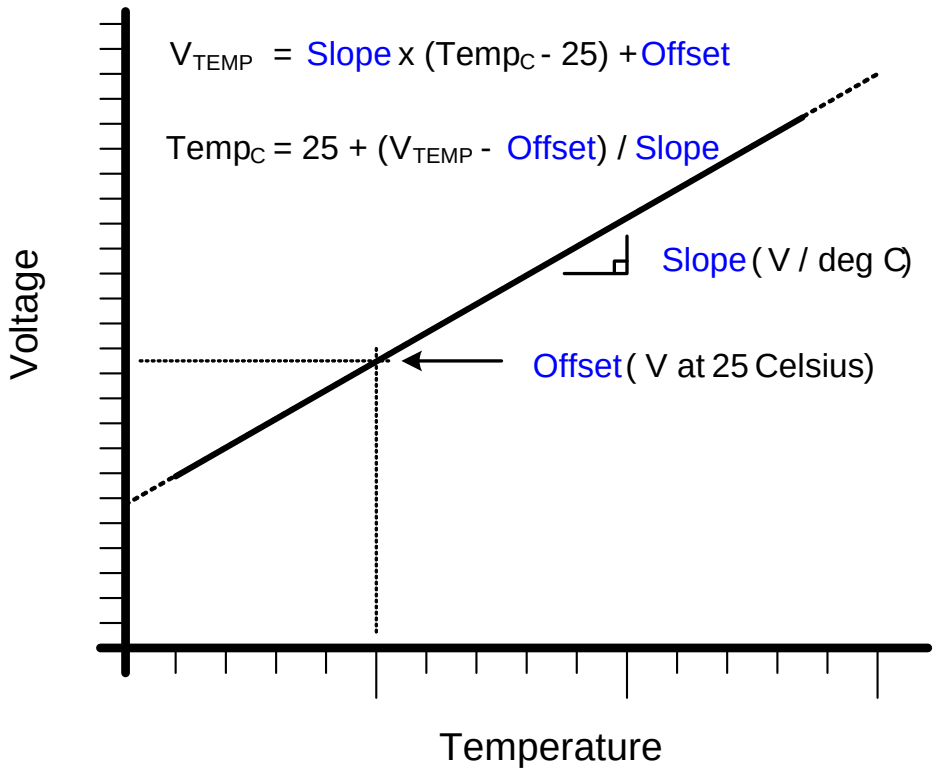


Figure 5.8. Temperature Sensor Transfer Function

5.8.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 4.13 for linearity specifications). For absolute temperature measurements, offset and/or gain calibration is recommended. Typically a 1-point (offset) calibration includes the following steps:

1. Control/measure the ambient temperature (this temperature must be known).
2. Power the device, and delay for a few seconds to allow for self-heating.
3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
4. Calculate the offset characteristics, and store this value in non-volatile memory for use with subsequent temperature sensor measurements.

7.5. Comparator Register Descriptions

The SFRs used to enable and configure the comparators are described in the following register descriptions. A Comparator must be enabled by setting the CPnEN bit to logic 1 before it can be used. From an enabled state, a comparator can be disabled and placed in a low power state by clearing the CPnEN bit to logic 0.

Important Note About Comparator Settings: False rising and falling edges can be detected by the Comparator while powering on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed. The Comparator Power Up Time is specified in Section “Table 4.16. Comparator Electrical Characteristics” on page 74.

SFR Definition 7.1. CPT0CN: Comparator 0 Control

Bit	7	6	5	4	3	2	1	0
Name	CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP[1:0]		CP0HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Page= 0x0; SFR Address = 0x9B

Bit	Name	Function
7	CP0EN	**Comparator0 Enable Bit.** 0: Comparator0 Disabled. 1: Comparator0 Enabled.
6	CP0OUT	**Comparator0 Output State Flag.** 0: Voltage on CP0+ < CP0–. 1: Voltage on CP0+ > CP0–.
5	CP0RIF	**Comparator0 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator0 Rising Edge has occurred since this flag was last cleared. 1: Comparator0 Rising Edge has occurred.
4	CP0FIF	**Comparator0 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator0 Falling-Edge has occurred since this flag was last cleared. 1: Comparator0 Falling-Edge has occurred.
3-2	CP0HYP[1:0]	**Comparator0 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1-0	CP0HYN[1:0]	**Comparator0 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

SFR Definition 7.6. CPT1MX: Comparator1 Input Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CMX1N[3:0]				CMX1P[3:0]			
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

SFR Page = 0x0; SFR Address = 0x9E

Bit	Name	Function			
7:4	CMX1N	**Comparator1 Negative Input Selection.** Selects the negative input channel for Comparator1.			
		0000:	P0.1	1000:	P2.1
		0001:	P0.3	1001:	P2.3
		0010:	P0.5	1010:	Reserved
		0011:	Reserved	1011:	Reserved
		0100:	Reserved	1100:	Compare
		0101:	Reserved	1101:	VBAT divided by 2
		0110:	P1.5	1110:	Digital Supply Voltage
		0111:	P1.7	1111:	Ground
3:0	CMX1P	**Comparator1 Positive Input Selection.** Selects the positive input channel for Comparator1.			
		0000:	P0.0	1000:	P2.0
		0001:	P0.2	1001:	P2.2
		0010:	P0.4	1010:	Reserved
		0011:	P0.6	1011:	Reserved
		0100:	Reserved	1100:	Compare
		0101:	Reserved	1101:	VBAT divided by 2
		0110:	P1.4	1110:	VBAT Supply Voltage
		0111:	P1.6	1111:	VDC Supply Voltage

10.5.1. Internal XRAM Only

When bits EMI0CF[3:2] are set to 00, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 8 kB boundaries. As an example, the addresses 0x2000 and 0x4000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

10.5.2. Split Mode without Bank Select

When bit EMI0CF.[3:2] are set to 01, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.5.3. Split Mode with Bank Select

When EMI0CF[3:2] are set to 10, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. The upper 8-bits of the Address Bus A[15:8] are determined by EMI0CN, and the lower 8-bits of the Address Bus A[7:0] are determined by R0 or R1. All 16-bits of the Address Bus A[15:0] are driven in “Bank Select” mode.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.5.4. External Only

When EMI0CF[3:2] are set to 11, all MOVX operations are directed to off-chip space. On-chip XRAM is not visible to the CPU. This mode is useful for accessing off-chip memory located between 0x0000 and the internal XRAM size boundary.

- 8-bit MOVX operations ignore the contents of EMI0CN. The upper Address bits A[15:8] are not driven (identical behavior to an off-chip access in “Split Mode without Bank Select” described above). This allows the user to manipulate the upper address bits at will by setting the Port state directly. The lower 8-bits of the effective address A[7:0] are determined by the contents of R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine the effective address A[15:0]. The full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

10.6. Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, $\overline{RD}$ and $\overline{WR}$ strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in SFR Definition 10.3, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for $\overline{RD}$ or $\overline{WR}$ pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for /ALE + 1 for $\overline{RD}$ or $\overline{WR}$ + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 10.2 lists the ac parameters for the External Memory Interface, and Figure 10.4 through Figure 10.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

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

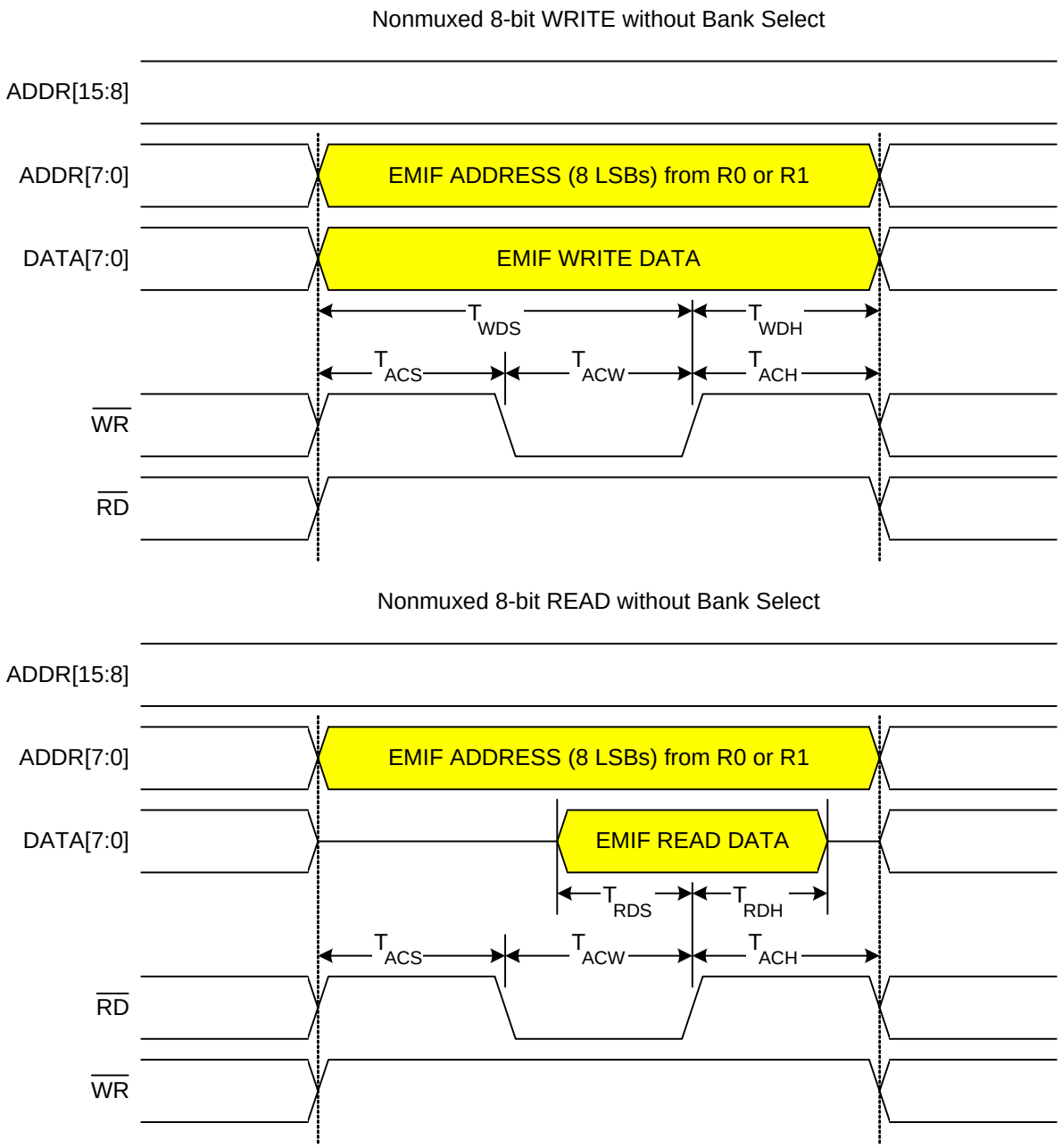


Figure 10.5. Non-multiplexed 8-bit MOVX without Bank Select Timing

SFR Definition 11.2. 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 occured on channel 6. 1: Full-length interrupt has not occured on channel 6.
5	CH5_INT	0: Full-length interrupt has not occured on channel 5. 1: Full-length interrupt has not occured on channel 5.
4	CH4_INT	**Channel 4 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 4. 1: Full-length interrupt has not occured on channel 4.
3	CH3_INT	**Channel 3 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 3. 1: Full-length interrupt has not occured on channel 3.
2	CH2_INT	**Channel 2 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 2. 1: Full-length interrupt has not occured on channel 2.
1	CH1_INT	**Channel 1 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 1. 1: Full-length interrupt has not occured on channel 1.
0	CH0_INT	**Channel 0 Full-Length Interrupt Flag.[1]** 0: Full-length interrupt has not occured on channel 0. 1: Full-length interrupt has not occured 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.

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 ($\overline{INT0}$)	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 ($\overline{INT1}$)	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 Comparator	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 17.6. EIP2: Extended Interrupt Priority 2

Bit	7	6	5	4	3	2	1	0
Name	PAES0	PENC0	PDMA0	PPC0	PSPI1	PRTC0F	PMAT	PWARN
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Page = All Pages; SFR Address = 0xF7

Bit	Name	Function
7	PAES0	**AES0 Interrupt Priority Control.** This bit sets the priority of the AES0 interrupt. 0: AES0 interrupt set to low priority level. 1: AES0 interrupt set to high priority level.
6	PENC0	**Encoder (ENC0) Interrupt Priority Control.** This bit sets the priority of the ENC0 interrupt. 0: ENC0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PDMA0	**DMA0 Interrupt Priority Control.** This bit sets the priority of the DMA0 interrupt. 0: DMA0 interrupt set to low priority level. 1: DMA0 interrupt set to high priority level.
4	PPC0	**Pulse Counter (PC0) Interrupt Priority Control.** This bit sets the priority of the PC0 interrupt. 0: PC0 interrupt set to low priority level. 1: PC0 interrupt set to high priority level.
3	PSPI0	**Serial Peripheral Interface (SPI1) Interrupt Priority Control.** This bit sets the priority of the SPI0 interrupt. 0: SPI1 interrupt set to low priority level. 1: SPI1 interrupt set to high priority level.
2	PRTC0F	**SmaRTClock Oscillator Fail Interrupt Priority Control.** This bit sets the priority of the SmaRTClock Alarm interrupt. 0: SmaRTClock Alarm interrupt set to low priority level. 1: SmaRTClock Alarm interrupt set to high priority level.
1	PMAT	**Port Match Interrupt Priority Control.** This bit sets the priority of the Port Match Event interrupt. 0: Port Match interrupt set to low priority level. 1: Port Match interrupt set to high priority level.
0	PWARN	**VDD/DC+ Supply Monitor Early Warning Interrupt Priority Control.** This bit sets the priority of the VDD/DC+ Supply Monitor Early Warning interrupt. 0: VDD/DC+ Supply Monitor Early Warning interrupt set to low priority level. 1: VDD/DC+ Supply Monitor Early Warning interrupt set to high priority level.

22.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to 1 and a MOVX write operation targets an address above the Lock Byte address.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above the Lock Byte address.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above the Lock Byte address.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section “18.3. Security Options” on page 252).
- A Flash write or erase is attempted while the V_{DD} Monitor is disabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

22.8. SmaRTClock (Real Time Clock) Reset

The SmaRTClock can generate a system reset on two events: SmaRTClock Oscillator Fail or SmaRTClock Alarm. The SmaRTClock Oscillator Fail event occurs when the SmaRTClock Missing Clock Detector is enabled and the SmaRTClock clock is below approximately 20 kHz. A SmaRTClock alarm event occurs when the SmaRTClock Alarm is enabled and the SmaRTClock timer value matches the ALARMn registers. The SmaRTClock can be configured as a reset source by writing a 1 to the RTC0RE flag (RSTSRC.7). The SmaRTClock reset remains functional even when the device is in the low power Suspend or Sleep mode. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

22.9. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit (RSTSRC.4). The SWRSF bit will read 1 following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

25.3. Programmable Pull-Up Resistors

The Pulse Counter features low-power pull-up resistors with a programmable resistance and duty-cycle. The average pull-up current will depend on the selected resistor, sample rate, and pull-up duty-cycle multiplier. Example code is available that will calculate the values for the Pull-Up configuration SFR (PC0PCF).

Table 25.1through Table 25.3 are used with Equation 25.1 to calculate the average pull-up resistor current. Table 25.4through Table 25.7 give the average current for all combinations.

$$I_{pull\text{-}up} = I_R \times D_{SR} \times D_{PU}$$

Equation 25.1. Average Pull-Up Current

Where:

I_R = Pull-up Resistor current selected by PC0PCF[4:2].

D_{SR} = Sample Rate Duty Cycle Multiplier selected by PC0MD[5:4].

D_{PU} = Pull-Up Duty Cycle Multiplier selected by PC0PCF[4:2].

Table 25.1. Pull-Up Resistor Current

PC0PCF[4:2]	I_R
000	0
001	1 μA
010	4 μA
011	16 μA
100	64 μA
101	256 μA
110	1 mA
111	4 mA

Table 25.2. Sample Rate Duty-Cycle Multiplier

PC0MD[5:4]	D_{SR}
000	1
001	1/2
010	1/4
011	1/8

Table 25.3. Pull-Up Duty-Cycle Multiplier

PC0PCF[4:2]	D_{PU}
000	1/4
001	3/8
010	1/2
011	3/4

26. LCD Segment Driver

C8051F96x devices contain an LCD segment driver and on-chip bias generation that supports static, 2-mux, 3-mux and 4-mux LCDs with 1/2 or 1/3 bias. The on-chip charge pump with programmable output voltage allows software contrast control which is independent of the supply voltage. LCD timing is derived from the SmaRTClock oscillator to allow precise control over the refresh rate.

The C8051F96x uses special function registers (SFRs) to store the enabled/disabled state of individual LCD segments. All LCD waveforms are generated on-chip based on the contents of the LCD0Dn registers An LCD blinking function is also supported. A block diagram of the LCD segment driver is shown in Figure 26.1.

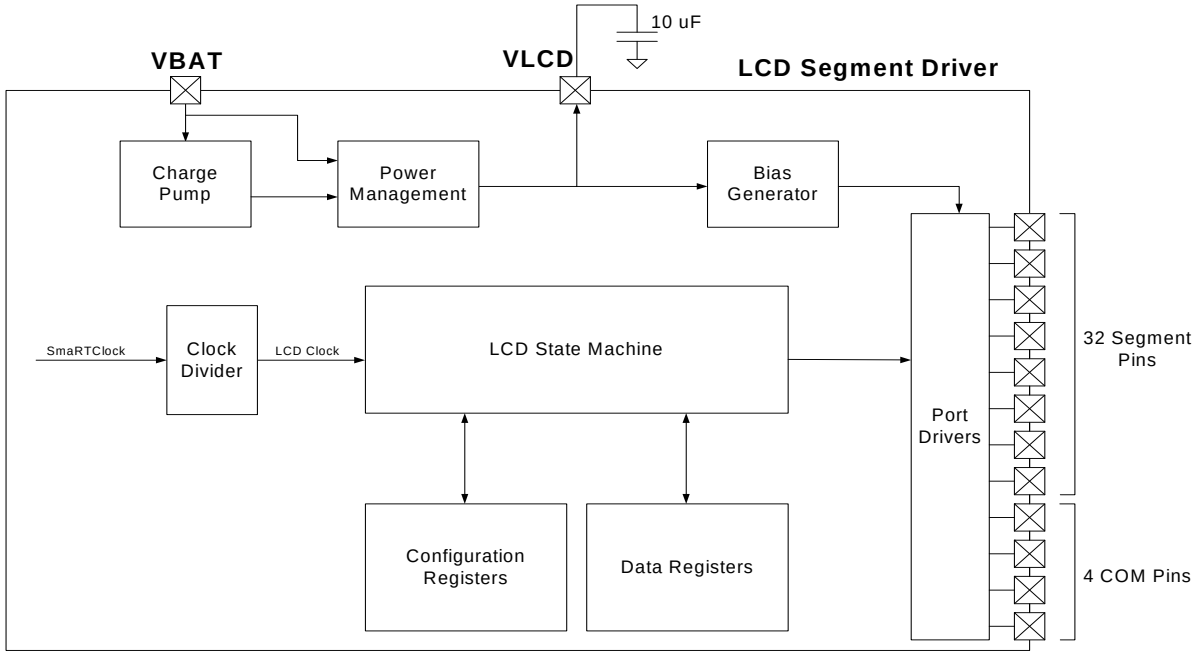


Figure 26.1. LCD Segment Driver Block Diagram

26.1. Configuring the LCD Segment Driver

The LCD segment driver supports multiple mux options: static, 2-mux, 3-mux, and 4-mux mode. It also supports 1/2 and 1/3 bias options. The desired mux mode and bias is configured through the LCD0CN register. A divide value may also be applied to the SmaRTClock output before being used as the LCD0 clock source.

The following procedure is recommended for using the LCD Segment Driver:

1. Initialize the SmaRTClock and configure the LCD clock divide settings in the LCD0CN register.
2. Determine the GPIO pins which will be used for the LCD function.
3. Configure the Port I/O pins to be used for LCD as Analog I/O.
4. Configure the LCD size, mux mode, and bias using the LCD0CN register.
5. Enable the LCD bias and clock gate by writing 0x50 to the LCD0MSCN register.
6. Configure the device into the desired Contrast Control Mode.
7. If VIO is internally or externally shorted to VBAT, disable the VLCD/VIO Supply Comparator using the

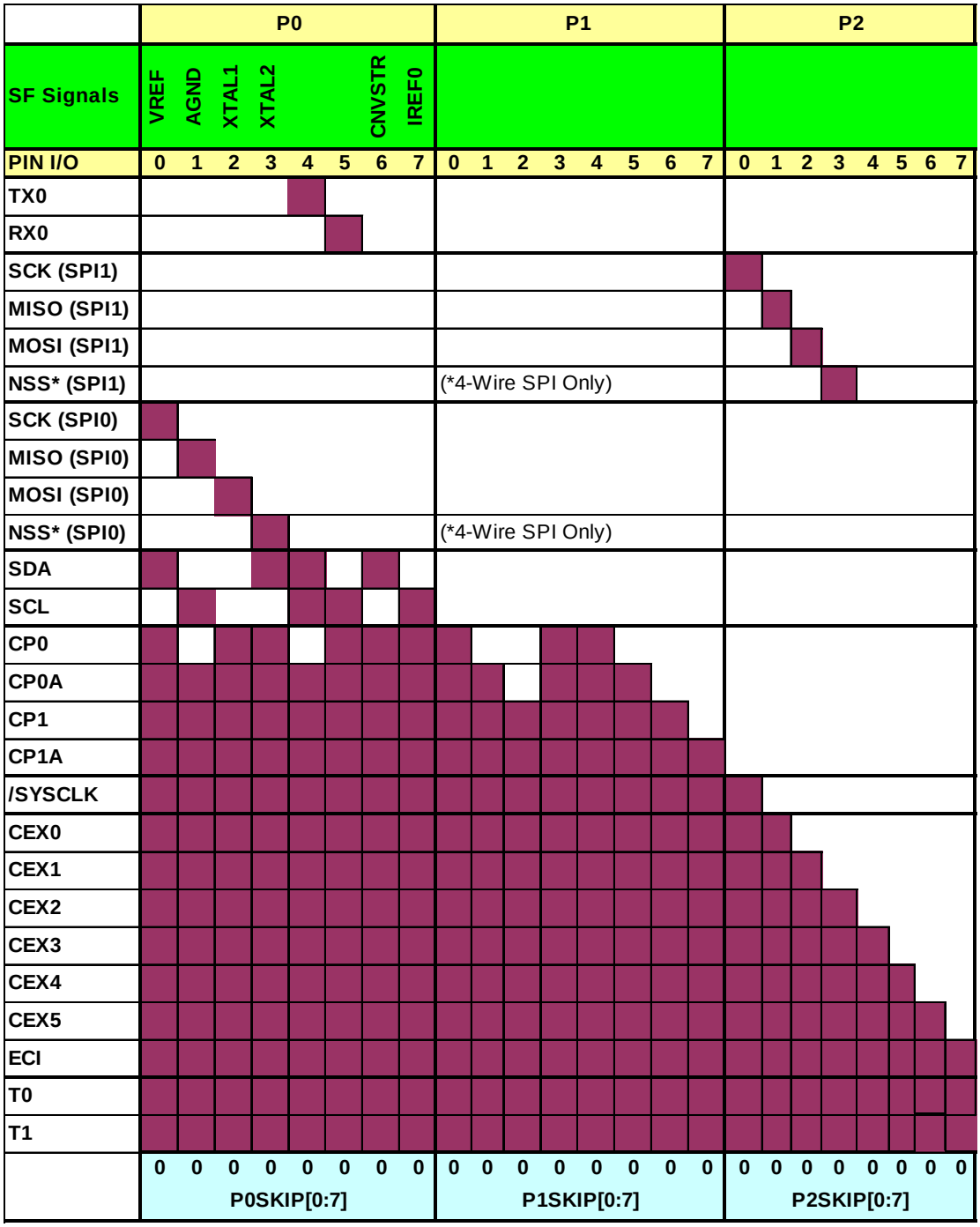


Figure 27.3. Crossbar Priority Decoder with No Pins Skipped

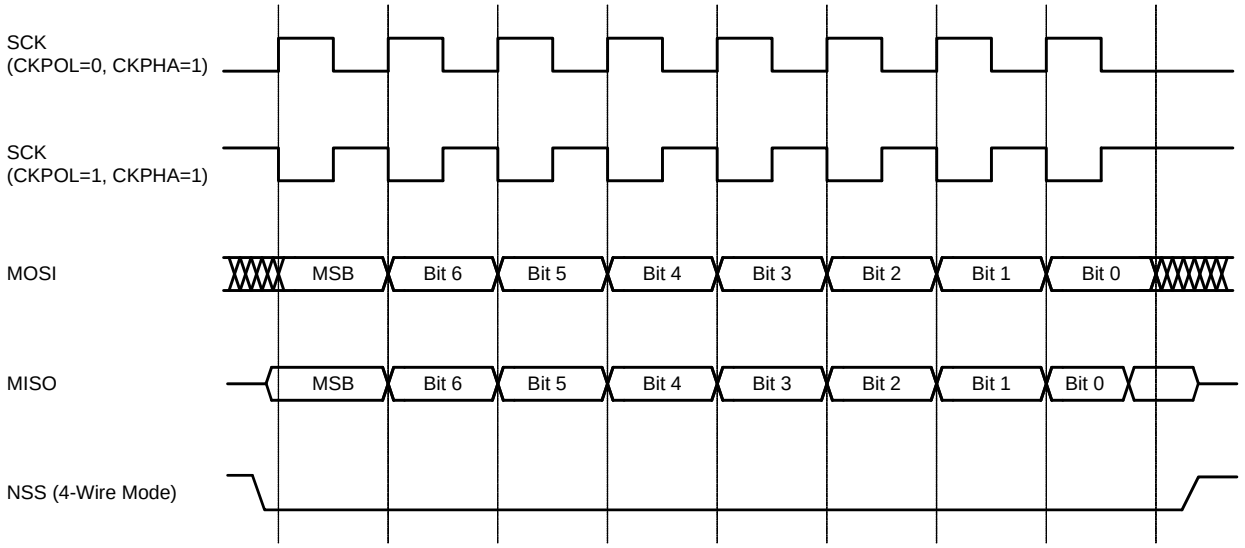


Figure 30.7. Slave Mode Data/Clock Timing (CKPHA = 1)

30.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 30.3. 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 Page = 0x0; 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 30.4. 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 Page = 0x0; 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.

31.10. Slave Mode DMA Transfers

SPI1 also supports using the DMA with Slave mode. The maximum SPI bit rate for a bidirectional Slave mode transfer is SYSCLK/10.

In master mode, the master is responsible for initiating the transfer, clocking the data, managing the NSS pin, and has control over the number of bytes transferred. In slave mode, the slave depends on the master for the clock and NSS signal. The slave also depends on the master to set the time between bytes and the number of bytes per transfer.

Firmware implementations of a SPI slave often have some restrictions on the time between bytes. When using SPI0 in slave mode, an interrupt service routine commonly processes each byte received. This imposes a limitation on the time between bytes. When using the SPI in Slave mode with the DMA, the time between bytes must be long enough to accommodate the DMA latency.

The time between bytes in master mode and the minimum time required between bytes in slave mode will depend on the DMA latency. The DMA latency will depend on a number of factors - the CPU state, the number of active DMA channels, and the DMA channel priority. Using only the two required DMA channels and putting the CPU in Idle mode will provide the lowest latency. If the CPU is actively executing instructions, the DMA may have to wait for the current instruction to execute before it can complete a transfer. If other DMA channels are active, the SPI DMA channels may have to wait for other DMA transfers to complete. This could be a very long time for long DMA transfers. Assigning the SPI to the first two DMA channels will ensure they have the highest DMA priority.

Note that in master mode, the time between bytes may prolong the DMA transfer, but does not usually result in data loss. In slave mode, the slave may drop data if the DMA cannot keep up with the master data coming in. Since the SPI slave data rate is limited to SYSCLK/10 and the longest instruction is 8 clock cycles, a delay between bytes of one SPI clock will prevent data loss. Using a SPI DMA slave with additional active DMA channels may result in data loss and is not recommended.

31.11. Bidirectional SPI Slave Mode DMA Transfer

A bidirectional SPI Slave mode DMA transfer will transfer a specified number of bytes out on the MISO pin and also receive the same number of bytes on the MOSI pin. The MISO data must be stored in XRAM before initiating the DMA transfers. After the complete transfer, the MOSI data will be stored in XRAM.

Since the MISO data must be stored in XRAM before the transfer, the MISO data is fixed and should not depend on the MOSI data received in the same transfer. The protocol designer should carefully consider this behavior when designing a SPI slave protocol. Firmware can easily modify the MISO data after each message. For example, one message can request data and a second message can read the data previously requested. This approach is much simpler and more efficient than attempting to modify the MISO data buffer on-the-fly.

If the slave transfer is a fixed constant length, the DMA interrupt will indicate one complete transfer. Firmware may implement a variable length slave transfer using an external interrupt connected to the NSS signal. In this case, firmware may use the DMA interrupt for a buffer overflow condition.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701

Please visit the Silicon Labs Technical Support web page:
https://www.silabs.com/support/pages/contacttechnicalsupport.aspx
and register to submit a technical support request.