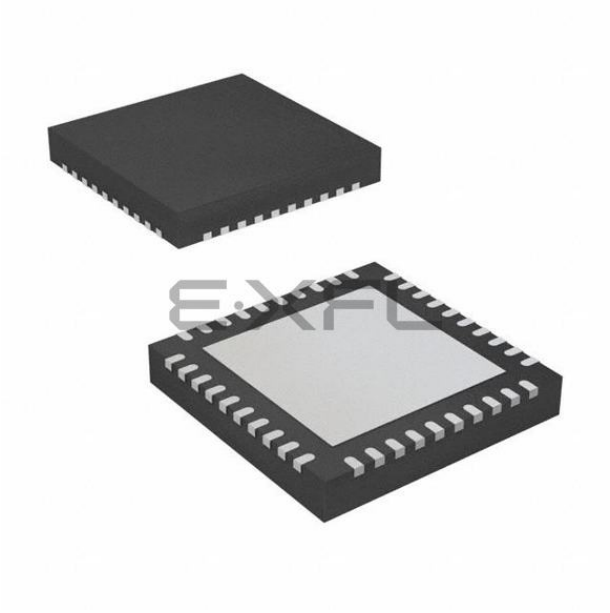


Welcome to E-XFL.COM

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-gmr

SFR Definition 10.3. EMI0TC: External Memory Timing Control

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

SFR Page = 0x0; SFR Address = 0xAF

Bit	Name	Function
7:6	EAS[1:0]	EMIF Address Setup Time Bits. 00: Address setup time = 0 SYSCLK cycles. 01: Address setup time = 1 SYSCLK cycle. 10: Address setup time = 2 SYSCLK cycles. 11: Address setup time = 3 SYSCLK cycles.
5:2	EWR[3:0]	EMIF $\overline{WR}$ and $\overline{RD}$ Pulse-Width Control Bits. 0000: $\overline{WR}$ and $\overline{RD}$ pulse width = 1 SYSCLK cycle. 0001: $\overline{WR}$ and $\overline{RD}$ pulse width = 2 SYSCLK cycles. 0010: $\overline{WR}$ and $\overline{RD}$ pulse width = 3 SYSCLK cycles. 0011: $\overline{WR}$ and $\overline{RD}$ pulse width = 4 SYSCLK cycles. 0100: $\overline{WR}$ and $\overline{RD}$ pulse width = 5 SYSCLK cycles. 0101: $\overline{WR}$ and $\overline{RD}$ pulse width = 6 SYSCLK cycles. 0110: $\overline{WR}$ and $\overline{RD}$ pulse width = 7 SYSCLK cycles. 0111: $\overline{WR}$ and $\overline{RD}$ pulse width = 8 SYSCLK cycles. 1000: $\overline{WR}$ and $\overline{RD}$ pulse width = 9 SYSCLK cycles. 1001: $\overline{WR}$ and $\overline{RD}$ pulse width = 10 SYSCLK cycles. 1010: $\overline{WR}$ and $\overline{RD}$ pulse width = 11 SYSCLK cycles. 1011: $\overline{WR}$ and $\overline{RD}$ pulse width = 12 SYSCLK cycles. 1100: $\overline{WR}$ and $\overline{RD}$ pulse width = 13 SYSCLK cycles. 1101: $\overline{WR}$ and $\overline{RD}$ pulse width = 14 SYSCLK cycles. 1110: $\overline{WR}$ and $\overline{RD}$ pulse width = 15 SYSCLK cycles. 1111: $\overline{WR}$ and $\overline{RD}$ pulse width = 16 SYSCLK cycles.
1:0	EAH[1:0]	EMIF Address Hold Time Bits. 00: Address hold time = 0 SYSCLK cycles. 01: Address hold time = 1 SYSCLK cycle. 10: Address hold time = 2 SYSCLK cycles. 11: Address hold time = 3 SYSCLK cycles.

Table 12.2. Example 32-bit CRC Outputs

Input	Output
0x63	0xF9462090
0xAA, 0xBB, 0xCC	0x41B207B3
0x00, 0x00, 0xAA, 0xBB, 0xCC	0x78D129BC

12.3. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should select the desired polynomial and set the initial value of the result. Two polynomials are available: 0x1021 (16-bit) and 0x04C11DB7 (32-bit). The CRC0 result may be initialized to one of two values: 0x00000000 or 0xFFFFFFFF. The following steps can be used to initialize CRC0.

1. Select a polynomial (Set CRC0SEL to 0 for 32-bit or 1 for 16-bit).
2. Select the initial result value (Set CRC0VAL to 0 for 0x00000000 or 1 for 0xFFFFFFFF).
3. Set the result to its initial value (Write 1 to CRC0INIT).

12.4. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more flash sectors. The following steps can be used to automatically perform a CRC on flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. If necessary, set the IFBANK bits in the PSBANK for the desired code bank.
3. Write the index of the starting page to CRC0AUTO.
4. Set the AUTOEN bit in CRC0AUTO.
5. Write the number of flash sectors to perform in the CRC calculation to CRC0CNT.
   Note: Each flash sector is 1024 bytes.
6. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes.
7. Clear the AUTOEN bit in CRC0AUTO.
8. Read the CRC result using the procedure below.

Setting the IFBANK bits in the PSBANK SFR is only necessary when accessing the upper banks on 128 kB code bank devices (‘F960/1/2/3). Multiple CRCs are required to cover the entire 128 kB Flash array. When writing to the PSBANK SFR, the code initiating the auto CRC of flash must be executing from the common area.

12.5. Accessing the CRC0 Result

The internal CRC0 result is 32-bits (CRC0SEL = 0b) or 16-bits (CRC0SEL = 1b). The CRC0PNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

14.6.4.1. CBC Decryption using DMA

Normally, the AES block is used with the DMA. This provides the best performance and lowest power consumption. Code examples are provided in 8051 compiler independent C code using the DMA. It is highly recommended to use with the code examples. The steps are documented in the datasheet for completeness.

- Prepare decryption Key, initialization vector, and data to be decrypted in xram.
- The initialization vector should be located immediately before the data to be decrypted to decrypt multiple blocks.
- Reset AES module by clearing bit 2 of AES0BCFG.
- Disable the first four DMA channels by clearing bits 0 to 3 in the DMA0EN sfr.
- Configure the first DMA channel for the AES0KIN sfr
  - Select the first DMA channel by writing 0x00 to the DMA0SEL sfr
  - Configure the first DMA channel to move xram to AES0KIN sfr by writing 0x05 to the DMA0NCF sfr
  - Write 0x01 to DMA0NMD to enable wrapping
  - Write the xram location of decryption key to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the key length in bytes to DMA0NSZL sfr
  - Clear the DMA0NSZH sfr
  - Clear the DMA0NAOH and DMA0NAOL sfrs
- Configure the second DMA channel for the AES0BIN sfr.
  - Select the second DMA channel by writing 0x01 to the DMA0SEL sfr.
  - Configure the second DMA channel to move xram to AES0BIN sfr by writing 0x06 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of the data to be decrypted to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the third DMA channel for the AES0XIN sfr.
  - Select the third DMA channel by writing 0x02 to the DMA0SEL sfr.
  - Configure the third DMA channel to move xram to AES0XIN sfr by writing 0x07 to the DMA0NCF sfr.
  - Clear DMA0NMD to disable wrapping.
  - Write the xram address of initialization vector to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Configure the fourth DMA channel for the AES0YOUT sfr
  - Select the fourth channel by writing 0x03 to the DMA0SEL sfr
  - Configure the fourth DMA channel to move the contents of the AES0YOUT sfr to xram by writing 0x08 to the DMA0NCF sfr
  - Enable transfer complete interrupt by setting bit 7 of DMA0NCF sfr
  - Clear DMA0NMD to disable wrapping
  - Write the xram address for decrypted data to the DMA0NBAH and DMA0NBAL sfrs.
  - Write the number of bytes to be decrypted in multiples of 16 bytes to the DMA0NSZH and DMA0NSZL sfrs.
  - Clear the DMA0NAOH and DMA0NAOL sfrs.
- Clear first four DMA interrupts by clearing bits 0 to 2 in the DMA0INT sfr.
- Enable first four DMA channels setting bits 0 to 2 in the DMA0EN sfr
- Configure the AES Module data flow for XOR on output data by writing 0x02 to the AES0DCFG sfr.
- Write key size to bits 1 and 0 of the AES0BCFG
- Configure the AES core for decryption by clearing bit 2 of AES0BCFG
- Initiate the decryption operation be setting bit 3 of AES0BCFG
- Wait on the DMA interrupt from DMA channel 3
- Disable the AES Module by clearing bit 2 of AES0BCFG
- Disable the DMA by writing 0x00 to DMA0EN

SFR Definition 14.2. AES0DCFG: AES Data Configuration

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

SFR Address = 0xEA; SFR page = 0x2; Not bit-Addressable

Bit	Name	Function
2:1	OUTSEL[1:0]	DATA Select. These bits select the output data source for the AES0YOUT sfr. 00: Direct AES Data 01: AES Data XOR with AES0XIN 10: Inverse Key 11: reserved
0	XORIN	XOR Input Enable. Setting this bit with enable the XOR data path on the AES input. If enabled, AES0BIN will be XORed with the AES0XIN and the results will feed into the AES data input. Clearing this bit to 0 will disable the XOR gate on the input. The contents of AES0BIN will go directly into the AES data input.

15. Encoder/Decoder

The Encoder/Decoder consists of three 8-bit data registers, a control register and an encoder/decoder logic block.

The size of the input data depends on the mode. The input data for Manchester encoding is one byte. For Manchester decoding it is two bytes. Three-out-of-Six encoding is two bytes. Three-out-of six decoding is three bytes.

The output size also depends on the mode selected. The input and output data size are shown below:

Table 15.1. Encoder Input and Output Data Sizes

	Input Data Size	Output Data Size
Operation	Bytes	Bytes
Manchester Encode	1	2
Manchester Decode	2	1
Three out of Six Encode	2	3
Three out of Six Decode	3	2

The input and output data is always right justified. So for Manchester mode the input uses only ENC0L and the output data is only in ENC0M and ENC0L. ENC0H is not used for Manchester mode

20. On-Chip DC-DC Buck Converter (DC0)

C8051F96x devices include an on-chip step down dc-dc converter to efficiently utilize the energy stored in the battery, thus extending the operational life time. The dc-dc converter is a switching buck converter with an input supply of 1.8 to 3.8 V and an output that is programmable from 1.8 to 3.5 V in steps of 0.1 V. The battery voltage should be at least 0.4 V higher than the programmed output voltage. The programmed output voltage has a default value of 1.9 V. The dc-dc converter can supply up to 250 mW. The dc-dc converter can be used to power the MCU and/or external devices in the system (e.g., an RF transceiver).

The dc-dc converter has a built in voltage reference and oscillator, and will automatically limit or turn off the switching activity in case the peak inductor current rises beyond a safe limit or the output voltage rises above the programmed target value. This allows the dc-dc converter output to be safely overdriven by a secondary power source (when available) in order to preserve battery life. When enabled, the dc-dc converter can source current into the output capacitor, but cannot sink current. The dc-dc converter's settings can be modified using SFR registers described in Section 20.8.

Figure 20.1 shows a block diagram of the buck converter.

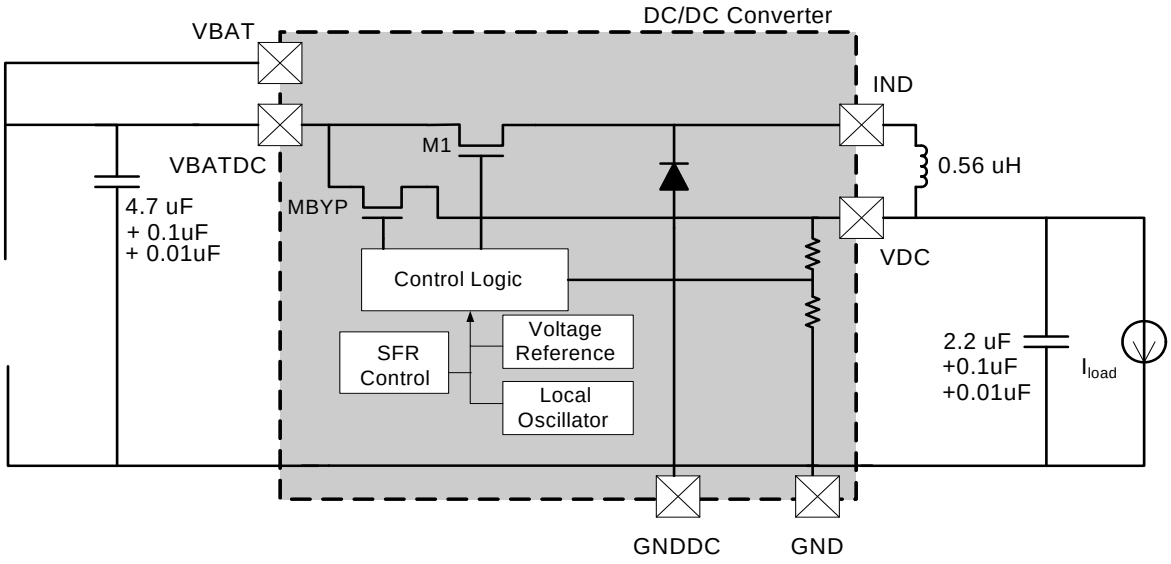


Figure 20.1. Step Down DC-DC Buck Converter Block Diagram

SFR Definition 20.3. DC0MD: DC-DC Converter Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved	ILIMIT			FORBYP	AUTOBYP	Reserved	DC0EN
Type	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	0	0	0

SFR Page = 0x2; SFR Address = 0xB3

Bit	Name	Function
7	Reserved	Read = 0b; Must write 0b.
6:4	ILIMIT	Peak Current Limit Threshold. 000: Reserved 001: Peak Inductor current is limited to 200 mA 010: Peak Inductor current is limited to 300 mA 011: Peak Inductor current is limited to 400 mA 100: Peak Inductor current is limited to 500 mA 101: Peak Inductor current is limited to 600 mA 110: Reserved 111: Reserved
3	FORBYP	Enable Forced Bypass Mode. 0: Forced bypass mode is disabled. 1: Forced bypass mode is enabled.
2	AUTOBYP	Enable Automatic Bypass Mode. 0: Automatic Bypass mode is disabled. 1: Automatic bypass mode is enabled.
1	Reserved	Read = 1b; Must write 1b.
0	DC0EN	DC-DC Converter Enable. 0: DC-DC converter is disabled. 1: DC-DC converter is enabled.

SFR Definition 25.1. PC0MD: PC0 Mode Configuration

Bit	7	6	5	4	3	2	1	0
Name	PCMODE[1:0]		PCRATE[1:0]		DUALCMPL	STPCNTFLTR	DUALSTCH	
Type	R/W		R/W		R/W	R	R/W	R
Reset	0	0	0	0	0	1	0	0

SFR Address = 0xD9; SFR Page = 0x2

Bit	Name	Function
7:6	PCMODE[1:0]	Counter Mode 00: Pulse Counter disabled. 01: Single Counter mode. 10: Dual Counter mode. 11: Quadrature Counter mode.
5:4	PCRATE[1:0]	PC Sample Rate 00: 250 µs 01: 500 µs 10: 1 ms 11: 2 ms
3	Reserved	
2	STPCNTFLTR	Stop Counting on Flutter (Only valid for quadrature counter mode and DUALSTCH off.) 0: Disabled. 1: Enabled.
1	DUALSTCH	Dual Mode Switch During Flutter (Only valid for quadrature counter mode.) 0: Disabled—quadrature mode remains set during flutter. 1: Enabled—quadrature mode changes to dual during flutter.
0	Reserved	

Note that writing to this register will clear the counter registers PC0CTR0H:M:L and PC0CTR1H:M:L.

SFR Definition 26.4. LCD0MSCN: LCD0 Master Control

Bit	7	6	5	4	3	2	1	0
Name		BIASEN	DCBIASOE	CLKOE		LOWDRV	LCDRST	LCDEN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xAB

Bit	Name	Function
7	Reserved	Read = 0b. Must write 0b.
6	BIASEN	LCD0 Bias Enable. LCD0 bias may be disabled when using a static LCD (single backplane), contrast control mode 1 (Bypass Mode) is selected, and the VLCD/VIO Supply Comparator is disabled (LCD0CF.5 = 1). It is required for all other modes. 0: LCD0 Bias is disabled. 1: LCD0 Bias is enabled
5	DCBIASOE	DCDC Converter Bias Output Enable. (Note 1) 0: The bias for the DCDC converter is gated off. 1: LCD0 provides the bias for the DCDC converter.
4	CLKOE	LCD Clock Output Enable. 0: The clock signal to the LCD0 module is gated off. 1: The SmaRTClock provides the undivided clock to the LCD0 Module.
3	Reserved	Read = 0b. Must write 0b.
2	LOWDRV	Charge Pump Reduced Drive Mode. This bit should be set to 1 in Contrast Control Mode 3 and Mode 4 for minimum power consumption. This bit may be set to 0 in these modes to support higher load current requirements. 0: The charge pump operates at full power. 1: The charge pump operates at reduced power.
1	LCDRST	LCD0 Reset. Writing a 1 to this bit will clear all the LCD0Dn registers to 0x00. This bit must be cleared by software.
0	LCDEN	LCD0 Enable. 0: LCD0 is disabled. 1: LCD0 is enabled.
Note 1: To same bias generator is shared by the DCDC Converter and LCD0.		

26.5. Setting the LCD Refresh Rate

The clock to the LCD0 module is derived from the SmaRTClock and may be divided down according to the settings in the LCD0CN register. The LCD refresh rate is derived from the LCD0 clock and can be programmed using the LCD0DIVH:LCD0DIVL registers. The LCD mux mode must be taken into account when determining the prescaler value. See the LCD0DIVH/LCD0DIVL register descriptions for more details. For maximum power savings, choose a slow LCD refresh rate and the minimum LCD0 clock frequency. For the least flicker, choose a fast LCD refresh rate.

SFR Definition 26.8. LCD0CLKDIVH: LCD0 Refresh Rate Prescaler High Byte

Bit	7	6	5	4	3	2	1	0
Name							LCD0DIV[9:8]	
Type	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 = 0xAA

Bit	Name	Function
7:2	Unused	Read = 000000. Write = Don't Care.
1:0	LCD0DIV[9:8]	LCD Refresh Rate Prescaler . Sets the LCD refresh rate according to the following equation: $\text{LCD Refresh Rate} = \frac{\text{LCD0 Clock Frequency}}{4 \times \text{mux_mode} \times (LCD0DIV + 1)}$

SFR Definition 26.9. LCD0CLKDIVL: LCD Refresh Rate Prescaler L ow Byte

Bit	7	6	5	4	3	2	1	0
Name	LCD0DIV[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Page = 0x2; SFR Address = 0xA9

Bit	Name	Function
7:0	LCD0DIV[7:0]	LCD Refresh Rate Prescaler . Sets the LCD refresh rate according to the following equation: $\text{LCD Refresh Rate} = \frac{\text{LCD0 Clock Frequency}}{4 \times \text{mux_mode} \times (LCD0DIV + 1)}$

SFR Definition 29.1. SCON0: Serial Port 0 Control

Bit	7	6	5	4	3	2	1	0
Name	S0MODE		MCE0	REN0	TB80	RB80	TI0	RI0
Type	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

SFR Page = 0x0; SFR Address = 0x98; Bit-Addressable

Bit	Name	Function
7	S0MODE	Serial Port 0 Operation Mode. Selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate. 1: 9-bit UART with Variable Baud Rate.
6	Unused	Read = 1b. Write = Don't Care.
5	MCE0	Multiprocessor Communication Enable. For Mode 0 (8-bit UART): Checks for valid stop bit. 0: Logic level of stop bit is ignored. 1: RI0 will only be activated if stop bit is logic level 1. For Mode 1 (9-bit UART): Multiprocessor Communications Enable. 0: Logic level of ninth bit is ignored. 1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.
4	REN0	Receive Enable. 0: UART0 reception disabled. 1: UART0 reception enabled.
3	TB80	Ninth Transmission Bit. The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB80	Ninth Receive Bit. RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI0	Transmit Interrupt Flag. Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.
0	RI0	Receive Interrupt Flag. Set to 1 by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to 1 causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

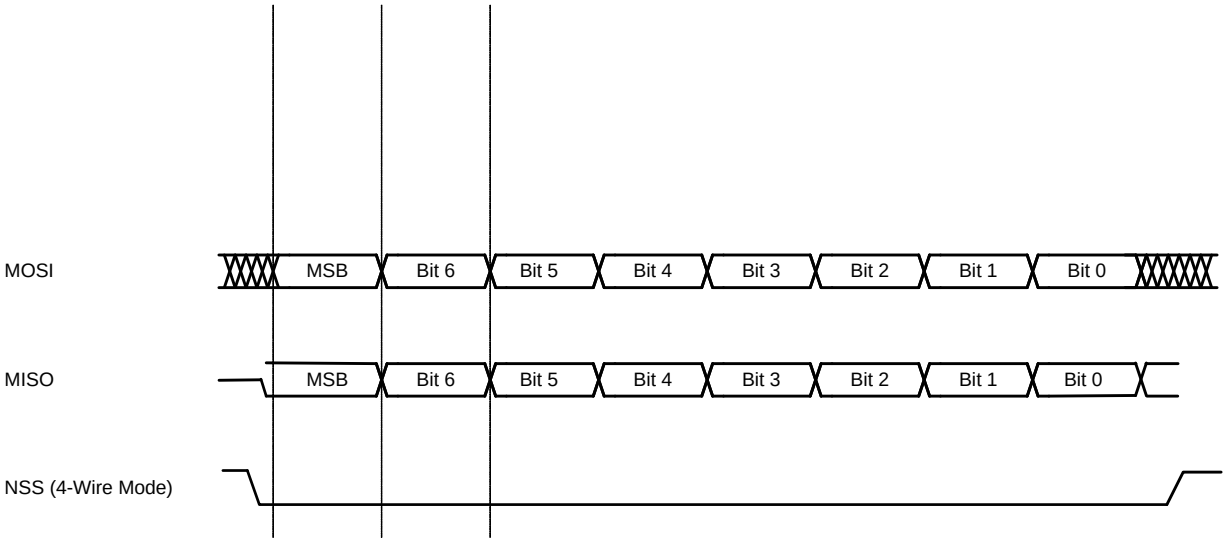
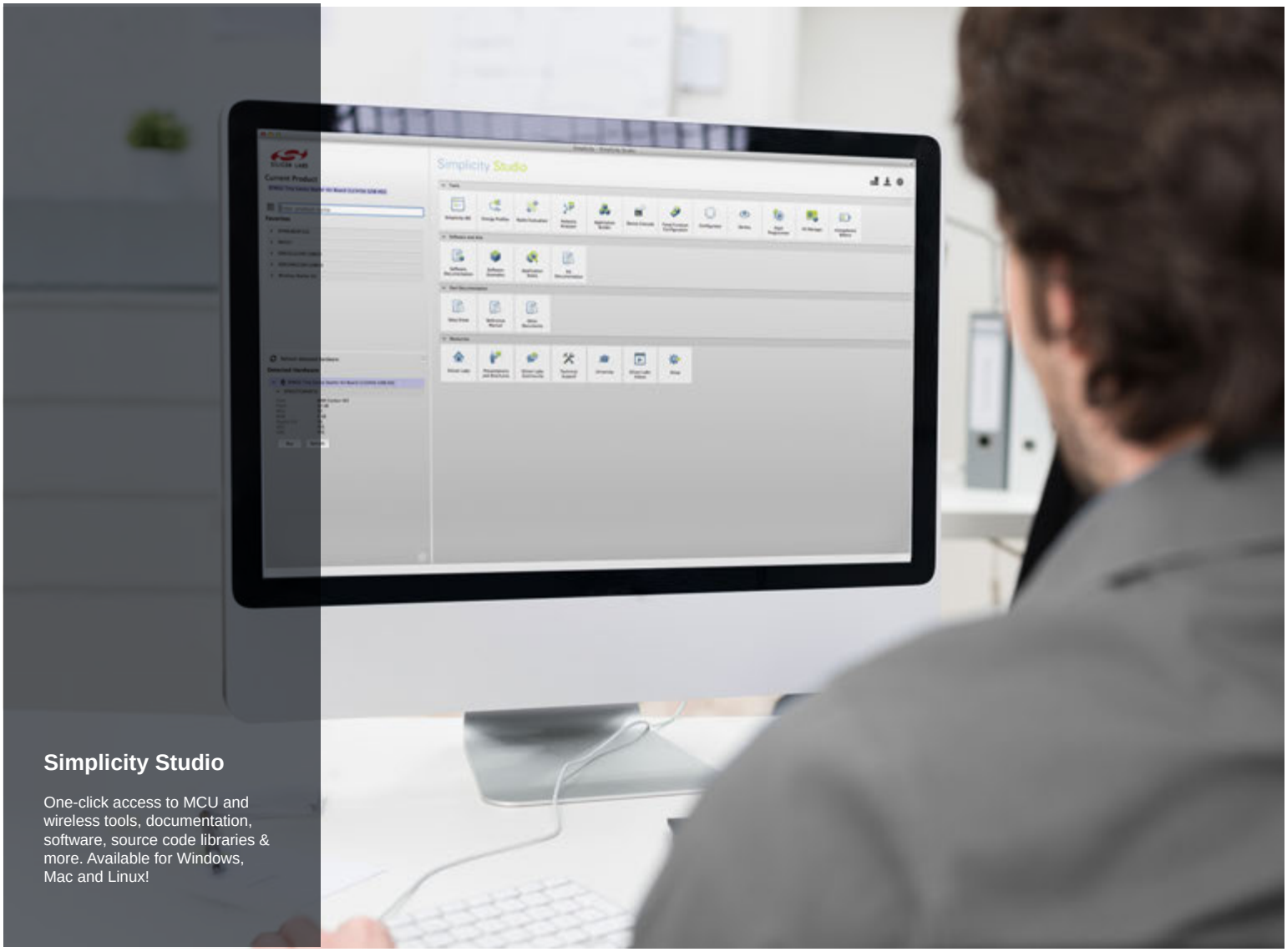


Figure 31.6. Slave Mode Data/Clock Timing (CKPHA = 0)

Figure 31.7. Slave Mode Data/Clock Timing (CKPHA = 1)



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are not designed or authorized for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com