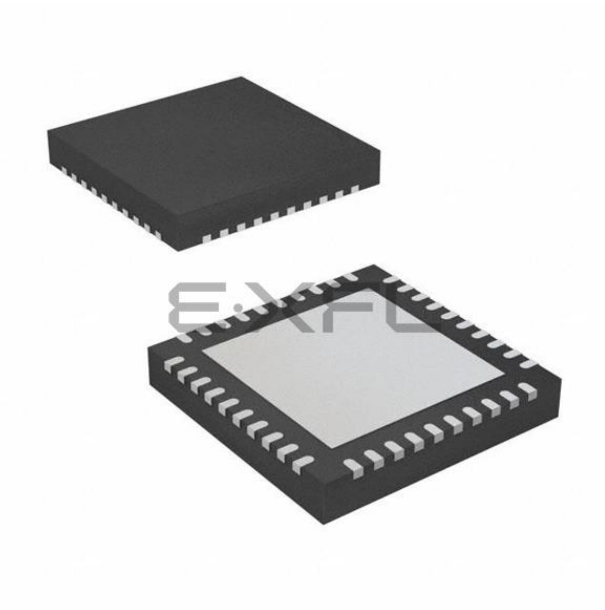


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	33
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/c8051f508-imr

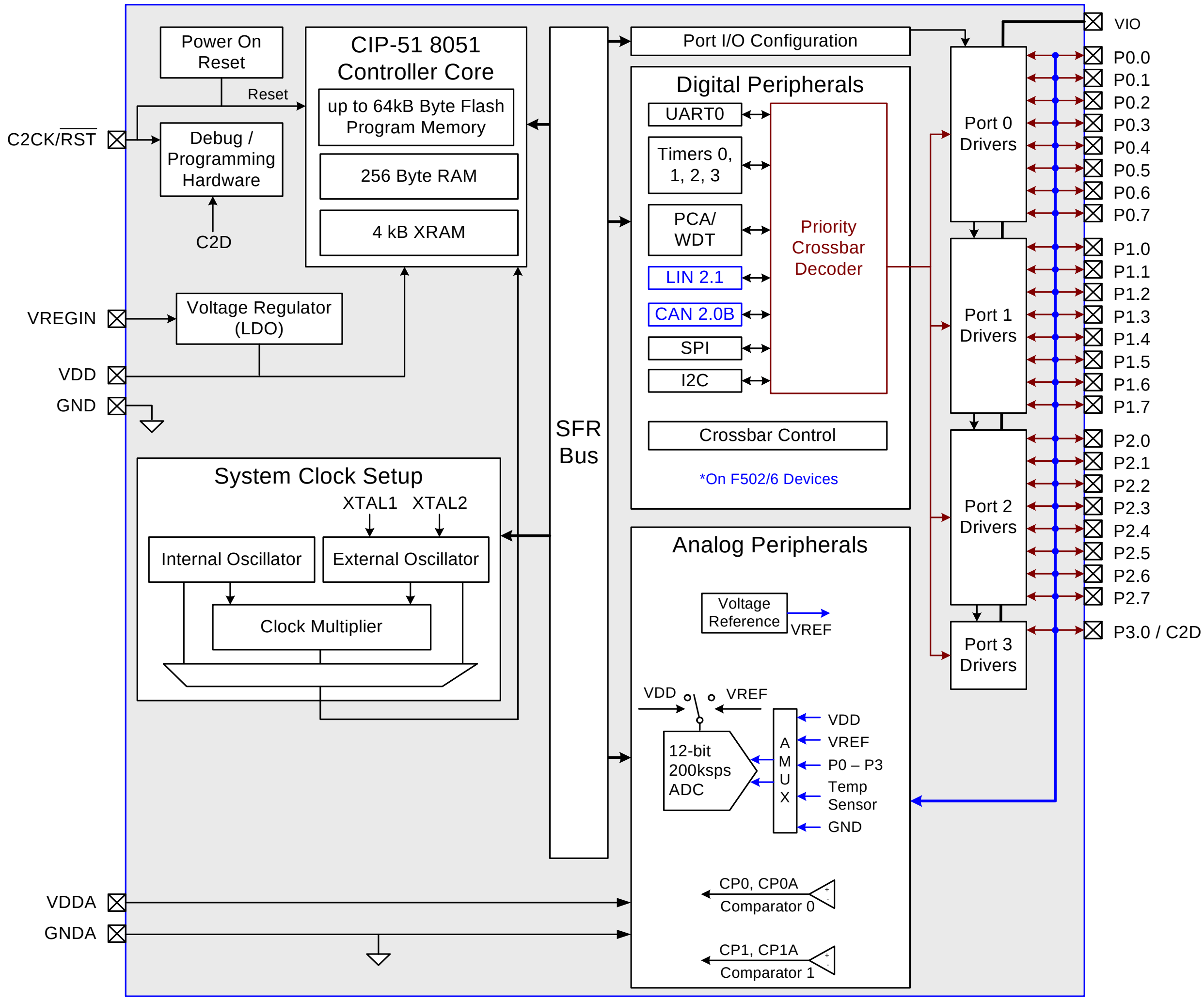


Figure 1.3. C8051F502/3/6/7 Block Diagram

2. Ordering Information

The following features are common to all devices in this family:

- 50 MHz system clock and 50 MIPS throughput (peak)
- 4352 bytes of RAM (256 internal bytes and 4096 XRAM bytes)
- SMBus/I^2C, Enhanced SPI, Enhanced UART
- Four Timers
- Six Programmable Counter Array channels
- Internal 24 MHz oscillator
- Internal Voltage Regulator
- 12-bit, 200 ksps ADC
- Internal Voltage Reference and Temperature Sensor
- Two Analog Comparators

Table 2.1 shows the feature that differentiate the devices in this family.

For example, if ADC0GNH = 0xFC, ADC0GNL = 0x00, and GAINADD = 1, GAIN = 0xFC0 = 4032, and the resulting equation is as follows:

$$\text{GAIN} = \left(\frac{4032}{4096}\right) + 1 \times \left(\frac{1}{64}\right) = 0.984 + 0.016 = 1.0$$

The table below equates values in the ADC0GNH, ADC0GNL, and ADC0GNA registers to the equivalent gain using this equation.

ADC0GNH Value	ADC0GNL Value	GAINADD Value	GAIN Value	Equivalent Gain
0xFC (default)	0x00 (default)	1 (default)	4032 + 64	1.0 (default)
0x7C	0x00	1	1984 + 64	0.5
0xBC	0x00	1	3008 + 64	0.75
0x3C	0x00	1	960 + 64	0.25
0xFF	0xF0	0	4095 + 0	~1.0
0xFF	0xF0	1	4096 + 64	1.016

For any desired gain value, the GAIN registers can be calculated by the following:

$$\text{GAIN} = \left(\text{gain} - \text{GAINADD} \times \left(\frac{1}{64}\right)\right) \times 4096$$

Equation 6.3. Calculating the ADC0GNH and ADC0GNL Values from the Desired Gain

Where:
GAIN is the 12-bit word of ADC0GNH[7:0] and ADC0GNL[7:4]
GAINADD is the value of the GAINADD bit (ADC0GNA.0)
gain is the equivalent gain value from 0 to 1.016

When calculating the value of GAIN to load into the ADC0GNH and ADC0GNL registers, the GAINADD bit can be turned on or off to reach a value closer to the desired gain value.

For example, the initial example in this section requires a gain of 0.44 to convert 5 V full scale to 2.2 V full scale. Using Equation 6.3:

$$\text{GAIN} = \left(0.44 - \text{GAINADD} \times \left(\frac{1}{64}\right)\right) \times 4096$$

If GAINADD is set to 1, this makes the equation:

$$\text{GAIN} = \left(0.44 - 1 \times \left(\frac{1}{64}\right)\right) \times 4096 = 0.424 \times 4096 = 1738 = 0\text{x}06\text{CA}$$

The actual gain from setting GAINADD to 1 and ADC0GNH and ADC0GNL to 0x6CA is 0.4399. A similar gain can be achieved if GAINADD is set to 0 with a different value for ADC0GNH and ADC0GNL.

7. Temperature Sensor

An on-chip temperature sensor is included on the C8051F50x/F51x devices which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC multiplexer channel should be configured to connect to the temperature sensor. The temperature sensor transfer function is shown in Figure 7.1. The output voltage (V_{TEMP}) is the positive ADC input is selected by bits AD0MX[4:0] in register ADC0MX. The TEMPE bit in register REF0CN enables/disables the temperature sensor, as described in SFR Definition 8.1. 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 5.10 for the slope and offset parameters of the temperature sensor.

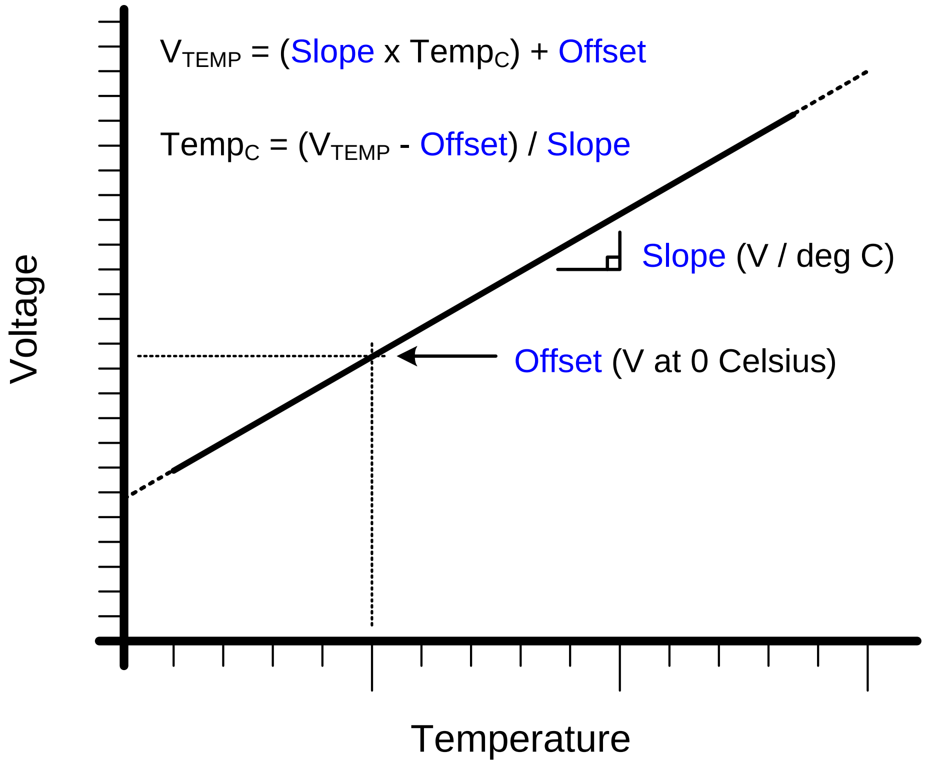


Figure 7.1. Temperature Sensor Transfer Function

SFR Definition 11.1. DPL: Data Pointer Low Byte

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

SFR Address = 0x82; SFR Page = All Pages

Bit	Name	Function
7:0	DPL[7:0]	**Data Pointer Low.** The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory or XRAM.

SFR Definition 11.2. DPH: Data Pointer High Byte

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

SFR Address = 0x83; SFR Page = All Pages

Bit	Name	Function
7:0	DPH[7:0]	**Data Pointer High.** The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed Flash memory or XRAM.

12.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F50x/F51x devices implement 64 kB or 32 kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0xFFFF in 64 kB devices and addresses 0x0000 to 0x7FFF in 32 kB devices. The address 0xFBFF in 64 kB devices and 0x7FFF in 32 kB devices serves as the security lock byte for the device. Addresses above 0xFDFF are reserved in the 64 kB devices.

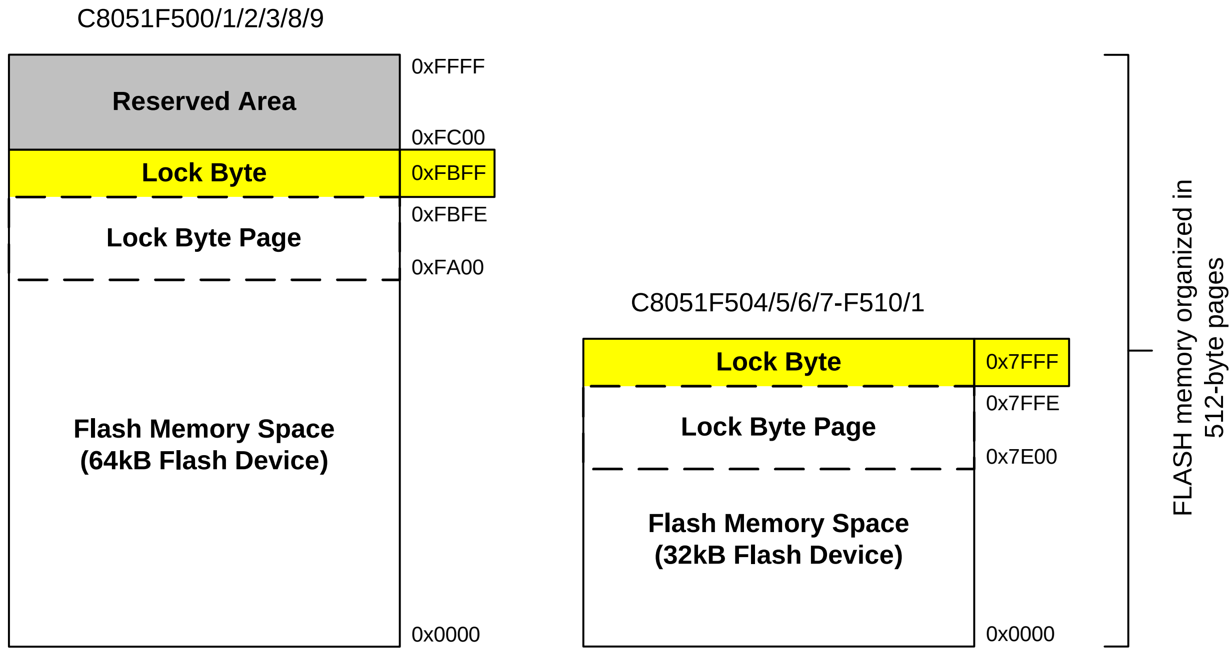


Figure 12.2. Flash Program Memory Map

12.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F50x/F51x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip Flash memory space. MOVC instructions are always used to read Flash memory, while MOVX write instructions are used to erase and write Flash. This Flash access feature provides a mechanism for the C8051F50x/F51x to update program code and use the program memory space for non-volatile data storage. Refer to Section "15. Flash Memory" on page 129 for further details.

12.2. Data Memory

The C8051F50x/F51x devices include 4352 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. The other 4096 bytes of this memory is on-chip "external" memory. The data memory map is shown in Figure 12.1 for reference.

12.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight

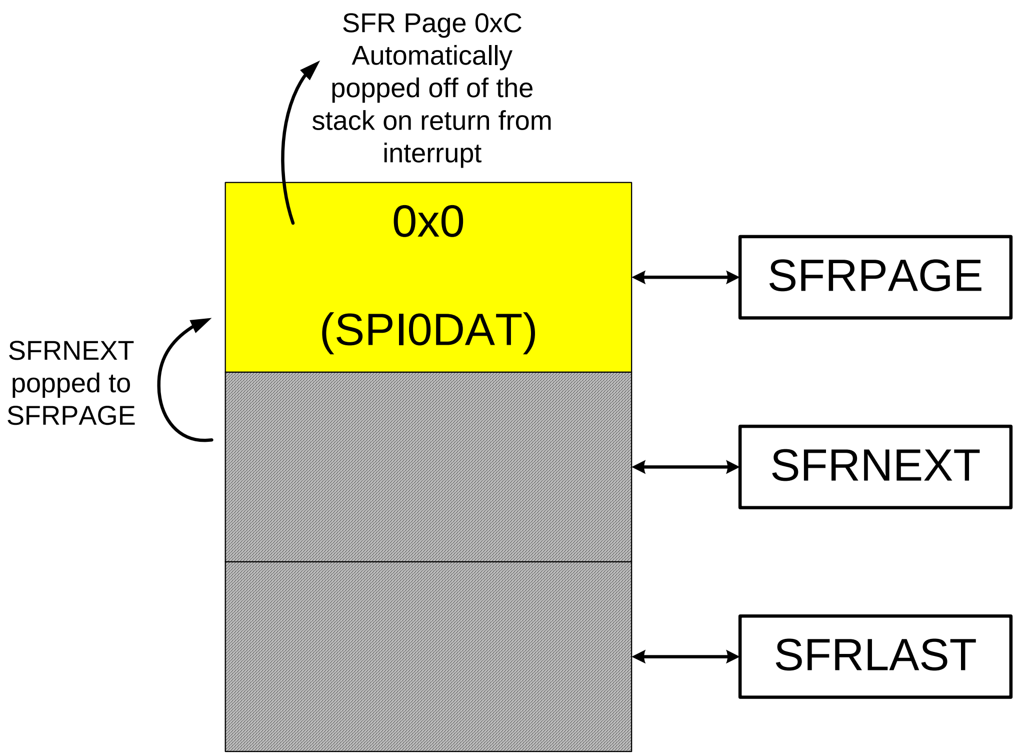


Figure 13.6. SFR Page Stack Upon Return From CAN0 Interrupt

In the example above, all three bytes in the SFR Page Stack are accessible via the SFRPAGE, SFRNEXT, and SFRLAST special function registers. If the stack is altered while servicing an interrupt, it is possible to return to a different SFR Page upon interrupt exit than selected prior to the interrupt call. Direct access to the SFR Page stack can be useful to enable real-time operating systems to control and manage context switching between multiple tasks.

Push operations on the SFR Page Stack only occur on interrupt service, and pop operations only occur on interrupt exit (execution on the RETI instruction). The automatic switching of the SFRPAGE and operation of the SFR Page Stack as described above can be disabled in software by clearing the SFR Automatic Page Enable Bit (SFRPGEN) in the SFR Page Control Register (SFR0CN). See SFR Definition 13.1.

SFR Definition 14.5. EIE2: Extended Interrupt Enable 2

Bit	7	6	5	4	3	2	1	0
Name						EMAT	ECAN0	EREG0
Type	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE7; SFR Page = All Pages

Bit	Name	Function
7:3	Unused	Read = 00000b; Write = Don’t Care.
2	EMAT	**Enable Port Match Interrupt.** This bit sets the masking of the Port Match interrupt. 0: Disable all Port Match interrupts. 1: Enable interrupt requests generated by a Port Match
1	ECAN0	**Enable CAN0 Interrupts.** This bit sets the masking of the CAN0 interrupt. 0: Disable all CAN0 interrupts. 1: Enable interrupt requests generated by CAN0.
0	EREG0	**Enable Voltage Regulator Dropout Interrupt.** This bit sets the masking of the Voltage Regulator Dropout interrupt. 0: Disable the Voltage Regulator Dropout interrupt. 1: Enable the Voltage Regulator Dropout interrupt.

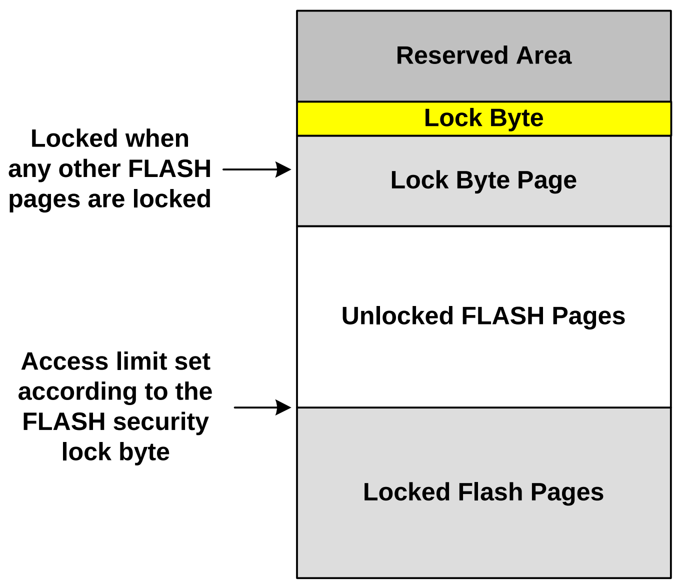
15.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

15.3. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to 1 before software can modify the Flash memory; both PSWE and PSEE must be set to 1 before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock *n* 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where *n* is the ones complement number represented by the Security Lock Byte. **Note that the page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are 1) and locked when any other Flash pages are locked (any bit of the Lock Byte is 0).** See example in Figure 15.1.



Security Lock Byte:	11111101b
1s Complement:	00000010b
Flash pages locked:	3 (First two Flash pages + Lock Byte Page)

Figure 15.1. Flash Program Memory Map

15.4.3. System Clock

1. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
2. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in ”AN201: Writing to Flash from Firmware" available from the Silicon Laboratories web site.

SFR Definition 15.1. PSCTL: Program Store R/W Control

Bit	7	6	5	4	3	2	1	0
Name							PSEE	PSWE
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x8F; SFR Page = 0x00

Bit	Name	Function
7:2	Unused	Read = 000000b, Write = don’t care.
1	PSEE	**Program Store Erase Enable.** Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter. 0: Flash program memory erasure disabled. 1: Flash program memory erasure enabled.
0	PSWE	**Program Store Write Enable.** Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The Flash location should be erased before writing data. 0: Writes to Flash program memory disabled. 1: Writes to Flash program memory enabled; the MOVX write instruction targets Flash memory.

17.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{RST}$ pin is driven low until V_{DD} settles above V_{RST}. A delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 17.2. plots the power-on and V_{DD} monitor reset timing.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000) software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset. The V_{DD} monitor is enabled following a power-on reset.

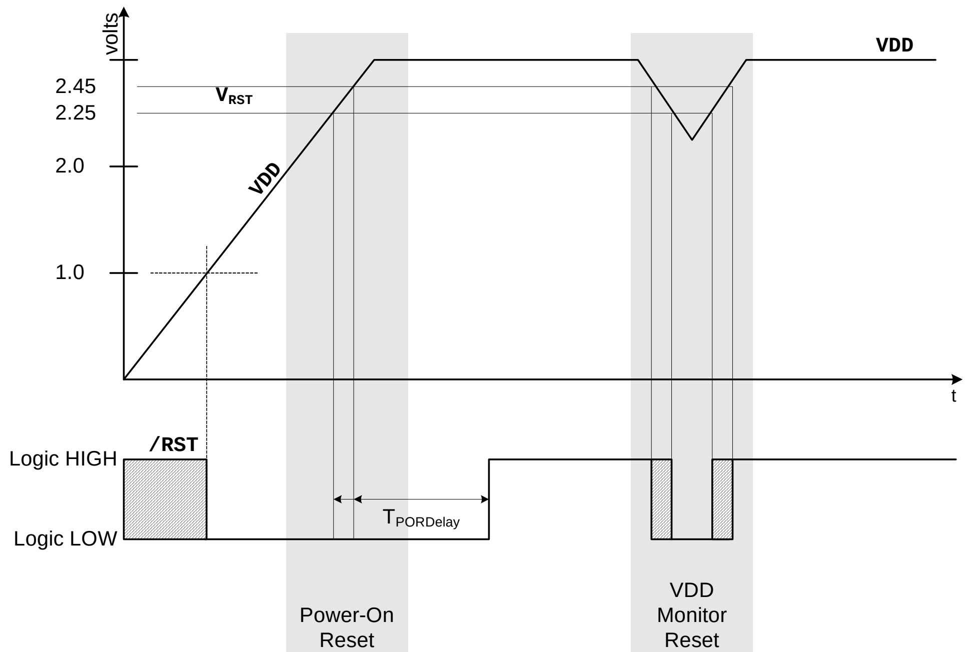


Figure 17.2. Power-On and V_{DD} Monitor Reset Timing

17.2. Power-Fail Reset/V_{DD} Monitor

When a power-down transition or power irregularity causes V_{DD} to drop below V_{RST}, the power supply monitor will drive the $\overline{RST}$ pin low and hold the CIP-51 in a reset state (see Figure 17.2). When V_{DD} returns to a level above V_{RST}, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if V_{DD} dropped below the level required for data retention. If the PORSF flag reads 1, the data may no longer be valid. The V_{DD} monitor is enabled after power-on resets. Its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} monitor is disabled by code and a software reset is performed, the V_{DD} monitor will still be disabled after the reset. **To protect the integrity of Flash contents, the V_{DD} monitor must be enabled to the higher setting (VDMLVL = 1) and selected as a reset source if software contains routines which erase or write Flash memory. If the V_{DD} monitor is not enabled and set to the high level, any erase or write performed on Flash memory will cause a Flash Error device reset.**

SFR Definition 20.17. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF2; SFR Page = 0x0F

Bit	Name	Function
7:0	P1MDIN[7:0]	**Analog Configuration Bits for P1.7–P1.0 (respectively).** Port pins configured for analog mode have their weak pull-up and digital receiver disabled. For analog mode, the pin also needs to be configured for open-drain mode in the P1MDOUT register. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.

SFR Definition 20.18. P1MDOUT: Port 1 Output Mode

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

SFR Address = 0xA5; SFR Page = 0x0F

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

SFR Definition 20.19. P1SKIP: Port 1 Skip

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

SFR Address = 0xD5; SFR Page = 0x0F

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

SFR Definition 20.20. P2: Port 2

Bit	7	6	5	4	3	2	1	0
Name	P2[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xA0; SFR Page = All Pages; Bit-Addressable

Bit	Name	Description	Write	Read
7:0	P2[7:0]	**Port 2Data.** Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P2.n Port pin is logic LOW. 1: P2.n Port pin is logic HIGH.

SFR Definition 21.3. LIN0CF: LIN0 Control Mode Register

Bit	7	6	5	4	3	2	1	0
Name	LINEN	MODE	ABAUD					
Type	R/W	R/W	R/W	R	R	R	R	R
Reset	0	1	1	0	0	0	0	0

SFR Address = 0xC9; SFR Page = 0x0F

Bit	Name	Function
7	LINEN	**LIN Interface Enable Bit.** 0: LIN0 is disabled. 1: LIN0 is enabled.
6	MODE	**LIN Mode Selection Bit.** 0: LIN0 operates in slave mode. 1: LIN0 operates in master mode.
5	ABAUD	**LIN Mode Automatic Baud Rate Selection.** This bit only has an effect when the MODE bit is configured for slave mode. 0: Manual baud rate selection is enabled. 1: Automatic baud rate selection is enabled.
4:0	Unused	Read = 00000b; Write = Don’t Care

SFR Definition 22.1. CAN0CFG: CAN Clock Configuration

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

SFR Address = 0x92; SFR Page = 0x0C

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = Don’t Care.
1:0	SYSDIV[1:0]	**CAN System Clock Divider Bits.** The CAN controller clock is derived from the CIP-51 system clock. The CAN controller clock must be less than or equal to 25 MHz. 00: CAN controller clock = System Clock/1. 01: CAN controller clock = System Clock/2. 10: CAN controller clock = System Clock/4. 11: CAN controller clock = System Clock/8.

Table 23.4. SMBus Status Decoding

Mode	Values Read: Status Vector	Values Read: ACKRQ	Values Read: ARBLOST	Values Read: ACK	Current SMbus State	Typical Response Options	Values to Write: STA	Values to Write: STO	Values to Write: ACK	Next Status Vector Expected
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

25.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 25.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 25.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

25.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

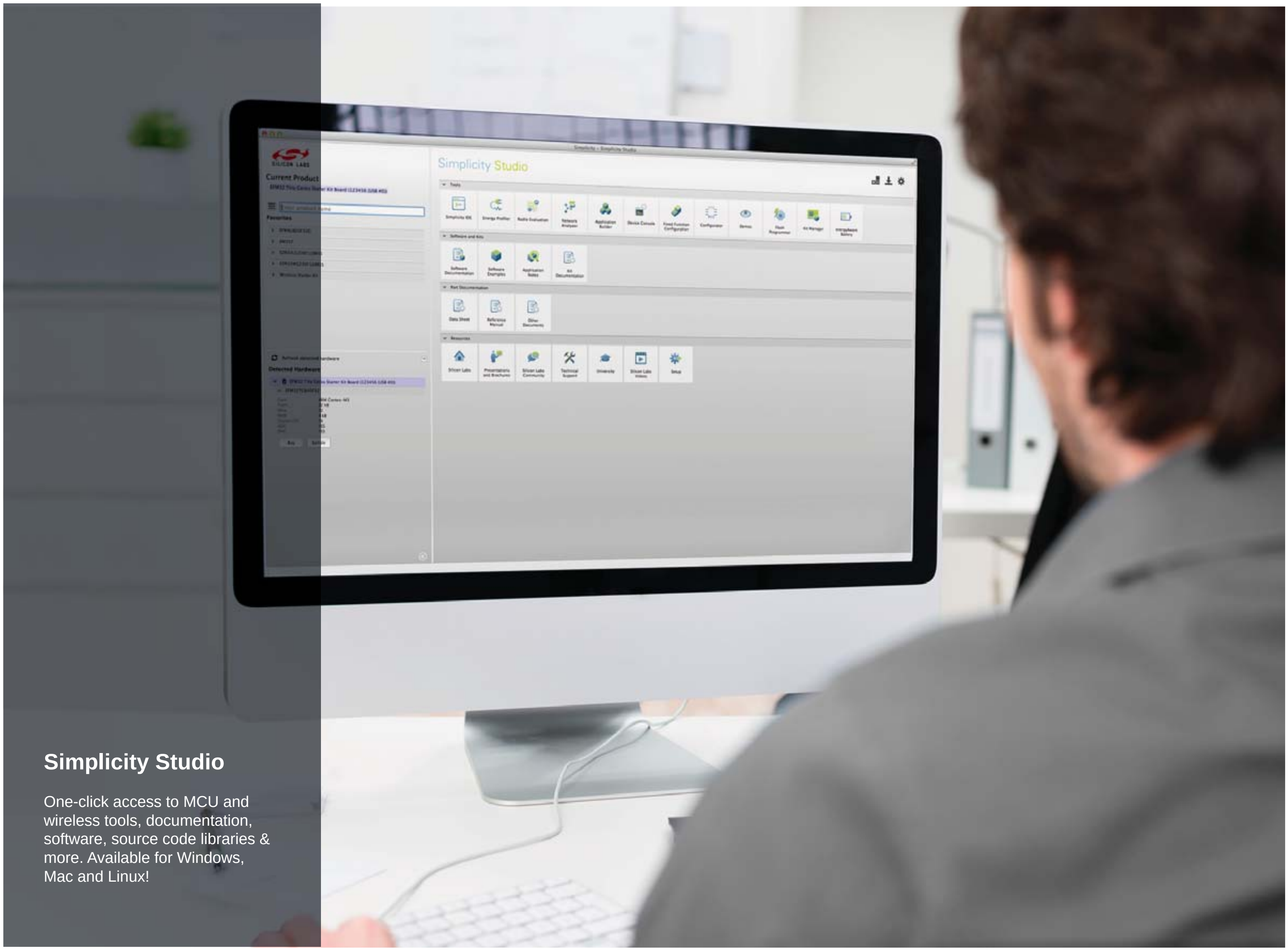
1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

SFR Definition 27.3. PCA0PWM: PCA PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL	ECOV	COVF				CLSEL[1:0]	
Type	R/W	R/W	R/W	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD9; SFR Page = 0x0F

Bit	Name	Function
7	ARSEL	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9, 10, and 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function. 0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn. 1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn.
6	ECOV	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt. 0: COVF will not generate PCA interrupts. 1: A PCA interrupt will be generated when COVF is set.
5	COVF	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th, 9th, 10th, or 11th bit of the main PCA counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or software, but must be cleared by software. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:2	Unused	Read = 000b; Write = Don’t care.
1:0	CLSEL[1:0]	**Cycle Length Select.** When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle, between 8, 9, 10, or 11 bits. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to16-bit PWM mode. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.



IoT Portfolio	SW/HW	Quality	Support and Community
www.silabs.com/IoT	*www.silabs.com/simplicity*	*www.silabs.com/quality*	*community.silabs.com*

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

http://www.silabs.com