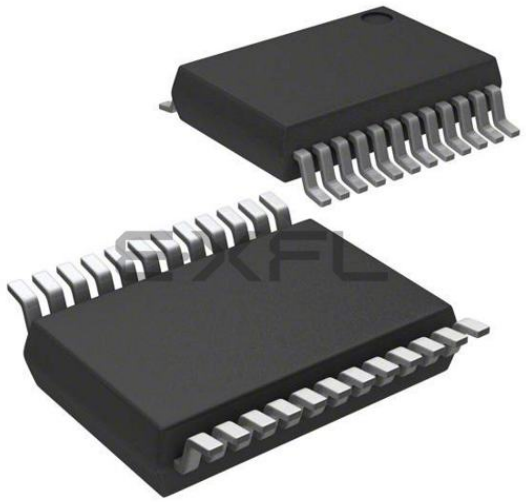


Welcome to E-XFL.COM

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	CIP-51 8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, Cap Sense, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	24-QSOP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f997-gur

6.2. IREF0 Specifications ... 92
7. Comparator ... 93
7.1. Comparator Inputs ... 93
7.2. Comparator Outputs ... 94
7.3. Comparator Response Time ... 94
7.4. Comparator Hysteresis ... 94
7.5. Comparator Register Descriptions ... 95
7.6. Comparator0 Analog Multiplexer ... 98
8. Capacitive Sense (CS0) ... 100
8.1. Configuring Port Pins as Capacitive Sense Inputs ... 101
8.2. Initializing the Capacitive Sensing Peripheral ... 101
8.3. Capacitive Sense Start-Of-Conversion Sources ... 101
8.4. CS0 Multiple Channel Enable ... 102
8.5. CS0 Gain Adjustment ... 102
8.6. Wake from Suspend ... 102
8.7. Using CS0 in Applications that Utilize Sleep Mode ... 102
8.8. Automatic Scanning (Method 1—CS0SMEN = 0) ... 103
8.9. Automatic Scanning (Method 2—CS0SMEN = 1) ... 104
8.10. CS0 Comparator ... 104
8.11. CS0 Conversion Accumulator ... 105
8.12. CS0 Pin Monitor ... 106
8.13. Adjusting CS0 For Special Situations ... 106
8.14. Capacitive Sense Multiplexer ... 117
9. CIP-51 Microcontroller ... 119
9.1. Performance ... 119
9.2. Programming and Debugging Support ... 120
9.3. Instruction Set ... 120
9.3.1. Instruction and CPU Timing ... 120
9.4. CIP-51 Register Descriptions ... 125
10. Memory Organization ... 128
10.1. Program Memory ... 129
10.1.1. MOVX Instruction and Program Memory ... 129
10.2. Data Memory ... 129
10.2.1. Internal RAM ... 129
10.2.2. External RAM ... 130
11. On-Chip XRAM ... 131
11.1. Accessing XRAM ... 131
11.1.1. 16-Bit MOVX Example ... 131
11.1.2. 8-Bit MOVX Example ... 131
12. Special Function Registers ... 132
12.1. SFR Paging ... 133
13. Interrupt Handler ... 138
13.1. Enabling Interrupt Sources ... 138
13.2. MCU Interrupt Sources and Vectors ... 138
13.3. Interrupt Priorities ... 139

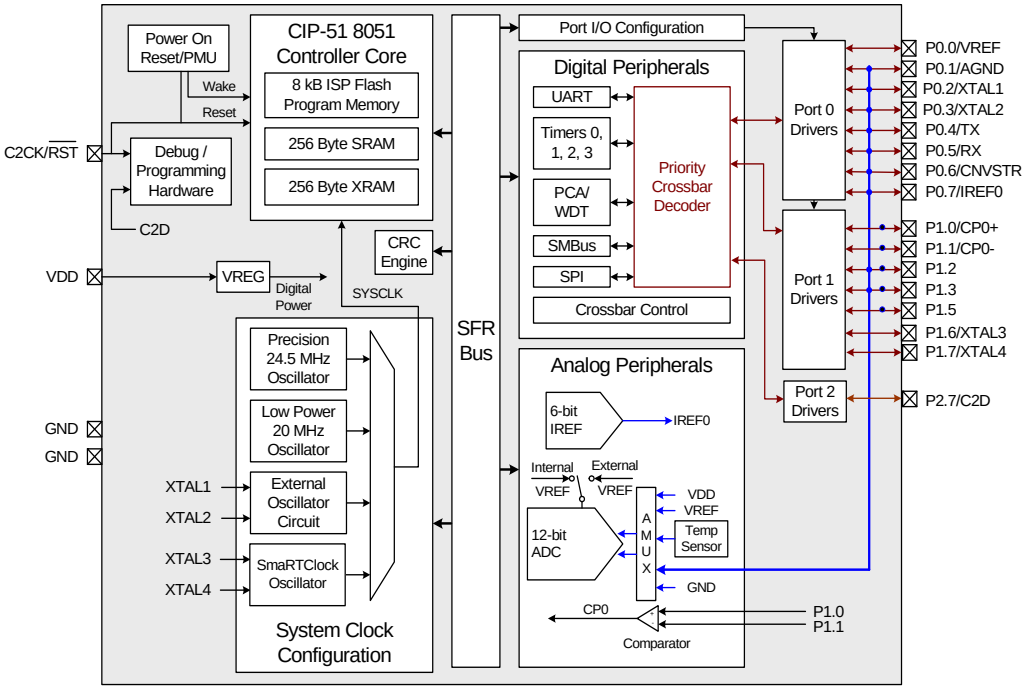


Figure 1.1. C8051F980 Block Diagram

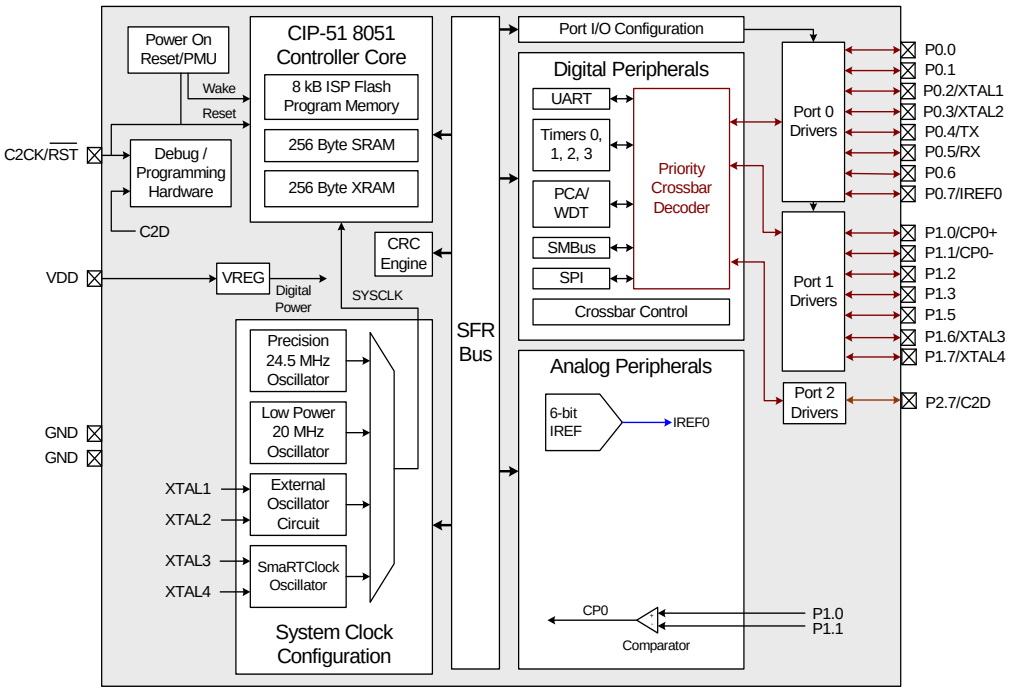


Figure 1.2. C8051F981 Block Diagram

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

The ADC0 on C8051F980/6 and C8051F990/6 devices is a 300 ksps, 10-bit or 75 ksps, 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. C8051F982 and C8051F988 devices only support the 10-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 83. The voltage reference for the ADC is selected as described in “5.9. Voltage and Ground Reference Options” on page 88.

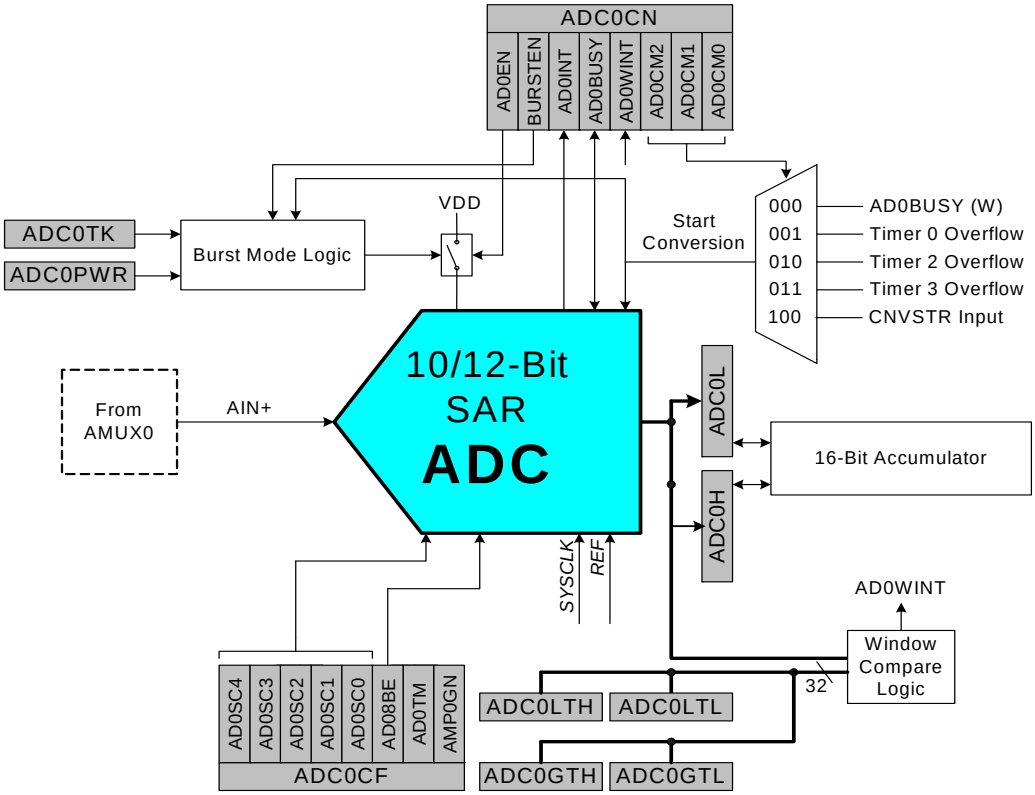


Figure 5.1. ADC0 Functional Block Diagram

5.2. Modes of Operation

ADC0 has a maximum conversion speed of 300 ksps in 10-bit mode. The ADC0 conversion clock (SARCLK) is a divided version of the system clock when burst mode is disabled (BURSTEN = 0), or a divided version of the low power oscillator when burst mode is enabled (BURSEN = 1). The clock divide value is determined by the AD0SC bits in the ADC0CF register.

5.2.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2–0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a 1 to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 3 overflow
5. A rising edge on the CNVSTR input signal (pin P0.6)

Writing a 1 to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. When Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See “25. Timers” on page 278 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as Port pin P0.6. When the CNVSTR input is used as the ADC0 conversion source, Port pin P0.6 should be skipped by the Digital Crossbar. To configure the Crossbar to skip P0.6, set to 1 Bit 6 in register P0SKIP. See “21. Port Input/Output” on page 215 for details on Port I/O configuration.

Table 5.1. Representative Conversion Times and Energy Consumption for the SAR ADC with 1.65 V High-Speed VREF

	Normal Power Mode			Low Power Mode		
	8 bit	**10 bit**	**12 bit**	**8 bit**	**10 bit**	**12 bit**
Highest nominal SAR clock frequency	8.17 MHz (24.5/3)	8.17 MHz (24.5/3)	6.67 MHz (20.0/3)	4.08 MHz (24.5/6)	4.08 MHz (24.5/6)	4.00 MHz (20.0/5)
Total number of conversion clocks required	11	13	52 (13 x 4)	11	13	52 (13*4)
Total tracking time (min)	1.5 µs	1.5 µs	4.8 µs (1.5+3 x 1.1)	1.5 µs	1.5 µs	4.8 µs (1.5+3 x 1.1)
Total time for one conversion	2.85 µs	3.09 µs	12.6 µs	4.19 µs	4.68 µs	17.8 µs
ADC Throughput	351 ksps	323 ksps	79 ksps	238 ksps	214 ksps	56 ksps
Energy per conversion	8.2 nJ	8.9 nJ	36.5 nJ	6.5 nJ	7.3 nJ	27.7 nJ

Note: This table assumes that the 24.5 MHz precision oscillator is used for 8- and 10-bit modes, and the 20 MHz low power oscillator is used for 12-bit mode. The values in the table assume that the oscillators run at their nominal frequencies. The maximum SAR clock values given in Table 4.10 allow for maximum oscillation frequencies of 25.0 MHz and 22 MHz for the precision and low-power oscillators, respectively, when using the given SAR clock divider values. Energy calculations are for the ADC subsystem only and do not include CPU current. 12-bit mode is only available on C8051F980/6 and C8051F990/6 devices.

SFR Definition 8.15. CS0MX: Capacitive Sense Mux Channel Select

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	CS0MX[3:0]			
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 = 0x0; SFR Address = 0xAB

Bit	Name	Description	
7:5	Reserved	Read = 0000b; Write = 0000b.	
4:0	CS0MX[4:0]	**CS0 Mux Channel Select.** Selects one of the 14 input channels for Capacitive Sense conversion.	
		Value	**Channel**
		0000	P0.0
		0001	P0.1
		0010	P0.2
		0011	P0.3
		0100	P0.4
		0101	P0.5
		0110	P0.6
		0111	P0.7
		1000	P1.0
		1001	P1.1
		1010	P1.2
		1011	P1.3
		1100	P1.4 (24-pin packages only)
		1101	P1.5
		1110	Reserved
		1111	Reserved

Table 9.1. 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

SFR Definition 14.3. 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 Page =All; SFR Address = 0x8F

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.

20.2.5. Automatic Gain Control (Crystal Mode Only) and SmaRTClock Bias Doubling

Automatic Gain Control allows the SmaRTClock oscillator to trim the oscillation amplitude of a crystal in order to achieve the lowest possible power consumption. Automatic Gain Control automatically detects when the oscillation amplitude has reached a point where it safe to reduce the drive current, therefore, it may be enabled during crystal startup. It is recommended to enable Automatic Gain Control in most systems which use the SmaRTClock oscillator in Crystal Mode. The following are recommended crystal specifications and operating conditions when Automatic Gain Control is enabled:

- ESR < 50 kΩ
- Load Capacitance < 10 pF
- Supply Voltage < 3.0 V
- Temperature > –20 °C

When using Automatic Gain Control, it is recommended to perform an oscillation robustness test to ensure that the chosen crystal will oscillate under the worst case condition to which the system will be exposed. The worst case condition that should result in the least robust oscillation is at the following system conditions: lowest temperature, highest supply voltage, highest ESR, highest load capacitance, and lowest bias current (AGC enabled, Bias Double Disabled).

To perform the oscillation robustness test, the SmaRTClock oscillator should be enabled and selected as the system clock source. Next, the SYSCLK signal should be routed to a port pin configured as a push-pull digital output. The positive duty cycle of the output clock can be used as an indicator of oscillation robustness. As shown in Figure 20.2, duty cycles less than 55% indicate a robust oscillation. As the duty cycle approaches 60%, oscillation becomes less reliable and the risk of clock failure increases. Increasing the bias current (by disabling AGC) will always improve oscillation robustness and will reduce the output clock's duty cycle. This test should be performed at the worst case system conditions, as results at very low temperatures or high supply voltage will vary from results taken at room temperature or low supply voltage.

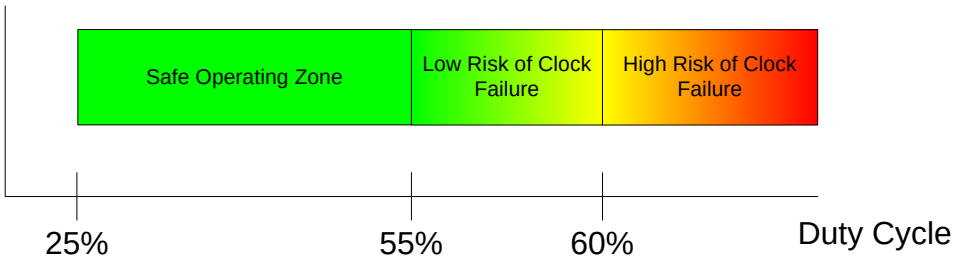


Figure 20.2. Interpreting Oscillation Robustness (Duty Cycle) Test Results

As an alternative to performing the oscillation robustness test, Automatic Gain Control may be disabled at the cost of increased power consumption (approximately 200 nA). Disabling Automatic Gain Control will provide the crystal oscillator with higher immunity against external factors which may lead to clock failure. Automatic Gain Control must be disabled if using the SmaRTClock oscillator in self-oscillate mode.

Table 20.3 shows a summary of the oscillator bias settings. The SmaRTClock Bias Doubling feature allows the self-oscillation frequency to be increased (almost doubled) and allows a higher crystal drive strength in crystal mode. High crystal drive strength is recommended when the crystal is exposed to poor environmental conditions such as excessive moisture. SmaRTClock Bias Doubling is enabled by setting BIASX2 (RTC0XCN.5) to 1.

Internal Register Definition 20.4. RTC0CN: SmaRTClock Control

Bit	7	6	5	4	3	2	1	0
Name	RTC0EN	MCLKEN	OSCFAIL	RTC0TR	RTC0AEN	ALRM	RTC0SET	RTC0CAP
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	Varies	0	0	0	0	0

SmaRTClock Address = 0x04

Bit	Name	Function		
7	RTC0EN	**SmaRTClock Enable.** Enables/disables the SmaRTClock oscillator and associated bias currents. 0: SmaRTClock oscillator disabled. 1: SmaRTClock oscillator enabled.		
6	MCLKEN	**Missing SmaRTClock Detector Enable.** Enables/disables the missing SmaRTClock detector. 0: Missing SmaRTClock detector disabled. 1: Missing SmaRTClock detector enabled.		
5	OSCFAIL	**SmaRTClock Oscillator Fail Event Flag.** Set by hardware when a missing SmaRTClock detector timeout occurs. Must be cleared by software. The value of this bit is not defined when the SmaRTClock oscillator is disabled.		
4	RTC0TR	**SmaRTClock Timer Run Control.** Controls if the SmaRTClock timer is running or stopped (holds current value). 0: SmaRTClock timer is stopped. 1: SmaRTClock timer is running.		
3	RTC0AEN	**SmaRTClock Alarm Enable.** Enables/disables the SmaRTClock alarm function. Also clears the ALRM flag. 0: SmaRTClock alarm disabled. 1: SmaRTClock alarm enabled.		
2	ALRM	**SmaRTClock Alarm Event Flag and Auto Reset Enable.** Reads return the state of the alarm event flag. Writes enable/disable the Auto Reset function.	Read: 0: SmaRTClock alarm event flag is de-asserted. 1: SmaRTClock alarm event flag is asserted.	Write: 0: Disable Auto Reset. 1: Enable Auto Reset.
1	RTC0SET	**SmaRTClock Timer Set.** Writing 1 initiates a SmaRTClock timer set operation. This bit is cleared to 0 by hardware to indicate that the timer set operation is complete.		
0	RTC0CAP	**SmaRTClock Timer Capture.** Writing 1 initiates a SmaRTClock timer capture operation. This bit is cleared to 0 by hardware to indicate that the timer capture operation is complete.		
Note: The ALRM flag will remain asserted for a maximum of one SmaRTClock cycle. See Section “Power Management” on page 162 for information on how to capture a SmaRTClock Alarm event using a flag which is not automatically cleared by hardware.				

22. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compatible with the System Management Bus Specification, version 1.1, and compatible with the I^2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/20th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. The SMBus peripheral can be fully driven by software (i.e. software accepts/rejects slave addresses, and generates ACKs), or hardware slave address recognition and automatic ACK generation can be enabled to minimize software overhead. A block diagram of the SMBus peripheral and the associated SFRs is shown in Figure 22.1.

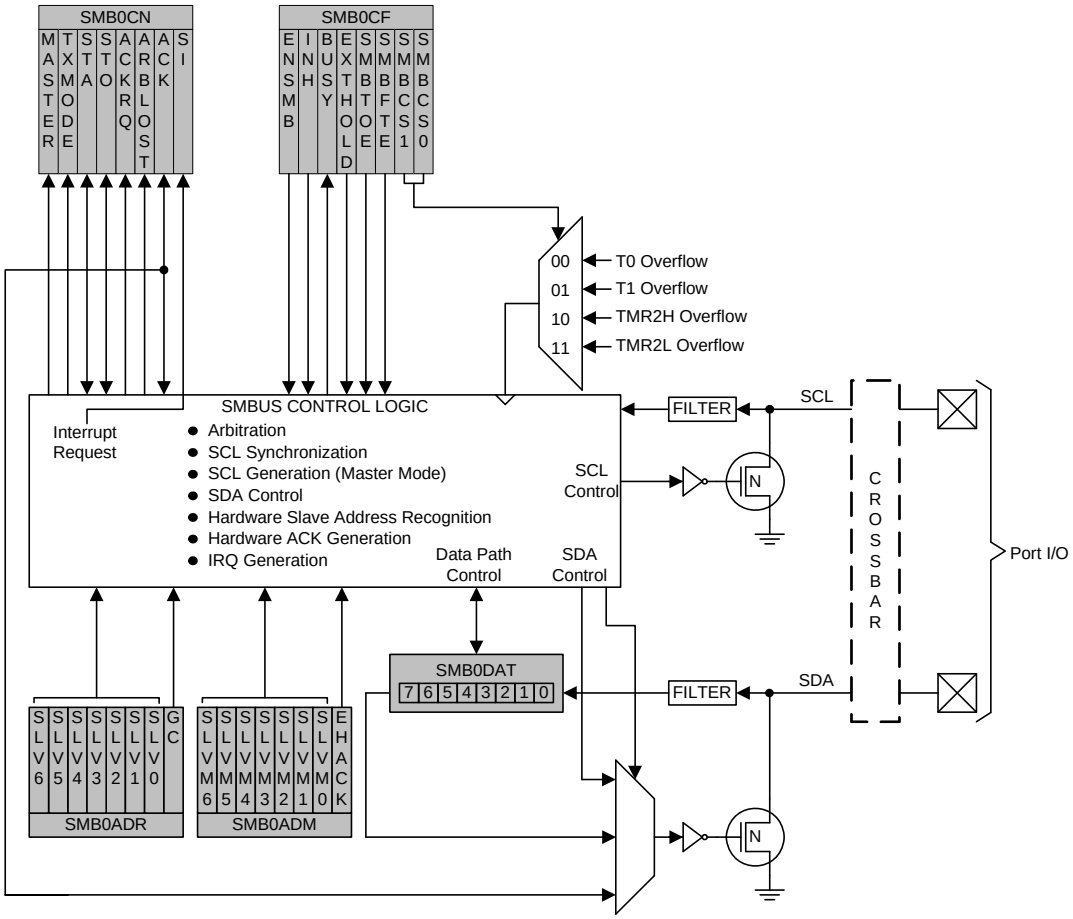


Figure 22.1. SMBus Block Diagram

SFR Definition 22.3. SMB0ADR: SMBus Slave Address

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

SFR Page = 0x0; SFR Address = 0xF4

Bit	Name	Function
7:1	SLV[6:0]	**SMBus Hardware Slave Address.** Defines the SMBus Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM[6:0] are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC	**General Call Address Enable.** When hardware address recognition is enabled (EHACK = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

SFR Definition 22.4. SMB0ADM: SMBus Slave Address Mask

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

SFR Page = 0x0; SFR Address = 0xF5

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

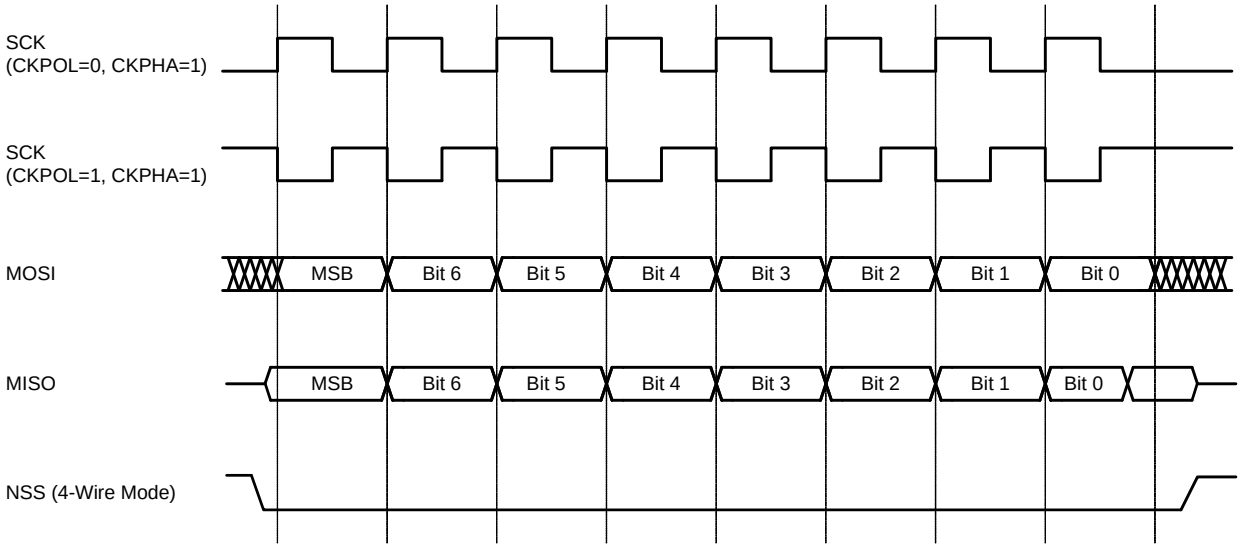


Figure 24.7. Slave Mode Data/Clock Timing (CKPHA = 1)

24.6. 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.

Table 24.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing (See Figure 24.8 and Figure 24.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing (See Figure 24.10 and Figure 24.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns
Note: T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).				

SFR Definition 25.3. TMOD: Timer Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1:0]		GATE0	C/T0	T0M[1:0]	
Type	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 = 0x89

Bit	Name	Function
7	GATE1	**Timer 1 Gate Control.** 0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{\text{INT1}}$ logic level. 1: Timer 1 enabled only when TR1 = 1 AND $\overline{\text{INT1}}$ is active as defined by bit IN1PL in register IT01CF (see SFR Definition 13.7).
6	C/T1	**Counter/Timer 1 Select.** 0: Timer: Timer 1 incremented by clock defined by T1M bit in register CKCON. 1: Counter: Timer 1 incremented by high-to-low transitions on external pin (T1).
5:4	T1M[1:0]	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	GATE0	**Timer 0 Gate Control.** 0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{\text{INT0}}$ logic level. 1: Timer 0 enabled only when TR0 = 1 AND $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF (see SFR Definition 13.7).
2	C/T0	**Counter/Timer 0 Select.** 0: Timer: Timer 0 incremented by clock defined by T0M bit in register CKCON. 1: Counter: Timer 0 incremented by high-to-low transitions on external pin (T0).
1:0	T0M[1:0]	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

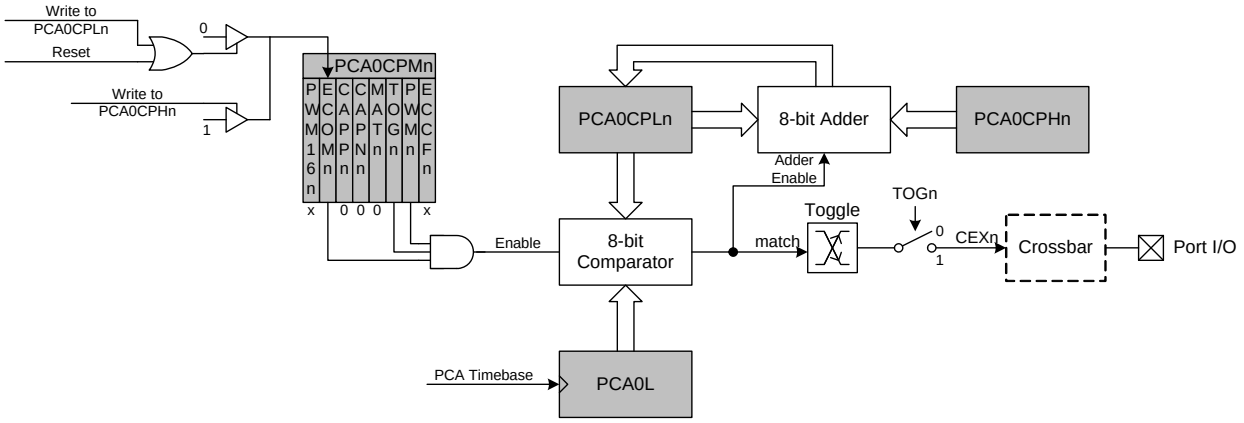


Figure 26.7. PCA Frequency Output Mode

26.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

27. C2 Interface

C8051F99x-C8051F98x devices include an on-chip Silicon Labs 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

27.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 27.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

Bit	Name	Function	
7:0	C2ADD[7:0]	**C2 Address.** The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.	
		Address	**Description**
		0x00	Selects the Device ID register for Data Read instructions
		0x01	Selects the Revision ID register for Data Read instructions
		0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
		0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions

27.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely “borrow” the C2CK ($\overline{RST}$) and C2D pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 27.1.

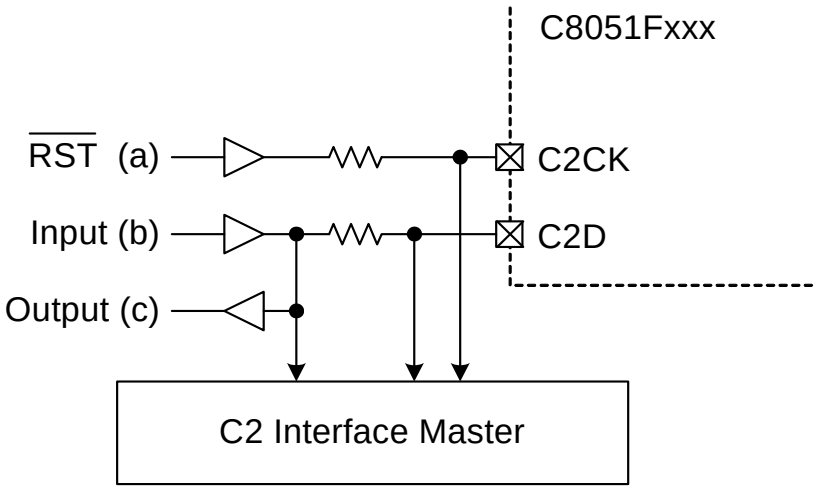


Figure 27.1. Typical C2 Pin Sharing

The configuration in Figure 27.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The $\overline{RST}$ pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

DOCUMENT CHANGE LIST

Revision 0.3 to Revision 0.4

- QFN-20 package and landing diagram updated.

Revision 0.4 to Revision 1.0

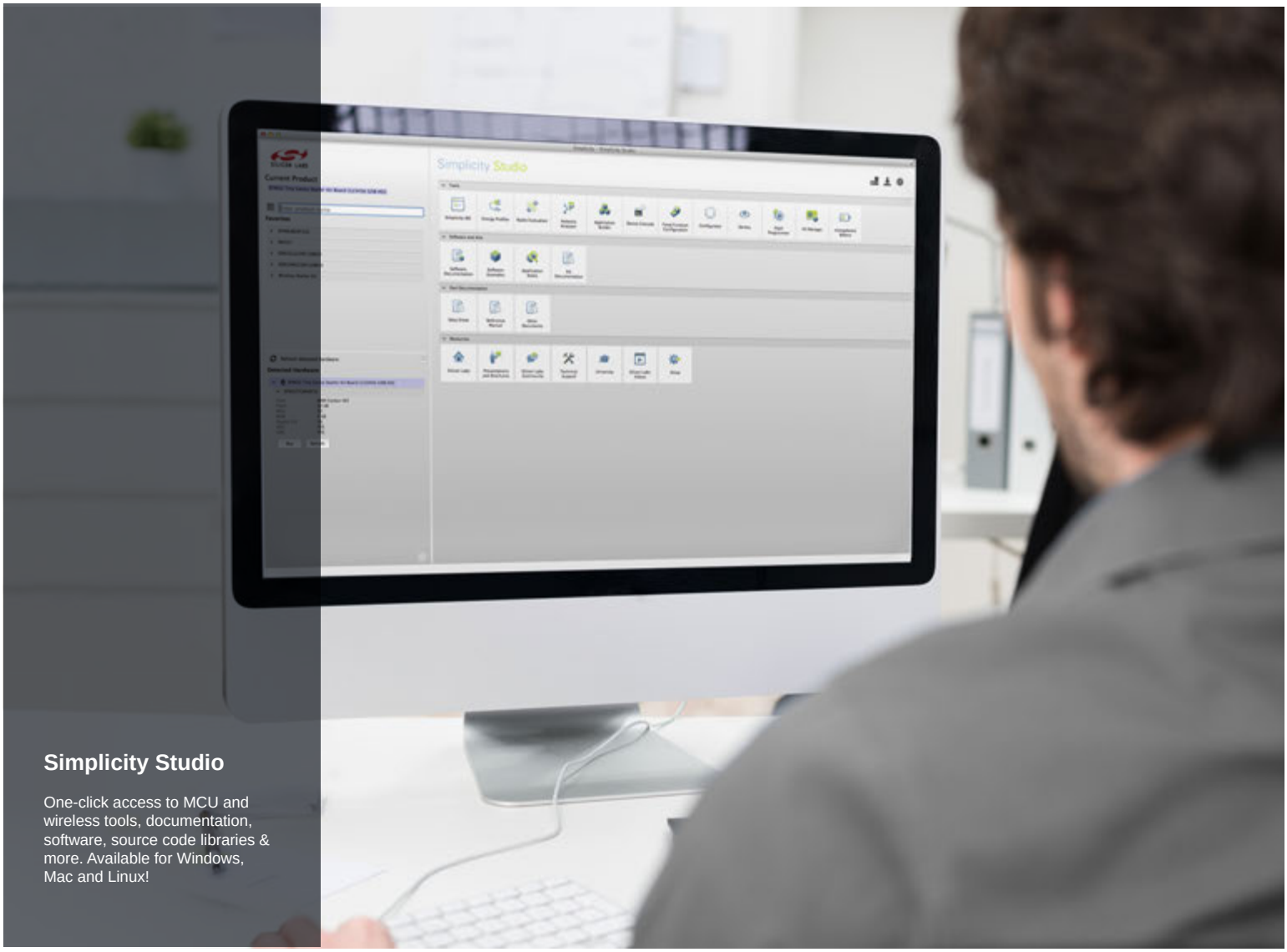
- IREF0CF register description updated.
- Updated ADC0 Chapter Text.
- Corrected an error in the Product Selector Guide.
- Updated SmaRTClock chapter to indicate how the Alarm value should be set when using Auto Reset and the LFO.
- Updated electrical specifications to fill TBDs and updated power specifications based on Rev B characterization data.
- Added a note to the OSCICL register description.
- Added a note to the CRC0CN register description.
- Updated equation in the CRC0CNT register description.
- Updated Power On Reset description.

Revision 1.0 to Revision 1.1

- Removed references to AN338.

Revision 1.1 to Revision 1.2

- Removed QuickSense references.
- Updated part numbers to Revision C in “Ordering Information” on page 31 and added Figure 3.4, Figure 3.5, and Figure 3.6 to identify the silicon revision.
- Updated REVID register (SFR Definition 14.2) and REVID C2 register (C2 Register Definition 27.3) with the 0x02 value for Revision C.
- Updated Figure “7.3 CP0 Multiplexer Block Diagram” on page 98 to remove the bar over the CPnOUT signals.
- Updated the “Reset Sources” on page 181 chapter to reflect the correct state of the $\overline{RST}$ pin during power-on reset.
- Updated Figure “1.14 Port I/O Functional Block Diagram” on page 26 and Figure “21.1 Port I/O Functional Block Diagram” on page 215 to mention P1.4 is not available on 20-pin devices.
- Removed references to the EMI0CN register, which does not exist.
- Updated Figure “8.2 Auto-Scan Example” on page 103 to refer to the correct pins.
- Updated POR Monitor Threshold (V_{POR}) Brownout Condition (VDD Falling) specification minimum, typical, and maximum values.
- Updated the reset value of the CLKSEL register (SFR Definition 19.1).
- Updated description of WEAKPUD in SFR Definition 21.3.
- Corrected SFR addresses for P0DRV (SFR Definition 21.12), P1DRV (SFR Definition 21.17), P2DRV (SFR Definition 21.20), PMU0MD (SFR Definition 15.3), FLSCL (SFR Definition 14.5), REF0CN (SFR Definition 5.15), CS0SCAN0 (SFR Definition 8.5), and CS0SCAN1 (SFR Definition 8.6).
- Replaced all instances of V_{BAT} with V_{DD}.
- Added a note to “11.1. Accessing XRAM” , “15.5. Sleep Mode” , and “18. Reset Sources” regarding an issue with the first address of XRAM.
- Added a note to “15.5. Sleep Mode” and “19. Clocking Sources” regarding using the internal low power



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