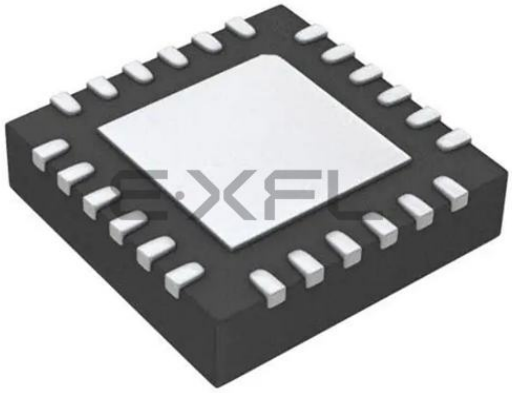

Welcome to E-XFL.COM

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	50MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	21
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K 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-WFQFN Exposed Pad
Supplier Device Package	24-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f371-a-gm

7. Electrical Characteristics

7.1. Absolute Maximum Specifications

Table 7.1. Absolute Maximum Ratings

Parameter	Test Condition	Min	Typ	Max	Unit
Ambient Temperature under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on any Port I/O Pin or $\overline{\text{RST}}$ with respect to GND		–0.3	—	V_{DD} + 0.3	V
Voltage on V_{DD} with Respect to GND		–0.3	—	4.2	V
Maximum Total Current through V_{DD} or GND		—	—	100	mA
Maximum Output Current Sunk by $\overline{\text{RST}}$ or any Port Pin		—	—	100	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

7.3. Typical Performance Curves

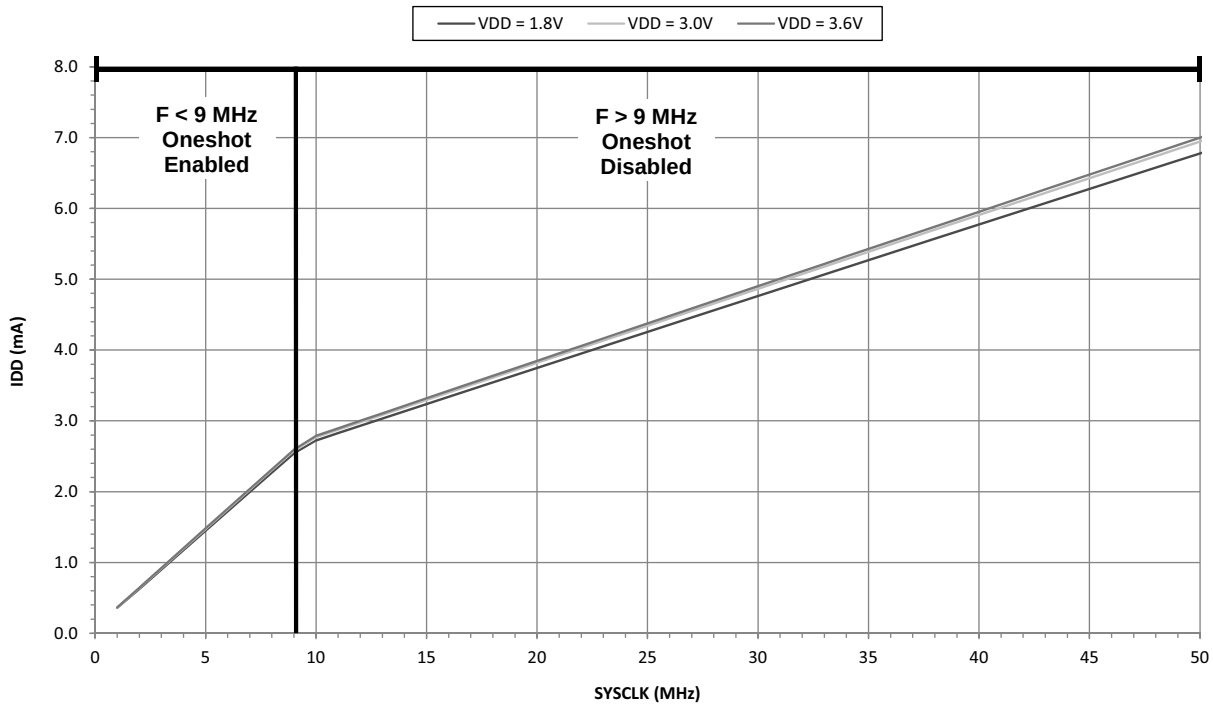

Figure 7.1. Normal Mode Digital Supply Current vs. Frequency

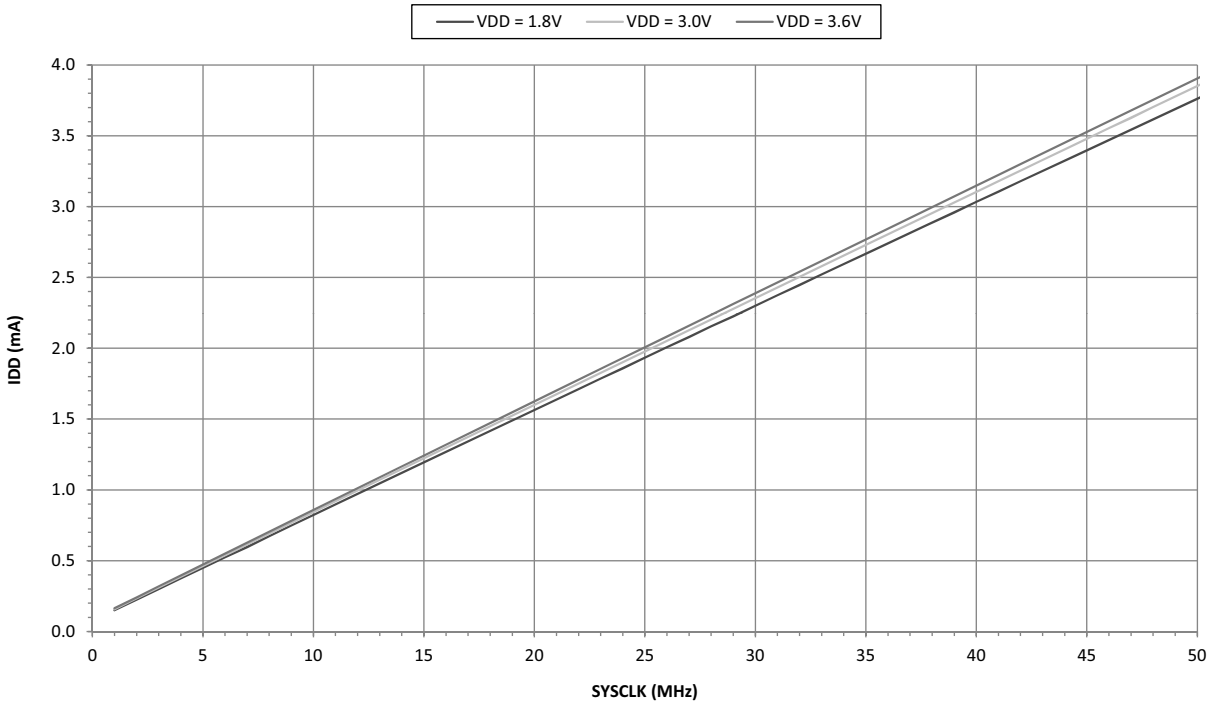

Figure 7.2. Idle Mode Digital Supply Current vs. Frequency

The Comparator response time may be configured in software via the CPT0MD register (see SFR Definition 14.2). Selecting a longer response time reduces the Comparator supply current.

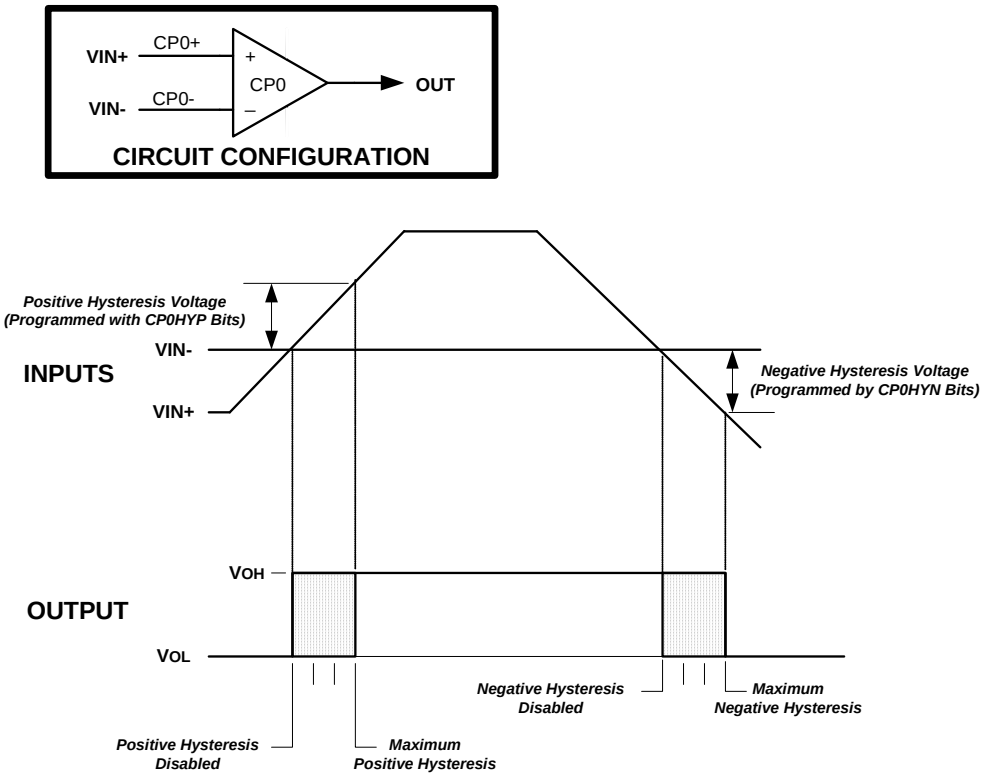

Figure 14.2. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPT0CN. The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3–0 in the Comparator Control Register CPT0CN (shown in SFR Definition 14.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 14.2, settings of 20, 10, or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “20.1. MCU Interrupt Sources and Vectors” on page 118). The CP0FIF flag is set to logic 1 upon a Comparator falling-edge occurrence, and the CP0RIF flag is set to logic 1 upon the Comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The Comparator rising-edge interrupt mask is enabled by setting CP0RIE to a logic 1. The Comparator0 falling-edge interrupt mask is enabled by setting CP0FIE to a logic 1.

The output state of the Comparator can be obtained at any time by reading the CP0OUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered 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.

Table 15.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2

SFR Definition 19.2. SFRPGCN: SFR Page Control

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

SFR Address = 0xCF; SFR Page = All Pages

Bit	Name	Function
7	Reserved	Must Write 0b
6:4	SFRPGIDX[2:0]	**SFR Page Stack Index.** This field can be used to access the SFRPAGE values stored in the SFR page stack. It selects which level of the stack is accessible when reading the SFRSTACK register. 000: SFRSTACK contains the value of SFRPAGE, the first/top byte of the SFR page stack 001: SFRSTACK contains the value of the second byte of the SFR page stack 010: SFRSTACK contains the value of the third byte of the SFR page stack 011: SFRSTACK contains the value of the forth byte of the SFR page stack 100: SFRSTACK contains the value of the fifth/bottom byte of the SFR page stack 101: Invalid index 11x: Invalid index
3:1	Reserved	Must Write 000b
0	SFRPGEN	**SFR Automatic Page Control Enable.** This bit is used to enable automatic page switching on ISR entry/exit. When set to 1, the current SFRPAGE value will be pushed onto the SFR page stack, and SFRPAGE will be set to the page corresponding to the flag which generated the interrupt; upon ISR exit, hardware will pop the value from the SFR page stack and restore SFRPAGE. 0: Disable automatic SFR paging. 1: Enable automatic SFR paging.

Table 19.3. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	SFR Page	Description	Page
ACC	0xE0	All Pages	Accumulator	89
ADC0CF	0xBC	All Pages	ADC0 Configuration	55
ADC0CN	0xE8	All Pages	ADC0 Control	57
ADC0GTH	0xC4	All Pages	ADC0 Greater-Than Compare High	58
ADC0GTL	0xC3	All Pages	ADC0 Greater-Than Compare Low	58
ADC0H	0xBE	All Pages	ADC0 High	56
ADC0L	0xBD	All Pages	ADC0 Low	56
ADC0LTH	0xC6	All Pages	ADC0 Less-Than Compare Word High	59
ADC0LTL	0xC5	All Pages	ADC0 Less-Than Compare Word Low	59
AMX0N	0xBA	All Pages	AMUX0 Negative Channel Select	63
AMX0P	0xBB	All Pages	AMUX0 Positive Channel Select	62
B	0xF0	All Pages	B Register	90
CKCON	0x8E	All Pages	Clock Control	243
CKCON1	0xF4	All Pages	Clock Control 1	244
CLKSEL	0xA9	All Pages	Clock Select	165
CPT0CN	0x9B	All Pages	Comparator0 Control	78
CPT0MD	0x9D	All Pages	Comparator0 Mode Selection	79
CPT0MX	0x9F	All Pages	Comparator0 MUX Selection	81
CRC0AUTO	0xDD	All Pages	CRC0 Automatic Control	152
CRC0CN	0xDF	All Pages	CRC0 Control	150
CRC0CNT	0xDE	All Pages	CRC0 Automatic Flash Sector Count	153
CRC0DAT	0x9E	All Pages	CRC0 Data Output	151
CRC0FLIP	0x9A	All Pages	CRC0 Bit Flip	154
CRC0IN	0x9C	All Pages	CRC0 Data Input	151
DERIVID	0xAB	0	Device Derivative ID	97
DPH	0x83	All Pages	Data Pointer High	88
DPL	0x82	All Pages	Data Pointer Low	88
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	123
EIE2	0xAF	All Pages	Extended Interrupt Enable 2	126
EIP1	0xF6	All Pages	Extended Interrupt Priority 1	124
EIP1H	0x85	All Pages	Extended Interrupt Priority 1 High	125
EIP2	0xBF	All Pages	Extended Interrupt Priority 2	127
EIP2H	0x86	All Pages	Extended Interrupt Priority 2 High	127

21.4.3. System Clock

12. 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.
13. 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.

23.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRC0VAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRC0INIT).

23.3. 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 256 byte blocks read from Flash. 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. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit to 1 in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRC0CNT.
5. 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. **See the note in SFR Definition 23.1. CRC0CN: CRC0 Control for more information on how to properly initiate a CRC calculation.**
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result using the procedure below.

23.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. 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.

23.5. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 23.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

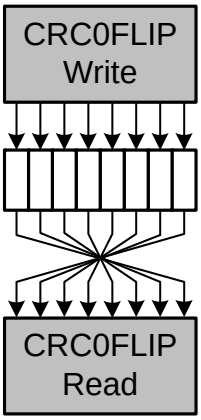

Figure 23.2. Bit Reverse Register

24.3. External Reset

The external RST pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the RST pin generates a reset; an external pullup and/or decoupling of the RST pin may be necessary to avoid erroneous noise-induced resets. See Section “7. Electrical Characteristics” on page 32 for complete RST pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

24.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than 100 µs, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read ‘1’, signifying the MCD as the reset source; otherwise, this bit reads ‘0’. Writing a ‘1’ to the MCDRSF bit enables the Missing Clock Detector; writing a ‘0’ disables it. The state of the RST pin is unaffected by this reset.

24.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a ‘1’ to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read ‘1’ signifying Comparator0 as the reset source; otherwise, this bit reads ‘0’. The state of the RST pin is unaffected by this reset.

24.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in Section “32.4. Watchdog Timer Mode” on page 286; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to ‘1’. The state of the RST pin is unaffected by this reset.

24.7. 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 address 0x3DFF.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above address 0x3DFF.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x3DFF.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section “21.3. Security Options” on page 133).

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the RST pin is unaffected by this reset.

24.8. 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 RST pin is unaffected by this reset.

SFR Definition 27.2. XBR1: Port I/O Crossbar Register 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE		PCA0ME[1:0]	
Type	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2; SFR Page = All Pages

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
4	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.
3	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
2	Unused	Read = 0b; Write = Don't Care.
1:0	PCA0ME[1:0]	**PCA Module I/O Enable Bits.** 00: All PCA I/O unavailable at Port pins. 01: CEX0 routed to Port pin. 10: CEX0, CEX1 routed to Port pins. 11: CEX0, CEX1, CEX2 routed to Port pins.

SFR Definition 28.3. SMBTC: SMBus Timing and Pin Control

Bit	7	6	5	4	3	2	1	0
Name	SMB1SWAP	SMB0SWAP			SMB1SDD[1:0]		SMB0SDD[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 Address = 0xC7; SFR Page = All Pages

Bit	Name	Function
7	SMB1SWAP	**SMBus1 Swap Pins** This bit swaps the order of the SMBus1 pins on the cross-bar. This should be set to 1 when accessing the EEPROM. 0: SDA1 is mapped to the lower-numbered port pin, and SCL1 is mapped to the higher-numbered port pin. 1: SCL1 is mapped to the lower-numbered port pin, and SDA1 is mapped to the higher-numbered port pin.
6	SMB0SWAP	**SMBus0 Swap Pins** This bit swaps the order of the SMBus1 pins on the cross-bar. This should be set to 1 when accessing the EEPROM. 0: SDA0 is mapped to the lower-numbered port pin, and SCL0 is mapped to the higher-numbered port pin. 1: SCL0 is mapped to the lower-numbered port pin, and SDA0 is mapped to the higher-numbered port pin.
5:4	Reserved	Must Write 00b.
3:2	SMB1SDD[1:0]	**SMBus1 Start Detection Window** These bits increase the hold time requirement between SDA falling and SCL falling for START detection. 00: No additional hold time requirement (0-1 SYSCLK). 01: Increase hold time window to 2-3 SYSCLKs. 10: Increase hold time window to 4-5 SYSCLKs. 11: Increase hold time window to 8-9 SYSCLKs.
1:0	SMB0SDD[1:0]	**SMBus0 Start Detection Window** These bits increase the hold time requirement between SDA falling and SCL falling for START detection. 00: No additional hold time window (0-1 SYSCLK). 01: Increase hold time window to 2-3 SYSCLKs. 10: Increase hold time window to 4-5 SYSCLKs. 11: Increase hold time window to 8-9 SYSCLKs.

SFR Definition 28.4. SMB0CN: SMBus Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER0	TXMODE0	STA0	STO0	ACKRQ0	ARBLOST0	ACK0	SI0
Type	R	R	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC0; SFR Page = 0; Bit-Addressable

Bit	Name	Description	Read	Write
7	MASTER0	**SMBus0 Master/Slave Indicator.** This read-only bit indicates when the SMBus0 is operating as a master.	0: SMBus0 operating in slave mode. 1: SMBus0 operating in master mode.	N/A
6	TXMODE0	**SMBus0 Transmit Mode Indicator.** This read-only bit indicates when the SMBus0 is operating as a transmitter.	0: SMBus0 in Receiver Mode. 1: SMBus0 in Transmitter Mode.	N/A
5	STA0	**SMBus0 Start Flag.**	0: No Start or repeated Start detected. 1: Start or repeated Start detected.	0: No Start generated. 1: When Configured as a Master, initiates a START or repeated START.
4	STO0	**SMBus0 Stop Flag.**	0: No Stop condition detected. 1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).	0: No STOP condition is transmitted. 1: When configured as a Master, causes a STOP condition to be transmitted after the next ACK cycle. Cleared by Hardware.
3	ACKRQ0	**SMBus0 Acknowledge Request.**	0: No ACK requested 1: ACK requested	N/A
2	ARBLOST0	**SMBus0 Arbitration Lost Indicator.**	0: No arbitration error. 1: Arbitration Lost	N/A
1	ACK0	**SMBus0 Acknowledge.**	0: NACK received. 1: ACK received.	0: Send NACK 1: Send ACK
0	SI0	**SMBus0 Interrupt Flag.** This bit is set by hardware under the conditions listed in Table 28.3. SI0 must be cleared by software. While SI0 is set, SCL0 is held low and the SMBus0 is stalled.	0: No interrupt pending 1: Interrupt Pending	0: Clear interrupt, and initiate next state machine event. 1: Force interrupt.

28.5.4. Read Sequence (Slave)

During a read sequence, an SMBus master reads data from a slave device. The slave in this transfer will be a receiver during the address byte, and a transmitter during all data bytes. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode (to receive the slave address) when a START followed by a slave address and direction bit (READ in this case) is received. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are transmitted. If the received slave address is acknowledged, data should be written to SMB0DAT to be transmitted. The interface enters slave transmitter mode, and transmits one or more bytes of data. After each byte is transmitted, the master sends an acknowledge bit; if the acknowledge bit is an ACK, SMB0DAT should be written with the next data byte. If the acknowledge bit is a NACK, SMB0DAT should not be written to before SI is cleared (an error condition may be generated if SMB0DAT is written following a received NACK while in slave transmitter mode). The interface exits slave transmitter mode after receiving a STOP. The interface will switch to slave receiver mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 28.8 shows a typical slave read sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur **after** the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

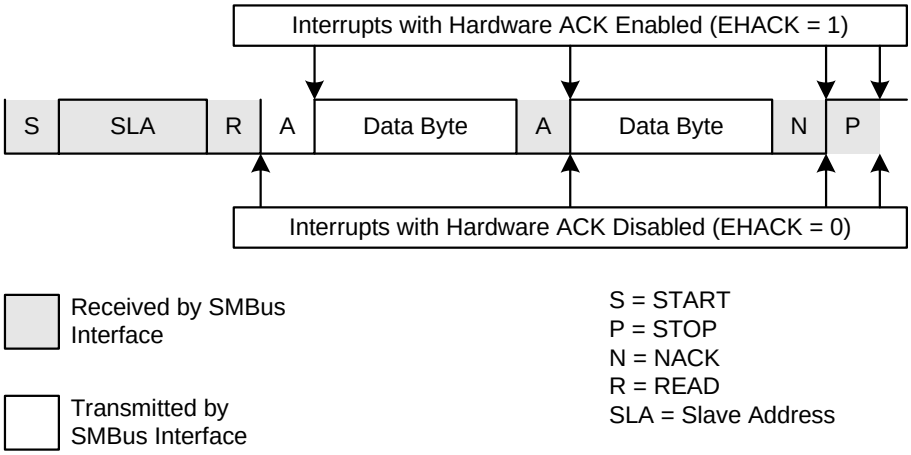

Figure 28.8. Typical Slave Read Sequence

28.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. The appropriate actions to take in response to an SMBus event depend on whether hardware slave address recognition and ACK generation is enabled or disabled. Table 28.5 describes the typical actions when hardware slave address recognition and ACK generation is disabled. Table 28.6 describes the typical actions when hardware slave address recognition and ACK generation is enabled. In the tables, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed by hardware but do not conform to the SMBus specification.

Table 28.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0) (Continued)

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

Table 28.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1) (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK	
Master Receiver	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
		0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	—
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	0101	0	X	X	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X	—

29.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to 1. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to 1.

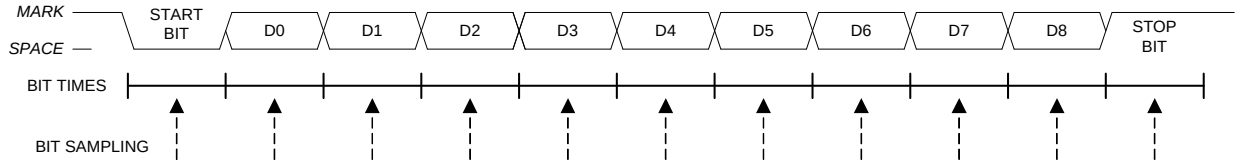

Figure 29.5. 9-Bit UART Timing Diagram

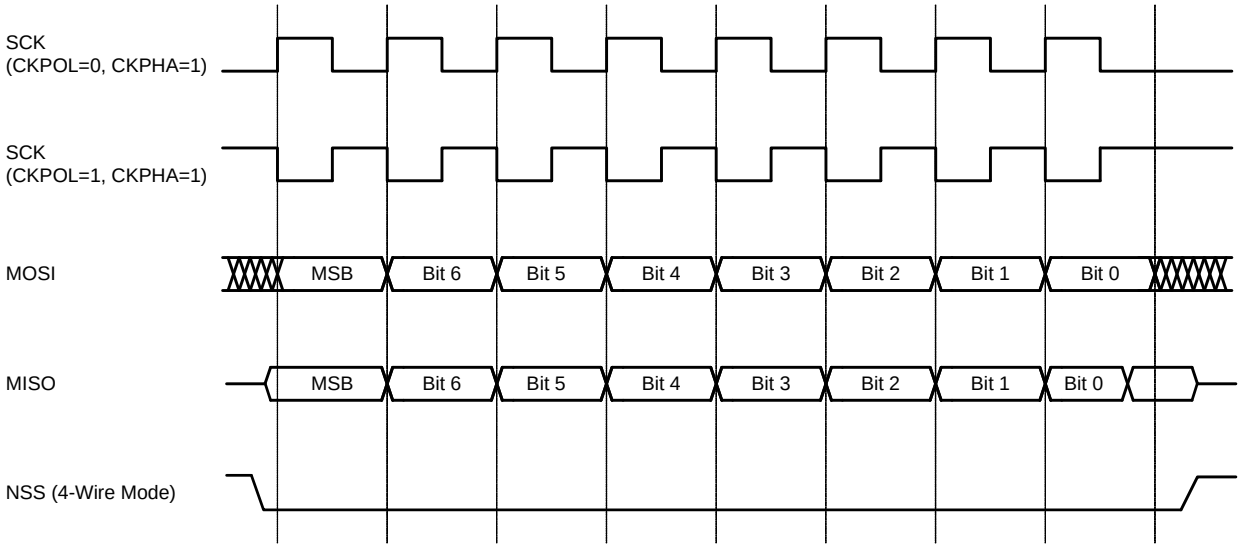

Figure 30.7. Slave Mode Data/Clock Timing (CKPHA = 1)

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.

31.4.2. 8-bit Timers with Auto-Reload

When T4SPLIT is 1 and T4CE = 0, Timer 4 operates as two 8-bit timers (TMR4H and TMR4L). Both 8-bit timers operate in auto-reload mode as shown in Figure 31.11. TMR4RLL holds the reload value for TMR4L; TMR4RLH holds the reload value for TMR4H. The TR4 bit in TMR4CN handles the run control for TMR4H. TMR4L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 4 Clock Select bits (T4MH and T4ML in CKCON1) select either SYSCLK or the clock defined by the Timer 4 External Clock Select bit (T4XCLK in TMR4CN), as follows:

T4MH	T4XCLK	TMR4H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T4ML	T4XCLK	TMR4L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

The TF4H bit is set when TMR4H overflows from 0xFF to 0x00; the TF4L bit is set when TMR4L overflows from 0xFF to 0x00. When Timer 4 interrupts are enabled, an interrupt is generated each time TMR4H overflows. If Timer 4 interrupts are enabled and TF4LEN (TMR4CN.5) is set, an interrupt is generated each time either TMR4L or TMR4H overflows. When TF4LEN is enabled, software must check the TF4H and TF4L flags to determine the source of the Timer 4 interrupt. The TF4H and TF4L interrupt flags are not cleared by hardware and must be manually cleared by software.

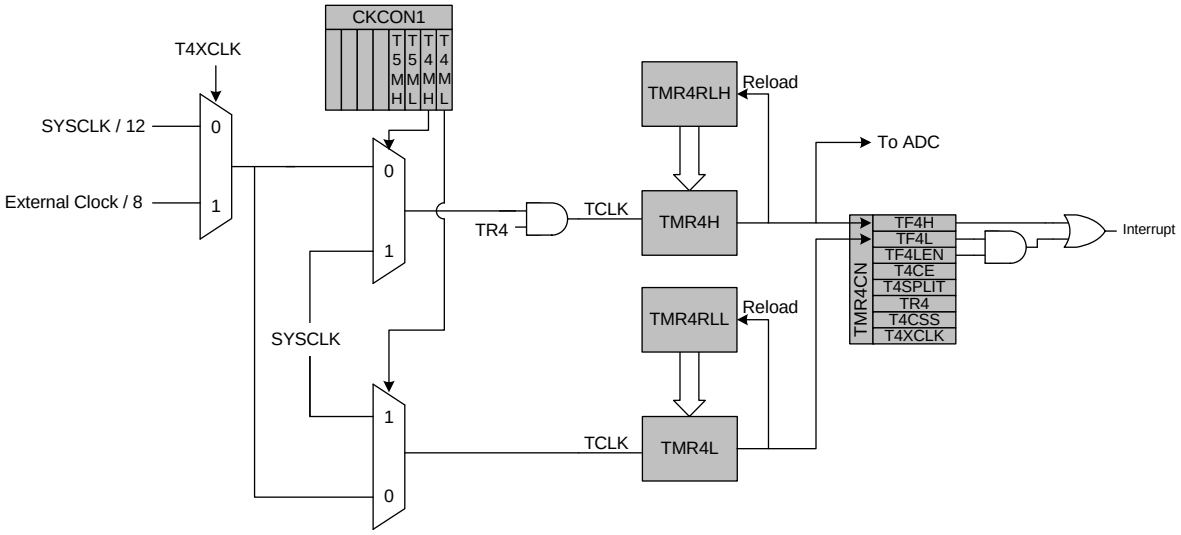

Figure 31.11. Timer 4 8-Bit Mode Block Diagram

C2 Register Definition 33.4. FPCTL: C2 Flash Programming Control

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

C2 Address: 0x02

Bit	Name	Function
7:0	FPCTL[7:0]	**Flash Programming Control Register.** This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 33.5. FPDAT: C2 Flash Programming Data

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

C2 Address: 0xB4

Bit	Name	Function	
7:0	FPDAT[7:0]	**C2 Flash Programming Data Register.** This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.	
		Code	**Command**
		0x06	Flash Block Read
		0x07	Flash Block Write
		0x08	Flash Page Erase
		0x03	Device Erase

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.7

- Added Section 8.1 “Temperature in Two’s Complement”
- Changed clock cycles for “CJNE A, direct, re” to “4/6” in Section 15. “CIP-15 Microcontroller”
- Changed bit 5 of CRC0CNT to reserved in Section 23. “Cyclic Redundancy Check Unit” (CRC0)
- Changed SFRPGCN reset value to “0x01” in Section 19. “Special Function Registers”
- Added Section 19.2 “Interrupts and Automatic SFR Paging”
- Added Section 19.3 “SFR Page Stack Example”
- Corrected incorrect references to C8501F34x in Section 2. “Ordering Information”
- Removed “The C8051F37x does not include the 4x clock multiplier” bullet point from Section 3.1
- Removed “External Oscillator C and RC Modes” bullet point from Section 3.1
- Updated the block diagram on the front page to show EEPROM and 500 ksps ADC
- Removed references to the REG0MD bit and low power mode in Section 13.1
- Removed REG0MD bit in the REG0CN SFR definition. This bit (bit 2) is now reserved.
- Section 22. “EEPROM” completely rewritten
- Moved the “from IPH, EIPH1 or EIPH2” text from the LSB column to the MSB column in Table 20.1
- Moved the “from IP, EIP1 or EIP2” text from the MSB column to the LSB column in Table 20.1
- Changed Figure 27.4 to show all five footnotes
- Changed Figure 27.5 to show correct SF signals and all five footnotes
- Added 5 V tolerance and lock byte address bullet points to Section 3.1. “Hardware Incompatibilities”

Revision 0.7 to Revision 0.71

- Updated part numbers in Table 2.1 on page 20.
- Updated replacement part numbers in Table 3.1 on page 21 to match Flash sizes.
- Corrected units for normal and active mode IDD (V_{DD} = 3.0 V, F = 80 kHz) in Table 7.2 on page 33.
- Updated maximum normal mode IDD in Table 7.2 on page 33.
- Added EESDA and EESCL DC electrical characteristics to Table 7.3 on page 35.
- Added EEPROM supply current to Table 7.6 on page 37.
- Added maximum EESCL clock frequency to Table 7.6 on page 37.
- Updated typical INL and DNL in Table 7.10 on page 39.
- Updated resolution in Table 7.12 on page 40.
- Updated typical and maximum INL and DNL in Table 7.15 on page 42.
- Updated typical full scale error in Table 7.15 on page 42.
- Updated references to Table 28.3 in the SMB0CN and SMB1CN SFR definitions.

Revision 0.71 to Revision 1.0

- Added typical precision temperature sensor curve Figure 7.3 on page 46.
- Added note to CLKSEL SFR definition.
- Corrected CPCPOL description in the PCA0CLR SFR definition.
- Updated C mode K factors in the OSCXCN SFR definition.
- Updated maximum normal mode IDD (F = 50 MHz, F = 25 MHz) in Table 7.2 on page 33.
- Updated typical suspend and stop mode digital supply current in Table 7.2 on page 33.
- Updated typical precision temperature sensor absolute error in Table 7.12 on page 40.
- Updated maximum precision temperature sensor INL in Table 7.12 on page 40.
- Updated minimum IDAC DNL in Table 7.15 on page 42.
- Updated minimum and maximum IDAC full scale error in Table 7.15 on page 42.
- Updated typical IDAC gain variation in Table 7.15 on page 42.
- Updated maximum comparator supply current at DC in Table 7.16 on page 43.
- Corrected flash security restrictions for erase page containing lock byte (if no pages are locked) in Table 21.1 on page 133.