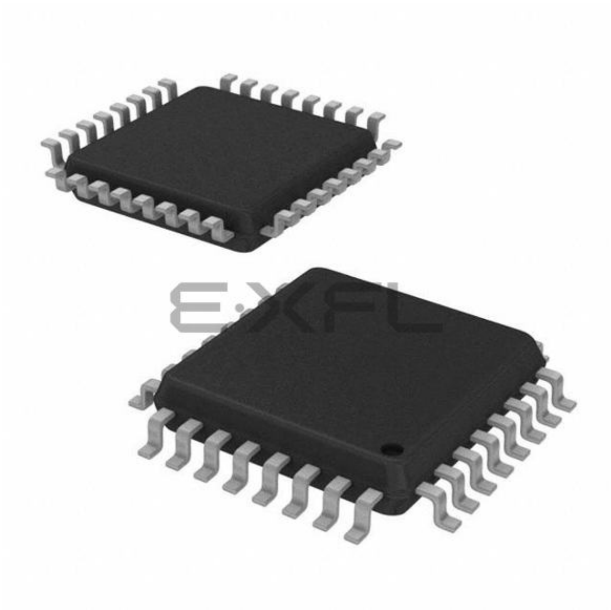

Welcome to E-XFL.COM

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	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x16b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f352-gq

11. Memory Organization and SFRs
Figure 11.1. Memory Map 99
12. Interrupt Handler
13. Prefetch Engine
14. Reset Sources
Figure 14.1. Reset Sources 115
Figure 14.2. Power-On and VDD Monitor Reset Timing 116
15. Flash Memory
Figure 15.1. Flash Memory Map 123
16. External RAM
17. Oscillators
Figure 17.1. Oscillator Diagram 129
Figure 17.2. 32.768 kHz External Crystal Example 132
18. Port Input/Output
Figure 18.1. Port I/O Functional Block Diagram 137
Figure 18.2. Port I/O Cell Block Diagram 138
Figure 18.3. Crossbar Priority Decoder with No Pins Skipped 139
Figure 18.4. Crossbar Priority Decoder with Crystal Pins Skipped 140
19. SMBus
Figure 19.1. SMBus Block Diagram 151
Figure 19.2. Typical SMBus Configuration 152
Figure 19.3. SMBus Transaction 153
Figure 19.4. Typical SMBus SCL Generation 157
Figure 19.5. Typical Master Transmitter Sequence 163
Figure 19.6. Typical Master Receiver Sequence 164
Figure 19.7. Typical Slave Receiver Sequence 165
Figure 19.8. Typical Slave Transmitter Sequence 166
20. UART0
Figure 20.1. UART0 Block Diagram 171
Figure 20.2. UART0 Baud Rate Logic 172
Figure 20.3. UART Interconnect Diagram 173
Figure 20.4. 8-Bit UART Timing Diagram 173
Figure 20.5. 9-Bit UART Timing Diagram 174
Figure 20.6. UART Multi-Processor Mode Interconnect Diagram 175
21. Serial Peripheral Interface (SPI0)
Figure 21.1. SPI Block Diagram 181
Figure 21.2. Multiple-Master Mode Connection Diagram 184
Figure 21.3. 3-Wire Single Master and Slave Mode Connection Diagram 184
Figure 21.4. 4-Wire Single Master and Slave Mode Connection Diagram 184
Figure 21.5. Data/Clock Timing Relationship 186
Figure 21.6. SPI Master Timing (CKPHA = 0) 191
Figure 21.7. SPI Master Timing (CKPHA = 1) 191
Figure 21.8. SPI Slave Timing (CKPHA = 0) 192
Figure 21.9. SPI Slave Timing (CKPHA = 1) 192

1.1. CIP-51™ Microcontroller

1.1.1. Fully 8051 Compatible Instruction Set

The C8051F35x devices use Silicon Labs' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set. Standard 803x/805x assemblers and compilers can be used to develop software. The C8051F35x family has a superset of all the peripherals included with a standard 8052.

1.1.2. Improved Throughput

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 to 24 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's system clock running at 50 MHz, it has a peak throughput of 50 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

1.1.3. Additional Features

The C8051F350/1/2/3 SoC family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

An extended interrupt handler allows the numerous analog and digital peripherals to operate independently of the controller core and interrupt the controller only when necessary. By requiring less intervention from the microcontroller core, an interrupt-driven system is more efficient and allows for easier implementation of multi-tasking, real-time systems.

Eight reset sources are available: power-on reset circuitry (POR), an on-chip V_{DD} monitor, a Watchdog Timer, a Missing Clock Detector, a voltage level detection from Comparator0, a forced software reset, an external reset pin, and an illegal Flash access protection circuit. Each reset source except for the POR, Reset Input Pin, or Flash error may be disabled by the user in software. The WDT may be permanently enabled in software after a power-on reset during MCU initialization.

The internal oscillator is factory calibrated to 24.5 MHz ±2%. An external oscillator drive circuit is also included, allowing an external crystal, ceramic resonator, capacitor, RC, or CMOS clock source to generate the system clock. A clock multiplier allows for operation at up to 50 MHz. An external oscillator can also be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) source, while periodically switching to the fast internal oscillator as needed.

5.2. Calibrating the ADC

ADC0 can be calibrated in-system for both gain and offset, using internal or system calibration modes. To ensure calibration accuracy, offset calibrations must be performed prior to gain calibrations. It is not necessary to perform both internal and system calibrations, as a system calibration will also compensate for any internal error sources.

Offset calibration is a single-point measurement that sets which input voltage produces a zero at the ADC output. When performing an offset calibration, any deviation from zero in the measurement is stored in the offset register. The offset value is subtracted from all conversions as they take place.

Gain calibration is a two-point measurement that sets the slope of the ADC transfer function. When performed, a gain calibration takes only a single measurement, which is assumed to be the desired full-scale value in the ADC transfer function. The offset calibration value is used as the other point in the gain calibration measurement, so that a gain factor can be calculated. After offset correction, conversions are multiplied by the gain factor.

Calibrations are initiated by writing the ADC System Mode bits (AD0SM) to one of the calibration options. During a calibration, the AD0CBSY bit is set to ‘1’. Upon completion of a calibration the the AD0SM bits will return to Idle mode, the AD0CBSY bit will be cleared to ‘0’, the AD0CALC bit will be set to ‘1’, and an ADC interrupt will be generated. Calibration results are also written to the appropriate calibration registers when the calibration is complete.

5.2.1. Internal Calibration

Internal calibration is performed without requiring a specific voltage on the ADC input pins. Internal calibrations can be performed in three different ways: offset only, gain only, or full (offset and gain). A full internal calibration consists of an internal offset calibration followed by an internal gain calibration. If offset and gain calibrations are performed independently, offset calibration must be performed prior to gain calibration. During an internal offset calibration, the ADC inputs are connected internally to AGND. For an internal gain calibration, the ADC inputs are connected internally to a full-scale Voltage that is equal to the selected Voltage reference divided by the PGA gain.

5.2.2. System Calibration

System calibration is performed using voltages which are applied to the ADC inputs. There are two system calibration options: offset calibration and gain calibration. For accurate calibration results, offset calibration must be performed prior to gain calibration. During a system offset calibration, the ADC inputs should be connected to a “zero” value. During a system gain calibration, the ADC inputs should be connected to the positive full-scale value for the current PGA gain setting.

5.2.3. Calibration Coefficient Storage

The calibration results for offset and gain are each 24-bits long. The calibration results are stored in SFRs that are both readable and writeable from software. This enables factory calibrations, as well as manual modification of the offset and gain parameters. The offset calibration results are stored as a two’s complement, 24-bit number in the ADC0COH, ADC0COM, and ADC0COL registers. The mapping of the offset register is shown in Figure 5.3. The gain calibration results are stored as a fixed-point, 24-bit number in the ADC0CGH, ADC0CGM, and ADC0CGL registers. The mapping of the gain register is shown in Figure 5.4.

SFR Definition 5.8. ADC0BUF: ADC0 Input Buffer Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0BPHE	AD0BPLE	AD0BPS		AD0BNHE	AD0BNLE	AD0BNS		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBD

Bit 7: AD0BPHE: Positive Channel High Buffer Enable.
0: Positive Channel High Input Buffer Disabled.
1: Positive Channel High Input Buffer Enabled.

Bit 6: AD0BPLE: Positive Channel Low Enable.
0: Positive Channel Low Input Buffer Disabled.
1: Positive Channel Low Input Buffer Enabled.

Bits 5–4: AD0BPS: Positive Channel Input Selection.
00 = Bypass Input Buffer (default).
01 = Select Low Input Buffer Range.
10 = Select High Input Buffer Range.
11 = Reserved.

Bit 3: AD0BNHE: Negative Channel High Buffer Enable.
0: Negative Channel High Input Buffer Disabled.
1: Negative Channel High Input Buffer Enabled.

Bit 2: AD0BNLE: Negative Channel Low Enable.
0: Negative Channel Low Input Buffer Disabled.
1: Negative Channel Low Input Buffer Enabled.

Bits 1–0: AD0BNS: Negative Channel Input Selection.
00 = Bypass Input Buffer (default).
01 = Select Low Input Buffer Range.
10 = Select High Input Buffer Range.
11 = Reserved.

This SFR can only be modified when ADC0 is in IDLE mode.

SFR Definition 5.10. ADC0COH: ADC0 Offset Calibration Register High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
OCAL23	OCAL22	OCAL21	OCAL20	OCAL19	OCAL18	OCAL17	OCAL16	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBC

Bits 7–0: OCAL[23:16]: ADC0 Offset Calibration Register High Byte.
This register contains the high byte of the 24-bit ADC Offset Calibration Value.

SFR Definition 5.11. ADC0COM: ADC0 Offset Calibration Register Middle Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
OCAL15	OCAL14	OCAL13	OCAL12	OCAL11	OCAL10	OCAL9	OCAL8	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBB

Bits 7–0: OCAL[15:8]: ADC0 Offset Calibration Register Middle Byte.
This register contains the middle byte of the 24-bit ADC Offset Calibration Value.

SFR Definition 5.12. ADC0COL: ADC0 Offset Calibration Register Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
OCAL7	OCAL6	OCAL5	OCAL4	OCAL3	OCAL2	OCAL1	OCAL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBA

Bits 7–0: OCAL[7:0]: ADC0 Offset Calibration Register Low Byte.
This register contains the low byte of the 24-bit ADC Offset Calibration Value.

SFR Definition 5.16. ADC0H: ADC0 Conversion Register (SINC3 Filter) High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADC0H								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC5

Bits 7–0: ADC0H: ADC0 Conversion Register (SINC3 Filter) High Byte.
C8051F350/1: This register contains bits 23–16 of the 24-bit ADC SINC3 filter conversion result.
C8051F352/3: This register contains bits 15–8 of the 16-bit ADC SINC3 filter conversion result.

SFR Definition 5.17. ADC0M: ADC0 Conversion Register (SINC3 Filter) Middle Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADC0M								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC4

Bits 7–0: ADC0M: ADC0 Conversion Register (SINC3 Filter) Middle Byte.
C8051F350/1: This register contains bits 15–8 of the 24-bit ADC SINC3 filter conversion result.
C8051F352/3: This register contains bits 7–0 of the 16-bit ADC SINC3 filter conversion result.

SFR Definition 5.18. ADC0L: ADC0 Conversion Register (SINC3 Filter) Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ADC0L								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC3

Bits 7–0: ADC0L: ADC0 Conversion Register (SINC3 Filter) Low Byte.
C8051F350/1: This register contains bits 7–0 of the 24-bit ADC SINC3 filter conversion result.
C8051F352/3: This register contains all zeros (00000000b).

5.6. Analog Multiplexer

ADC0 includes analog multiplexer circuitry with independent selection capability for the AIN+ and AIN– inputs. Each input can be connected to one of ten possible input sources: AIN0.0 though AIN0.7, AGND, or the on-chip temperature sensor circuitry (Figure 5.5). The ADC0MUX register (SFR Definition 5.22) controls the input mux selection for both input channels. The multiplexer configuration allows for measurement of single-ended or differential signals. A single-ended measurement can be performed by connecting one of the ADC inputs to AGND. Additionally, the temperature sensor can be measured in single-ended or differential mode. The temperature sensor is automatically enabled when it is selected with the ADC multiplexer. See Section “8. Temperature Sensor’ on page 77 for more details on the temperature sensor.

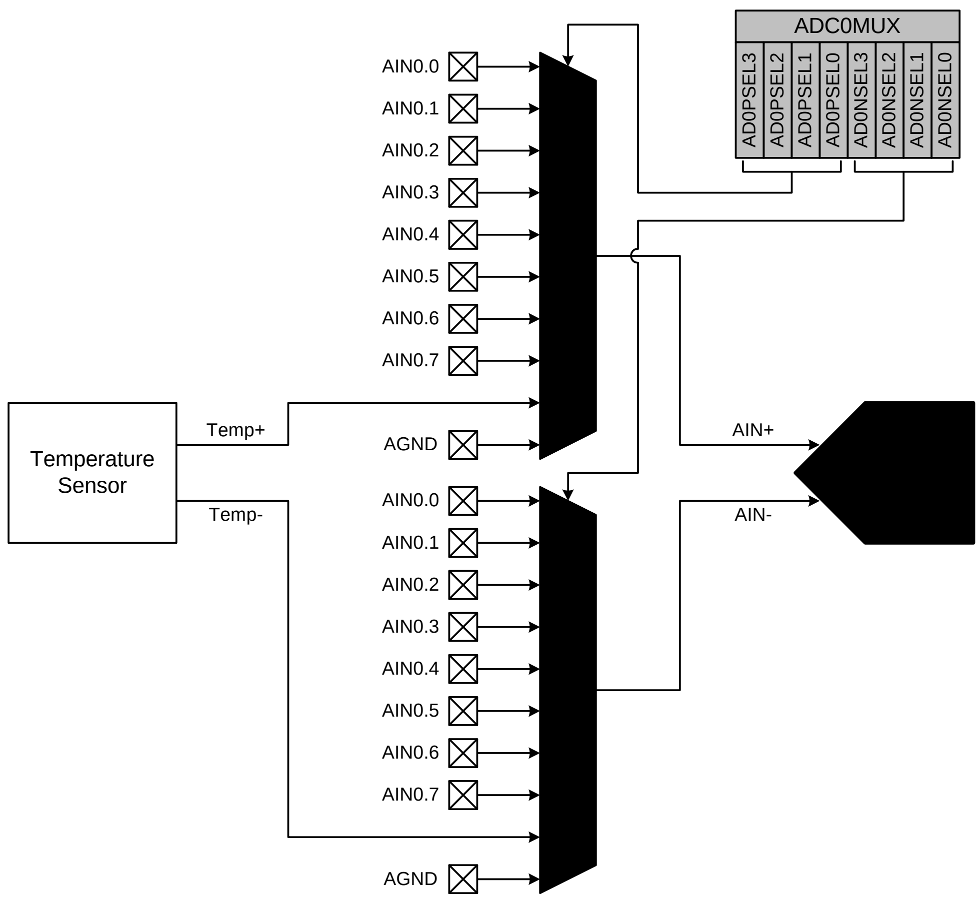

Figure 5.5. ADC0 Multiplexer Connections

Table 5.3. ADC0 Electrical Characteristics

V_{DD} = AV+ = 3.0 V, VREF = 2.5 V External, PGA Gain = 1, MDCLK = 2.4576 MHz,
Decimation Ratio = 1920, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
24-bit ADC (C8051F350/1)					
Resolution			24		bits
No Missing Codes			24		bits
16-bit ADC (C8051F352/3)					
Resolution			16		bits
No Missing Codes			16		bits
All Devices					
Integral Nonlinearity		—	—	±15	ppm FS
Offset Error (Calibrated)		—	±5	—	ppm
Offset Drift vs. Temperature		—	10	—	nV/°C
Gain Error (Calibrated)		—	±0.002	—	%
Gain Drift vs. Temperature		—	±0.5	—	ppm/°C
Modulator Clock (MDCLK)		—	2.4576	—	MHz
Modulator Sampling Frequency			MDCLK/128		Hz
Output Word Rate		—	—	1000	sps
Analog Inputs					
Analog Input Voltage Range (AIN+ – AIN–)	PGA Gain = 1, Bipolar PGA Gain = 1, Unipolar	–VREF 0	— —	+VREF +VREF	V
Absolute Voltage on AIN+ or AIN– pin with respect to AGND	Input Buffers OFF	0	—	AV+	V
Input Current	Input Buffer ON	—	± 1.5	30	nA
Input Impedance	Input Buffer OFF, Gain = 1	—	7	—	MΩ
Common Mode Rejection Ratio	DC 50/60 Hz	95	110 100	— —	dB dB
Input Buffers					
High Buffer Input Range with respect to AGND	PGA Gain = 1, 2, 4, or 8 PGA Gain = 16 PGA Gain = 32 PGA Gain = 64 or 128	1.4 1.45 1.5 1.6	— — — —	AV+ – 0.1 AV+ – 0.15 AV+ – 0.2 AV+ – 0.25	V V V V
Low Buffer Input Range with respect to AGND	PGA Gain = 1, 2, 4, or 8 PGA Gain = 16 PGA Gain = 32 PGA Gain = 64 or 128	0.1 0.15 0.2 0.25	— — — —	AV+ – 1.4 AV+ – 1.45 AV+ – 1.5 AV+ – 1.6	V V V V
Burnout Current Sources					
Positive (AIN+) Channel Current	VREF = 2.5 V	0.9	2	2.9	µA
Negative (AIN–) Channel Current	VREF = 2.5 V	–0.9	–2	–2.9	µA

6. 8-Bit Current Mode DACS (IDA0 and IDA1)

The C8051F350/1/2/3 devices include two 8-bit current-mode Digital-to-Analog Converters (IDACs). The maximum current output of the IDACs can be adjusted for four different current settings; 0.25 mA, 0.5 mA, 1 mA, and 2 mA. The IDACs can be individually enabled or disabled using the enable bits in the corresponding IDAC Control Register (IDA0CN or IDA1CN). An internal bandgap bias generator is used to generate a reference current for the IDACs whenever they are enabled. IDAC updates can be performed on-demand, scheduled on a Timer overflow, or synchronized with an external pin edge. Figure 6.1 shows a block diagram of the IDAC circuitry.

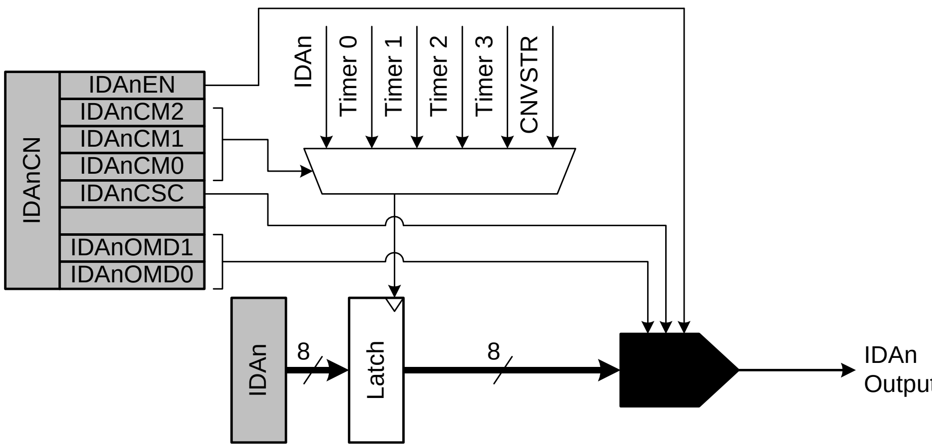

Figure 6.1. IDAC Functional Block Diagram

SFR Definition 10.4. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xD0

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: RS1–RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00 – 0x07
0	1	1	0x08 – 0x0F
1	0	2	0x10 – 0x17
1	1	3	0x18 – 0x1F

Bit2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
• An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
• A MUL instruction results in an overflow (result is greater than 255).
• A DIV instruction causes a divide-by-zero condition.
The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

SFR Definition 18.5. P0MDOUT: Port0 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA4

Bits7–0: Output Configuration Bits for P0.7–P0.0 (respectively): ignored if corresponding bit in register P0MDIN is logic 0.
0: Corresponding P0.n Output is open-drain.
1: Corresponding P0.n Output is push-pull.

(Note: When SDA and SCL appear on any of the Port I/O, each are open-drain regardless of the value of P0MDOUT).

SFR Definition 18.6. P0SKIP: Port0 Skip

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD4

Bits7–0: P0SKIP[7:0]: Port0 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (VREF input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P0.n pin is not skipped by the Crossbar.
1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 18.11. P2: Port2

R	R	R	R	R	R	R	R/W	Reset Value
—	—	—	—	—	—	—	P2.0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xA0

Bits7–1: Unused. Read = 0000000b. Write = don’t care.
Bit0: P2.0
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P2MDOUT.n bit = 0).
Read - Directly reads Port pin.
0: P2.n pin is logic low.
1: P2.n pin is logic high.

SFR Definition 18.12. P2MDOUT: Port2 Output Mode

R	R	R	R	R	R	R	R/W	Reset Value
—	—	—	—	—	—	—		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA6

Bits7–1: Unused. Read = 0000000b. Write = don’t care.
Bit0: Output Configuration Bit for P2.0.
0: P2.0 Output is open-drain.
1: P2.0 Output is push-pull.

19. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I2C 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. Three SFRs are associated with the SMBus: SMB0CF configures the SMBus; SMB0CN controls the status of the SMBus; and SMB0DAT is the data register, used for both transmitting and receiving SMBus data and slave addresses.

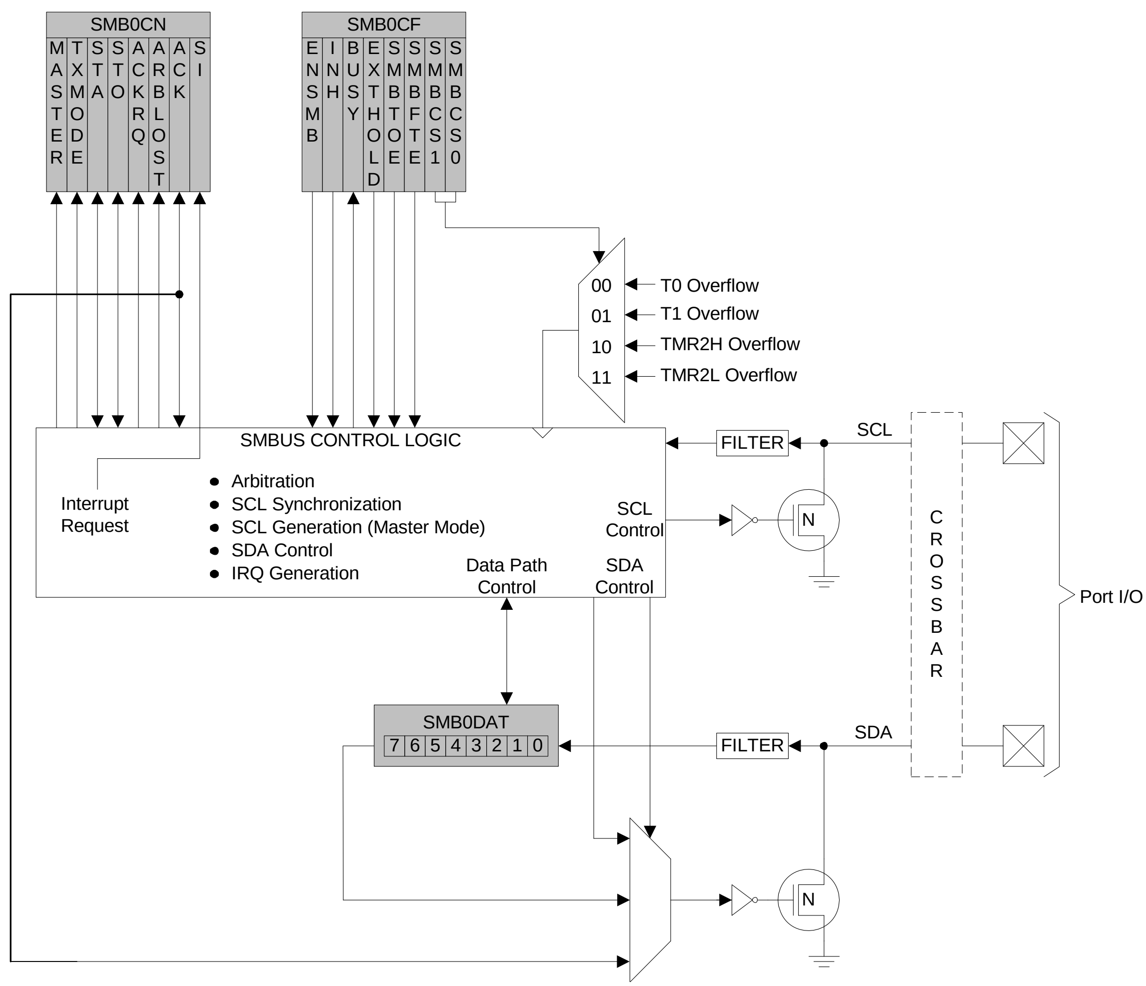

Figure 19.1. SMBus Block Diagram

19.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

Table 19.1. SMBus Clock Source Selection

SMBCS1	SMBCS0	SMBus Clock Source
0	0	Timer 0 Overflow
0	1	Timer 1 Overflow
1	0	Timer 2 High Byte Overflow
1	1	Timer 2 Low Byte Overflow

The SMBCS1–0 bits select the SMBus clock source, which is used only when operating as a master or when the Free Timeout detection is enabled. When operating as a master, overflows from the selected source determine the absolute minimum SCL low and high times as defined in Equation 19.1. Note that the selected clock source may be shared by other peripherals so long as the timer is left running at all times. For example, Timer 1 overflows may generate the SMBus and UART baud rates simultaneously. Timer configuration is covered in Section “22. Timers’ on page 195.

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

Equation 19.1. Minimum SCL High and Low Times

The selected clock source should be configured to establish the minimum SCL High and Low times as per Equation 19.1. When the interface is operating as a master (and SCL is not driven or extended by any other devices on the bus), the typical SMBus bit rate is approximated by Equation 19.2.

$$BitRate = \frac{f_{ClockSourceOverflow}}{3}$$

Equation 19.2. Typical SMBus Bit Rate

19.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 19.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER and TXMODE indicate the master/slave state and transmit/receive modes, respectively.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a ‘1’ to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a ‘1’ to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 19.3 for more details.

Important note about the SI bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

Table 19.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 19.4 for SMBus status decoding using the SMB0CN register.

19.5.3. Slave Receiver Mode

Serial data is received on SDA and the clock is received on SCL. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. Upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. Software responds to the received slave address with an ACK, or ignores the received slave address with a NACK. If the received slave address is ignored, slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received. Software must write the ACK bit after each received byte to ACK or NACK the received byte. The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 19.7 shows a typical Slave Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

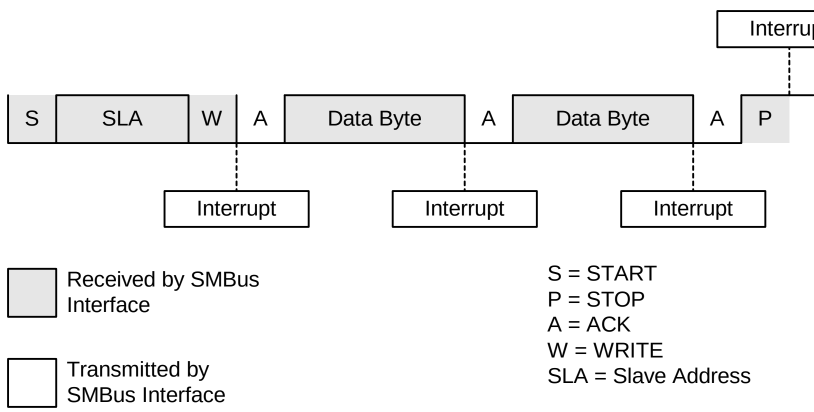

Figure 19.7. Typical Slave Receiver Sequence

20.2.2. 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’.

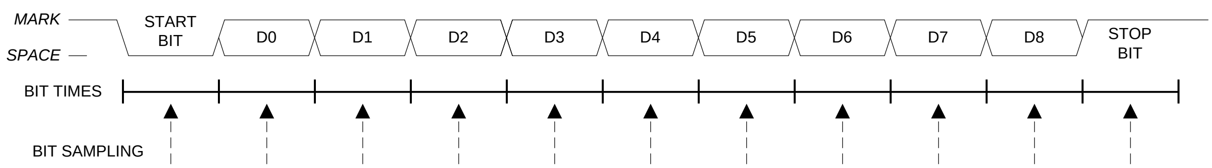

Figure 20.5. 9-Bit UART Timing Diagram

20.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

SFR Definition 21.1. SPI0CFG: SPI0 Configuration

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA1

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic 1 when a SPI transfer is in progress (Master or Slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*

Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: SLVSEL: Slave Selected Flag (read only).
This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.

Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.

Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0.
NOTE: RXBMT = 1 when in Master Mode.

***Note:** See Table 21.1 for timing parameters.

SFR Definition 22.3. CKCON: Clock Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8E

Bit7: T3MH: Timer 3 High Byte Clock Select.
This bit selects the clock supplied to the Timer 3 high byte if Timer 3 is configured in split 8-bit timer mode. T3MH is ignored if Timer 3 is in any other mode.
0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 high byte uses the system clock.

Bit6: T3ML: Timer 3 Low Byte Clock Select.
This bit selects the clock supplied to Timer 3. If Timer 3 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 low byte uses the system clock.

Bit5: T2MH: Timer 2 High Byte Clock Select.
This bit selects the clock supplied to the Timer 2 high byte if Timer 2 is configured in split 8-bit timer mode. T2MH is ignored if Timer 2 is in any other mode.
0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 high byte uses the system clock.

Bit4: T2ML: Timer 2 Low Byte Clock Select.
This bit selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 low byte uses the system clock.

Bit3: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic 1.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.

Bit2: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic 1.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Counter/Timer 0 uses the system clock.

Bits1–0: SCA1–SCA0: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48
1	1	External clock divided by 8

Note: External clock divided by 8 is synchronized with the system clock.

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 must not 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 generally not intended 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, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world’s most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem ®, Precision32®, ProSLIC®, SiPHY®, 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