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	25MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 16x12b
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	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f851-c-gu

1.2.3. Port I/O Output Drive

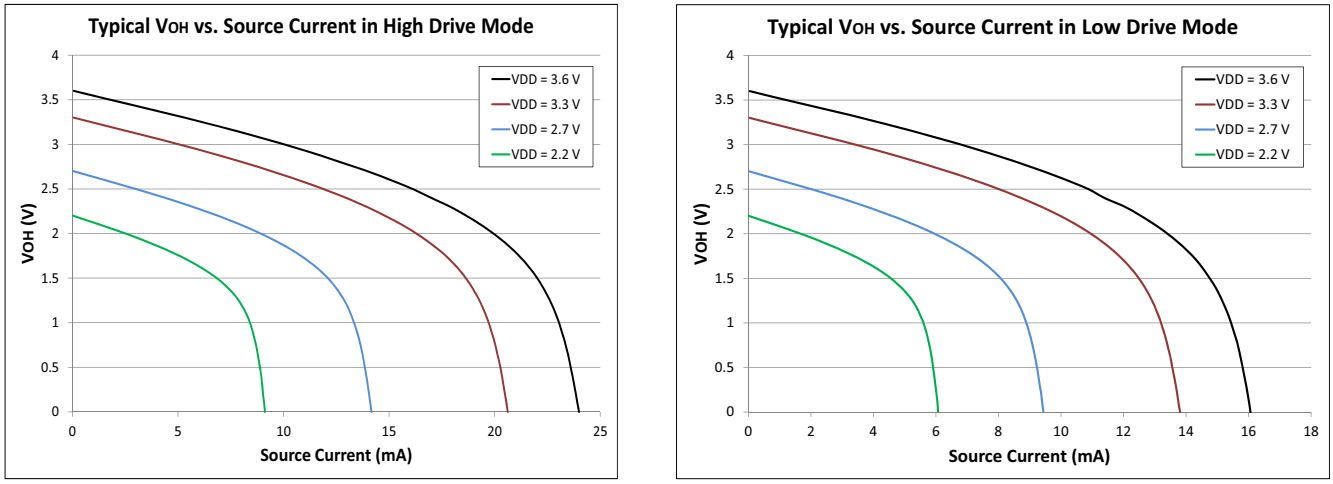

Figure 1.5. Typical V_{OH} vs. Source Current

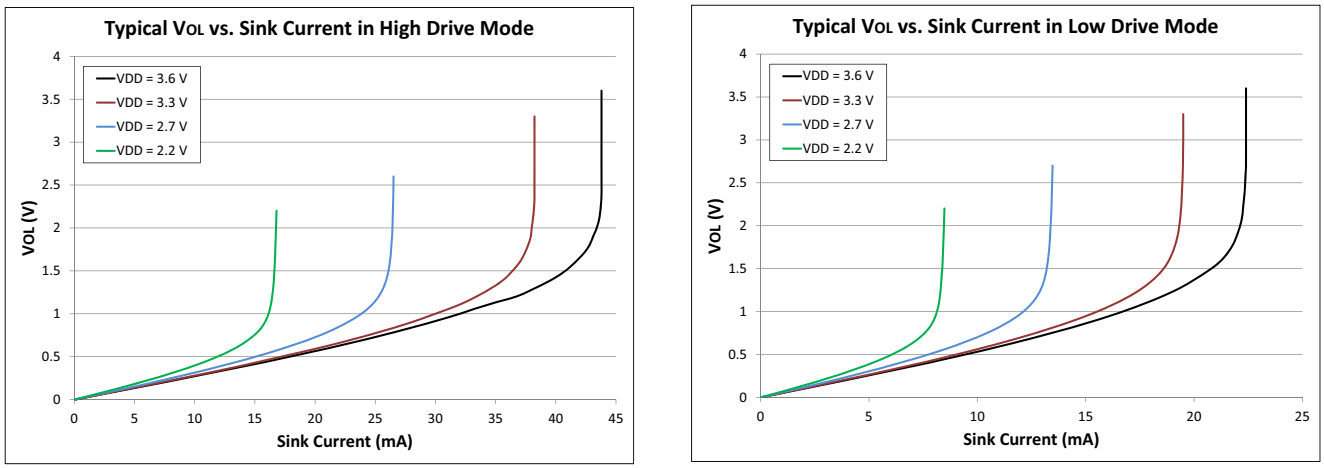

Figure 1.6. Typical V_{OL} vs. Sink Current

1.3. Thermal Conditions

Table 1.12. Thermal Conditions

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Thermal Resistance*	θ_{JA}	SOIC-16 Packages	—	70	—	°C/W
		QFN-20 Packages	—	60	—	°C/W
		QSOP-24 Packages	—	65	—	°C/W
***Note:** Thermal resistance assumes a multi-layer PCB with any exposed pad soldered to a PCB pad.						

6. QFN-20 Package Specifications

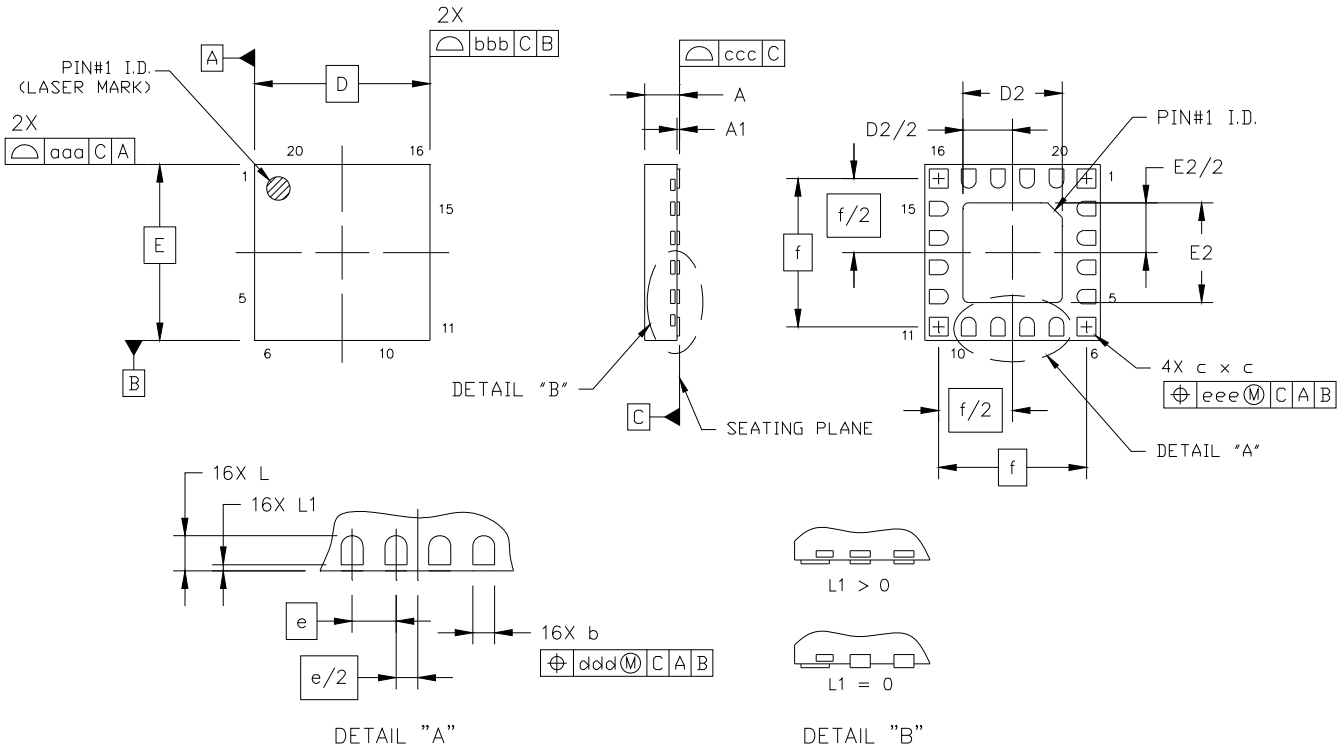

Figure 6.1. QFN-20 Package Drawing

Table 6.1. QFN-20 Package Dimensions

Symbol	Millimeters		
	Min	Nom	Max
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.20	0.25	0.30
c	0.25	0.30	0.35
D		3.00 BSC	
D2	1.6	1.70	1.8
e		0.50 BSC	
E		3.00 BSC	
E2	1.6	1.70	1.8

Symbol	Millimeters		
	Min	Nom	Max
f		2.53 BSC	
L	0.3	0.40	0.5
L1	0.00	—	0.10
aaa	—	—	0.05
bbb	—	—	0.05
ccc	—	—	0.08
ddd	—	—	0.10
eee	—	—	0.10

Notes:
1. All dimensions are shown in millimeters unless otherwise noted.
2. Dimensioning and tolerancing per ANSI Y14.5M-1994.

8.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F85x/86x family implements 8 kB, 4 kB or 2 kB of this program memory space as in-system, re-programmable flash memory. The last address in the flash block (0x1FFF on 8 kB devices, 0x0FFF on 4 kB devices and 0x07FF on 2 kB devices) serves as a security lock byte for the device, and provides read, write and erase protection. Addresses above the lock byte within the 64 kB address space are reserved.

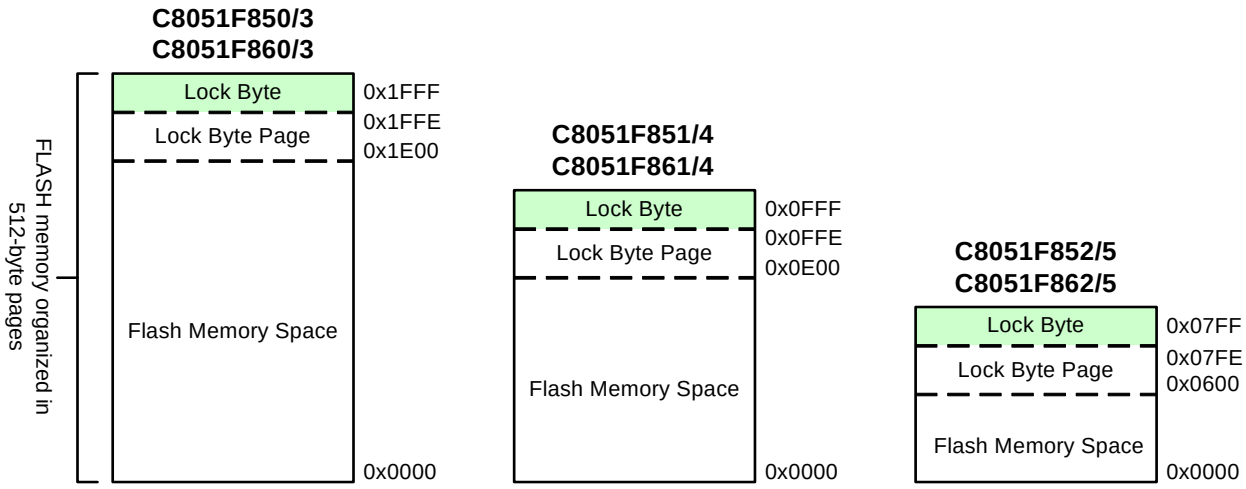

Figure 8.2. Flash Program Memory Map

8.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F85x/86x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip flash memory space. MOVC instructions are always used to read flash memory, while MOVX write instructions are used to erase and write flash. This flash access feature provides a mechanism for the C8051F85x/86x to update program code and use the program memory space for non-volatile data storage. Refer to Section “10. Flash Memory” on page 61 for further details.

8.2. Data Memory

The C8051F85x/86x device family includes up to 512 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. On devices with 512 bytes total RAM, 256 additional bytes of memory are available as on-chip “external” memory. The data memory map is shown in Figure 8.1 for reference.

8.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

9. Special Function Register Memory Map

This section details the special function register memory map for the C8051F85x/86x devices.

Table 9.1. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	P0MAT	P0MASK	VDM0CN
F0	B	P0MDIN	P1MDIN	EIP1	-	-	PRTDRV	PCA0PWM
E8	ADC0CN0	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	P1MAT	P1MASK	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF	-	EIE1	-
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	CRC0IN	CRC0DAT	ADC0PWR
D0	PSW	REF0CN	CRC0AUTO	CRC0CNT	P0SKIP	P1SKIP	SMB0ADM	SMB0ADR
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	CRC0CN	CRC0FLIP
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	OSCICL
B8	IP	ADC0TK	-	ADC0MX	ADC0CF	ADC0L	ADC0H	CPT1CN
B0	-	OSCLCN	ADC0CN1	ADC0AC	-	DEVICEID	REVID	FLKEY
A8	IE	CLKSEL	CPT1MX	CPT1MD	SMB0TC	DERIVID	-	-
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	-
98	SCON0	SBUF0	-	CPT0CN	PCA0CLR	CPT0MD	PCA0CENT	CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	PCA0POL	WDTCN
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	-	-	-	PCON
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

(bit addressable)

Table 9.2. Special Function Registers

Register	Address	Register Description	Page
ACC	0xE0	Accumulator	122
ADC0AC	0xB3	ADC0 Accumulator Configuration	102
ADC0CF	0xBC	ADC0 Configuration	101
ADC0CN0	0xE8	ADC0 Control 0	99
ADC0CN1	0xB2	ADC0 Control 1	100
ADC0GTH	0xC4	ADC0 Greater-Than High Byte	107
ADC0GTL	0xC3	ADC0 Greater-Than Low Byte	108
ADC0H	0xBE	ADC0 Data Word High Byte	105

Table 9.2. Special Function Registers (Continued)

Register	Address	Register Description	Page
IE	0xA8	Interrupt Enable	75
IP	0xB8	Interrupt Priority	77
IT01CF	0xE4	INT0 / INT1 Configuration	150
OSCICL	0xC7	High Frequency Oscillator Calibration	127
OSCLCN	0xB1	Low Frequency Oscillator Control	128
P0	0x80	Port 0 Pin Latch	199
P0MASK	0xFE	Port 0 Mask	197
P0MAT	0xFD	Port 0 Match	198
P0MDIN	0xF1	Port 0 Input Mode	200
P0MDOUT	0xA4	Port 0 Output Mode	201
P0SKIP	0xD4	Port 0 Skip	202
P1	0x90	Port 1 Pin Latch	205
P1MASK	0xEE	Port 1 Mask	203
P1MAT	0xED	Port 1 Match	204
P1MDIN	0xF2	Port 1 Input Mode	206
P1MDOUT	0xA5	Port 1 Output Mode	207
P1SKIP	0xD5	Port 1 Skip	208
P2	0xA0	Port 2 Pin Latch	209
P2MDOUT	0xA6	Port 2 Output Mode	210
PCA0CENT	0x9E	PCA Center Alignment Enable	177
PCA0CLR	0x9C	PCA Comparator Clear Control	170
PCA0CN	0xD8	PCA Control	167
PCA0CPH0	0xFC	PCA Capture Module High Byte 0	175
PCA0CPH1	0xEA	PCA Capture Module High Byte 1	181
PCA0CPH2	0xEC	PCA Capture Module High Byte 2	183
PCA0CPL0	0xFB	PCA Capture Module Low Byte 0	174
PCA0CPL1	0xE9	PCA Capture Module Low Byte 1	180
PCA0CPL2	0xEB	PCA Capture Module Low Byte 2	182

10.2. Programming the Flash Memory

Writes to flash memory clear bits from logic 1 to logic 0, and can be performed on single byte locations. Flash erasures set bits back to logic 1, and occur only on full pages. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a flash write/erase operation.

The simplest means of programming the flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device.

To ensure the integrity of flash contents, it is strongly recommended that the on-chip supply monitor be enabled in any system that includes code that writes and/or erases flash memory from software.

10.2.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a flash write or erase is attempted before the key codes have been written properly. The flash lock resets after each write or erase; the key codes must be written again before a following flash operation can be performed.

10.2.2. Flash Erase Procedure

The flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to flash memory using MOVX, flash write operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit in the PSCTL register to logic 1 (this directs the MOVX writes to target flash memory); and (2) Writing the flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

A write to flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in flash. **A byte location to be programmed should be erased before a new value is written.** Erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire page, perform the following steps:

1. Disable interrupts (recommended).
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. Write the first key code to FLKEY: 0xA5.
5. Write the second key code to FLKEY: 0xF1.
6. Using the MOVX instruction, write a data byte to any location within the page to be erased.
7. Clear the PSWE and PSEE bits.

10.2.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the flash page containing the target location, as described in Section 10.2.2.
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the desired

Table 12.4. EIE1 Register Bit Descriptions

Bit	Name	Function
0	ESMB0	**Enable SMBus (SMB0) Interrupt.** This bit sets the masking of the SMB0 interrupt. 0: Disable all SMB0 interrupts. 1: Enable interrupt requests generated by SMB0.

Register 14.4. ADC0AC: ADC0 Accumulator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AD12BE	ADAE	ADSJST			ADRPT		
Type	RW	RW	RW			RW		
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xB3								

Table 14.7. ADC0AC Register Bit Descriptions

Bit	Name	Function
7	AD12BE	**12-Bit Mode Enable.** Enables 12-bit Mode. In 12-bit mode, the ADC throughput is reduced by a factor of 4. 0: 12-bit Mode Disabled. 1: 12-bit Mode Enabled.
6	ADAE	**Accumulate Enable.** Enables multiple conversions to be accumulated when burst mode is disabled. 0: ADC0H:ADC0L contain the result of the latest conversion when Burst Mode is disabled. 1: ADC0H:ADC0L contain the accumulated conversion results when Burst Mode is disabled. Software must write 0x0000 to ADC0H:ADC0L to clear the accumulated result.
5:3	ADSJST	**Accumulator Shift and Justify.** Specifies the format of data read from ADC0H:ADC0L. All remaining bit combinations are reserved. 000: Right justified. No shifting applied. 001: Right justified. Shifted right by 1 bit. 010: Right justified. Shifted right by 2 bits. 011: Right justified. Shifted right by 3 bits. 100: Left justified. No shifting applied. 101-111: Reserved.
2:0	ADRPT	**Repeat Count.** Selects the number of conversions to perform and accumulate in Burst Mode. This bit field must be set to 000 if Burst Mode is disabled. 000: Perform and Accumulate 1 conversion (not used in 12-bit mode). 001: Perform and Accumulate 4 conversions (1 conversion in 12-bit mode). 010: Perform and Accumulate 8 conversions (2 conversions in 12-bit mode). 011: Perform and Accumulate 16 conversions (4 conversions in 12-bit mode). 100: Perform and Accumulate 32 conversions (8 conversions in 12-bit mode). 101: Perform and Accumulate 64 conversions (16 conversions in 12-bit mode). 110-111: Reserved.

Register 15.2. DPH: Data Pointer High

Bit	7	6	5	4	3	2	1	0
Name	DPH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0x83								

Table 15.3. DPH Register Bit Descriptions

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

Register 15.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS		OV	F1	PARITY
Type	RW	RW	RW	RW		RW	RW	R
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD0 (bit-addressable)								

Table 15.7. PSW Register Bit Descriptions

Bit	Name	Function
7	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 logic 0 by all other arithmetic operations.
6	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 logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under software control.
4:3	RS	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: 1. An ADD, ADDC, or SUBB instruction causes a sign-change overflow. 2. A MUL instruction results in an overflow (result is greater than 255). 3. 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.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under software control.
0	PARITY	**Parity Flag.** This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

Register 20.8. PCA0CPL0: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPL0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xFB								

Table 20.10. PCA0CPL0 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPL0	**PCA Capture Module Low Byte.** The PCA0CPL0 register holds the low byte (LSB) of the 16-bit capture module.This register address also allows access to the low byte of the corresponding PCA channels auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOM bit to a 0.		

Register 20.16. PCA0CPL2: PCA Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPL2							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xEB								

Table 20.18. PCA0CPL2 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPL2	**PCA Capture Module Low Byte.** The PCA0CPL2 register holds the low byte (LSB) of the 16-bit capture module.This register address also allows access to the low byte of the corresponding PCA channels auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the modules ECOM bit to a 0.		

Note: The SPI can be operated in either 3-wire or 4-wire modes, pending the state of the NSSMD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a port pin. The order in which SMBus pins are assigned is defined by the SWAP bit in the SMB0TC register.

Register 21.17. P2: Port 2 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	Reserved						P2	
Type	R						RW	
Reset	0	0	0	0	0	0	1	1
SFR Address: 0xA0 (bit-addressable)								

Table 21.20. P2 Register Bit Descriptions

Bit	Name	Function
7:2	Reserved	Must write reset value.
1:0	P2	**Port 2 Data.** Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.
Note: Port 2 consists of 2 bits (P2.0-P2.1) on QSOP24 devices and 1 bit (P2.0) on QFN20 and SOIC16 packages.		

Table 23.3. SPI0CN Register Bit Descriptions

Bit	Name	Function
0	SPIEN	**SPI0 Enable.** 0: SPI disabled. 1: SPI enabled.

24.1. Supporting Documents

It is assumed the reader is familiar with or has access to the following supporting documents:

1. The I^{2}C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I^{2}C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

24.2. SMBus Configuration

Figure 24.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. However, the maximum voltage on any port pin must conform to the electrical characteristics specifications. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

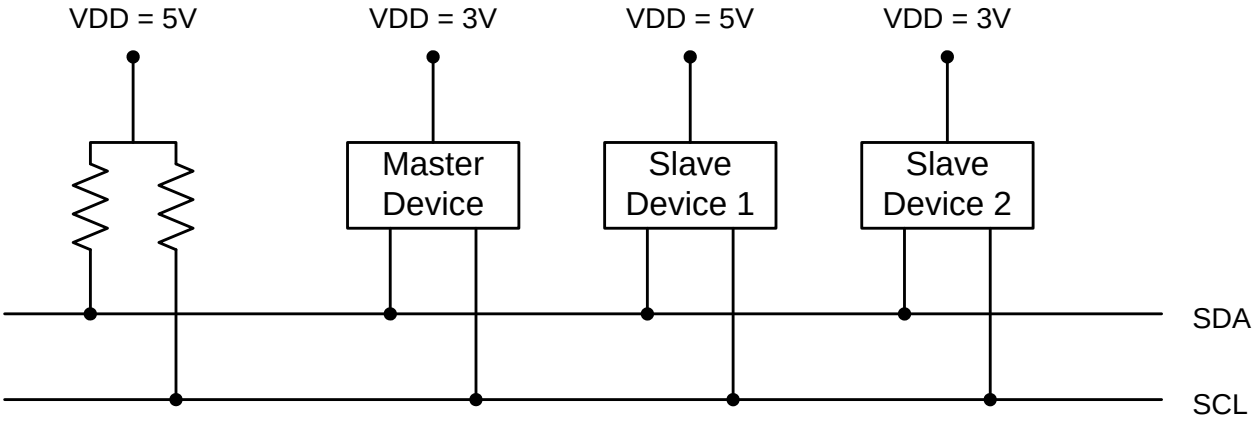

Figure 24.2. Typical SMBus Configuration

24.3. SMBus Operation

Two types of data transfers are possible: data transfers from a master transmitter to an addressed slave receiver (WRITE), and data transfers from an addressed slave transmitter to a master receiver (READ). The master device initiates both types of data transfers and provides the serial clock pulses on SCL. The SMBus interface may operate as a master or a slave, and multiple master devices on the same bus are supported. If two or more masters attempt to initiate a data transfer simultaneously, an arbitration scheme is employed with a single master always winning the arbitration. It is not necessary to specify one device as the Master in a system; any device who transmits a START and a slave address becomes the master for the duration of that transfer.

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 24.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address byte. The direction bit is set to logic 1 to indicate a "READ" operation and cleared to logic 0 to indicate a "WRITE" operation.

Table 24.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV	Slave Address Mask SLVM	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

24.4.6. Data Register

The SMBus Data register SMB0DAT holds a byte of serial data to be transmitted or one that has just been received. Software may safely read or write to the data register when the SI flag is set. Software should not attempt to access the SMB0DAT register when the SMBus is enabled and the SI flag is cleared to logic 0, as the interface may be in the process of shifting a byte of data into or out of the register.

Data in SMB0DAT is always shifted out MSB first. After a byte has been received, the first bit of received data is located at the MSB of SMB0DAT. While data is being shifted out, data on the bus is simultaneously being shifted in. SMB0DAT always contains the last data byte present on the bus. In the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data or address in SMB0DAT.

24.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. 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. 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 received.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

The interface exits Slave Receiver Mode after receiving a STOP. The interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 24.7 shows a typical slave write sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs **before** the ACK with hardware ACK generation disabled, and **after** the ACK when hardware ACK generation is enabled.

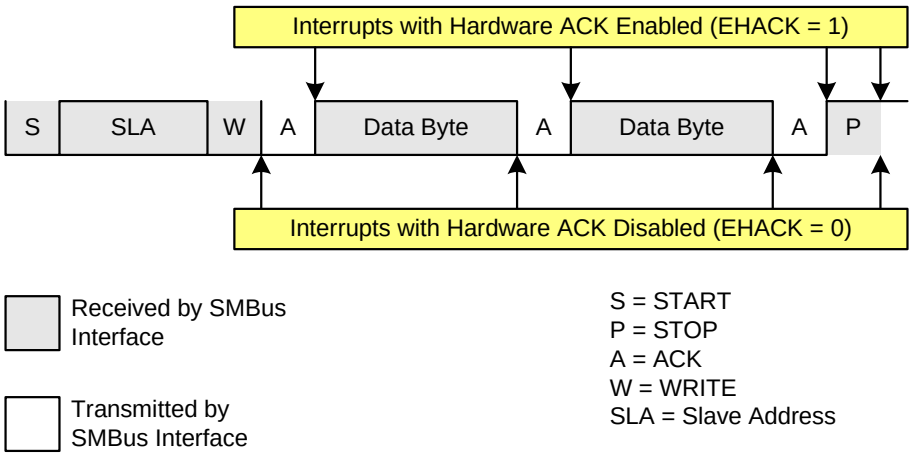

Figure 24.7. Typical Slave Write Sequence

Register 29.2. C2DEVID: C2 Device ID

Bit	7	6	5	4	3	2	1	0
Name	C2DEVID							
Type	R							
Reset	0	0	1	1	0	0	0	0
C2 Address: 0x00								

Table 29.2. C2DEVID Register Bit Descriptions

Bit	Name	Function
7:0	C2DEVID	**Device ID.** This read-only register returns the 8-bit device ID: 0x30 (C8051F85x/86x).