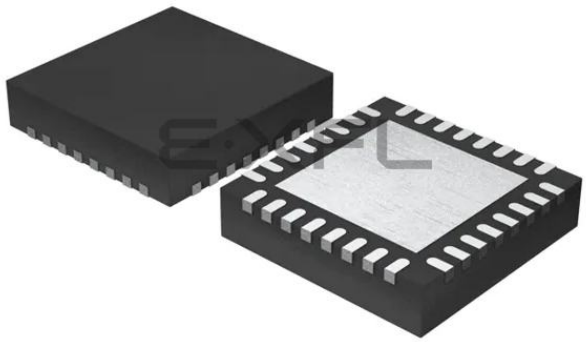


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

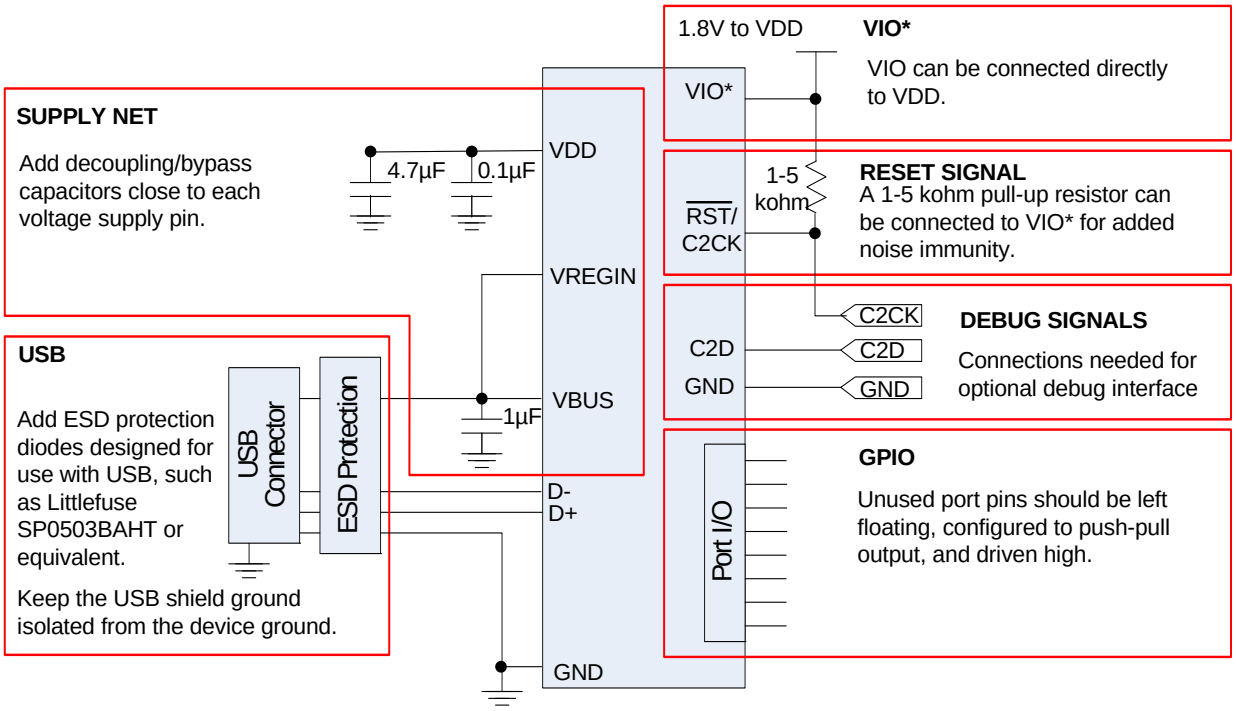
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	48 MIPS
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	24
Program Memory Size	64KB (64K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 21x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051t626-b-gmr

Table of Contents

List of Figures



*Note : VIO only appears on certain package options. If VIO is not present, the $\overline{\text{RST}}$ pullup can connect to VDD.

Figure 1.5. Typical Bus-Powered Connections

8. 10-Bit ADC (ADC0, C8051T620/6/7 and C8051T320/1 Only)

ADC0 on the C8051T620/6/7 and C8051T320/1 is a 500 ksps, 10-bit successive-approximation-register (SAR) ADC with integrated track-and-hold, a gain stage programmable to 1x or 0.5x, and a programmable window detector. The ADC is fully configurable under software control via Special Function Registers. The ADC may be configured to measure various different signals using the analog multiplexer described in Section “8.5. ADC0 Analog Multiplexer (C8051T620/6/7 and C8051T320/1 Only)” on page 55. The voltage reference for the ADC is selected as described in Section “10. Voltage Reference Options” on page 59. The ADC0 subsystem is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.

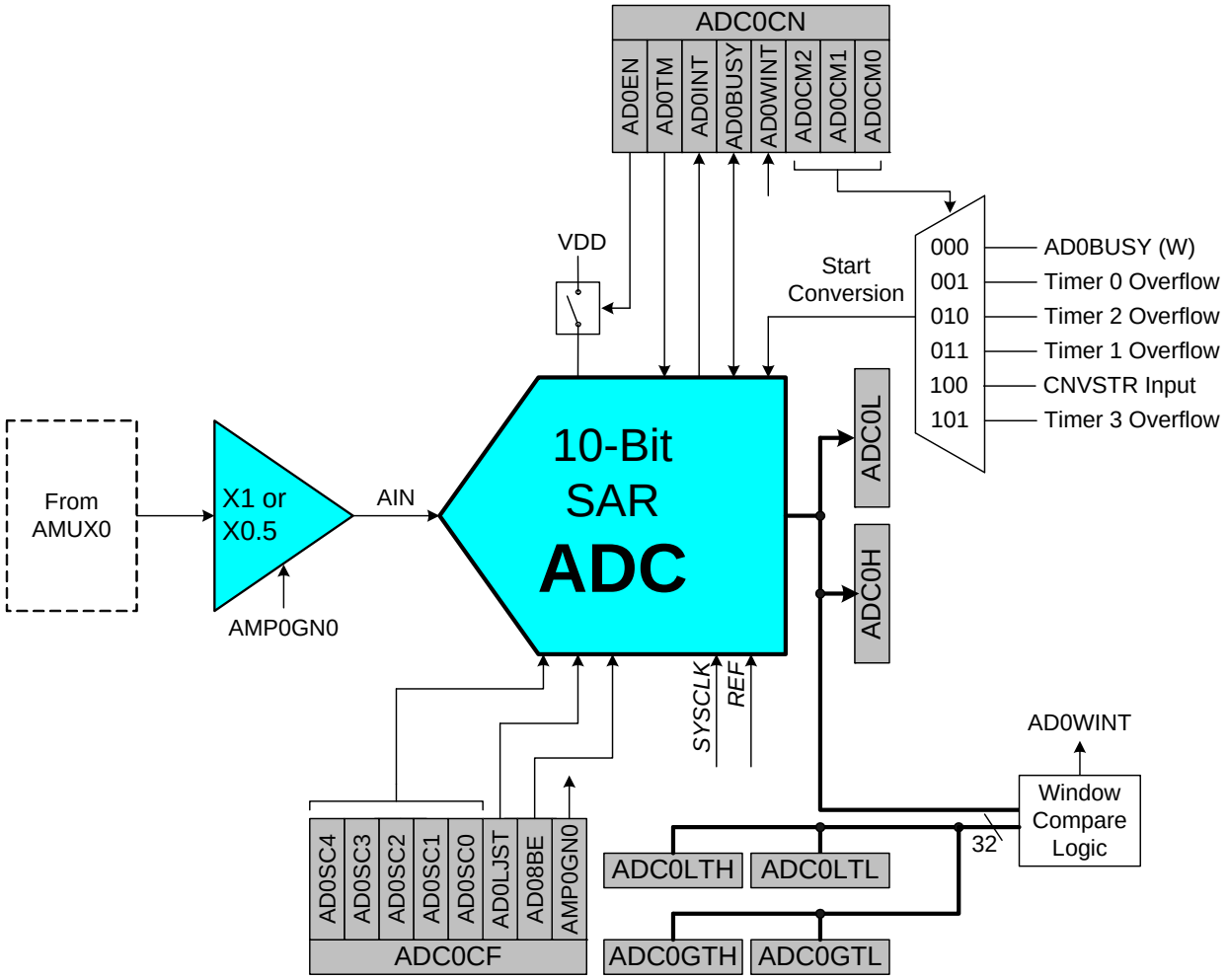


Figure 8.1. ADC0 Functional Block Diagram

9.1. Calibration

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 7.11 on page 41 for specifications). For absolute temperature measurements, offset and/or gain calibration is recommended.

A single-point offset measurement of the temperature sensor is performed on each device during production test. The TOFFH and TOFFL calibration values represent the output of the ADC when reading the temperature sensor at 0 degrees Celsius, and using the internal regulator as a voltage reference. The TOFFH and TOFFL values can be read from EPROM memory and are located at 0x3FFB (TOFFH) and 0x3FFA (TOFFL). The temperature sensor offset information is left-justified, so TOFFH contains the 8 most-significant bits of the calibration value and TOFFL.7-6 contain the 2 least-significant bits of the calibration value, as shown in Figure 9.2. One LSB of this measurement is equivalent to one LSB of the ADC output under the measurement conditions.

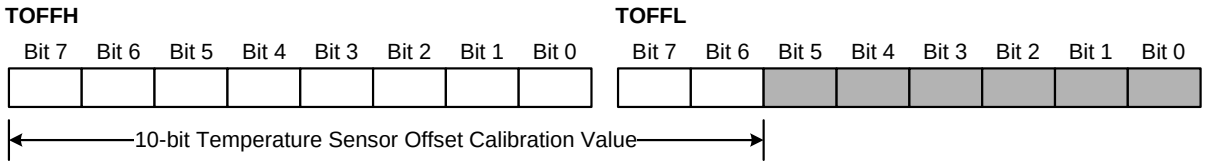


Figure 9.2. TOFFH and TOFFL Calibration Value Orientation

Figure 9.3 shows the typical temperature sensor error assuming a 1-point calibration at 0 °C. **Parameters that affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

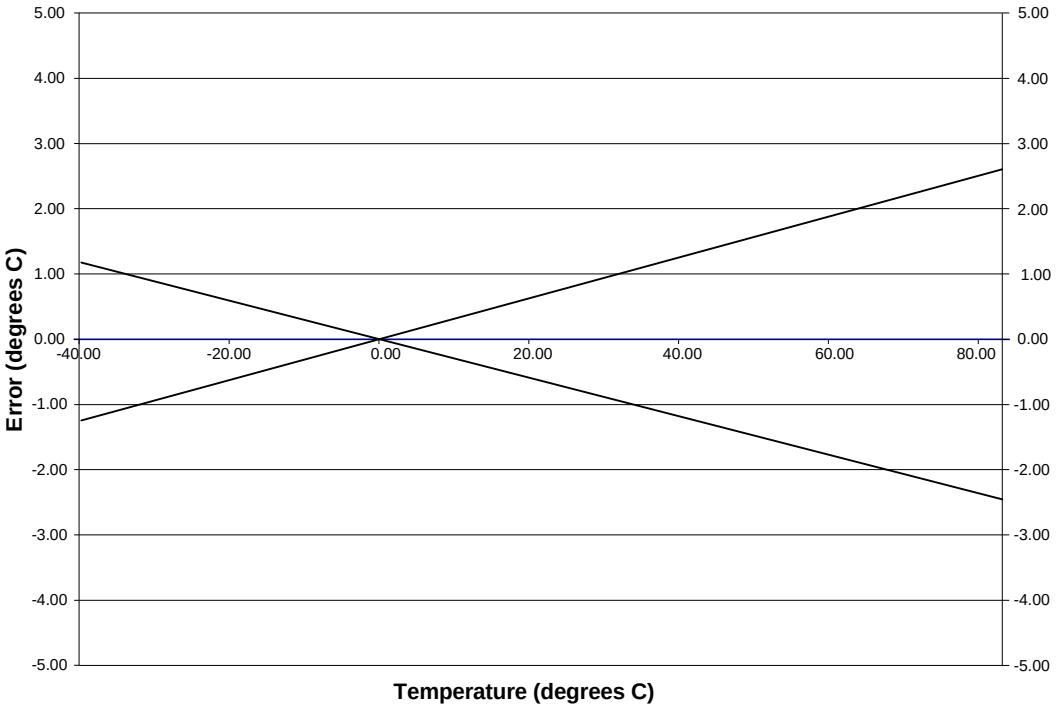


Figure 9.3. Temperature Sensor Error with 1-Point Calibration at 0 Celsius

12. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. 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 MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 30), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 12.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 48 MIPS Peak Throughput with 48 MHz Clock
- 0 to 48 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

Performance

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 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.

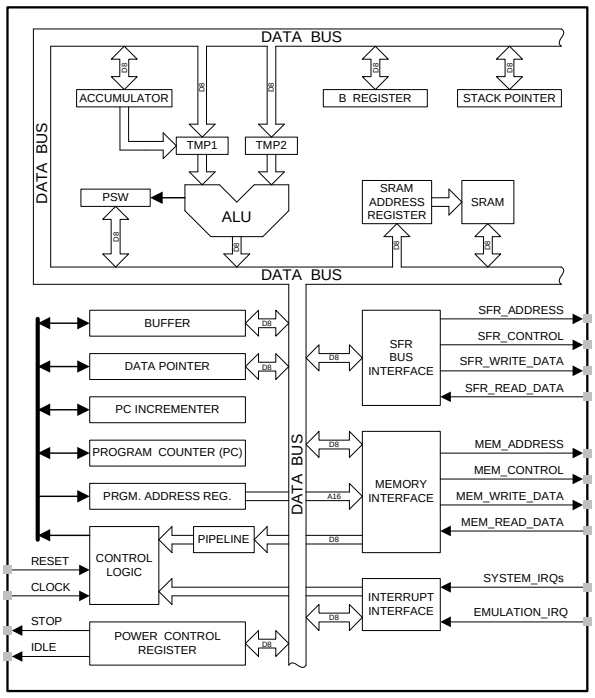


Figure 12.1. CIP-51 Block Diagram

SFR Definition 12.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS[1:0]		OV	F1	PARITY
Type	R/W	R/W	R/W	R/W		R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD0; Bit-Addressable

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[1:0]	**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: • 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.
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.

15.2.2. External RAM

There are 1024 bytes (C8051T620/1/320/1/2/3 devices) or 3072 bytes (C8051T626/7 devices) of on-chip RAM mapped into the external data memory space. All of these address locations may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN as shown in SFR Definition 15.1).

For a 16-bit MOVX operation (@DPTR), the upper three or five bits of the 16-bit external data memory address word are "don't cares" (when USBFAE is cleared to 0). As a result, the XRAM is mapped modulo style over the entire 64 k external data memory address range. For example, on the C8051T620/1 the XRAM byte at address 0x0000 is shadowed at addresses 0x0400, 0x0800, 0x0C00, 0x1000, etc. This is a useful feature when performing a linear memory fill, as the address pointer doesn't have to be reset when reaching the RAM block boundary.

SFR Definition 15.1. EMI0CN: External Memory Interface Control

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

SFR Address = 0xAA

Bit	Name	Function
7:5	Unused	Read = 00000b; Write = Don't Care.
4:0	PGSEL[4:0]	**XRAM Page Select.** The EMI0CN register provides the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. Since the upper (unused) bits of the register are always zero, the PGSEL determines which page of XRAM is accessed. For Example: If EMI0CN = 0x01, addresses 0x0100 through 0x01FF will be accessed. **Note:** PGSEL[4:3] are only valid on the C8051T626/7 devices.

15.2.3. Accessing USB FIFO Space

The C8051T620/1/6/7 & C8051T320/1/2/3 include 1k of RAM which functions as USB FIFO space. Figure 15.4 shows an expanded view of the FIFO space and user XRAM. FIFO space is normally accessed via USB FIFO registers; see Section "23.5. FIFO Management" on page 169 for more information on accessing these FIFOs. The MOVX instruction should not be used to load or modify USB data in the FIFO space.

Unused areas of the USB FIFO space may be used as general purpose XRAM if necessary. The FIFO block operates on the USB clock domain; thus the USB clock must be active when accessing FIFO space. Note that the number of SYSCLK cycles required by the MOVX instruction is increased when accessing USB FIFO space.

To access the FIFO RAM directly using MOVX instructions, the following conditions must be met: (1) the USBFAE bit in register EMI0CF must be set to 1, and (2) the USB clock frequency must be greater than or

Table 16.2. Special Function Registers (Continued)

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

Register	Address	Description	Page
IAPCN	0xF5	In-Application Programming Control	117
IE	0xA8	Interrupt Enable	104
IP	0xB8	Interrupt Priority	105
IT01CF	0xE4	INT0/INT1 Configuration	111
MEMKEY	0xB7	EPROM Memory Lock and Key	116
OSCICL	0xB3	Internal Oscillator Calibration	130
OSCICN	0xB2	Internal Oscillator Control	131
OSCLCN	0x86	Low-Frequency Oscillator Control	133
OSCXCN	0xB1	External Oscillator Control	137
P0	0x80	Port 0 Latch	152
P0MASK	0xAE	Port 0 Mask Configuration	150
P0MAT	0x84	Port 0 Match Configuration	150
P0MDIN	0xF1	Port 0 Input Mode Configuration	153
P0MDOUT	0xA4	Port 0 Output Mode Configuration	153
P0SKIP	0xD4	Port 0 Skip	154
P1	0x90	Port 1 Latch	154
P1MASK	0xBA	Port 1Mask Configuration	151
P1MAT	0xB6	Port 1 Match Configuration	151
P1MDIN	0xF2	Port 1 Input Mode Configuration	155
P1MDOUT	0xA5	Port 1 Output Mode Configuration	155
P1SKIP	0xD5	Port 1 Skip	156
P2	0xA0	Port 2 Latch	156
P2MDIN	0xF3	Port 2 Input Mode Configuration	157
P2MDOUT	0xA6	Port 2 Output Mode Configuration	157
P2SKIP	0xD6	Port 2 Skip	158
P3	0xB0	Port 3 Latch	158
P3MDOUT	0xA7	Port 3 Output Mode Configuration	159
PCA0CN	0xD8	PCA Control	282

Table 16.2. Special Function Registers (Continued)

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

Register	Address	Description	Page
PCA0CPH0	0xFC	PCA Capture 0 High	287
PCA0CPH1	0xEA	PCA Capture 1 High	287
PCA0CPH2	0xEC	PCA Capture 2 High	287
PCA0CPH3	0xEE	PCA Capture 3 High	287
PCA0CPH4	0xFE	PCA Capture 4 High	287
PCA0CPL0	0xFB	PCA Capture 0 Low	287
PCA0CPL1	0xE9	PCA Capture 1 Low	287
PCA0CPL2	0xEB	PCA Capture 2 Low	287
PCA0CPL3	0xED	PCA Capture 3 Low	287
PCA0CPL4	0xFD	PCA Capture 4 Low	287
PCA0CPM0	0xDA	PCA Module 0 Mode Register	285
PCA0CPM1	0xDB	PCA Module 1 Mode Register	285
PCA0CPM2	0xDC	PCA Module 2 Mode Register	285
PCA0CPM3	0xDD	PCA Module 3 Mode Register	285
PCA0CPM4	0xDE	PCA Module 4 Mode Register	285
PCA0H	0xFA	PCA Counter High	286
PCA0L	0xF9	PCA Counter Low	286
PCA0MD	0xD9	PCA Mode	283
PCA0PWM	0xF4	PCA PWM Configuration	284
PCON	0x87	Power Control	120
PFE0CN	0xAF	Prefetch Engine Control	76
PSCTL	0x8F	Program Store R/W Control	116
PSW	0xD0	Program Status Word	75
REF0CN	0xD1	Voltage Reference Control	60
REG01CN	0xC9	Voltage Regulator Control	65
RSTSRC	0xEF	Reset Source Configuration/Status	126
SBCON1	0xAC	UART1 Baud Rate Generator Control	231
SBRLH1	0xB5	UART1 Baud Rate Generator High Byte	231

Table 16.2. Special Function Registers (Continued)

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

Register	Address	Description	Page
SBRLL1	0xB4	UART1 Baud Rate Generator Low Byte	232
SBUF0	0x99	UART0 Data Buffer	221
SBUF1	0xD3	UART1 Data Buffer	230
SCON0	0x98	UART0 Control	220
SCON1	0xD2	UART1 Control	228
SMB0ADM	0xCF	SMBus Slave Address Mask	205
SMB0ADR	0xC7	SMBus Slave Address	204
SMB0CF	0xC1	SMBus Configuration	200
SMB0CN	0xC0	SMBus Control	202
SMB0DAT	0xC2	SMBus Data	206
SMOD1	0xE5	UART1 Mode	229
SP	0x81	Stack Pointer	74
SPI0CFG	0xA1	SPI Configuration	240
SPI0CKR	0xA2	SPI Clock Rate Control	242
SPI0CN	0xF8	SPI Control	241
SPI0DAT	0xA3	SPI Data	242
TCON	0x88	Timer/Counter Control	252
TH0	0x8C	Timer/Counter 0 High	255
TH1	0x8D	Timer/Counter 1 High	255
TL0	0x8A	Timer/Counter 0 Low	254
TL1	0x8B	Timer/Counter 1 Low	254
TMOD	0x89	Timer/Counter Mode	253
TMR2CN	0xC8	Timer/Counter 2 Control	259
TMR2H	0xCD	Timer/Counter 2 High	261
TMR2L	0xCC	Timer/Counter 2 Low	260
TMR2RLH	0xCB	Timer/Counter 2 Reload High	260
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	260
TMR3CN	0x91	Timer/Counter 3Control	265

SFR Definition 17.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA8; Bit-Addressable

Bit	Name	Function
7	EA	**Enable All Interrupts.** Globally enables/disables all interrupts. It overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	**Enable Serial Peripheral Interface (SPI0) Interrupt.** This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	**Enable Timer 2 Interrupt.** This bit sets the masking of the Timer 2 interrupt. 0: Disable Timer 2 interrupt. 1: Enable interrupt requests generated by the TF2L or TF2H flags.
4	ES0	**Enable UART0 Interrupt.** This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	**Enable Timer 1 Interrupt.** This bit sets the masking of the Timer 1 interrupt. 0: Disable all Timer 1 interrupt. 1: Enable interrupt requests generated by the TF1 flag.
2	EX1	**Enable External Interrupt 1.** This bit sets the masking of External Interrupt 1. 0: Disable external interrupt 1. 1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.
1	ET0	**Enable Timer 0 Interrupt.** This bit sets the masking of the Timer 0 interrupt. 0: Disable all Timer 0 interrupt. 1: Enable interrupt requests generated by the TF0 flag.
0	EX0	**Enable External Interrupt 0.** This bit sets the masking of External Interrupt 0. 0: Disable external interrupt 0. 1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

SFR Definition 17.7. IT01CF: $\overline{\text{INT0}}$/$\overline{\text{INT1}}$ Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
Type	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	1

SFR Address = 0xE4

Bit	Name	Function
7	IN1PL	**$\overline{\text{INT1}}$ Polarity.** 0: $\overline{\text{INT1}}$ input is active low. 1: $\overline{\text{INT1}}$ input is active high.
6:4	IN1SL[2:0]	**$\overline{\text{INT1}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7
3	IN0PL	**$\overline{\text{INT0}}$ Polarity.** 0: $\overline{\text{INT0}}$ input is active low. 1: $\overline{\text{INT0}}$ input is active high.
2:0	IN0SL[2:0]	**$\overline{\text{INT0}}$ Port Pin Selection Bits.** These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0 001: Select P0.1 010: Select P0.2 011: Select P0.3 100: Select P0.4 101: Select P0.5 110: Select P0.6 111: Select P0.7

6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

18.3.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (PSCTL.0) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write EPROM bytes.
8. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area.
9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to '0'. Any interrupts posted during the EPROM write operation will be serviced in priority order after the EPROM operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the EPROM write pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write EPROM memory to ensure that a routine called with an illegal address does not result in modification of the EPROM.

18.3.3. 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 an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during EPROM write operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the EPROM operation has completed.

18.4. Program Memory CRC

A CRC engine is included on-chip which provides a means of verifying EPROM contents once the device has been programmed. The CRC engine is available for EPROM verification even if the device is fully read and write locked, allowing for verification of code contents at any time.

The CRC engine is operated through the C2 debug and programming interface, and performs 16-bit CRCs on individual 256-Byte blocks of program memory, or a 32-bit CRC on the entire memory space. To prevent hacking and extrapolation of security-locked source code, the CRC engine will only allow CRCs to be performed on contiguous 256-Byte blocks beginning on 256-Byte boundaries (lowest 8-bits of address are 0x00). For example, the CRC engine can perform a CRC for locations 0x0400 through 0x04FF, but it cannot perform a CRC for locations 0x0401 through 0x0500, or on block sizes smaller or larger than 256 Bytes.

18.4.1. Performing 32-bit CRCs on Full EPROM Content

A 32-bit CRC on the entire EPROM space is initiated by writing to the CRC1 byte over the C2 interface. The CRC calculation begins at address 0x0000, and ends at the end of user EPROM space. The EPBusy bit in register C2ADD will be set during the CRC operation, and cleared once the operation is complete. The 32-bit results will be available in the CRC3-0 registers. CRC3 is the MSB, and CRC0 is the LSB. The polynomial used for the 32-bit CRC calculation is 0x04C11DB7. Note: If a 16-bit CRC has been performed since the last device reset, a device reset should be initiated before performing a 32-bit CRC operation.

18.4.2. Performing 16-bit CRCs on 256-Byte EPROM Blocks

A 16-bit CRC of individual 256-byte blocks of EPROM can be initiated by writing to the CRC0 byte over the C2 interface. The value written to CRC0 is the high byte of the beginning address for the CRC. For example, if CRC0 is written to 0x02, the CRC will be performed on the 256-bytes beginning at address 0x0200, and ending at address 0x2FF. The EPBusy bit in register C2ADD will be set during the CRC operation, and cleared once the operation is complete. The 16-bit results will be available in the CRC1-0 registers. CRC1 is the MSB, and CRC0 is the LSB. The polynomial for the 16-bit CRC calculation is 0x1021.

20.6. 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 “29.4. Watchdog Timer Mode” on page 279; 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.

20.7. EPROM Error Reset

If an EPROM program read or write targets an illegal address, a system reset is generated. This may occur due to any of the following:

- Programming hardware attempts to write or read an EPROM location which is above the user code space address limit.
- An EPROM read from firmware is attempted above user code space. This occurs when a MOVC operation is attempted above the user code space address limit.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above the user code space address limit.

The MEMERR bit (RSTSRC.6) is set following an EPROM error reset. The state of the RST pin is unaffected by this reset.

20.8. 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.

20.9. USB Reset

Writing 1 to the USBRSF bit in register RSTSRC selects USB0 as a reset source. With USB0 selected as a reset source, a system reset will be generated when either of the following occur:

1. RESET signaling is detected on the USB network. The USB Function Controller (USB0) must be enabled for RESET signaling to be detected. See Section “23. Universal Serial Bus Controller (USB0)” on page 160 for information on the USB Function Controller.
2. A falling or rising voltage on the VBUS pin matches the edge polarity selected by the VBPOL bit in register REG01CN. See Section “11. Voltage Regulators (REG0 and REG1)” on page 61 for details on the VBUS detection circuit.

The USBRSF bit will read 1 following a USB reset. The state of the /RST pin is unaffected by this reset.

SFR Definition 22.3. XBR2: Port I/O Crossbar Register 2

Bit	7	6	5	4	3	2	1	0
Name							Reserved	URT1E
Type	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE3

Bit	Name	Function
7:2	Unused	Read = 0000000b; Write = Don't Care.
1	Reserved	Must write 0.
0	URT1E	**UART1 I/O Output Enable Bit.** 0: UART1 I/O unavailable at Port pins. 1: UART1 TX1, RX1 routed to Port pins.

22.5. Port Match

Port match functionality allows system events to be triggered by a logic value change on P0 or P1. A software controlled value stored in the PnMATCH registers specifies the expected or normal logic values of P0 and P1. A Port mismatch event occurs if the logic levels of the Port's input pins no longer match the software controlled value. This allows Software to be notified if a certain change or pattern occurs on P0 or P1 input pins regardless of the XBRn settings.

The PnMASK registers can be used to individually select which P0 and P1 pins should be compared against the PnMATCH registers. A Port mismatch event is generated if (P0 & P0MASK) does not equal (P0MATCH & P0MASK) or if (P1 & P1MASK) does not equal (P1MATCH & P1MASK).

A Port mismatch event may be used to generate an interrupt or wake the device from a low power mode, such as IDLE or SUSPEND. See the Interrupts and Power Options chapters for more details on interrupt and wake-up sources.

24.4.4. 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.

SFR Definition 24.5. SMB0DAT: SMBus Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC2

Bit	Name	Function
7:0	SMB0DAT[7:0]	**SMBus Data.** The SMB0DAT register contains a byte of data to be transmitted on the SMBus serial interface or a byte that has just been received on the SMBus serial interface. The CPU can read from or write to this register whenever the SI serial interrupt flag (SMB0CN.0) is set to logic 1. The serial data in the register remains stable as long as the SI flag is set. When the SI flag is not set, the system may be in the process of shifting data in/out and the CPU should not attempt to access this register.

Table 25.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX[2]	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 28.1.
2. X = Don't care.

Table 25.2. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

	Frequency: 22.1184 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from External Osc.	230400	0.00%	96	SYSCLK	XX[2]	1	0xD0
	115200	0.00%	192	SYSCLK	XX	1	0xA0
	57600	0.00%	384	SYSCLK	XX	1	0x40
	28800	0.00%	768	SYSCLK / 12	00	0	0xE0
	14400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	9600	0.00%	2304	SYSCLK / 12	00	0	0xA0
	2400	0.00%	9216	SYSCLK / 48	10	0	0xA0
	1200	0.00%	18432	SYSCLK / 48	10	0	0x40
SYSCLK from Internal Osc.	230400	0.00%	96	EXTCLK / 8	11	0	0xFA
	115200	0.00%	192	EXTCLK / 8	11	0	0xF4
	57600	0.00%	384	EXTCLK / 8	11	0	0xE8
	28800	0.00%	768	EXTCLK / 8	11	0	0xD0
	14400	0.00%	1536	EXTCLK / 8	11	0	0xA0
	9600	0.00%	2304	EXTCLK / 8	11	0	0x70

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 28.1.
2. X = Don't care.

eout intervals for typical system clocks.

Table 29.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL4	Timeout Interval (ms)
12,000,000	255	65.5
12,000,000	128	33.0
12,000,000	32	8.4
24,000,000	255	32.8
24,000,000	128	16.5
24,000,000	32	4.2
1,500,000[2]	255	524.3
1,500,000[2]	128	264.2
1,500,000[2]	32	67.6
32,768	255	24,000
32,768	128	12,093.75
32,768	32	3,093.75

Notes:
1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 8.

29.5. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of the PCA.