

Welcome to [E-XFL.COM](https://www.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	25MHz
Connectivity	SMBus (2-Wire/I ² C), 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 16x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	20-MLP (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f330

C8051F330/1, C8051F330D

1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

Program memory consists of 8k bytes of Flash. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. See Figure 1.5 for the MCU system memory map.

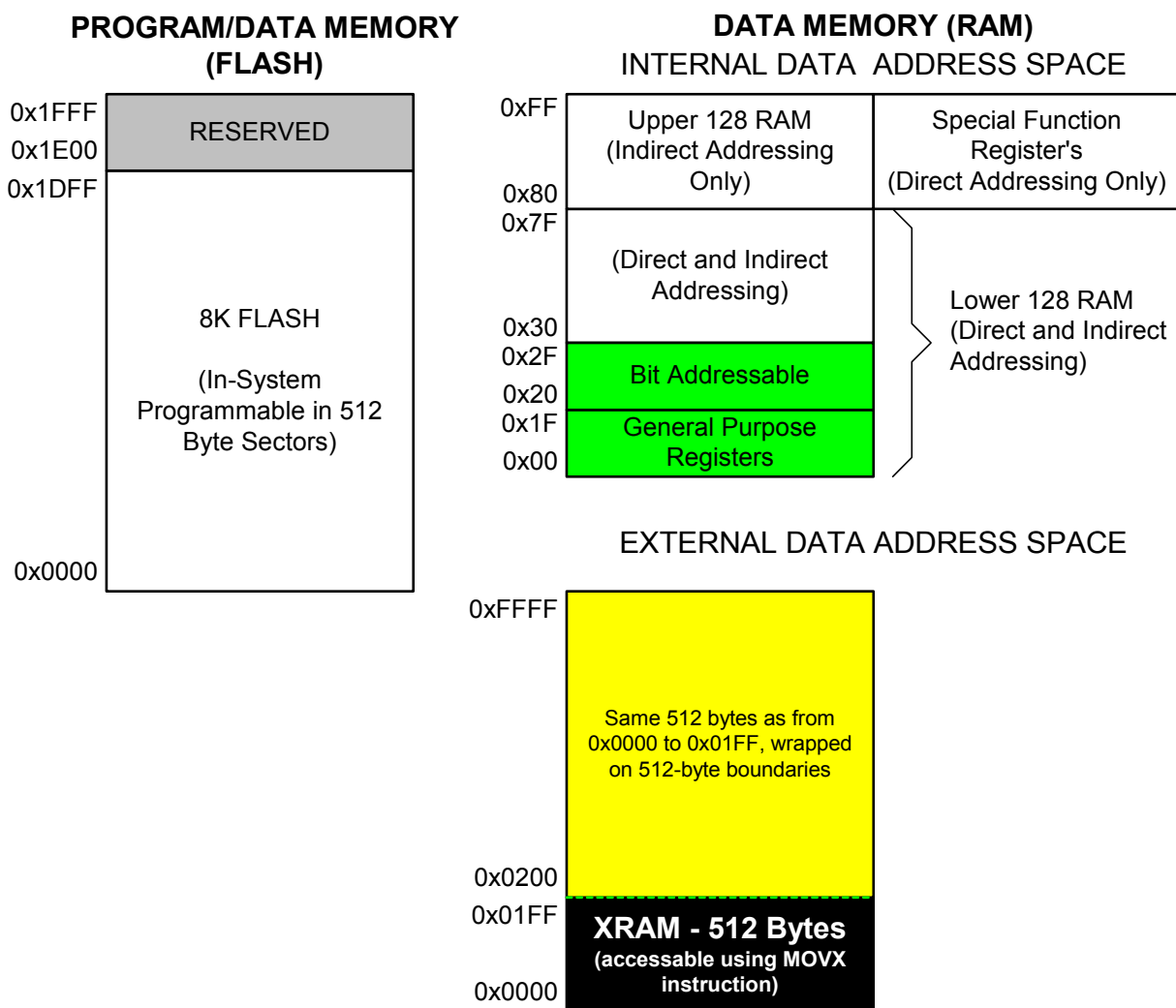


Figure 1.5. On-Board Memory Map

5. 10-Bit ADC (ADC0, C8051F330 and C8051F330D only)

The ADC0 subsystem for the C8051F330 and C8051F330D consists of two analog multiplexers (referred to collectively as AMUX0) with 16 total input selections, and a 200 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector. The AMUX0, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. ADC0 operates in both Single-ended and Differential modes, and may be configured to measure Ports0-1, the Temperature Sensor output, or V_{DD} with respect to Ports0-1 or GND. 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.

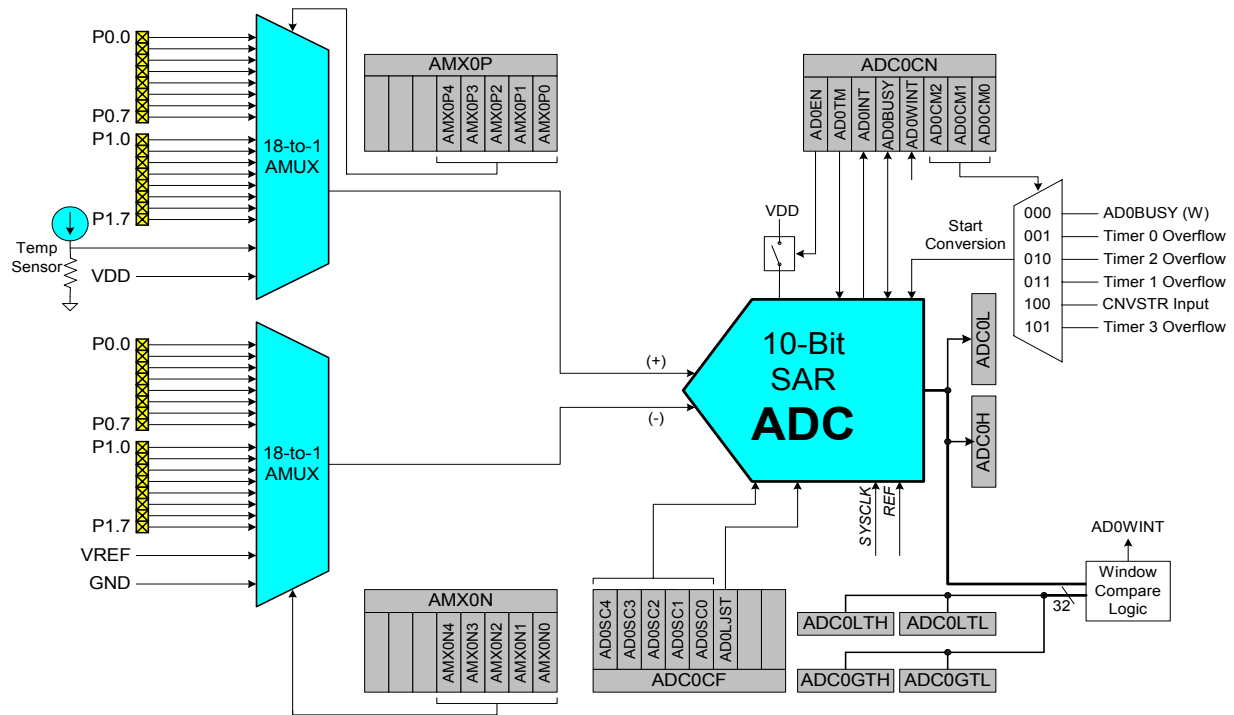


Figure 5.1. ADC0 Functional Block Diagram

5.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

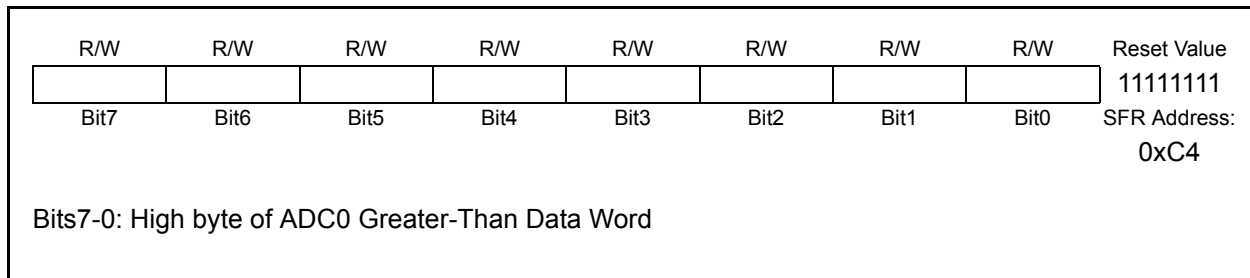


Figure 5.11. ADC0GTH: ADC0 Greater-Than Data High Byte Register

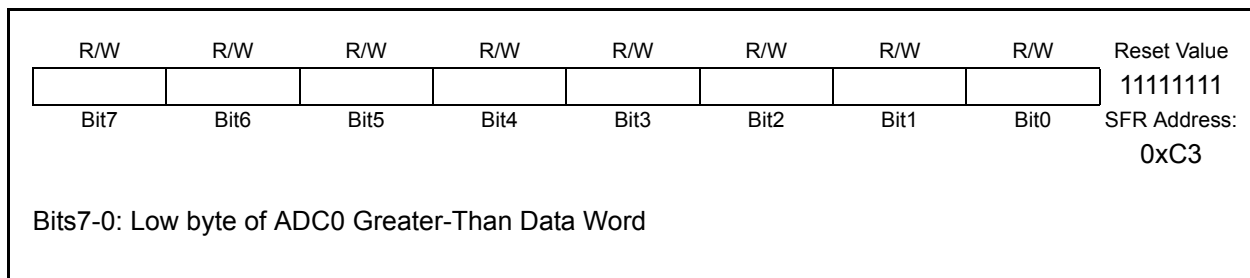


Figure 5.12. ADC0GTL: ADC0 Greater-Than Data Low Byte Register

C8051F330/1, C8051F330D

The Comparator output can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to less than 100 nA. See [Section “14.1. Priority Crossbar Decoder” on page 115](#) for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from -0.25 V to $(V_{DD}) + 0.25\text{ V}$ without damage or upset. The complete Comparator electrical specifications are given in Table 8.1.

The Comparator response time may be configured in software via the CPT0MD register (see Figure 8.5). Selecting a longer response time reduces the Comparator supply current. See Table 8.1 for complete timing and power consumption specifications.

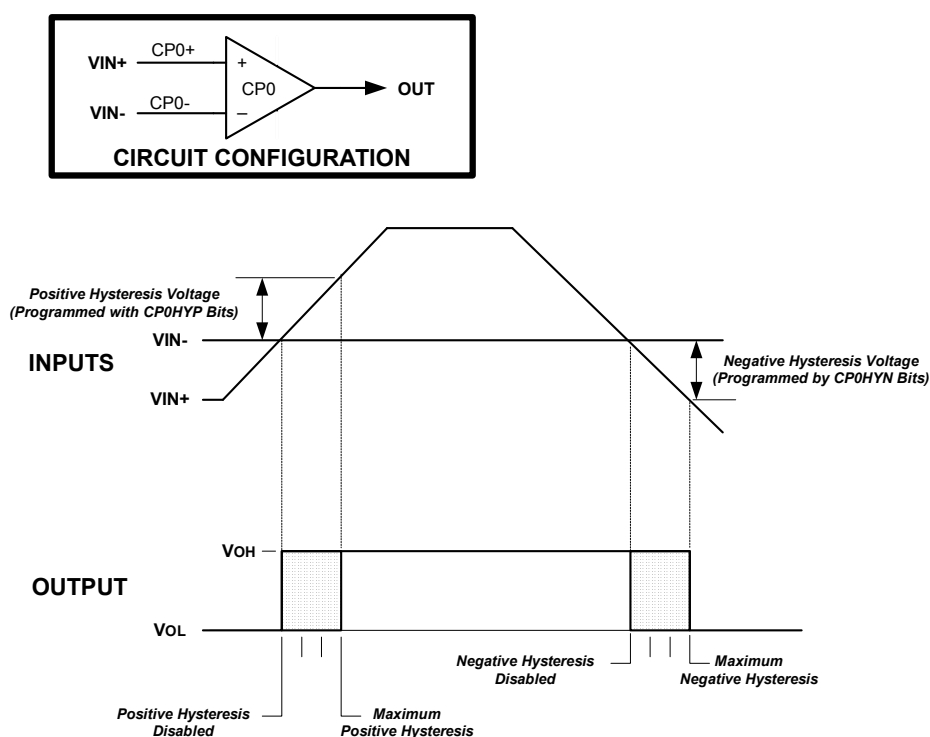


Figure 8.2. Comparator Hysteresis Plot

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

The Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPT0CN (shown in Figure 8.3). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Figure 8.2, settings of 20, 10 or 5 mV of negative hysteresis can be programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see [Section “8.3. Interrupt Handler” on page 58](#)). The CP0FIF flag is set

9.4. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the peripherals and clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not effected). Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. Figure 1.15 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

Although the CIP-51 has Idle and Stop modes built in (as with any standard 8051 architecture), power management of the entire MCU is better accomplished by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off the oscillators lowers power consumption considerably; however a reset is required to restart the MCU.

9.4.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

Note: If the instruction following the write of the IDLE bit is a single-byte instruction and an interrupt occurs during the execution phase of the instruction that sets the IDLE bit, the CPU may not wake from Idle mode when a future interrupt occurs. Therefore, instructions that set the IDLE bit should be followed by an instruction that has 2 or more opcode bytes, for example:

```
// in 'C':
PCON |= 0x01;           // set IDLE bit
PCON = PCON;           // ... followed by a 3-cycle dummy instruction

; in assembly:
ORL PCON, #01h          ; set IDLE bit
MOV PCON, PCON          ; ... followed by a 3-cycle dummy instruction
```

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to [Section “10.6. PCA Watchdog Timer Reset” on page 92](#) for more information on the use and configuration of the WDT.

10.8. 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.

R	R	R/W	R/W	R	R/W	R/W	R	Reset Value
-	FERROR	CORSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SFR Address: 0xEF								
Bit7:	UNUSED. Read = 0. Write = don't care.							
Bit6:	FERROR: Flash Error Indicator. 0: Source of last reset was not a Flash read/write/erase error. 1: Source of last reset was a Flash read/write/erase error.							
Bit5:	CORSEF: Comparator0 Reset Enable and Flag. 0: Read: Source of last reset was not Comparator0. Write: Comparator0 is not a reset source. 1: Read: Source of last reset was Comparator0. Write: Comparator0 is a reset source (active-low).							
Bit4:	SWRSF: Software Reset Force and Flag. 0: Read: Source of last reset was not a write to the SWRSF bit. Write: No Effect. 1: Read: Source of last reset was a write to the SWRSF bit. Write: Forces a system reset.							
Bit3:	WDTRSF: Watchdog Timer Reset Flag. 0: Source of last reset was not a WDT timeout. 1: Source of last reset was a WDT timeout.							
Bit2:	MCDRSF: Missing Clock Detector Flag. 0: Read: Source of last reset was not a Missing Clock Detector timeout. Write: Missing Clock Detector disabled. 1: Read: Source of last reset was a Missing Clock Detector timeout. Write: Missing Clock Detector enabled; triggers a reset if a missing clock condition is detected.							
Bit1:	PORSF: Power-On Reset Force and Flag. This bit is set anytime a power-on reset occurs. Writing this bit enables/disables the V_{DD} monitor as a reset source. Note: writing '1' to this bit before the V_{DD} monitor is enabled and stabilized may cause a system reset. See register VDM0CN (Figure 10.3) 0: Read: Last reset was not a power-on or V_{DD} monitor reset. Write: V_{DD} monitor is not a reset source. 1: Read: Last reset was a power-on or V_{DD} monitor reset; all other reset flags indeterminate. Write: V_{DD} monitor is a reset source.							
Bit0:	PINRSF: HW Pin Reset Flag. 0: Source of last reset was not /RST pin. 1: Source of last reset was /RST pin.							

Figure 10.4. RSTSRC: Reset Source Register

C8051F330/1, C8051F330D

Table 10.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
/RST Output Low Voltage	$I_{OL} = 8.5 \text{ mA}$, $V_{DD} = 2.7 \text{ V to } 3.6 \text{ V}$	—	—	0.6	V
/RST Input High Voltage		$0.7 \times V_{DD}$	—	—	V
/RST Input Low Voltage		—	—	$0.3 \times V_{DD}$	
/RST Input Pullup Current	/RST = 0.0 V	—	25	40	μA
V_{DD} POR Threshold (V_{RST})		2.40	2.55	2.70	V
Missing Clock Detector Time-out	Time from last system clock rising edge to reset initiation	100	220	600	μs
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	5.0	—	—	μs
Minimum /RST Low Time to Generate a System Reset		15	—	—	μs
V_{DD} Monitor Turn-on Time		100	—	—	μs
V_{DD} Monitor Supply Current		—	20	50	μA

C8051F330/1, C8051F330D

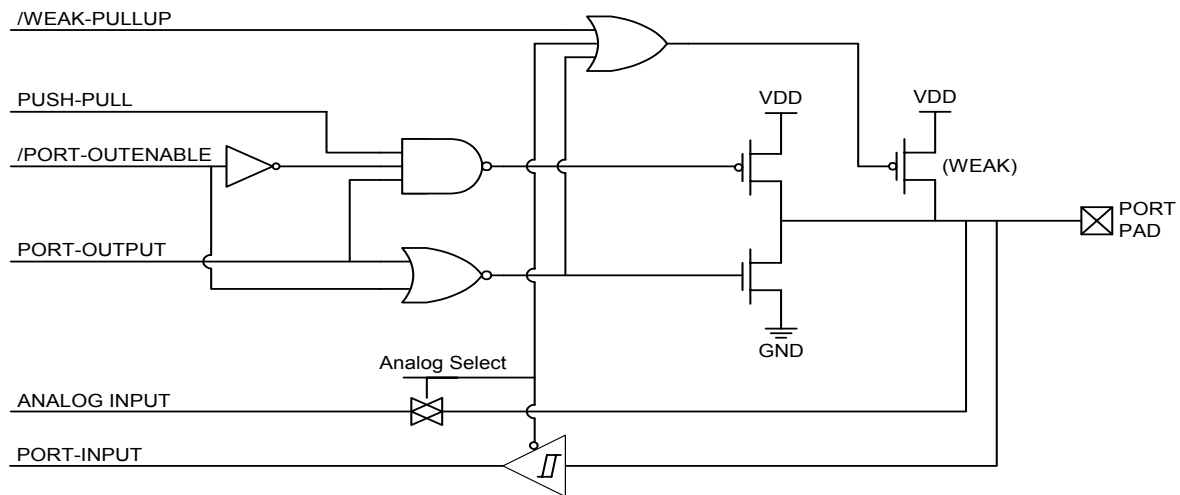


Figure 14.2. Port I/O Cell Block Diagram

C8051F330/1, C8051F330D

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).

Figure 14.9. P0MDOUT: Port0 Output Mode Register

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.

Figure 14.10. P0SKIP: Port0 Skip Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x90

(bit addressable)

Bits7-0: P1.[7:0]
 Write - Output appears on I/O pins per Crossbar Registers.
 0: Logic Low Output.
 1: Logic High Output (high impedance if corresponding P1MDOUT.n bit = 0).
 Read - Always reads '0' if selected as analog input in register P1MDIN. Directly reads Port pin when configured as digital input.
 0: P1.n pin is logic low.
 1: P1.n pin is logic high.

Figure 14.11. P1: Port1 Register

C8051F330/1, C8051F330D

15.1. Supporting Documents

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

1. The I2C-Bus and How to Use It (including specifications), Philips Semiconductor.
2. The I2C-Bus Specification—Version 2.0, Philips Semiconductor.
3. System Management Bus Specification—Version 1.1, SBS Implementers Forum.

15.2. SMBus Configuration

Figure 15.2 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. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pull-up 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.

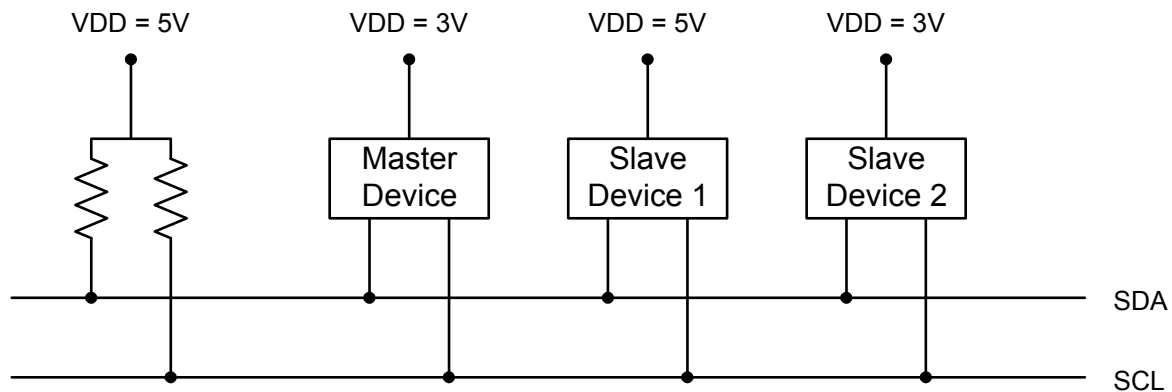


Figure 15.2. Typical SMBus Configuration

C8051F330/1, C8051F330D

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value															
ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS1	SMBCS0	00000000															
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0																
SFR Address: 0xC1																							
Bit7:	ENSMB: SMBus Enable. This bit enables/disables the SMBus interface. When enabled, the interface constantly monitors the SDA and SCL pins. 0: SMBus interface disabled. 1: SMBus interface enabled.																						
Bit6:	INH: SMBus Slave Inhibit. When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected. 0: SMBus Slave Mode enabled. 1: SMBus Slave Mode inhibited.																						
Bit5:	BUSY: SMBus Busy Indicator. This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.																						
Bit4:	EXTHOLD: SMBus Setup and Hold Time Extension Enable. This bit controls the SDA setup and hold times according to . 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.																						
Bit3:	SMBTOE: SMBus SCL Timeout Detection Enable. This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.																						
Bit2:	SMBFTE: SMBus Free Timeout Detection Enable. When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.																						
Bits1-0:	SMBCS1-SMBCS0: SMBus Clock Source Selection. These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 15.1.																						
<table><tr><th>SMBCS1</th><th>SMBCS0</th><th>SMBus Clock Source</th></tr><tr><td>0</td><td>0</td><td>Timer 0 Overflow</td></tr><tr><td>0</td><td>1</td><td>Timer 1 Overflow</td></tr><tr><td>1</td><td>0</td><td>Timer 2 High Byte Overflow</td></tr><tr><td>1</td><td>1</td><td>Timer 2 Low Byte Overflow</td></tr></table>									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
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																					

Figure 15.5. SMB0CF: SMBus Clock/Configuration Register

C8051F330/1, C8051F330D

Table 16.6. Timer Settings for Standard Baud Rates Using an External Oscillator

Frequency: 3.6864 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%	16	SYSCLK	XX ²	1	0xF8
	115200	0.00%	32	SYSCLK	XX	1	0xF0
	57600	0.00%	64	SYSCLK	XX	1	0xE0
	28800	0.00%	128	SYSCLK	XX	1	0xC0
	14400	0.00%	256	SYSCLK	XX	1	0x80
	9600	0.00%	384	SYSCLK	XX	1	0x40
	2400	0.00%	1536	SYSCLK / 12	00	0	0xC0
	1200	0.00%	3072	SYSCLK / 12	00	0	0x80
SYSCLK from Internal Osc.	230400	0.00%	16	EXTCLK / 8	11	0	0xFF
	115200	0.00%	32	EXTCLK / 8	11	0	0xFE
	57600	0.00%	64	EXTCLK / 8	11	0	0xFC
	28800	0.00%	128	EXTCLK / 8	11	0	0xF8
	14400	0.00%	256	EXTCLK / 8	11	0	0xF0
	9600	0.00%	384	EXTCLK / 8	11	0	0xE8

Notes:

1. SCA1-SCA0 and T1M bit definitions can be found in [Section 18.1](#).
2. X = Don't care.

17.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers the contents of its shift register to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 17.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used, and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 17.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin, and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 17.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

18.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 2 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

18.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 18.11, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

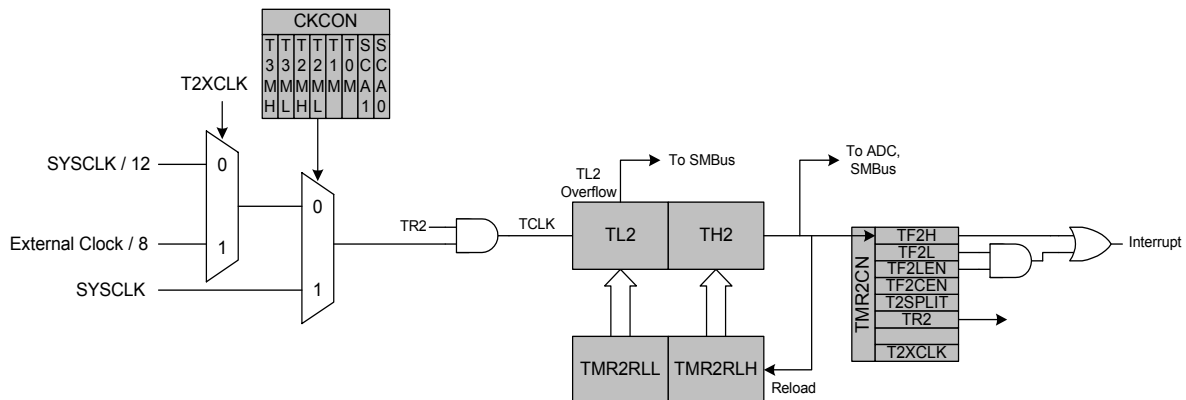


Figure 18.11. Timer 2 16-Bit Mode Block Diagram

C8051F330/1, C8051F330D

18.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TH3 and TL3). Both 8-bit timers operate in auto-reload mode as shown in Figure 18.12. TMR3RLL holds the reload value for TL3; TMR3RLH holds the reload value for TH3. The TR3 bit in TMR3CN handles the run control for TH3. TL3 is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bit (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TH3 Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK	TL3 Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF3H bit is set when TH3 overflows from 0xFF to 0x00; the TF3L bit is set when TL3 overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled (IE.5), an interrupt is generated each time TH3 overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TL3 or TH3 overflows. When TF3LEN is enabled, software must check the TF3H and TF3L flags to determine the source of the Timer 3 interrupt. The TF3H and TF3L interrupt flags are not cleared by hardware and must be manually cleared by software.

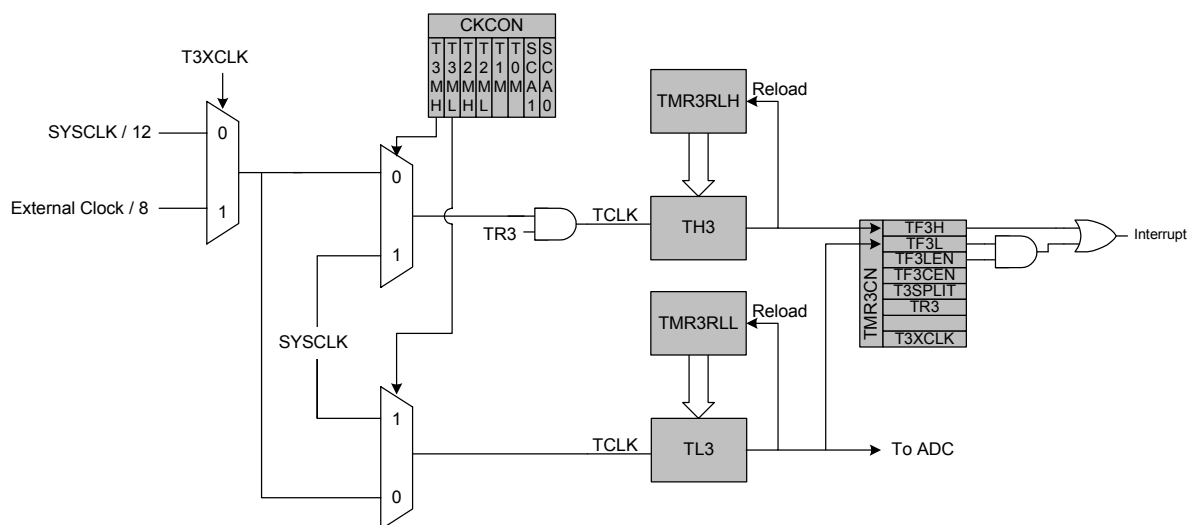


Figure 18.19. Timer 3 8-Bit Mode Block Diagram

C8051F330/1, C8051F330D

NOTES:

C8051F330/1, C8051F330D

CONTACT INFORMATION

Silicon Laboratories Inc.

4635 Boston Lane

Austin, TX 78735

Tel: 1+(512) 416-8500

Fax: 1+(512) 416-9669

Toll Free: 1+(877) 444-3032

Email: productinfo@silabs.com

Internet: www.silabs.com

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders