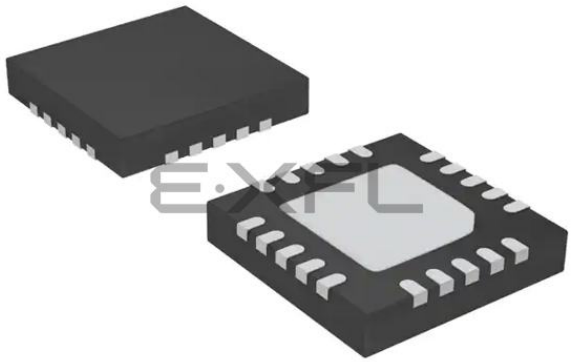


Welcome to E-XFL.COM

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	15
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	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VFQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f854-c-imr

1.2.3. Port I/O Output Drive

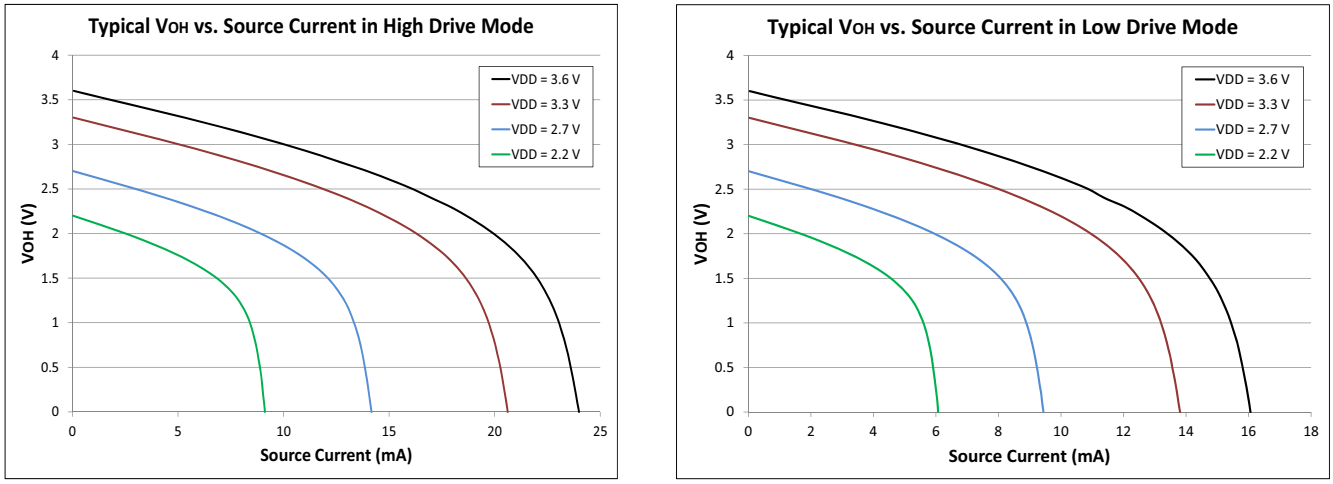


Figure 1.5. Typical V_{OH} vs. Source Current

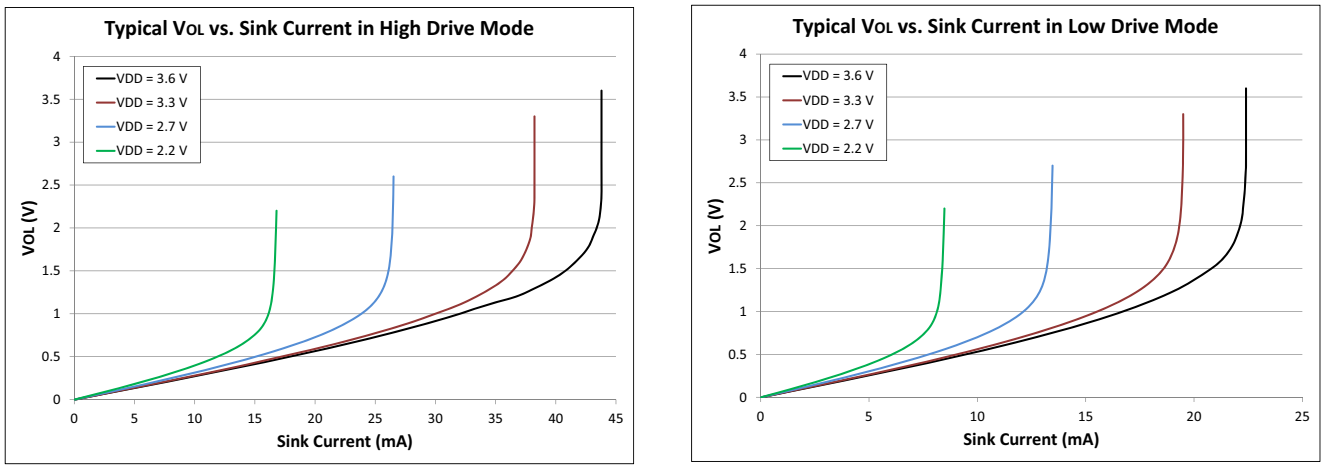


Figure 1.6. Typical V_{OL} vs. Sink Current

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.						

3. Pin Definitions

3.1. C8051F850/1/2/3/4/5 QSOP24 Pin Definitions

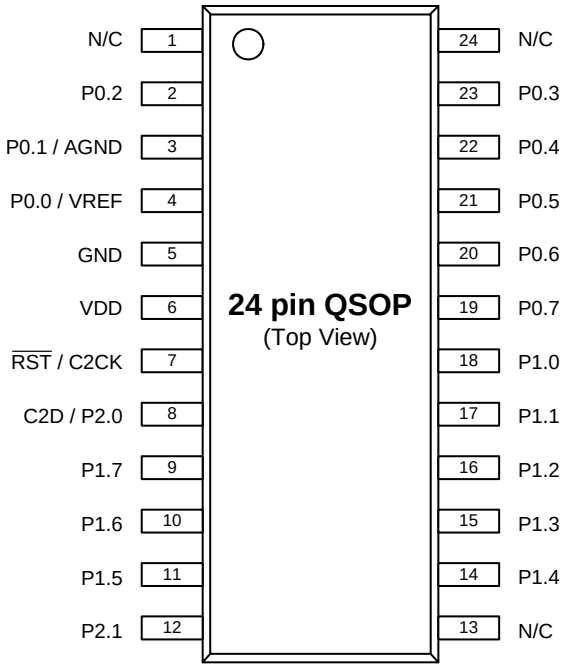


Figure 3.1. C8051F850/1/2/3/4/5-GU and C8051F850/1/2/3/4/5-IU Pinout

Table 3.1. Pin Definitions for C8051F850/1/2/3/4/5-GU and C8051F850/1/2/3/4/5-IU

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
GND	Ground	5			
VDD	Power	6			
$\overline{\text{RST}}$ / C2CK	Active-low Reset / C2 Debug Clock	7			

Register 10.2. FLKEY: Flash Lock and Key

Bit	7	6	5	4	3	2	1	0
Name	FLKEY							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xB7								

Table 10.3. FLKEY Register Bit Descriptions

Bit	Name	Function
7:0	FLKEY	**Flash Lock and Key Register.** Write: This register provides a lock and key function for flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a flash write or erase operation is attempted while these operations are disabled, the flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to flash, it can intentionally lock the flash by writing a non-0xA5 value to FLKEY from software. Read: When read, bits 1-0 indicate the current flash lock state. 00: Flash is write/erase locked. 01: The first key code has been written (0xA5). 10: Flash is unlocked (writes/erases allowed). 11: Flash writes/erases are disabled until the next reset.

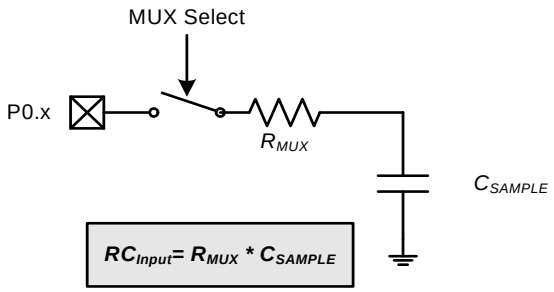
14.1. ADC0 Analog Multiplexer

ADC0 on C8051F85x/86x has an analog multiplexer capable of selecting any pin on ports P0 and P1 (up to 16 total), the on-chip temperature sensor, the internal regulated supply, the VDD supply, or GND. ADC0 input channels are selected using the ADC0MX register.

Table 14.1. ADC0 Input Multiplexer Channels

ADC0MX setting	Signal Name	QSOP24 Pin Name	QFN20 Pin Name	SOIC16 Pin Name
00000	ADC0.0	P0.0	P0.0	P0.0
00001	ADC0.1	P0.1	P0.1	P0.1
00010	ADC0.2	P0.2	P0.2	P0.2
00011	ADC0.3	P0.3	P0.3	P0.3
00100	ADC0.4	P0.4	P0.4	P0.4
00101	ADC0.5	P0.5	P0.5	P0.5
00110	ADC0.6	P0.6	P0.6	P0.6
00111	ADC0.7	P0.7	P0.7	P0.7
01000	ADC0.8	P1.0	P1.0	P1.0
01001	ADC0.9	P1.1	P1.1	P1.1
01010	ADC0.10	P1.2	P1.2	P1.2
01011	ADC0.11	P1.3	P1.3	P1.3
01100	ADC0.12	P1.4	P1.4	Reserved
01101	ADC0.13	P1.5	P1.5	Reserved
01110	ADC0.14	P1.6	P1.6	Reserved
01111	ADC0.15	P1.7	Reserved	Reserved
10000	Temp Sensor	Internal Temperature Sensor		
10001	LDO	Internal 1.8 V LDO Output		
10010	VDD	VDD Supply Pin		
10011	GND	GND Supply Pin		
10100-11111	None	No connection		

n is the ADC resolution in bits (8/10/12).



Note: The value of CSAMPLE depends on the PGA Gain. See electrical specifications for details.

Figure 14.4. ADC0 Equivalent Input Circuits

14.2.5. Gain Setting

The ADC has gain settings of 1x and 0.5x. In 1x mode, the full scale reading of the ADC is determined directly by VREF. In 0.5x mode, the full-scale reading of the ADC occurs when the input voltage is VREF x 2. The 0.5x gain setting can be useful to obtain a higher input voltage range when using a small VREF voltage, or to measure input voltages that are between VREF and VDD. Gain settings for the ADC are controlled by the ADGN bit in register ADC0CF. Note that even with a gain setting of 0.5, voltages above the supply rail cannot be measured directly by the ADC.

14.3. 8-Bit Mode

Setting the ADC08BE bit in register ADC0CF to 1 will put the ADC in 8-bit mode. In 8-bit mode, only the 8 MSBs of data are converted, allowing the conversion to be completed in fewer SAR clock cycles than a 10-bit conversion. The two LSBs of a conversion are always 00 in this mode, and the ADC0L register will always read back 0x00.

14.4. 12-Bit Mode

When configured for 12-bit conversions, the ADC performs four 10-bit conversions using four different reference voltages and combines the results into a single 12-bit value. Unlike simple averaging techniques, this method provides true 12-bit resolution of AC or DC input signals without depending on noise to provide dithering. The converter also employs a hardware dynamic element matching algorithm that reconfigures the largest elements of the internal DAC for each of the four 10-bit conversions. This reconfiguration cancels any matching errors and enables the converter to achieve 12-bit linearity performance to go along with its 12-bit resolution.

The 12-bit mode is enabled by setting the AD12BE bit in register ADC0AC to logic 1 and configuring the ADC in burst mode (ADBMEN = 1) for four or more conversions. The conversion can be initiated using any of the conversion start sources, and the 12-bit result will appear in the ADC0H and ADC0L registers. Since the 12-bit result is formed from a combination of four 10-bit results, the maximum output value is 4 x (1023) = 4092, rather than the max value of (2^12 – 1) = 4095 that is produced by a traditional 12-bit converter. To further increase resolution, the burst mode repeat value may be configured to any multiple of four conversions. For example, if a repeat value of 16 is selected, the ADC0 output will be a 14-bit number (sum of four 12-bit numbers) with 13 effective bits of resolution.

The AD12SM bit in register ADC0TK controls when the ADC will track and sample the input signal. When AD12SM is set to 1, the selected input signal will be tracked before the first conversion of a set and held internally during all four conversions. When AD12SM is cleared to 0, the ADC will track and sample the selected input before each of the four conversions in a set. When maximum throughput (180-200 ksps) is

Register 14.2. ADC0CN1: ADC0 Control 1

Bit	7	6	5	4	3	2	1	0
Name	Reserved							ADCMBE
Type	R							RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xB2								

Table 14.5. ADC0CN1 Register Bit Descriptions

Bit	Name	Function
7:1	Reserved	Must write reset value.
0	ADCMBE	**Common Mode Buffer Enable.** 0: Disable the common mode buffer. This setting should be used only if the tracking time of the signal is greater than 1.5 us. 1: Enable the common mode buffer. This setting should be used in most cases, and will give the best dynamic ADC performance. The common mode buffer must be enabled if signal tracking time is less than or equal to 1.5 us.

Table 15.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
ORL direct, #data	OR immediate to direct byte	3	3

20.4.2. Center Aligned PWM

When configured for center-aligned mode, a module will generate an edge transition at two points for every $2^{(N+1)}$ PCA clock cycles, where N is the selected PWM resolution in bits. In center-aligned mode, these two edges are referred to as the “up” and “down” edges. The polarity at the output pin is selectable, and can be inverted by setting the appropriate channel bit to ‘1’ in the PCA0POL register.

The generated waveforms are centered about the points where the lower N bits of the PCA0 counter are zero. The $(N+1)^{th}$ bit in the PCA0 counter acts as a selection between up and down edges. In 16-bit mode, a special 17th bit is implemented internally for this purpose. At the center point, the (non-inverted) channel output will be low when the $(N+1)^{th}$ bit is ‘0’ and high when the $(N+1)^{th}$ bit is ‘1’, except for cases of 0% and 100% duty cycle. Prior to inversion, an up edge sets the channel to logic high, and a down edge clears the channel to logic low.

Down edges occur when the $(N+1)^{th}$ bit in the PCA0 counter is one, and a logical inversion of the value in the module’s PCA0CPn register matches the main PCA0 counter register for the lowest N bits. For example, with 10-bit PWM, the down edge will occur when the one’s complement of bits 9-0 of the PCA0CPn register match bits 9-0 of the PCA0 counter, and bit 10 of the PCA0 counter is ‘1’.

Up edges occur when the $(N+1)^{th}$ bit in the PCA0 counter is zero, and the lowest N bits of the module’s PCA0CPn register match the value of (PCA0 - 1). For example, with 10-bit PWM, the up edge will occur when bits 9-0 of the PCA0CPn register are one less than bits 9-0 of the PCA0 counter, and bit 10 of the PCA0 counter is ‘0’.

An example of the PWM timing in center-aligned mode for two channels is shown in Figure 20.7. In this example, the CEX0POL and CEX1POL bits are cleared to 0.

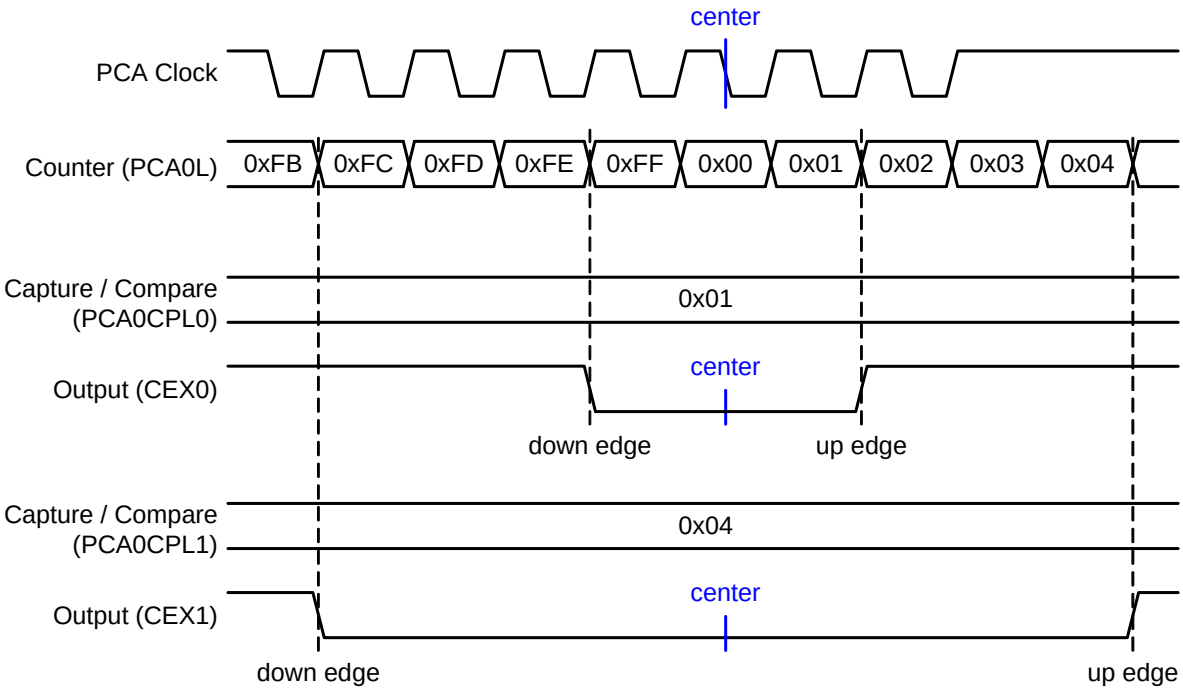


Figure 20.7. Center-Aligned PWM Timing

21.2. Assigning Port I/O Pins to Analog and Digital Functions

Port I/O pins can be assigned to various analog, digital, and external interrupt functions. The port pins assigned to analog functions should be configured for analog I/O, and port pins assigned to digital or external interrupt functions should be configured for digital I/O.

21.2.1. Assigning Port I/O Pins to Analog Functions

Table 21.1 shows all available analog functions that require port I/O assignments. Table 21.1 shows the potential mapping of port I/O to each analog function.

Table 21.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins	SFR(s) used for Assignment
ADC Input	P0.0 - P1.7	ADC0MX, PnSKIP, PnMDIN
Comparator0 Input	P0.0 - P1.7	CPT0MX, PnSKIP, PnMDIN
Comparator1 Input	P0.0 - P1.7	CPT1MX, PnSKIP, PnMDIN
Voltage Reference (VREF)	P0.0	REF0CN, PnSKIP, PnMDIN
Reference Ground (AGND)	P0.1	REF0CN, PnSKIP, PnMDIN

21.2.2. Assigning Port I/O Pins to Digital Functions

Any port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the crossbar for pin assignment; however, some digital functions bypass the crossbar in a manner similar to the analog functions listed above. Table 21.2 shows all digital functions available through the crossbar and the potential mapping of port I/O to each function.

Table 21.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins	SFR(s) Used for Assignment
UART0, SPI0, SMBus0, CP0, CP0A, CP1, CP1A, SYSCLK, PCA0 (CEX0-2 and ECI), T0, T1 or T2.	Any port pin available for assignment by the crossbar. This includes P0.0 - P1.7 pins which have their PnSKIP bit set to ‘0’. **Note:** The crossbar will always assign UART0 pins to P0.4 and P0.5.	XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0 - P2.1	P0SKIP, P1SKIP, P2SKIP

21.4.3. Port Drive Strength

Port drive strength can be controlled on a port-by-port basis using the PRTDRV register. Each port has a bit in PRTDRV to select the high or low drive strength setting for all pins on that port. By default, all ports are configured for high drive strength.

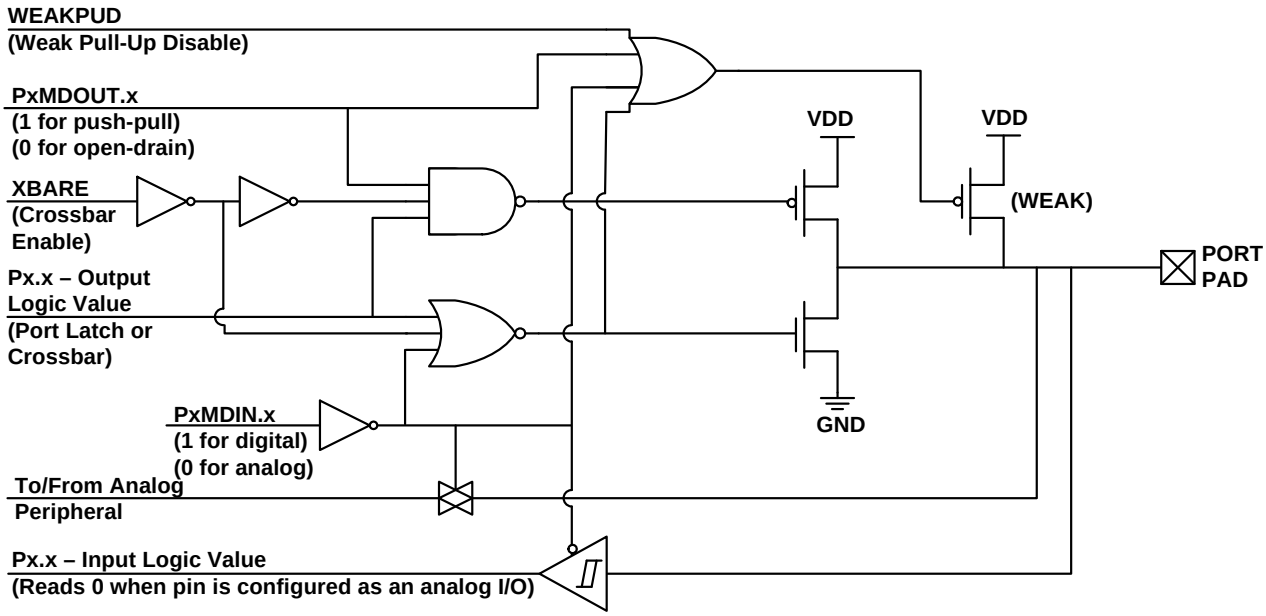


Figure 21.4. Port I/O Cell Block Diagram

21.5. Port Match

Port match functionality allows system events to be triggered by a logic value change on one or more port I/O pins. A software controlled value stored in the PnMATCH registers specifies the expected or normal logic values of the associated port pins (for example, P0MATCH.0 would correspond to P0.0). 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 the input pins regardless of the XBRn settings.

The PnMASK registers can be used to individually select which pins should be compared against the PnMATCH registers. A port mismatch event is generated if (Pn & PnMASK) does not equal (PnMATCH & PnMASK) for all ports with a PnMAT and PnMASK register.

A port mismatch event may be used to generate an interrupt or wake the device from idle mode. See the interrupts and power options chapters for more details on interrupt and wake-up sources.

21.6. Direct Read/Write Access to Port I/O Pins

All port I/O are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable. When writing to a port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the crossbar, the port register can always read its corresponding port I/O pin). The exception to this is the execution of the read-modify-write instructions that target a Port Latch register as the destination. The read-modify-write instructions when operating on a port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a port SFR. For these instructions, the value of the latch register (not the pin) is read, modified, and written back to the SFR.

Register 21.16. P1SKIP: Port 1 Skip

Bit	7	6	5	4	3	2	1	0
Name	P1SKIP							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD5								

Table 21.19. P1SKIP Register Bit Descriptions

Bit	Name	Function
7:0	P1SKIP	**Port 1 Skip.** These bits select port pins to be skipped by the crossbar decoder. Port pins used for analog, special functions or GPIO should be skipped. 0: Corresponding P1.x pin is not skipped by the crossbar. 1: Corresponding P1.x pin is skipped by the crossbar.
Note: Port 1 consists of 8 bits (P1.0-P1.7) on QSOP24 packages and 7 bits (P1.0-P1.6) on QFN20 packages and 4 bits (P1.0-P1.3) on SOIC16 packages.		

22. Reset Sources and Supply Monitor

Reset circuitry allows the controller to be easily placed in a predefined default condition. Upon entering this reset state, the following events occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External port pins are placed in a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain, low-drive mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state. Note that during a power-on event, there may be a short delay before the POR circuitry fires and the $\overline{RST}$ pin is driven low. During that time, the $\overline{RST}$ pin will be weakly pulled to the V_{DD} supply pin.

On exit from the reset state, the program counter (PC) is reset, the Watchdog Timer is enabled and the system clock defaults to the internal oscillator. Program execution begins at location 0x0000.

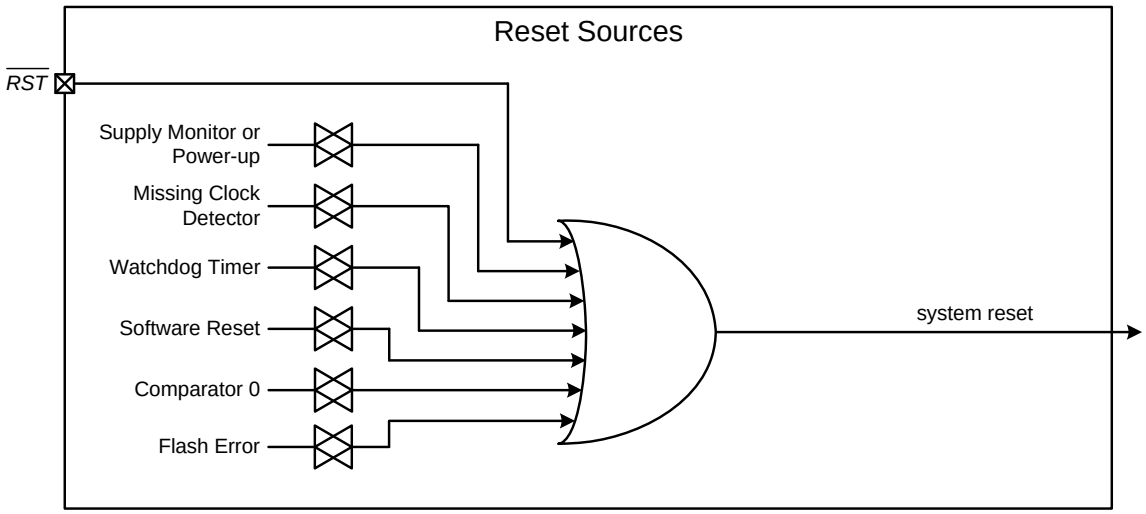


Figure 22.1. Reset Sources

23.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

23.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

23.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

23.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

23.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 23.2, Figure 23.3, and Figure 23.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device.

should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 23.5. For slave mode, the clock and data relationships are shown in Figure 23.6 and Figure 23.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

The SPI0 Clock Rate Register (SPI0CKR) controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

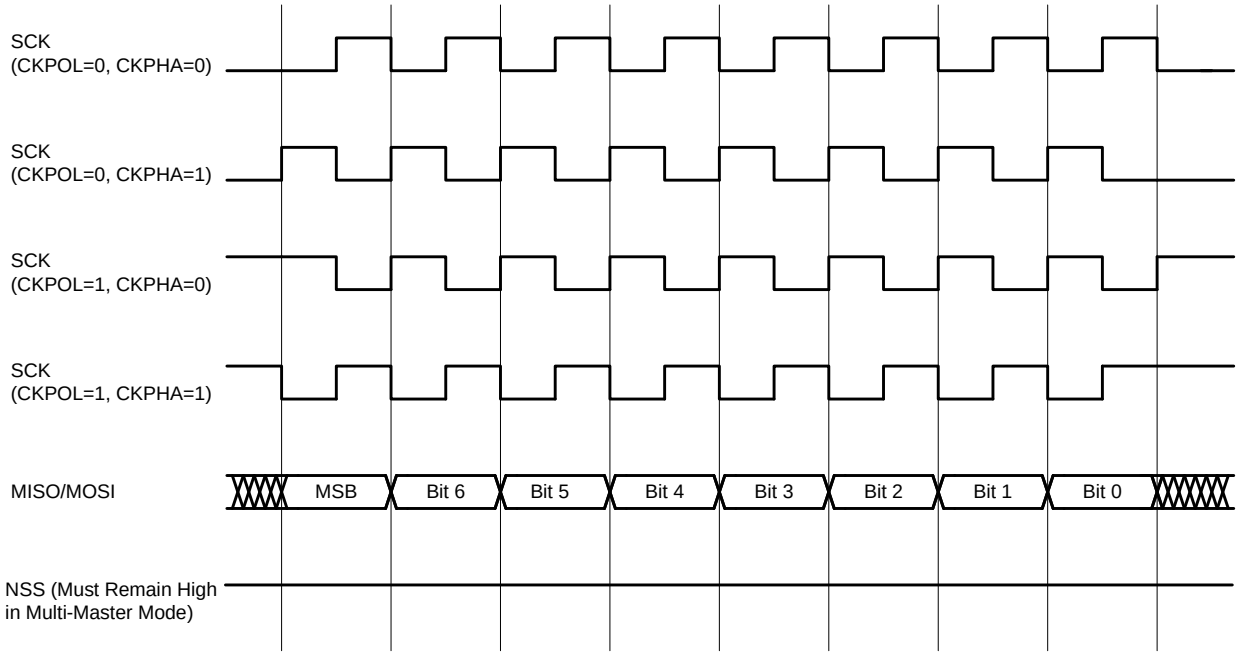


Figure 23.5. Master Mode Data/Clock Timing

Register 24.2. SMB0TC: SMBus0 Timing and Pin Control

Bit	7	6	5	4	3	2	1	0
Name	SWAP	Reserved					SDD	
Type	RW	R					RW	
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xAC								

Table 24.8. SMB0TC Register Bit Descriptions

Bit	Name	Function
7	SWAP	**SMBus0 Swap Pins.** This bit swaps the order of the SMBus0 pins on the crossbar. 0: SDA is mapped to the lower-numbered port pin, and SCL is mapped to the higher-numbered port pin. 1: SCL is mapped to the lower-numbered port pin, and SDA is mapped to the higher-numbered port pin.
6:2	Reserved	Must write reset value.
1:0	SDD	**SMBus0 Start Detection Window.** These bits increase the hold time requirement between SDA falling and SCL falling for START detection. 00: No additional hold time window (0-1 SYSCLK). 01: Increase hold time window to 2-3 SYSCLKs. 10: Increase hold time window to 4-5 SYSCLKs. 11: Increase hold time window to 8-9 SYSCLKs.

Register 24.4. SMB0ADR: SMBus0 Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV							GC
Type	RW							RW
Reset	0	0	0	0	0	0	0	0
SFR Address: 0xD7								

Table 24.10. SMB0ADR Register Bit Descriptions

Bit	Name	Function
7:1	SLV	**SMBus Hardware Slave Address.** Defines the SMBus0 Slave Address(es) for automatic hardware acknowledgement. Only address bits which have a 1 in the corresponding bit position in SLVM are checked against the incoming address. This allows multiple addresses to be recognized.
0	GC	**General Call Address Enable.** When hardware address recognition is enabled (EHACK = 1), this bit will determine whether the General Call Address (0x00) is also recognized by hardware. 0: General Call Address is ignored. 1: General Call Address is recognized.

25.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 in the TCON register is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit enables the timer when either GATE0 in the TMOD register is logic 0 or when the input signal $\overline{INT0}$ is active as defined by bit IN0PL in register IT01CF.

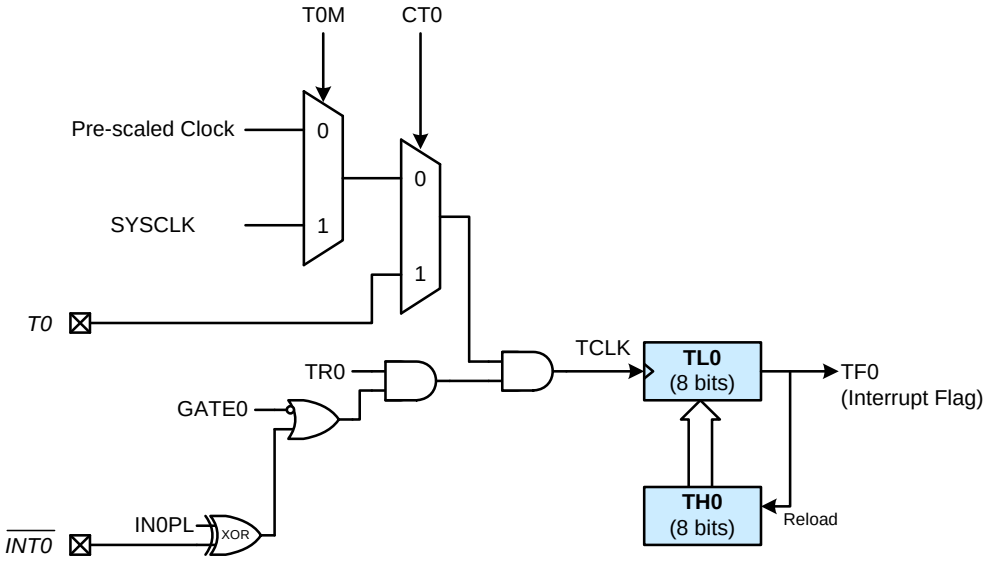


Figure 25.2. T0 Mode 2 Block Diagram

25.2.2. 8-bit Timers with Auto-Reload

When TnSPLIT is set, the timer operates as two 8-bit timers (TMRnH and TMRnL). Both 8-bit timers operate in auto-reload mode as shown in Figure 25.5. TMRnRLL holds the reload value for TMRnL; TMRnRLH holds the reload value for TMRnH. The TRn bit in TMRnCN handles the run control for TMRnH. TMRnL is always running when configured for 8-bit auto-reload mode.

Each 8-bit timer may be configured to clock from SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Clock Select bits (TnMH and TnML in CKCON) select either SYSCLK or the clock defined by the External Clock Select bit (TnXCLK in TMRnCN), as follows:

TnMH	TnXCLK	TMRnH Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

TnML	TnXCLK	TMRnL Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TFnH bit is set when TMRnH overflows from 0xFF to 0x00; the TFnL bit is set when TMRnL overflows from 0xFF to 0x00. When timer interrupts are enabled, an interrupt is generated each time TMRnH overflows. If timer interrupts are enabled and TFnLEN is set, an interrupt is generated each time either TMRnL or TMRnH overflows. When TFnLEN is enabled, software must check the TFnH and TFnL flags to determine the source of the timer interrupt. The TFnH and TFnL interrupt flags are not cleared by hardware and must be manually cleared by software.

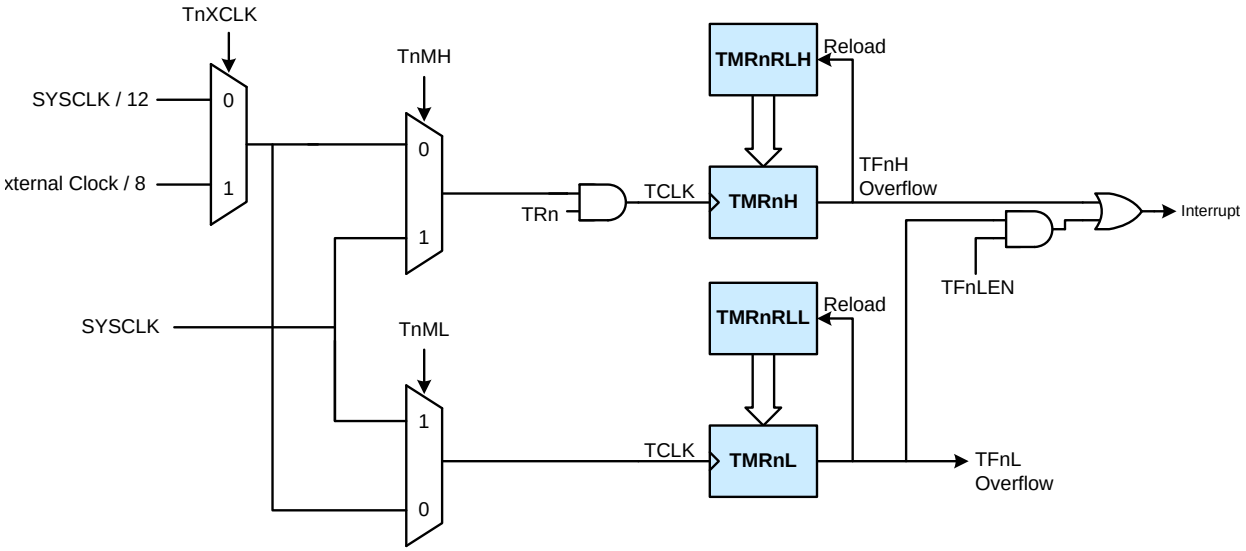


Figure 25.5. 8-Bit Mode Block Diagram

27.5. Watchdog Timer Control Registers

Register 27.1. WDTCN: Watchdog Timer Control

Bit	7	6	5	4	3	2	1	0
Name	WDTCN							
Type	RW							
Reset	0	0	0	1	0	1	1	1
SFR Address: 0x97								

Table 27.1. WDTCN Register Bit Descriptions

Bit	Name	Function
7:0	WDTCN	**WDT Control.** The WDT control field has different behavior for reads and writes. Read: When reading the WDTCN register, the lower three bits (WDTCN[2:0]) indicate the current timeout interval. Bit WDTCN.4 indicates whether the WDT is active (logic 1) or inactive (logic 0). Write: Writing the WDTCN register can set the timeout interval, enable the WDT, disable the WDT, reset the WDT, or lock the WDT to prevent disabling. Writing to WDTCN with the MSB (WDTCN.7) cleared to 0 will set the timeout interval to the value in bits WDTCN[2:0]. Writing 0xA5 both enables and reloads the WDT. Writing 0xDE followed within 4 system clocks by 0xAD disables the WDT. Writing 0xFF locks out the disable feature until the next device reset.