



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	Cap Sense, POR, PWM, Temp Sensor, WDT
Number of I/O	29
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 3x10b
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/c8051f716-gm

C8051F70x/71x

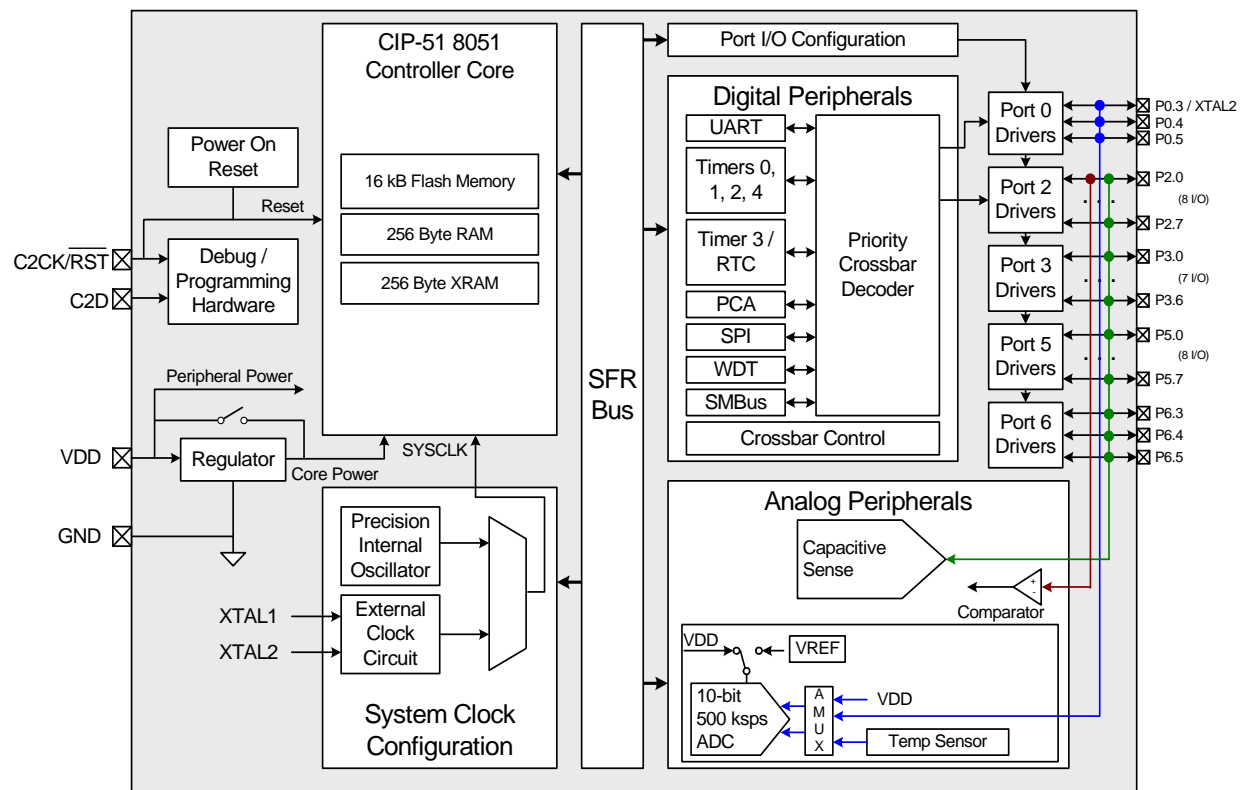


Figure 1.7. C8051F716 Block Diagram

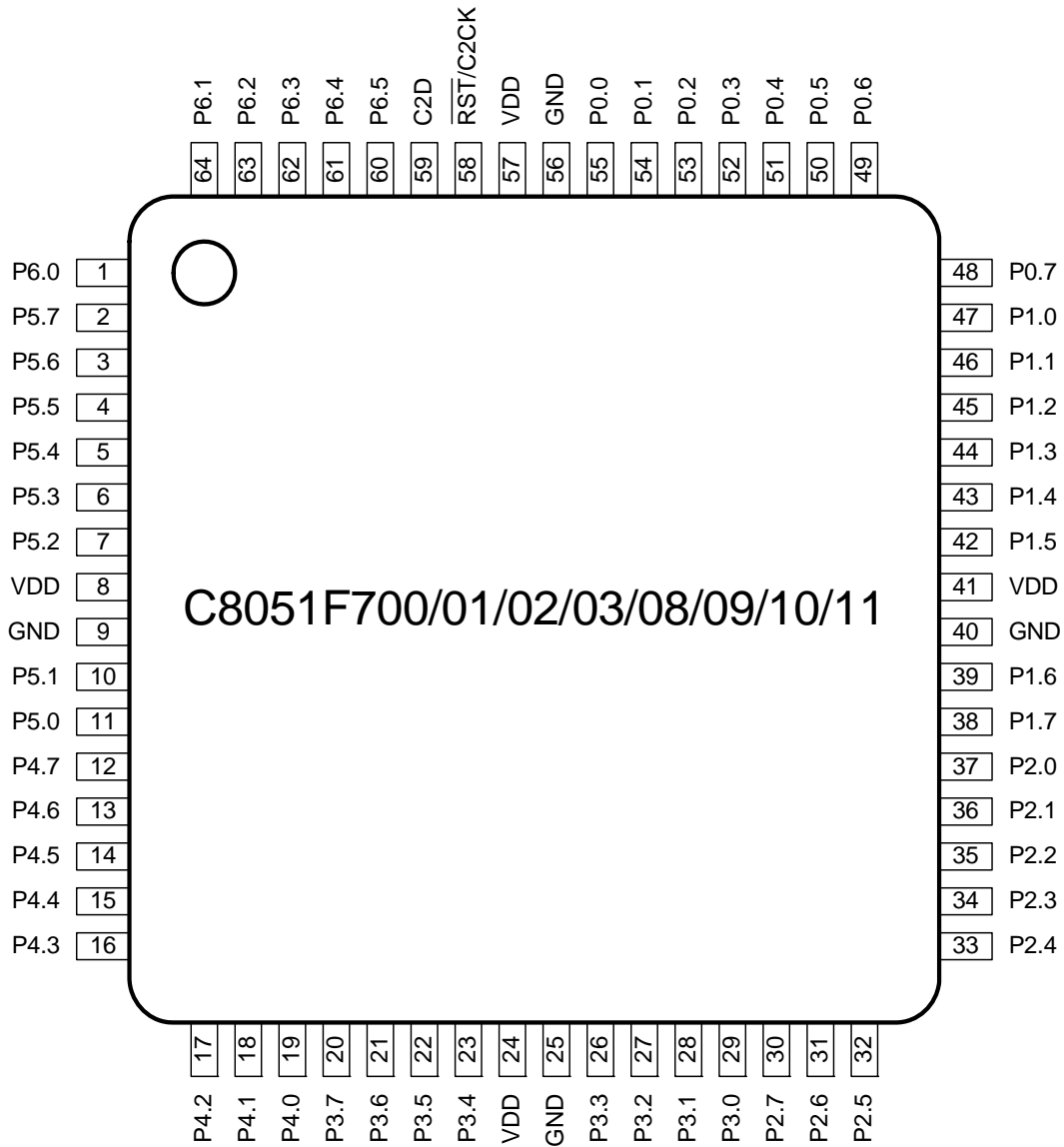


Figure 3.1. C8051F7xx-GQ TQFP64 Pinout Diagram (Top View)

C8051F70x/71x

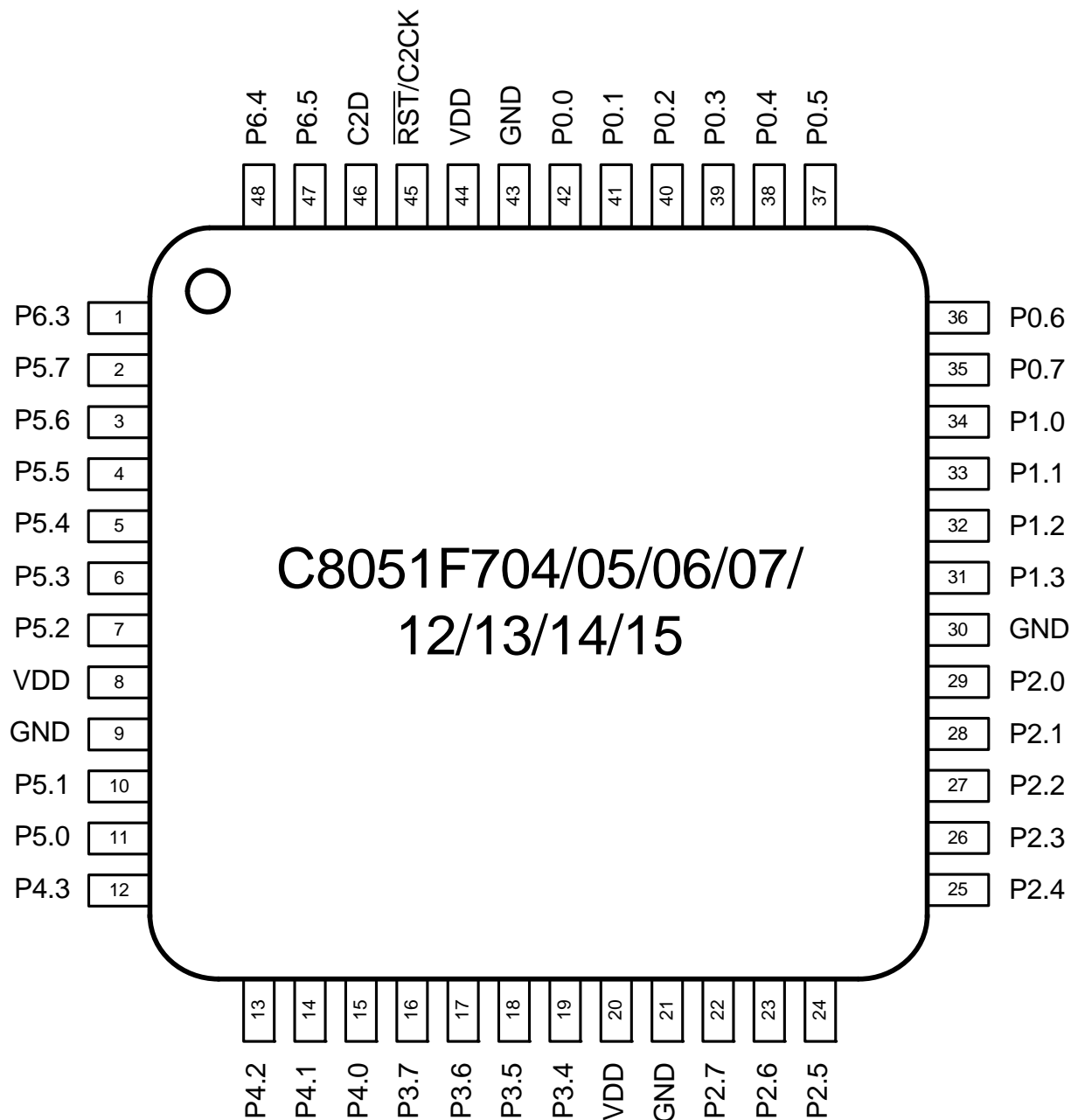


Figure 3.2. C8051F7xx-GQ QFP48 Pinout Diagram (Top View)

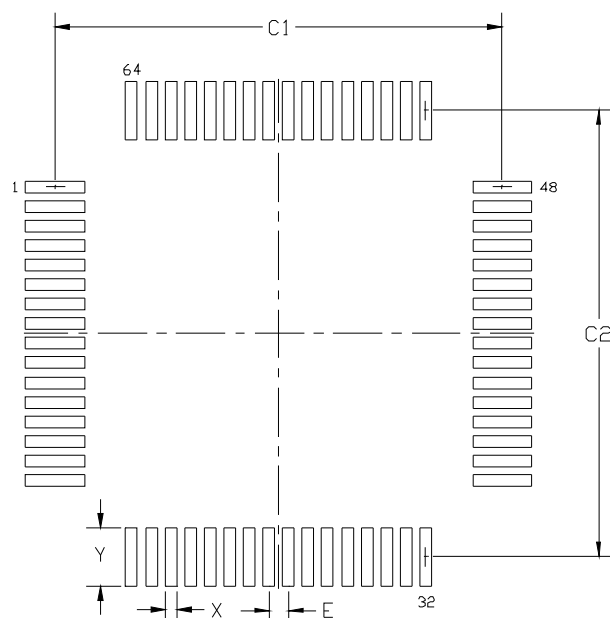


Figure 4.2. TQFP-64 PCB Land Pattern

Table 4.2. TQFP-64 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	11.30	11.40
C2	11.30	11.40
E	0.50 BSC	
X	0.20	0.30
Y	1.40	1.50

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This land pattern design is based on the IPC-7351 guidelines.

Solder Mask Design

3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 μ m minimum, all the way around the pad.

Stencil Design

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pins.

Card Assembly

7. A No-Clean, Type-3 solder paste is recommended.
8. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

12. Voltage and Ground Reference Options

The voltage reference MUX is configurable to use an externally connected voltage reference, the on-chip voltage reference, or one of two power supply voltages (see Figure 12.1). The ground reference MUX allows the ground reference for ADC0 to be selected between the ground pin (GND) or a port pin dedicated to analog ground (P0.1/AGND).

The voltage and ground reference options are configured using the REF0CN SFR described on page 71. Electrical specifications can be found in the Electrical Specifications Chapter.

Important Note About the V_{REF} and AGND Inputs: Port pins are used as the external V_{REF} and AGND inputs. When using an external voltage reference, P0.0/VREF should be configured as an analog input and skipped by the Digital Crossbar. When using AGND as the ground reference to ADC0, P0.1/AGND should be configured as an analog input and skipped by the Digital Crossbar. Refer to Section “28. Port Input/Output” on page 180 for complete Port I/O configuration details. The external reference voltage must be within the range $0 \leq V_{REF} \leq V_{DD}$ and the external ground reference must be at the same DC voltage potential as GND.

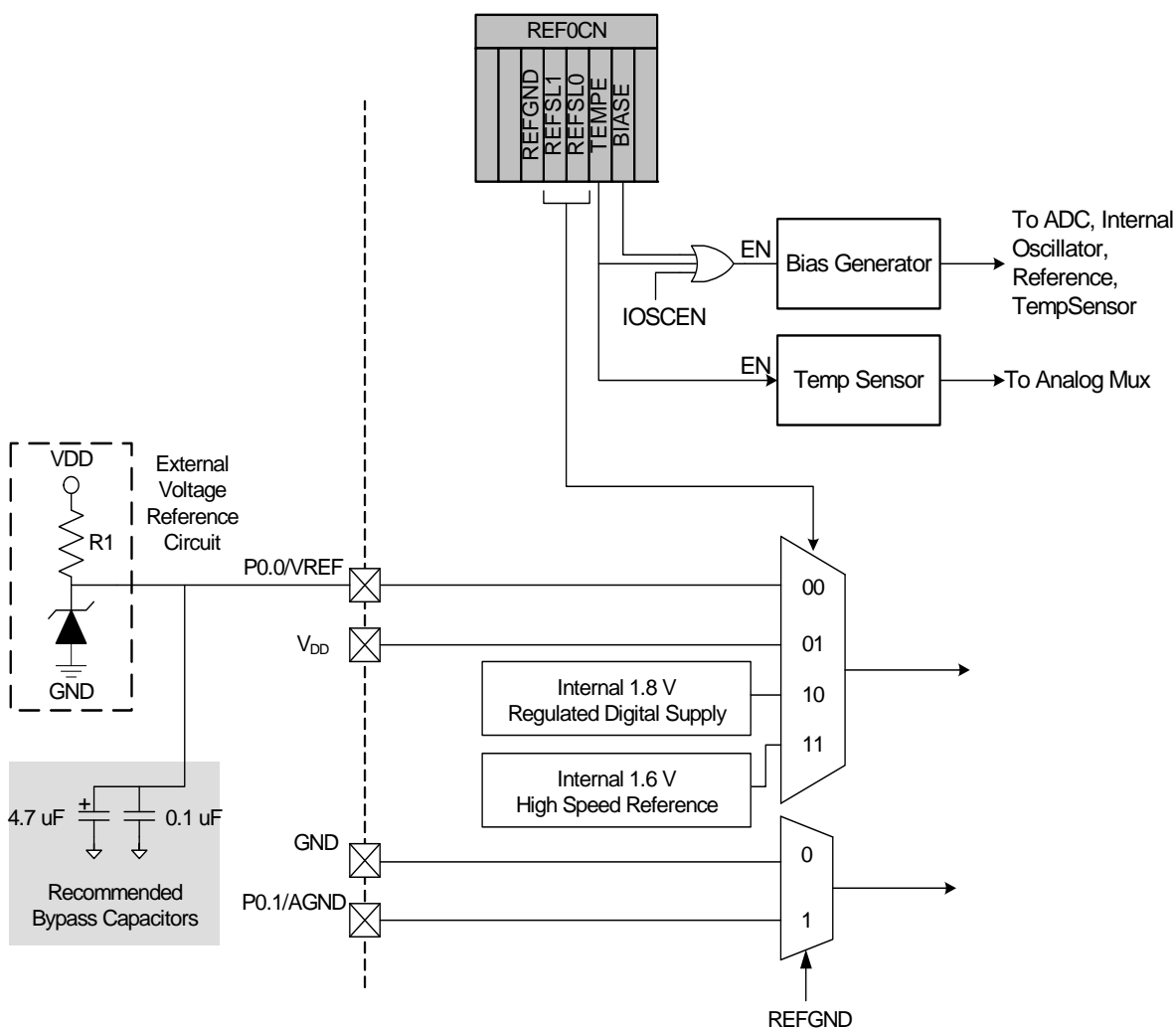
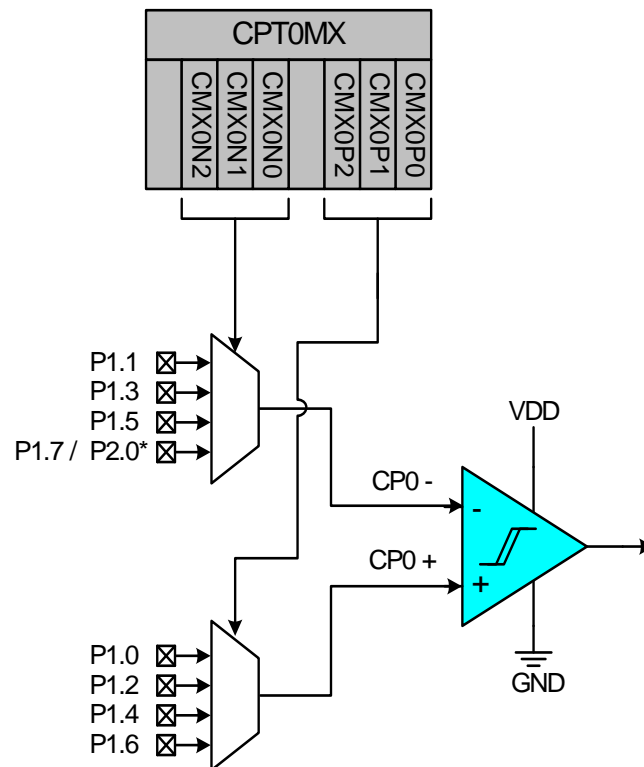


Figure 12.1. Voltage Reference Functional Block Diagram

14.1. Comparator Multiplexer

C8051F70x/71x devices include an analog input multiplexer to connect Port I/O pins to the comparator inputs. The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 14.3). The CMX0P2–CMX0P0 bits select the Comparator0 positive input; the CMX0N2–CMX0N0 bits select the Comparator0 negative input.

Important Note About Comparator Inputs: The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “28.6. Special Function Registers for Accessing and Configuring Port I/O” on page 194).



*P1.7 on 64 and 48-pin devices, P2.0 on 32 and 24-pin devices

Figure 14.3. Comparator Input Multiplexer Block Diagram

15.4. Automatic Scanning

CS0 can be configured to automatically scan a sequence of contiguous CS0 input channels by configuring and enabling auto-scan. Using auto-scan with the CS0 comparator interrupt enabled allows a system to detect a change in measured capacitance without requiring any additional dedicated MCU resources.

Auto-scan is enabled by setting the CS0 start-of-conversion bits (CS0CF6:4) to 111b. After enabling auto-scan, the starting and ending channels should be set to appropriate values in CS0SS and CS0SE, respectively. Writing to CS0SS when auto-scan is enabled will cause the value written to CS0SS to be copied into CS0MX. After being enabled, writing a 1 to CS0BUSY will start auto-scan conversions. When auto-scan completes the number of conversions defined in the CS0 accumulator bits (CS0CF2:0), auto-scan configures CS0MX to the next sequential port pin configured as an analog input and begins a conversion on that channel. The scan sequence continues until CS0MX reaches the ending input channel value defined in CS0SE.

Note: All other CS0 pins configured for analog input with a 0 in the port latch are grounded during the conversion.

After the final channel conversion, auto-scan configures CS0MX back to the starting input channel. For an example system configured to use auto-scan, please see Figure “15.2 Auto-Scan Example” on page 83.

Note: Auto-scan attempts one conversion on a CS0MX channel regardless of whether that channel's port pin has been configured as an analog input. Auto-scan will also complete the current rotation when the device is halted for debugging.

If auto-scan is enabled when the device enters suspend mode, auto-scan will remain enabled and running. This feature allows the device to wake from suspend through CS0 greater-than comparator event on any configured capacitive sense input included in the auto-scan sequence of inputs.

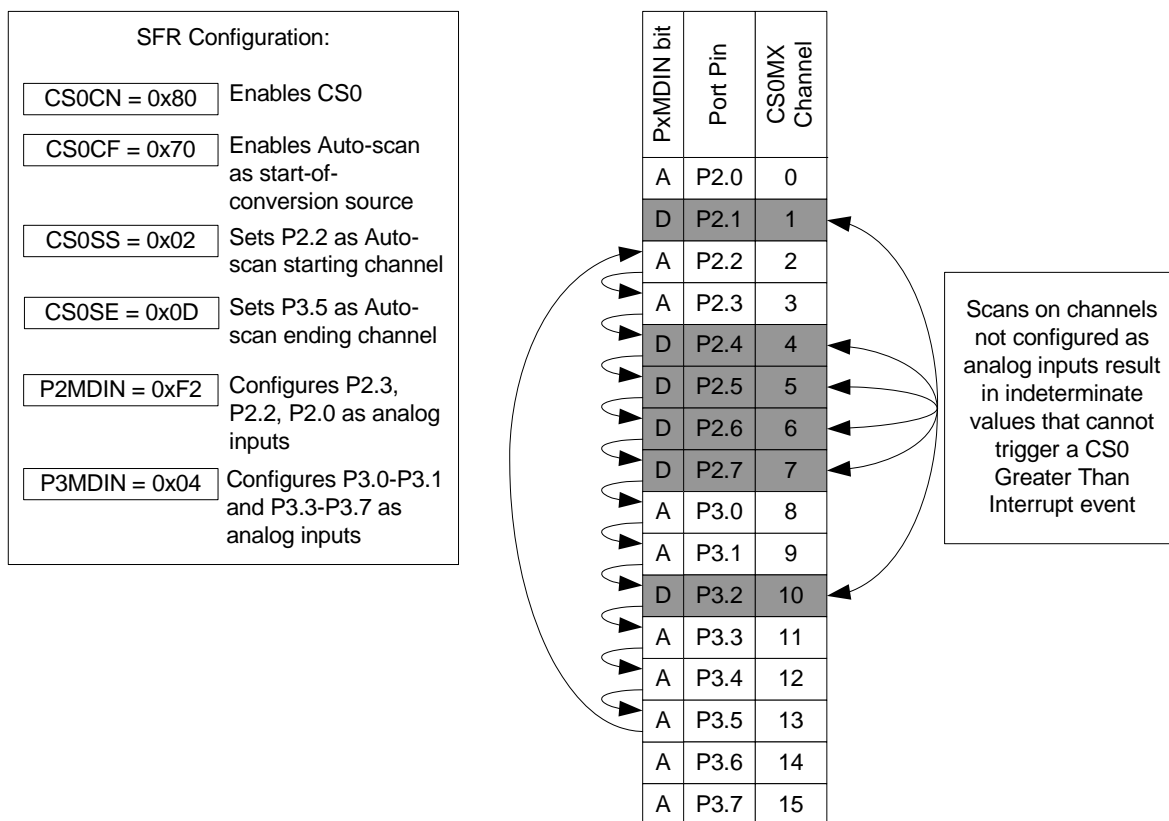


Figure 15.2. Auto-Scan Example

15.9. Capacitive Sense Multiplexer

The input multiplexer can be controlled through two methods. The CS0MX register can be written to through firmware, or the register can be configured automatically using the modules auto-scan functionality (see “15.4. Automatic Scanning”).

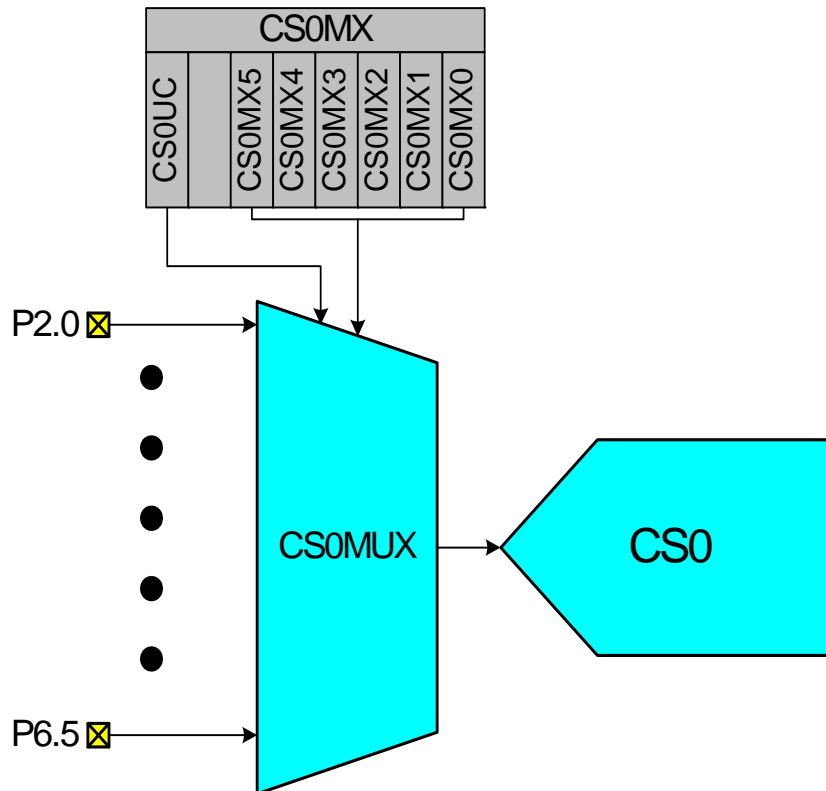


Figure 15.3. CS0 Multiplexer Block Diagram

Table 16.1. 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
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2

18.5.3. Split Mode with Bank Select

When EMI0CF[3:2] are set to 10, the XRAM memory map is split into two areas, on-chip space and off-chip space.

- Effective addresses below the internal XRAM size boundary will access on-chip XRAM space.
- Effective addresses above the internal XRAM size boundary will access off-chip space.
- 8-bit MOVX operations use the contents of EMI0CN to determine whether the memory access is on-chip or off-chip. The upper 8-bits of the Address Bus A[15:8] are determined by EMI0CN, and the lower 8-bits of the Address Bus A[7:0] are determined by R0 or R1. All 16-bits of the Address Bus A[15:0] are driven in "Bank Select" mode.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

18.5.4. External Only

When EMI0CF[3:2] are set to 11, all MOVX operations are directed to off-chip space. On-chip XRAM is not visible to the CPU. This mode is useful for accessing off-chip memory located between 0x0000 and the internal XRAM size boundary.

- 8-bit MOVX operations ignore the contents of EMI0CN. The upper Address bits A[15:8] are not driven (identical behavior to an off-chip access in "Split Mode without Bank Select" described above). This allows the user to manipulate the upper address bits at will by setting the Port state directly. The lower 8-bits of the effective address A[7:0] are determined by the contents of R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine the effective address A[15:0]. The full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

18.6. Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, \overline{RD} and \overline{WR} strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in SFR Definition 18.3, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for \overline{RD} or \overline{WR} pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for \overline{ALE} + 1 for \overline{RD} or \overline{WR} + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 18.1 lists the ac parameters for the External Memory Interface, and Figure 18.4 through Figure 18.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

Table 20.1. Special Function Register (SFR) Memory Map

Addr	SFR Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0 F	SPI0CN	PCA0L P0DRV	PCA0H P1DRV	PCA0CPL0 P2DRV	PCA0CPH0 P3DRV	P4DRV	P5DRV	VDM0CN
F0	0 F	B	P0MDIN	P1MDIN	P0MAT P2MDIN	P0MASK P3MDIN	P4MDIN	P5MDIN	P6MDIN
E8	0 F	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2 DERIVID	PCA0MD	EMI0TC	RSTSRC
E0	0 F	ACC	P1MAT XBR0	P1MASK XBR1	WDTCN	IT01CF		EIE1	EIE2
D8	0 F	PCA0CN	CRC0DATA	PCA0CPM0	PCA0CPM1	PCA0CPM2			
D0	0 F	PSW	EEDATA	REF0CN		P0SKIP	P1SKIP	P2SKIP	
C8	0 F	TMR2CN		TMR2RLL	TMR2RLH	TMR2L	TMR2H	EIP1	EIP2
C0	0 F	SMB0CN	SMB0CF P6DRV	SMB0DAT	ADC0GTL	ADC0GTH HWID	ADC0LTL EECNTL	ADC0LTH EEKEY	EMI0CF
B8	0 F	IP	REG0CN	SMB0ADR	ADC0MX SMB0ADM	ADC0CF	ADC0L CLKSEL	ADC0H CS0MD2	OSCICL
B0	0 F	P3		P6	P5		OSCXCN	EEADDR	FLKEY
A8	0 F	IE	CS0DL OSCICN	CS0DH EMI0CN		P4	CS0MD1 REVID		P3MDOUT
A0	0 F	P2	SPI0CFG PCA0PWM	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	SFRPAGE
98	0 F	SCON0	SBUF0	CS0CN P4MDOUT	CPT0CN P5MDOUT	CS0MX P6MDOUT	CPT0MD	CS0CF	CPT0MX CS0PM
90	0 F	P1	TMR3CN CRC0CN	TMR3RLL CS0SS	TMR3RLH CS0SE	TMR3L CRC0IN	TMR3H CRC0FLIP	CS0THL CRC0AUTO	CS0THH CRC0CNT
88	0 F	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	0 F	P0	SP	DPL	DPH				PCON
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Notes:

1. SFR addresses ending in 0x0 or 0x8 (leftmost column) are bit-addressable.
2. SFRs indicated with bold lettering and shaded cells are available on both SFR Page 0 and F.

22. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during Flash write/erase operations. Refer to Table 9.6 for complete Flash memory electrical characteristics.

22.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “35. C2 Interface” on page 301.

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 programming Flash memory using MOVX, Flash programming operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) 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.

Note: A minimum SYSCLK frequency is required for writing or erasing Flash memory, as detailed in Section “Table 9.6. Flash Electrical Characteristics” on page 50.

For detailed guidelines on programming Flash from firmware, please see Section “22.4. Flash Write and Erase Guidelines” on page 150.

To ensure the integrity of the Flash contents, the on-chip VDD Monitor must be enabled and enabled as a reset source in any system that includes code that writes and/or erases Flash memory from software. Furthermore, there should be no delay between enabling the V_{DD} Monitor and enabling the V_{DD} Monitor as a reset source. Any attempt to write or erase Flash memory while the V_{DD} Monitor is disabled, or not enabled as a reset source, will cause a Flash Error device reset.

22.1.1. 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. The FLKEY register is detailed in SFR Definition 22.2.

22.1.2. Flash Erase Procedure

The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

1. Save current interrupt state and disable interrupts.
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.

C8051F70x/71x

The following guidelines are recommended for any system that contains routines which write or erase Flash from code.

22.4.1. VDD Maintenance and the VDD Monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum VDD rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external VDD brownout circuit to the /RST pin of the device that holds the device in reset until VDD reaches the minimum device operating voltage and re-asserts /RST if VDD drops below the minimum device operating voltage.
3. Keep the on-chip VDD Monitor enabled and enable the VDD Monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Code examples showing this can be found in "AN201: Writing to Flash from Firmware," available from the Silicon Laboratories web site.

Note: On C8051F70x/71x devices, both the VDD Monitor and the VDD Monitor reset source must be enabled to write or erase Flash without generating a Flash Error Device Reset.

On C8051F70x/71x devices, both the VDD Monitor and the VDD Monitor reset source are enabled by hardware after a power-on reset.

4. As an added precaution, explicitly enable the VDD Monitor and enable the VDD Monitor as a reset source inside the functions that write and erase Flash memory. The VDD Monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct, but "RSTSRC |= 0x02" is incorrect.
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 or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

22.4.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write Flash bytes and one routine in code that sets both PSWE and PSEE both to a 1 to erase Flash pages.
8. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in "AN201: Writing to Flash from Firmware," available from the Silicon Laboratories web site.
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 Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the Flash write and erase 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 or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

27.2. Programmable Internal High-Frequency (H-F) Oscillator

All C8051F70x/71x devices include a programmable internal high-frequency oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register as defined by SFR Definition 27.2.

On C8051F70x/71x devices, OSCICL is factory calibrated to obtain a 24.5 MHz base frequency.

The internal oscillator output frequency may be divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN. The divide value defaults to 8 following a reset.

The precision oscillator supports a spread spectrum mode which modulates the output frequency in order to reduce the EMI generated by the system. When enabled (SSE = 1), the oscillator output frequency is modulated by a stepped triangle wave whose frequency is equal to the oscillator frequency divided by 384 (63.8 kHz using the factory calibration). The maximum deviation from the center frequency is $\pm 0.75\%$. The output frequency updates occur every 32 cycles and the step size is typically 0.25% of the center frequency.

SFR Definition 27.2. OSCICL: Internal H-F Oscillator Calibration

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

SFR Address = 0xBF; SFR Page = F

Bit	Name	Function
6:0	OSCICL[7:0]	Internal Oscillator Calibration Bits. These bits determine the internal oscillator period. When set to 00000000b, the H-F oscillator operates at its fastest setting. When set to 11111111b, the H-F oscillator operates at its slowest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

28.1.3. Interfacing Port I/O to 5 V Logic

All Port I/O configured for digital, open-drain operation are capable of interfacing to digital logic operating at a supply voltage up to 2 V higher than VDD and less than 5.25 V. An external pull-up resistor to the higher supply voltage is typically required for most systems.

Important Note: In a multi-voltage interface, the external pull-up resistor should be sized to allow a current of at least 150 μA to flow into the Port pin when the supply voltage is between (VDD + 0.6V) and (VDD + 1.0V). Once the Port pin voltage increases beyond this range, the current flowing into the Port pin is minimal. Figure 28.3 shows the input current characteristics of port pins driven above VDD. The port pin requires 150 μA peak overdrive current when its voltage reaches approximately (VDD + 0.7 V).

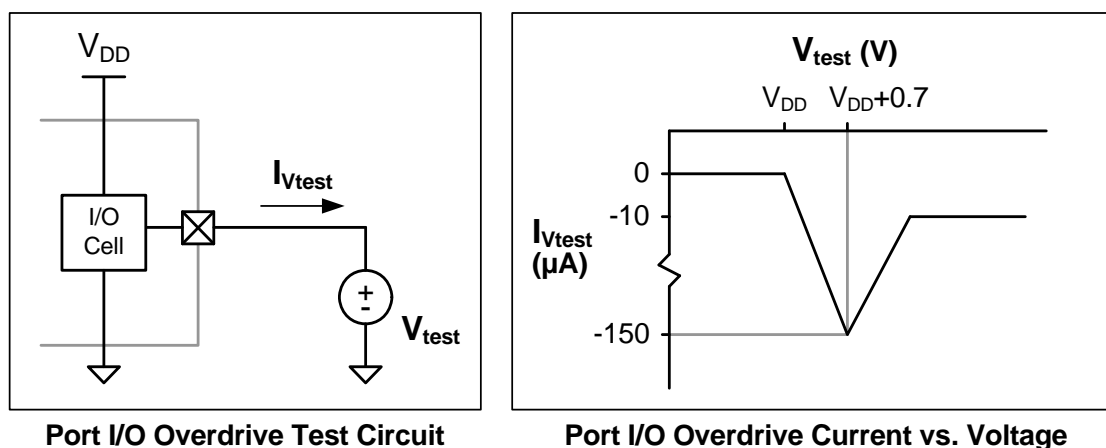


Figure 28.3. Port I/O Overdrive Current

28.1.4. Increasing Port I/O Drive Strength

Port I/O output drivers support a high and low drive strength; the default is low drive strength. The drive strength of a Port I/O can be configured using the PnDRV registers. See Section “9. Electrical Characteristics” on page 47 for the difference in output drive strength between the two modes.

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

Port I/O pins P0.0–P2.7 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.

28.2.1. Assigning Port I/O Pins to Analog Functions

Table 28.1 shows all available analog functions that require Port I/O assignments. **Port pins selected for these analog functions should have their corresponding bit in PnSKIP set to 1.** This reserves the pin for use by the analog function and does not allow it to be claimed by the Crossbar. Table 28.1 shows the potential mapping of Port I/O to each analog function.

28.3. Priority Crossbar Decoder

The Priority Crossbar Decoder assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which is always at pins 4 and 5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. The potential crossbar pin assignments are shown in Figure 28.4. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 28.5 shows an example crossbar configuration with no pins skipped. Figure 28.6 shows the same example with pins P0.2, P0.3 and P1.0 skipped.

If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P0.0 if VREF is used, P0.1 if AGND is used, P0.3 and/or P0.2 if the external oscillator circuit is enabled, P0.6 if the ADC is configured to use the external conversion start signal (CNVSTR), and any selected ADC or Comparator inputs. It is also important to skip any pins that do not exist for the package being used.

Registers XBR0 and XBR1 are used to assign the digital I/O resources to the physical I/O Port pins. When the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL); when the UART is selected, the Crossbar assigns both pins associated with the UART (TX and RX). UART0 pin assignments are fixed for bootloading purposes: UART TX0 is always assigned to P0.4; UART RX0 is always assigned to P0.5. Standard Port I/Os appear contiguously after the prioritized functions have been assigned.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, pending the state of the NSS-MD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

C8051F70x/71x

SFR Definition 28.31. P5MDIN: Port 5 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P5MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF6; SFR Page = F

Bit	Name	Function
7:0	P5MDIN[7:0]	Analog Configuration Bits for P5.7–P5.0 (respectively). Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled. 0: Corresponding P5.n pin is configured for analog mode. 1: Corresponding P5.n pin is not configured for analog mode.

SFR Definition 28.32. P5MDOUT: Port 5 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P5MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9B; SFR Page = F

Bit	Name	Function
7:0	P5MDOUT[7:0]	Output Configuration Bits for P5.7–P5.0 (respectively). These bits are ignored if the corresponding bit in register P5MDIN is logic 0. 0: Corresponding P5.n Output is open-drain. 1: Corresponding P5.n Output is push-pull.

imum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 30.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
0	$T_{low} - 4$ system clocks or 1 system clock + s/w delay *	3 system clocks
1	11 system clocks	12 system clocks
Note: Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgement, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI is cleared. Note that if SI is cleared in the same write that defines the outgoing ACK value, s/w delay is zero.		

With the SMBTOE bit set, Timer 3 should be configured to overflow after 25 ms in order to detect SCL low timeouts (see Section “30.3.4. SCL Low Timeout” on page 221). The SMBus interface will force Timer 3 to reload while SCL is high, and allow Timer 3 to count when SCL is low. The Timer 3 interrupt service routine should be used to reset SMBus communication by disabling and re-enabling the SMBus.

SMBus Free Timeout detection can be enabled by setting the SMBFTE bit. When this bit is set, the bus will be considered free if SDA and SCL remain high for more than 10 SMBus clock source periods (see Figure 30.4).

32.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 32.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

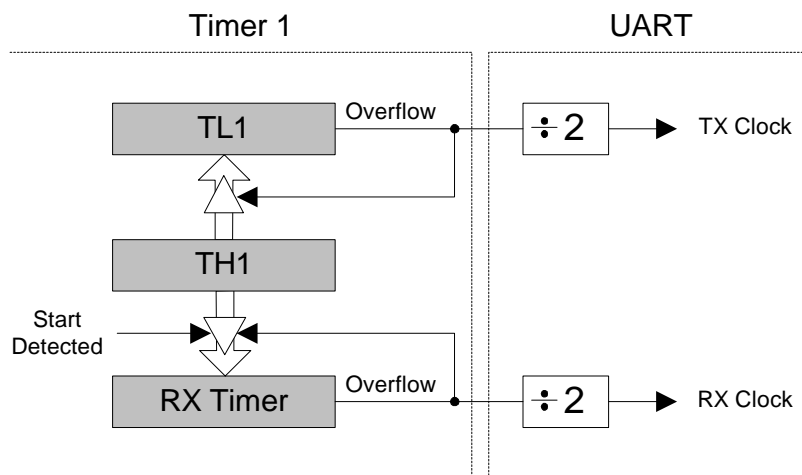


Figure 32.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section “33.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload” on page 265). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK/4, SYSCLK/12, SYSCLK/48, the external oscillator clock/8, or an external input T1. For any given Timer 1 clock source, the UART0 baud rate is determined by Equation 32.1-A and Equation 32.1-B.

$$A) \quad \text{UartBaudRate} = \frac{1}{2} \times \text{T1_Overflow_Rate}$$

$$B) \quad \text{T1_Overflow_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$

Equation 32.1. UART0 Baud Rate

Where T1_{CLK} is the frequency of the clock supplied to Timer 1, and T1H is the high byte of Timer 1 (reload value). Timer 1 clock frequency is selected as described in Section “33. Timers” on page 262. A quick reference for typical baud rates and system clock frequencies is given in Table 32.1 through Table 32.2. The internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

33.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 33.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode. Timer 3 can also be used in capture mode to capture rising edges of the Comparator 0 output.

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 bits (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF3H bit is set when TMR3H overflows from 0xFF to 0x00; the TF3L bit is set when TMR3L overflows from 0xFF to 0x00. When Timer 3 interrupts are enabled, an interrupt is generated each time TMR3H overflows. If Timer 3 interrupts are enabled and TF3LEN (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H 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.

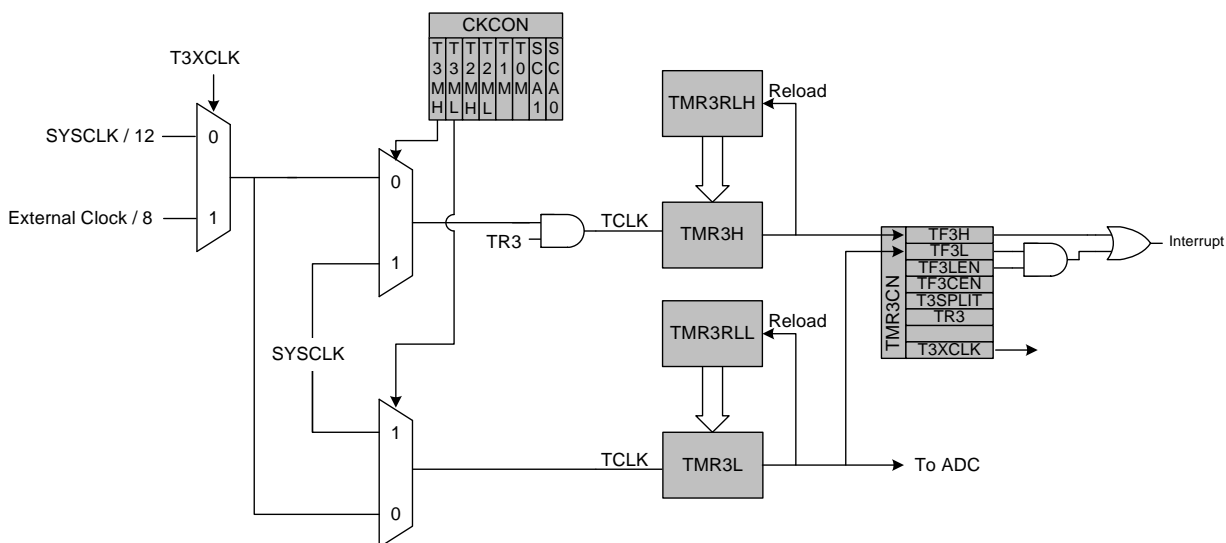


Figure 33.8. Timer 3 8-Bit Mode Block Diagram