C8051F360/1/2/3/4/5/6/7/8/9

Mixed Signal ISP Flash MCU Family

Analog Peripherals
- **10-Bit ADC (‘F360/1/2/6/7/8/9 only)**
  - Up to 200 ksps
  - Up to 21 external single-ended or differential inputs
  - VREF from internal VREF, external pin or V_{DD}
  - Internal or external start of conversion source
  - Built-in temperature sensor
- **10-Bit Current Output DAC (‘F360/1/2/6/7/8/9 only)**
- **Two Comparators**
  - Programmable hysteresis and response time
  - Configurable as interrupt or reset source
  - Low current (0.4 µA)
- **Brown-out detector and POR Circuitry**

On-Chip Debug
- On-chip debug circuitry facilitates full speed, non-intrusive in-system debug (no emulator required)
- Provides breakpoints, single stepping, inspect/modify memory and registers
- Superior performance to emulation systems using ICE-chips, target pods, and sockets
- Low cost, **complete** development kit

Supply Voltage
- Range: 2.7–3.6 V (50 MIPS) 3.0–3.6 V (100 MIPS)
- Power saving suspend and shutdown modes

High Speed 8051 µC Core
- Pipelined instruction architecture; executes 70% of instructions in 1 or 2 system clocks
- 100 MIPS or 50 MIPS throughput with on-chip PLL
- Expanded interrupt handler
- 2-cycle 16 x 16 MAC engine

Memory
- 1280 bytes internal data RAM (256 + 1024)
- 32 kB (‘F360/1/2/3/4/5/6/7) or 16 kB (‘F368/9) Flash; In-system programmable in 1024-byte Sectors—1024 bytes are reserved in the 32 kB devices

Digital Peripherals
- up to 39 Port I/O; All 5 V tolerant with high sink current
- Hardware enhanced UART, SMBus™, and enhanced SPI™ serial ports
- Four general purpose 16-bit counter/timers
- 16-Bit programmable counter array (PCA) with six capture/compare modules
- Real time clock mode using PCA or timer and external clock source
- External Memory Interface (EMIF)

Clock Sources
- Two internal oscillators:
  - 24.5 MHz with ±2% accuracy supports crystal-less UART operation
  - 80/40/20/10 kHz low frequency, low power
- Flexible PLL technology
- External oscillator: Crystal, RC, C, or clock (1 or 2 pin modes)
- Can switch between clock sources on-the-fly; useful in power saving modes

Packages
- 48-pin TQFP (C8051F360/3)
- 32-pin LQFP (C8051F361/4/6/8)
- 28-pin QFN (C8051F362/5/7/9)

Temperature Range: –40 to +85 °C

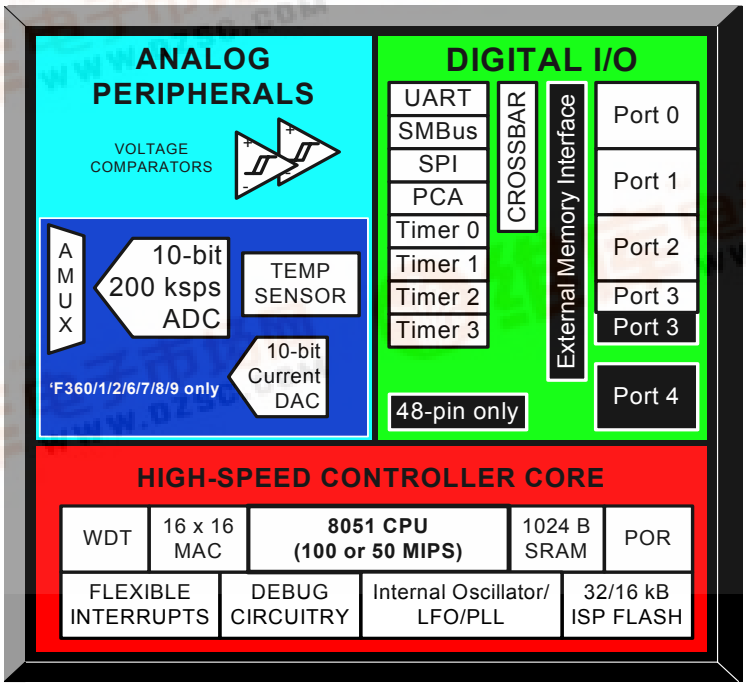


Table of Contents

List of Figures

List of Tables

List of Registers

1. System Overview

C8051F36x devices are fully integrated mixed-signal System-on-a-Chip MCUs. Highlighted features are listed below. Refer to Table 1.1 for specific product feature selection.

- High-speed pipelined 8051-compatible microcontroller core (up to 100 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 10-bit 200 ksps 16-channel single-ended/differential ADC with analog multiplexer
- 10-bit Current Output DAC
- 2-cycle 16 by 16 Multiply and Accumulate Engine
- Precision programmable 25 MHz internal oscillator
- Up to 32 kB of on-chip Flash memory—1024 bytes are reserved
- 1024 bytes of on-chip RAM
- External Data Memory Interface with 64 kB address space
- SMBus/I2C, Enhanced UART, and Enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- Programmable Counter/Timer Array (PCA) with six capture/compare modules and Watchdog Timer function
- On-chip Power-On Reset, V_{DD} Monitor, and Temperature Sensor
- Two on-chip Voltage Comparators
- up to 39 Port I/O (5 V tolerant)

With on-chip Power-On Reset, V_{DD} Monitor, Watchdog Timer, and clock oscillator, the C8051F36x devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed even in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The on-chip Silicon Labs 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system debugging without occupying package pins.

Each device is specified for 3.0 to 3.6 V (100 MIPS) operation or 2.7 to 3.6 V (50 MIPS) operation over the industrial temperature range (–40 to +85 °C). The Port I/O and $\overline{\text{RST}}$ pins are tolerant of input signals up to 5 V. The C8051F36x devices are available in 48-pin TQFP packages, and C8051F36x devices are available in 32-pin LQFP and 28-pin QFN packages (also referred to as MLP or MLF packages). All package types are lead-free (RoHS compliant). See Table 1.1 for ordering part numbers. Block diagrams are included in Figure 1.1, Figure 1.2, and Figure 1.3.

Table 1.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	Flash Memory (kB)	RAM (bytes)	2-cycle 16 by 16 MAC	Calibrated Internal 24.5 MHz Oscillator	Internal 80 kHz Oscillator	External Memory Interface	SMBus/I^2C	Enhanced SPI	UART	Timers (16-bit)	Programmable Counter Array	Digital Port I/Os	10-bit 200ksps ADC	10-bit Current Output DAC	Internal Voltage Reference	Temperature Sensor	Analog Comparators	Lead-free (RoHS Compliant)	Package
C8051F360-GQ	100	32	1024	✓	✓	✓	✓	✓	✓	✓	4	✓	39	✓	✓	✓	✓	2	✓	TQFP-48
C8051F361-GQ[1]	100	32	1024	✓	✓	✓	—	✓	✓	✓	4	✓	29	✓	✓	✓	✓	2	✓	LQFP-32
C8051F362-GM[2]	100	32	1024	✓	✓	✓	—	✓	✓	✓	4	✓	25	✓	✓	✓	✓	2	✓	QFN-28
C8051F363-GQ	100	32	1024	✓	✓	✓	✓	✓	✓	✓	4	✓	39	—	—	—	—	2	✓	TQFP-48
C8051F364-GQ[1]	100	32	1024	✓	✓	✓	—	✓	✓	✓	4	✓	29	—	—	—	—	2	✓	LQFP-32
C8051F365-GM[2]	100	32	1024	✓	✓	✓	—	✓	✓	✓	4	✓	25	—	—	—	—	2	✓	QFN-28
C8051F366-GQ[1]	50	32	1024	✓	✓	✓	—	✓	✓	✓	4	✓	29	✓	✓	✓	✓	2	✓	LQFP-32
C8051F367-GM[2]	50	32	1024	✓	✓	✓	—	✓	✓	✓	4	✓	25	✓	✓	✓	✓	2	✓	QFN-28
C8051F368-GQ[1]	50	16	1024	✓	✓	✓	—	✓	✓	✓	4	✓	29	✓	✓	✓	✓	2	✓	LQFP-32
C8051F369-GM[2]	50	16	1024	✓	✓	✓	—	✓	✓	✓	4	✓	25	✓	✓	✓	✓	2	✓	QFN-28

Notes:
1. Pin compatible with the C8051F310-GQ.
2. Pin compatible with the C8051F311-GM.

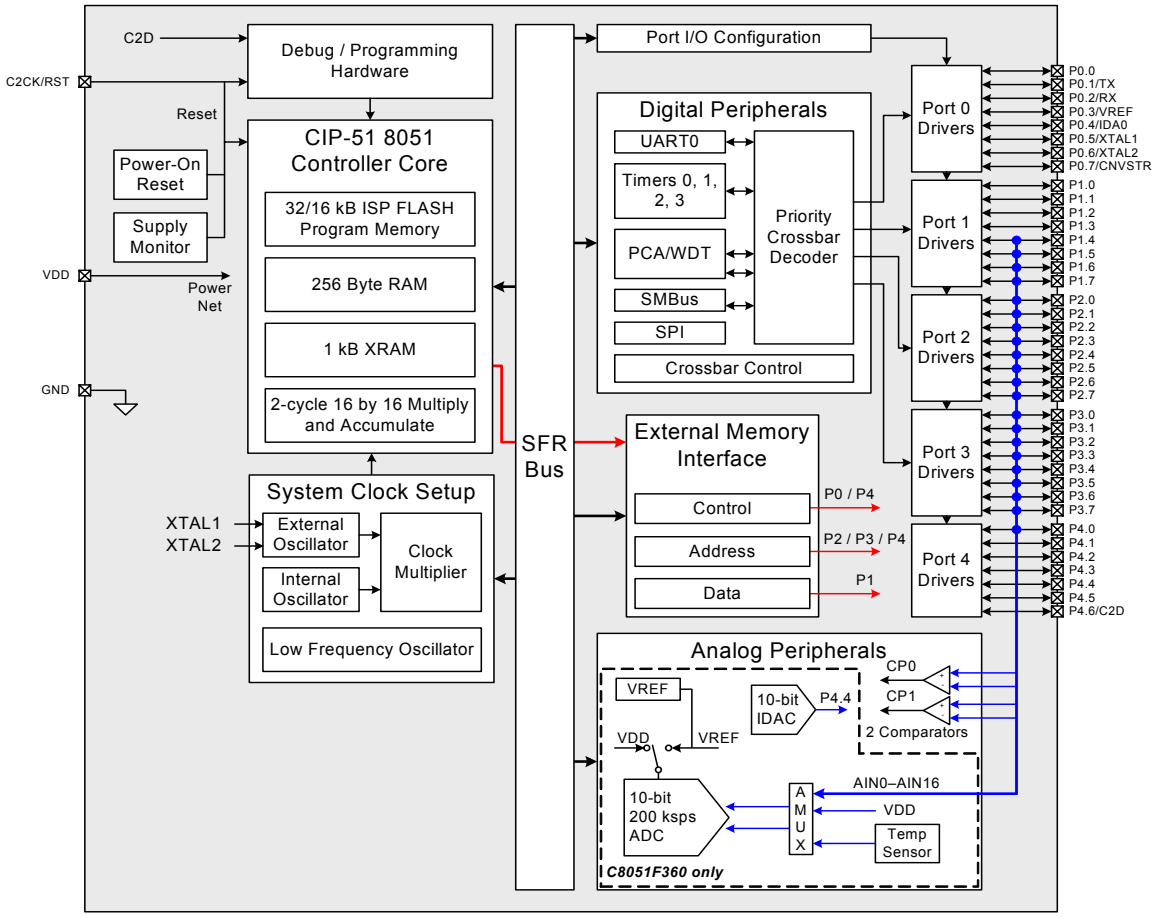


Figure 1.1. C8051F360/3 Block Diagram

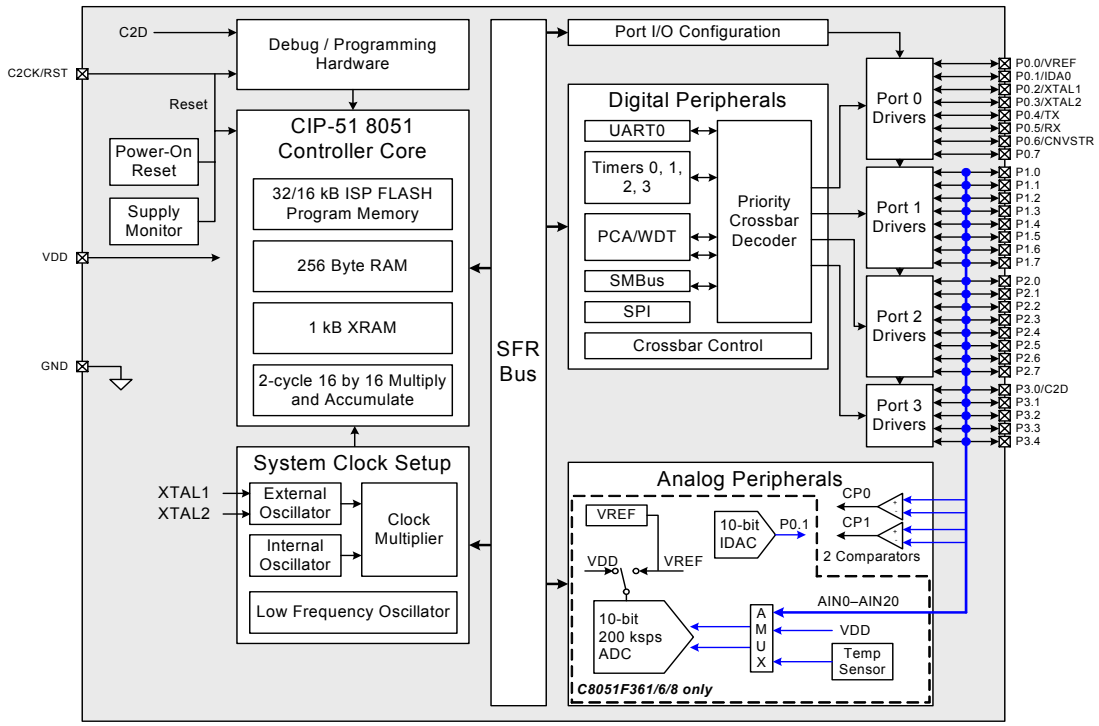


Figure 1.2. C8051F361/4/6/8 Block Diagram

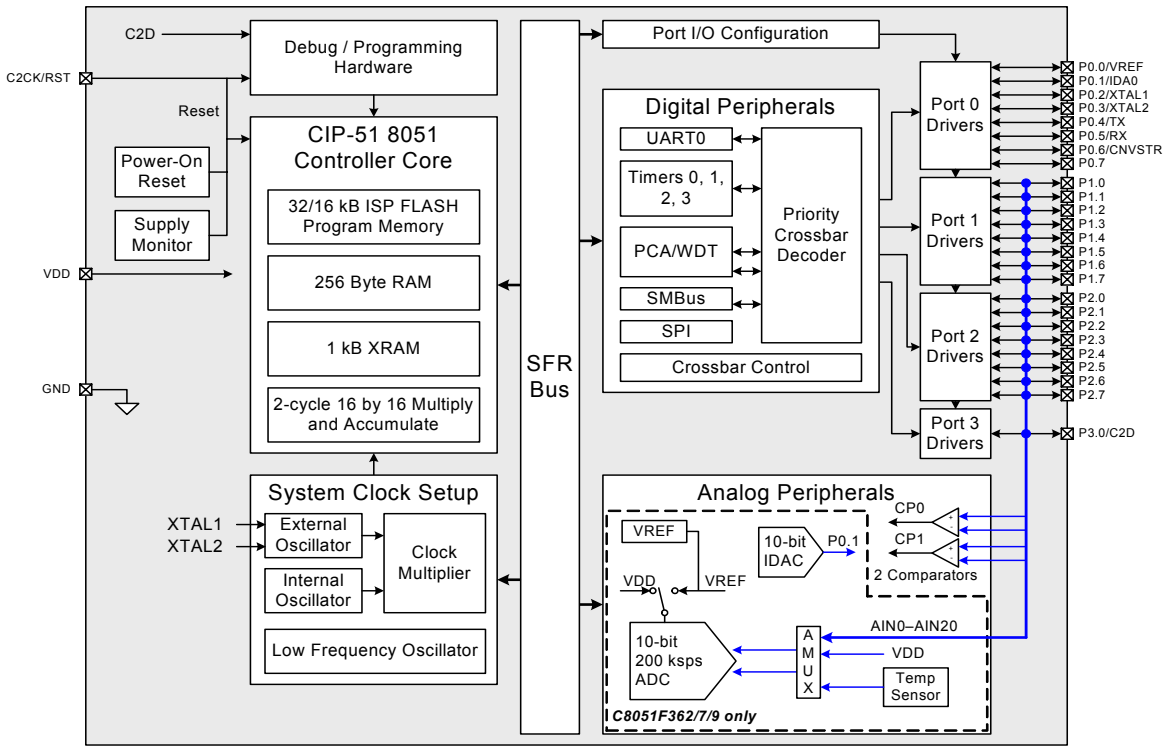


Figure 1.3. C8051F362/5/7/9 Block Diagram

1.1. CIP-51™ Microcontroller Core

1.1.1. Fully 8051 Compatible

The C8051F36x family utilizes Silicon Labs' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The CIP-51 core offers all the peripherals included with a standard 8052, including four 16-bit counter/timers, a full-duplex UART with extended baud rate configuration, an enhanced SPI port, 1024 bytes of internal RAM, 128 byte Special Function Register (SFR) address space, and up to 39 I/O pins.

1.1.2. Improved Throughput

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute with a maximum system clock of 12-to-24 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with only four instructions taking more than four system clock cycles.

The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

With the CIP-51's maximum system clock at 100 MHz, it has a peak throughput of 100 MIPS. Figure 1.4 shows a comparison of peak throughputs for various 8-bit microcontroller cores with their maximum system clocks.

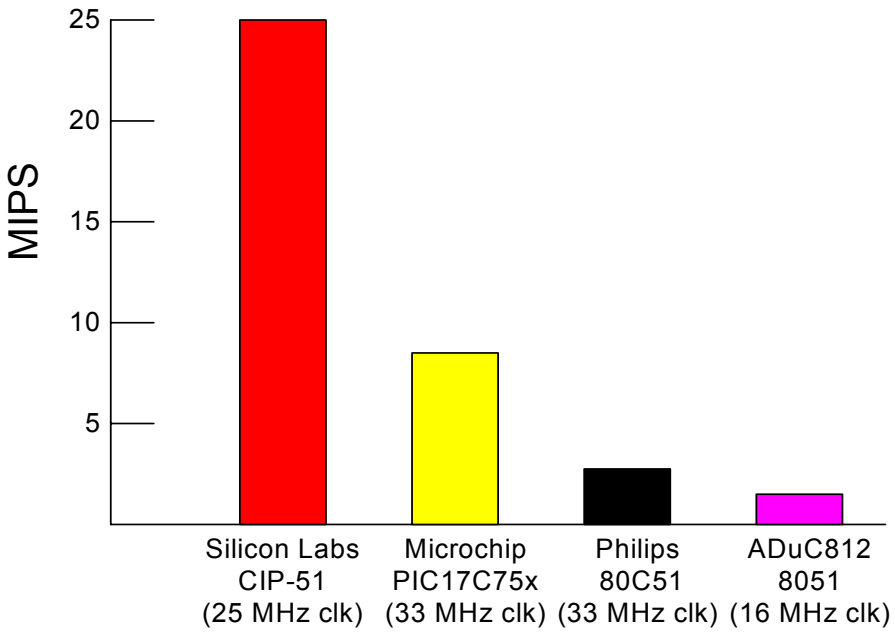


Figure 1.4. Comparison of Peak MCU Execution Speeds

1.1.3. Additional Features

The C8051F36x SoC family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

The extended interrupt handler provides 16 interrupt sources into the CIP-51 (as opposed to 7 for the standard 8051), allowing numerous analog and digital peripherals to interrupt the controller. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The extra interrupt sources are very useful when building multi-tasking, real-time systems.

Eight reset sources are available: power-on reset circuitry (POR), an on-chip V_{DD} Monitor (forces reset when power supply voltage drops below V_{RST} as given in Table 12.1 on page 134), a Watchdog Timer, a Missing Clock Detector, a voltage level detection from Comparator0, a forced software reset, an external reset pin, and an illegal Flash access protection circuit. Each reset source except for the POR, Reset Input Pin, or Flash error may be disabled by the user in software. The WDT may be permanently enabled in software after a power-on reset during MCU initialization.

The internal oscillator factory calibrated to 24.5 MHz ±2%. This internal oscillator period may be user programmed in ~0.5% increments. An additional low-frequency oscillator is also available which facilitates low-power operation. An external oscillator drive circuit is included, allowing an external crystal, ceramic resonator, capacitor, RC, or CMOS clock source to generate the system clock. If desired, the system clock source may be switched on-the-fly between both internal and external oscillator circuits. An external oscillator can also be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) source, while periodically switching to the fast (up to 25 MHz) internal oscillator as needed. Additionally, an on-chip PLL is provided to achieve higher system clock speeds for increased throughput.

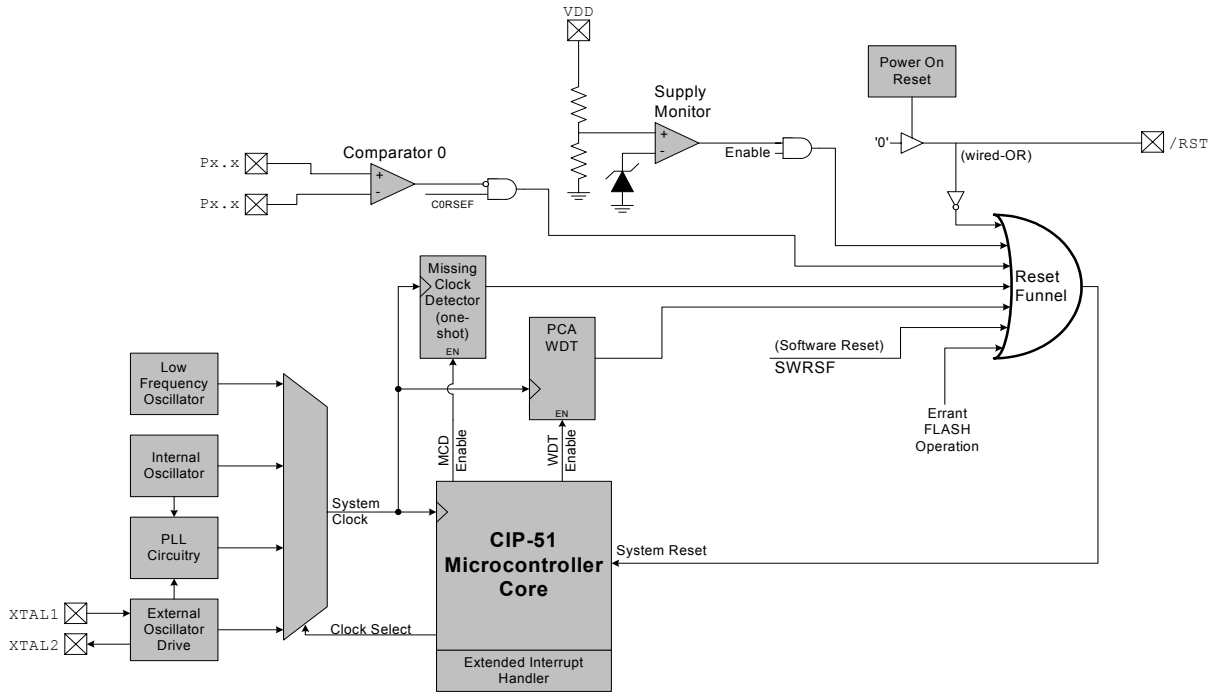


Figure 1.5. On-Chip Clock and Reset

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 32/16 kB of Flash. This memory may be reprogrammed in-system in 1024 byte sectors, and requires no special off-chip programming voltage. See Figure 1.6 for the MCU system memory map.

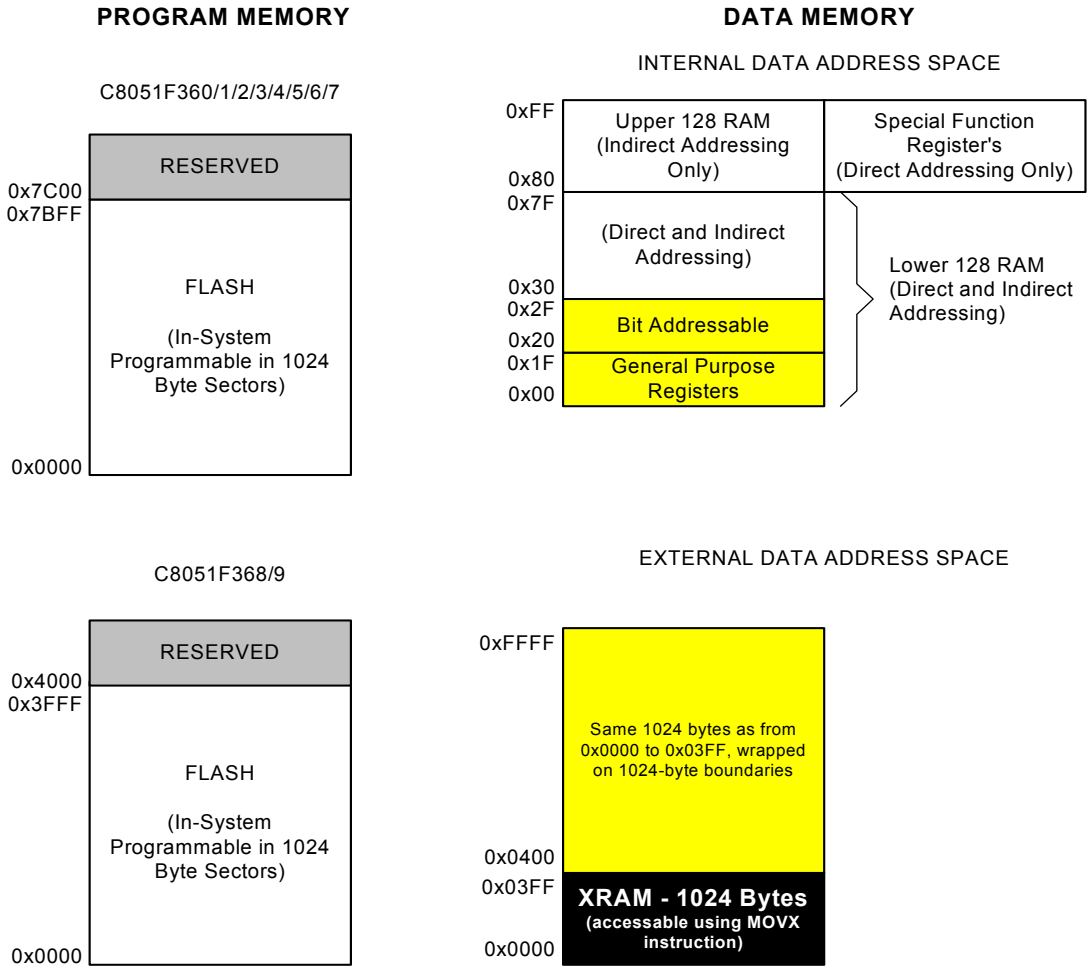


Figure 1.6. On-Board Memory Map

1.3. On-Chip Debug Circuitry

The C8051F36x devices include on-chip Silicon Labs 2-Wire (C2) debug circuitry that provides non-intrusive, full speed, in-circuit debugging of the production part *installed in the end application.*

Silicon Labs' debugging system supports inspection and modification of memory and registers, breakpoints, and single stepping. No additional target RAM, program memory, timers, or communications chan-

nels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the ADC and SMBus) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized.

The C8051F360DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F36x MCUs. The kit includes software with a developer's studio and debugger, an integrated 8051 assembler, and a debug adapter. It also has a target application board with the associated MCU installed and prototyping area, plus the required cables, and wall-mount power supply. The Development Kit requires a PC running Windows98SE or later.

The Silicon Labs IDE interface is a vastly superior developing and debugging configuration, compared to standard MCU emulators that use on-board "ICE Chips" and require the MCU in the application board to be socketed. Silicon Labs' debug paradigm increases ease of use and preserves the performance of the precision analog peripherals.

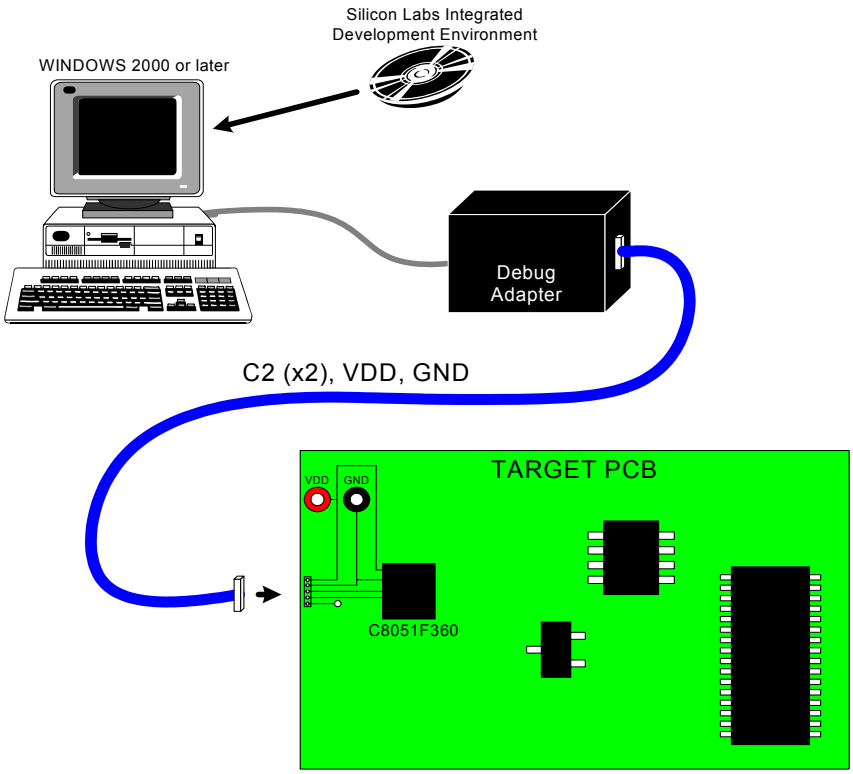


Figure 1.7. Development/In-System Debug Diagram

1.4. Programmable Digital I/O and Crossbar

C8051F36x devices include up to 39 I/O pins (four byte-wide Ports and one 7-bit-wide Port). The C8051F36x Ports behave like typical 8051 Ports with a few enhancements. Each Port pin may be configured as an analog input or a digital I/O pin. Pins selected as digital I/Os may additionally be configured for push-pull or open-drain output. The “weak pullups” that are fixed on typical 8051 devices may be globally disabled, providing power savings capabilities.

The Digital Crossbar allows mapping of internal digital system resources to Port I/O pins. (See Figure 1.8.) On-chip counter/timers, serial buses, HW interrupts, comparator output, and other digital signals in the controller can be configured to appear on the Port I/O pins specified in the Crossbar Control registers. This allows the user to select the exact mix of general purpose Port I/O and digital resources needed for the particular application.

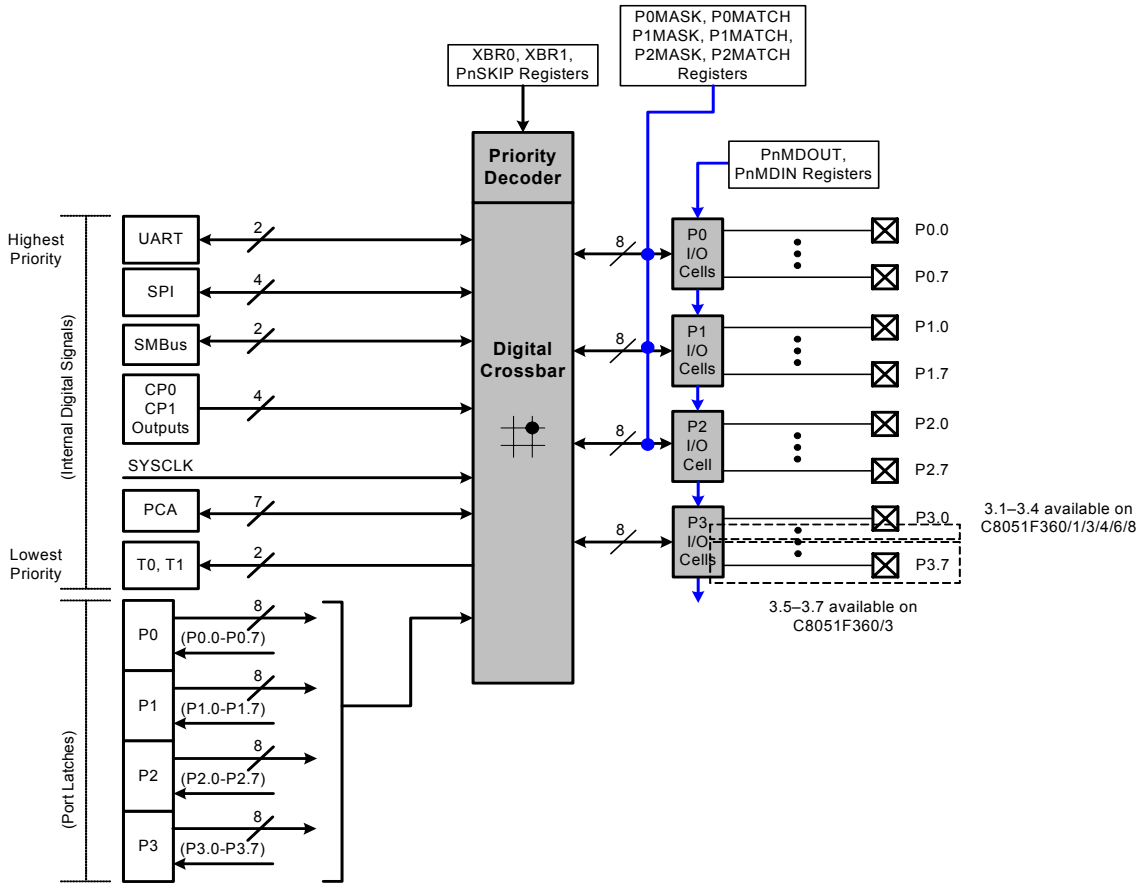


Figure 1.8. Digital Crossbar Diagram (Port 0 to Port 3)

1.5. Serial Ports

The C8051F36x Family includes an SMBus/I^2C interface, a full-duplex UART with enhanced baud rate configuration, and an Enhanced SPI interface. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little CPU intervention.

1.6. Programmable Counter Array

An on-chip Programmable Counter/Timer Array (PCA) is included in addition to the four 16-bit general purpose counter/timers. The PCA consists of a dedicated 16-bit counter/timer time base with three programmable capture/compare modules. The PCA clock is derived from one of six sources: the system clock divided by 12, the system clock divided by 4, Timer 0 overflows, an External Clock Input (ECI), the system clock, or the external oscillator clock source divided by 8. The external clock source selection is useful for

real-time clock functionality, where the PCA is clocked by an external source while the internal oscillator drives the system clock.

Each capture/compare module can be configured to operate in one of six modes: Edge-Triggered Capture, Software Timer, High Speed Output, 8- or 16-bit Pulse Width Modulator, or Frequency Output. Additionally, Capture/Compare Module 5 offers watchdog timer (WDT) capabilities. Following a system reset, Module 5 is configured and enabled in WDT mode. The PCA Capture/Compare Module I/O and External Clock Input may be routed to Port I/O via the Digital Crossbar.

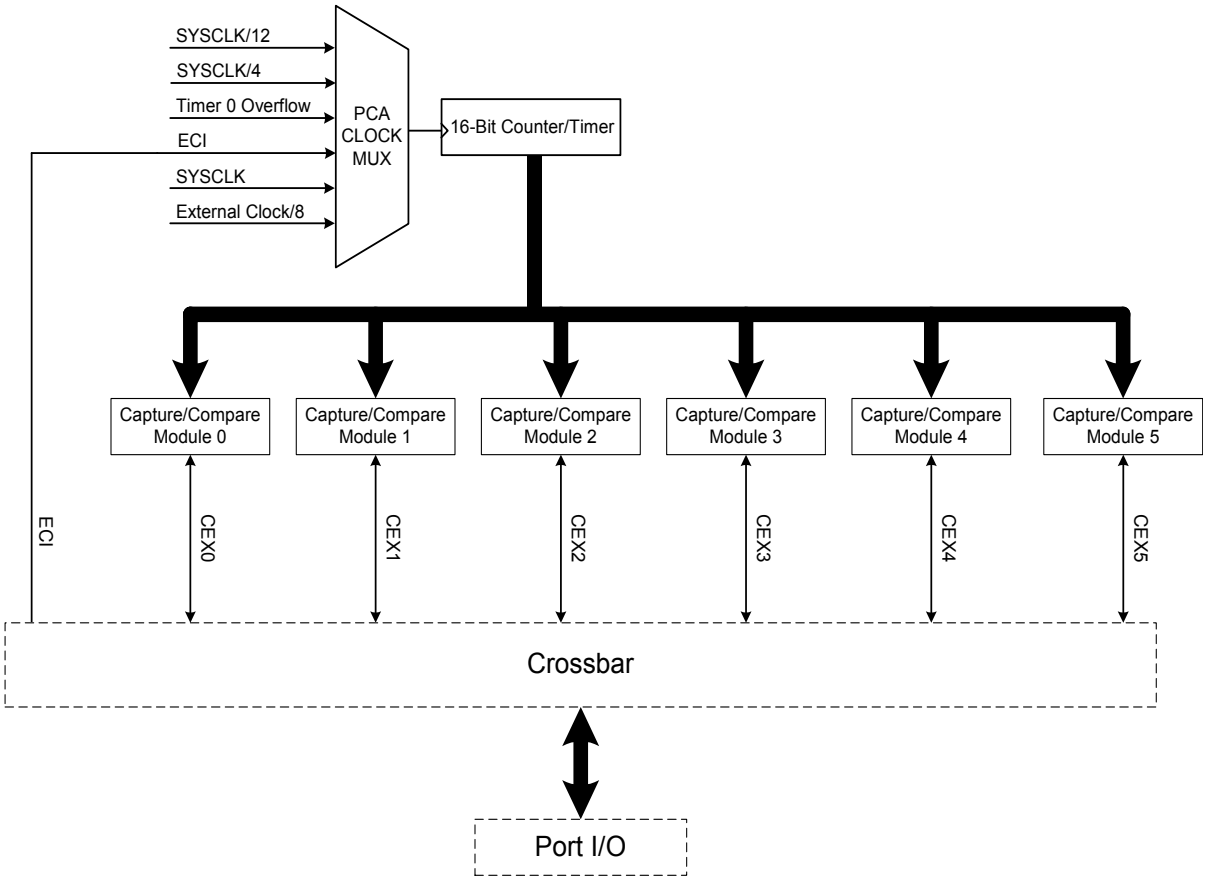


Figure 1.10. PCA Block Diagram

1.7. 10-Bit Analog to Digital Converter

The C8051F360/1/2/6/7/8/9 devices include an on-chip 10-bit SAR ADC with up to 21 channels for the differential input multiplexer. With a maximum throughput of 200 ksps, the ADC offers true 10-bit linearity with an INL and DNL of ±1 LSB. The ADC system includes a configurable analog multiplexer that selects both positive and negative ADC inputs. Ports1-3 are available as an ADC inputs; additionally, the on-chip Temperature Sensor output and the power supply voltage (V_{DD}) are available as ADC inputs. User firmware may shut down the ADC to save power.

Conversions can be started in six ways: a software command, an overflow of Timer 0, 1, 2, or 3, or an external convert start signal (CNVSTR). This flexibility allows the start of conversion to be triggered by software events, a periodic signal (timer overflows), or external HW signals. Conversion completions are indi-

cated by a status bit and an interrupt (if enabled). The resulting 10-bit data word is latched into the ADC data SFRs upon completion of a conversion.

Window compare registers for the ADC data can be configured to interrupt the controller when ADC data is either within or outside of a specified range. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within/outside the specified range.

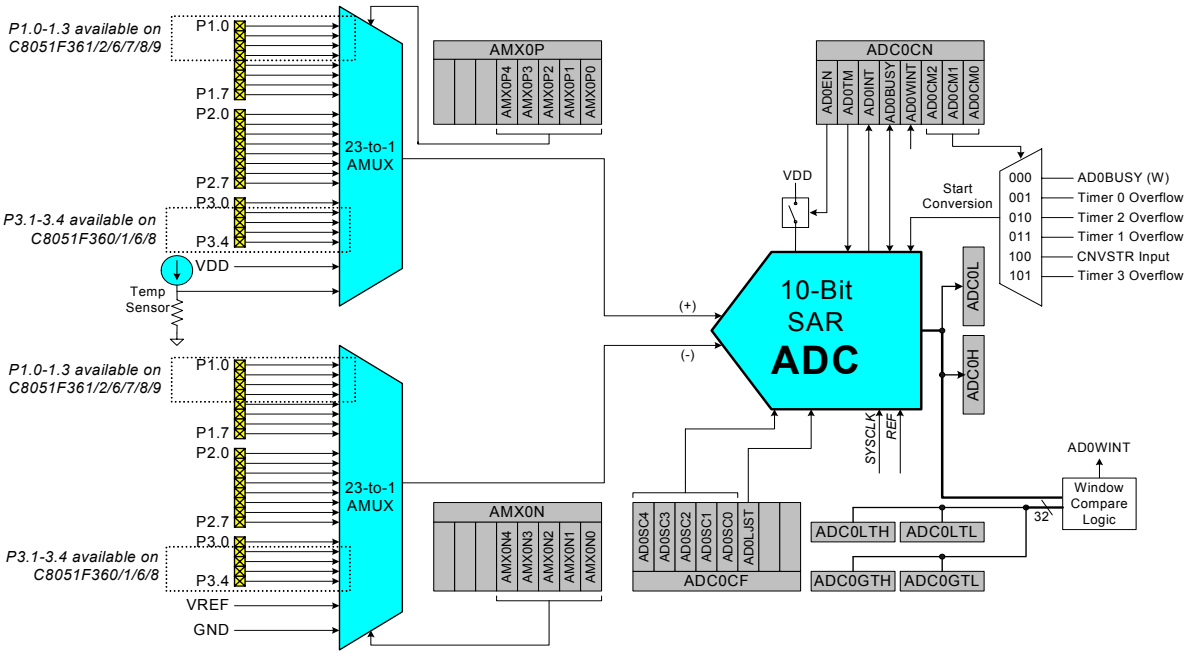


Figure 1.11. 10-Bit ADC Block Diagram

1.8. Comparators

C8051F36x devices include two on-chip voltage comparators that are enabled/disabled and configured via user software. Port I/O pins may be configured as comparator inputs via a selection mux. Two comparator outputs may be routed to a Port pin if desired: a latched output and/or an unlatched (asynchronous) output. Comparator response time is programmable, allowing the user to select between high-speed and low-power modes. Positive and negative hysteresis are also configurable.

Comparator interrupts may be generated on rising, falling, or both edges. When in IDLE mode, these interrupts may be used as a “wake-up” source. Comparator0 may also be configured as a reset source. Figure 1.12 shows the Comparator0 block diagram, and Figure 1.13 shows the Comparator1 block diagram.

Note: The first Port I/O pins shown in Figure 1.12 and Figure 1.13 are for the 48-pin (C8051F360/3) devices. The second set of Port I/O pins are for the 32-pin and 28-pin (C8051F361/2/4/5/6/7/8/9) devices. Please refer to the CPTnMX registers (SFR Definition 8.2 and SFR Definition 8.5) for more information.

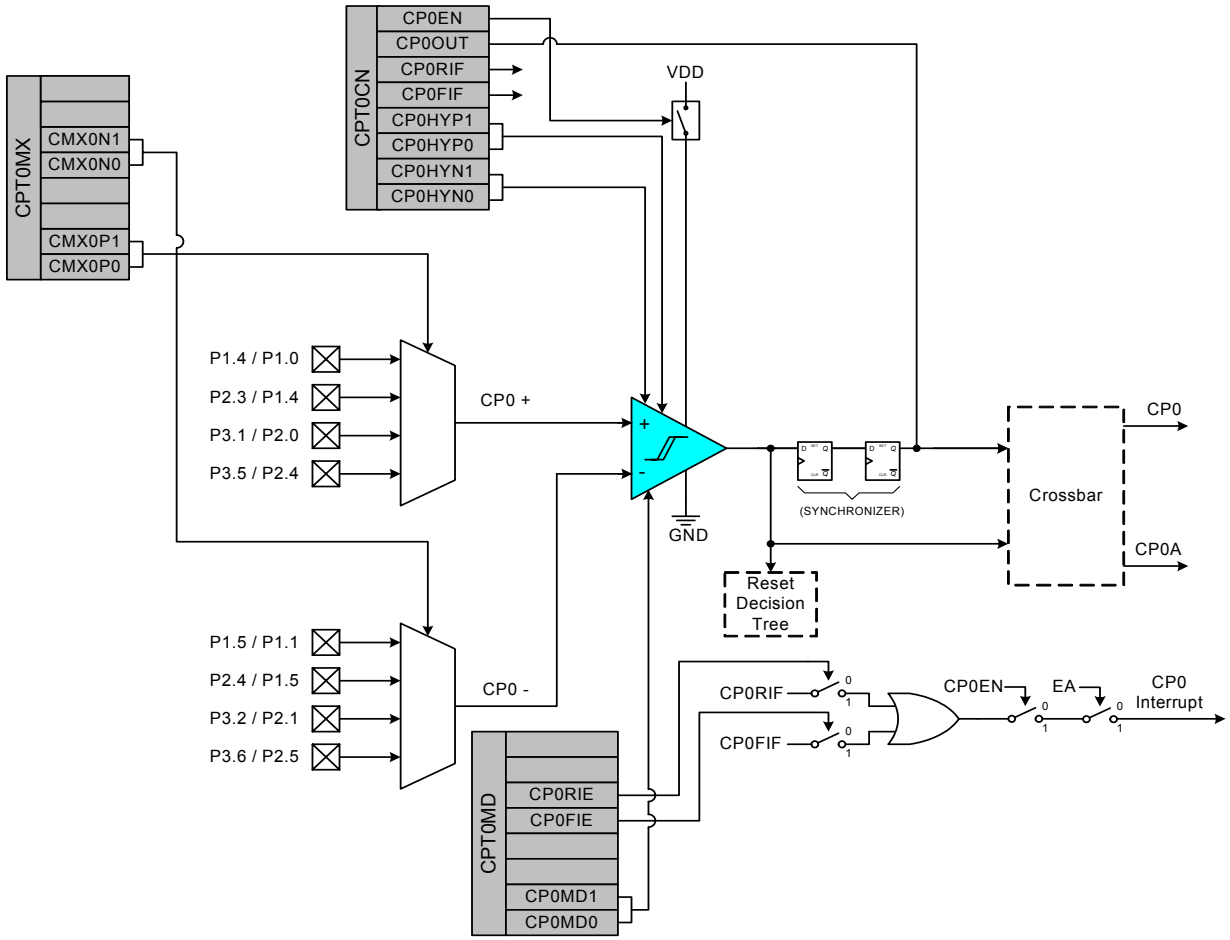


Figure 1.12. Comparator0 Block Diagram

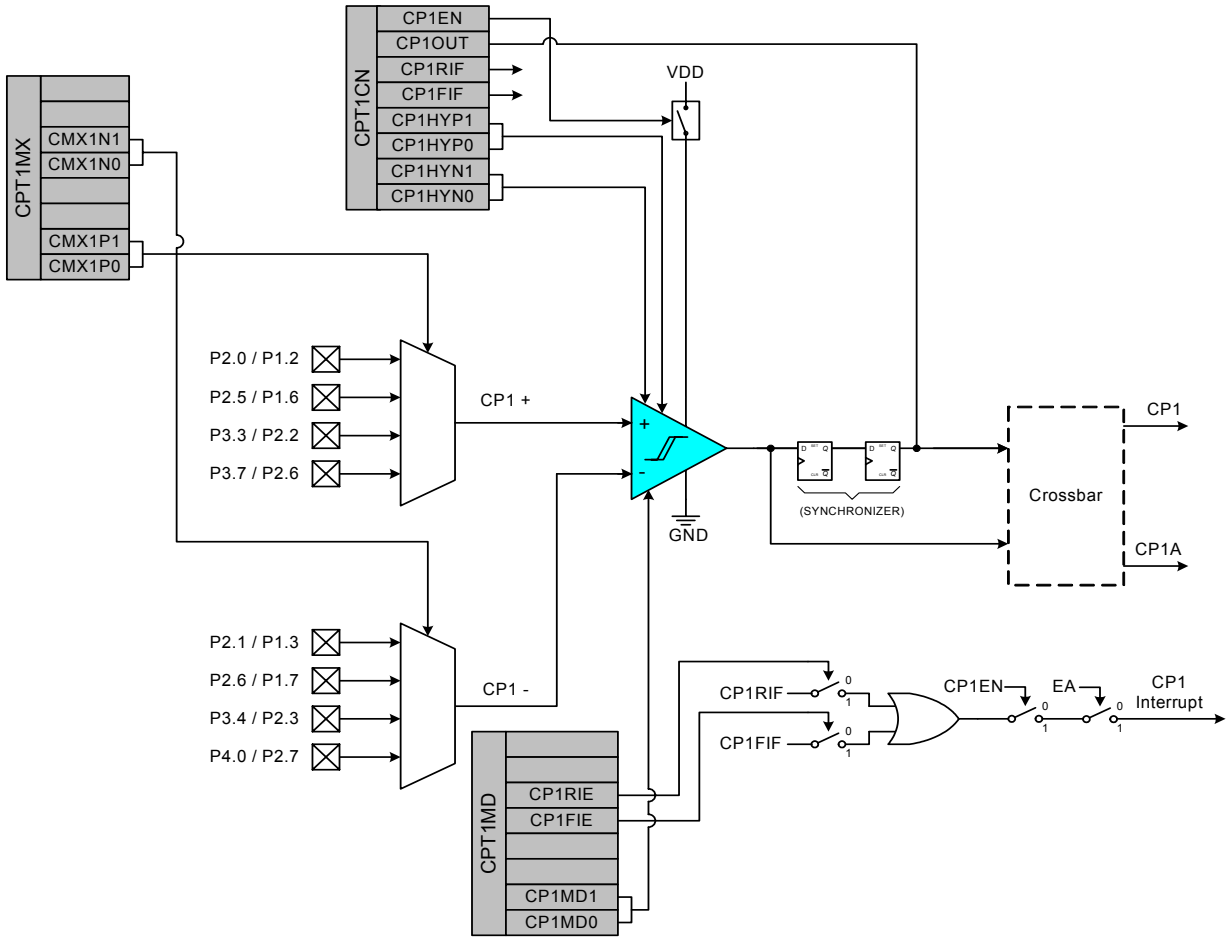


Figure 1.13. Comparator1 Block Diagram

1.9. 10-bit Current Output DAC

The C8051F360/1/2/6/7/8/9 devices includes a 10-bit current-mode Digital-to-Analog Converter (IDA0). The maximum current output of the IDA0 can be adjusted for three different current settings; 0.5 mA, 1 mA, and 2 mA. IDA0 features a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. Three update modes are provided, allowing IDA0 output updates on a write to IDA0H, on a Timer overflow, or on an external pin edge.

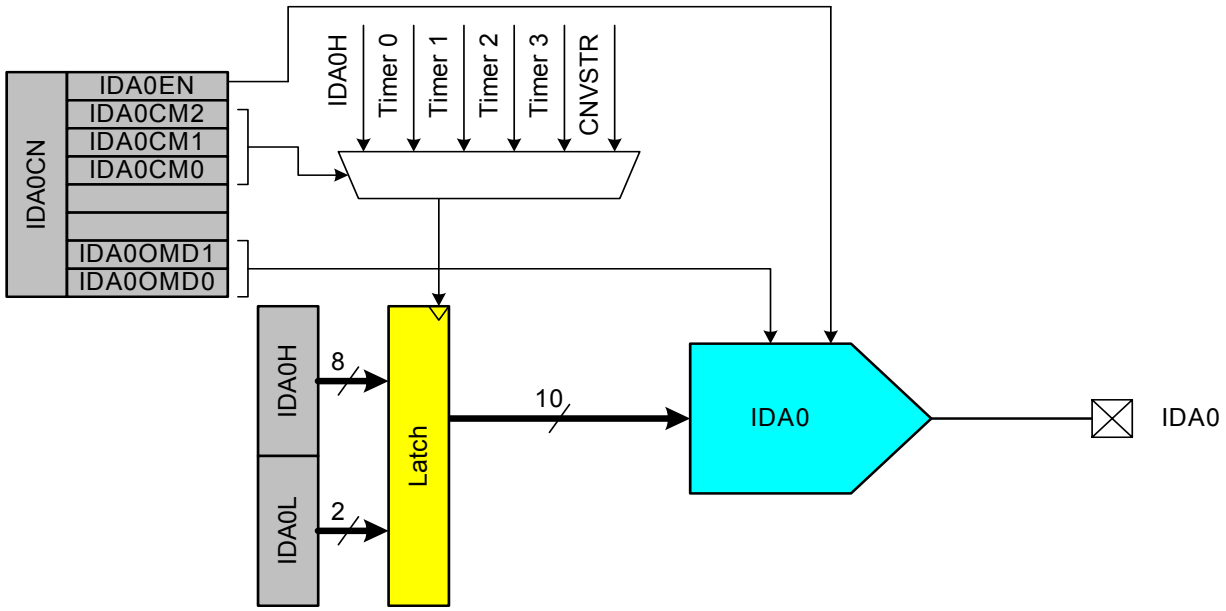


Figure 1.14. IDA0 Functional Block Diagram

2. Absolute Maximum Ratings

Table 2.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient temperature under bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on any Port I/O Pin or $\overline{\text{RST}}$ with respect to GND		–0.3	—	5.8	V
Voltage on V_{DD} with respect to GND		–0.3	—	4.2	V
Maximum Total current through V_{DD} or GND		—	—	500	mA
Maximum output current sunk by $\overline{\text{RST}}$ or any Port pin		—	—	100	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

3. Global Electrical Characteristics

Table 3.1. Global Electrical Characteristics

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Voltage	SYSCLK = 0 to 50 MHz SYSCLK > 50 MHz	2.7 3.0	3.0 3.3	3.6 3.6	V
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock)[1,2]	C8051F360/1/2/3/4/5 C8051F366/7/8/9	0 0	— —	100 50	MHz MHz
Specified Operating Temperature Range		–40	—	+85	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[2]	V_{DD} = 3.6 V, F = 100 MHz	—	68	75	mA
	V_{DD} = 3.6 V, F = 25 MHz	—	21	25	mA
	V_{DD} = 3.0 V, F = 100 MHz	—	54	60	mA
	V_{DD} = 3.0 V, F = 25 MHz	—	16	18	mA
	V_{DD} = 3.0 V, F = 1 MHz	—	0.48	—	mA
	V_{DD} = 3.0 V, F = 80 kHz	—	36	—	µA
I_{DD} Supply Sensitivity[3]	F = 25 MHz	—	56	—	%/V
	F = 1 MHz	—	57	—	%/V
I_{DD} Frequency Sensitivity[3,4]	V_{DD} = 3.0 V, F <= 20 MHz, T = 25 °C	—	0.45	—	mA/MHz
	V_{DD} = 3.0 V, F > 20 MHz, T = 25 °C	—	0.38	—	mA/MHz
	V_{DD} = 3.6 V, F <= 20 MHz, T = 25 °C	—	0.61	—	mA/MHz
	V_{DD} = 3.6 V, F > 20 MHz, T = 25 °C	—	0.51	—	mA/MHz

Table 3.1. Global Electrical Characteristics (Continued)

–40 to +85 °C, 25 MHz system clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash)					
I_{DD}[2]	V_{DD} = 3.6 V, F = 100 MHz	—	36	40	mA
	V_{DD} = 3.6 V, F = 25 MHz	—	9	12	mA
	V_{DD} = 3.0 V, F = 100 MHz	—	30	35	mA
	V_{DD} = 3.0 V, F = 25 MHz	—	7	9	mA
	V_{DD} = 3.0 V, F = 1 MHz	—	0.24	—	mA
	V_{DD} = 3.0 V, F = 80 kHz	—	19	—	µA
I_{DD} Supply Sensitivity[3]	F = 25 MHz	—	44	—	%/V
	F = 1 MHz	—	43.7	—	%/V
I_{DD} Frequency Sensitivity[3,5]	V_{DD} = 3.0 V, F <= 1 MHz, T = 25 °C	—	0.24	—	mA/MHz
	V_{DD} = 3.0 V, F > 1 MHz, T = 25 °C	—	0.25	—	mA/MHz
	V_{DD} = 3.6 V, F <= 1 MHz, T = 25 °C	—	0.31	—	mA/MHz
	V_{DD} = 3.6 V, F > 1 MHz, T = 25 °C	—	0.32	—	mA/MHz
Digital Supply Current (Stop Mode, shutdown)	Oscillator not running, V_{DD} Monitor Disabled	—	< 0.1	—	µA

Notes:

1. SYSCLK must be at least 32 kHz to enable debugging.
2. SYSCLK is the internal device clock. For operational speeds in excess of 30 MHz, SYSCLK must be derived from the Phase-Locked Loop (PLL).
3. Based on device characterization data; Not production tested.
4. IDD can be estimated for frequencies ≤ 20 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} for >20 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 3.0 V; F = 20 MHz, I_{DD} = 15.9 mA - (25 MHz - 20 MHz) * 0.38 mA/MHz = 14 mA.
5. Idle IDD can be estimated for frequencies ≤ 1 MHz by simply multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate Idle I_{DD} for >1 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 3.0 V; F = 5 MHz, Idle I_{DD} = 7.2 mA - (25 MHz - 5 MHz) * 0.25 mA/MHz = 2.2 mA.

Other electrical characteristics tables are found in the data sheet section corresponding to the associated peripherals. For more information on electrical characteristics for a specific peripheral, refer to the page indicated in Table 3.2.

Table 3.2. Index to Electrical Characteristics Tables

Peripheral Electrical Characteristics	Page No.
ADC0 Electrical Characteristics	62
IDAC Electrical Characteristics	66
Voltage Reference Electrical Characteristics	69
Comparator Electrical Characteristics	79
Reset Electrical Characteristics	134
Flash Electrical Characteristics	144
Internal High Frequency Oscillator Electrical Characteristics	171
Internal Low Frequency Oscillator Electrical Characteristics	172
PLL Frequency Characteristics	182
Port I/O DC Electrical Characteristics	201

4. Pinout and Package Definitions

Table 4.1. Pin Definitions for the C8051F36x

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
V_{DD}	19, 31, 43	4	4		Power Supply Voltage.
GND	18, 30, 42	3	3		Ground.
AGND	6	—	—		Analog Ground.
AV+	7	—	—		Analog Supply Voltage. Must be tied to +2.7 to +3.6 V.
$\overline{RST}$/	8	5	5	D I/O	Device Reset. Open-drain output of internal POR or V_{DD} Monitor. An external source can initiate a system reset by driving this pin low for at least 10 µs.
C2CK				D I/O	Clock signal for the C2 Debug Interface.
P4.6/	9	—	—	D I/O or A In	Port 4.6. See Section 17 for a complete description.
C2D				D I/O	Bi-directional data signal for the C2 Debug Interface.
P3.0/	—	6	6	D I/O or A In	Port 3.0. See Section 17 for a complete description.
C2D				D I/O	Bi-directional data signal for the C2 Debug Interface.
P0.0	5	2	2	D I/O or A In	Port 0.0. See Section 17 for a complete description.
P0.1	4	1	1	D I/O or A In	Port 0.1. See Section 17 for a complete description.
P0.2	3	32	28	D I/O or A In	Port 0.2. See Section 17 for a complete description.
P0.3	2	31	27	D I/O or A In	Port 0.3. See Section 17 for a complete description.
P0.4	1	30	26	D I/O or A In	Port 0.4. See Section 17 for a complete description.
P0.5	48	29	25	D I/O or A In	Port 0.5. See Section 17 for a complete description.
P0.6	47	28	24	D I/O or A In	Port 0.6. See Section 17 for a complete description.
P0.7	46	27	23	D I/O or A In	Port 0.7. See Section 17 for a complete description.

Table 4.1. Pin Definitions for the C8051F36x (Continued)

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
P1.0	45	26	22	D I/O or A In	Port 1.0. See Section 17 for a complete description.
P1.1	44	25	21	D I/O or A In	Port 1.1. See Section 17 for a complete description.
P1.2	41	24	20	D I/O or A In	Port 1.2. See Section 17 for a complete description.
P1.3	40	23	19	D I/O or A In	Port 1.3. See Section 17 for a complete description.
P1.4	39	22	18	D I/O or A In	Port 1.4. See Section 17 for a complete description.
P1.5	38	21	17	D I/O or A In	Port 1.5. See Section 17 for a complete description.
P1.6	37	20	16	D I/O or A In	Port 1.6. See Section 17 for a complete description.
P1.7	36	19	15	D I/O or A In	Port 1.7. See Section 17 for a complete description.
P2.0	35	18	14	D I/O or A In	Port 2.0. See Section 17 for a complete description.
P2.1	34	17	13	D I/O or A In	Port 2.1. See Section 17 for a complete description.
P2.2	33	16	12	D I/O or A In	Port 2.2. See Section 17 for a complete description.
P2.3	32	15	11	D I/O or A In	Port 2.3. See Section 17 for a complete description.
P2.4	29	14	10	D I/O or A In	Port 2.4. See Section 17 for a complete description.
P2.5	28	13	9	D I/O or A In	Port 2.5. See Section 17 for a complete description.
P2.6	27	12	8	D I/O or A In	Port 2.6. See Section 17 for a complete description.
P2.7	26	11	7	D I/O or A In	Port 2.7. See Section 17 for a complete description.
P3.0	25	—	—	D I/O or A In	Port 3.0. See Section 17 for a complete description.

Table 4.1. Pin Definitions for the C8051F36x (Continued)

Name	Pin 'F360/3 (48-pin)	Pin 'F361/4/6/8 (32-pin)	Pin 'F362/5/7/9 (28-pin)	Type	Description
P3.1	24	7	—	D I/O or A In	Port 3.1. See Section 17 for a complete description.
P3.2	23	8	—	D I/O or A In	Port 3.2. See Section 17 for a complete description.
P3.3	22	9	—	D I/O or A In	Port 3.3. See Section 17 for a complete description.
P3.4	21	10	—	D I/O or A In	Port 3.4. See Section 17 for a complete description.
P3.5	20	—	—	D I/O or A In	Port 3.5. See Section 17 for a complete description.
P3.6	17	—	—	D I/O or A In	Port 3.6. See Section 17 for a complete description.
P3.7	16	—	—	D I/O or A In	Port 3.7. See Section 17 for a complete description.
P4.0	15	—	—	D I/O or A In	Port 4.0. See Section 17 for a complete description.
P4.1	14	—	—	D I/O	Port 4.1. See Section 17 for a complete description.
P4.2	13	—	—	D I/O	Port 4.2. See Section 17 for a complete description.
P4.3	12	—	—	D I/O	Port 4.3. See Section 17 for a complete description.
P4.4	11	—	—	D I/O	Port 4.4. See Section 17 for a complete description.
P4.5	10	—	—	D I/O	Port 4.5. See Section 17 for a complete description.

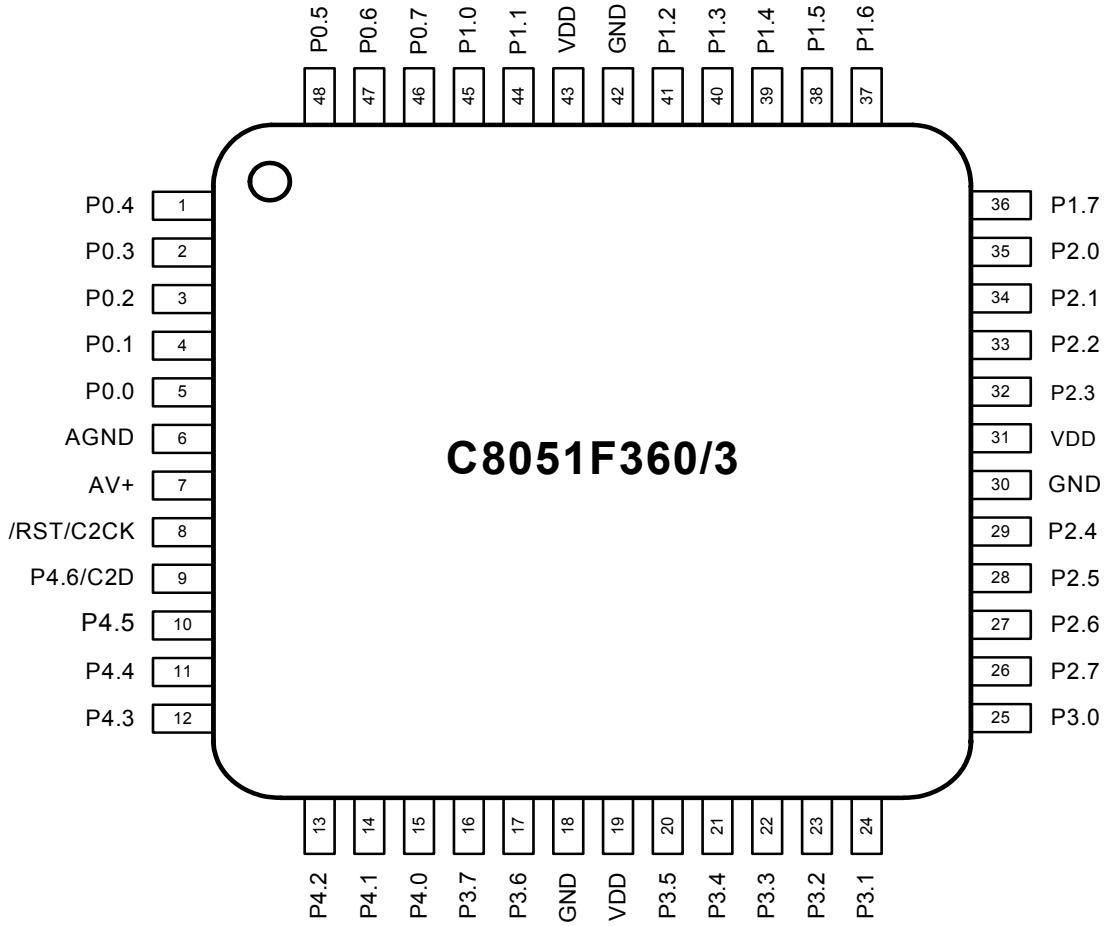


Figure 4.1. TQFP-48 Pinout Diagram (Top View)

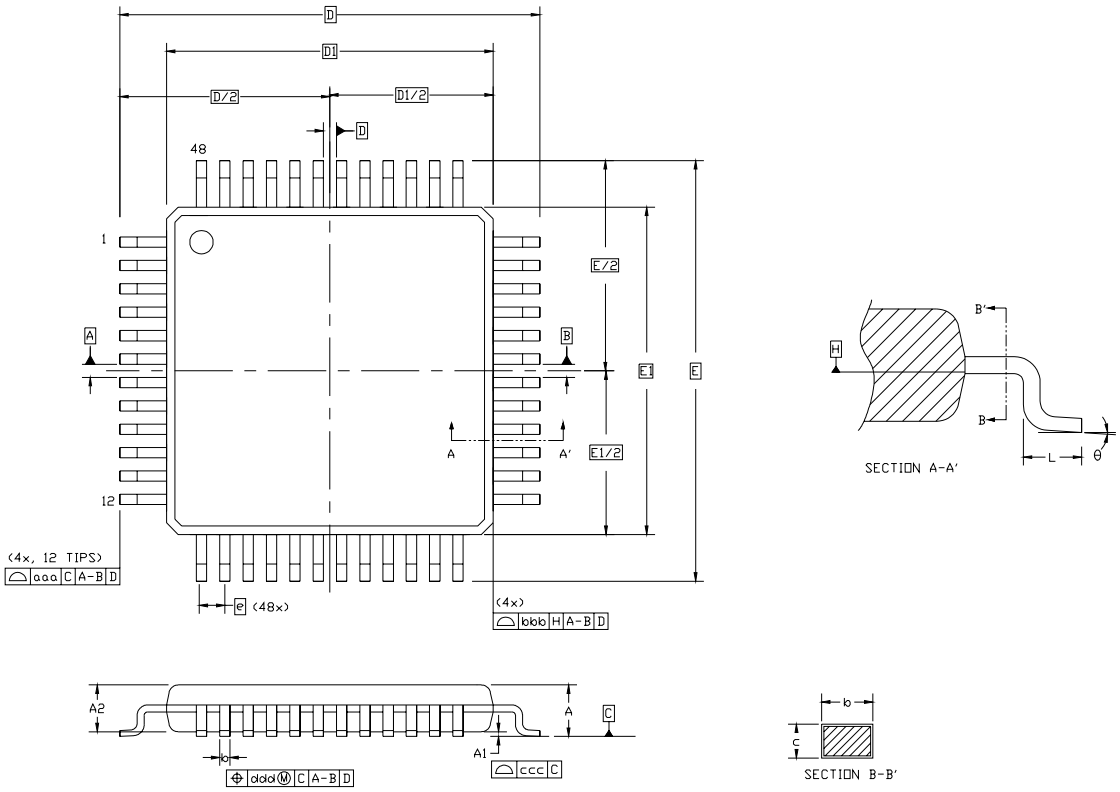


Figure 4.2. TQFP-48 Package Diagram

Table 4.2. TQFP-48 Package Dimensions

Dimension	Min	Nom	Max	Dimension	Min	Nom	Max
A	—	—	1.20	E	9.00 BSC.		
A1	0.05	—	0.15	E1	7.00 BSC.		
A2	0.95	1.00	1.05	L	0.45	0.60	0.75
b	0.17	0.22	0.27	aaa	0.20		
c	0.09	—	0.20	bbb	0.20		
D	9.00 BSC.			ccc	0.08		
D1	7.00 BSC.			ddd	0.08		
e	0.50 BSC.			θ	0°	3.5°	7°

Notes:
1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MS-026, variation ABC.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

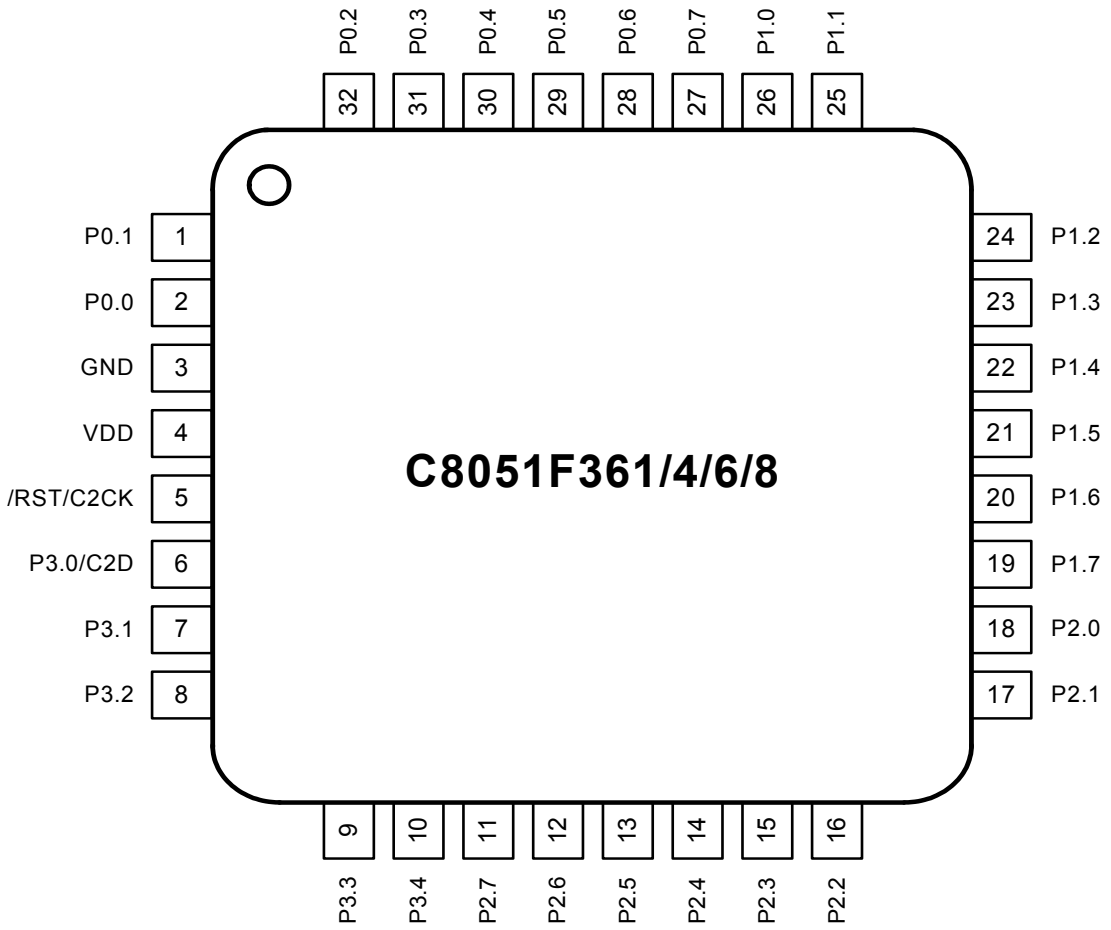


Figure 4.3. LQFP-32 Pinout Diagram (Top View)

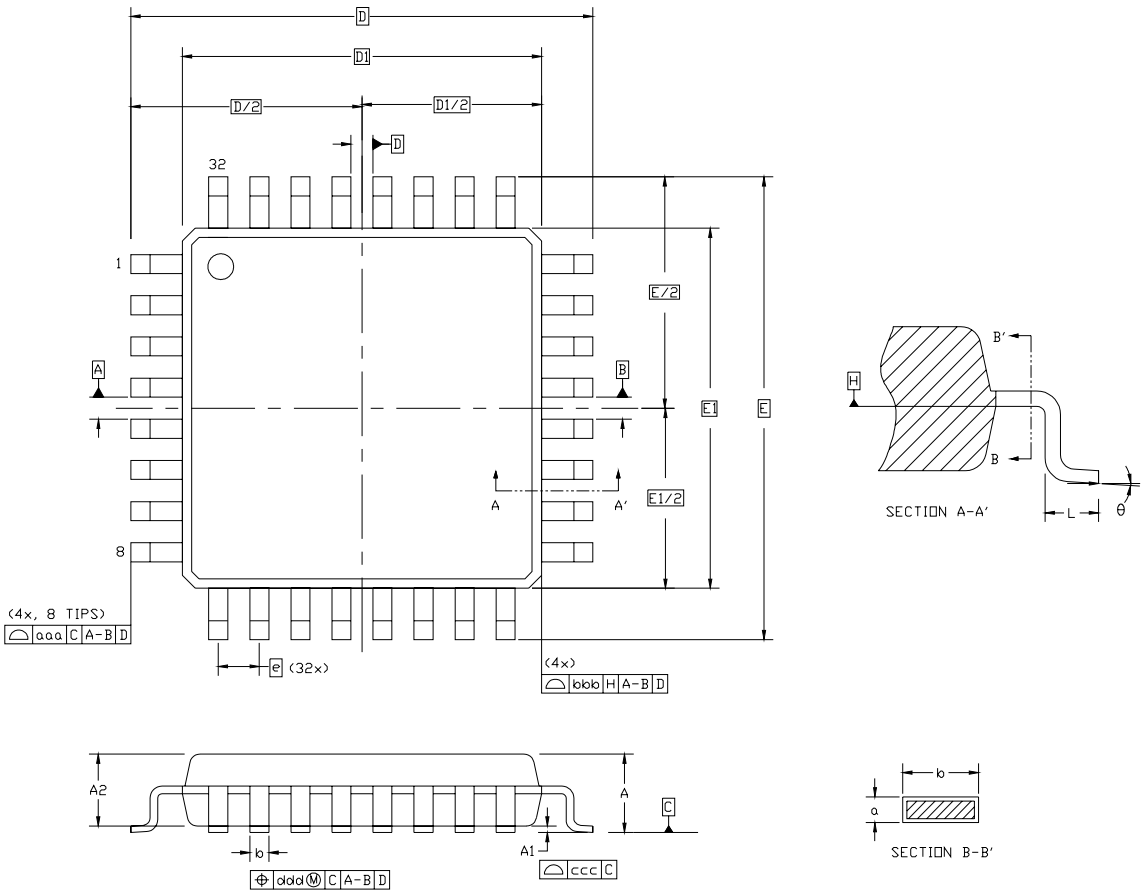


Figure 4.4. LQFP-32 Package Diagram

Table 4.3. LQFP-32 Package Dimensions

Dimension	Min	Nom	Max
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
b	0.30	0.37	0.45
c	0.09	—	0.20
D	9.00 BSC.		
D1	7.00 BSC.		
e	0.80 BSC.		

Dimension	Min	Nom	Max
E	9.00 BSC.		
E1	7.00 BSC.		
L	0.45	0.60	0.75
aaa	0.20		
bbb	0.20		
ccc	0.10		
ddd	0.20		
θ	0°	3.5°	7°

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MS-026, variation BBA.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020B specification for Small Body Components.

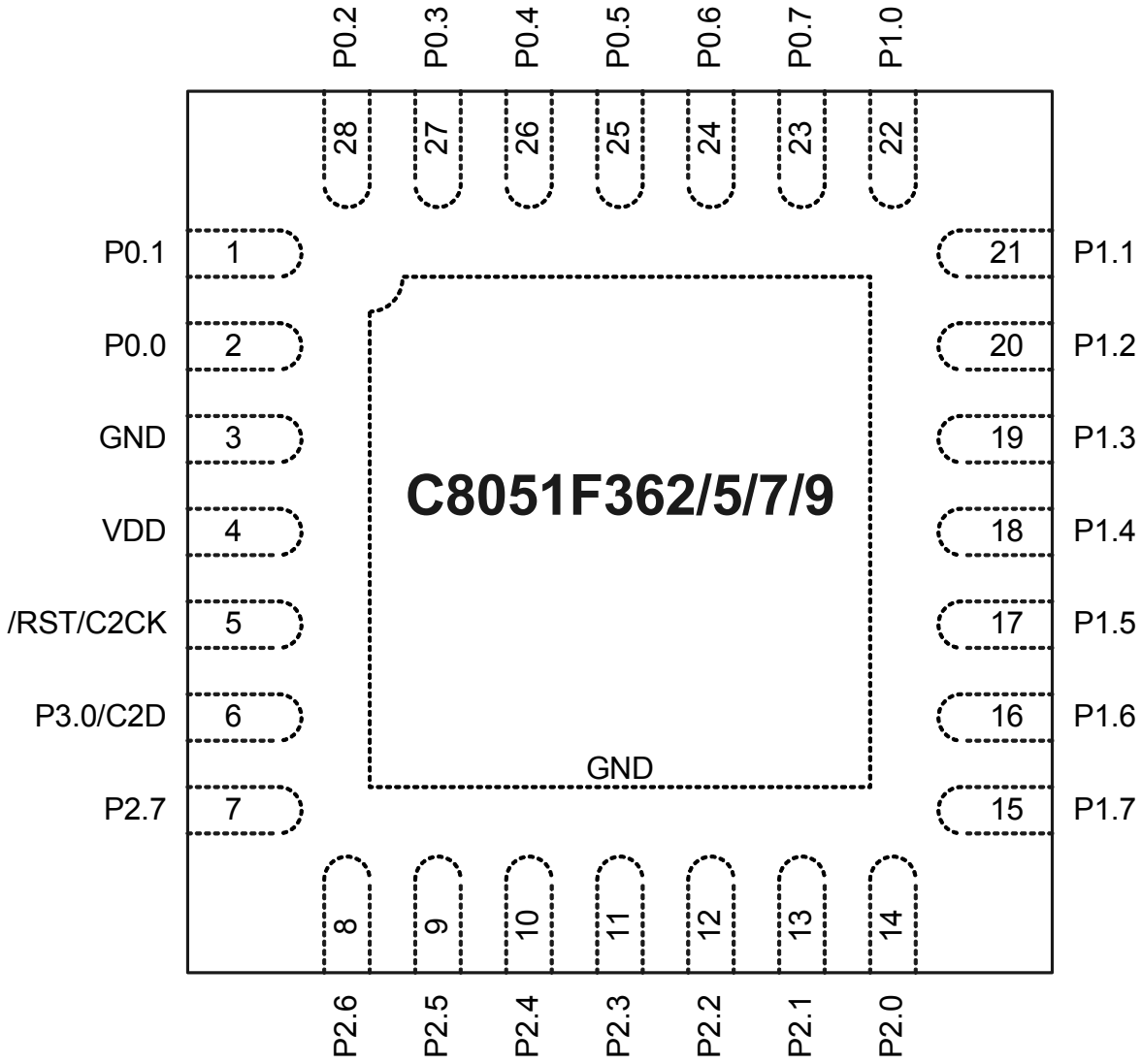


Figure 4.5. QFN-28 Pinout Diagram (Top View)

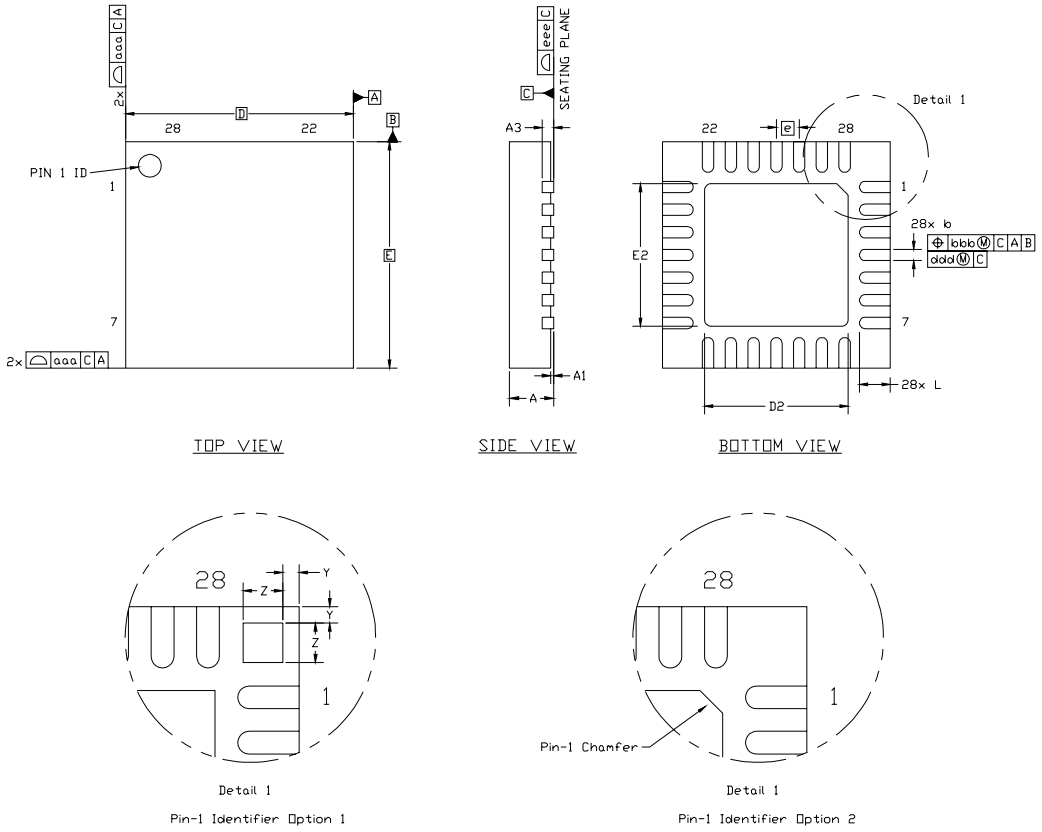


Figure 4.6. QFN-28 Package Drawing

Table 4.4. QFN-28 Package Dimensions

Dimension	Min	Nom	Max
A	0.80	0.90	1.00
A1	0.03	0.07	0.11
A3		0.25 REF	
b	0.18	0.25	0.30
D		5.00 BSC.	
D2	2.90	3.15	3.35
e		0.50 BSC.	
E		5.00 BSC.	

Dimension	Min	Nom	Max
E2	2.90	3.15	3.35
L	0.45	0.55	0.65
aaa		0.15	
bbb		0.10	
ddd		0.05	
eee		0.08	
Z		0.435	
Y		0.18	

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MO-243, variation VHHD except for custom features D2, E2, L, Z, and Y which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

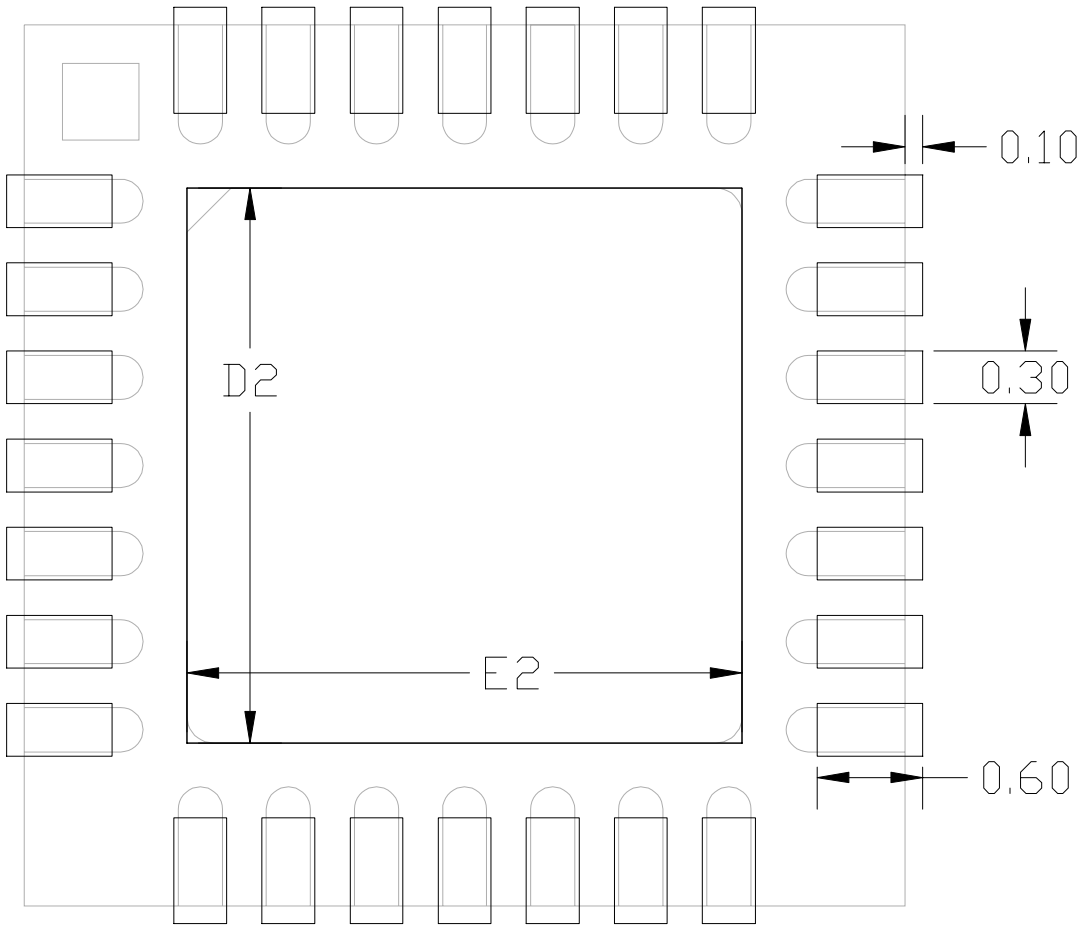


Figure 4.7. Typical QFN-28 Landing Diagram

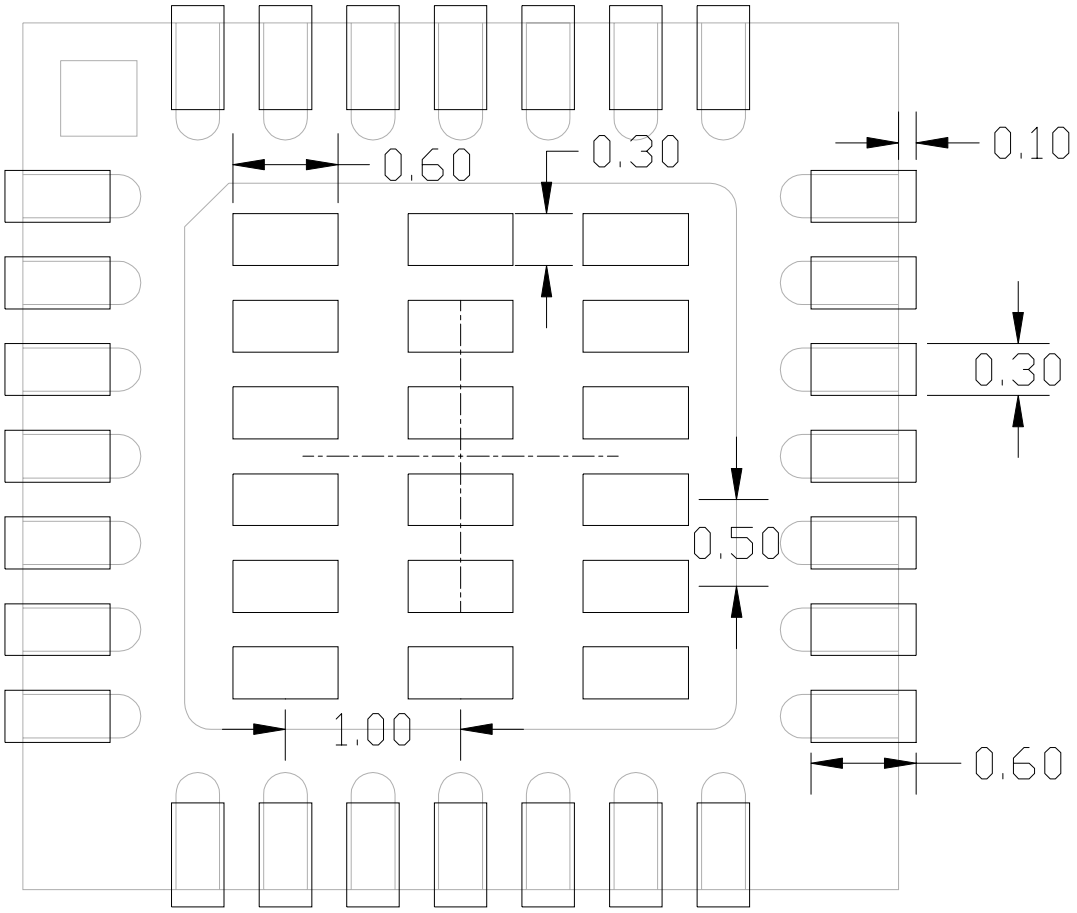


Figure 4.8. QFN-28 Solder Paste Recommendation

5. 10-Bit ADC (ADC0, C8051F360/1/2/6/7/8/9)

The ADC0 subsystem for the C8051F360/1/2/6/7/8/9 consists of two analog multiplexers (referred to collectively as AMUX0) with 23 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 P1.0-P3.4 (where available), the Temperature Sensor output, or V_{DD} with respect to P1.0-P3.4, VREF, 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'.

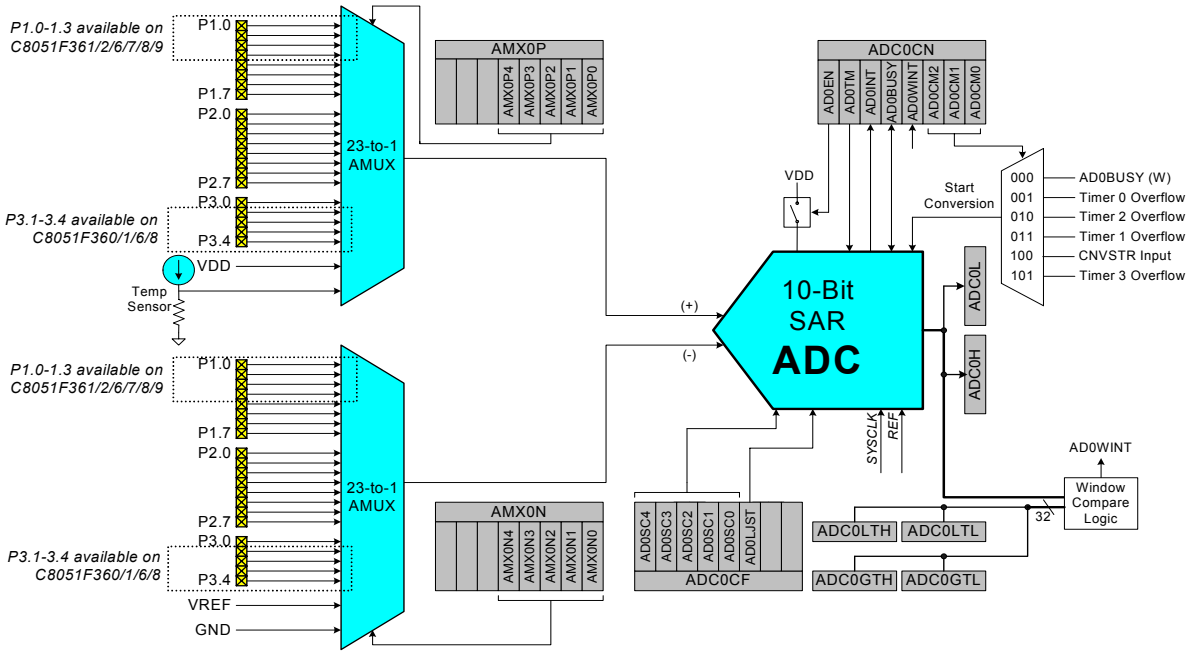


Figure 5.1. ADC0 Functional Block Diagram

5.1. Analog Multiplexer

AMUX0 selects the positive and negative inputs to the ADC. Any of the following may be selected as the positive input: the AMUX0 Port I/O inputs, the on-chip temperature sensor, or the positive power supply (V_{DD}). Any of the following may be selected as the negative input: the AMUX0 Port I/O inputs, VREF, or GND. **When GND is selected as the negative input, ADC0 operates in Single-ended Mode; all other times, ADC0 operates in Differential Mode.** The ADC0 input channels are selected in the AMX0P and AMX0N registers as described in SFR Definition 5.1 and SFR Definition 5.2.

The conversion code format differs between Single-ended and Differential modes. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit (ADC0CN.0). When in Single-ended Mode, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from '0' to VREF * 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC0H and ADC0L registers are set to '0'.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When in Differential Mode, conversion codes are represented as 10-bit signed 2's complement numbers. Inputs are measured from -VREF to VREF * 511/512. Example codes are shown below for both right-justified and left-justified data. For right-justified data, the unused MSBs of ADC0H are a sign-extension of the data word. For left-justified data, the unused LSBs in the ADC0L register are set to '0'.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
VREF x 511/512	0x01FF	0x7FC0
VREF x 256/512	0x0100	0x4000
0	0x0000	0x0000
–VREF x 256/512	0xFF00	0xC000
–VREF	0xFE00	0x8000

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to '0' the corresponding bit in register PnMDIN (for n = 0,1,2,3). To force the Crossbar to skip a Port pin, set to '1' the corresponding bit in register PnSKIP (for n = 0,1,2,3). See Section "17. Port Input/Output" on page 183 for more Port I/O configuration details.

5.2. Temperature Sensor

The typical temperature sensor transfer function is shown in Figure 5.2. The output voltage (V_{TEMP}) is the positive ADC input when the temperature sensor is selected by bits AMX0P4-0 in register AMX0P.

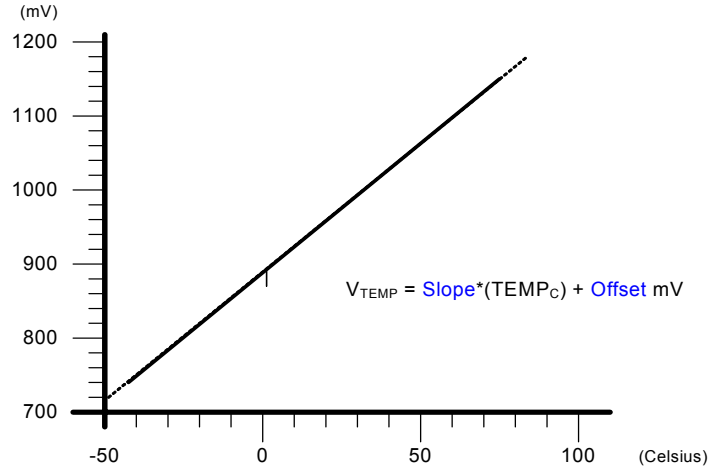


Figure 5.2. Typical Temperature Sensor Transfer Function

The uncalibrated temperature sensor output is extremely linear and suitable for relative temperature measurements (see Table 5.1 for linearity specifications). For absolute temperature measurements, gain and/or offset calibration is recommended. Typically a 1-point calibration includes the following steps:

Step 1. Control/measure the ambient temperature (this temperature must be known).
Step 2. Power the device, and delay for a few seconds to allow for self-heating.
Step 3. Perform an ADC conversion with the temperature sensor selected as the positive input and GND selected as the negative input.
Step 4. Calculate the offset and/or gain characteristics, and store these values in non-volatile memory for use with subsequent temperature sensor measurements.

Figure 5.3 shows the typical temperature sensor error assuming a 1-point calibration at 25 °C. **Note that parameters which affect ADC measurement, in particular the voltage reference value, will also affect temperature measurement.**

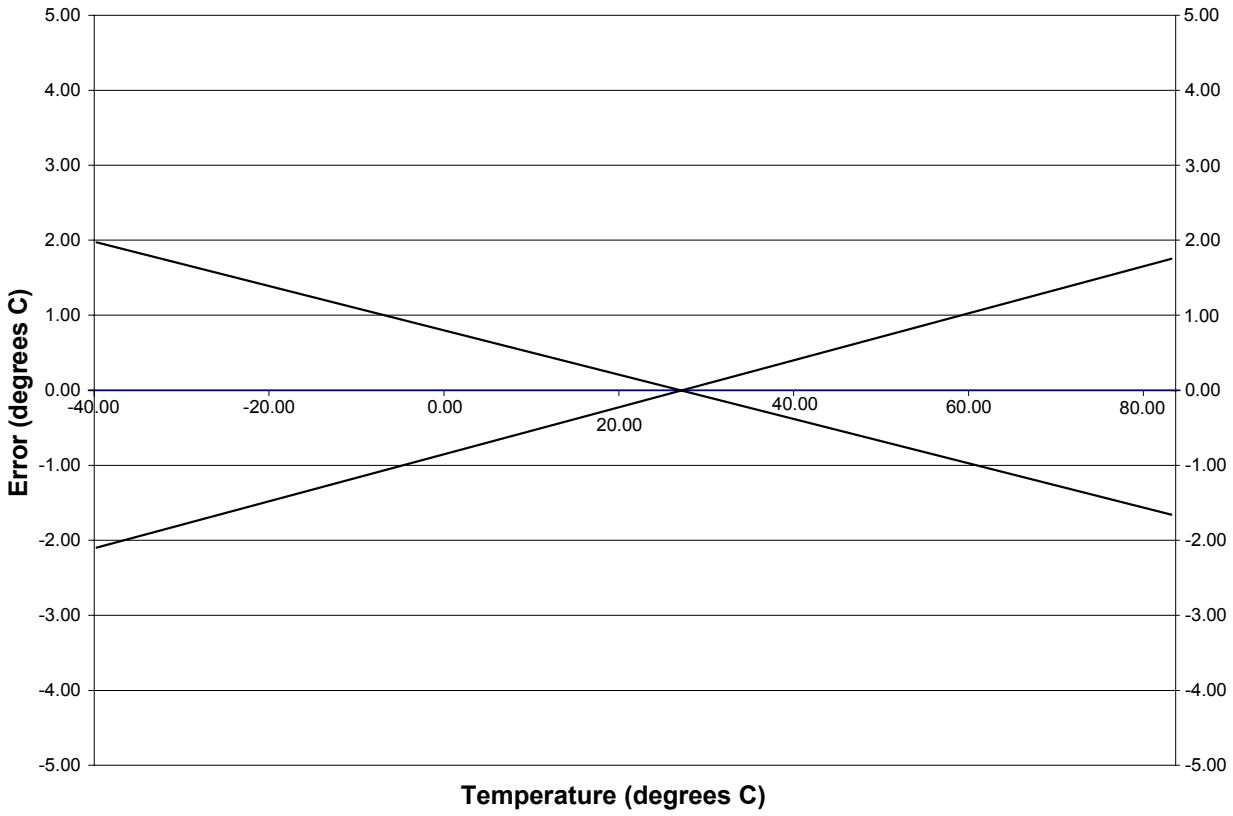


Figure 5.3. Temperature Sensor Error with 1-Point Calibration

5.3. Modes of Operation

ADC0 has a maximum conversion speed of 200 ksps. The ADC0 conversion clock is a divided version of the system clock, determined by the AD0SC bits in the ADC0CF register (system clock divided by (AD0SC + 1) for $0 \leq AD0SC \leq 31$).

5.3.1. Starting a Conversion

A conversion can be initiated in one of five ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM2-0) in register ADC0CN. Conversions may be initiated by one of the following:

1. Writing a ‘1’ to the AD0BUSY bit of register ADC0CN
2. A Timer 0 overflow (i.e., timed continuous conversions)
3. A Timer 2 overflow
4. A Timer 1 overflow
5. A rising edge on the CNVSTR input signal
6. A Timer 3 overflow

Writing a ‘1’ to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand". During conversion, the AD0BUSY bit is set to logic ‘1’ and reset to logic ‘0’ when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic ‘1’. Note that when Timer 2 or Timer 3 overflows are used as the conversion source, Low Byte overflows are used if Timer 2/3 is in 8-bit mode; High byte overflows are used if Timer 2/3 is in 16-bit mode. See Section “21. Timers” on page 247 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as Port pin P0.7 on the C8051F360 devices and Port pin P0.6 on the C8051F361/2/6/7/8/9 devices. When the CNVSTR input is used as the ADC0 conversion source, the corresponding port pin should be skipped by the Digital Crossbar. To configure the Crossbar to skip the port pin, set the appropriate bit to ‘1’ in register P0SKIP. See Section “17. Port Input/Output” on page 183 for details on Port I/O configuration.

5.3.2. Tracking Modes

According to Table 5.1, each ADC0 conversion must be preceded by a minimum tracking time for the converted result to be accurate. The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state, the ADC0 input is continuously tracked, except when a conversion is in progress. When the AD0TM bit is logic ‘1’, ADC0 operates in low-power track-and-hold mode. In this mode, each conversion is preceded by a tracking period of 3 SAR clocks (after the start-of-conversion signal). When the CNVSTR signal is used to initiate conversions in low-power tracking mode, ADC0 tracks only when CNVSTR is low; conversion begins on the rising edge of CNVSTR (see Figure 5.4). Tracking can also be disabled (shutdown) when the device is in low power standby or sleep modes. Low-power track-and-hold mode is also useful when AMUX settings are frequently changed, due to the settling time requirements described in Section “5.3.3. Settling Time Requirements” on page 53.

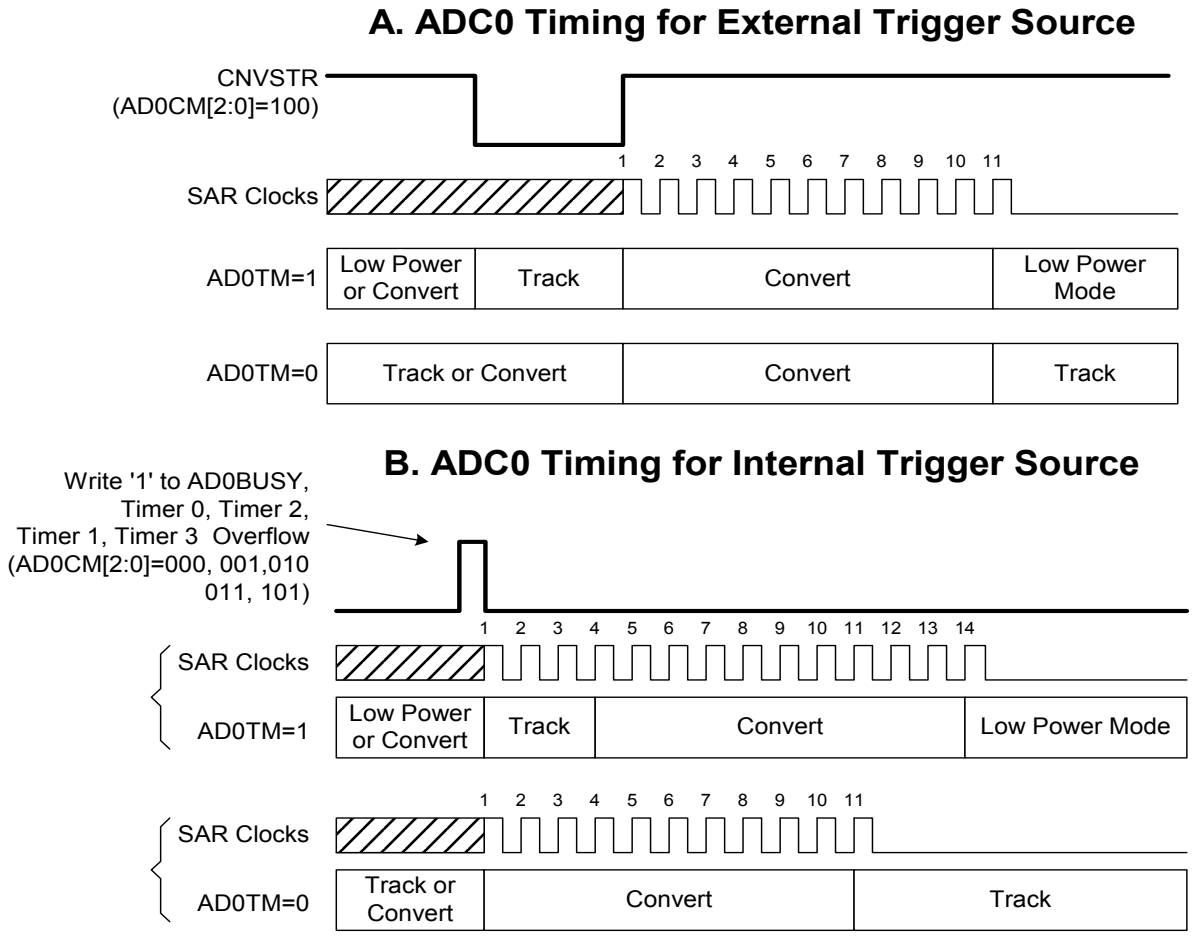


Figure 5.4. 10-Bit ADC Track and Conversion Example Timing

5.3.3. Settling Time Requirements

When the ADC0 input configuration is changed (i.e., a different AMUX0 selection is made), a minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion. In low-power tracking mode, three SAR clocks are used for tracking at the start of every conversion. For most applications, these three SAR clocks will meet the minimum tracking time requirements.

Figure 5.5 shows the equivalent ADC0 input circuits for both Differential and Single-ended modes. Notice that the equivalent time constant for both input circuits is the same. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 5.1. When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 5.1 for ADC0 minimum settling time requirements.

Equation 5.1. ADC0 Settling Time Requirements

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (10).

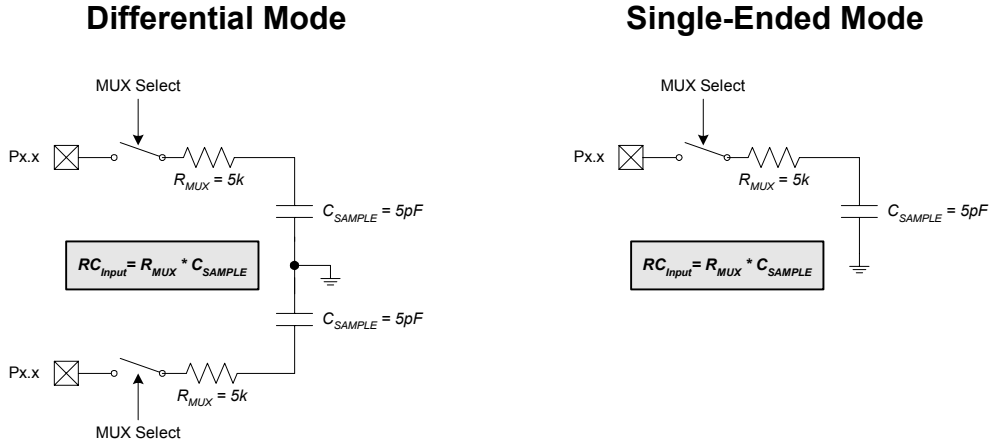


Figure 5.5. ADC0 Equivalent Input Circuits

SFR Definition 5.1. AMX0P: AMUX0 Positive Channel Select

SFR Page: all pages
SFR Address: 0xBB

R	R	R	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	AMX0P4	AMX0P3	AMX0P2	AMX0P1	AMX0P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–5: UNUSED. Read = 000b; Write = don't care.
Bits 4–0: AMX0P4–0: AMUX0 Positive Input Selection

AMX0P4-0	ADC0 Positive Input
00000(1)	P1.0(1)
00001(1)	P1.1(1)
00010(1)	P1.2(1)
00011(1)	P1.3(1)
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P2.0
01001	P2.1
01010	P2.2
01011	P2.3
01100	P2.4
01101	P2.5
01110	P2.6
01111	P2.7
10000	P3.0
10001(2)	P3.1(2)
10010(2)	P3.2(2)
10011(2)	P3.3(2)
10100(2)	P3.4(2)
10101–11101	RESERVED
11110	Temp Sensor
11111	V_{DD}

Notes:
1. Only applies to C8051F361/2/6/7/8/9 (32-pin and 28-pin); selection RESERVED on C8051F360 (48-pin) device.
2. Only applies to C8051F360/1/6/8 (48-pin and 32-pin); selection RESERVED on C8051F362/7/9 (28-pin) devices.

SFR Definition 5.2. AMX0N: AMUX0 Negative Channel Select

SFR Page: all pages
SFR Address: 0xBA

R	R	R	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	AMX0N4	AMX0N3	AMX0N2	AMX0N1	AMX0N0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–5: UNUSED. Read = 000b; Write = don't care.
Bits 4–0: AMX0N4–0: AMUX0 Negative Input Selection.
Note that when GND is selected as the Negative Input, ADC0 operates in Single-ended mode. For all other Negative Input selections, ADC0 operates in Differential mode.

AMX0N4-0	ADC0 Negative Input
00000(1)	P1.0(1)
00001(1)	P1.1(1)
00010(1)	P1.2(1)
00011(1)	P1.3(1)
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P2.0
01001	P2.1
01010	P2.2
01011	P2.3
01100	P2.4
01101	P2.5
01110	P2.6
01111	P2.7
10000	P3.0
10001(2)	P3.1(2)
10010(2)	P3.2(2)
10011(2)	P3.3(2)
10100(2)	P3.4(2)
10101–11101	RESERVED
11110	VREF
11111	GND

Notes:
1. Only applies to C8051F361/2/6/7/8/9 (32-pin and 28-pin); selection RESERVED on C8051F360 (48-pin) device.
2. Only applies to C8051F360/1/6/8 (48-pin and 32-pin); selection RESERVED on C8051F362/7/9 (28-pin) devices.

SFR Definition 5.3. ADC0CF: ADC0 Configuration

SFR Page: all pages
SFR Address: 0xBC

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AD0LJST	–	–	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–3: AD0SC4–0: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from system clock by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4–0. SAR Conversion clock requirements are given in Table 5.1.

$$AD0SC = \frac{SYSCLK}{CLK_{SAR}} - 1$$

Bit 2: AD0LJST: ADC0 Left Justify Select.
0: Data in ADC0H:ADC0L registers are right-justified.
1: Data in ADC0H:ADC0L registers are left-justified.
Bits 1–0: UNUSED. Read = 00b; Write = don't care.

SFR Definition 5.4. ADC0H: ADC0 Data Word MSB

SFR Page: all pages
SFR Address: 0xBE

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	

Bits 7–0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0: Bits 7–2 are the sign extension of Bit1. Bits 1–0 are the upper 2 bits of the 10-bit ADC0 Data Word.
For AD0LJST = 1: Bits 7–0 are the most-significant bits of the 10-bit ADC0 Data Word.

SFR Definition 5.5. ADC0L: ADC0 Data Word LSB

SFR Page: all pages
SFR Address: 0xBD

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	

Bits 7–0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the 10-bit Data Word.
For AD0LJST = 1: Bits 7–6 are the lower 2 bits of the 10-bit Data Word. Bits 5–0 will always read '0'.

SFR Definition 5.6. ADC0CN: ADC0 Control

SFR Page: all pages
SFR Address: 0xE8 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0EN	AD0TM	AD0INT	AD0BUSY	AD0WINT	AD0CM2	AD0CM1	AD0CM0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit 6: AD0TM: ADC0 Track Mode Bit.
0: Normal Track Mode: When ADC0 is enabled, tracking is continuous unless a conversion is in progress.
1: Low-power Track Mode: Tracking Defined by AD0CM2-0 bits (see below).

Bit 5: AD0INT: ADC0 Conversion Complete Interrupt Flag.
0: ADC0 has not completed a data conversion since the last time AD0INT was cleared.
1: ADC0 has completed a data conversion.

Bit 4: AD0BUSY: ADC0 Busy Bit.
Read:
0: ADC0 conversion is complete or a conversion is not currently in progress. AD0INT is set to logic ‘1’ on the falling edge of AD0BUSY.
1: ADC0 conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0CM2-0 = 000b

Bit 3: AD0WINT: ADC0 Window Compare Interrupt Flag.
0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC0 Window Comparison Data match has occurred.

Bits 2–0: AD0CM2–0: ADC0 Start of Conversion Mode Select.
When AD0TM = 0:
000: ADC0 conversion initiated on every write of ‘1’ to AD0BUSY.
001: ADC0 conversion initiated on overflow of Timer 0.
010: ADC0 conversion initiated on overflow of Timer 2.
011: ADC0 conversion initiated on overflow of Timer 1.
100: ADC0 conversion initiated on rising edge of external CNVSTR.
101: ADC0 conversion initiated on overflow of Timer 3.
11x: Reserved.
When AD0TM = 1:
000: Tracking initiated on write of ‘1’ to AD0BUSY and lasts 3 SAR clocks, followed by conversion.
001: Tracking initiated on overflow of Timer 0 and lasts 3 SAR clocks, followed by conversion.
010: Tracking initiated on overflow of Timer 2 and lasts 3 SAR clocks, followed by conversion.
011: Tracking initiated on overflow of Timer 1 and lasts 3 SAR clocks, followed by conversion.
100: ADC0 tracks only when CNVSTR input is logic low; conversion starts on rising CNVSTR edge.
101: Tracking initiated on overflow of Timer 3 and lasts 3 SAR clocks, followed by conversion.
11x: Reserved.

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.

SFR Definition 5.7. ADC0GTH: ADC0 Greater-Than Data High Byte

SFR Page: all pages
SFR Address: 0xC4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: High byte of ADC0 Greater-Than Data Word.

SFR Definition 5.8. ADC0GTL: ADC0 Greater-Than Data Low Byte

SFR Page: all pages
SFR Address: 0xC3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Low byte of ADC0 Greater-Than Data Word.

SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte

SFR Page: all pages
SFR Address: 0xC6

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	

Bits 7–0: High byte of ADC0 Less-Than Data Word.

SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte

SFR Page: all pages
SFR Address: 0xC5

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	

Bits 7–0: Low byte of ADC0 Less-Than Data Word.

5.4.1. Window Detector In Single-Ended Mode

Figure 5.6 shows two example window comparisons for right-justified, single-ended data, with ADC0LTH:ADC0LTL = 0x0080 (128d) and ADC0GTH:ADC0GTL = 0x0040 (64d). In single-ended mode, the input voltage can range from '0' to VREF x (1023/1024) with respect to GND, and is represented by a 10-bit unsigned integer value. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0040 < ADC0H:ADC0L < 0x0080). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0040 or ADC0H:ADC0L > 0x0080). Figure 5.7 shows an example using left-justified data with the same comparison values.

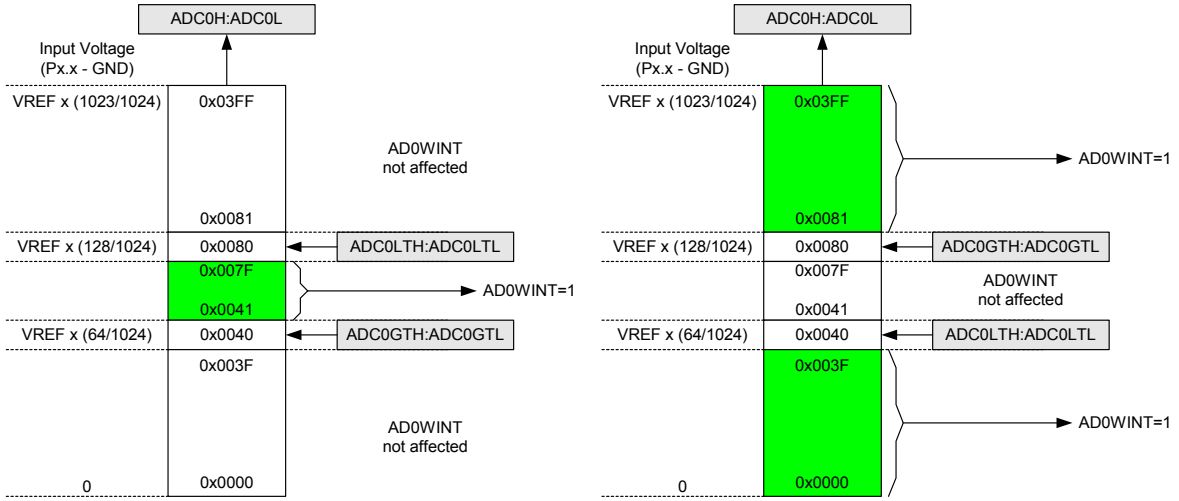


Figure 5.6. ADC Window Compare Example: Right-Justified Single-Ended Data

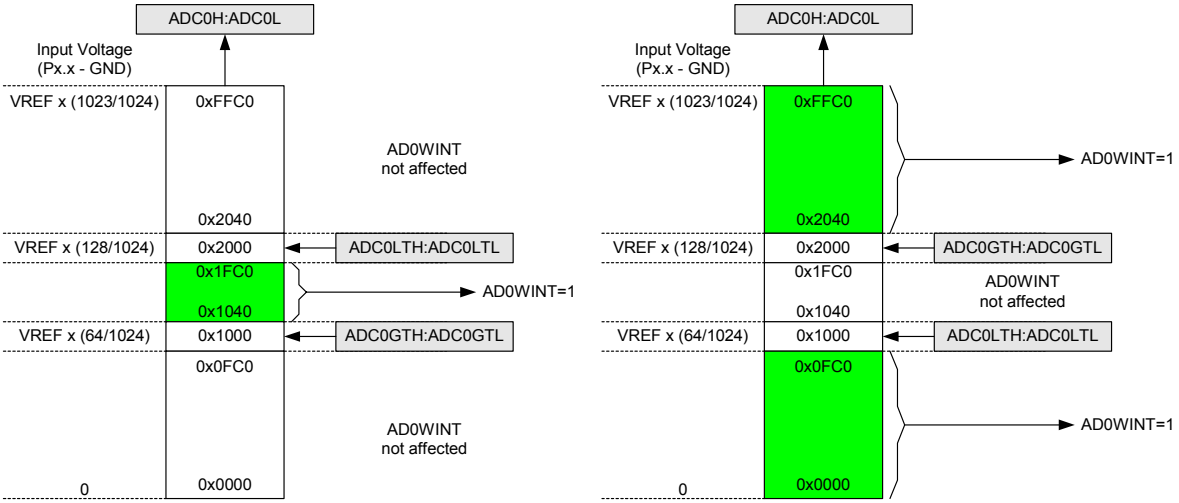


Figure 5.7. ADC Window Compare Example: Left-Justified Single-Ended Data

5.4.2. Window Detector In Differential Mode

Figure 5.8 shows two example window comparisons for right-justified, differential data, with ADC0LTH:ADC0LTL = 0x0040 (+64d) and ADC0GTH:ADC0GTH = 0xFFFF (-1d). In differential mode, the measurable voltage between the input pins is between -VREF and VREF*(511/512). Output codes are represented as 10-bit 2's complement signed integers. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0xFFFF (-1d) < ADC0H:ADC0L < 0x0040 (64d)). In the right example, an AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0xFFFF (-1d) or ADC0H:ADC0L > 0x0040 (+64d)). Figure 5.9 shows an example using left-justified data with the same comparison values.

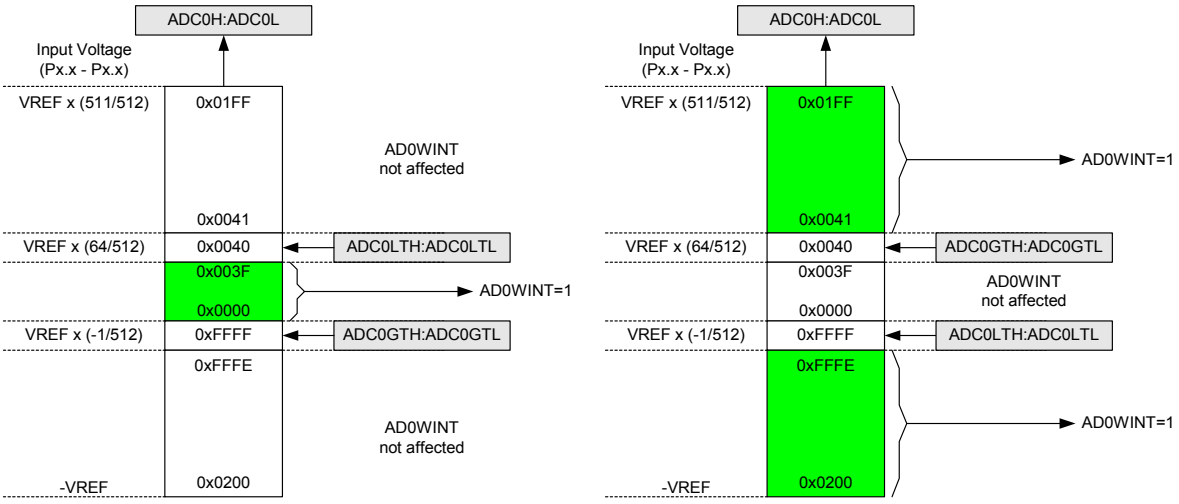


Figure 5.8. ADC Window Compare Example: Right-Justified Differential Data

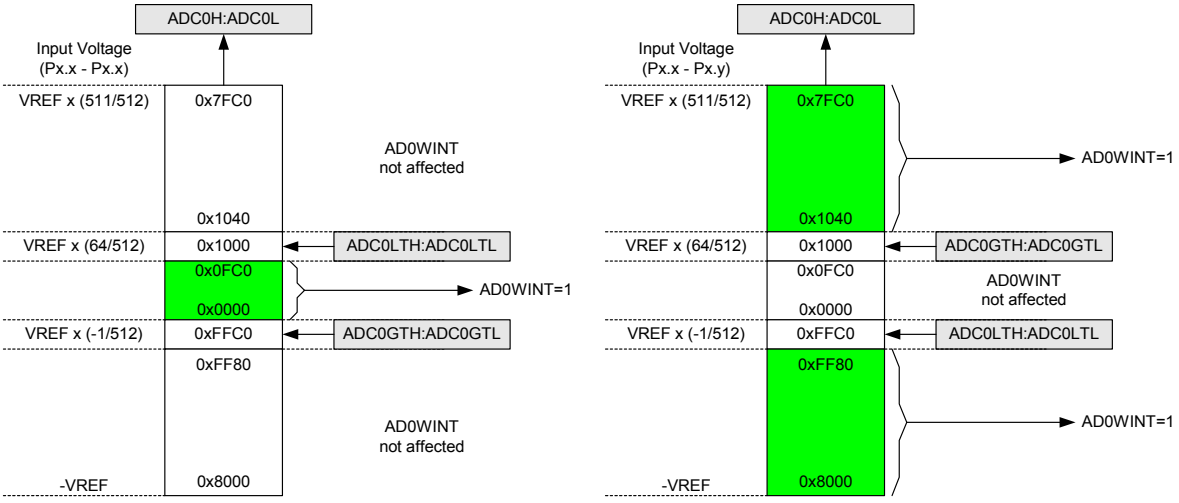


Figure 5.9. ADC Window Compare Example: Left-Justified Differential Data

Table 5.1. ADC0 Electrical Characteristics

V_{DD} = 3.0 V, VREF = 2.40 V (REFSL=0), –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			10		bits
Integral Nonlinearity		—	±0.5	±1	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	±0.5	±1	LSB
Offset Error		–12	3	12	LSB
Full Scale Error	Differential mode	–5	1	5	LSB
Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion		53	58	—	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	–75	—	dB
Spurious-Free Dynamic Range		—	75	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	3	MHz
Conversion Time in SAR Clocks		13	—	—	clocks
Track/Hold Acquisition Time		300	—	—	ns
Throughput Rate		—	—	200	ksps
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ – GND) Differential (AIN+ – AIN–)	0 –VREF	— —	VREF VREF	V V
Absolute Pin Voltage with respect to GND	Single Ended or Differential	0	—	V_{DD}	V
Input Capacitance		—	5	—	pF
Temperature Sensor					
Linearity*		—	±0.2	—	°C
Slope		—	2.18	—	mV/ºC
Slope Error*		—	±172	—	µV/ºC
Offset	(Temp = 0 °C)	—	802	—	mV
Offset Error*		—	±18.5	—	mV
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Operating Mode, 200 ksps	—	450	900	µA
Power Supply Rejection		—	3	—	mV/V

***Note:** Represents one standard deviation from the mean. Includes ADC offset, gain, and linearity variations.

6. 10-Bit Current Mode DAC (IDA0, C8051F360/1/2/6/7/8/9)

The C8051F360/1/2/6/7/8/9 devices include a 10-bit current-mode Digital-to-Analog Converter (IDAC). The maximum current output of the IDAC can be adjusted for three different current settings; 0.5 mA, 1 mA, and 2 mA. The IDAC is enabled or disabled with the IDA0EN bit in the IDA0 Control Register (see SFR Definition 6.1). When IDA0EN is set to '0', the IDAC port pin (P0.4 for C8051F360, P0.1 for C8051F361/2/6/7/8/9) behaves as a normal GPIO pin. When IDA0EN is set to '1', the digital output drivers and weak pullup for the IDAC pin are automatically disabled, and the pin is connected to the IDAC output. An internal bandgap bias generator is used to generate a reference current for the IDAC whenever it is enabled. When using the IDAC, the appropriate bit in the P0SKIP register should be set to '1' to force the Crossbar to skip the IDAC pin.

6.1. IDA0 Output Scheduling

IDA0 features a flexible output update mechanism which allows for seamless full-scale changes and supports jitter-free updates for waveform generation. Three update modes are provided, allowing IDAC output updates on a write to IDA0H, on a Timer overflow, or on an external pin edge.

6.1.1. Update Output On-Demand

In its default mode (IDA0CN.[6:4] = '111') the IDA0 output is updated "on-demand" on a write to the high-byte of the IDA0 data register (IDA0H). It is important to note that writes to IDA0L are held in this mode, and have no effect on the IDA0 output until a write to IDA0H takes place. If writing a full 10-bit word to the IDAC data registers, the 10-bit data word is written to the low byte (IDA0L) and high byte (IDA0H) data registers. **Data is latched into IDA0 after a write to the IDA0H register, so the write sequence should be IDA0L followed by IDA0H** if the full 10-bit resolution is required. The IDAC can be used in 8-bit mode by initializing IDA0L to the desired value (typically 0x00), and writing data to only IDA0H (see Section 6.2 for information on the format of the 10-bit IDAC data word within the 16-bit SFR space).

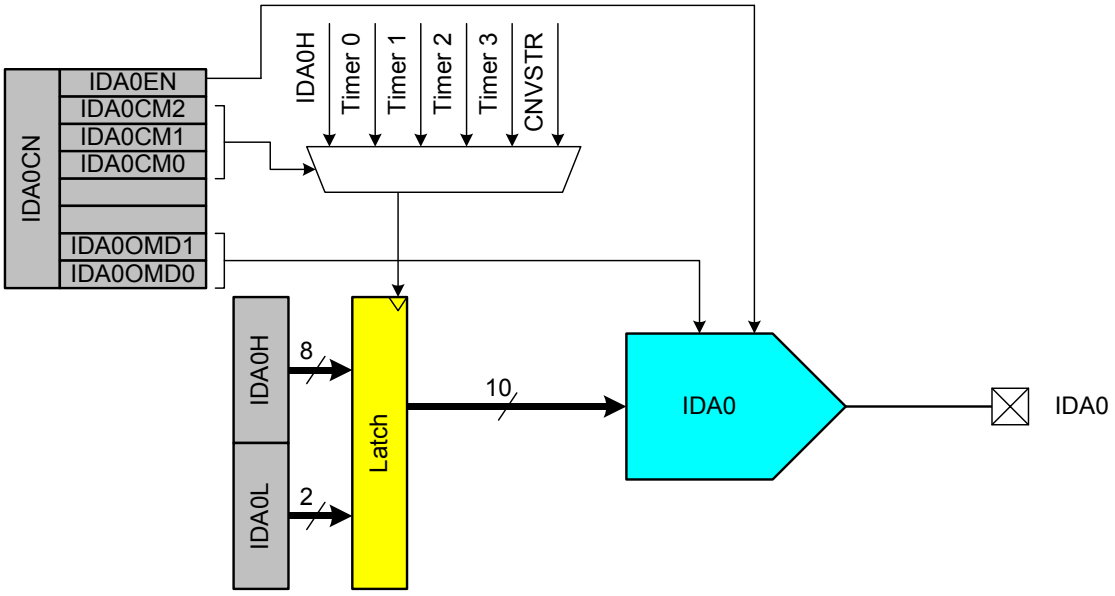


Figure 6.1. IDA0 Functional Block Diagram

6.1.2. Update Output Based on Timer Overflow

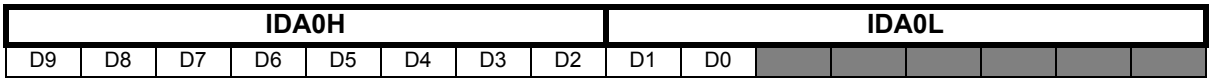
Similar to the ADC operation, in which an ADC conversion can be initiated by a timer overflow independently of the processor, the IDAC outputs can use a Timer overflow to schedule an output update event. This feature is useful in systems where the IDAC is used to generate a waveform of a defined sampling rate by eliminating the effects of variable interrupt latency and instruction execution on the timing of the IDAC output. When the IDA0CM bits (IDA0CN.[6:4]) are set to '000', '001', '010' or '011', writes to both IDAC data registers (IDA0L and IDA0H) are held until an associated Timer overflow event (Timer 0, Timer 1, Timer 2 or Timer 3, respectively) occurs, at which time the IDA0H:IDA0L contents are copied to the IDAC input latches, allowing the IDAC output to change to the new value.

6.1.3. Update Output Based on CNVSTR Edge

The IDAC output can also be configured to update on a rising edge, falling edge, or both edges of the external CNVSTR signal. When the IDA0CM bits (IDA0CN.[6:4]) are set to '100', '101', or '110', writes to both IDAC data registers (IDA0L and IDA0H) are held until an edge occurs on the CNVSTR input pin. The particular setting of the IDA0CM bits determines whether IDAC outputs are updated on rising, falling, or both edges of CNVSTR. When a corresponding edge occurs, the IDA0H:IDA0L contents are copied to the IDAC input latches, allowing the IDAC output to change to the new value.

6.2. IDAC Output Mapping

The IDAC data registers (IDA0H and IDA0L) are left-justified, meaning that the eight MSBs of the IDAC output word are mapped to bits 7–0 of the IDA0H register, and the two LSBs of the IDAC output word are mapped to bits 7 and 6 of the IDA0L register. The data word mapping for the IDAC is shown in Figure 6.2.



Input Data Word (D9–D0)	Output Current IDA0OMD[1:0] = '1x'	Output Current IDA0OMD[1:0] = '01'	Output Current IDA0OMD[1:0] = '00'
0x000	0 mA	0 mA	0 mA
0x001	1/1024 x 2 mA	1/1024 x 1 mA	1/1024 x 0.5 mA
0x200	512/1024 x 2 mA	512/1024 x 1 mA	512/1024 x 0.5 mA
0x3FF	1023/1024 x 2 mA	1023/1024 x 1 mA	1023/1024 x 0.5 mA

Figure 6.2. IDA0 Data Word Mapping

The full-scale output current of the IDAC is selected using the IDA0OMD bits (IDA0CN[1:0]). By default, the IDAC is set to a full-scale output current of 2 mA. The IDA0OMD bits can also be configured to provide full-scale output currents of 1 mA or 0.5 mA, as shown in SFR Definition 6.1.

SFR Definition 6.1. IDA0CN: IDA0 Control

SFR Page: all pages
SFR Address: 0xB9

R/W	R/W	R/W	R/W	R	R	R/W	R/W	Reset Value
IDA0EN	IDA0CM			–	–	IDA0OMD		01110010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: IDA0EN: IDA0 Enable.
0: IDA0 Disabled.
1: IDA0 Enabled.

Bits 6–4: IDA0CM[2:0]: IDA0 Update Source Select bits.
000: DAC output updates on Timer 0 overflow.
001: DAC output updates on Timer 1 overflow.
010: DAC output updates on Timer 2 overflow.
011: DAC output updates on Timer 3 overflow.
100: DAC output updates on rising edge of CNVSTR.
101: DAC output updates on falling edge of CNVSTR.
110: DAC output updates on any edge of CNVSTR.
111: DAC output updates on write to IDA0H. (default)

Bits 3–2: UNUSED. Read = 00b. Write = don't care.

Bits 1–0: IDA0OMD[1:0]: IDA0 Output Mode Select bits.
00: 0.5 mA full-scale output current.
01: 1.0 mA full-scale output current.
1x: 2.0 mA full-scale output current. (default)

SFR Definition 6.2. IDA0H: IDA0 Data Word MSB

SFR Page: all pages
SFR Address: 0x97

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	

Bits 7–0: IDA0 Data Word High-Order Bits.
Bits 7–0 are the most-significant bits of the 10-bit IDA0 Data Word.

SFR Definition 6.3. IDA0L: IDA0 Data Word LSB

SFR Page: all pages
SFR Address: 0x96

R/W	R/W	R	R	R	R	R	R	Reset Value
		—	—	—	—	—	—	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: IDA0 Data Word Low-Order Bits.
Lower 2 bits of the 10-bit Data Word.
Bits 5–0: UNUSED. Read = 000000b, Write = don't care.

Table 6.1. IDAC Electrical Characteristics

–40 to +85 °C, V_{DD} = 3.0 V Full-scale output current set to 2 mA unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Static Performance					
Resolution			10		bits
Integral Nonlinearity		—	±0.5	±2	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	±0.5	±1	LSB
Output Compliance Range		—	—	V_{DD} – 1.2	V
Offset Error		—	0	—	LSB
Full Scale Error	2 mA Full Scale Output Current	–15	0	15	LSB
Full Scale Error Tempco		—	30	—	ppm/°C
V_{DD} Power Supply Rejection Ratio		—	6.5	—	µA/V
Dynamic Performance					
Output Settling Time to 1/2 LSB	IDA0H:L = 0x3FF to 0x000	—	5	—	µs
Startup Time		—	5	—	µs
Gain Variation	1 mA Full Scale Output Current 0.5 mA Full Scale Output Current	— —	±1 ±1	— —	% %
Power Consumption					
Power Supply Current (V_{DD} supplied to IDAC)	2 mA Full Scale Output Current 1 mA Full Scale Output Current 0.5 mA Full Scale Output Current	— — —	2140 1140 640	— — —	µA µA µA

7. Voltage Reference (C8051F360/1/2/6/7/8/9)

The Voltage reference MUX on the C8051F360/1/2/6/7/8/9 devices is configurable to use an externally connected voltage reference, the internal reference voltage generator, or the V_{DD} power supply voltage (see Figure 7.1). The REFSL bit in the Reference Control register (REF0CN) selects the reference source. For an external source or the internal reference, REFSL should be set to '0'. To use V_{DD} as the reference source, REFSL should be set to '1'.

The BIASE bit enables the internal voltage bias generator, which is used by the ADC, Temperature Sensor, internal oscillators, and Current DAC. This bias is enabled when any of the aforementioned peripherals are enabled. The bias generator may be enabled manually by writing a '1' to the BIASE bit in register REF0CN; see SFR Definition 7.1 for REF0CN register details. The electrical specifications for the voltage reference circuit are given in Table 7.1.

The internal voltage reference circuit consists of a 1.2 V, temperature stable bandgap voltage reference generator and a gain-of-two output buffer amplifier. The internal voltage reference can be driven out on the VREF pin by setting the REFBE bit in register REF0CN to a '1' (see SFR Definition 7.1). The maximum load seen by the VREF pin must be less than 200 µA to GND. When using the internal voltage reference, bypass capacitors of 0.1 µF and 4.7 µF are recommended from the VREF pin to GND. If the internal reference is not used, the REFBE bit should be cleared to '0'. Electrical specifications for the internal voltage reference are given in Table 7.1.

Important Note about the VREF Pin: Port pin P0.3 on the C8051F360 device and P0.0 on C8051F361/2/6/7/89 devices is used as the external VREF input and as an output for the internal VREF. When using either an external voltage reference or the internal reference circuitry, the port pin should be configured as an analog pin, and skipped by the Digital Crossbar. To configure the port pin as an analog pin, set the appropriate bit to '0' in register P0MDIN. To configure the Crossbar to skip the VREF port pin, set the appropriate bit to '1' in register P0SKIP. Refer to Section "17. Port Input/Output" on page 183 for

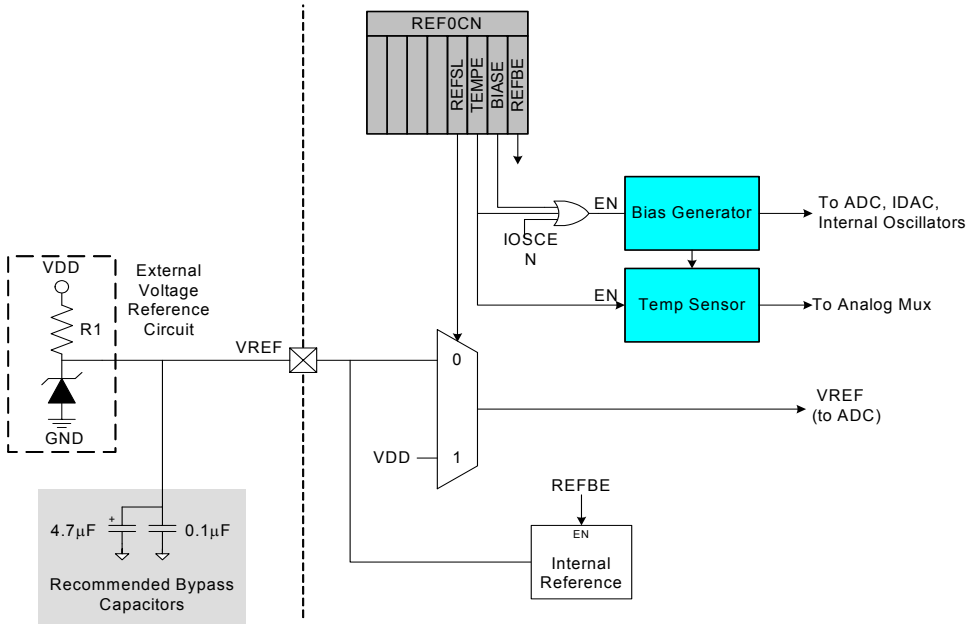


Figure 7.1. Voltage Reference Functional Block Diagram

complete Port I/O configuration details. The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data.

SFR Definition 7.1. REF0CN: Reference Control

SFR Page: all pages
SFR Address: 0xD1

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	REFSL	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–4: UNUSED. Read = 0000b; Write = don't care.
Bit 3: REFSL: Voltage Reference Select.
This bit selects the source for the internal voltage reference.
0: VREF pin used as voltage reference.
1: V_{DD} used as voltage reference.
Bit 2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.
Bit 1: BIASE: Internal Analog Bias Generator Enable Bit.
0: Internal Bias Generator off.
1: Internal Bias Generator on.
Bit 0: REFBE: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer disabled.
1: Internal Reference Buffer enabled. Internal voltage reference driven on the VREF pin.

Table 7.1. Voltage Reference Electrical Characteristics

V_{DD} = 3.0 V; –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal Reference (REFBE = 1)					
Output Voltage	25 °C ambient	2.35	2.42	2.50	V
VREF Short-Circuit Current		—	—	10	mA
VREF Temperature Coefficient		—	25	—	ppm/°C
Load Regulation	Load = 0 to 200 µA to AGND	—	3	—	µV/µA
VREF Turn-on Time 1	4.7 µF tantalum, 0.1 µF ceramic bypass	—	7.5	—	ms
VREF Turn-on Time 2	0.1 µF ceramic bypass	—	200	—	µs
Power Supply Rejection		—	1.4	—	mV/V
External Reference (REFBE = 0)					
Input Voltage Range		0	—	V_{DD}	V
Input Current	Sample Rate = 200 ksps; VREF = 3.0 V	—	3	—	µA
Power Specifications					
ADC Bias Generator	BIASE = ‘1’ or AD0EN = ‘1’ or IOSCEN = ‘1’	—	100	150	µA
Reference Bias Generator	REFBE = ‘1’ or TEMPE = ‘1’ or IDA0EN = ‘1’	—	30	50	µA

8. Comparators

C8051F36x devices include two on-chip programmable voltage comparators, Comparator0 and Comparator1, shown in Figure 8.1 and Figure 8.2 (**Note:** the port pin Comparator inputs differ between C8051F36x devices. The first Port I/O pin shown is for C8051F360/3 devices).

The comparators offer programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0 and CP1), or an asynchronous “raw” output (CP0A and CP1A). The asynchronous CP0A and CP1A signals are available even when the system clock is not active. This allows the Comparators to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator outputs may be configured as open drain or push-pull (see Section “17.2. Port I/O Initialization” on page 187). Comparator0 may also be used as a reset source (see Section “12.5. Comparator0 Reset” on page 131).

The Comparator inputs are selected in the CPT0MX and CPT1MX registers (SFR Definition 8.2 and SFR Definition 8.5). The CMXnP1–CMXnP0 bits select the Comparator positive input; the CMXnN1–CMXnN0 bits select the Comparator 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 “17.3. General Purpose Port I/O” on page 190).

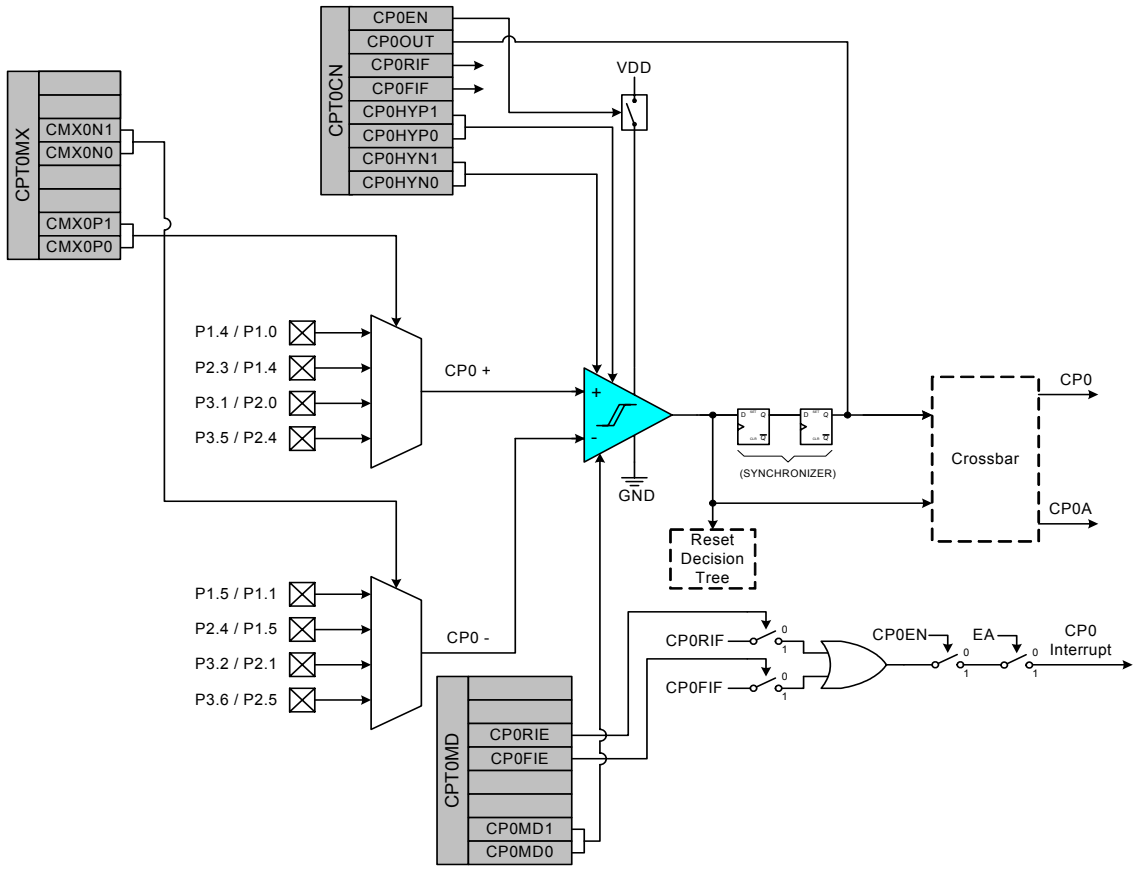


Figure 8.1. Comparator0 Functional Block Diagram

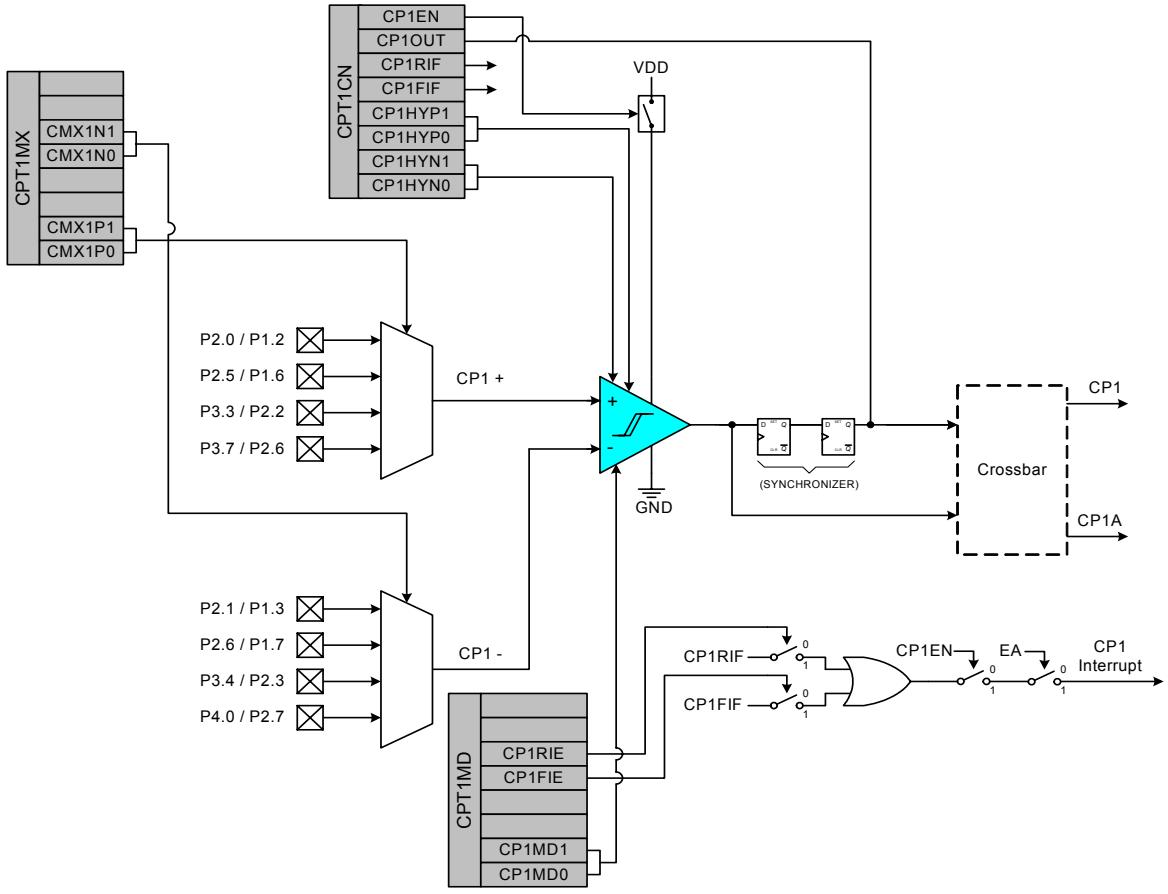


Figure 8.2. Comparator1 Functional Block Diagram

A 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 outputs are available asynchronous or synchronous to the system clock; the asynchronous outputs are available even in STOP mode (with no system clock active). When disabled, the Comparator outputs (if assigned to a Port I/O pin via the Crossbar) default to the logic low state, and their supply current falls to less than 100 nA. See Section “17.1. Priority Crossbar Decoder” on page 185 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 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 and CPT1MD registers (see SFR Definition 8.3 and SFR Definition 8.6). Selecting a longer response time reduces the Comparator supply current. See Table 8.1 for complete timing and power consumption specifications.

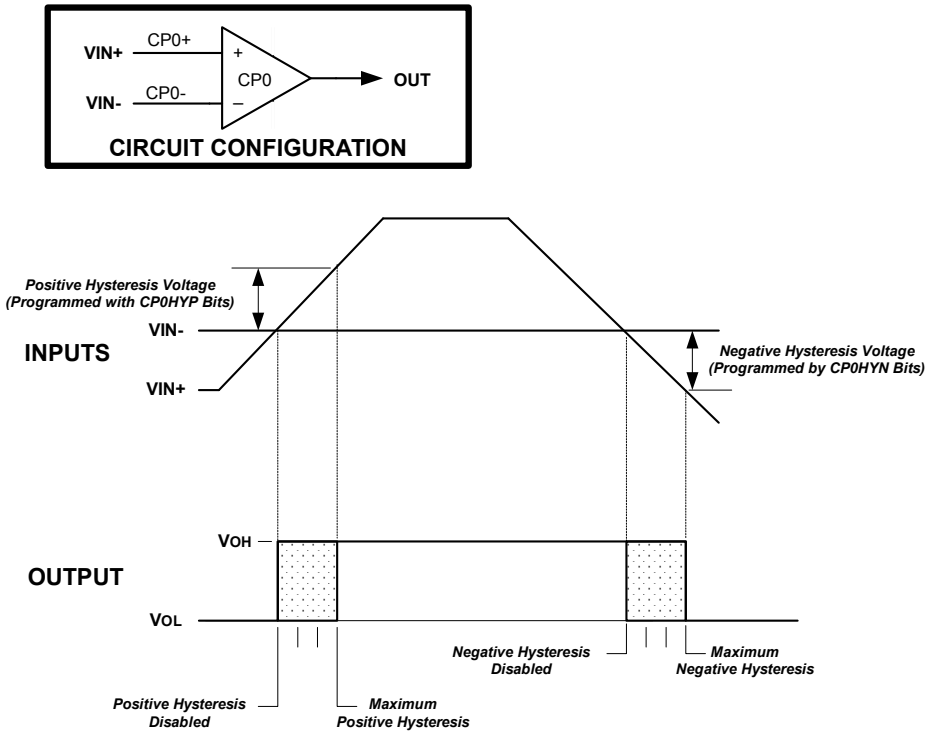


Figure 8.3. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via the Comparator Control registers CPT0CN and CPT1CN. 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 registers CPT0CN and CPT1CN (shown in SFR Definition 8.1 and SFR Definition 8.4). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN and CP1HYN bits. As shown in Figure 8.3, 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 and CP1HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see Section “10. Interrupt Handler” on page 107). The CP0FIF or CP1FIF flag is set to logic ‘1’ upon a Comparator falling-edge occurrence, and the CP0RIF or CP1RIF flag is set to logic ‘1’ upon the Comparator rising-edge occurrence. Once set, these bits remain set until cleared by software. The Comparator rising-edge interrupt mask is enabled by setting CP0RIE or CP1RIE to a logic ‘1’. The Comparator falling-edge interrupt mask is enabled by setting CP0FIE or CP1FIE to a logic ‘1’.

The output state of the Comparator can be obtained at any time by reading the CP0OUT or CP1OUT bit. The Comparator is enabled by setting the CP0EN or CP1EN bit to logic ‘1’, and is disabled by clearing this bit to logic ‘0’.

Note that false rising edges and falling edges can be detected when the comparator is first powered on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic ‘0’ a short time after the comparator is enabled or its mode bits have been changed. This Power Up Time is specified in Table 8.1 on page 79.

SFR Definition 8.1. CPT0CN: Comparator0 Control

SFR Page: all pages
SFR Address: 0x9B

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP1	CP0HYP0	CP0HYN1	CP0HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CP0EN: Comparator0 Enable Bit.
0: Comparator0 Disabled.
1: Comparator0 Enabled.

Bit 6: CP0OUT: Comparator0 Output State Flag.
0: Voltage on CP0+ < CP0–.
1: Voltage on CP0+ > CP0–.

Bit 5: CP0RIF: Comparator0 Rising-Edge Flag. Must be cleared by software.
0: No Comparator0 Rising Edge has occurred since this flag was last cleared.
1: Comparator0 Rising Edge has occurred.

Bit 4: CP0FIF: Comparator0 Falling-Edge Flag. Must be cleared by software.
0: No Comparator0 Falling-Edge has occurred since this flag was last cleared.
1: Comparator0 Falling-Edge has occurred.

Bits 3–2: CP0HYP1–0: Comparator0 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 20 mV.

Bits 1–0: CP0HYN1–0: Comparator0 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 20 mV.

SFR Definition 8.2. CPT0MX: Comparator0 MUX Selection

SFR Page: all pages
SFR Address: 0x9F

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	CMX0N1	CMX0N0	–	–	CMX0P1	CMX0P0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 11b, Write = don't care.
Bits 5–4: CMX0N1–CMX0N0: Comparator0 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator0 negative input.

CMX0N1	CMX0N0	C8051F360/3	C8051F361/2/4/5/6/7/8/9
		Negative Input	Negative Input
0	0	P1.5	P1.1
0	1	P2.4	P1.5
1	0	P3.2	P2.1
1	1	P3.6	P2.5

Bits 3–2: UNUSED. Read = 11b, Write = don't care.
Bits 1–0: CMX0P3–CMX0P0: Comparator0 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator0 positive input.

CMX0P1	CMX0P0	C8051F360/3	C8051F361/2/4/5/6/7/8/9
		Positive Input	Positive Input
0	0	P1.4	P1.0
0	1	P2.3	P1.4
1	0	P3.1	P2.0
1	1	P3.5	P2.4

SFR Definition 8.3. CPT0MD: Comparator0 Mode Selection

SFR Page: all pages
SFR Address: 0x9D

R	R	R/W	R/W	R	R	R/W	R/W	Reset Value
–	–	CP0RIE	CP0FIE	–	–	CP0MD1	CP0MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 00b, Write = don't care.
Bit 5: CP0RIE: Comparator0 Rising-Edge Interrupt Enable.
0: Comparator0 Rising-edge interrupt disabled.
1: Comparator0 Rising-edge interrupt enabled.
Bit 4: CP0FIE: Comparator0 Falling-Edge Interrupt Enable.
0: Comparator0 Falling-edge interrupt disabled.
1: Comparator0 Falling-edge interrupt enabled.
Bits 3–2: UNUSED. Read = 00b, Write = don't care.
Bits 1–0: CP0MD1–CP0MD0: Comparator0 Mode Select
These bits select the response time for Comparator0.

Mode	CP0MD1	CP0MD0	CP0 Response Time (TYP)
0	0	0	100 ns
1	0	1	175 ns
2	1	0	320 ns
3	1	1	1050 ns

SFR Definition 8.4. CPT1CN: Comparator1 Control

SFR Page: all pages
SFR Address: 0x9A

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP1	CP1HYP0	CP1HYN1	CP1HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CP1EN: Comparator1 Enable Bit.
0: Comparator1 Disabled.
1: Comparator1 Enabled.

Bit 6: CP1OUT: Comparator1 Output State Flag.
0: Voltage on CP1+ < CP1–.
1: Voltage on CP1+ > CP1–.

Bit 5: CP1RIF: Comparator1 Rising-Edge Flag. Must be cleared by software.
0: No Comparator1 Rising Edge has occurred since this flag was last cleared.
1: Comparator1 Rising Edge has occurred.

Bit 4: CP1FIF: Comparator1 Falling-Edge Flag. Must be cleared by software.
0: No Comparator1 Falling-Edge has occurred since this flag was last cleared.
1: Comparator1 Falling-Edge has occurred.

Bits 3–2: CP1HYP1–0: Comparator1 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 20 mV.

Bits 1–0: CP1HYN1–0: Comparator1 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 20 mV.

SFR Definition 8.5. CPT1MX: Comparator1 MUX Selection

SFR Page: all pages
SFR Address: 0x9E

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	CMX1N1	CMX1N0	–	–	CMX1P1	CMX1P0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 11b, Write = don't care.
Bits 5–4: CMX1N1–CMX1N0: Comparator1 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator1 negative input.

CMX1N1	CMX1N0	C8051F360/3	C8051F361/2/4/5/6/7/8/9
		Negative Input	Negative Input
0	0	P2.1	P1.3
0	1	P2.6	P1.7
1	0	P3.4	P2.3
1	1	P4.0	P2.7

Bits 3–2: UNUSED. Read = 11b, Write = don't care.
Bits 1–0: CMX1P1–CMX1P0: Comparator1 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator1 positive input.

CMX1P1	CMX1P0	C8051F360/3	C8051F361/2/4/5/6/7/8/9
		Positive Input	Positive Input
0	0	P2.0	P1.2
0	1	P2.5	P1.6
1	0	P3.3	P2.2
1	1	P3.7	P2.6

SFR Definition 8.6. CPT1MD: Comparator1 Mode Selection

SFR Page: all pages
SFR Address: 0x9C

R	R	R/W	R/W	R	R	R/W	R/W	Reset Value
–	–	CP1RIE	CP1FIE	–	–	CP1MD1	CP1MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 00b, Write = don't care.
Bit 5: CP1RIE: Comparator1 Rising-Edge Interrupt Enable.
0: Comparator1 Rising-edge interrupt disabled.
1: Comparator1 Rising-edge interrupt enabled.
Bit 4: CP1FIE: Comparator1 Falling-Edge Interrupt Enable.
0: Comparator1 Falling-edge interrupt disabled.
1: Comparator1 Falling-edge interrupt enabled.
Bits 3–2: UNUSED. Read = 00b, Write = don't care.
Bits 1–0: CP1MD1–CP1MD0: Comparator1 Mode Select
These bits select the response time for Comparator1.

Mode	CP1MD1	CP1MD0	CP1 Response Time (TYP)
0	0	0	100 ns
1	0	1	175 ns
2	1	0	320 ns
3	1	1	1050 ns

Table 8.1. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +85 °C unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm* = 1.5 V	CPx+ – CPx– = 100 mV	—	100	—	ns
	CPx+ – CPx– = –100 mV	—	250	—	ns
Response Time: Mode 1, Vcm* = 1.5 V	CPx+ – CPx– = 100 mV	—	175	—	ns
	CPx+ – CPx– = –100 mV	—	500	—	ns
Response Time: Mode 2, Vcm* = 1.5 V	CPx+ – CPx– = 100 mV	—	320	—	ns
	CPx+ – CPx– = –100 mV	—	1100	—	ns
Response Time: Mode 3, Vcm* = 1.5 V	CPx+ – CPx– = 100 mV	—	1050	—	ns
	CPx+ – CPx– = –100 mV	—	5200	—	ns
Common-Mode Rejection Ratio		—	1.26	5	mV/V
Positive Hysteresis 1	CPxHYP1–0 = 00	—	0	1	mV
Positive Hysteresis 2	CPxHYP1–0 = 01	1	5	10	mV
Positive Hysteresis 3	CPxHYP1–0 = 10	6	10	20	mV
Positive Hysteresis 4	CPxHYP1–0 = 11	12	20	30	mV
Negative Hysteresis 1	CPxHYN1–0 = 00	—	0	1	mV
Negative Hysteresis 2	CPxHYN1–0 = 01	1	5	10	mV
Negative Hysteresis 3	CPxHYN1–0 = 10	6	10	20	mV
Negative Hysteresis 4	CPxHYN1–0 = 11	12	20	30	mV
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{DD} + 0.25	V
Input Capacitance		—	4	—	pF
Input Bias Current		—	0.001	—	nA
Input Offset Voltage		–5	—	+5	mV
Power Supply					
Power Supply Rejection		—	0.3	—	mV/V
Power-up Time		—	10	—	µs
Supply Current at DC	Mode 0	—	11.4	20	µA
	Mode 1	—	4.6	10	µA
	Mode 2	—	1.9	5	µA
	Mode 3	—	0.4	2.5	µA

***Note:** Vcm is the common-mode voltage on CPx+ and CPx–.

9. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. Included are five 16-bit counter/timers (see description in Section 21), one full-duplex UART (see description in Section 19), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (see Section 9.4.6), and up to four byte-wide and one 7-bit-wide I/O Ports (see description in Section 17). The CIP-51 also includes on-chip debug hardware (see description in Section 24), and interfaces directly with the MCU's analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 9.1 for a block diagram).

- Fully Compatible with MCS-51 Instruction Set
- 100 or 50 MIPS Peak Using the On-Chip PLL
- 256 Bytes of Internal RAM
- 8/4 Byte-Wide I/O Ports
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic

The CIP-51 includes the following features:

9.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's system clock running at 100 MHz, it has a peak throughput of 100 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/3	3	3/4	4	4/5	5	8
Number of Instructions	26	50	5	14	7	3	1	2	1

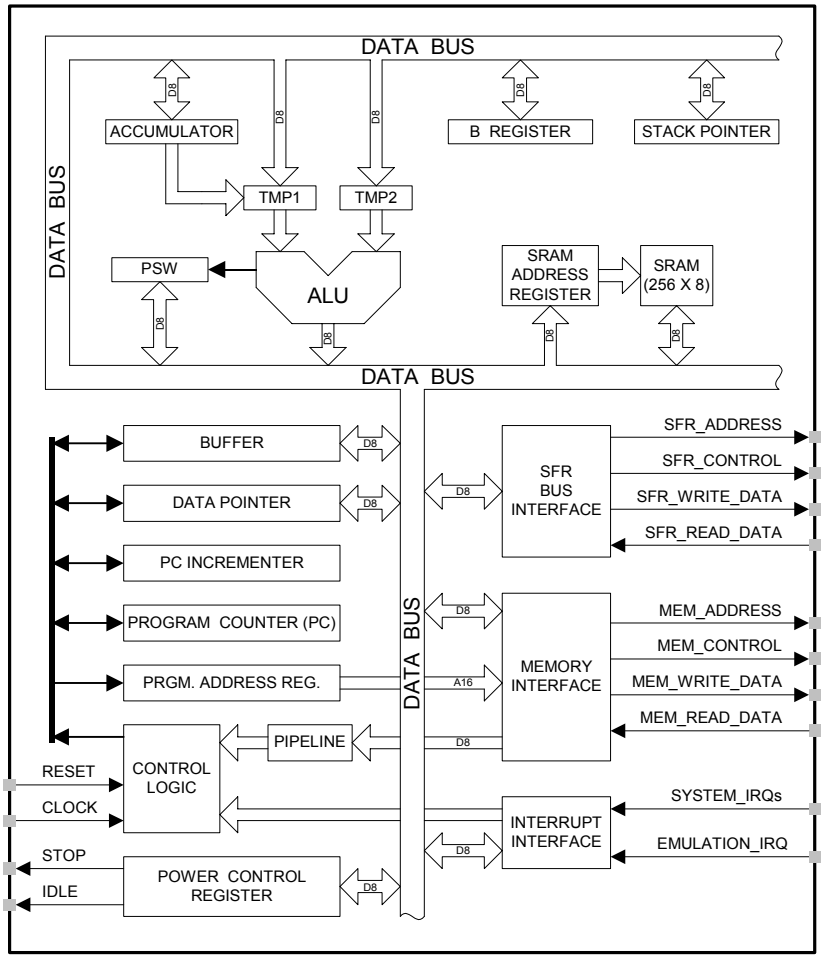


Figure 9.1. CIP-51 Block Diagram

9.2. Programming and Debugging Support

A C2-based serial interface is provided for in-system programming of the Flash program memory and communication with on-chip debug support logic. The re-programmable Flash can also be read and changed by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints and watch points, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debug is completely non-intrusive and non-invasive, requiring no RAM, Stack, timers, or other on-chip resources.

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, macro assembler, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via its C2 interface to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

9.3. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set; standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

9.3.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 9.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

9.3.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip XRAM, and accessing on-chip program Flash memory. The Flash access feature provides a mechanism for user software to update program code and use the program memory space for non-volatile data storage (see Section “13. Flash Memory” on page 135). The External Memory Interface provides a fast access to off-chip XRAM (or memory-mapped peripherals) via the MOVX instruction. Refer to Section “15. External Data Memory Interface and On-Chip XRAM” on page 153 for details.

Table 9.1. 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

Table 9.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
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
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3

Table 9.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3*
JNC rel	Jump if Carry is not set	2	2/3*
JB bit, rel	Jump if direct bit is set	3	3/4*
JNB bit, rel	Jump if direct bit is not set	3	3/4*
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4*
Program Branching			
ACALL addr11	Absolute subroutine call	2	3*
LCALL addr16	Long subroutine call	3	4*
RET	Return from subroutine	1	5*
RETI	Return from interrupt	1	5*
AJMP addr11	Absolute jump	2	3*
LJMP addr16	Long jump	3	4*
SJMP rel	Short jump (relative address)	2	3*
JMP @A+DPTR	Jump indirect relative to DPTR	1	3*

Table 9.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
JZ rel	Jump if A equals zero	2	2/3*
JNZ rel	Jump if A does not equal zero	2	2/3*
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/4*
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4*
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4*
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5*
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3*
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4*
NOP	No operation	1	1
* Branch instructions will incur a cache-miss penalty if the branch target location is not already stored in the Branch Target Cache. See Section “14. Branch Target Cache” on page 145 for more details.			

Notes on Registers, Operands and Addressing Modes:

Rn - Register R0-R7 of the currently selected register bank.

@Ri - Data RAM location addressed indirectly through R0 or R1.

rel - 8-bit, signed (2s complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct - 8-bit internal data location’s address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).

#data - 8-bit constant

#data16 - 16-bit constant

bit - Direct-accessed bit in Data RAM or SFR

addr11 - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-byte page of program memory as the first byte of the following instruction.

addr16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 64K-byte program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

9.4. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. There are 256 bytes of internal data memory and 32k bytes (C8051F360/1/2/3/4/5/6/7) or 16k bytes (C8051F368/9) of internal program memory address space implemented within the CIP-51. The CIP-51 memory organization is shown in Figure 9.2.

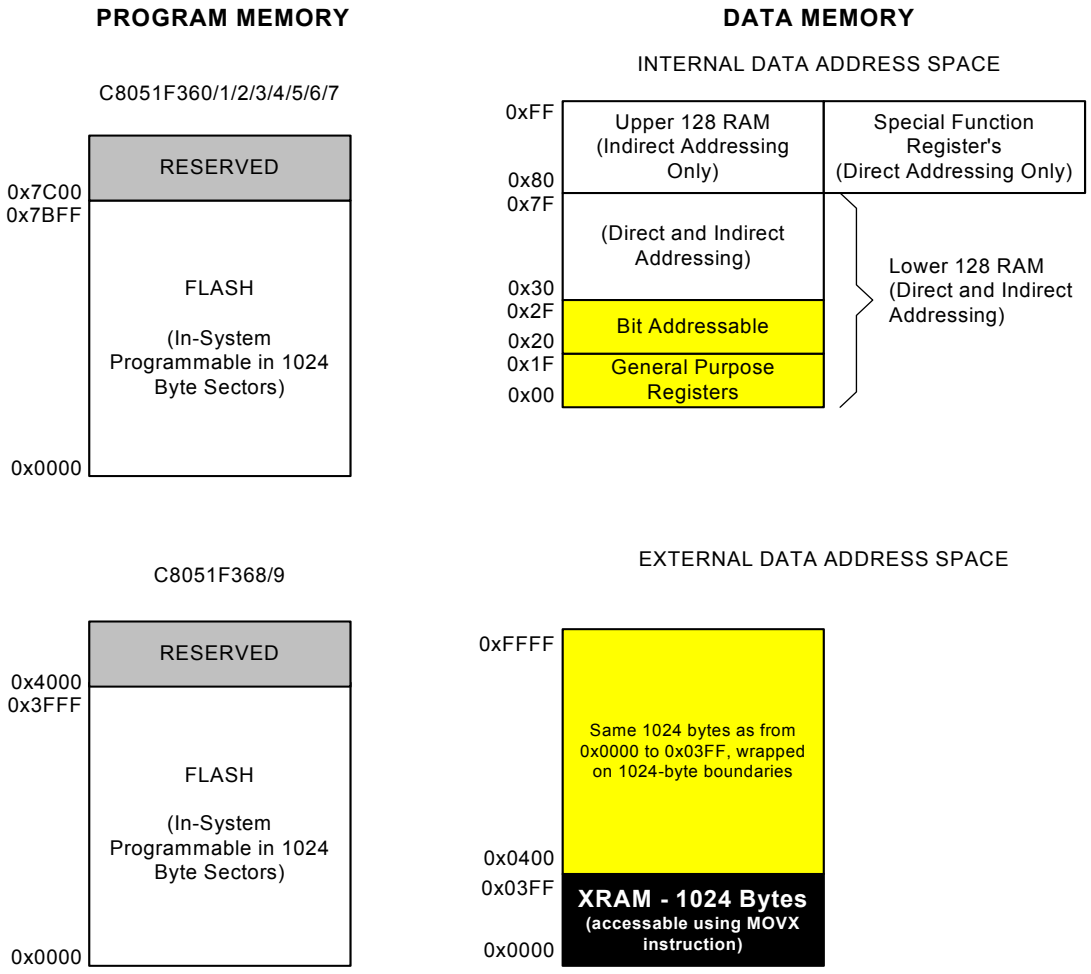


Figure 9.2. Memory Map

9.4.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F360/1/2/3/4/5/6/7 implement 32 kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x7BFF. Addresses above 0x7BFF are reserved on the 32 kB devices. The C8051F368/9 implement 16 kB of Flash from addresses 0x0000 to 0x3FFF.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section “13. Flash Memory” on page 135 for further details.

9.4.2. Data Memory

The CIP-51 implements 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFR's. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 9.2 illustrates the data memory organization of the CIP-51.

9.4.3. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 9.8). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

9.4.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination). The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte.

For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

9.4.5. Stack

A programmer's stack can be located anywhere in the 256 byte data memory. The stack area is designated using the Stack Pointer (SP, address 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07; therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

The MCUs also have built-in hardware for a stack record which is accessed by the debug logic. The stack record is a 32-bit shift register, where each PUSH or increment SP pushes one record bit onto the register,

and each CALL pushes two record bits onto the register. (A POP or decrement SP pops one record bit, and a RET pops two record bits, also.) The stack record circuitry can also detect an overflow or underflow on the 32-bit shift register, and can notify the debug software even with the MCU running at speed.

9.4.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFR's). The SFR's provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFR's found in a typical 8051 implementation as well as implementing additional SFR's used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 9.2 lists the SFR's implemented in the CIP-51 System Controller.

The SFR registers are accessed whenever the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFR's with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, P1, SCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFR's are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 9.3, for a detailed description of each register.

9.4.6.1. SFR Paging

The CIP-51 features *SFR paging,* allowing the device to map many SFR's into the 0x80 to 0xFF memory address space. The SFR memory space has 256 *pages*. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFR's. The C8051F36x family of devices utilizes two SFR pages: 0 and F. SFR pages are selected using the Special Function Register Page Selection register, SFRPAGE (see SFR Definition 9.2). The procedure for reading and writing an SFR is as follows:

1. Select the appropriate SFR page number using the SFRPAGE register.
2. Use direct accessing mode to read or write the special function register (MOV instruction).

9.4.6.2. Interrupts and SFR Paging

When an interrupt occurs, the SFR Page Register will automatically switch to SFR page 0, where all registers containing the interrupt flag bits are accessible. The automatic SFR Page switch function conveniently removes the burden of switching SFR pages from the interrupt service routine. Upon execution of the RETI instruction, the SFR page is automatically restored to the SFR Page in use prior to the interrupt. This is accomplished via a three-byte *SFR Page Stack*. The top byte of the stack is SFRPAGE, the current SFR Page. The second byte of the SFR Page Stack is SFRNEXT. The third, or bottom byte of the SFR Page Stack is SFRLAST. On interrupt, the current SFRPAGE value is pushed to the SFRNEXT byte, and the value of SFRNEXT is pushed to SFRLAST. Hardware then loads SFRPAGE with the SFR Page containing the flag bit associated with the interrupt. On a return from interrupt, the SFR Page Stack is popped resulting in the value of SFRNEXT returning to the SFRPAGE register, thereby restoring the SFR page context without software intervention. The value in SFRLAST (0x00 if there is no SFR Page value in the bottom of the stack) of the stack is placed in SFRNEXT register. If desired, the values stored in SFRNEXT and SFRLAST may be modified during an interrupt, enabling the CPU to return to a different SFR Page upon execution of the RETI instruction (on interrupt exit). Modifying registers in the SFR Page Stack does not cause a push or pop of the stack. Only interrupt calls and returns will cause push/pop operations on the SFR Page Stack.

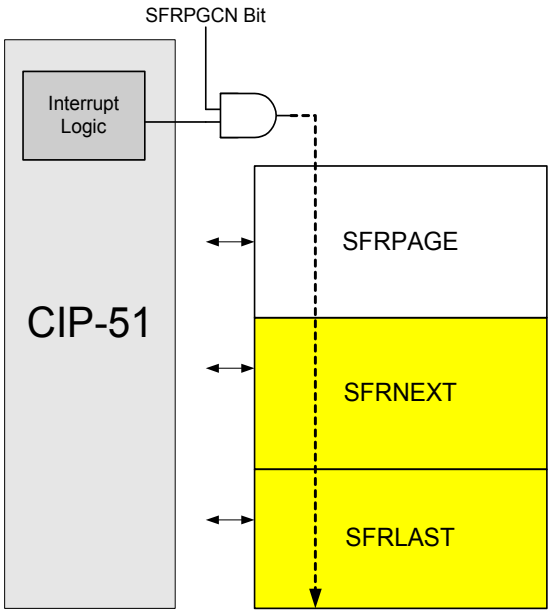


Figure 9.3. SFR Page Stack

Automatic hardware switching of the SFR Page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR Page Control Register (SFR0CN). This function defaults to ‘enabled’ upon reset. In this way, the autoswitching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) is provided in Table 9.2. in the form of an SFR memory map. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Note that certain SFR’s are accessible from ALL SFR pages, and are denoted by the background shading in the table. For example, the Port I/O registers P0, P1, P2, and P3 all have a shaded background, indicating these SFR’s are accessible from all SFR pages regardless of the SFRPAGE register value.

9.4.6.3. SFR Page Stack Example

The following is an example that shows the operation of the SFR Page Stack during interrupts.

In this example, the SFR Page Control is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to OSCICN (SFR “OSCICN”, located at address 0xB6 on SFR Page 0x0F). The device is also using the Programmable Counter Array (PCA) and the 10-bit ADC (ADC0) window comparator to monitor a voltage. The PCA is timing a critical control function in its interrupt service routine (ISR), so its interrupt is enabled and is set to *high* priority. The ADC0 is monitoring a voltage that is less important, but to minimize the software overhead its window comparator is being used with an associated ISR that is set to *low* priority. At this point, the SFR page is set to access the OSCICN SFR (SFRPAGE = 0x0F). See Figure 9.4 below.

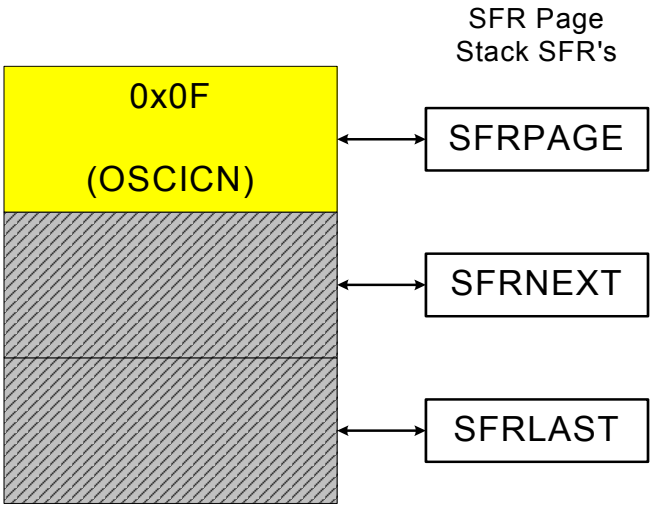


Figure 9.4. SFR Page Stack While Using SFR Page 0x0F To Access OSCICN

While CIP-51 executes in-line code (writing values to OSCICN in this example), ADC0 Window Comparator Interrupt occurs. The CIP-51 vectors to the ADC0 Window Comparator ISR and pushes the current SFR Page value (SFR Page 0x0F) into SFRNEXT in the SFR Page Stack. SFR page 0x00 is then automatically placed in the SFRPAGE register. SFRPAGE is considered the “top” of the SFR Page Stack. Software can now access the ADC0 SFR’s. Software may switch to any SFR Page by writing a new value to the SFRPAGE register at any time during the ADC0 ISR to access SFR’s that are not on SFR Page 0x00. See Figure 9.5 below.

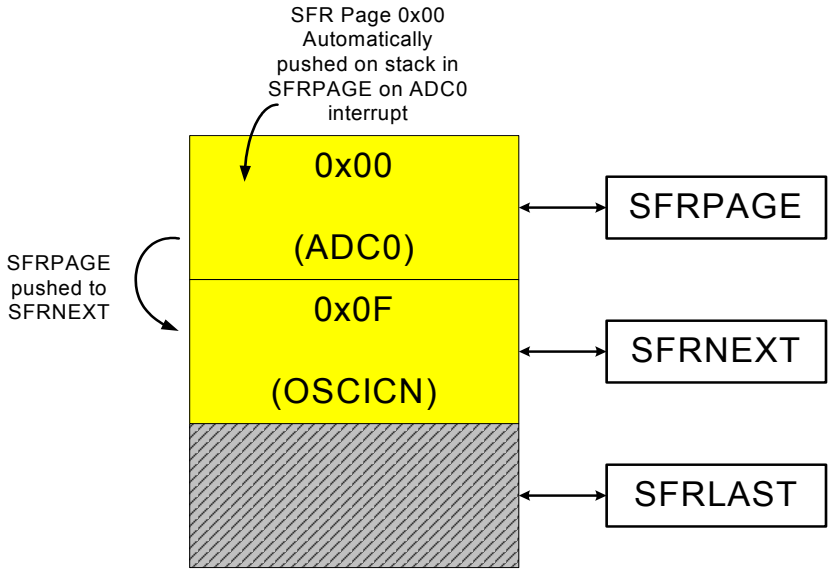


Figure 9.5. SFR Page Stack After ADC0 Window Comparator Interrupt Occurs

While in the ADC0 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a *high* priority interrupt, while the ADC0 interrupt is configured as a *low* priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place SFR page 0x00 into the SFRPAGE register. The value that was in the SFRPAGE register before the PCA interrupt (SFR Page 0x00 for ADC0) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR Page 0x0F for OSCICN) is pushed down to the SFRLAST register, the “bottom” of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 9.6 below.

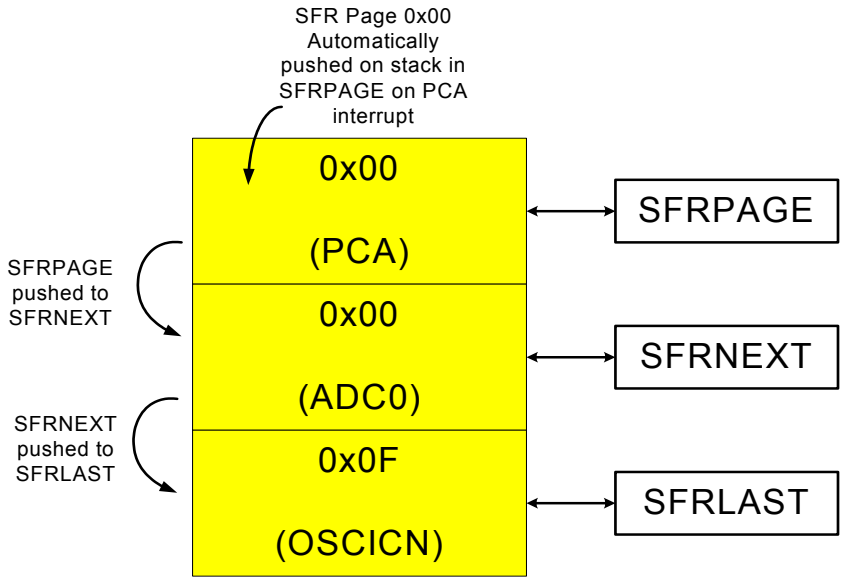


Figure 9.6. SFR Page Stack Upon PCA Interrupt Occurring During an ADC0 ISR

On exit from the PCA interrupt service routine, the CIP-51 will return to the ADC0 Window Comparator ISR. On execution of the RETI instruction, SFR Page 0x00 used to access the PCA registers will be automatically popped off of the SFR Page Stack, and the contents of the SFRNEXT register will be moved to the SFRPAGE register. Software in the ADC0 ISR can continue to access SFR's as it did prior to the PCA interrupt. Likewise, the contents of SFRLAST are moved to the SFRNEXT register. Recall this was the SFR Page value 0x0F being used to access OSCICN before the ADC0 interrupt occurred. See Figure 9.7 below.

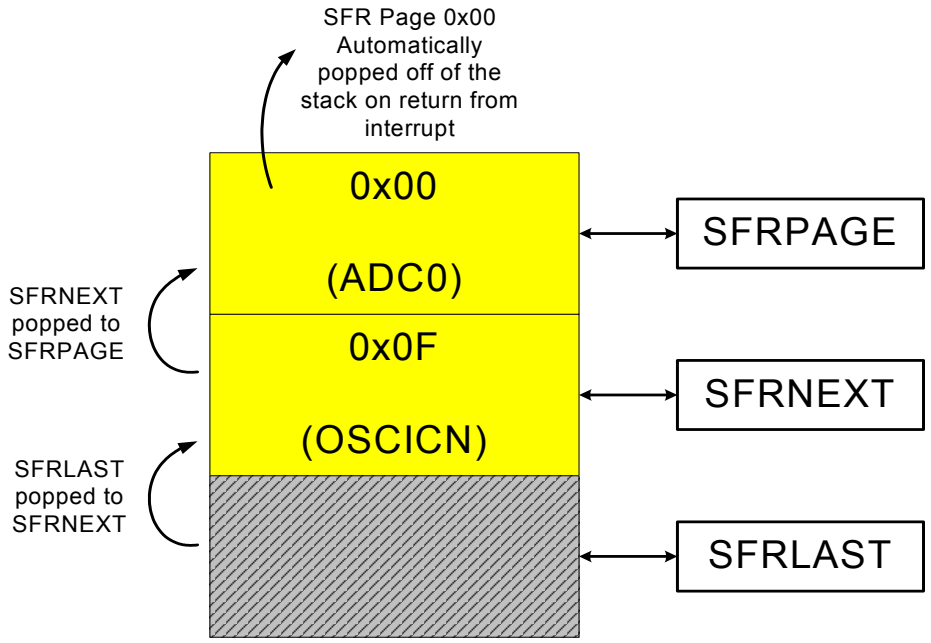


Figure 9.7. SFR Page Stack Upon Return From PCA Interrupt

On the execution of the RETI instruction in the ADC0 Window Comparator ISR, the value in SFRPAGE register is overwritten with the contents of SFRNEXT. The CIP-51 may now access the OSCICN SFR bits as it did prior to the interrupts occurring. See Figure 9.8 below.

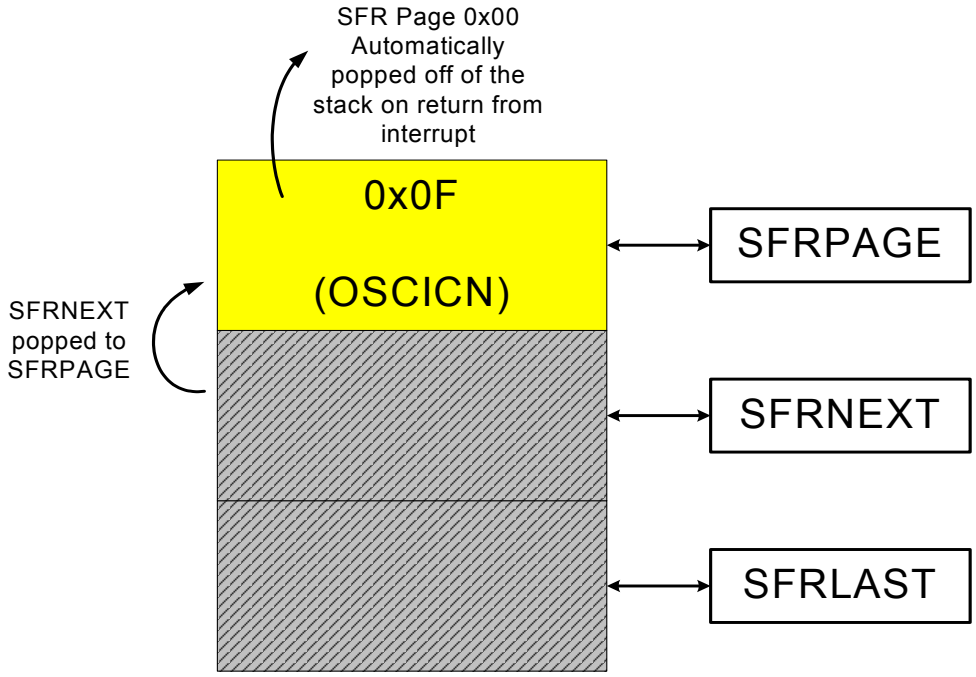


Figure 9.8. SFR Page Stack Upon Return From ADC2 Window Interrupt

Note that in the above example, all three bytes in the SFR Page Stack are accessible via the SFRPAGE, SFRNEXT, and SFRLAST special function registers. If the stack is altered while servicing an interrupt, it is possible to return to a different SFR Page upon interrupt exit than selected prior to the interrupt call. Direct access to the SFR Page stack can be useful to enable real-time operating systems to control and manage context switching between multiple tasks.

Push operations on the SFR Page Stack only occur on interrupt service, and pop operations only occur on interrupt exit (execution on the RETI instruction). The automatic switching of the SFRPAGE and operation of the SFR Page Stack as described above can be disabled in software by clearing the SFR Automatic Page Enable Bit (SFRPGEN) in the SFR Page Control Register (SFR0CN). See SFR Definition 9.1.

SFR Definition 9.1. SFR0CN: SFR Page Control

SFR Page: F
SFR Address: 0xE5

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SFRPGEN	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–1: RESERVED. Read = 0000000b. Must Write 0000000b.
Bit 0: SFRPGEN: SFR Automatic Page Control Enable.
Upon interrupt, the C8051 Core will vector to the specified interrupt service routine and automatically switch to SFR page 0. This bit is used to control this autopaging function.
0: SFR Automatic Paging disabled. C8051 core will not automatically change to SFR page 0.
1: SFR Automatic Paging enabled. Upon interrupt, the C8051 will automatically switch to SFR page 0.

SFR Definition 9.2. SFRPAGE: SFR Page

SFR Page: all pages
SFR Address: 0xA7

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	

Bits 7–0: SFR Page Bits: Byte Represents the SFR Page the C8051 MCU uses when reading or modifying SFR's.
Write: Sets the SFR Page.
Read: Byte is the SFR page the C8051 MCU is using.

When enabled in the SFR Page Control Register (SFR0CN), the C8051 will automatically switch to SFR Page 0x00 and return to the previous SFR page upon return from interrupt (unless SFR Stack was altered before a returning from the interrupt).
SFRPAGE is the top byte of the SFR Page Stack, and push/pop events of this stack are caused by interrupts (and **not** by reading/writing to the SFRPAGE register)

SFR Definition 9.3. SFRNEXT: SFR Next Register

SFR Page: all pages
SFR Address: 0x85

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	

Bits 7–0: SFR Page Stack Bits: SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to 'push' or 'pop'. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt.

Read: Returns the value of the SFR page contained in the second byte of the SFR stack. This is the value that will go to the SFR Page register upon a return from interrupt.

SFR Definition 9.4. SFRLAST: SFR Last Register

SFR Page: all pages
SFR Address: 0x86

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	

Bits 7–0: SFR Page Stack Bits: SFR page context is retained upon interrupts/return from interrupts in a 3 byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to 'push' or 'pop'. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Write: Sets the SFR Page in the last entry of the SFR Stack. This will cause the SFRNEXT SFR to have this SFR page value upon a return from interrupt.

Read: Returns the value of the SFR page contained in the last entry of the SFR stack.

Table 9.2. Special Function Register (SFR) Memory Map

ADDRESS	SFR Page	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8	0 F	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0	0 F	B	MAC0BL P0MDIN	MAC0BH P1MDIN	P0MAT P2MDIN	P0MASK P3MDIN	PCA0CPL5	PCA0CPH5	- EMI0TC
E8	0 F	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	0 F	ACC	P1MAT XBR0	P1MASK XBR1	- -	IT01CF	- SFR0CN	EIE1	EIE2
D8	0 F	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	PCA0CPM5
D0	0 F	PSW	REF0CN	MAC0ACC0 CCH0LC	MAC0ACC1 CCH0MA	MAC0ACC2 P0SKIP	MAC0ACC3 P1SKIP	MAC0OVR P2SKIP	MAC0CF P3SKIP
C8	0 F	TMR2CN	- CCH0TN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	- EIP1	MAC0STA EIP2
C0	0 F	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	- EMI0CF
B8	0 F	IP	IDA0CN	AMX0N	AMX0P	ADC0CF	ADC0L	ADC0H	- OSCICL
B0	0 F	P3	P2MAT PLL0MUL	P2MASK PLL0FLT	- PLL0CN	- -	P4	FLSCL OSCXCN	FLKEY OSCICN
A8	0 F	IE	- PLL0DIV	EMI0CN	- -	- FLSTAT	- OSCLCN	MAC0RNDL P4MDOUT	MAC0RNDH P3MDOUT
A0	0 F	P2	SPI0CFG	SPI0CKR	SPI0DAT	MAC0AL P0MDOUT	MAC0AH P1MDOUT	- P2MDOUT	SFRPAGE
98	0 F	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
90	0 F	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	IDA0L	IDA0H
88	0 F	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL CLKSEL
80	0 F	P0	SP	DPL	DPH	- CCH0CN	SFRNEXT	SFRLAST	PCON
		0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

bit-addressable shaded SFRs are accessible on all SFR Pages regardless of the contents of SFRPAGE

Table 9.3. Special Function Registers

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

Register	Address	SFR Page	Description	Page No.
ACC	0xE0	All Pages	Accumulator	page 104
ADC0CF	0xBC	All Pages	ADC0 Configuration	page 56[1]
ADC0CN	0xE8	All Pages	ADC0 Control	page 57[1]
ADC0GTH	0xC4	All Pages	ADC0 Greater-Than High Byte	page 58[1]
ADC0GTL	0xC3	All Pages	ADC0 Greater-Than Low Byte	page 58[1]
ADC0H	0xBE	All Pages	ADC0 Data Word High Byte	page 56[1]
ADC0L	0xBD	All Pages	ADC0 Data Word Low Byte	page 56[1]
ADC0LTH	0xC6	All Pages	ADC0 Less-Than High Byte	page 59[1]
ADC0LTL	0xC5	All Pages	ADC0 Less-Than Low Byte	page 59[1]
AMX0N	0xBA	All Pages	AMUX0 Negative Channel Select	page 55[1]
AMX0P	0xBB	All Pages	AMUX0 Positive Channel Select	page 54[1]
B	0xF0	All Pages	B Register	page 104
CCH0CN	0x84	F	Cache Control	page 149
CCH0LC	0xD2	F	Cache Lock	page 151
CCH0MA	0xD3	F	Cache Miss Accumulator	page 152
CCH0TN	0xC9	F	Cache Tuning	page 150
CKCON	0x8E	All Pages	Clock Control	page 254
CLKSEL	0x8F	F	System Clock Select	page 174
CPT0CN	0x9B	All Pages	Comparator0 Control	page 73
CPT0MD	0x9D	All Pages	Comparator0 Configuration	page 75
CPT0MX	0x9F	All Pages	Comparator0 MUX Selection	page 74
CPT1CN	0x9A	All Pages	Comparator1 Control	page 76
CPT1MD	0x9C	All Pages	Comparator1 Configuration	page 78
CPT1MX	0x9E	All Pages	Comparator1 MUX Selection	page 77
DPH	0x83	All Pages	Data Pointer High Byte	page 102
DPL	0x82	All Pages	Data Pointer Low Byte	page 102
EIE1	0xE6	All Pages	Extended Interrupt Enable 1	page 112
EIE2	0xE7	All Pages	Extended Interrupt Enable 2	page 114
EIP1	0xCE	F	Extended Interrupt Priority 1	page 113
EIP2	0xCF	F	Extended Interrupt Priority 2	page 114

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

Table 9.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
EMI0CF	0xC7	F	EMIF Configuration	page 156[2]
EMI0CN	0xAA	All Pages	EMIF Control	page 155[2]
EMI0TC	0xF7	F	EMIF Timing Control	page 161[2]
FLKEY	0xB7	0	Flash Lock and Key	page 142
FLSCL	0xB6	0	Flash Scale	page 143
FLSTAT	0xAC	F	Flash Status	page 152
IDA0CN	0xB9	All Pages	IDAC0 Control	page 65[1]
IDA0H	0x97	All Pages	IDAC0 High Byte	page 65[1]
IDA0L	0x96	All Pages	IDAC0 Low Byte	page 66[1]
IE	0xA8	All Pages	Interrupt Enable	page 110
IP	0xB8	All Pages	Interrupt Priority	page 111
IT01CF	0xE4	All Pages	INT0/INT1 Configuration	page 116
MAC0ACC0	0xD2	0	MAC0 Accumulator Byte 0 (LSB)	page 126
MAC0ACC1	0xD3	0	MAC0 Accumulator Byte 1	page 125
MAC0ACC2	0xD4	0	MAC0 Accumulator Byte 2	page 125
MAC0ACC3	0xD5	0	MAC0 Accumulator Byte 3 (MSB)	page 125
MAC0AH	0xA5	0	MAC0 A Register High Byte	page 123
MAC0AL	0xA4	0	MAC0 A Register Low Byte	page 124
MAC0BH	0xF2	0	MAC0 B Register High Byte	page 124
MAC0BL	0xF1	0	MAC0 B Register Low Byte	page 124
MAC0CF	0xD7	0	MAC0 Configuration	page 122
MAC0OVR	0xD6	0	MAC0 Accumulator Overflow	page 126
MAC0RNDH	0xAF	0	MAC0 Rounding Register High Byte	page 126
MAC0RNDL	0xAE	0	MAC0 Rounding Register Low Byte	page 127
MAC0STA	0xCF	0	MAC0 Status Register	page 123
OSCICL	0xBF	F	Internal Oscillator Calibration	page 170
OSCICN	0xB7	F	Internal Oscillator Control	page 171
OSCLCN	0xAD	F	Internal L-F Oscillator Control	page 172
OSCXCN	0xB6	F	External Oscillator Control	page 175
P0	0x80	All Pages	Port 0 Latch	page 190
P0MASK	0xF4	0	Port 0 Mask	page 192

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

Table 9.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
P0MAT	0xF3	0	Port 0 Match	page 192
P0MDIN	0xF1	F	Port 0 Input Mode	page 191
P0MDOUT	0xA4	F	Port 0 Output Mode Configuration	page 191
P0SKIP	0xD4	F	Port 0 Skip	page 192
P1	0x90	All Pages	Port 1 Latch	page 193
P1MASK	0xE2	0	Port 1 Mask	page 195
P1MAT	0xE1	0	Port 1 Match	page 194
P1MDIN	0xF2	F	Port 1 Input Mode	page 193
P1MDOUT	0xA5	F	Port 1 Output Mode Configuration	page 194
P1SKIP	0xD5	F	Port 1 Skip	page 194
P2	0xA0	All Pages	Port 2 Latch	page 195
P2MASK	0xB2	0	Port 2 Mask	page 197
P2MAT	0xB1	0	Port 2 Match	page 197
P2MDIN	0xF3	F	Port 2 Input Mode	page 196
P2MDOUT	0xA6	F	Port 2 Output Mode Configuration	page 196
P2SKIP	0xD6	F	Port 2 Skip	page 197
P3	0xB0	All Pages	Port 3 Latch	page 198
P3MDIN	0xF4	F	Port 3 Input Mode	page 198
P3MDOUT	0xAF	F	Port 3 Output Mode Configuration	page 199
P3SKIP	0xD7	F	Port 3 Skip	page 199
P4	0xB5	All Pages	Port 4 Latch	page 200
P4MDOUT	0xAE	F	Port 4 Output Mode Configuration	page 200
PCA0CN	0xD8	All Pages	PCA Control	page 276
PCA0CPH0	0xFC	All Pages	PCA Module 0 Capture/Compare High Byte	page 280
PCA0CPH1	0xEA	All Pages	PCA Module 1 Capture/Compare High Byte	page 280
PCA0CPH2	0xEC	All Pages	PCA Module 2 Capture/Compare High Byte	page 280
PCA0CPH3	0xEE	All Pages	PCA Module 3 Capture/Compare High Byte	page 280
PCA0CPH4	0xFE	All Pages	PCA Module 4 Capture/Compare High Byte	page 280
PCA0CPH5	0xF6	All Pages	PCA Module 5 Capture/Compare High Byte	page 280
PCA0CPL0	0xFB	All Pages	PCA Module 0 Capture/Compare Low Byte	page 279
PCA0CPL1	0xE9	All Pages	PCA Module 1 Capture/Compare Low Byte	page 279

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

Table 9.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
PCA0CPL2	0xEB	All Pages	PCA Module 2 Capture/Compare Low Byte	page 279
PCA0CPL3	0xED	All Pages	PCA Module 3 Capture/Compare Low Byte	page 279
PCA0CPL4	0xFD	All Pages	PCA Module 4 Capture/Compare Low Byte	page 279
PCA0CPL5	0xF5	All Pages	PCA Module 5 Capture/Compare Low Byte	page 279
PCA0CPM0	0xDA	All Pages	PCA Module 0 Mode	page 278
PCA0CPM1	0xDB	All Pages	PCA Module 1 Mode	page 278
PCA0CPM2	0xDC	All Pages	PCA Module 2 Mode	page 278
PCA0CPM3	0xDD	All Pages	PCA Module 3 Mode	page 278
PCA0CPM4	0xDE	All Pages	PCA Module 4 Mode	page 278
PCA0CPM5	0xDF	All Pages	PCA Module 5 Mode	page 278
PCA0H	0xFA	All Pages	PCA Counter High Byte	page 279
PCA0L	0xF9	All Pages	PCA Counter Low Byte	page 279
PCA0MD	0xD9	All Pages	PCA Mode	page 277
PCON	0x87	All Pages	Power Control	page 106
PLL0CN	0xB3	F	PLL Control	page 180
PLL0DIV	0xA9	F	PLL Divider	page 180
PLL0FLT	0xB2	F	PLL Filter	page 181
PLL0MUL	0xB1	F	PLL Multiplier	page 181
PSCTL	0x8F	0	Flash Write/Erase Control	page 142
PSW	0xD0	All Pages	Program Status Word	page 103
REF0CN	0xD1	All Pages	Voltage Reference Control	page 68[1]
RSTSRC	0xEF	All Pages	Reset Source	page 133
SBUF0	0x99	All Pages	UART 0 Data Buffer	page 226
SCON0	0x98	All Pages	UART 0 Control	page 225
SFR0CN	0xE5	F	SFR Page Control	page 94
SFRLAST	0x86	All Pages	SFR Stack Last Page	page 95
SFRNEXT	0x85	All Pages	SFR Stack Next Page	page 95
SFRPAGE	0xA7	All Pages	SFR Page Select	page 94
SMB0CF	0xC1	All Pages	SMBus Configuration	page 208
SMB0CN	0xC0	All Pages	SMBus Control	page 210
SMB0DAT	0xC2	All Pages	SMBus Data	page 212

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

Table 9.3. Special Function Registers (Continued)

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

Register	Address	SFR Page	Description	Page No.
SP	0x81	All Pages	Stack Pointer	page 102
SPI0CFG	0xA1	All Pages	SPI Configuration	page 241
SPI0CKR	0xA2	All Pages	SPI Clock Rate Control	page 243
SPI0CN	0xF8	All Pages	SPI Control	page 242
SPI0DAT	0xA3	All Pages	SPI Data	page 243
TCON	0x88	All Pages	Timer/Counter Control	page 252
TH0	0x8C	All Pages	Timer/Counter 0 High Byte	page 255
TH1	0x8D	All Pages	Timer/Counter 1 High Byte	page 255
TL0	0x8A	All Pages	Timer/Counter 0 Low Byte	page 255
TL1	0x8B	All Pages	Timer/Counter 1 Low Byte	page 255
TMOD	0x89	All Pages	Timer/Counter Mode	page 253
TMR2CN	0xC8	All Pages	Timer/Counter 2 Control	page 258
TMR2H	0xCD	All Pages	Timer/Counter 2 High Byte	page 259
TMR2L	0xCC	All Pages	Timer/Counter 2 Low Byte	page 259
TMR2RLH	0xCB	All Pages	Timer 2 Reload Register High Byte	page 259
TMR2RLL	0xCA	All Pages	Timer 2 Reload Register Low Byte	page 259
TMR3CN	0x91	All Pages	Timer 3 Control	page 262
TMR3H	0x95	All Pages	Timer 3 High Byte	page 263
TMR3L	0x94	All Pages	Timer 3 Low Byte	page 263
TMR3RLH	0x93	All Pages	Timer 3 Reload Register High Byte	page 263
TMR3RLL	0x92	All Pages	Timer 3 Reload Register Low Byte	page 263
VDM0CN	0xFF	All Pages	V_{DD} Monitor Control	page 131
XBR0	0xE1	F	Port I/O Crossbar Control 0	page 188
XBR1	0xE2	F	Port I/O Crossbar Control 1	page 189

Notes:
1. Refers to a register in the C8051F360/1/2/6/7/8/9 only.
2. Refers to a register in the C8051F360/3 only.

9.4.7. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic ‘1’. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic ‘0’, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the data sheet associated with their corresponding system function.

SFR Definition 9.5. SP: Stack Pointer

SFR Page: all pages
SFR Address: 0x81

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SP: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 9.6. DPL: Data Pointer Low Byte

SFR Page all pages
SFR Address: 0x82

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	

Bits 7–0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

SFR Definition 9.7. DPH: Data Pointer High Byte

SFR Page: all pages
SFR Address: 0x83

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	

Bits 7–0: DPH: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

SFR Definition 9.8. PSW: Program Status Word

SFR Page: all pages
SFR Address: 0xD0
(bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CY: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to 0 by all other arithmetic operations.

Bit 6: AC: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to 0 by all other arithmetic operations.

Bit 5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits 4–3: RS1–RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00–0x07
0	1	1	0x08–0x0F
1	0	2	0x10–0x17
1	1	3	0x18–0x1F

Bit 2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
• An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
• A MUL instruction results in an overflow (result is greater than 255).
• A DIV instruction causes a divide-by-zero condition.
The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit 1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit 0: PARITY: Parity Flag.
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

SFR Definition 9.9. ACC: Accumulator

SFR Page: all pages
SFR Address: 0xE0 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: ACC: Accumulator.
This register is the accumulator for arithmetic operations.

SFR Definition 9.10. B: B Register

SFR Page: all pages
SFR Address: 0xF0 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: B: B Register.
This register serves as a second accumulator for certain arithmetic operations.

9.5. 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 external peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the system clock is stopped. 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. SFR Definition 9.11 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 put into low power mode. Digital peripherals, such as timers or serial buses, draw little power whenever they are not in use. Turning off the Flash memory saves power, similar to entering Idle mode. Turning off the oscillator saves even more power, but requires a reset to restart the MCU.

The C8051F36x devices feature an additional low-power SUSPEND mode, which stops the internal oscillator until an awakening event occurs. See Section “16.1.1. Internal Oscillator Suspend Mode” on page 170 for more information.

9.5.1. 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. 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 or $\overline{\text{RST}}$ is asserted. 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.

If enabled, the 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 22.3 for more information on the use and configuration of the WDT.

Note: Any instruction which sets the IDLE bit should be immediately followed by an instruction which has two 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 the instruction following the write to the IDLE bit is a single-byte instruction and an interrupt occurs during the execution of the instruction of the instruction which sets the IDLE bit, the CPU may not wake from IDLE mode when a future interrupt occurs.

9.5.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes. In Stop mode, the CPU and oscillators are stopped, effectively shutting down all digital peripherals. Each analog peripheral must be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to sleep for longer than the MCD timeout of 100 µs.

9.5.3. Suspend Mode

The C8051F36x devices feature a low-power SUSPEND mode, which stops the internal oscillator until an awakening event occurs. See Section “16.1.1. Internal Oscillator Suspend Mode” on page 170.

SFR Definition 9.11. PCON: Power Control

SFR Page: all pages
SFR Address: 0x87

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–3: RESERVED. Read = 000000b. Must Write 000000b.
Bit 1: STOP: STOP Mode Select.
Writing a '1' to this bit will place the CIP-51 into STOP mode. This bit will always read '0'.
1: CIP-51 forced into power-down mode. (Turns off oscillator).
Bit 0: IDLE: IDLE Mode Select.
Writing a '1' to this bit will place the CIP-51 into IDLE mode. This bit will always read '0'.
1: CIP-51 forced into IDLE mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, and all peripherals remain active.)

10. Interrupt Handler

The C8051F36x family includes an extended interrupt system supporting a total of 16 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic ‘1’.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic ‘1’ regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable and Extended Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic ‘1’ before the individual interrupt enables are recognized. Setting the EA bit to logic ‘0’ disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic ‘0’ will be held in a pending state, and will not be serviced until the EA bit is set back to logic ‘1’.

Note: Any instruction that clears a bit to disable an interrupt should be immediately followed by an instruction that has two or more opcode bytes. Using EA (global interrupt enable) as an example:

```
// in 'C':
EA = 0; // clear EA bit.
EA = 0; // this is a dummy instruction with two-byte opcode.
; in assembly:
CLR EA ; clear EA bit.
CLR EA ; this is a dummy instruction with two-byte opcode.
```

For example, if an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction which clears a bit to disable an interrupt source), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. However, a read of the enable bit will return a '0' inside the interrupt service routine. When the bit-clearing opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

10.1. MCU Interrupt Sources and Vectors

The C8051F36x MCUs support 16 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic ‘1’. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order, and control bits are summarized in Table 10.1 on page 108. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

10.2. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP, EIP1, or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 10.1.

10.3. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. Additional clock cycles will be required if a cache miss occurs (see Section 14 for more details). If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) is when the CPU is performing an RETI instruction followed by a DIV as the next instruction, and a cache miss event also occurs. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

Table 10.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
RESERVED	0x0043	8	N/A	N/A	N/A	N/A	N/A
ADC0 Window Comparator	0x004B	9	AD0WINT (ADC0CN.5)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 End of Conversion	0x0053	10	AD0INT (ADC0STA.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)

Table 10.1. Interrupt Summary (Continued)

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Programmable Counter Array	0x005B	11	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.4)	PPCA0 (EIP1.4)
Comparator0	0x0063	12	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECP0 (EIE1.5)	PCP0 (EIP1.5)
Comparator1	0x006B	13	CP1FIF (CPT1CN.4) CP1RIF (CPT1CN.5)	N	N	ECP1 (EIE1.6)	PCP1 (EIP1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
RESERVED	0x007B	15	N/A	N/A	N/A	N/A	N/A
Port Match	0x0083	16	N/A	N/A	N/A	EMAT (EIE2.1)	PMAT (EIP2.1)

10.4. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described below. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 10.1. IE: Interrupt Enable

SFR Page: all pages
SFR Address: 0xA8
(bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: EA: Global Interrupt Enable.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit 6: ESPI0: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of the SPI0 interrupts.
0: Disable all SPI0 interrupts.
1: Enable interrupt requests generated by SPI0.

Bit 5: ET2: Enable Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable interrupt requests generated by the TF2L or TF2H flags.

Bit 4: ES0: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit 3: ET1: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit 2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the /INT1 input.

Bit 1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit 0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the /INT0 input.

SFR Definition 10.2. IP: Interrupt Priority

SFR Page: **all pages**
SFR Address: **0xB8** (bit addressable)

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	PSPI0	PT2	PS0	PT1	PX1	PT0	PX0	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: UNUSED. Read = 1b, Write = don't care.
Bit 6: PSPI0: Serial Peripheral Interface (SPI0) Interrupt Priority Control.
This bit sets the priority of the SPI0 interrupt.
0: SPI0 interrupt set to low priority level.
1: SPI0 interrupt set to high priority level.
Bit 5: PT2: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt set to low priority level.
1: Timer 2 interrupt set to high priority level.
Bit 4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupt set to high priority level.
Bit 3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority level.
1: Timer 1 interrupt set to high priority level.
Bit 2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.
Bit 1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.
Bit 0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

SFR Definition 10.3. EIE1: Extended Interrupt Enable 1

SFR Page: all pages
SFR Address: 0xE6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	–	ESMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: ET3: Enable Timer 3 Interrupt.
This bit sets the masking of the Timer 3 interrupt.
0: Disable Timer 3 interrupts.
1: Enable interrupt requests generated by the TF3L or TF3H flags.

Bit 6: ECP1: Enable Comparator1 (CP1) Interrupt.
This bit sets the masking of the CP1 interrupt.
0: Disable CP1 interrupts.
1: Enable interrupt requests generated by the CP1RIF or CP1FIF flags.

Bit 5: ECP0: Enable Comparator0 (CP0) Interrupt.
This bit sets the masking of the CP0 interrupt.
0: Disable CP0 interrupts.
1: Enable interrupt requests generated by the CP0RIF or CP0FIF flags.

Bit 4: EPCA0: Enable Programmable Counter Array (PCA0) Interrupt.
This bit sets the masking of the PCA0 interrupts.
0: Disable all PCA0 interrupts.
1: Enable interrupt requests generated by PCA0.

Bit 3: EADC0: Enable ADC0 Conversion Complete Interrupt.
This bit sets the masking of the ADC0 Conversion Complete interrupt.
0: Disable ADC0 Conversion Complete interrupt.
1: Enable interrupt requests generated by the AD0INT flag.

Bit 2: EWADC0: Enable ADC0 Window Comparison Interrupt.
This bit sets the masking of the ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison interrupt.
1: Enable interrupt requests generated by the AD0WINT flag.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

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

SFR Definition 10.4. EIP1: Extended Interrupt Priority 1

SFR Page: F
SFR Address: 0xCE

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	–	PSMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: PT3: Timer 3 Interrupt Priority Control.
This bit sets the priority of the Timer 3 interrupt.
0: Timer 3 interrupts set to low priority level.
1: Timer 3 interrupts set to high priority level.

Bit 6: PCP1: Comparator1 (CP1) Interrupt Priority Control.
This bit sets the priority of the CP1 interrupt.
0: CP1 interrupt set to low priority level.
1: CP1 interrupt set to high priority level.

Bit 5: PCP0: Comparator0 (CP0) Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: CP0 interrupt set to low priority level.
1: CP0 interrupt set to high priority level.

Bit 4: PPCA0: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority level.
1: PCA0 interrupt set to high priority level.

Bit 3: PADC0: ADC0 Conversion Complete Interrupt Priority Control.
This bit sets the priority of the ADC0 Conversion Complete interrupt.
0: ADC0 Conversion Complete interrupt set to low priority level.
1: ADC0 Conversion Complete interrupt set to high priority level.

Bit 2: PWADC0: ADC0 Window Comparison Interrupt Priority Control.
This bit sets the priority of the ADC0 Window Comparison interrupt.
0: ADC0 Window Comparison interrupt set to low priority level.
1: ADC0 Window Comparison interrupt set to high priority level.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: PSMB0: SMBus (SMB0) Interrupt Priority Control.
This bit sets the priority of the SMB0 interrupt.
0: SMB0 interrupt set to low priority level.
1: SMB0 interrupt set to high priority level.

SFR Definition 10.5. EIE2: Extended Interrupt Enable 2

SFR Page: all pages
SFR Address: 0xE7

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	–	–	EMAT	–	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–2: UNUSED. Read = 000000b. Write = don't care.
Bit 1: EMAT: Enable Port Match Interrupt.
This bit sets the masking of the Port Match interrupt.
0: Disable the Port Match interrupt.
1: Enable the Port Match interrupt.
Bit 0: UNUSED. Read = 0b. Write = don't care.

SFR Definition 10.6. EIP2: Extended Interrupt Priority 2

SFR Page: F
SFR Address: 0xCF

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	–	–	PMAT	–	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–2: UNUSED. Read = 000000b. Write = don't care.
Bit 1: PMAT: Port Match Interrupt Priority Control.
This bit sets the priority of the Port Match interrupt.
0: Port Match interrupt set to low priority level.
1: Port Match interrupt set to high priority level.
Bit 0: UNUSED. Read = 0b. Write = don't care.

10.5. External Interrupts

The /INT0 and /INT1 external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL (/INT0 Polarity) and IN1PL (/INT1 Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (Section “21.1. Timer 0 and Timer 1” on page 248) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	/INT0 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	/INT1 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

/INT0 and /INT1 are assigned to Port pins as defined in the IT01CF register (see SFR Definition 10.7). Note that /INT0 and /INT0 Port pin assignments are independent of any Crossbar assignments. /INT0 and /INT1 will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to /INT0 and/or /INT1, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “17.1. Priority Crossbar Decoder” on page 185 for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic ‘1’ while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic ‘0’ while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

SFR Definition 10.7. IT01CF: INT0/INT1 Configuration

SFR Page: all pages
SFR Address: 0xE4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Note: Refer to SFR Definition 21.1. “TCON: Timer Control” on page 252 for INT0/1 edge- or level-sensitive interrupt selection.

Bit 7: IN1PL: /INT1 Polarity
0: /INT1 input is active low.
1: /INT1 input is active high.

Bits 6–4: IN1SL2-0: /INT1 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT1. Note that this pin assignment is independent of the Crossbar; /INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to ‘1’ the corresponding bit in register P0SKIP).

IN1SL2-0	/INT1 Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

Bit 3: IN0PL: /INT0 Polarity
0: /INT0 interrupt is active low.
1: /INT0 interrupt is active high.

Bits 2–0: INT0SL2-0: /INT0 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT0. Note that this pin assignment is independent of the Crossbar. /INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to ‘1’ the corresponding bit in register P0SKIP).

IN0SL2-0	/INT0 Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

11. Multiply And Accumulate (MAC0)

The C8051F36x devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. MAC0 contains a 16-by-16 bit multiplier and a 40-bit adder, which can perform integer or fractional multiply-accumulate and multiply operations on signed input values in two SYSCLK cycles. A rounding engine provides a rounded 16-bit fractional result after an additional (third) SYSCLK cycle. MAC0 also contains a 1-bit arithmetic shifter that will left or right-shift the contents of the 40-bit accumulator in a single SYSCLK cycle. Figure 11.1 shows a block diagram of the MAC0 unit and its associated Special Function Registers.

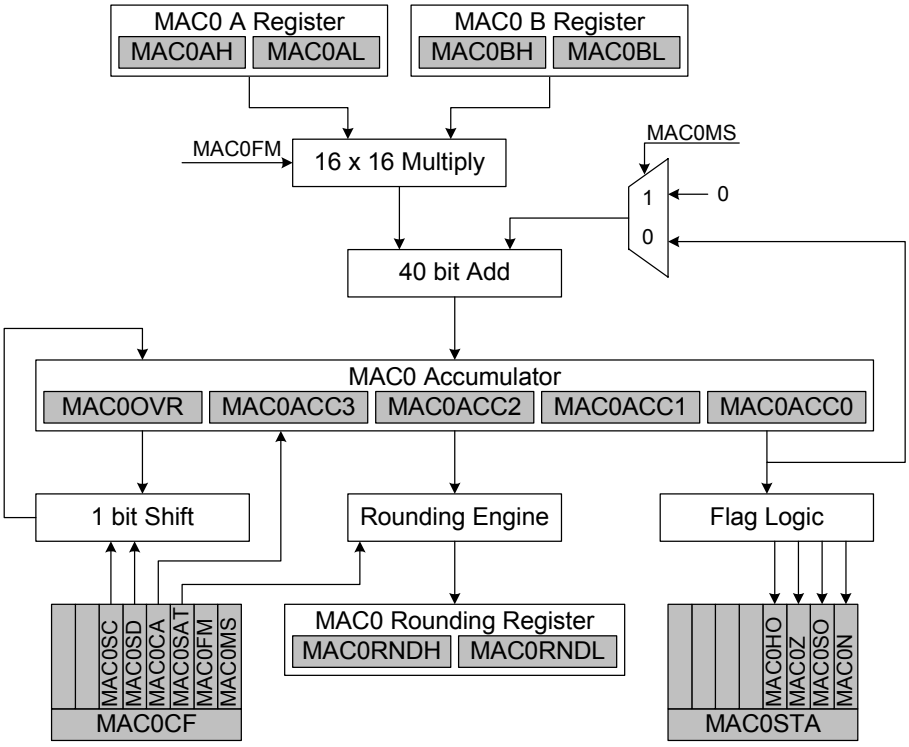


Figure 11.1. MAC0 Block Diagram

11.1. Special Function Registers

There are thirteen Special Function Register (SFR) locations associated with MAC0. Two of these registers are related to configuration and operation, while the other eleven are used to store multi-byte input and output data for MAC0. The Configuration register MAC0CF (SFR Definition 11.1) is used to configure and control MAC0. The Status register MAC0STA (SFR Definition 11.2) contains flags to indicate overflow conditions, as well as zero and negative results. The 16-bit MAC0A (MAC0AH:MAC0AL) and MAC0B (MAC0BH:MAC0BL) registers are used as inputs to the multiplier. The MAC0 Accumulator register is 40 bits long, and consists of five SFRs: MAC0OVR, MAC0ACC3, MAC0ACC2, MAC0ACC1, and MAC0ACC0. The primary results of a MAC0 operation are stored in the Accumulator registers. If they are needed, the rounded results are stored in the 16-bit Rounding Register MAC0RND (MAC0RNDH:MAC0RNDL).

11.2. Integer and Fractional Math

MAC0 is capable of interpreting the 16-bit inputs stored in MAC0A and MAC0B as signed integers or as signed fractional numbers. When the MAC0FM bit (MAC0CF.1) is cleared to '0', the inputs are treated as 16-bit, 2's complement, integer values. After the operation, the accumulator will contain a 40-bit, 2's complement, integer value. Figure 11.2 shows how integers are stored in the SFRs.

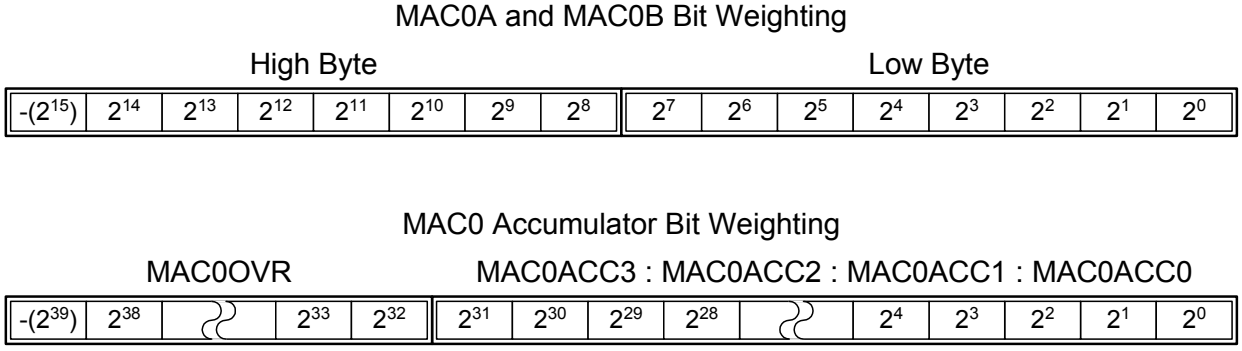
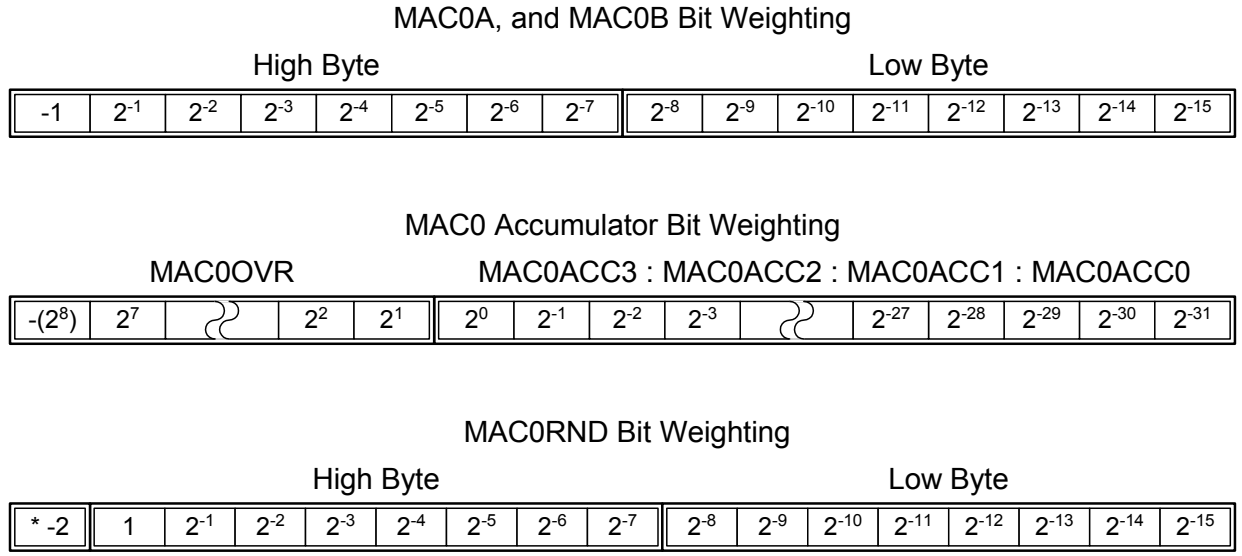


Figure 11.2. Integer Mode Data Representation

When the MAC0FM bit is set to '1', the inputs are treated at 16-bit, 2's complement, fractional values. The decimal point is located between bits 15 and 14 of the data word. After the operation, the accumulator will contain a 40-bit, 2's complement, fractional value, with the decimal point located between bits 31 and 30. Figure 11.3 shows how fractional numbers are stored in the SFRs.



* The MAC0RND register contains the 16 LSBs of a two's complement number. The MAC0N Flag can be used to determine the sign of the MAC0RND register.

Figure 11.3. Fractional Mode Data Representation

11.3. Operating in Multiply and Accumulate Mode

MAC0 operates in Multiply and Accumulate (MAC) mode when the MAC0MS bit (MAC0CF.0) is cleared to '0'. When operating in MAC mode, MAC0 performs a 16-by-16 bit multiply on the contents of the MAC0A and MAC0B registers, and adds the result to the contents of the 40-bit MAC0 accumulator. Figure 11.4 shows the MAC0 pipeline. There are three stages in the pipeline, each of which takes exactly one SYSCLK cycle to complete. The MAC operation is initiated with a write to the MAC0BL register. After the MAC0BL register is written, MAC0A and MAC0B are multiplied on the first SYSCLK cycle. During the second stage of the MAC0 pipeline, the results of the multiplication are added to the current accumulator contents, and the result of the addition is stored in the MAC0 accumulator. The status flags in the MAC0STA register are set after the end of the second pipeline stage. During the second stage of the pipeline, the next multiplication can be initiated by writing to the MAC0BL register, if it is desired. The rounded (and optionally, saturated) result is available in the MAC0RNDH and MAC0RNDL registers at the end of the third pipeline stage. If the MAC0CA bit (MAC0CF.3) is set to '1' when the MAC operation is initiated, the accumulator and all MAC0STA flags will be cleared during the next cycle of the controller's clock (SYSCLK). The MAC0CA bit will clear itself to '0' when the clear operation is complete.

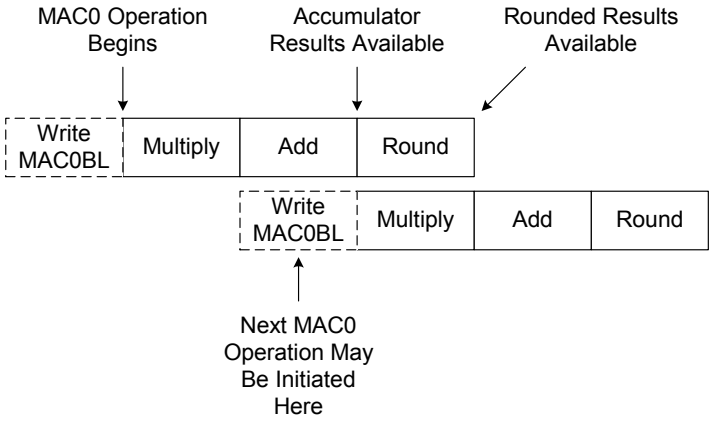


Figure 11.4. MAC0 Pipeline

11.4. Operating in Multiply Only Mode

MAC0 operates in Multiply Only mode when the MAC0MS bit (MAC0CF.0) is set to '1'. Multiply Only mode is identical to Multiply and Accumulate mode, except that the multiplication result is added with a value of zero before being stored in the MAC0 accumulator (i.e. it overwrites the current accumulator contents). The result of the multiplication is available in the MAC0 accumulator registers at the end of the second MAC0 pipeline stage (two SYSCLKs after writing to MAC0BL). As in MAC mode, the rounded result is available in the MAC0 Rounding Registers after the third pipeline stage. Note that in Multiply Only mode, the MAC0HO flag is not affected.

11.5. Accumulator Shift Operations

MAC0 contains a 1-bit arithmetic shift function which can be used to shift the contents of the 40-bit accumulator left or right by one bit. The accumulator shift is initiated by writing a '1' to the MAC0SC bit (MAC0CF.5), and takes one SYSCLK cycle (the rounded result is available in the MAC0 Rounding Registers after a second SYSCLK cycle, and MAC0SC is cleared to '0'). The direction of the arithmetic shift is controlled by the MAC0SD bit (MAC0CF.4). When this bit is cleared to '0', the MAC0 accumulator will shift left. When the MAC0SD bit is set to '1', the MAC0 accumulator will shift right. Right-shift operations are sign-extended with the current value of bit 39. Note that the status flags in the MAC0STA register are not affected by shift operations.

11.6. Rounding and Saturation

A Rounding Engine is included, which can be used to provide a rounded result when operating on fractional numbers. MAC0 uses an unbiased rounding algorithm to round the data stored in bits 31–16 of the accumulator, as shown in Table 11.1. Rounding occurs during the third stage of the MAC0 pipeline, after any shift operation, or on a write to the LSB of the accumulator. The rounded results are stored in the rounding registers: MAC0RNDH (SFR Definition 11.12) and MAC0RNDL (SFR Definition 11.13). The accumulator registers are not affected by the rounding engine. Although rounding is primarily used for fractional data, the data in the rounding registers is updated in the same way when operating in integer mode.

Table 11.1. MAC0 Rounding (MAC0SAT = 0)

Accumulator Bits 15–0 (MAC0ACC1:MAC0ACC0)	Accumulator Bits 31–16 (MAC0ACC3:MAC0ACC2)	Rounding Direction	Rounded Results (MAC0RNDH:MAC0RNDL)
Greater Than 0x8000	Anything	Up	(MAC0ACC3:MAC0ACC2) + 1
Less Than 0x8000	Anything	Down	(MAC0ACC3:MAC0ACC2)
Equal To 0x8000	Odd (LSB = 1)	Up	(MAC0ACC3:MAC0ACC2) + 1
Equal To 0x8000	Even (LSB = 0)	Down	(MAC0ACC3:MAC0ACC2)

The rounding engine can also be used to saturate the results stored in the rounding registers. If the MAC0SAT bit is set to '1' and the rounding register overflows, the rounding registers will saturate. When a positive overflow occurs, the rounding registers will show a value of 0x7FFF when saturated. For a negative overflow, the rounding registers will show a value of 0x8000 when saturated. If the MAC0SAT bit is cleared to '0', the rounding registers will not saturate.

11.7. Usage Examples

This section details some software examples for using MAC0. Section 11.7.1 shows a series of two MAC operations using fractional numbers. Section 11.7.2 shows a single operation in Multiply Only mode with integer numbers. The last example, shown in Section 11.7.3, demonstrates how the left-shift and right-shift operations can be used to modify the accumulator. All of the examples assume that all of the flags in the MAC0STA register are initially set to '0'.

11.7.1. Multiply and Accumulate Example

The example below implements the equation:

$$(0.5 \times 0.25) + (0.5 \times -0.25) = 0.125 - 0.125 = 0.0$$

```
MOV   MAC0CF, #0Ah    ; Set to Clear Accumulator, Use fractional numbers
MOV   MAC0AH, #40h    ; Load MAC0A register with 4000 hex = 0.5 decimal
MOV   MAC0AL, #00h
MOV   MAC0BH, #20h    ; Load MAC0B register with 2000 hex = 0.25 decimal
MOV   MAC0BL, #00h    ; This line initiates the first MAC operation
MOV   MAC0BH, #E0h    ; Load MAC0B register with E000 hex = -0.25 decimal
MOV   MAC0BL, #00h    ; This line initiates the second MAC operation
NOP
NOP                   ; After this instruction, the Accumulator should be equal to 0,
                      ; and the MAC0STA register should be 0x04, indicating a zero
NOP                   ; After this instruction, the Rounding register is updated
```

11.7.2. Multiply Only Example

The example below implements the equation:

$$4660 \times -292 = -1360720$$

```
MOV   MAC0CF, #01h    ; Use integer numbers, and multiply only mode (add to zero)
MOV   MAC0AH, #12h    ; Load MAC0A register with 1234 hex = 4660 decimal
MOV   MAC0AL, #34h
MOV   MAC0BH, #FEh    ; Load MAC0B register with FEDC hex = -292 decimal
MOV   MAC0BL, #DCh    ; This line initiates the Multiply operation
NOP
NOP                   ; After this instruction, the Accumulator should be equal to
                      ; FFFFEB3CB0 hex = -1360720 decimal. The MAC0STA register should
                      ; be 0x01, indicating a negative result.
NOP                   ; After this instruction, the Rounding register is updated
```

11.7.3. MAC0 Accumulator Shift Example

The example below shifts the MAC0 accumulator left one bit, and then right two bits:

```
MOV   MAC0OVR, #40h   ; The next few instructions load the accumulator with the value
MOV   MAC0ACC3, #88h  ; 4088442211 Hex.
MOV   MAC0ACC2, #44h
MOV   MAC0ACC1, #22h
MOV   MAC0ACC0, #11h
MOV   MAC0CF, #20h    ; Initiate a Left-shift
NOP                   ; After this instruction, the accumulator should be 0x8110884422
NOP                   ; The rounding register is updated after this instruction
MOV   MAC0CF, #30h    ; Initiate a Right-shift
MOV   MAC0CF, #30h    ; Initiate a second Right-shift
NOP                   ; After this instruction, the accumulator should be 0xE044221108
NOP                   ; The rounding register is updated after this instruction
```

SFR Definition 11.1. MAC0CF: MAC0 Configuration

SFR Page: 0
SFR Address: 0xD7

R	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	MAC0SC	MAC0SD	MAC0CA	MAC0SAT	MAC0FM	MAC0MS	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED: Read = 00b, Write = don't care.

Bit 5: MAC0SC: Accumulator Shift Control.
When set to 1, the 40-bit MAC0 Accumulator register will be shifted during the next SYSCLK cycle. The direction of the shift (left or right) is controlled by the MAC0SD bit.
This bit is cleared to '0' by hardware when the shift is complete.

Bit 4: MAC0SD: Accumulator Shift Direction.
This bit controls the direction of the accumulator shift activated by the MAC0SC bit.
0: MAC0 Accumulator will be shifted left.
1: MAC0 Accumulator will be shifted right.

Bit 3: MAC0CA: Clear Accumulator.
This bit is used to reset MAC0 before the next operation.
When set to '1', the MAC0 Accumulator will be cleared to zero and the MAC0 Status register will be reset during the next SYSCLK cycle.
This bit will be cleared to '0' by hardware when the reset is complete.

Bit 2: MAC0SAT: Saturate Rounding Register.
This bit controls whether the Rounding Register will saturate. If this bit is set and a Soft Overflow occurs, the Rounding Register will saturate. This bit does not affect the operation of the MAC0 Accumulator. See Section 11.6 for more details about rounding and saturation.
0: Rounding Register will not saturate.
1: Rounding Register will saturate.

Bit 1: MAC0FM: Fractional Mode.
This bit selects between Integer Mode and Fractional Mode for MAC0 operations.
0: MAC0 operates in Integer Mode.
1: MAC0 operates in Fractional Mode.

Bit 0: MAC0MS: Mode Select
This bit selects between MAC Mode and Multiply Only Mode.
0: MAC (Multiply and Accumulate) Mode.
1: Multiply Only Mode.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.2. MAC0STA: MAC0 Status

SFR Page: 0
SFR Address: 0xCF

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	MAC0HO	MAC0Z	MAC0SO	MAC0N	00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

Bits 7–4: UNUSED: Read = 0000b, Write = don't care.

Bit 3: MAC0HO: Hard Overflow Flag.
This bit is set to '1' whenever an overflow out of the MAC0OVR register occurs during a MAC operation (i.e. when MAC0OVR changes from 0x7F to 0x80 or from 0x80 to 0x7F). The hard overflow flag must be cleared in software by directly writing it to '0', or by resetting the MAC logic using the MAC0CA bit in register MAC0CF.

Bit 2: MAC0Z: Zero Flag.
This bit is set to '1' if a MAC0 operation results in an Accumulator value of zero. If the result is non-zero, this bit will be cleared to '0'.

Bit 1: MAC0SO: Soft Overflow Flag.
This bit is set to '1' when a MAC operation causes an overflow into the sign bit (bit 31) of the MAC0 Accumulator. If the overflow condition is corrected after a subsequent MAC operation, this bit is cleared to '0'.

Bit 0: MAC0N: Negative Flag.
If the MAC Accumulator result is negative, this bit will be set to '1'. If the result is positive or zero, this flag will be cleared to '0'.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.3. MAC0AH: MAC0 A High Byte

SFR Page: 0
SFR Address: 0xA5

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: High Byte (bits 15–8) of MAC0 A Register.

SFR Definition 11.4. MAC0AL: MAC0 A Low Byte

SFR Page: 0
SFR Address: 0xA4

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Low Byte (bits 7–0) of MAC0 A Register.

SFR Definition 11.5. MAC0BH: MAC0 B High Byte

SFR Page: 0
SFR Address: 0xF2

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: High Byte (bits 15–8) of MAC0 B Register.

SFR Definition 11.6. MAC0BL: MAC0 B Low Byte

SFR Page: 0
SFR Address: 0xF1

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Low Byte (bits 7–0) of MAC0 B Register.
A write to this register initiates a Multiply or Multiply and Accumulate operation.

Note: The contents of this register should not be changed by software during the first MAC0 pipeline stage.

SFR Definition 11.7. MAC0ACC3: MAC0 Accumulator Byte 3

SFR Page: 0
SFR Address: 0xD5

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Byte 3 (bits 31–24) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.8. MAC0ACC2: MAC0 Accumulator Byte 2

SFR Page: 0
SFR Address: 0xD4

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Byte 2 (bits 23–16) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.9. MAC0ACC1: MAC0 Accumulator Byte 1

SFR Page: 0
SFR Address: 0xD3

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Byte 1 (bits 15–8) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.10. MAC0ACC0: MAC0 Accumulator Byte 0

SFR Page: 0
SFR Address: 0xD2

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Byte 0 (bits 7–0) of MAC0 Accumulator.

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.11. MAC0OVR: MAC0 Accumulator Overflow

SFR Page: 0
SFR Address: 0xD6

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: MAC0 Accumulator Overflow Bits (bits 39–32).

Note: The contents of this register should not be changed by software during the first two MAC0 pipeline stages.

SFR Definition 11.12. MAC0RNDH: MAC0 Rounding Register High Byte

SFR Page: 0
SFR Address: 0xAF

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: High Byte (bits 15–8) of MAC0 Rounding Register.

SFR Definition 11.13. MAC0RNDL: MAC0 Rounding Register Low Byte

SFR Page: 0
SFR Address: 0xAE

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Low Byte (bits 7–0) of MAC0 Rounding Register.

12. Reset Sources

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

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to 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 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.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. Refer to Section “16. Oscillators” on page 169 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source (Section “22.3. Watchdog Timer Mode” on page 272 details the use of the Watchdog Timer). Program execution begins at location 0x0000.

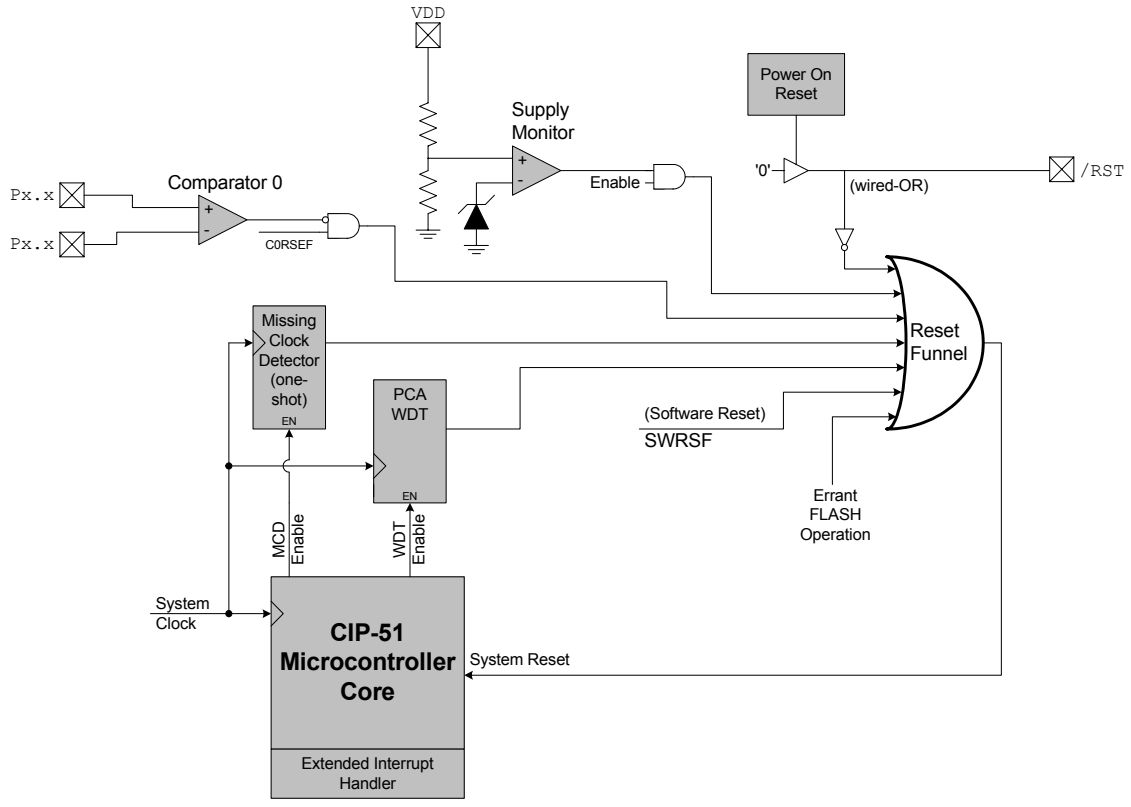


Figure 12.1. Reset Sources

12.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin is driven low until V_{DD} settles above V_{RST}. A delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 12.2. plots the power-on and V_{DD} Monitor reset timing. For ramp times less than 1 ms, the power-on reset delay ($T_{PORDelay}$) is typically less than 0.3 ms.

Note: The maximum V_{DD} ramp time is 1 ms; slower ramp times may cause the device to be released from reset before V_{DD} reaches the V_{RST} level.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic '1'. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000) software can read the PORSF flag to determine if a power-up was the cause of reset. The content of internal data memory should be assumed to be undefined after a power-on reset. The V_{DD} Monitor is enabled following a power-on reset.

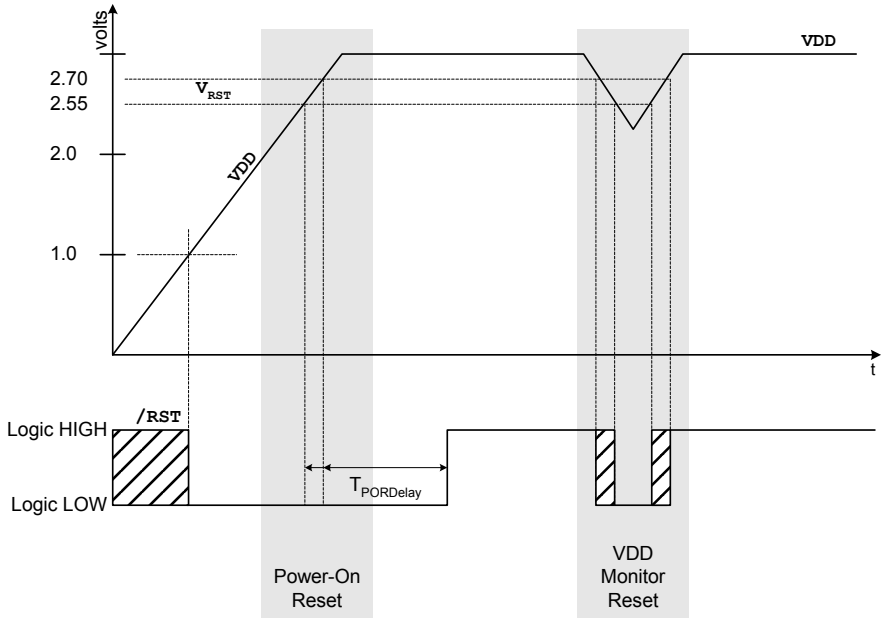


Figure 12.2. Power-On and V_{DD} Monitor Reset Timing

12.2. Power-Fail Reset/V_{DD} Monitor

When a power-down transition or power irregularity causes V_{DD} to drop below V_{RST}, the power supply monitor will drive the $\overline{RST}$ pin low and hold the CIP-51 in a reset state (see Figure 12.2). When V_{DD} returns to a level above V_{RST}, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if V_{DD} dropped below the level required for data retention. If the PORSF flag reads '1', the data may no longer be valid. The V_{DD} Monitor is enabled after power-on resets; however its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} Monitor is disabled and a software reset is performed, the V_{DD} Monitor will still be disabled after the reset. **To protect the integrity of Flash contents, the VDD Monitor must be enabled and selected as a reset source if software contains routines which erase or write Flash memory. If the VDD Monitor is not enabled, any erase or write performed on Flash memory will cause a Flash Error device reset.**

The V_{DD} Monitor must be enabled before it is selected as a reset source. Selecting the V_{DD} Monitor as a reset source before it is enabled and stabilized may cause a system reset. The procedure for configuring the V_{DD} Monitor as a reset source is shown below:

Step 1. Enable the V_{DD} Monitor (VDMEN bit in VDM0CN = '1').
Step 2. Wait for the VDD Monitor to stabilize (approximately 5 µs).
Note: This delay should be omitted if software contains routines which erase or write Flash memory.
Step 3. Select the V_{DD} Monitor as a reset source (PORSF bit in RSTSRC = '1').

See Table 12.1 for complete electrical characteristics of the V_{DD} Monitor.

Note: Software should take care not to inadvertently disable the VDD Monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to '1' to keep the VDD Monitor enabled as a reset source.

SFR Definition 12.1. VDM0CN: V_{DD} Monitor Control

SFR Page: all pages
SFR Address: 0xFF

R/W	R	R	R	R	R	R	R	Reset Value
VDMEN	VDDSTAT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: VDMEN: V_{DD} Monitor Enable.
This bit turns the V_{DD} Monitor circuit on/off. The V_{DD} Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (SFR Definition 12.2). The V_{DD} Monitor must be allowed to stabilize before it is selected as a reset source. **Selecting the V_{DD} Monitor as a reset source before it has stabilized may generate a system reset.**
0: V_{DD} Monitor Disabled.
1: V_{DD} Monitor Enabled.

Bit 6: V_{DD} STAT: V_{DD} Status.
This bit indicates the current power supply status (V_{DD} Monitor output).
0: V_{DD} is at or below the V_{DD} Monitor threshold.
1: V_{DD} is above the V_{DD} Monitor threshold.

Bits 5–0: RESERVED. Read = Variable. Write = don't care.

12.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 12.1 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

12.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than 100 µs, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read '1', signifying the MCD as the reset source; otherwise, this bit reads '0'. Writing a '1' to the MCDRSF bit enables the Missing Clock Detector; writing a '0' disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.5. Comparator0 Reset

Comparator0 can be configured as a reset source by writing a '1' to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read '1' signifying Comparator0 as the reset source; otherwise, this bit reads '0'. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in Section “22.3. Watchdog Timer Mode” on page 272; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to ‘1’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.7. Flash Error Reset

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

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to ‘1’ and a MOVX write operation targets an address above address 0x7BFF.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above address 0x7BFF.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x7BFF.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section “13.2. Security Options” on page 138).
- A Flash write or erase is attempted while the V_{DD} Monitor is disabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.8. 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 $\overline{\text{RST}}$ pin is unaffected by this reset.

SFR Definition 12.2. RSTSRC: Reset Source

SFR Page: all pages
SFR Address: 0xEF

R	R	R/W	R/W	R	R/W	R/W	R	Reset Value
–	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Note: For bits that act as both reset source enables (on a write) and reset indicator flags (on a read), read-modify-write instructions read and modify the source enable only. [This applies to bits: C0RSEF, SWRSF, MCDRSF, PORSF].

Bit 7: UNUSED. Read = 0b. Write = don't care.

Bit 6: 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.

Bit 5: C0RSEF: 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).

Bit 4: 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.

Bit 3: WDTRSF: Watchdog Timer Reset Flag.
0: Source of last reset was not a WDT timeout.
1: Source of last reset was a WDT timeout.

Bit 2: 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.

Bit 1: 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 (SFR Definition 12.1)
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.

Bit 0: PINRSF: HW Pin Reset Flag.
0: Source of last reset was not $\overline{RST}$ pin.
1: Source of last reset was $\overline{RST}$ pin.

Table 12.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{RST}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.7 V to 3.6 V	—	—	0.6	V
$\overline{RST}$ Input High Voltage		0.7 x V_{DD}	—	—	V
$\overline{RST}$ Input Low Voltage		—	—	0.7	V
$\overline{RST}$ Input Pullup Impedance		—	100	—	kΩ
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	40	—	—	µs
Minimum $\overline{RST}$ Low Time to Generate a System Reset		15	—	—	µs
V_{DD} Monitor Supply Current		—	19	40	µA

13. Flash Memory

All devices include either 32 kB (C8051F360/1/2/3/4/5/6/7) or 16 kB (C8051F368/9) of on-chip, reprogrammable Flash memory for program code or non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface, or by software using the MOVX write instructions. Once cleared to logic ‘0’, a Flash bit must be erased to set it back to logic ‘1’. Bytes should be erased (set to 0xFF) before being reprogrammed. Flash write and erase operations are automatically timed by hardware for proper execution. During a Flash erase or write, the FLBUSY bit in the FLSTAT register is set to ‘1’ (see SFR Definition 14.5). During this time, instructions that are located in the prefetch buffer or the branch target cache can be executed, but the processor will stall until the erase or write is completed if instruction data must be fetched from Flash memory. Interrupts that have been pre-loaded into the branch target cache can also be serviced at this time, if the current code is also executing from the prefetch engine or cache memory. Any interrupts that are not pre-loaded into cache, or that occur while the core is halted, will be held in a pending state during the Flash write/erase operation, and serviced in priority order once the Flash operation has completed. Refer to Table 13.2 for the electrical characteristics of the Flash memory.

13.1. Programming the Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “24. C2 Interface” on page 284. For detailed guidelines on writing or erasing Flash from firmware, please see Section “13.3. Flash Write and Erase Guidelines” on page 140.

The Flash memory can be programmed from software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic ‘1’. This directs the MOVX writes to Flash memory instead of to XRAM, which is the default target. The PSWE bit remains set until cleared by software. To avoid errant Flash writes, it is recommended that interrupts be disabled while the PSWE bit is logic ‘1’.

Flash memory is read using the MOVC instruction. MOVX reads are always directed to XRAM, regardless of the state of PSWE.

Note: To ensure the integrity of the Flash contents, the on-chip V_{DD} Monitor must be enabled 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 disabled will cause a Flash Error device reset.

A write to Flash memory can clear bits but cannot set them; only an erase operation can set bits in Flash. **A byte location to be programmed must be erased before a new value can be written**.

Write/Erase timing is automatically controlled by hardware. Note that on the 32 k Flash devices, 1024 bytes beginning at location 0x7C00 are reserved. Flash writes and erases targeting the reserved area should be avoided.

13.1.1. 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 13.2.

13.1.2. Erasing Flash Pages From Software

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

A write to Flash memory can clear bits to logic '0' but cannot set them; only an erase operation can set bits to logic '1' in Flash. **A byte location to be programmed should be erased before a new value is written.** The Flash memory is organized in 1024-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 1024-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set PSEE (PSCTL.1) to enable Flash erases.
Step 5. Set PSWE (PSCTL.0) to redirect MOVX commands to write to Flash.
Step 6. Use the MOVX instruction to write a data byte to any location within the page to be erased.
Step 7. Clear PSEE to disable Flash erases.
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Re-enable interrupts.

13.1.3. Writing Flash Memory From Software

Bytes in Flash memory can be written one byte at a time, or in small blocks. The CHBLKW bit in register CCH0CN (SFR Definition 14.1) controls whether a single byte or a block of bytes is written to Flash during a write operation. When CHBLKW is cleared to '0', the Flash will be written one byte at a time. When CHBLKW is set to '1', the Flash will be written in blocks of four bytes for addresses in code space. Block writes are performed in the same amount of time as single byte writes, which can save time when storing large amounts of data to Flash memory.

For single-byte writes to Flash, bytes are written individually, and the Flash write is performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is as follows:

Step 1. Disable interrupts.
Step 2. Clear CHBLKW (register CCH0CN) to select single-byte write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Use the MOVX instruction to write a data byte to the desired location (repeat as necessary).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Re-enable interrupts.

Steps 3–8 must be repeated for each byte to be written

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. When writing to addresses located in any of the four code banks, a Flash write block is four bytes long, from addresses ending in 00b to addresses ending in 11b. Writes must be performed sequentially (i.e. addresses ending in 00b, 01b, 10b, and 11b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 11b. The Flash write will be performed following the MOVX write that targets the address ending in 1b. If any bytes in the block do not need to be updated in Flash, they should be written to 0xFF. The recommended procedure for writing Flash in blocks is as follows:

Step 1. Disable interrupts.
Step 2. Set CHBLKW (register CCH0CN) to select block write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Using the MOVX instruction, write the first data byte to the first block location (ending in 00b).
Step 8. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 9. Write the first key code to FLKEY: 0xA5.
Step 10. Write the second key code to FLKEY: 0xF1.
Step 11. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 12. Clear the PSEE bit (register PSCTL).
Step 13. Using the MOVX instruction, write the second data byte to the second block location (ending in 01b).
Step 14. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 15. Write the first key code to FLKEY: 0xA5.
Step 16. Write the second key code to FLKEY: 0xF1.
Step 17. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 18. Clear the PSEE bit (register PSCTL).
Step 19. Using the MOVX instruction, write the third data byte to the third block location (ending in 10b).
Step 20. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 21. Write the first key code to FLKEY: 0xA5.
Step 22. Write the second key code to FLKEY: 0xF1.
Step 23. Set PSWE (register PSCTL) to redirect MOVX commands to write to Flash.
Step 24. Clear the PSEE bit (register PSCTL).
Step 25. Using the MOVX instruction, write the fourth data byte to the last block location (ending in 11b).
Step 26. Clear the PSWE bit to redirect MOVX commands to the XRAM data space.
Step 27. Re-enable interrupts.

Steps 3-26 must be repeated for each block to be written.

13.1.4. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written and erased using the MOVX write instruction (as described in Section 13.1.2 and Section 13.1.3) and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

13.2. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to '1' before software can modify the Flash memory; both PSWE and PSEE must be set to '1' before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock *n* 1024-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x03FF), where *n* is the 1's complement number represented by the Security Lock Byte. **Note that the page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are '1') and locked when any other Flash pages are locked (any bit of the Lock Byte is '0').** See the example below for an C8051F360.

Security Lock Byte:	11111101b
1's Complement:	00000010b
Flash pages locked:	3 (First two Flash pages + Lock Byte Page)
Addresses locked:	0x0000 to 0x07FF (first two Flash pages) and 0x7800 to 0x7BFF (Lock Byte Page)

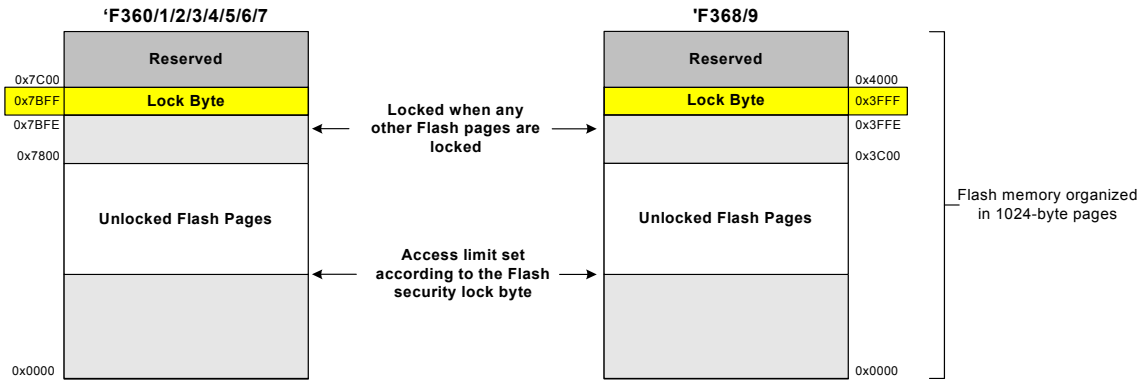


Figure 13.1. Flash Program Memory Map

13.2.1. Summary of Flash Security Options

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 13.1 summarizes the Flash security features of the C8051F36x devices.

Table 13.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	FEDR	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	FEDR	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	FEDR	FEDR
Erase page containing Lock Byte - Unlock all pages (if any page is locked)	Only C2DE	FEDR	FEDR
Lock additional pages (change '1's to '0's in the Lock Byte)	Not Permitted	FEDR	FEDR
Unlock individual pages (change '0's to '1's in the Lock Byte)	Not Permitted	FEDR	FEDR
Read, Write or Erase Reserved Area	Not Permitted	FEDR	FEDR

C2DE - C2 Device Erase (Erases all Flash pages including the page containing the Lock Byte)
FEDR - Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is '1' after reset)

- All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).
- Locking any Flash page also locks the page containing the Lock Byte.
- Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.
- If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.

13.3. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of V_{DD}, system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

To help prevent the accidental modification of Flash by firmware, the V_{DD} Monitor must be enabled and enabled as a reset source on C8051F36x devices for the Flash to be successfully modified. **If either the V_{DD} Monitor or the V_{DD} Monitor reset source is not enabled, a Flash Error Device Reset will be generated when the firmware attempts to modify the Flash.**

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

13.3.1. V_{DD} Maintenance and the V_{DD} 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 V_{DD} rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external V_{DD} brownout circuit to the /RST pin of the device that holds the device in reset until V_{DD} reaches V_{RST} and re-asserts /RST if V_{DD} drops below V_{RST}. Please see Table 12.1, “Reset Electrical Characteristics,” on page 134 for more information on the VDD Monitor Threshold voltage (V_{RST}).
3. Keep the on-chip V_{DD} Monitor enabled and enable the V_{DD} 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 V_{DD} Monitor and enabling the V_{DD} 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 C8051F36x devices, both the V_{DD} Monitor and the V_{DD} Monitor reset source must be enabled to write or erase Flash without generating a Flash Error Device Reset.

4. As an added precaution, explicitly enable the V_{DD} Monitor and enable the V_{DD} Monitor as a reset source inside the functions that write and erase Flash memory. The V_{DD} 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.

13.3.2. 16.4.2 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.

13.3.3. System Clock

12. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

SFR Definition 13.1. PSCTL: Program Store Read/Write Control

SFR Page: 0
SFR Address: 0x8F

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	–	–	–	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–2: UNUSED. Read = 000000b, Write = don't care.
Bit 1: PSEE: Program Store Erase Enable.
Setting this bit allows an entire page of the Flash program memory to be erased provided the PSWE bit is also set. After setting this bit, a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter. **Note: The Flash page containing the Read Lock Byte and Write/Erase Lock Byte cannot be erased by software.**
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.
Bit 0: PSWE: Program Store Write Enable.
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The location must be erased prior to writing data.
0: Write to Flash program memory disabled. MOVX write operations target External RAM.
1: Write to Flash program memory enabled. MOVX write operations target Flash memory.

SFR Definition 13.2. FLKEY: Flash Lock and Key

SFR Page: 0
SFR Address: 0xB7

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	

Bits 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 disabled until the next reset.

13.4. Flash Read Timing

On reset, the C8051F36x Flash read timing is configured for operation with system clocks up to 25 MHz. If the system clock will not be increased above 25 MHz, then the Flash timing registers may be left at their reset value.

For every Flash read or fetch, the system provides an internal Flash read strobe to the Flash memory. The Flash read strobe lasts for one or two system clock cycles, based on the FLRT bits (FLSCL.4 and FLSCL.5). **If the system clock is greater than 25 MHz, the FLRT bit must be changed to the appropriate setting.** Otherwise, data read or fetched from Flash may not represent the actual contents of Flash. When the Flash read strobe is asserted, Flash memory is active. When it is de-asserted, Flash memory is in a low power state.

The recommended procedure for updating FLRT is:

Step 1. Select SYSCLK to 25 MHz or less.
Step 2. Disable the prefetch engine (CHPFEN = '0' in CCH0CN register).
Step 3. Set the FLRT bits to the appropriate setting for the SYSCLK.
Step 4. Enable the prefetch engine (CHPFEN = '1' in CCH0CN register).

SFR Definition 13.3. FLSCL: Flash Memory Control

SFR Page: 0
SFR Address: 0xB6

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	FLRT		Reserved	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 00b. Write = don't care.
Bits 5–4: FLRT: Flash Read Time.
These bits should be programmed to the smallest allowed value, according to the system clock speed.
00: SYSCLK ≤ 25 MHz.
01: SYSCLK ≤ 50 MHz.
10: SYSCLK ≤ 75 MHz.
11: SYSCLK ≤ 100 MHz.
Bits 3–0: RESERVED. Read = 0000b. Must Write 0000b.

Important Note: When changing the FLRT bits to a lower setting (e.g. when changing from a value of 11b to 00b), cache reads, cache writes, and the prefetch engine should be disabled using the CCH0CN register (see SFR Definition 14.1).

Table 13.2. Flash Electrical Characteristics

V_{DD} = 2.7 to 3.6 V; –40 to +85 °C.

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F360/1/2/3/4/5/6/7		32768*		Bytes
	C8051F368/9		16384		
Endurance		20 k	250 k		Erase/Write
Erase Cycle Time		8	10	12	ms
Write Cycle Time		37	47	57	µs
***Note:** 1024 Bytes at location 0x7C00 to 0x7FFF are reserved.					

14. Branch Target Cache

The C8051F36x device families incorporate a 32x4 byte branch target cache with a 4-byte prefetch engine. Because the access time of the Flash memory is 40 ns, and the minimum instruction time is 10 ns (C8051F360/1/2/3/4/5/6/7) or 20 ns (C8051F368/9), the branch target cache and prefetch engine are necessary for full-speed code execution. Instructions are read from Flash memory four bytes at a time by the prefetch engine, and given to the CIP-51 processor core to execute. When running linear code (code without any jumps or branches), the prefetch engine alone allows instructions to be executed at full speed. When a code branch occurs, a search is performed for the branch target (destination address) in the cache. If the branch target information is found in the cache (called a “cache hit”), the instruction data is read from the cache and immediately returned to the CIP-51 with no delay in code execution. If the branch target is not found in the cache (called a “cache miss”), the processor may be stalled for up to four clock cycles while the next set of four instructions is retrieved from Flash memory. Each time a cache miss occurs, the requested instruction data is written to the cache if allowed by the current cache settings. A data flow diagram of the interaction between the CIP-51 and the Branch Target Cache and Prefetch Engine is shown in Figure 14.1.

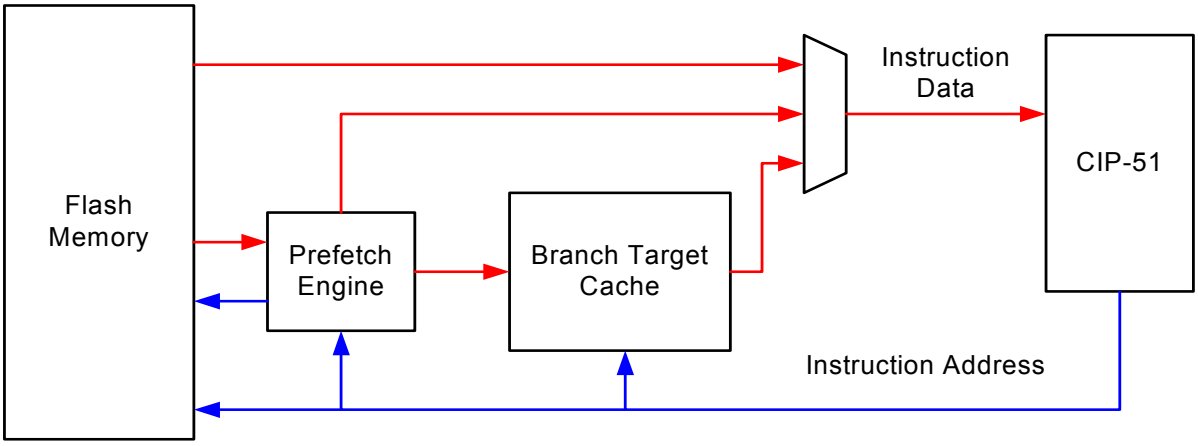


Figure 14.1. Branch Target Cache Data Flow

14.1. Cache and Prefetch Operation

The branch target cache maintains two sets of memory locations: “slots” and “tags”. A slot is where the cached instruction data from Flash is stored. Each slot holds four consecutive code bytes. A tag contains the 13 most significant bits of the corresponding Flash address for each four-byte slot. Thus, instruction data is always cached along four-byte boundaries in code space. A tag also contains a “valid bit”, which indicates whether a cache location contains valid instruction data. A special cache location (called the linear tag and slot), is reserved for use by the prefetch engine. The cache organization is shown in Figure 14.2. Each time a Flash read is requested, the address is compared with all valid cache tag locations (including the linear tag). If any of the tag locations match the requested address, the data from that slot is immediately provided to the CIP-51. If the requested address matches a location that is currently being read by the prefetch engine, the CIP-51 will be stalled until the read is complete. If a match is not found, the current prefetch operation is abandoned, and a new prefetch operation is initiated for the requested instruction data. When the prefetch operation is finished, the CIP-51 begins executing the instructions that were retrieved, and the prefetch engine begins reading the next four-byte word from Flash memory. If the newly-fetched data also meets the criteria necessary to be cached, it will be written to the cache in the slot indicated by the current replacement algorithm.

The replacement algorithm is selected with the Cache Algorithm bit, CHALGM (CCH0TN.3). When CHALGM is cleared to ‘0’, the cache will use the rebound algorithm to replace cache locations. The rebound algorithm replaces locations in order from the beginning of cache memory to the end, and then from the end of cache memory to the beginning. When CHALGM is set to ‘1’, the cache will use the pseudo-random algorithm to replace cache locations. The pseudo-random algorithm uses a pseudo-random number to determine which cache location to replace. The cache can be manually emptied by writing a ‘1’ to the CHFLUSH bit (CCH0CN.4).

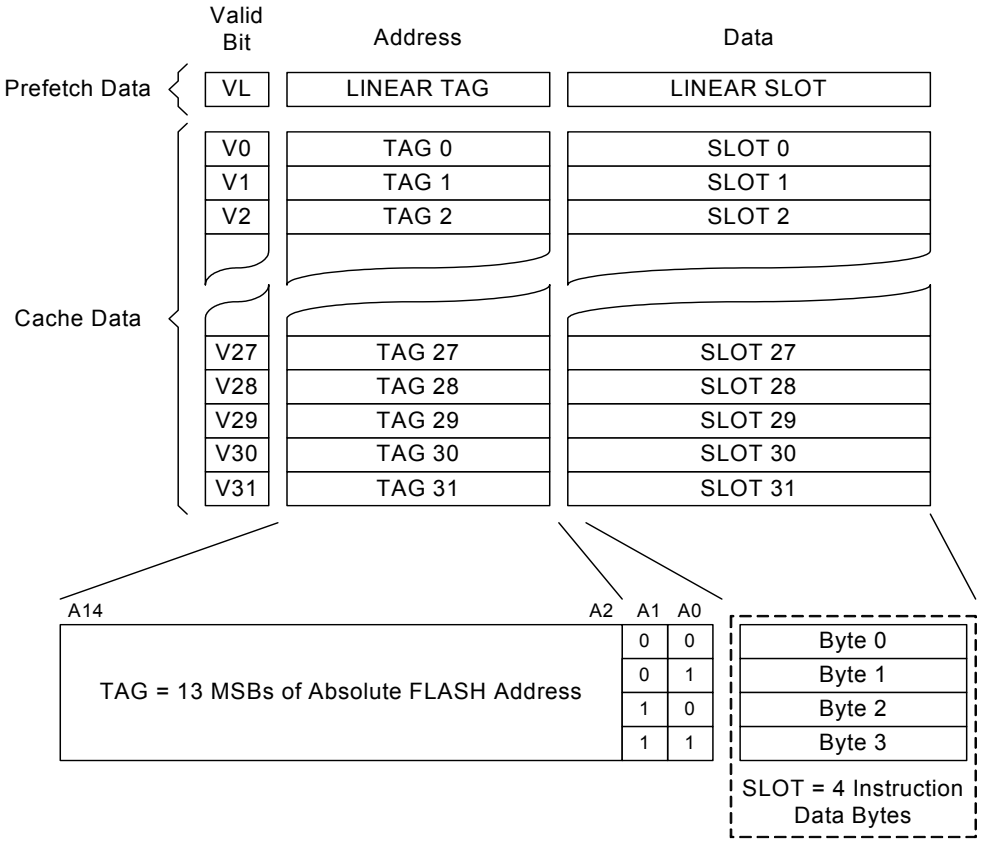


Figure 14.2. Branch Target Cache Organization

14.2. Cache and Prefetch Optimization

By default, the branch target cache is configured to provide code speed improvements for a broad range of circumstances. **In most applications, the cache control registers should be left in their reset states.** Sometimes it is desirable to optimize the execution time of a specific routine or critical timing loop. The branch target cache includes options to exclude caching of certain types of data, as well as the ability to pre-load and lock time-critical branch locations to optimize execution speed.

The most basic level of cache control is implemented with the Cache Miss Penalty Threshold bits, CHMSTH (CCH0TN.1–0). If the processor is stalled during a prefetch operation for more clock cycles than the number stored in CHMSTH, the requested data will be cached when it becomes available. The CHMSTH bits are set to zero by default, meaning that any time the processor is stalled, the new data will be cached. If, for example, CHMSTH is equal to 2, any cache miss causing a delay of 3 or 4 clock cycles will be cached, while a cache miss causing a delay of 1–2 clock cycles will not be cached.

Certain types of instruction data or certain blocks of code can also be excluded from caching. The destinations of RETI instructions are, by default, excluded from caching. To enable caching of RETI destinations, the CHRETI bit (CCH0CN.3) can be set to '1'. It is generally not beneficial to cache RETI destinations unless the same instruction is likely to be interrupted repeatedly (such as a code loop that is waiting for an interrupt to happen). Instructions that are part of an interrupt service routine (ISR) can also be excluded from caching. By default, ISR instructions are cached, but this can be disabled by clearing the CHISR bit (CCH0CN.2) to '0'. The other information that can be explicitly excluded from caching are the data returned by MOVC instructions. Clearing the CHMOV bit (CCH0CN.1) to '0' will disable caching of MOVC data. If MOVC caching is allowed, it can be restricted to only use slot 0 for the MOVC information (excluding cache push operations). The CHFIXM bit (CCH0TN.2) controls this behavior.

Further cache control can be implemented by disabling all cache writes. Cache writes can be disabled by clearing the CHWREN bit (CCH0CN.7) to '0'. Although normal cache writes (such as those after a cache miss) are disabled, data can still be written to the cache with a cache push operation. Disabling cache writes can be used to prevent a non-critical section of code from changing the cache contents. Note that regardless of the value of CHWREN, a Flash write or erase operation automatically removes the affected bytes from the cache. Cache reads and the prefetch engine can also be individually disabled. Disabling cache reads forces all instructions data to execute from Flash memory or from the prefetch engine. To disable cache reads, the CHRDEN bit (CCH0CN.6) can be cleared to '0'. Note that when cache reads are disabled, cache writes will still occur (if CHWREN is set to '1'). Disabling the prefetch engine is accomplished using the CHPFEN bit (CCH0CN.5). When this bit is cleared to '0', the prefetch engine will be disabled. If both CHPFEN and CHRDEN are '0', code will execute at a fixed rate, as instructions become available from the Flash memory.

Cache locations can also be pre-loaded and locked with time-critical branch destinations. For example, in a system with an ISR that must respond as fast as possible, the entry point for the ISR can be locked into a cache location to minimize the response latency of the ISR. Up to 30 locations can be locked into the cache at one time. Instructions are locked into cache by enabling cache push operations with the CHPUSH bit (CCH0LC.7). When CHPUSH is set to '1', a MOVC instruction will cause the four-byte segment containing the data byte to be written to the cache slot location indicated by CHSLOT (CCH0LC.4-0). CHSLOT is them decremented to point to the next lockable cache location. This process is called a cache push operation. Cache locations that are above CHSLOT are "locked", and cannot be changed by the processor core, as shown in Figure 14.3. Cache locations can be unlocked by using a cache pop operation. A cache pop is performed by writing a '1' to the CHPOP bit (CCH0LC.6). When a cache pop is initiated, the value of CHSLOT is incremented. This unlocks the most recently locked cache location, but does not remove the information from the cache. Note that a cache pop should not be initiated if CHSLOT is equal to 11110b. Doing so may have an adverse effect on cache performance. **Important: Although locking cache location 1 is not explicitly disabled by hardware, the entire cache will be unlocked when CHSLOT is equal to 00000b. Therefore, cache locations 1 and 0 must remain unlocked at all times.**

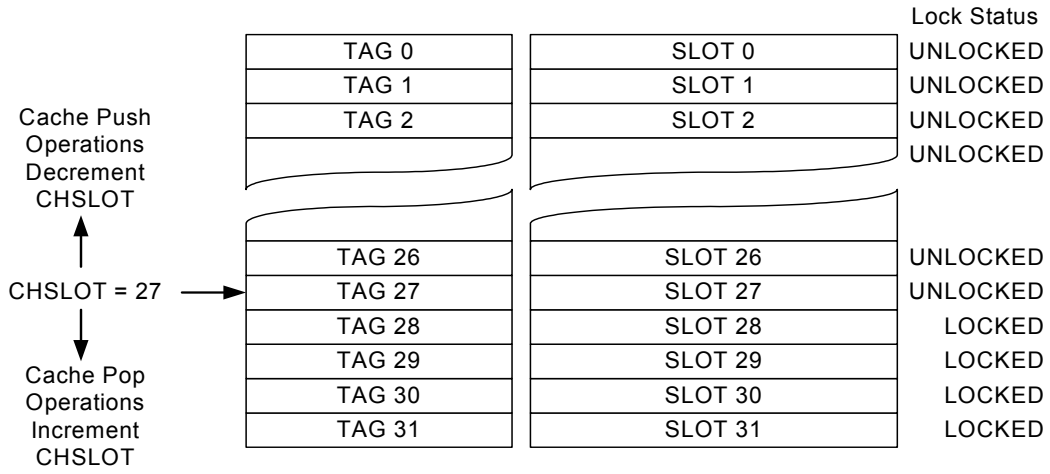


Figure 14.3. Cache Lock Operation

SFR Definition 14.1. CCH0CN: Cache Control

SFR Page: F
SFR Address: 0x84

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CHWREN	CHRDEN	CHPFEN	CHFLSH	CHRETI	CHISR	CHMOVC	CHBLKW	11100110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CHWREN: Cache Write Enable.
This bit enables the processor to write to the cache memory.
0: Cache contents are not allowed to change, except during Flash writes/erasures or cache locks.
1: Writes to cache memory are allowed.

Bit 6: CHRDEN: Cache Read Enable.
This bit enables the processor to read instructions from the cache memory.
0: All instruction data comes from Flash memory or the prefetch engine.
1: Instruction data is obtained from cache (when available).

Bit 5: CHPFEN: Cache Prefetch Enable.
This bit enables the prefetch engine.
0: Prefetch engine is disabled.
1: Prefetch engine is enabled.

Bit 4: CHFLSH: Cache Flush.
When written to a ‘1’, this bit clears the cache contents. This bit always reads ‘0’.

Bit 3: CHRETI: Cache RETI Destination Enable.
This bit enables the destination of a RETI address to be cached.
0: Destinations of RETI instructions will not be cached.
1: RETI destinations will be cached.

Bit 2: CHISR: Cache ISR Enable.
This bit allows instructions which are part of an Interrupt Service Routine (ISR) to be cached.
0: Instructions in ISRs will not be loaded into cache memory.
1: Instructions in ISRs can be cached.

Bit 1: CHMOVC: Cache MOVC Enable.
This bit allows data requested by a MOVC instruction to be loaded into the cache memory.
0: Data requested by MOVC instructions will not be cached.
1: Data requested by MOVC instructions will be loaded into cache memory.

Bit 0: CHBLKW: Block Write Enable.
This bit allows block writes to Flash memory from software.
0: Each byte of a software Flash write is written individually.
1: Flash bytes are written in groups of four (for code space writes).

SFR Definition 14.2. CCH0TN: Cache Tuning

SFR Page: F
SFR Address: 0xC9

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CHMSCTL				CHALGM	CHFIXM	CHMSTH		00000100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–4: CHMSCTL: Cache Miss Penalty Accumulator (Bits 4–1).
These are bits 4-1 of the Cache Miss Penalty Accumulator. To read these bits, they must first be latched by reading the CHMSCTH bits in the CCH0MA Register (See SFR Definition 14.4).

Bit 3: CHALGM: Cache Algorithm Select.
This bit selects the cache replacement algorithm.
0: Cache uses Rebound algorithm.
1: Cache uses Pseudo-random algorithm.

Bit 2: CHFIXM: Cache Fix MOVC Enable.
This bit forces MOVC writes to the cache memory to use slot 0.
0: MOVC data is written according to the current algorithm selected by the CHALGM bit.
1: MOVC data is always written to cache slot 0.

Bits 1–0: CHMSTH: Cache Miss Penalty Threshold.
These bits determine when missed instruction data will be cached.
If data takes longer than CHMSTH clocks to obtain, it will be cached.

SFR Definition 14.3. CCH0LC: Cache Lock Control

SFR Page: F
SFR Address: 0xD2

R/W	R/W	R	R	R	R	R	R	Reset Value
CHPUSH	CHPOP	RESERVED	CHSLOT					00011111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CHPUSH: Cache Push Enable.
This bit enables cache push operations, which will lock information in cache slots using MOVC instructions.
0: Cache push operations are disabled.
1: Cache push operations are enabled. When a MOVC read is executed, the requested 4-byte segment containing the data is locked into the cache at the location indicated by CHSLOT, and CHSLOT is decremented.

Note: No more than 30 cache slots should be locked at one time, since the entire cache will be unlocked when CHSLOT is equal to 0.

Bit 6: CHPOP: Cache Pop.
Writing a '1' to this bit will increment CHSLOT and then unlock that location. This bit always reads '0'. Note that Cache Pop operations should not be performed while CHSLOT = 11110b. "Pop"ing more Cache slots than have been "Push"ed will have indeterminate results on the Cache performance.

Bit 5: RESERVED. Read = 0b. Must Write 0b.

Bits 4–0: CHSLOT: Cache Slot Pointer.
These read-only bits are the pointer into the cache lock stack. Locations above CHSLOT are locked, and will not be changed by the processor, except when CHSLOT equals 0.

SFR Definition 14.4. CCH0MA: Cache Miss Accumulator

SFR Page: F
SFR Address: 0xD3

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CHMSOV	CHMSCTH							00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CHMSOV: Cache Miss Penalty Overflow.
This bit indicates when the Cache Miss Penalty Accumulator has overflowed since it was last written.
0: The Cache Miss Penalty Accumulator has not overflowed since it was last written.
1: An overflow of the Cache Miss Penalty Accumulator has occurred since it was last written.

Bits 6–0: CHMSCTH: Cache Miss Penalty Accumulator (bits 11–5)
These are bits 11-5 of the Cache Miss Penalty Accumulator. The next four bits (bits 4-1) are stored in CHMSCTL in the CCH0TN register.
The Cache Miss Penalty Accumulator is incremented every clock cycle that the processor is delayed due to a cache miss. This is primarily used as a diagnostic feature, when optimizing code for execution speed.
Writing to CHMSCTH clears the lower 5 bits of the Cache Miss Penalty Accumulator.
Reading from CHMSCTH returns the current value of CHMSTCH, and latches bits 4-1 into CHMSTCL so that they can be read. Because bit 0 of the Cache Miss Penalty Accumulator is not available, the Cumulative Miss Penalty is equal to 2 * (CCHMSTCH:CCHMSTCL).

SFR Definition 14.5. FLSTAT: Flash Status

SFR Page: F
SFR Address: 0xAC

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FLBUSY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7–1: RESERVED. Read = 0000000b. Must Write 0000000b.

Bit 0: FLBUSY: Flash Busy
This bit indicates when a Flash write or erase operation is in progress.
0: Flash is idle or reading.
1: Flash write/erase operation is currently in progress.

15. External Data Memory Interface and On-Chip XRAM

For C8051F36x devices, 1k Bytes of RAM are included on-chip and mapped into the external data memory space (XRAM). Additionally, an External Memory Interface (EMIF) is available on the C8051F360/3 devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 15.1). Note: the MOVX instruction can also be used for writing to the FLASH memory. See Section "13. Flash Memory" on page 135 for details. The MOVX instruction accesses XRAM by default.

15.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

15.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

15.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

15.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Configure the Output Modes of the associated port pins as either push-pull or open-drain (push-pull is most common), and skip the associated pins in the crossbar.
2. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic ‘1’).
3. Select Multiplexed mode or Non-multiplexed mode.
4. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
5. Set up timing to interface with off-chip memory or peripherals.

Each of these five steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in SFR Definition 15.2.

15.3. Port Configuration

The External Memory Interface appears on Ports 1, 2 (non-multiplexed mode only), 3, and 4 when it is used for off-chip memory access. When the EMIF is used in multiplexed mode, the Crossbar should be configured to skip over the ALE control line (P0.0) using the P0SKIP register. The other control lines, /RD (P4.4) and /WR (P4.5), are not available on the Crossbar and do not need to be skipped. For more information about configuring the Crossbar, see Section “17.3. General Purpose Port I/O” on page 190. The EMIF pinout is shown in Table 15.1 on page 155.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar settings for those pins. See Section “17. Port Input/Output” on page 183 for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to ‘park’ the External Memory Interface pins in a dormant state, most commonly by setting them to a logic ‘1’**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode.

Table 15.1. EMIF Pinout (C8051F360/3)

Multiplexed Mode		Non Multiplexed Mode	
Signal Name	**Port Pin**	**Signal Name**	**Port Pin**
/RD	P4.4	/RD	P4.4
/WR	P4.5	/WR	P4.5
ALE	P0.0	ALE	P0.0
D0/A0	P1.0	D0	P1.0
D1/A1	P1.1	D1	P1.1
D2/A2	P1.2	D2	P1.2
D3/A3	P1.3	D3	P1.3
D4/A4	P1.4	D4	P1.4
D5/A5	P1.5	D5	P1.5
D6/A6	P1.6	D6	P1.6
D7/A7	P1.7	D7	P1.7
A8	P3.4	A0	P2.0
A9	P3.5	A1	P2.1
A10	P3.6	A2	P2.2
A11	P3.7	A3	P2.3
A12	P4.0	A4	P2.4
A13	P4.1	A5	P2.5
A14	P4.2	A6	P2.6
A15	P4.3	A7	P2.7
–	–	A8	P3.4
–	–	A9	P3.5
–	–	A10	P3.6
–	–	A11	P3.7
–	–	A12	P4.0
–	–	A13	P4.1
–	–	A14	P4.2
–	–	A15	P4.3

SFR Definition 15.1. EMI0CN: External Memory Interface Control

SFR Page: all pages
SFR Address: 0xAA

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PGSEL7	PGSEL6	PGSEL5	PGSEL4	PGSEL3	PGSEL2	PGSEL1	PGSEL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: PGSEL[7:0]: XRAM Page Select Bits.
The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM.
0x00: 0x0000 to 0x00FF
0x01: 0x0100 to 0x01FF
...
0xFE: 0xFE00 to 0xFEFF
0xFF: 0xFF00 to 0xFFFF

SFR Definition 15.2. EMI0CF: External Memory Configuration

SFR Page: F
SFR Address: 0xC7

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	EMD2	EMD1	EMD0	EALE1	EALE0	00000011
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–5: UNUSED. Read = 000b. Write = don't care.

Bit 4: EMD2: EMIF Multiplex Mode Select.
0: EMIF operates in multiplexed address/data mode.
1: EMIF operates in non-multiplexed mode (separate address and data pins).

Bits 3–2: EMD1–0: EMIF Operating Mode Select.
These bits control the operating mode of the External Memory Interface.
00: Internal Only: MOVX accesses on-chip XRAM only. All effective addresses alias to on-chip memory space.
01: Split Mode without Bank Select: Accesses below the 1 k boundary are directed on-chip. Accesses above the 1 k boundary are directed off-chip. 8-bit off-chip MOVX operations use the current contents of the Address High port latches to resolve upper address byte. Note that in order to access off-chip space, EMI0CN must be set to a page that is not contained in the on-chip address space.
10: Split Mode with Bank Select: Accesses below the 1 k boundary are directed on-chip. Accesses above the 1 k boundary are directed off-chip. 8-bit off-chip MOVX operations use the contents of EMI0CN to determine the high-byte of the address.
11: External Only: MOVX accesses off-chip XRAM only. On-chip XRAM is not visible to the CPU.

Bits 1–0: EALE1–0: ALE Pulse-Width Select Bits (only has effect when EMD2 = 0).
00: ALE high and ALE low pulse width = 1 SYSCLK cycle.
01: ALE high and ALE low pulse width = 2 SYSCLK cycles.
10: ALE high and ALE low pulse width = 3 SYSCLK cycles.
11: ALE high and ALE low pulse width = 4 SYSCLK cycles.

15.4. Multiplexed and Non-multiplexed Selection

The External Memory Interface is capable of acting in a Multiplexed mode or a Non-multiplexed mode, depending on the state of the EMD2 (EMI0CF.4) bit.

15.4.1. Multiplexed Configuration

In Multiplexed mode, the Data Bus and the lower 8-bits of the Address Bus share the same Port pins: AD[7:0]. In this mode, an external latch (74HC373 or equivalent logic gate) is used to hold the lower 8-bits of the RAM address. The external latch is controlled by the ALE (Address Latch Enable) signal, which is driven by the External Memory Interface logic. An example of a Multiplexed Configuration is shown in Figure 15.1.

In Multiplexed mode, the external MOVX operation can be broken into two phases delineated by the state of the ALE signal. During the first phase, ALE is high and the lower 8-bits of the Address Bus are presented to AD[7:0]. During this phase, the address latch is configured such that the 'Q' outputs reflect the states of the 'D' inputs. When ALE falls, signaling the beginning of the second phase, the address latch outputs remain fixed and are no longer dependent on the latch inputs. Later in the second phase, the Data Bus controls the state of the AD[7:0] port at the time /RD or /WR is asserted.

See Section "15.6.2. Multiplexed Mode" on page 165 for more information.

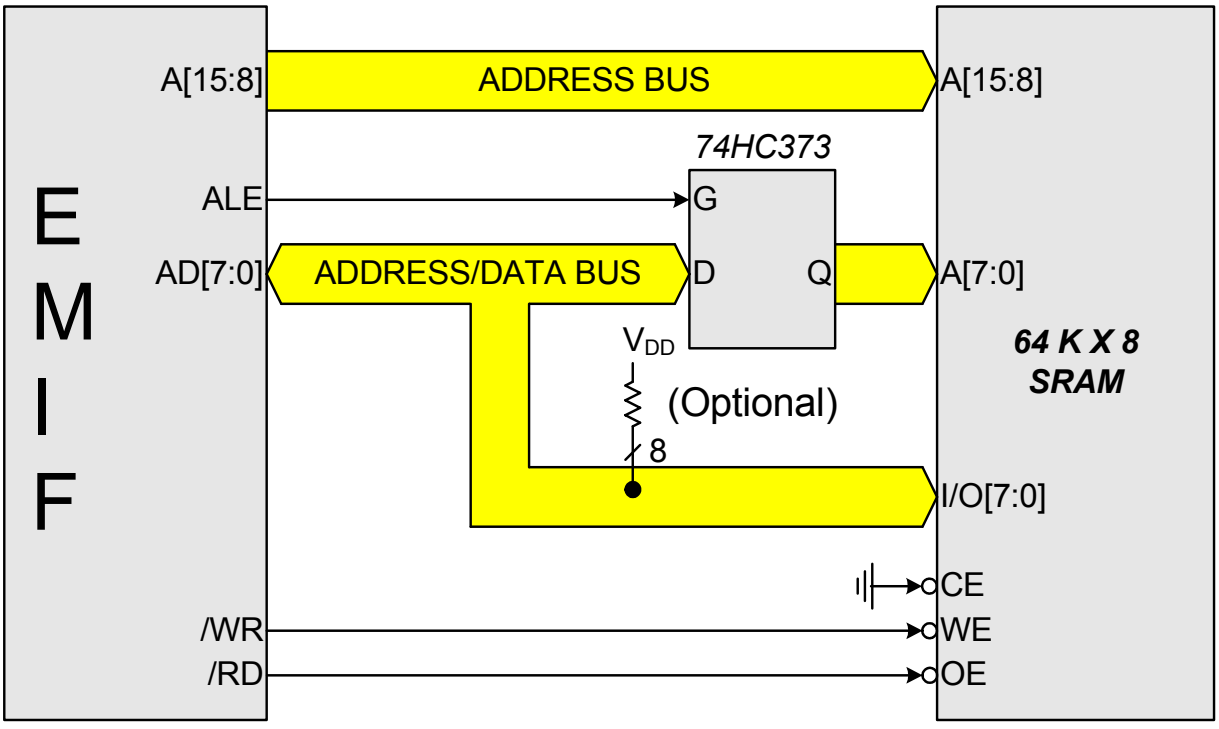


Figure 15.1. Multiplexed Configuration Example

15.4.2. Non-multiplexed Configuration

In Non-multiplexed mode, the Data Bus and the Address Bus pins are not shared. An example of a Non-multiplexed Configuration is shown in Figure 15.2. See Section “15.6.1. Non-multiplexed Mode” on page 162 for more information about Non-multiplexed operation.

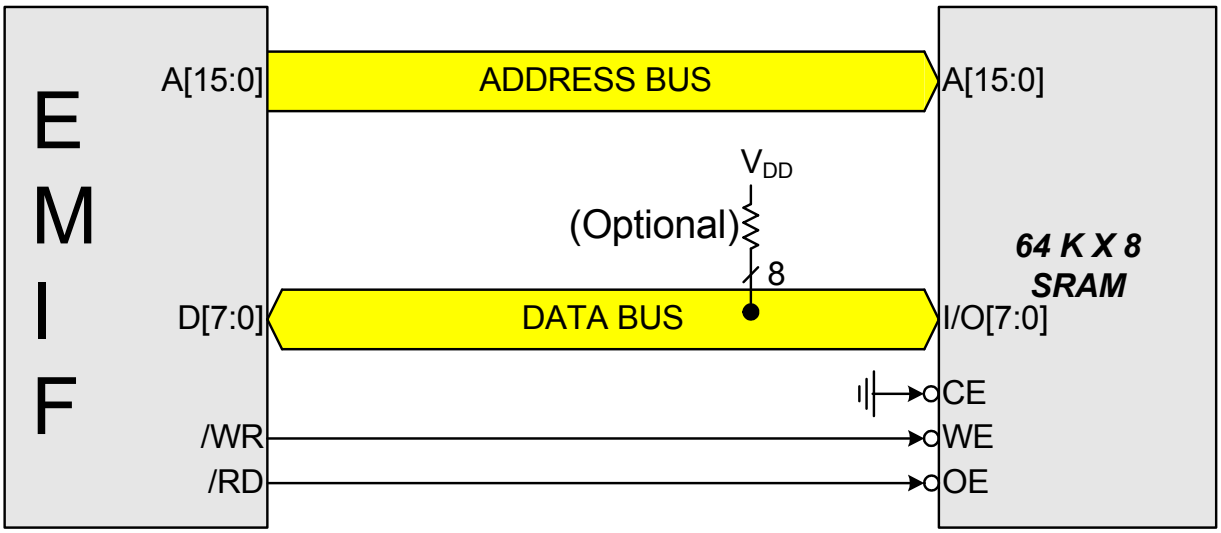


Figure 15.2. Non-multiplexed Configuration Example

15.5. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 15.3, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 15.2). These modes are summarized below. More information about the different modes can be found in Section “15.6. Timing” on page 160.

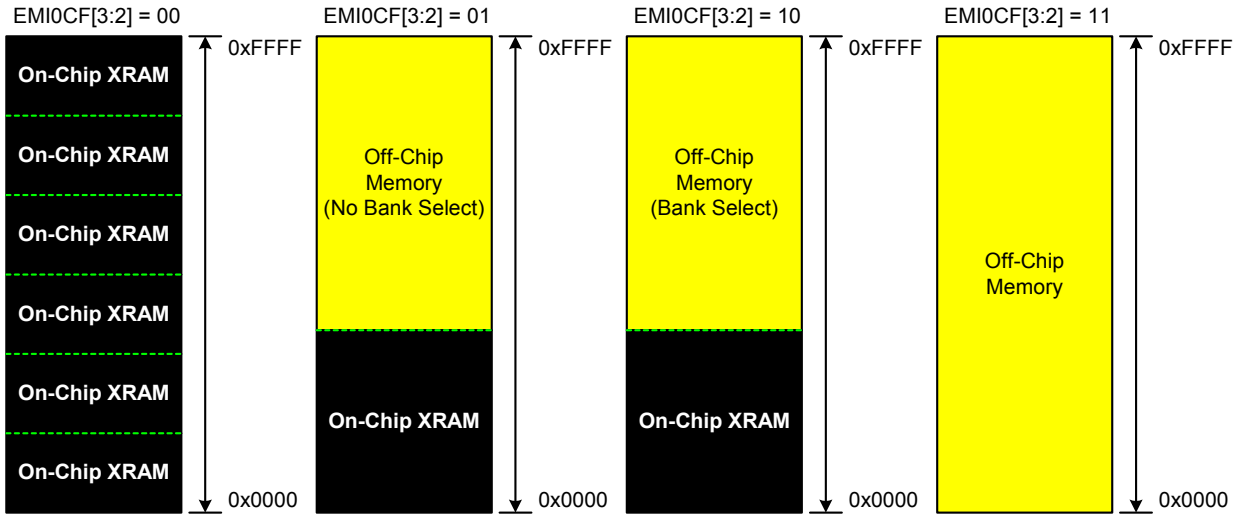


Figure 15.3. EMIF Operating Modes

15.5.1. Internal XRAM Only

When EMI0CF.[3:2] are set to ‘00’, all MOVX instructions will target the internal XRAM space on the device. Memory accesses to addresses beyond the populated space will wrap on 1k boundaries. As an example, the addresses 0x0400 and 0x1000 both evaluate to address 0x0000 in on-chip XRAM space.

- 8-bit MOVX operations use the contents of EMI0CN to determine the high-byte of the effective address and R0 or R1 to determine the low-byte of the effective address.
- 16-bit MOVX operations use the contents of the 16-bit DPTR to determine the effective address.

15.5.2. Split Mode without Bank Select

When EMI0CF.[3:2] are set to ‘01’, 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. However, in the “No Bank Select” mode, an 8-bit MOVX operation will not drive the upper 8-bits A[15:8] of the Address Bus during an off-chip access. This allows the user to manipulate the upper address bits at will by setting the Port state directly via the port latches. This behavior is in contrast with “Split Mode with Bank Select” described below. The lower 8-bits of the Address Bus A[7:0] are driven, determined by R0 or R1.
- 16-bit MOVX operations use the contents of DPTR to determine whether the memory access is on-chip or off-chip, and unlike 8-bit MOVX operations, the full 16-bits of the Address Bus A[15:0] are driven during the off-chip transaction.

15.5.3. 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.

15.5.4. 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.

15.6. 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, /RD and /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 15.3, 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 /RD or /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 /ALE + 1 for /RD or /WR + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 15.2 lists the AC parameters for the External Memory Interface, and Figure 15.4 through Figure 15.9 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

SFR Definition 15.3. EMI0TC: External Memory Timing Control

SFR Page: F
SFR Address: 0xF7

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EAS1	EAS0	ERW3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: EAS1–0: EMIF Address Setup Time Bits.
00: Address setup time = 0 SYSCLK cycles.
01: Address setup time = 1 SYSCLK cycle.
10: Address setup time = 2 SYSCLK cycles.
11: Address setup time = 3 SYSCLK cycles.

Bits 5–2: EWR3–0: EMIF /WR and /RD Pulse-Width Control Bits.
0000: /WR and /RD pulse width = 1 SYSCLK cycle.
0001: /WR and /RD pulse width = 2 SYSCLK cycles.
0010: /WR and /RD pulse width = 3 SYSCLK cycles.
0011: /WR and /RD pulse width = 4 SYSCLK cycles.
0100: /WR and /RD pulse width = 5 SYSCLK cycles.
0101: /WR and /RD pulse width = 6 SYSCLK cycles.
0110: /WR and /RD pulse width = 7 SYSCLK cycles.
0111: /WR and /RD pulse width = 8 SYSCLK cycles.
1000: /WR and /RD pulse width = 9 SYSCLK cycles.
1001: /WR and /RD pulse width = 10 SYSCLK cycles.
1010: /WR and /RD pulse width = 11 SYSCLK cycles.
1011: /WR and /RD pulse width = 12 SYSCLK cycles.
1100: /WR and /RD pulse width = 13 SYSCLK cycles.
1101: /WR and /RD pulse width = 14 SYSCLK cycles.
1110: /WR and /RD pulse width = 15 SYSCLK cycles.
1111: /WR and /RD pulse width = 16 SYSCLK cycles.

Bits 1–0: EAH1–0: EMIF Address Hold Time Bits.
00: Address hold time = 0 SYSCLK cycles.
01: Address hold time = 1 SYSCLK cycle.
10: Address hold time = 2 SYSCLK cycles.
11: Address hold time = 3 SYSCLK cycles.

15.6.1. Non-multiplexed Mode

15.6.1.1.16-bit MOVX: EMI0CF[4:2] = ‘101’, ‘110’, or ‘111’.

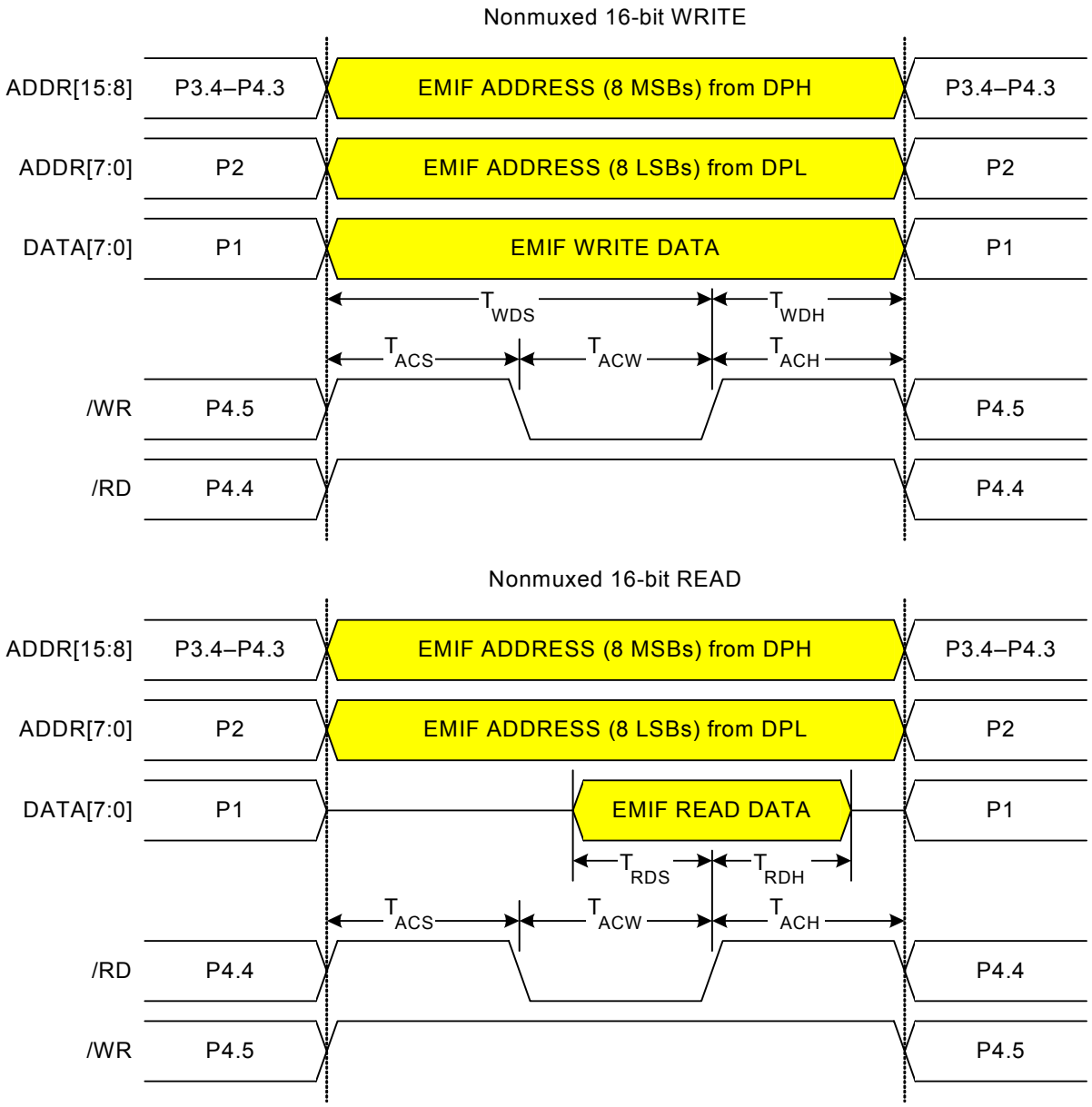


Figure 15.4. Non-multiplexed 16-bit MOVX Timing

15.6.1.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = '101' or '111'.

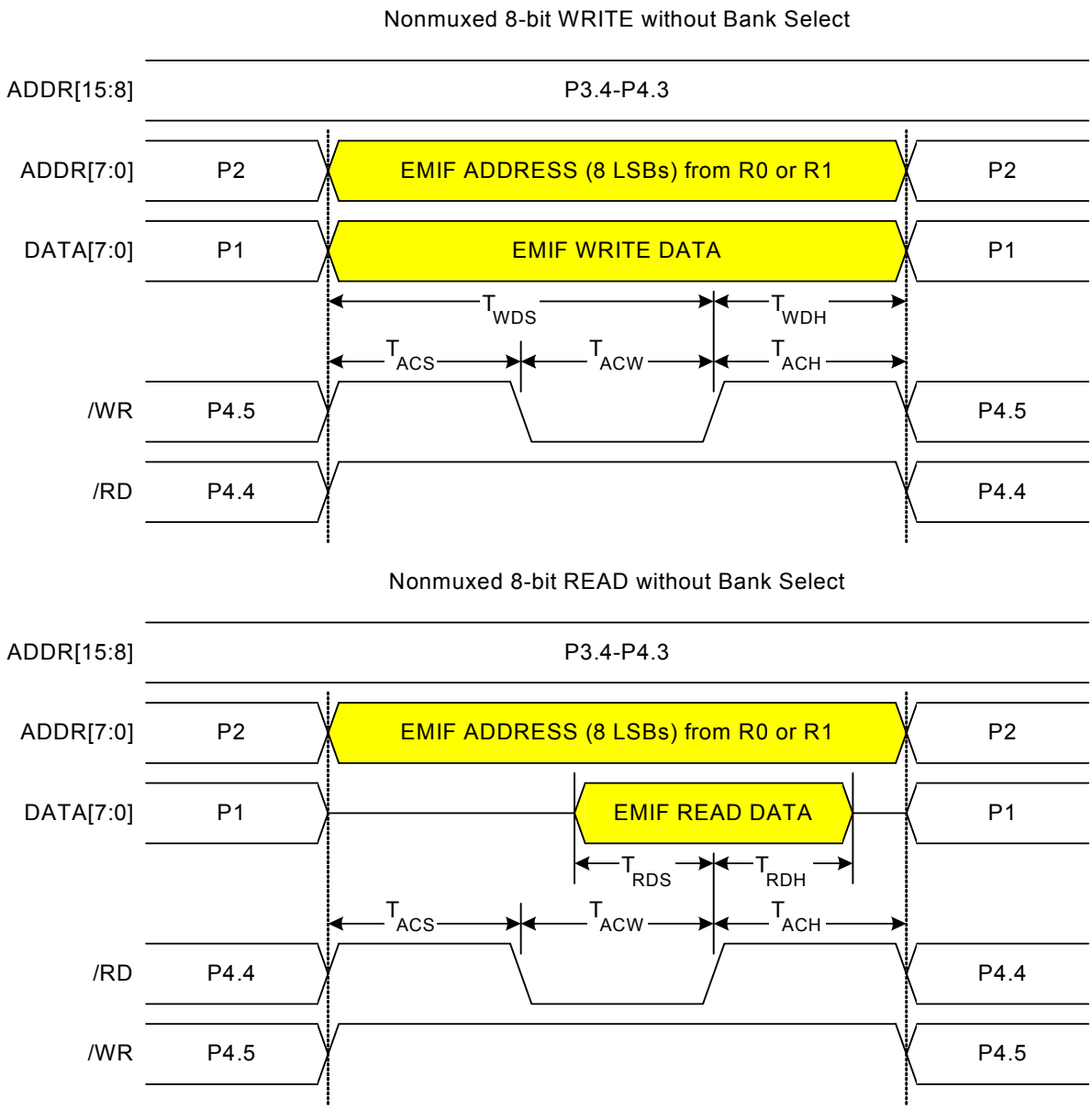


Figure 15.5. Non-multiplexed 8-bit MOVX without Bank Select Timing

15.6.1.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = '110'.

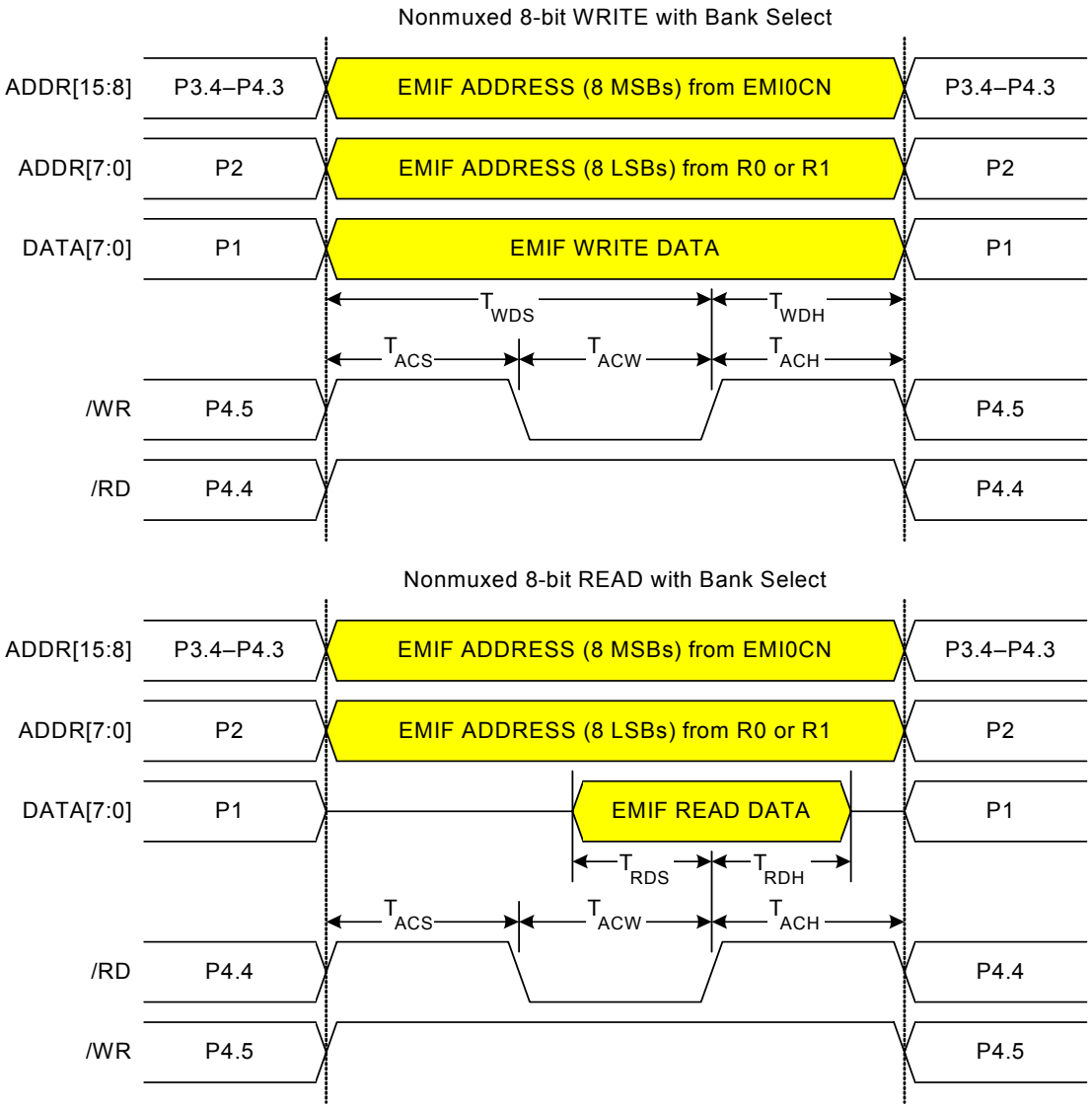


Figure 15.6. Non-multiplexed 8-bit MOVX with Bank Select Timing

15.6.2. Multiplexed Mode

15.6.2.1.16-bit MOVX: EMI0CF[4:2] = ‘001’, ‘010’, or ‘011’.

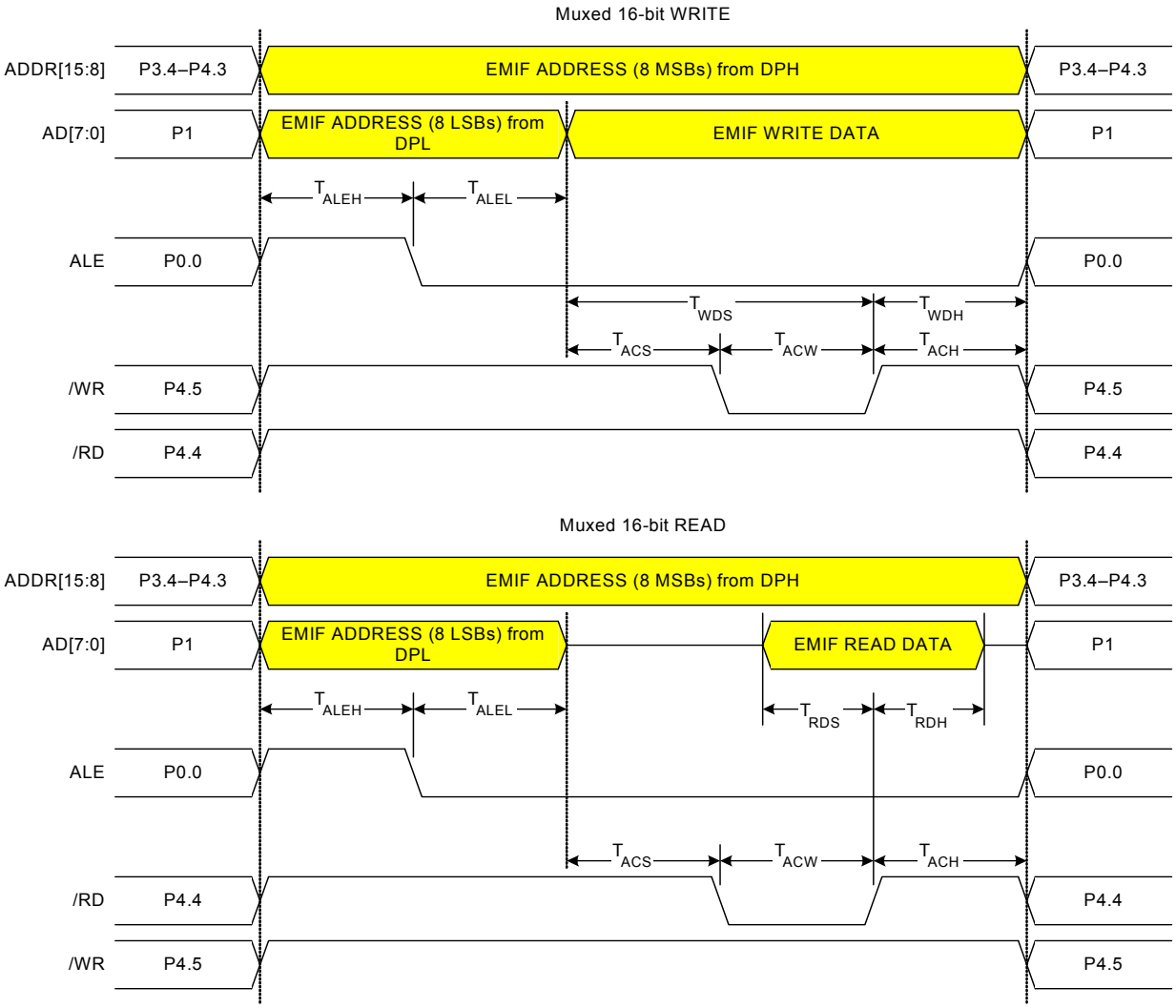


Figure 15.7. Multiplexed 16-bit MOVX Timing

15.6.2.2.8-bit MOVX without Bank Select: EMI0CF[4:2] = ‘001’ or ‘011’.

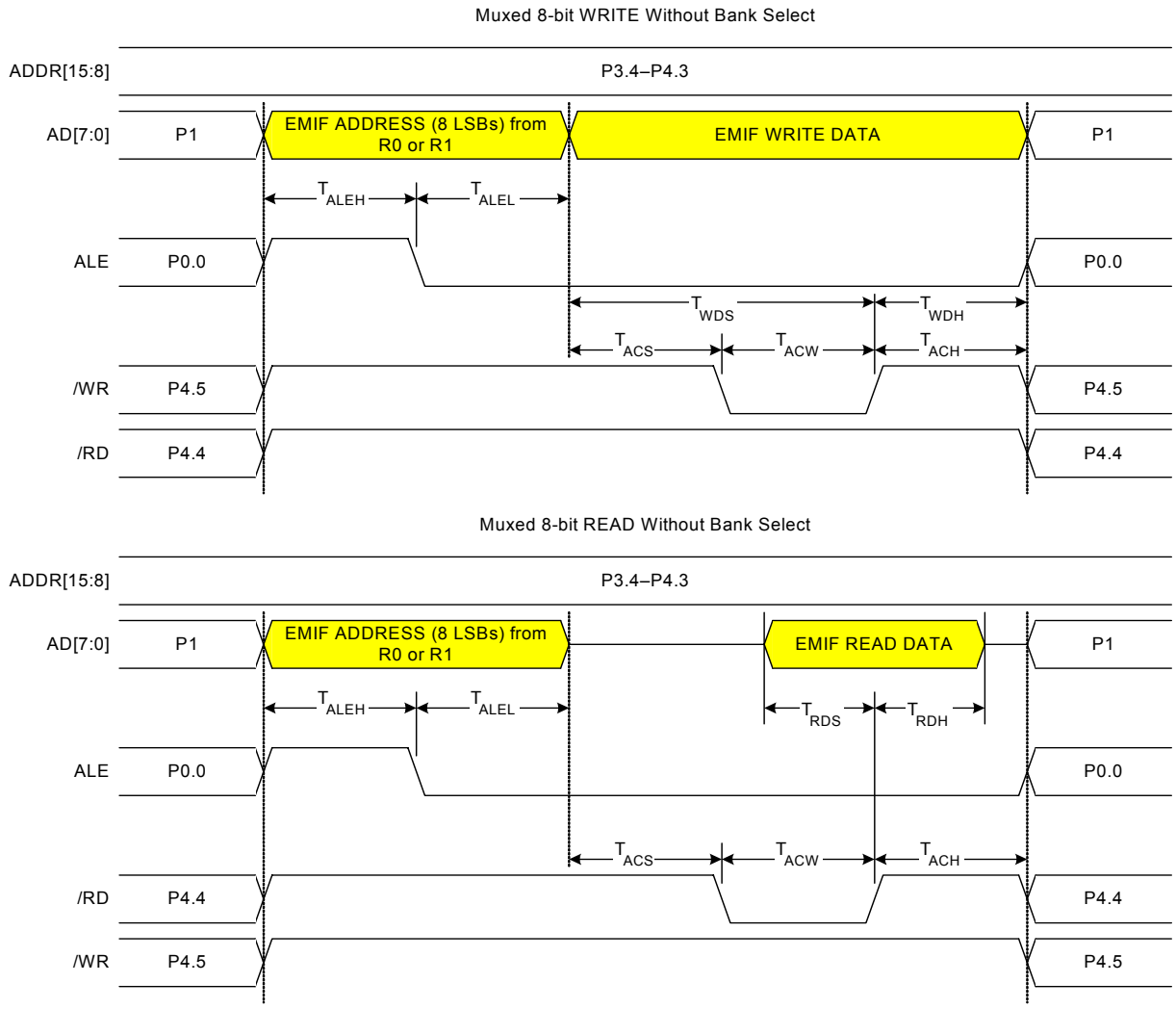


Figure 15.8. Multiplexed 8-bit MOVX without Bank Select Timing

15.6.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘010’.

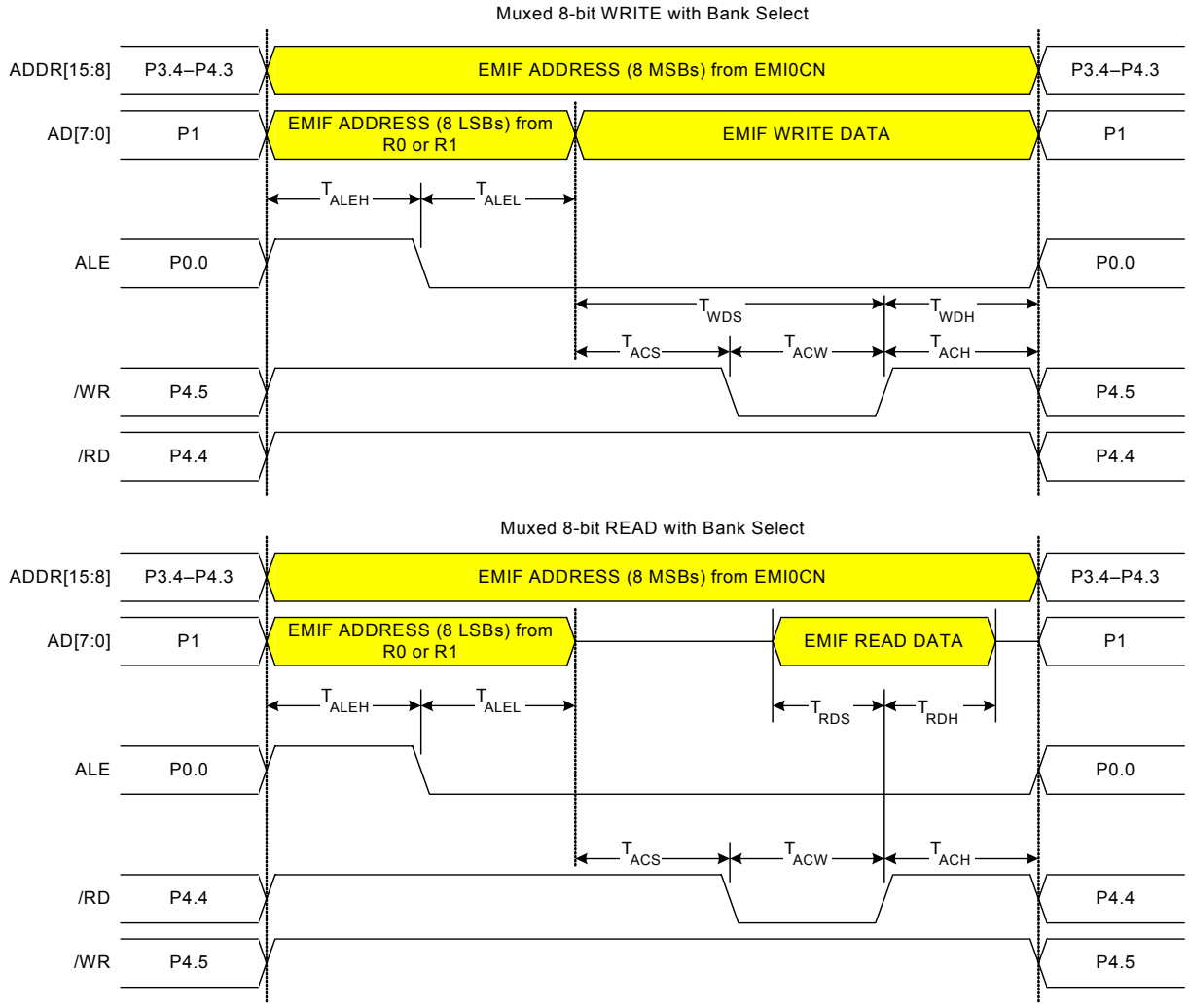


Figure 15.9. Multiplexed 8-bit MOVX with Bank Select Timing

Table 15.2. AC Parameters for External Memory Interface

Parameter	Description	Min*	Max*	Units
T_{ACS}	Address/Control Setup Time	0	3 x T_{SYSCLK}	ns
T_{ACW}	Address/Control Pulse Width	1 x T_{SYSCLK}	16 x T_{SYSCLK}	ns
T_{ACH}	Address/Control Hold Time	0	3 x T_{SYSCLK}	ns
T_{ALEH}	Address Latch Enable High Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{ALEL}	Address Latch Enable Low Time	1 x T_{SYSCLK}	4 x T_{SYSCLK}	ns
T_{WDS}	Write Data Setup Time	1 x T_{SYSCLK}	19 x T_{SYSCLK}	ns
T_{WDH}	Write Data Hold Time	0	3 x T_{SYSCLK}	ns
T_{RDS}	Read Data Setup Time	20		ns
T_{RDH}	Read Data Hold Time	0		ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

16. Oscillators

The C8051F36x devices include a programmable internal high-frequency oscillator, a programmable internal low-frequency oscillator, and an external oscillator drive circuit. The internal high-frequency oscillator can be enabled, disabled, and calibrated using the OSCICN and OSCICL registers, as shown in Figure 16.1. The internal low-frequency oscillator can be enabled/disabled and calibrated using the OSCLCN register, as shown in SFR Definition 16.3. Both internal oscillators offer a selectable post-scaling feature. The system clock can be sourced by the external oscillator circuit, either internal oscillator, or the on-chip phase-locked loop (PLL). The internal oscillator's electrical specifications are given in Table 16.1 on page 171 and Table 16.2 on page 172.

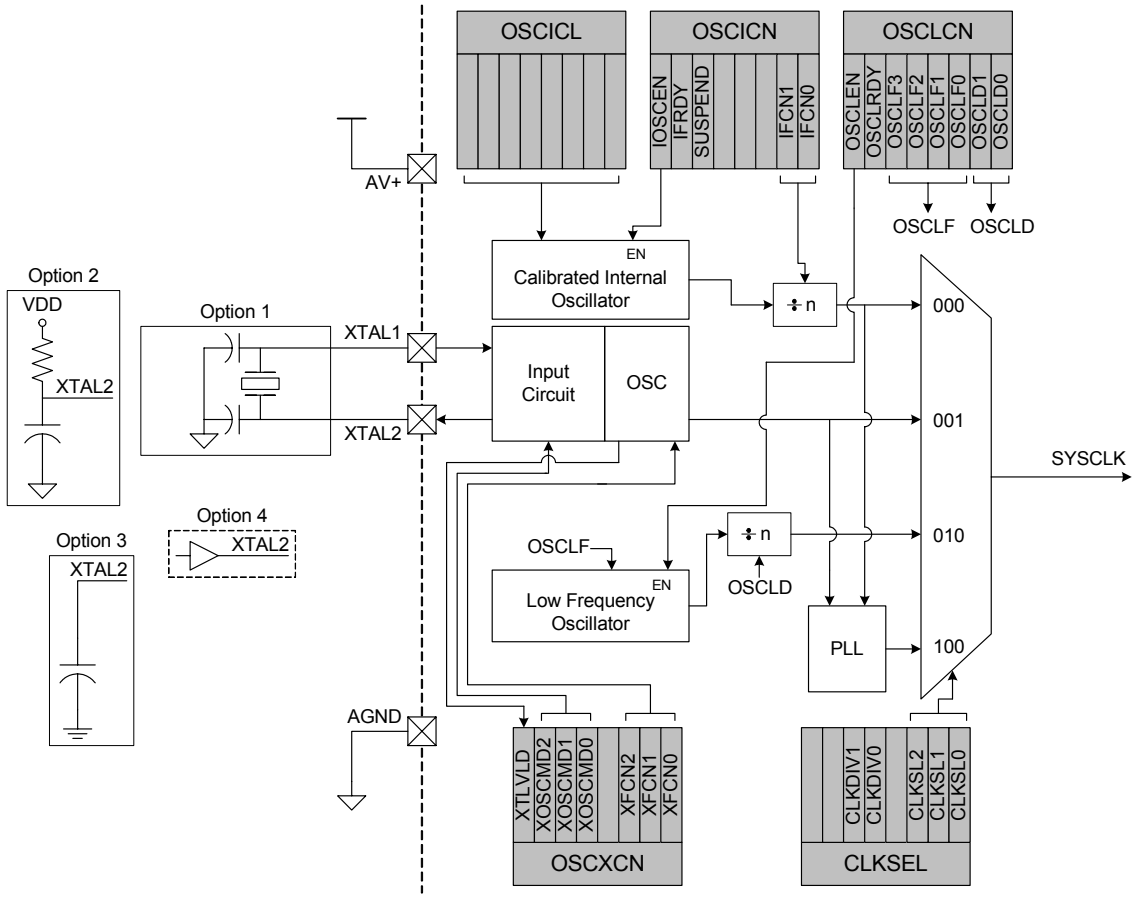


Figure 16.1. Oscillator Diagram

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

All devices include a calibrated 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 16.1. OSCICL is factory calibrated to obtain a 24.5 MHz frequency.

Electrical specifications for the precision internal oscillator are given in Table 16.1 on page 171 and Table 16.2 on page 172. Note that the system clock may be derived from the programmed internal oscillator 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.

16.1.1. Internal Oscillator Suspend Mode

When software writes a logic ‘1’ to SUSPEND (OSCICN.5), the internal oscillator is suspended. If the system clock is derived from the internal oscillator, the input clock to the peripheral or CIP-51 will be stopped until one of the following events occur:

- Port 0 Match Event.
- Port 1 Match Event.
- Port 2 Match Event.
- Comparator 0 enabled and output is logic ‘0’.
- Comparator 1 enabled and output is logic ‘0’.

When one of the internal oscillator awakening events occur, the internal oscillator, CIP-51, and affected peripherals resume normal operation, regardless of whether the event also causes an interrupt. The CPU resumes execution at the instruction following the write to SUSPEND.

Note: Before entering SUSPEND mode, SYSCLK should be switched to run off of the internal oscillator and not the PLL. When the CPU wakes due to the awakening event, the PLL must be reinitialized before switching back to it as the SYSCLK source.

SFR Definition 16.1. OSCICL: Internal Oscillator Calibration.

SFR Page: F
SFR Address: 0xBF

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: OSCICL: Internal Oscillator Calibration Register.
This register calibrates the internal oscillator period. The reset value for OSCICL defines the internal oscillator base frequency. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

SFR Definition 16.2. OSCICN: Internal Oscillator Control

SFR Page: F
SFR Address: 0xB7

R/W	R	R/W	R	R/W	R/W	R/W	R/W	Reset Value
IOSCEN	IFRDY	SUSPEND	Reserved	Reserved	Reserved	IFCN1	IFCN0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: IOSCEN: Internal Oscillator Enable Bit.
0: Internal Oscillator Disabled.
1: Internal Oscillator Enabled.
Bit 6: IFRDY: Internal Oscillator Frequency Ready Flag.
0: Internal Oscillator not running at programmed frequency.
1: Internal Oscillator running at programmed frequency.
Bits 5: SUSPEND: Internal Oscillator Suspend Enable Bit.
Setting this bit to logic ‘1’ places the internal oscillator in SUSPEND mode. The internal oscillator resumes operation when one of the SUSPEND mode awakening events occur.
Bits 4–2: RESERVED. Read = 000b. Must Write 000b.
Bits 1–0: IFCN1-0: Internal Oscillator Frequency Control Bits.
00: Internal Oscillator is divided by 8. (default)
01: Internal Oscillator is divided by 4.
10: Internal Oscillator is divided by 2.
11: Internal Oscillator is divided by 1.

Table 16.1. Internal High Frequency Oscillator Electrical Characteristics

–40°C to +85°C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Calibrated Internal Oscillator Frequency		24	24.5	25	MHz
Internal Oscillator Supply Current (from V_{DD})	OSCICN.7 = 1	—	450	600	µA
Power Supply Sensitivity	Constant Temperature	—	0.12	—	%/V
Temperature Sensitivity	Constant Supply	—	60	—	ppm/°C
External Clock Frequency		0	—	30	MHz
T_{XCH} (External Clock High Time)		15	—	—	ns
T_{XCL} (External Clock Low Time)		15	—	—	ns

16.2. Programmable Internal Low-Frequency (L-F) Oscillator

All C8051F36x devices include a programmable low-frequency internal oscillator, which is calibrated to a nominal frequency of 80 kHz. The low-frequency oscillator circuit includes a divider that can be changed to divide the clock by 1, 2, 4, or 8, using the OSCLD bits in the OSCLCN register (see SFR Definition 16.3). Additionally, the OSCLF bits (OSCLCN5:2) can be used to adjust the oscillator’s output frequency.

16.2.1. Calibrating the Internal L-F Oscillator

Timers 2 and 3 include capture functions that can be used to capture the oscillator frequency, when running from a known time base. When either Timer 2 or Timer 3 is configured for L-F Oscillator Capture Mode, a falling edge (Timer 2) or rising edge (Timer 3) of the low-frequency oscillator's output will cause a capture event on the corresponding timer. As a capture event occurs, the current timer value (TMRnH:TMRnL) is copied into the timer reload registers (TMRnRLH:TMRnRLL). By recording the difference between two successive timer capture values, the low-frequency oscillator's period can be calculated. The OSCLF bits can then be adjusted to produce the desired oscillator frequency.

SFR Definition 16.3. OSCLCN: Internal L-F Oscillator Control

SFR Page: F
SFR Address: 0xAD

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
OSCLEN	OSCLRDY	OSCLF3	OSCLF2	OSCLF1	OSCLF0	OSCLD1	OSCLD0	00vvvv00
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: OSCLEN: Internal L-F Oscillator Enable.
0: Internal L-F Oscillator Disabled.
1: Internal L-F Oscillator Enabled.

Bit 6: OSCLRDY: Internal L-F Oscillator Ready.
0: Internal L-F Oscillator frequency not stabilized.
1: Internal L-F Oscillator frequency stabilized.

Bits 5–2: OSCLF[3:0]: Internal L-F Oscillator Frequency Control bits.
Fine-tune control bits for the Internal L-F oscillator frequency. When set to 0000b, the L-F oscillator operates at its fastest setting. When set to 1111b, the L-F oscillator operates at its slowest setting.

Bits 1–0: OSCLD[1:0]: Internal L-F Oscillator Divider Select.
00: Divide by 8 selected.
01: Divide by 4 selected.
10: Divide by 2 selected.
11: Divide by 1 selected.

Table 16.2. Internal Low Frequency Oscillator Electrical Characteristics
–40°C to +85°C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	OSCLD = 11b	72	80	88	kHz
Oscillator Supply Current (from V_{DD})	25 °C, V_{DD} = 3.0 V, OSCLCN.7 = 1	—	5.5	10	µA
Power Supply Sensitivity	Constant Temperature	—	2.4	—	%/V
Temperature Sensitivity	Constant Supply	—	30	—	ppm/°C

16.3. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 16.1. A 10 MΩ resistor also must be wired across the XTAL1 and XTAL2 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 16.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 16.5).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.5 and P0.6 (C8051F360/3) or P0.2 and P0.3 (C8051F361/2/4/5/6/7/8/9) are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.6 (C8051F360/3) or P0.3 (C8051F361/2/4/5/6/7/8/9) is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “17.1. Priority Crossbar Decoder” on page 185 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “17.2. Port I/O Initialization” on page 187 for details on Port input mode selection.

16.4. System Clock Selection

The internal oscillator requires little start-up time, and may be enabled and selected as the system clock in the same write to OSCICN. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use as the system clock. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to ‘1’ by hardware when the external oscillator is settled. To avoid reading a false XTLVLD, in crystal mode software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD. RC and C modes typically require no startup time. The PLL also requires time to lock onto the desired frequency, and the PLL Lock Flag (PLLLCK in register PLL0CN) is set to ‘1’ by hardware once the PLL is locked on the correct frequency.

The CLKSL1-0 bits in register CLKSEL select which oscillator source generates the system clock. CLKSL1-0 must be set to ‘01’ for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals, such as the timers and PCA, when the internal oscillator or the PLL is selected as the system clock. The system clock may be switched on-the-fly between the internal and external oscillators or the PLL, so long as the selected oscillator source is enabled and settled.

SFR Definition 16.4. CLKSEL: System Clock Selection

SFR Page: F
SFR Address: 0x8F

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Reserved	Reserved	CLKDIV1	CLKDIV0	Reserved	CLKSL2	CLKSL1	CLKSL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: RESERVED. Read = 00b. Must Write 00b.
Bits 5–4: CLKDIV1-0: Output SYSCLK Divide Factor.
These bits can be used to pre-divide SYSCLK before it is output to a port pin through the crossbar.
00: Output will be SYSCLK.
01: Output will be SYSCLK/2.
10: Output will be SYSCLK/4.
11: Output will be SYSCLK/8.
See Section “17. Port Input/Output” on page 183 for more details about routing this output to a port pin.
Bit 3: RESERVED. Read = 0b. Must Write 0b.
Bits 2–0: CLKSL2–0: System Clock Source Select Bits.
000: SYSCLK derived from the high-frequency Internal Oscillator, and scaled as per the IFCN bits in OSCICN.
001: SYSCLK derived from the External Oscillator circuit.
010: SYSCLK derived from the low-frequency Internal Oscillator, and scaled as per the OSCLD bits in OSCLCN.
011: RESERVED.
100: SYSCLK derived from the PLL.
101-11x: RESERVED.

SFR Definition 16.5. OSCXCN: External Oscillator Control

SFR Page: F
SFR Address: 0xB6

R	R/W	R/W	R/W	R	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	Reserved	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: XTLVLD: Crystal Oscillator Valid Flag.
(**Valid only when XOSCMD = 11x.**)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits 6–4: XOSCMD2–0: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode.
011: External CMOS Clock Mode with divide by 2 stage.
100: RC Oscillator Mode.
101: Capacitor Oscillator Mode.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit 3: RESERVED. Read = 0b. Write = don't care.

Bits 2–0: XFCN2–0: External Oscillator Frequency Control Bits.
000-111: see table below:

XFCN	Crystal (XOSCMD = 11x)	RC (XOSCMD = 100)	C (XOSCMD = 101)
000	$f \leq 32$ kHz	$f \leq 25$ kHz	K Factor = 0.87
001	32 kHz $< f \leq$ 84 kHz	25 kHz $< f \leq$ 50 kHz	K Factor = 2.6
010	84 kHz $< f \leq$ 225 kHz	50 kHz $< f \leq$ 100 kHz	K Factor = 7.7
011	225 kHz $< f \leq$ 590 kHz	100 kHz $< f \leq$ 200 kHz	K Factor = 22
100	590 kHz $< f \leq$ 1.5 MHz	200 kHz $< f \leq$ 400 kHz	K Factor = 65
101	1.5 MHz $< f \leq$ 4 MHz	400 kHz $< f \leq$ 800 kHz	K Factor = 180
110	4 MHz $< f \leq$ 10 MHz	800 kHz $< f \leq$ 1.6 MHz	K Factor = 664
111	10 MHz $< f \leq$ 30 MHz	1.6 MHz $< f \leq$ 3.2 MHz	K Factor = 1590

CRYSTAL MODE (Circuit from Figure 16.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal frequency.

RC MODE (Circuit from Figure 16.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
$f = 1.23(10^3)/(R * C)$, where
f = frequency of oscillation in MHz
C = capacitor value in pF
R = Pullup resistor value in kΩ

C MODE (Circuit from Figure 16.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
$f = KF/(C * V_{DD})$, where
f = frequency of oscillation in MHz
C = capacitor value on XTAL1, XTAL2 pins in pF
V_{DD} = Power Supply on MCU in Volts

16.5. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 16.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 16.5 (OSCXCN register). For example, an 11.0592 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is enabled, the oscillator amplitude detection circuit requires a settle time to achieve proper bias. Waiting at least 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Force the XTAL1 and XTAL2 pins low by writing 0's to the port latch.
Step 2. Configure XTAL1 and XTAL2 as analog inputs.
Step 3. Enable the external oscillator.
Step 4. Wait at least 1 ms.
Step 5. Poll for XTLVLD => '1'.
Step 6. Switch the system clock to the external oscillator.

Note: Tuning-fork crystals may require additional settling time before XTLVLD returns a valid result.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 kHz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 16.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 16.2.

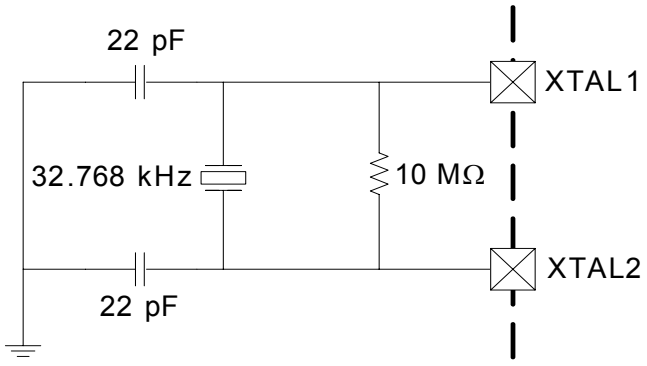


Figure 16.2. 32.768 kHz External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

16.6. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 16.1, Option 2. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation. If the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

f = 1.23(10^3)/RC = 1.23 (10^3)/[246 x 50] = 0.1 MHz = 100 kHz

Referring to the table in SFR Definition 16.5, the required XFCN setting is 010b. Programming XFCN to a higher setting in RC mode will improve frequency accuracy at a slightly increased external oscillator supply current.

16.7. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 16.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the capacitor to be used and find the frequency of oscillation from the equations below. Assume V_{DD} = 3.0 V and f = 75 kHz:

f = KF / (C x V_{DD})
0.075 MHz = KF / (C x 3.0)

Since the frequency of roughly 75 kHz is desired, select the K Factor from the table in SFR Definition 16.5 as KF = 7.7:

0.075 MHz = 7.7 / (C x 3.0)

C x 3.0 = 7.7 / 0.075 MHz

C = 102.6 / 3.0 pF = 34.2 pF

Therefore, the XFCN value to use in this example is 010b.

16.8. Phase-Locked Loop (PLL)

A Phase-Locked-Loop (PLL) is included, which is used to multiply the internal oscillator or an external clock source to achieve higher CPU operating frequencies. The PLL circuitry is designed to produce an output frequency between 25 MHz and 100 MHz, from a divided reference frequency between 5 MHz and 30 MHz. A block diagram of the PLL is shown in Figure 16.3.

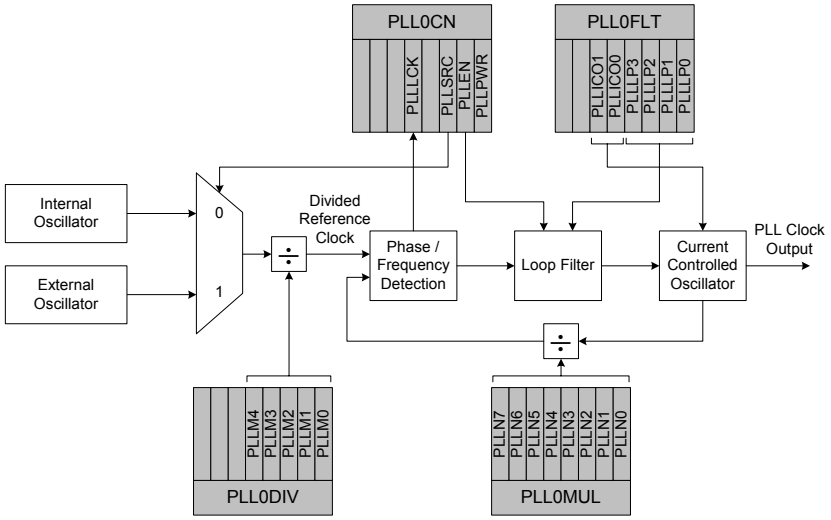


Figure 16.3. PLL Block Diagram

16.8.1. PLL Input Clock and Pre-divider

The PLL circuitry can derive its reference clock from either the internal oscillator or an external clock source. The PLLSRC bit (PLL0CN.2) controls which clock source is used for the reference clock (see SFR Definition 16.6). If PLLSRC is set to '0', the internal oscillator source is used. Note that the internal oscillator divide factor (as specified by bits IFCN1-0 in register OSCICN) will also apply to this clock. When PLLSRC is set to '1', an external oscillator source will be used. The external oscillator should be active and settled before it is selected as a reference clock for the PLL circuit. The reference clock is divided down prior to the PLL circuit, according to the contents of the PLLM4-0 bits in the PLL Pre-divider Register (PLL0DIV), shown in SFR Definition 16.7.

16.8.2. PLL Multiplication and Output Clock

The PLL circuitry will multiply the divided reference clock by the multiplication factor stored in the PLL0MUL register shown in SFR Definition 16.8. To accomplish this, it uses a feedback loop consisting of a phase/frequency detector, a loop filter, and a current-controlled oscillator (ICO). It is important to configure the loop filter and the ICO for the correct frequency ranges. The PLLLP3–0 bits (PLL0FLT.3–0) should be set according to the divided reference clock frequency. Likewise, the PLLICO1–0 bits (PLL0FLT.5–4) should be set according to the desired output frequency range. SFR Definition 16.9 describes the proper settings to use for the PLLLP3–0 and PLLICO1–0 bits. When the PLL is locked and stable at the desired frequency, the PLLLCK bit (PLL0CN.5) will be set to a '1'. The resulting PLL frequency will be set according to the equation:

$$\text{PLL Frequency} = \text{Reference Frequency} \times \frac{\text{PLLN}}{\text{PLLM}}$$

Where "Reference Frequency" is the selected source clock frequency, PLLN is the PLL Multiplier, and PLLM is the PLL Pre-divider.

16.8.3. Powering on and Initializing the PLL

To set up and use the PLL as the system clock after power-up of the device, the following procedure should be implemented:

Step 1. Ensure that the reference clock to be used (internal or external) is running and stable.
Step 2. Set the PLLSRC bit (PLL0CN.2) to select the desired clock source for the PLL.
Step 3. Program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see Section “13. Flash Memory” on page 135).
Step 4. Enable power to the PLL by setting PLLPWR (PLL0CN.0) to ‘1’.
Step 5. Program the PLL0DIV register to produce the divided reference frequency to the PLL.
Step 6. Program the PLLLP3–0 bits (PLL0FLT.3–0) to the appropriate range for the divided reference frequency.
Step 7. Program the PLLICO1–0 bits (PLL0FLT.5–4) to the appropriate range for the PLL output frequency.
Step 8. Program the PLL0MUL register to the desired clock multiplication factor.
Step 9. Wait at least 5 µs, to provide a fast frequency lock.
Step 10. Enable the PLL by setting PLLEN (PLL0CN.1) to ‘1’.
Step 11. Poll PLLLCK (PLL0CN.4) until it changes from ‘0’ to ‘1’.
Step 12. Switch the System Clock source to the PLL using the CLKSEL register.

If the PLL characteristics need to be changed when the PLL is already running, the following procedure should be implemented:

Step 1. The system clock should first be switched to either the internal oscillator or an external clock source that is running and stable, using the CLKSEL register.
Step 2. Ensure that the reference clock to be used for the new PLL setting (internal or external) is running and stable.
Step 3. Set the PLLSRC bit (PLL0CN.2) to select the new clock source for the PLL.
Step 4. If moving to a faster frequency, program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see Section “13. Flash Memory” on page 135).
Step 5. Disable the PLL by setting PLLEN (PLL0CN.1) to ‘0’.
Step 6. Program the PLL0DIV register to produce the divided reference frequency to the PLL.
Step 7. Program the PLLLP3–0 bits (PLL0FLT.3–0) to the appropriate range for the divided reference frequency.
Step 8. Program the PLLICO1-0 bits (PLL0FLT.5–4) to the appropriate range for the PLL output frequency.
Step 9. Program the PLL0MUL register to the desired clock multiplication factor.
Step 10. Enable the PLL by setting PLLEN (PLL0CN.1) to ‘1’.
Step 11. Poll PLLLCK (PLL0CN.4) until it changes from ‘0’ to ‘1’.
Step 12. Switch the System Clock source to the PLL using the CLKSEL register.
Step 13. If moving to a slower frequency, program the Flash read timing bits, FLRT (FLSCL.5–4) to the appropriate value for the new clock rate (see Section “13. Flash Memory” on page 135). **Important Note: Cache reads, cache writes, and the prefetch engine should be disabled whenever the FLRT bits are changed to a lower setting.**

To shut down the PLL, the system clock should be switched to the internal oscillator or a stable external clock source, using the CLKSEL register. Next, disable the PLL by setting PLLEN (PLL0CN.1) to '0'. Finally, the PLL can be powered off, by setting PLLPWR (PLL0CN.0) to '0'. Note that the PLLEN and PLLPWR bits can be cleared at the same time.

SFR Definition 16.6. PLL0CN: PLL Control

SFR Page: F
SFR Address: 0xB3

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
–	–	–	PLLLCK	Reserved	PLLSRC	PLLEN	PLLPWR	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–5: UNUSED. Read = 000b. Write = don't care.
Bit 4: PLLLCK: PLL Lock Flag.
0: PLL Frequency is not locked.
1: PLL Frequency is locked.
Bit 3: RESERVED. Read = 0b. Must Write 0b.
Bit 2: PLLSRC: PLL Reference Clock Source Select Bit.
0: PLL Reference Clock Source is Internal Oscillator.
1: PLL Reference Clock Source is External Oscillator.
Bit 1: PLLEN: PLL Enable Bit.
0: PLL is held in reset.
1: PLL is enabled. PLLPWR must be '1'.
Bit 0: PLLPWR: PLL Power Enable.
0: PLL bias generator is de-activated. No static power is consumed.
1: PLL bias generator is active. Must be set for PLL to operate.

SFR Definition 16.7. PLL0DIV: PLL Pre-divider

SFR Page: F
SFR Address: 0xA9

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	–	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–5: UNUSED. Read = 000b. Write = don't care.
Bits 4–0: PLLM4–0: PLL Reference Clock Pre-divider.
These bits select the pre-divide value of the PLL reference clock. When set to any non-zero value, the reference clock will be divided by the value in PLLM4–0. When set to '00000b', the reference clock will be divided by 32.

SFR Definition 16.8. PLL0MUL: PLL Clock Scaler

SFR Page: F
SFR Address: 0xB1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PLLN7	PLLN6	PLLN5	PLLN4	PLLN3	PLLN2	PLLN1	PLLN0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: PLLN7–0: PLL Multiplier.
These bits select the multiplication factor of the divided PLL reference clock. When set to any non-zero value, the multiplication factor will be equal to the value in PLLN7-0. When set to ‘00000000b’, the multiplication factor will be equal to 256.

SFR Definition 16.9. PLL0FLT: PLL Filter

SFR Page: F
SFR Address: 0xB2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	–	PLLICO1	PLLICO0	PLLLP3	PLLLP2	PLLLP1	PLLLP0	00110001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–6: UNUSED. Read = 00b. Write = don’t care.
Bits 5–4: PLLICO1-0: PLL Current-Controlled Oscillator Control Bits.
Selection is based on the desired output frequency, according to the following table:

PLL Output Clock	PLLICO1-0
65–100 MHz	00
45–80 MHz	01
30–60 MHz	10
25–50 MHz	11

Bits 3–0: PLLLP3-0: PLL Loop Filter Control Bits.
Selection is based on the divided PLL reference clock, according to the following table:

Divided PLL Reference Clock	PLLLP3-0
19–30 MHz	0001
12.2–19.5 MHz	0011
7.8–12.5 MHz	0111
5–8 MHz	1111

All other states of PLLLP3–0 are RESERVED.

Table 16.3. PLL Frequency Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Input Frequency (Divided Reference Frequency)		5		30	MHz
PLL Output Frequency		25		100*	MHz
***Note:** The maximum operating frequency of the C8051F366/7/8/9 is 50 MHz.					

Table 16.4. PLL Lock Timing Characteristics

–40 to +85 °C unless otherwise specified

Input Frequency	Multiplier (Pll0mul)	Pll0flt Setting	Output Frequency	Min	Typ	Max	Units
5 MHz	20	0x0F	100 MHz		202		µs
	13	0x0F	65 MHz		115		µs
	16	0x1F	80 MHz		241		µs
	9	0x1F	45 MHz		116		µs
	12	0x2F	60 MHz		258		µs
	6	0x2F	30 MHz		112		µs
	10	0x3F	50 MHz		263		µs
	5	0x3F	25 MHz		113		µs
25 MHz	4	0x01	100 MHz		42		µs
	2	0x01	50 MHz		33		µs
	3	0x11	75 MHz		48		µs
	2	0x11	50 MHz		17		µs
	2	0x21	50 MHz		42		µs
	1	0x21	25 MHz		33		µs
	2	0x31	50 MHz		60		µs
	1	0x31	25 MHz		25		µs

17. Port Input/Output

Digital and analog resources are available through up to 39 I/O pins. On the largest devices (C8051F360/3), port pins are organized as four byte-wide Ports and one 7-bit-wide Port. On the other devices (C8051F361/2/4/5/6/7/8/9), port pins are three byte-wide Ports and one partial port. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input/output; Port pins P0.0–P3.7 can be assigned to one of the internal digital resources as shown in Figure 17.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the peripheral priority order of the Priority Decoder (Figure 17.3 and Figure 17.4). The registers XBR0 and XBR1, defined in SFR Definition 17.1 and SFR Definition 17.2, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 17.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1,2,3,4). Complete Electrical Specifications for Port I/O are given in Table 17.1 on page 201.

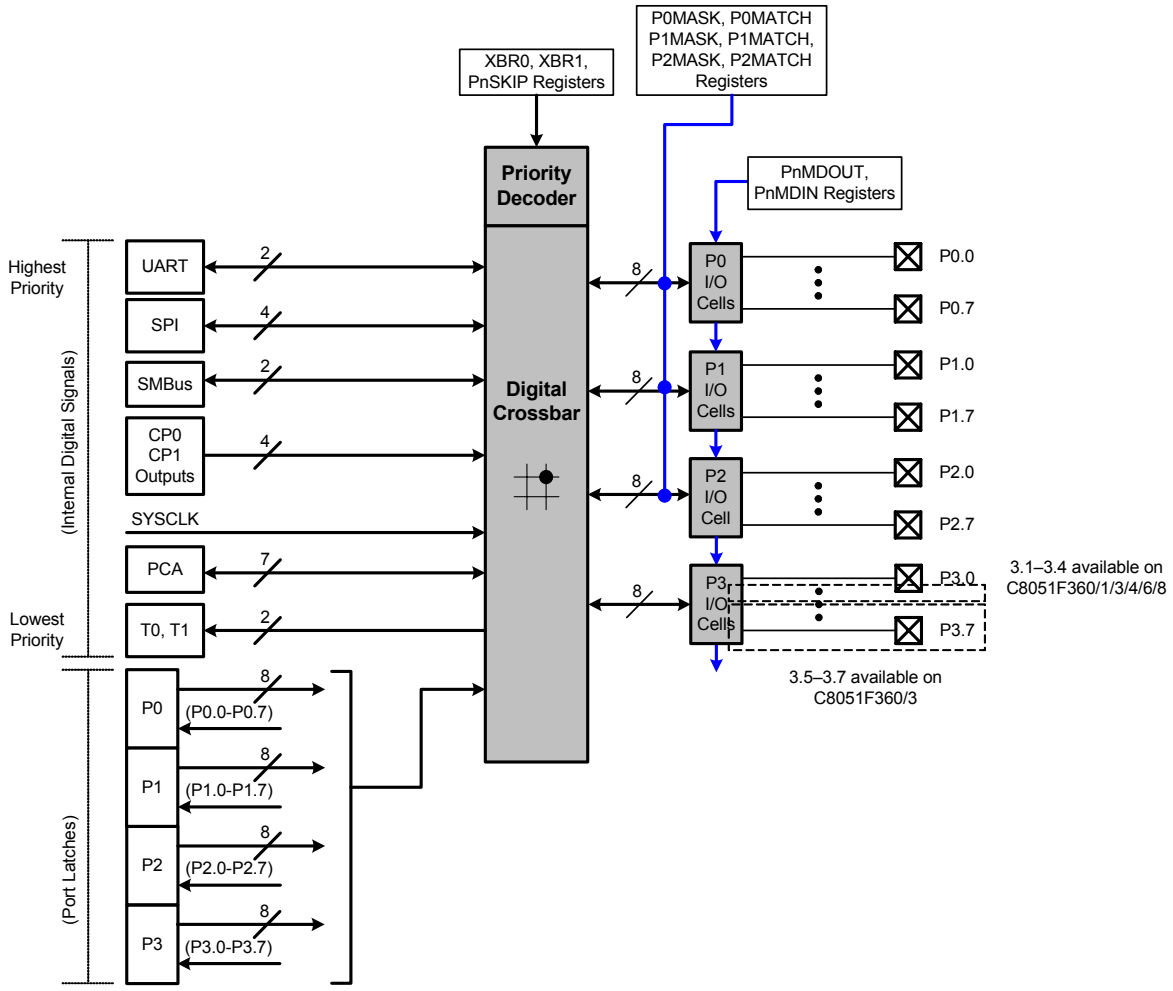


Figure 17.1. Port I/O Functional Block Diagram (Port 0 through Port 3)

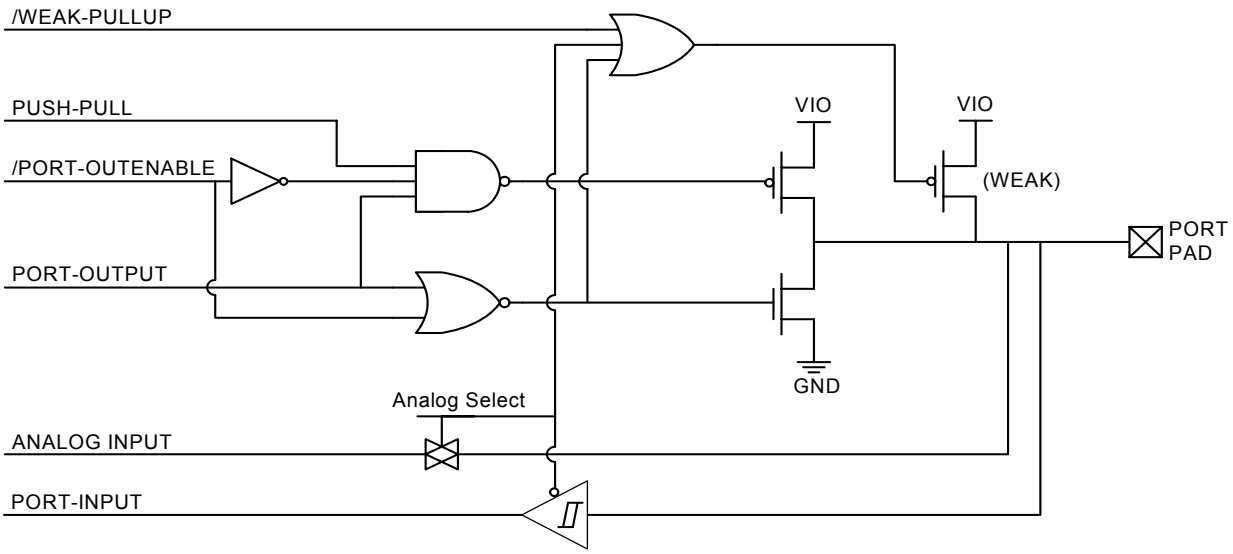


Figure 17.2. Port I/O Cell Block Diagram

17.1. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 17.3) 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 will be assigned to specific port pins (P0.1 and P0.2 in the C8051F360/3 devices, P0.4 and P0.5 in the C8051F361/2/4/5/6/7/8/9 devices). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. 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.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to the port pins associated with the external oscillator, V_{REF}, external CNVSTR signal, IDA0, and any selected ADC or comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 17.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP, P2SKIP, P3SKIP = 0x00); Figure 17.4 shows the Crossbar Decoder priority with the P1.0 and P1.1 pins skipped (P1SKIP = 0x03).

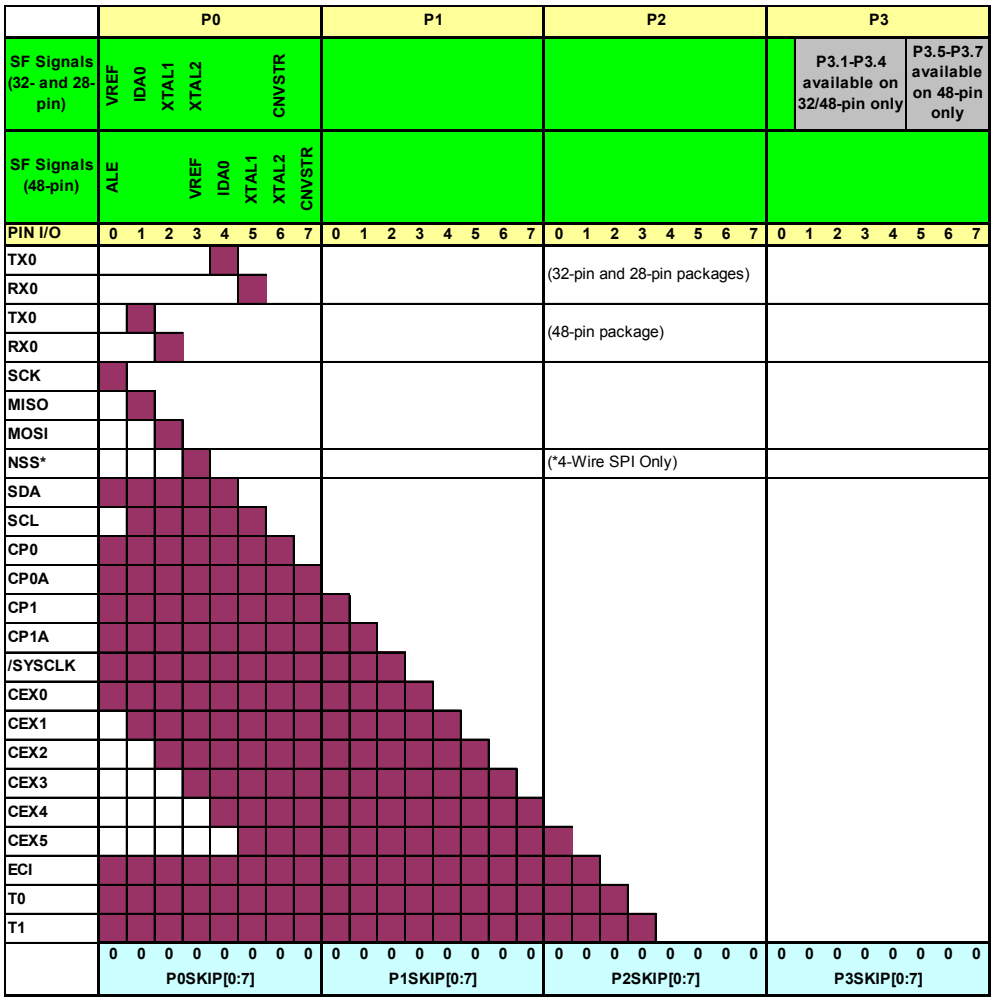


Figure 17.3. Crossbar Priority Decoder with No Pins Skipped

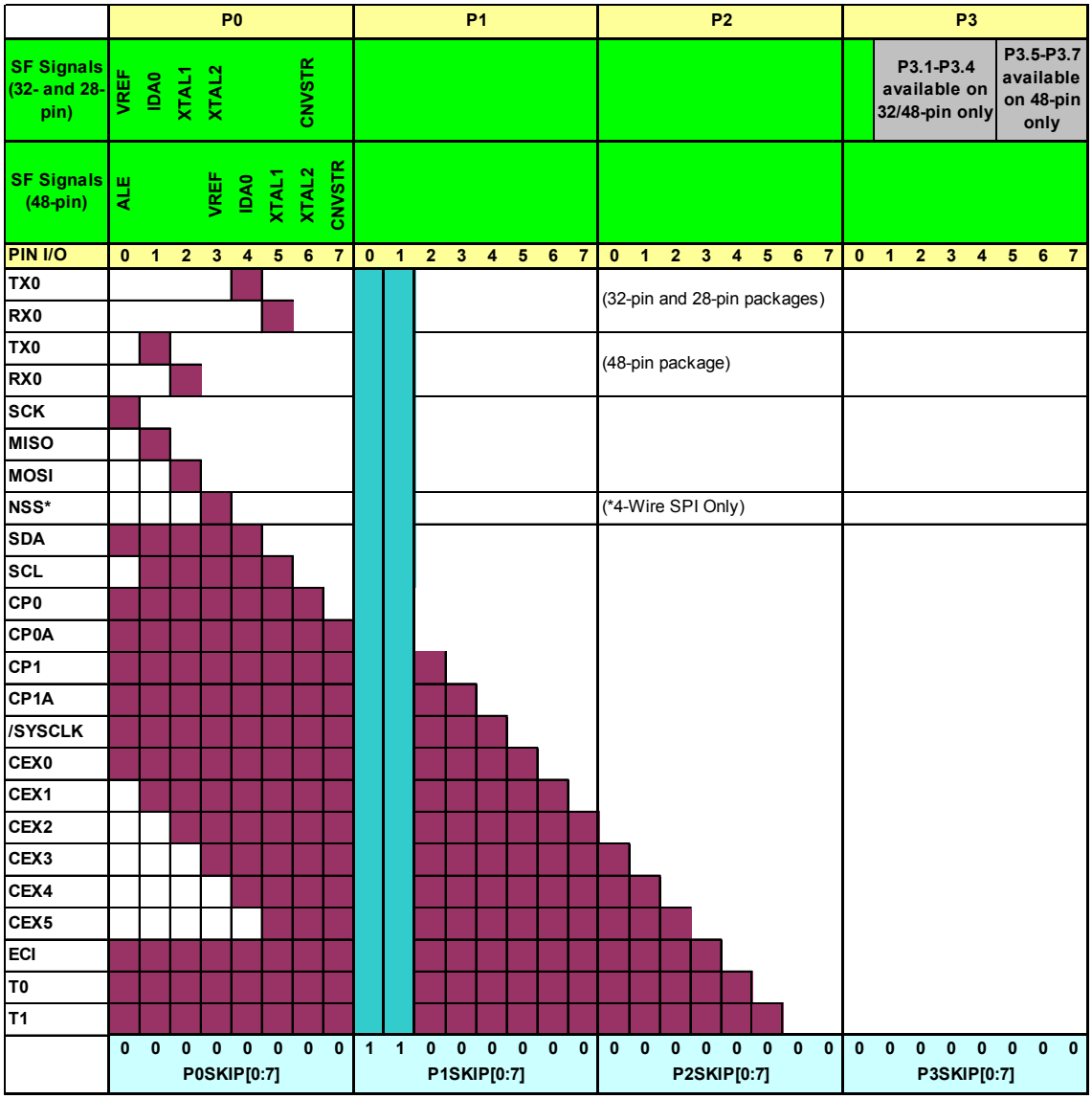


Figure 17.4. Crossbar Priority Decoder with Port Pins Skipped

Registers XBR0 and XBR1 are used to assign the digital I/O resources to the physical I/O Port pins. Note that 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 P0.1 (C8051F360/3) or P0.4 (C8051F361/2/4/5/6/7/8/9); UART RX0 is always assigned to P0.2 (C8051F360/3) or P0.5 (C8051F361/2/4/5/6/7/8/9). Standard Port I/Os appear contiguously starting at P0.0 after prioritized functions and skipped pins are assigned.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1-NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

17.2. Port I/O Initialization

Port I/O initialization consists of the following steps:

Step 1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
Step 2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
Step 3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
Step 4. Assign Port pins to desired peripherals using the XBRn registers.
Step 5. Enable the Crossbar (XBARE = ‘1’).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a ‘1’ indicates a digital input, and a ‘0’ indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 17.4 for the PnMDIN register details.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. The only exception to this is the SMBus (SDA, SCL) pins, which are configured as open-drain regardless of the PnMDOUT settings. When the WEAKPUD bit in XBR1 is ‘0’, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a ‘0’ and for pins configured for analog input mode to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to ‘1’ enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table; as an alternative, the Configuration Wizard utility of the Silicon Labs IDE software will determine the Port I/O pin-assignments based on the XBRn Register settings.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. **Port output drivers are disabled while the Crossbar is disabled.**

SFR Definition 17.1. XBR0: Port I/O Crossbar Register 0

SFR Page: F
SFR Address: 0xE1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP1AE	CP1E	CP0AE	CP0E	SYSCKE	SMB0E	SPI0E	URT0E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CP1AE: Comparator1 Asynchronous Output Enable
0: Asynchronous CP1 unavailable at Port pin.
1: Asynchronous CP1 routed to Port pin.

Bit 6: CP1E: Comparator1 Output Enable
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

Bit 5: CP0AE: Comparator0 Asynchronous Output Enable
0: Asynchronous CP0 unavailable at Port pin.
1: Asynchronous CP0 routed to Port pin.

Bit 4: CP0E: Comparator0 Output Enable
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.

Bit 3: SYSCKE: /SYSCLK Output Enable
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK (divided by 1, 2, 4, or 8) routed to Port pin. The divide factor is determined by the CLKDIV1–0 bits in register CLKSEL (See Section Section “16. Oscillators” on page 169).

Bit 2: SMB0E: SMBus I/O Enable
0: SMBus I/O unavailable at Port pins.
1: SMBus I/O routed to Port pins.

Bit 1: SPI0E: SPI I/O Enable
0: SPI I/O unavailable at Port pins.
1: SPI I/O routed to Port pins. Note that the SPI can be assigned either 3 or 4 GPIO pins.

Bit 0: URT0E: UART I/O Output Enable
0: UART I/O unavailable at Port pin.
1: UART TX0, RX0 routed to Port pins P0.1 and P0.2 (C8051F360/3) or P0.4 and P0.5 (C8051F361/2/4/5/6/7/8/9).

SFR Definition 17.2. XBR1: Port I/O Crossbar Register 1

SFR Page: F
SFR Address: 0xE2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: WEAKPUD: Port I/O Weak Pullup Disable.
0: Weak Pullups enabled (except for Ports whose I/O are configured as analog input).
1: Weak Pullups disabled.

Bit 6: XBARE: Crossbar Enable.
0: Crossbar disabled.
1: Crossbar enabled.

Bit 5: T1E: T1 Enable
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.

Bit 4: T0E: T0 Enable
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.

Bit 3: ECIE: PCA0 External Counter Input Enable
0: ECI unavailable at Port pin.
1: ECI routed to Port pin.

Bits 2–0: PCA0ME: PCA Module I/O Enable Bits.
000: All PCA I/O unavailable at Port pins.
001: CEX0 routed to Port pin.
010: CEX0, CEX1 routed to Port pins.
011: CEX0, CEX1, CEX2 routed to Port pins.
100: CEX0, CEX1, CEX2, CEX3 routed to Port pins.
101: CEX0, CEX1, CEX2, CEX3, CEX4 routed to Port pins.
110: CEX0, CEX1, CEX2, CEX3, CEX4, CEX5 routed to Port pins.
111: Reserved.

17.3. General Purpose Port I/O

Port pins that remain unassigned by the Crossbar and are not used by analog peripherals can be used for general purpose I/O. Ports P0-P3 are accessed through corresponding special function registers (SFRs) that are both byte-addressable and bit-addressable. Port 4 (C8051F360/3 only) uses an SFR which is byte-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.

In addition to performing general purpose I/O, P0, P1, and P2 can generate a port match event if the logic levels of the Port's input pins match a software controlled value. A port match event is generated if (P0 & P0MASK) does not equal (P0MATCH & P0MASK), if (P1 & P1MASK) does not equal (P1MATCH & P1MASK), or if (P2 & P2MASK) does not equal (P2MATCH & P2MASK). This allows Software to be notified if a certain change or pattern occurs on P0, P1, or P2 input pins regardless of the XBRn settings. A port match event can cause an interrupt if EMAT (EIE2.1) is set to '1' or cause the internal oscillator to awaken from SUSPEND mode. See Section "16.1.1. Internal Oscillator Suspend Mode" on page 170 for more information.

SFR Definition 17.3. P0: Port0

SFR Page: **all pages**
SFR Address: **0x80** (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P0.[7:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P0MDOUT.n bit = 0).
Read - Always reads '0' if selected as analog input in register P0MDIN. Directly reads Port pin when configured as digital input.
0: P0.n pin is logic low.
1: P0.n pin is logic high.

SFR Definition 17.4. P0MDIN: Port0 Input Mode

SFR Page: F
SFR Address: 0xF1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Analog Input Configuration Bits for P0.7-P0.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P0.n pin is configured as an analog input.
1: Corresponding P0.n pin is not configured as an analog input.

SFR Definition 17.5. P0MDOUT: Port0 Output Mode

SFR Page: F
SFR Address: 0xA4

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	

Bits 7–0: 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.

SFR Definition 17.6. P0SKIP: Port0 Skip

SFR Page: F
SFR Address: 0xD4

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	

Bits 7–0: 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 (V_{REF} 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.

SFR Definition 17.7. P0MAT: Port0 Match

SFR Page: 0
SFR Address: 0xF3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P0MAT[7:0]: Port0 Match Value.
These bits control the value that unmasked P0 Port pins are compared against. A Port Match event is generated if (P0 & P0MASK) does not equal (P0MAT & P0MASK).

SFR Definition 17.8. P0MASK: Port0 Mask

SFR Page: 0
SFR Address: 0xF4

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	

Bits 7–0: P0MASK[7:0]: Port0 Mask Value.
These bits select which Port pins will be compared to the value stored in P0MAT.
0: Corresponding P0.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P0.n pin is compared to the corresponding bit in P0MAT.

SFR Definition 17.9. P1: Port1

SFR Page: **all pages**
SFR Address: **0x90** (bit addressable)

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	

Bits 7–0: 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.

SFR Definition 17.10. P1MDIN: Port1 Input Mode

SFR Page: **F**
SFR Address: **0xF2**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Analog Input Configuration Bits for P1.7-P1.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P1.n pin is configured as an analog input.
1: Corresponding P1.n pin is not configured as an analog input.

SFR Definition 17.11. P1MDOUT: Port1 Output Mode

SFR Page: F
SFR Address: 0xA5

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	

Bits 7–0: Output Configuration Bits for P1.7-P1.0 (respectively): ignored if corresponding bit in register P1MDIN is logic '0'.
0: Corresponding P1.n Output is open-drain.
1: Corresponding P1.n Output is push-pull.

SFR Definition 17.12. P1SKIP: Port1 Skip

SFR Page: F
SFR Address: 0xD5

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	

Bits 7–0: P1SKIP[7:0]: Port1 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 (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P1.n pin is not skipped by the Crossbar.
1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 17.13. P1MAT: Port1 Match

SFR Page: 0
SFR Address: 0xE1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P1MAT[7:0]: Port1 Match Value.
These bits control the value that unmasked P0 Port pins are compared against. A Port Match event is generated if (P1 & P1MASK) does not equal (P1MAT & P1MASK).

SFR Definition 17.14. P1MASK: Port1 Mask

SFR Page: 0
SFR Address: 0xE2

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	

Bits 7–0: P1MASK[7:0]: Port1 Mask Value.
These bits select which Port pins will be compared to the value stored in P1MAT.
0: Corresponding P1.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P1.n pin is compared to the corresponding bit in P1MAT.

SFR Definition 17.15. P2: Port2

SFR Page: all pages
SFR Address: 0xA0 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P2.[7:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P2MDOUT.n bit = 0).
Read - Always reads ‘0’ if selected as analog input in register P2MDIN. Directly reads Port pin when configured as digital input.
0: P2.n pin is logic low.
1: P2.n pin is logic high.

SFR Definition 17.16. P2MDIN: Port2 Input Mode

SFR Page: F
SFR Address: 0xF3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Analog Input Configuration Bits for P2.7-P2.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P2.n pin is configured as an analog input.
1: Corresponding P2.n pin is not configured as an analog input.

SFR Definition 17.17. P2MDOUT: Port2 Output Mode

SFR Page: F
SFR Address: 0xA6

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	

Bits 7–0: Output Configuration Bits for P2.7-P2.0 (respectively): ignored if corresponding bit in register P2MDIN is logic ‘0’.
0: Corresponding P2.n Output is open-drain.
1: Corresponding P2.n Output is push-pull.

SFR Definition 17.18. P2SKIP: Port2 Skip

SFR Page: F
SFR Address: 0xD6

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	

Bits 7–0: P2SKIP[7:0]: Port2 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 (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P2.n pin is not skipped by the Crossbar.
1: Corresponding P2.n pin is skipped by the Crossbar.

SFR Definition 17.19. P2MAT: Port2 Match

SFR Page: 0
SFR Address: 0xB1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P2MAT[7:0]: Port2 Match Value.
These bits control the value that unmasked P2 Port pins are compared against. A Port Match event is generated if (P2 & P2MASK) does not equal (P2MAT & P2MASK).

SFR Definition 17.20. P2MASK: Port2 Mask

SFR Page: 0
SFR Address: 0xB2

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	

Bits 7–0: P2MASK[7:0]: Port2 Mask Value.
These bits select which Port pins will be compared to the value stored in P2MAT.
0: Corresponding P2.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P2.n pin is compared to the corresponding bit in P2MAT.

SFR Definition 17.21. P3: Port3

SFR Page: all pages
SFR Address: 0xB0 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: P3.[7:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P3MDOUT.n bit = 0).
Read - Always reads '0' if selected as analog input in register P3MDIN. Directly reads Port pin when configured as digital input.
0: P3.n pin is logic low.
1: P3.n pin is logic high.

SFR Definition 17.22. P3MDIN: Port3 Input Mode

SFR Page: F
SFR Address: 0xF4

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: Analog Input Configuration Bits for P3.7-P3.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P3.n pin is configured as an analog input.
1: Corresponding P3.n pin is not configured as an analog input.

SFR Definition 17.23. P3MDOUT: Port3 Output Mode

SFR Page: F
SFR Address: 0xAF

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	

Bits 7–0: Output Configuration Bits for P3.7-P3.0 (respectively): ignored if corresponding bit in register P3MDIN is logic '0'.
0: Corresponding P3.n Output is open-drain.
1: Corresponding P3.n Output is push-pull.

SFR Definition 17.24. P3SKIP: Port3 Skip

SFR Page: F
SFR Address: 0xD7

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	

Bits 7–0: P3SKIP[7:0]: Port3 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 (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P3.n pin is not skipped by the Crossbar.
1: Corresponding P3.n pin is skipped by the Crossbar.

SFR Definition 17.25. P4: Port4

SFR Page: all pages
SFR Address: 0xB5

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
–	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	01111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: UNUSED. Read = 0b. Write = don't care.
Bits 6–0: P4.[6:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P4MDOUT.n bit = 0).
Read - Directly reads Port pin.
0: P4.n pin is logic low.
1: P4.n pin is logic high.

SFR Definition 17.26. P4MDOUT: Port4 Output Mode

SFR Page: F
SFR Address: 0xAE

R	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	

Bit 7: UNUSED. Read = 0b. Write = don't care.
Bits 6–0: Output Configuration Bits for P4.6-P4.0 (respectively).
0: Corresponding P4.n Output is open-drain.
1: Corresponding P4.n Output is push-pull.

Table 17.1. Port I/O DC Electrical Characteristics

V_{DD} = 2.7 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	I_{OH} = –3 mA, Port I/O push-pull	V_{DD} – 0.7	—	—	V
	I_{OH} = –10 µA, Port I/O push-pull	V_{DD} – 0.1	—	—	
	I_{OH} = –10 mA, Port I/O push-pull	—	V_{DD} – 0.8	—	
Output Low Voltage	I_{OL} = 8.5 mA	—	—	0.6	V
	I_{OL} = 10 µA	—	—	0.1	
	I_{OL} = 25 mA	—	1.0	—	
Input High Voltage		2.0	—	—	V
Input Low Voltage		—	—	0.8	V
Input Leakage Current	Weak Pullup Off	—	—	±1	µA
	Weak Pullup On, V_{IN} = 0 V	—	25	50	

18. SMBus

The SMBus I/O interface is a two-wire, bi-directional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I^2C serial bus. Reads and writes to the interface by the system controller are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/10th of the system clock as a master or slave (this can be faster than allowed by the SMBus specification, depending on the system clock used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The SMBus interface may operate as a master and/or slave, and may function on a bus with multiple masters. The SMBus provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and START/STOP control and generation. Three SFRs are associated with the SMBus: SMB0CF configures the SMBus; SMB0CN controls the status of the SMBus; and SMB0DAT is the data register, used for both transmitting and receiving SMBus data and slave addresses.

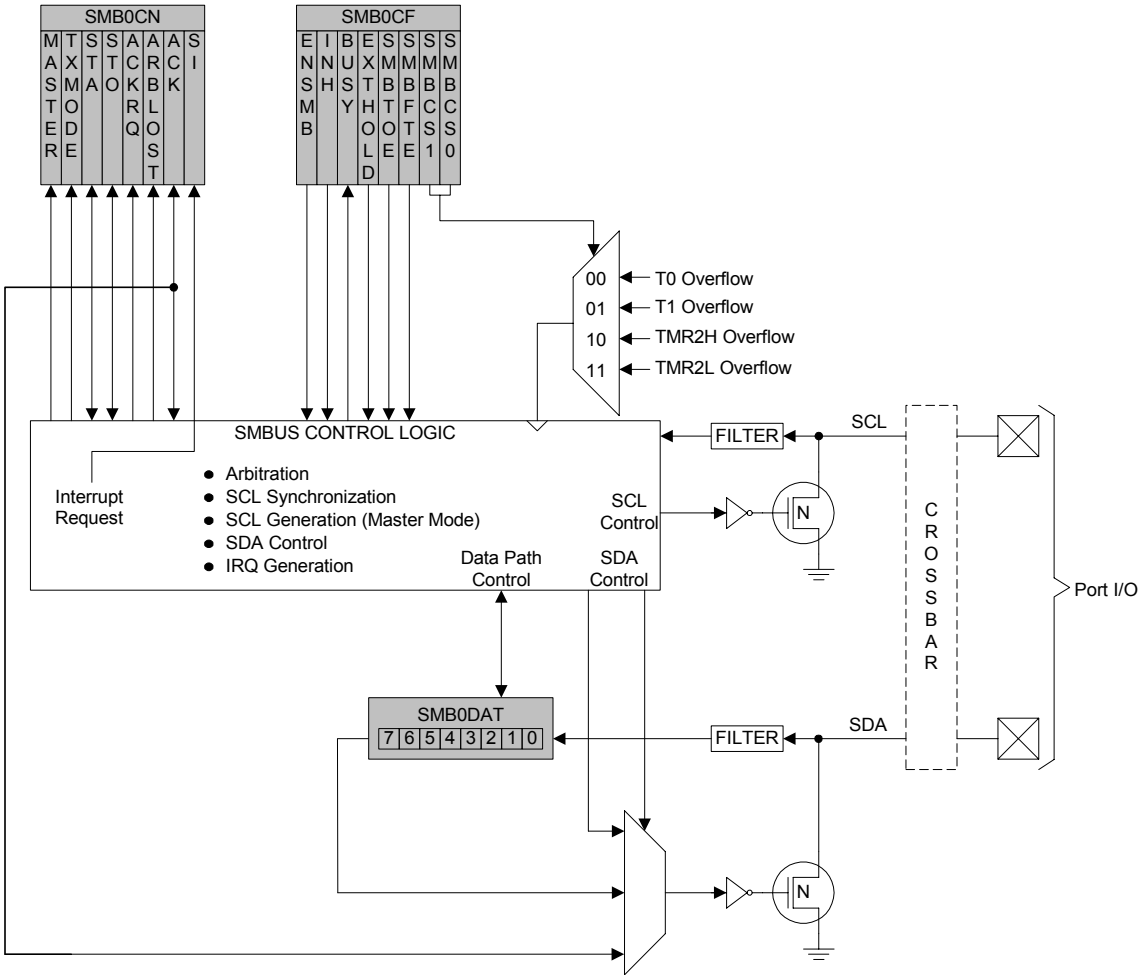


Figure 18.1. SMBus Block Diagram

18.1. Supporting Documents

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

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

18.2. SMBus Configuration

Figure 18.2 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 pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

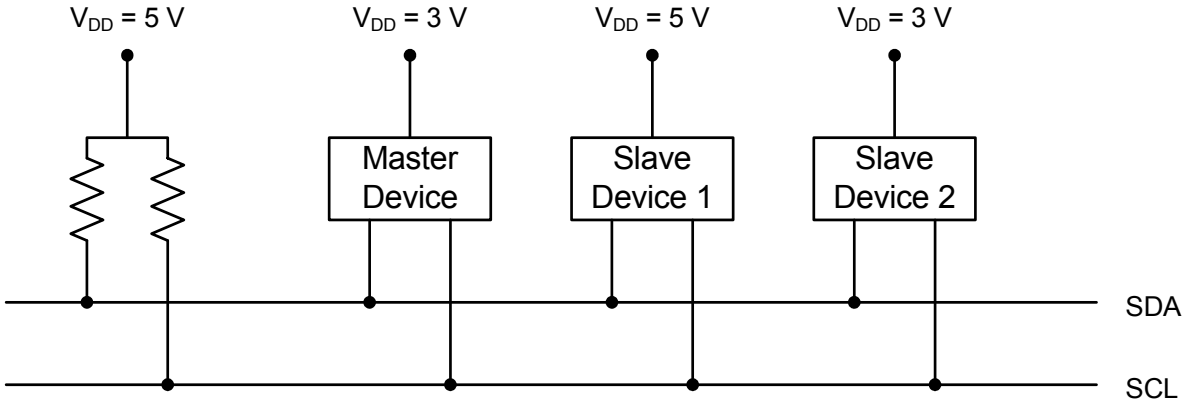


Figure 18.2. Typical SMBus Configuration

18.3. SMBus Operation

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

A typical SMBus transaction consists of a START condition followed by an address byte (Bits7–1: 7-bit slave address; Bit0: R/W direction bit), one or more bytes of data, and a STOP condition. Each byte that is received (by a master or slave) must be acknowledged (ACK) with a low SDA during a high SCL (see Figure 18.3). If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

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

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the START condition and then transmits the slave address and direction bit. If the transaction is a WRITE operation from the master to the slave, the master transmits the data a byte at a time waiting for an ACK from the slave at the end of each byte. For READ operations, the slave transmits the data waiting for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a STOP condition to terminate the transaction and free the bus. Figure 18.3 illustrates a typical SMBus transaction.

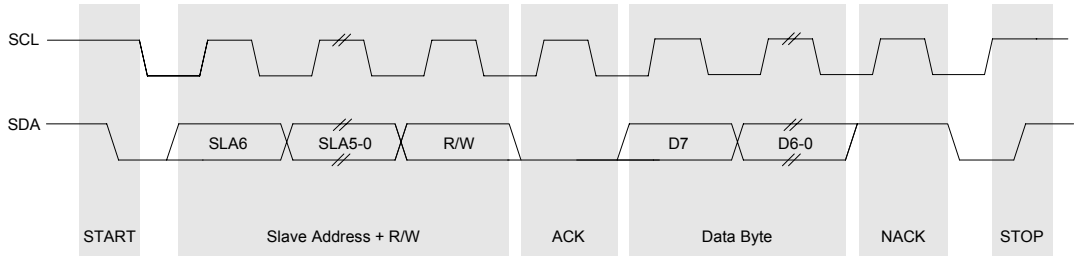


Figure 18.3. SMBus Transaction

18.3.1. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a STOP condition or after the SCL and SDA lines remain high for a specified time (see Section “18.3.4. SCL High (SMBus Free) Timeout” on page 205). In the event that two or more devices attempt to begin a transfer at the same time, an arbitration scheme is employed to force one master to give up the bus. The master devices continue transmitting until one attempts a HIGH while the other transmits a LOW. Since the bus is open-drain, the bus will be pulled LOW. The master attempting the HIGH will detect a LOW SDA and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave and receives the rest of the transfer if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

18.3.2. Clock Low Extension

SMBus provides a clock synchronization mechanism, similar to I2C, which allows devices with different speed capabilities to coexist on the bus. A clock-low extension is used during a transfer in order to allow slower slave devices to communicate with faster masters. The slave may temporarily hold the SCL line LOW to extend the clock low period, effectively decreasing the serial clock frequency.

18.3.3. SCL Low Timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset the communication no later than 10 ms after detecting the timeout condition.

When the SMBTOE bit in SMB0CF is set, Timer 3 is used to detect SCL low timeouts. Timer 3 is forced to reload when SCL is high, and allowed to count when SCL is low. With Timer 3 enabled and configured to

overflow after 25 ms (and SMBTOE set), the Timer 3 interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

18.3.4. SCL High (SMBus Free) Timeout

The SMBus specification stipulates that if the SCL and SDA lines remain high for more that 50 µs, the bus is designated as free. When the SMBFTE bit in SMB0CF is set, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods. If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. Note that a clock source is required for free timeout detection, even in a slave-only implementation.

18.4. Using the SMBus

The SMBus can operate in both Master and Slave modes. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The SMBus interface provides the following application-independent features:

- Byte-wise serial data transfers
- Clock signal generation on SCL (Master Mode only) and SDA data synchronization
- Timeout/bus error recognition, as defined by the SMB0CF configuration register
- START/STOP timing, detection, and generation
- Bus arbitration
- Interrupt generation
- Status information

SMBus interrupts are generated for each data byte or slave address that is transferred. When transmitting, this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data, this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. See Section “18.5. SMBus Transfer Modes” on page 213 for more details on transmission sequences.

Interrupts are also generated to indicate the beginning of a transfer when a master (START generated), or the end of a transfer when a slave (STOP detected). Software should read the SMB0CN (SMBus Control register) to find the cause of the SMBus interrupt. The SMB0CN register is described in Section “18.4.2. SMB0CN Control Register” on page 209; Table 18.4 provides a quick SMB0CN decoding reference.

SMBus configuration options include:

- Timeout detection (SCL Low Timeout and/or Bus Free Timeout)
- SDA setup and hold time extensions
- Slave event enable/disable
- Clock source selection

These options are selected in the SMB0CF register, as described in Section “18.4.1. SMBus Configuration Register” on page 206.

18.4.1. SMBus Configuration Register

The SMBus Configuration register (SMB0CF) is used to enable the SMBus Master and/or Slave modes, select the SMBus clock source, and select the SMBus timing and timeout options. When the ENSMB bit is set, the SMBus is enabled for all master and slave events. Slave events may be disabled by setting the INH bit. With slave events inhibited, the SMBus interface will still monitor the SCL and SDA pins; however, the interface will NACK all received addresses and will not generate any slave interrupts. When the INH bit is set, all slave events will be inhibited following the next START (interrupts will continue for the duration of the current transfer).

Table 18.1. SMBus Clock Source Selection

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

The SMBCS1–0 bits select the SMBus clock source, which is used only when operating as a master or when the Free Timeout detection is enabled. When operating as a master, overflows from the selected source determine the absolute minimum SCL low and high times as defined in Equation 18.1. Note that the selected clock source may be shared by other peripherals so long as the timer is left running at all times. For example, Timer 1 overflows may generate the SMBus and UART baud rates simultaneously. Timer configuration is covered in Section “21. Timers” on page 247.

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

Equation 18.1. Minimum SCL High and Low Times

The selected clock source should be configured to establish the minimum SCL High and Low times as per Equation 18.1. When the interface is operating as a master (and SCL is not driven or extended by any other devices on the bus), the typical SMBus bit rate is approximated by Equation 18.2.

$$BitRate = \frac{f_{ClockSourceOverflow}}{3}$$

Equation 18.2. Typical SMBus Bit Rate

Figure 18.4 shows the typical SCL generation described by Equation 18.2. Notice that T_{HIGH} is typically twice as large as T_{LOW}. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 18.1.

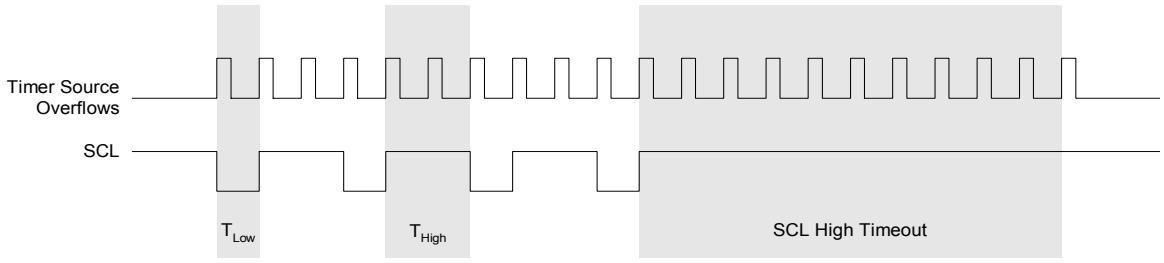


Figure 18.4. Typical SMBus SCL Generation

Setting the EXTHOLD bit extends the minimum setup and hold times for the SDA line. The minimum SDA setup time defines the absolute minimum time that SDA is stable before SCL transitions from low-to-high. The minimum SDA hold time defines the absolute minimum time that the current SDA value remains stable after SCL transitions from high-to-low. EXTHOLD should be set so that the minimum setup and hold times meet the SMBus Specification requirements of 250 ns and 300 ns, respectively. Table 18.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary when SYSCLK is above 10 MHz.

Table 18.2. 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. 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 “18.3.3. SCL Low Timeout” on page 204). 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 18.4). When a Free Timeout is detected, the interface will respond as if a STOP was detected (an interrupt will be generated, and STO will be set).

SFR Definition 18.1. SMB0CF: SMBus Clock/Configuration

SFR Page: all pages
SFR Address: 0xC1

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	

Bit 7: 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.

Bit 6: 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.

Bit 5: 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.

Bit 4: 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.

Bit 3: 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.

Bit 2: 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.

Bits 1–0: 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 18.1.

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

18.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 18.2). The higher four bits of SMB0CN (MASTER, TXMODE, STA, and STO) form a status vector that can be used to jump to service routines. MASTER and TXMODE indicate the master/slave state and transmit/receive modes, respectively.

STA and STO indicate that a START and/or STOP has been detected or generated since the last SMBus interrupt. STA and STO are also used to generate START and STOP conditions when operating as a master. Writing a ‘1’ to STA will cause the SMBus interface to enter Master Mode and generate a START when the bus becomes free (STA is not cleared by hardware after the START is generated). Writing a ‘1’ to STO while in Master Mode will cause the interface to generate a STOP and end the current transfer after the next ACK cycle. If STO and STA are both set (while in Master Mode), a STOP followed by a START will be generated.

As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. ACKRQ is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQ is set, software should write the desired outgoing value to the ACK bit before clearing SI. A NACK will be generated if software does not write the ACK bit before clearing SI. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however SCL will remain low until SI is cleared. If a received slave address is not acknowledged, further slave events will be ignored until the next START is detected.

The ARBLOST bit indicates that the interface has lost an arbitration. This may occur anytime the interface is transmitting (master or slave). A lost arbitration while operating as a slave indicates a bus error condition. ARBLOST is cleared by hardware each time SI is cleared.

The SI bit (SMBus Interrupt Flag) is set at the beginning and end of each transfer, after each byte frame, or when an arbitration is lost; see Table 18.3 for more details.

Important Note About the SI Bit: The SMBus interface is stalled while SI is set; thus SCL is held low, and the bus is stalled until software clears SI.

Table 18.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 18.4 for SMBus status decoding using the SMB0CN register.

SFR Definition 18.2. SMB0CN: SMBus Control

SFR Page: all pages
SFR Address: 0xC0 (bit addressable)

R	R	R/W	R/W	R	R	R/W	R/W	Reset Value
MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: MASTER: SMBus Master/Slave Indicator.
This read-only bit indicates when the SMBus is operating as a master.
0: SMBus operating in Slave Mode.
1: SMBus operating in Master Mode.

Bit 6: TXMODE: SMBus Transmit Mode Indicator.
This read-only bit indicates when the SMBus is operating as a transmitter.
0: SMBus in Receiver Mode.
1: SMBus in Transmitter Mode.

Bit 5: STA: SMBus Start Flag.
Write:
0: No Start generated.
1: When operating as a master, a START condition is transmitted if the bus is free (If the bus is not free, the START is transmitted after a STOP is received or a timeout is detected). If STA is set by software as an active Master, a repeated START will be generated after the next ACK cycle.
Read:
0: No Start or repeated Start detected.
1: Start or repeated Start detected.

Bit 4: STO: SMBus Stop Flag.
Write:
0: No STOP condition is transmitted.
1: Setting STO to logic ‘1’ causes a STOP condition to be transmitted after the next ACK cycle. When the STOP condition is generated, hardware clears STO to logic ‘0’. If both STA and STO are set, a STOP condition is transmitted followed by a START condition.
Read:
0: No Stop condition detected.
1: Stop condition detected (if in Slave Mode) or pending (if in Master Mode).

Bit 3: ACKRQ: SMBus Acknowledge Request
This read-only bit is set to logic ‘1’ when the SMBus has received a byte and needs the ACK bit to be written with the correct ACK response value.

Bit 2: ARBLOST: SMBus Arbitration Lost Indicator.
This read-only bit is set to logic ‘1’ when the SMBus loses arbitration while operating as a transmitter. A lost arbitration while a slave indicates a bus error condition.

Bit 1: ACK: SMBus Acknowledge Flag.
This bit defines the out-going ACK level and records incoming ACK levels. It should be written each time a byte is received (when ACKRQ=1), or read after each byte is transmitted.
0: A "not acknowledge" has been received (if in Transmitter Mode) OR will be transmitted (if in Receiver Mode).
1: An "acknowledge" has been received (if in Transmitter Mode) OR will be transmitted (if in Receiver Mode).

Bit 0: SI: SMBus Interrupt Flag.
This bit is set by hardware under the conditions listed in Table 18.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.

Table 18.3. Sources for Hardware Changes to SMB0CN

Bit	Set by Hardware When:	Cleared by Hardware When:
MASTER	• A START is generated.	• A STOP is generated. • Arbitration is lost.
TXMODE	• START is generated. • SMB0DAT is written before the start of an SMBus frame.	• A START is detected. • Arbitration is lost. • SMB0DAT is not written before the start of an SMBus frame.
STA	• A START followed by an address byte is received.	• Must be cleared by software.
STO	• A STOP is detected while addressed as a slave. • Arbitration is lost due to a detected STOP.	• A pending STOP is generated.
ACKRQ	• A byte has been received and an ACK response value is needed.	• After each ACK cycle.
ARBLOST	• A repeated START is detected as a MASTER when STA is low (unwanted repeated START). • SCL is sensed low while attempting to generate a STOP or repeated START condition. • SDA is sensed low while transmitting a ‘1’ (excluding ACK bits).	• Each time SI is cleared.
ACK	• The incoming ACK value is low (ACKNOWLEDGE).	• The incoming ACK value is high (NOT ACKNOWLEDGE).
SI	• A START has been generated. • Lost arbitration. • A byte has been transmitted and an ACK/NACK received. • A byte has been received. • A START or repeated START followed by a slave address + R/W has been received. • A STOP has been received.	• Must be cleared by software.

18.4.3. Data Register

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

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

SFR Definition 18.3. SMB0DAT: SMBus Data

SFR Page: all pages
SFR Address: 0xC2

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	

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

18.5. SMBus Transfer Modes

The SMBus interface may be configured to operate as master and/or slave. At any particular time, it will be operating in one of the following four modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. The SMBus interface enters Master Mode any time a START is generated, and remains in Master Mode until it loses an arbitration or generates a STOP. An SMBus interrupt is generated at the end of all SMBus byte frames; however, note that the interrupt is generated before the ACK cycle when operating as a receiver, and after the ACK cycle when operating as a transmitter.

18.5.1. Master Transmitter Mode

Serial data is transmitted on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic ‘0’ (WRITE). The master then transmits one or more bytes of serial data. After each byte is transmitted, an acknowledge bit is generated by the slave. The transfer is ended when the STO bit is set and a STOP is generated. Note that the interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 18.5 shows a typical Master Transmitter sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that the ‘data byte transferred’ interrupts occur **after** the ACK cycle in this mode.

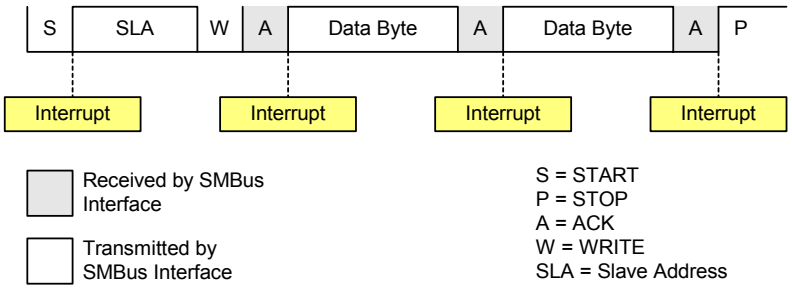


Figure 18.5. Typical Master Transmitter Sequence

18.5.2. Master Receiver Mode

Serial data is received on SDA while the serial clock is output on SCL. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic ‘1’ (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. After each byte is received, ACKRQ is set to ‘1’ and an interrupt is generated. Software must write the ACK bit (SMB0CN.1) to define the outgoing acknowledge value (Note: writing a ‘1’ to the ACK bit generates an ACK; writing a ‘0’ generates a NACK). Software should write a ‘0’ to the ACK bit after the last byte is received, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 18.6 shows a typical Master Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

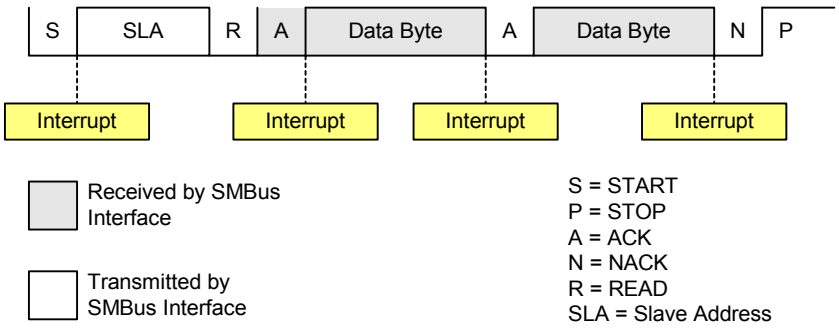


Figure 18.6. Typical Master Receiver Sequence

18.5.3. Slave Receiver Mode

Serial data is received on SDA and the clock is received on SCL. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode when a START followed by a slave address and direction bit (WRITE in this case) is received. Upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. Software responds to the received slave address with an ACK, or ignores the received slave address with a NACK. If the received slave address is ignored, slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are received. Software must write the ACK bit after each received byte to ACK or NACK the received byte. The interface exits Slave Receiver Mode after receiving a STOP. Note that the interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 18.7 shows a typical Slave Receiver sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur **before** the ACK cycle in this mode.

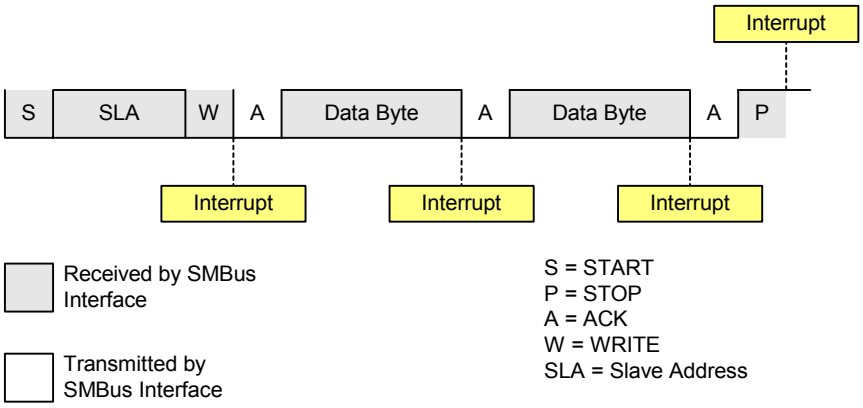


Figure 18.7. Typical Slave Receiver Sequence

18.5.4. Slave Transmitter Mode

Serial data is transmitted on SDA and the clock is received on SCL. When slave events are enabled (INH = 0), the interface enters Slave Receiver Mode (to receive the slave address) when a START followed by a slave address and direction bit (READ in this case) is received. Upon entering Slave Transmitter Mode, an interrupt is generated and the ACKRQ bit is set. Software responds to the received slave address with an ACK, or ignores the received slave address with a NACK. If the received slave address is ignored, slave interrupts will be inhibited until a START is detected. If the received slave address is acknowledged, data should be written to SMB0DAT to be transmitted. The interface enters Slave Transmitter Mode, and transmits one or more bytes of data. After each byte is transmitted, the master sends an acknowledge bit; if the acknowledge bit is an ACK, SMB0DAT should be written with the next data byte. If the acknowledge bit is a NACK, SMB0DAT should not be written to before SI is cleared (Note: an error condition may be generated if SMB0DAT is written following a received NACK while in Slave Transmitter Mode). The interface exits Slave Transmitter Mode after receiving a STOP. Note that the interface will switch to Slave Receiver Mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 18.8 shows a typical Slave Transmitter sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. Notice that the 'data byte transferred' interrupts occur **after** the ACK cycle in this mode.

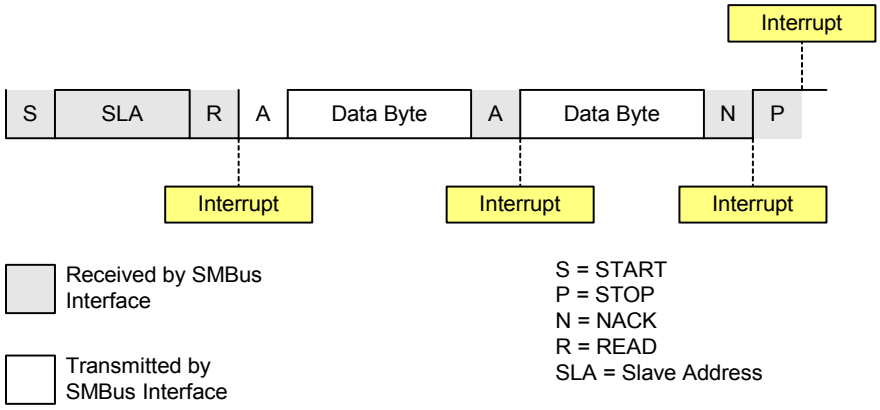


Figure 18.8. Typical Slave Transmitter Sequence

18.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. In the table below, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed but do not conform to the SMBus specification.

Table 18.4. SMBus Status Decoding

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X
						Abort transfer.	0	1	X
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X
						End transfer with STOP.	0	1	X
						End transfer with STOP and start another transfer.	1	1	X
						Send repeated START.	1	0	X
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X

Table 18.4. SMBus Status Decoding (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1
						Send NACK to indicate last byte, and send STOP.	0	1	0
						Send NACK to indicate last byte, and send STOP followed by START.	1	1	0
						Send ACK followed by repeated START.	1	0	1
						Send NACK to indicate last byte, and send repeated START.	1	0	0
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0
Slave Transmitter	0100	0	0	0	A slave byte was transmitted; NACK received.	No action required (expecting STOP condition).	0	0	X
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X
	0101	0	X	X	A STOP was detected while an addressed Slave Transmitter.	No action required (transfer complete).	0	0	X

Table 18.4. SMBus Status Decoding (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Slave Receiver	0010	1	0	X	A slave address was received; ACK requested.	Acknowledge received address.	0	0	1
						Do not acknowledge received address.	0	0	0
		1	1	X	Lost arbitration as master; slave address received; ACK requested.	Acknowledge received address.	0	0	1
						Do not acknowledge received address.	0	0	0
						Reschedule failed transfer; do not acknowledge received address.	1	0	0
		0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X
						Reschedule failed transfer.	1	0	X
	0001	1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0
		0	0	X	A STOP was detected while an addressed slave receiver.	No action required (transfer complete).	0	0	X
		0	1	X	Lost arbitration due to a detected STOP.	Abort transfer.	0	0	X
						Reschedule failed transfer.	1	0	X
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1
						Do not acknowledge received byte.	0	0	0
		1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0
						Reschedule failed transfer.	1	0	0

19. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in Section “19.1. Enhanced Baud Rate Generation” on page 221). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte.

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

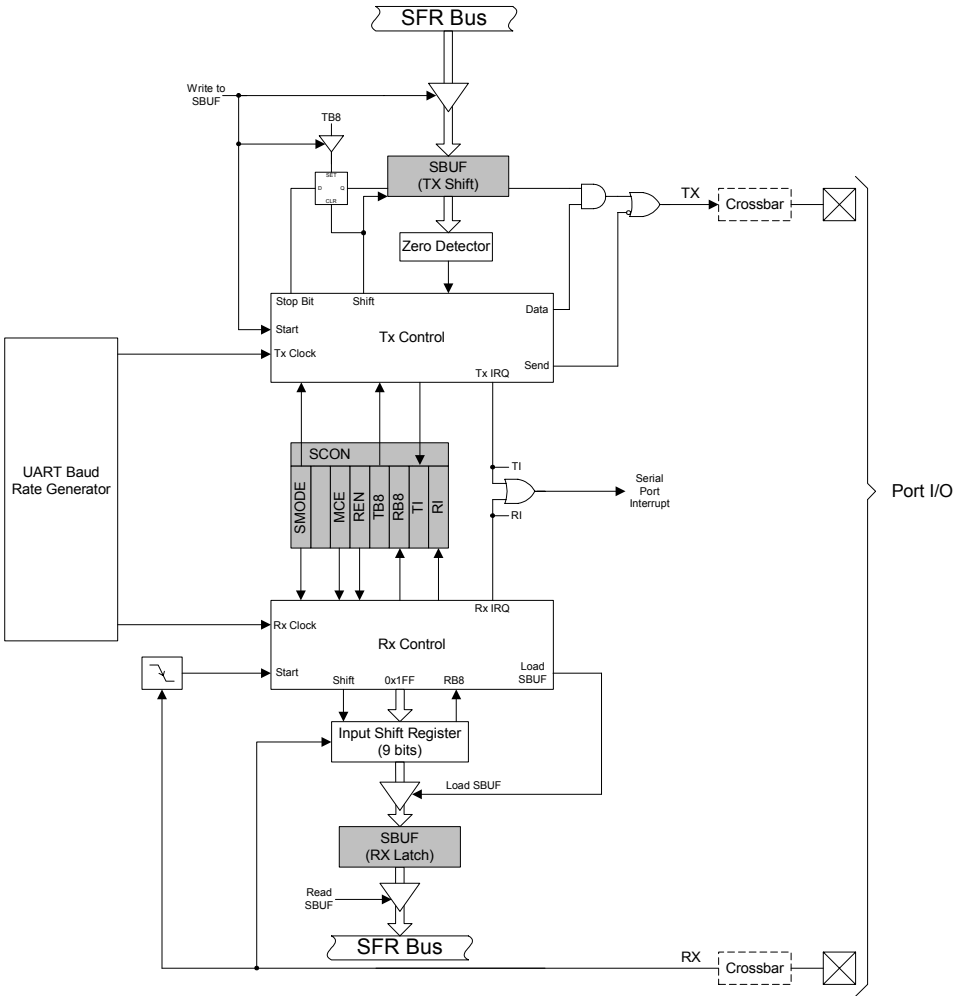


Figure 19.1. UART0 Block Diagram

19.1. 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 19.2), 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.

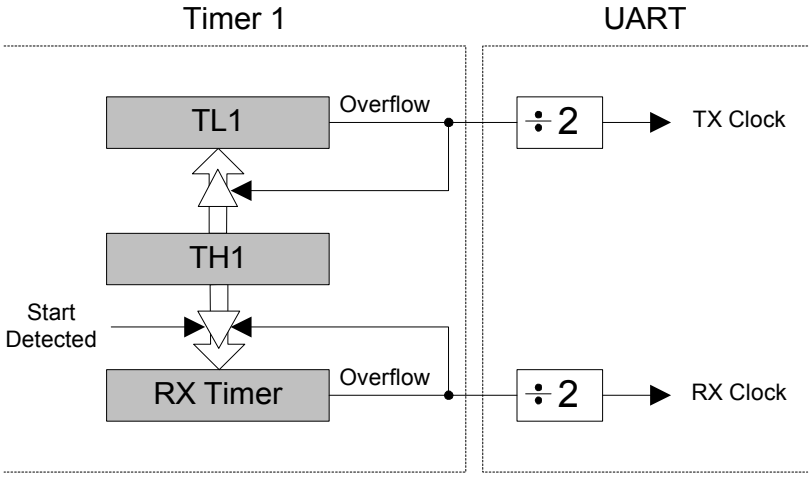


Figure 19.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see Section "21.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload" on page 249). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that 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 19.1-A and Equation 19.1-B.

A) $$\text{UartBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

B) $$\text{T1\_Overflow\_Rate} = \frac{\text{T1}_{\text{CLK}}}{256 - \text{TH1}}$$

Equation 19.1. 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 "21. Timers" on page 247. A quick reference for typical baud rates and system clock frequencies is given in Table 19.1 through Table 19.6. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

19.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown below.

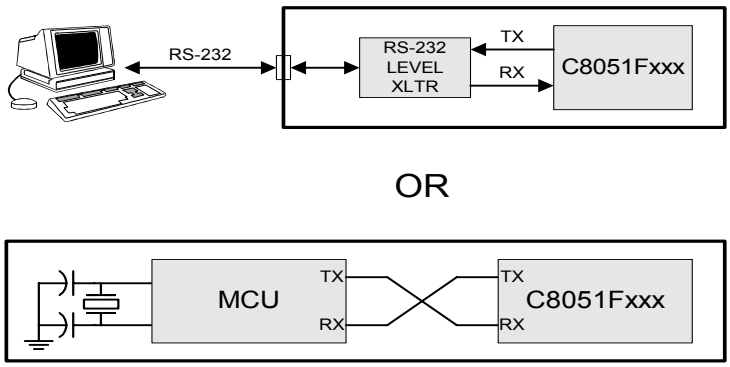


Figure 19.3. UART Interconnect Diagram

19.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic '1'. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic '0', and if MCE0 is logic '1', the stop bit must be logic '1'. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

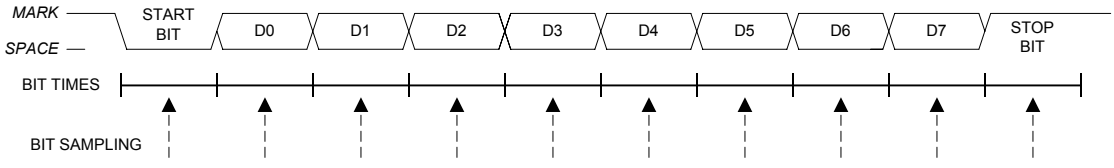


Figure 19.4. 8-Bit UART Timing Diagram

19.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic '0', and (2) if MCE0 is logic '1', the 9th bit must be logic '1' (when MCE0 is logic '0', the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to '1'. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to '1'. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to '1'.

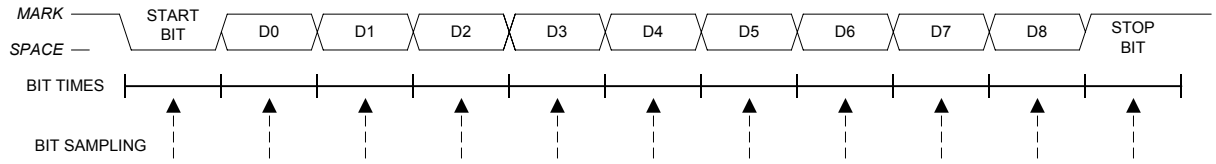


Figure 19.5. 9-Bit UART Timing Diagram

19.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic ‘1’; in a data byte, the ninth bit is always set to logic ‘0’.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic ‘1’ (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

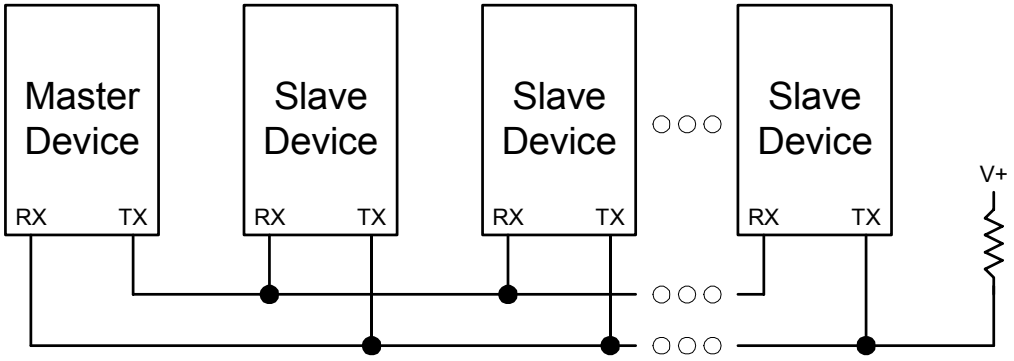


Figure 19.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 19.1. SCON0: Serial Port 0 Control

SFR Page: all pages
SFR Address: 0x98 (bit addressable)

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S0MODE	–	MCE0	REN0	TB80	RB80	TI0	RI0	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: S0MODE: Serial Port 0 Operation Mode.
This bit selects the UART0 Operation Mode.
0: 8-bit UART with Variable Baud Rate.
1: 9-bit UART with Variable Baud Rate.

Bit 6: UNUSED. Read = 1b. Write = don't care.

Bit 5: MCE0: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
S0MODE = 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI0 will only be activated if stop bit is logic level '1'.
S0MODE = 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI0 is set and an interrupt is generated only when the ninth bit is logic '1'.

Bit 4: REN0: Receive Enable.
This bit enables/disables the UART receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit 3: TB80: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit 2: RB80: Ninth Receive Bit.
RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit 1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

Bit 0: RI0: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

SFR Definition 19.2. SBUF0: Serial (UART0) Port Data Buffer

SFR Page: all pages
SFR Address: 0x99

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	

Bits 7–0: SBUF0[7:0]: Serial Data Buffer Bits 7–0 (MSB–LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Table 19.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

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

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

Table 19.2. Timer Settings for Standard Baud Rates Using an External 25.0 MHz Oscillator

	Frequency: 25.0 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 and Timer Clock from External Osc.	230400	–0.47%	108	SYSCLK	XX[2]	1	0xCA
	115200	0.45%	218	SYSCLK	XX	1	0x93
	57600	–0.01%	434	SYSCLK	XX	1	0x27
	28800	0.45%	872	SYSCLK/4	01	0	0x93
	14400	–0.01%	1736	SYSCLK/4	01	0	0x27
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D
	2400	0.45%	10464	SYSCLK/48	10	0	0x93
	1200	–0.01%	20832	SYSCLK/48	10	0	0x27
SYSCLK from Internal Osc., Timer Clock from External Osc.	57600	–0.47%	432	EXTCLK/8	11	0	0xE5
	28800	–0.47%	864	EXTCLK/8	11	0	0xCA
	14400	0.45%	1744	EXTCLK/8	11	0	0x93
	9600	0.15%	2608	EXTCLK/8	11	0	0x5D

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

Table 19.3. Timer Settings for Standard Baud Rates Using an External 22.1184 MHz Oscillator

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

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

Table 19.4. Timer Settings for Standard Baud Rates Using an External 18.432 MHz Oscillator

	Frequency: 18.432 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 and Timer Clock from External Osc.	230400	0.00%	80	SYSCLK	XX[2]	1	0xD8
	115200	0.00%	160	SYSCLK	XX	1	0xB0
	57600	0.00%	320	SYSCLK	XX	1	0x60
	28800	0.00%	640	SYSCLK/4	01	0	0xB0
	14400	0.00%	1280	SYSCLK/4	01	0	0x60
	9600	0.00%	1920	SYSCLK/12	00	0	0xB0
	2400	0.00%	7680	SYSCLK/48	10	0	0xB0
	1200	0.00%	15360	SYSCLK/48	10	0	0x60
SYSCLK from Internal Osc., Timer Clock from External Osc.	230400	0.00%	80	EXTCLK/8	11	0	0xFB
	115200	0.00%	160	EXTCLK/8	11	0	0xF6
	57600	0.00%	320	EXTCLK/8	11	0	0xEC
	28800	0.00%	640	EXTCLK/8	11	0	0xD8
	14400	0.00%	1280	EXTCLK/8	11	0	0xB0
	9600	0.00%	1920	EXTCLK/8	11	0	0x88

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

Table 19.5. Timer Settings for Standard Baud Rates Using an External 11.0592 MHz Oscillator

	Frequency: 11.0592 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 and Timer Clock from External Osc.	230400	0.00%	48	SYSCLK	XX[2]	1	0xE8
	115200	0.00%	96	SYSCLK	XX	1	0xD0
	57600	0.00%	192	SYSCLK	XX	1	0xA0
	28800	0.00%	384	SYSCLK	XX	1	0x40
	14400	0.00%	768	SYSCLK/12	00	0	0xE0
	9600	0.00%	1152	SYSCLK/12	00	0	0xD0
	2400	0.00%	4608	SYSCLK/12	00	0	0x40
	1200	0.00%	9216	SYSCLK/48	10	0	0xA0
SYSCLK from Internal Osc., Timer Clock from External Osc.	230400	0.00%	48	EXTCLK/8	11	0	0xFD
	115200	0.00%	96	EXTCLK/8	11	0	0xFA
	57600	0.00%	192	EXTCLK/8	11	0	0xF4
	28800	0.00%	384	EXTCLK/8	11	0	0xE8
	14400	0.00%	768	EXTCLK/8	11	0	0xD0
	9600	0.00%	1152	EXTCLK/8	11	0	0xB8

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

Table 19.6. Timer Settings for Standard Baud Rates Using an External 3.6864 MHz 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 and Timer Clock 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., Timer Clock from External 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 21.1.
2. X = Don't care.

Table 19.7. Timer Settings for Standard Baud Rates Using the PLL

Frequency: 50.0 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)
230400	0.45%	218	SYSCLK	XX[2]	1	0x93
115200	–0.01%	434	SYSCLK	XX	1	0x27
57600	0.45%	872	SYSCLK/4	01	0	0x93
28800	–0.01%	1736	SYSCLK/4	01	0	0x27
14400	0.22%	3480	SYSCLK/12	00	0	0x6F
9600	–0.01%	5208	SYSCLK/12	00	0	0x27
2400	–0.01%	20832	SYSCLK/48	10	0	0x27

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

Table 19.8. Timer Settings for Standard Baud Rates Using the PLL

Frequency: 100.0 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)
230400	–0.01%	434	SYSCLK	XX[2]	1	0x27
115200	0.45%	872	SYSCLK/4	01	0	0x93
57600	–0.01%	1736	SYSCLK/4	01	0	0x27
28800	0.22%	3480	SYSCLK/12	00	0	0x6F
14400	–0.47%	6912	SYSCLK/48	10	0	0xB8
9600	0.45%	10464	SYSCLK/48	10	0	0x93

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Section 21.1.
2. X = Don't care.

20. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

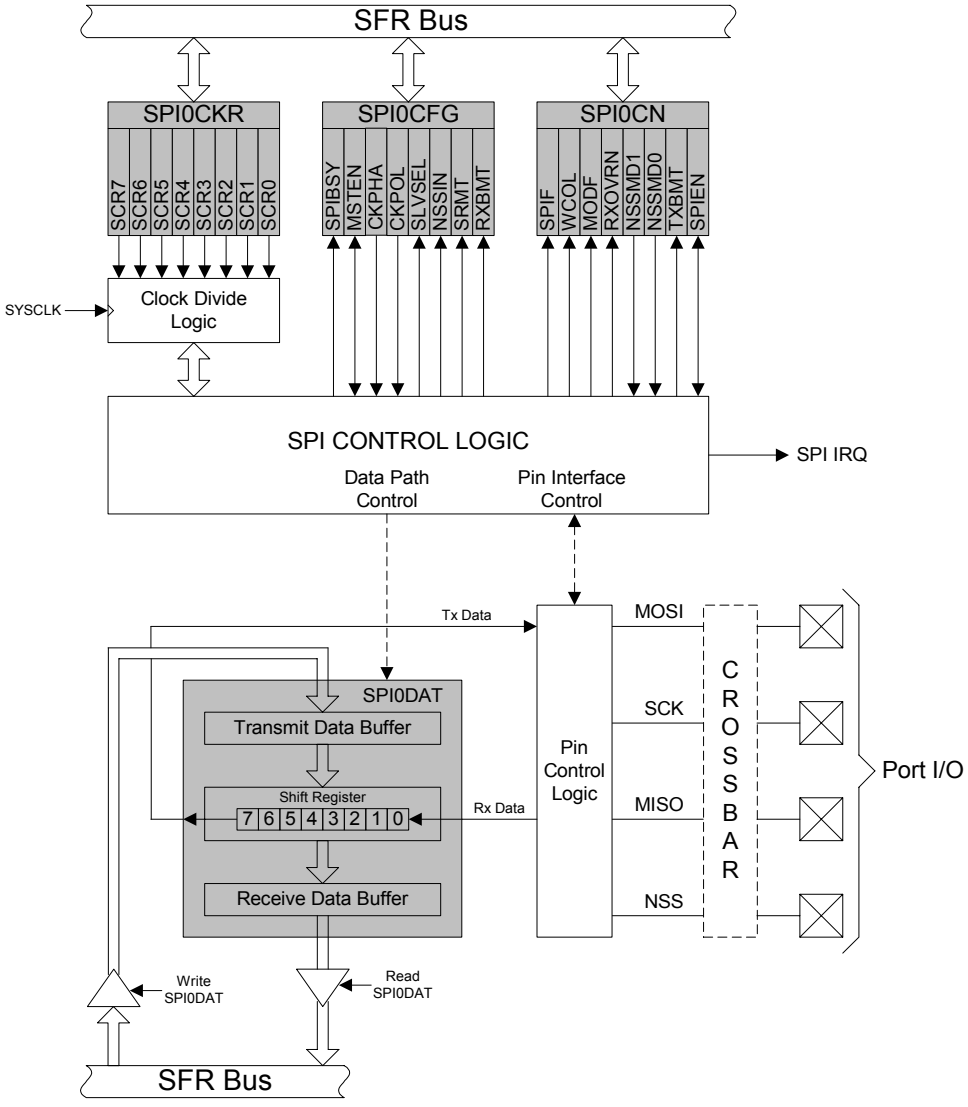


Figure 20.1. SPI Block Diagram

20.1. Signal Descriptions

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

20.1.1. 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.

20.1.2. 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.

20.1.3. 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.

20.1.4. 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 20.2, Figure 20.3, and Figure 20.4 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. See Section “17. Port Input/Output” on page 183 for general purpose port I/O and crossbar information.

20.2. 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 20.2 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 20.3 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 20.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

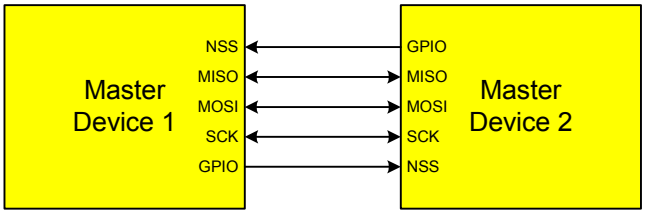


Figure 20.2. Multiple-Master Mode Connection Diagram

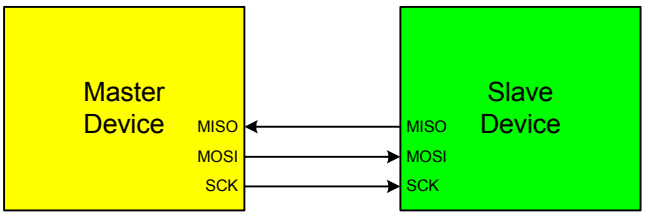


Figure 20.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

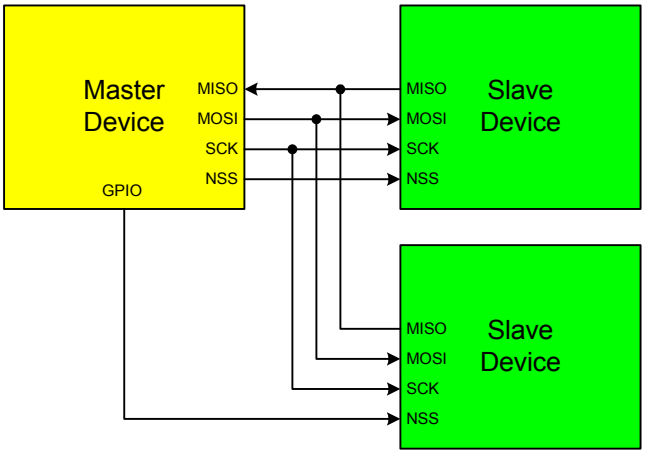


Figure 20.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

20.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted through the shift register, the SPIF flag is set to logic ‘1’, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer’s contents after the last SCK edge of the next (or current) SPI transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic ‘0’, and disabled when NSS is logic ‘1’. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 20.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 20.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

20.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic ‘1’:

All of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic ‘1’ at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic ‘1’ if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic ‘1’ when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic ‘0’ to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic ‘1’ when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

20.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 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 20.5. For slave mode, the clock and data relationships are shown in Figure 20.6 and Figure 20.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, C8051F33x, and C8051F36x.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 20.3 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.

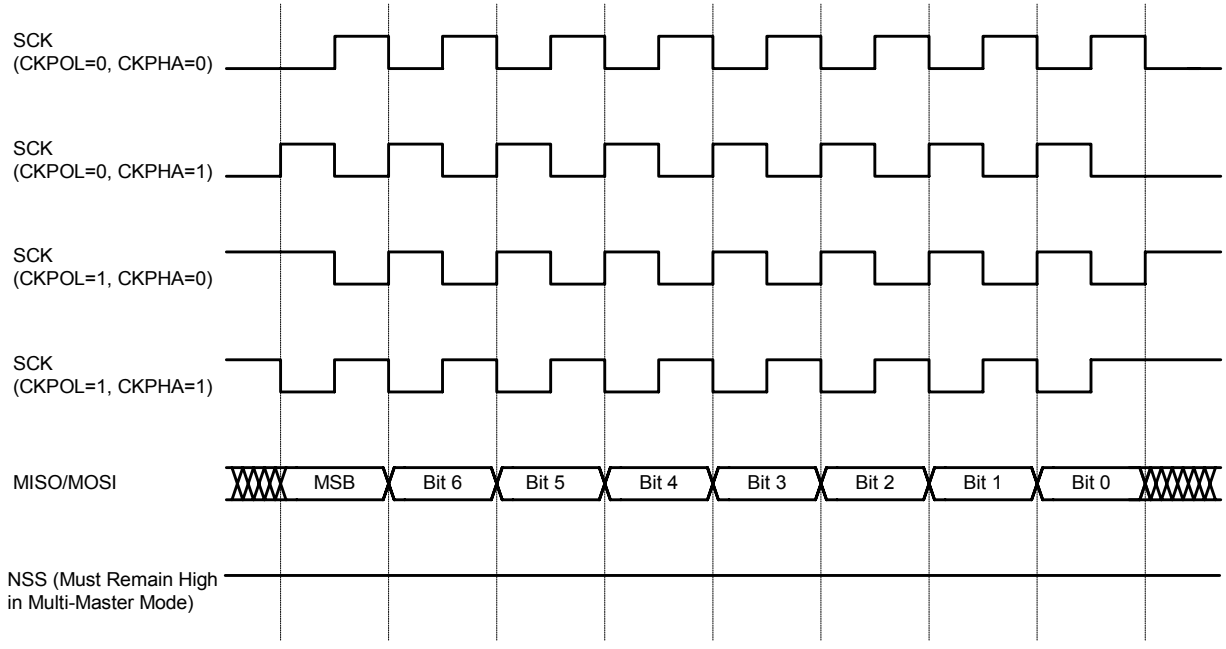


Figure 20.5. Master Mode Data/Clock Timing

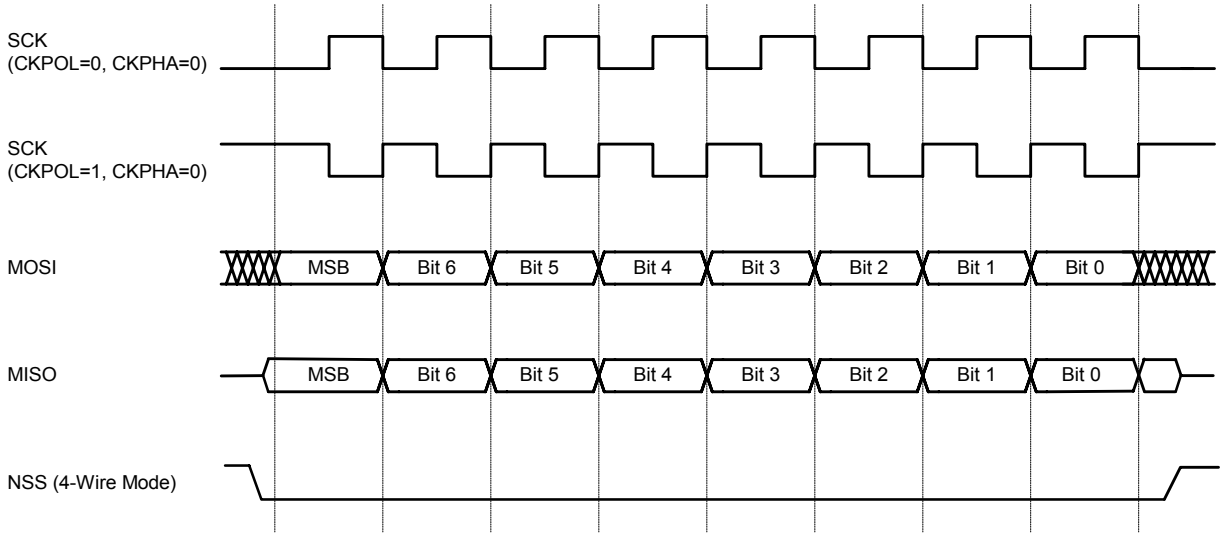


Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

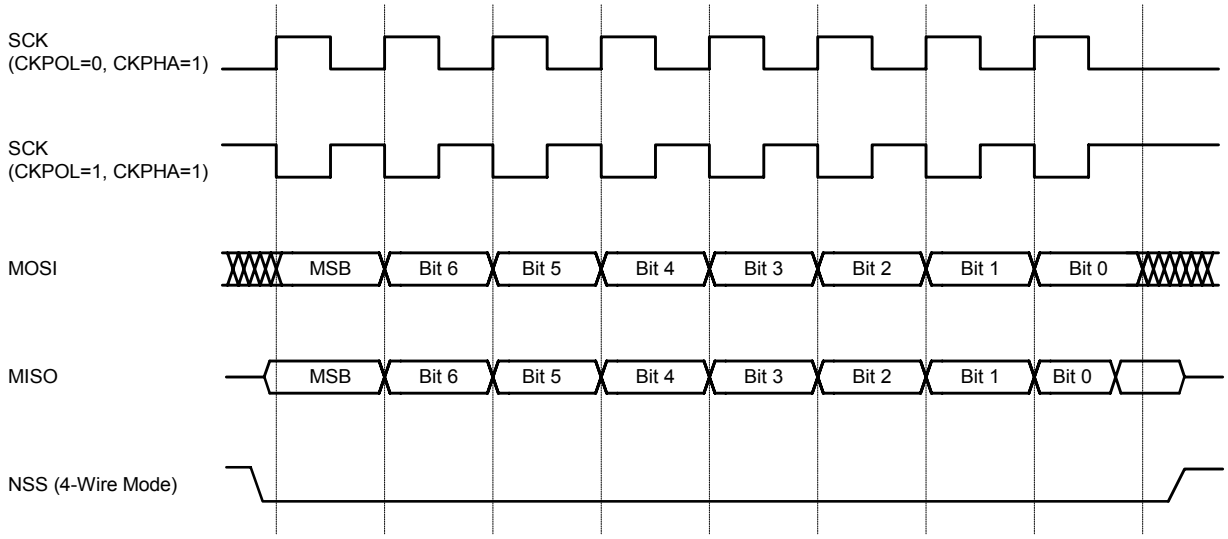


Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)

20.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 20.1. SPI0CFG: SPI0 Configuration

SFR Page: all pages
SFR Address: 0xA1

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: SPIBSY: SPI Busy (read only).
This bit is set to logic ‘1’ when a SPI transfer is in progress (Master or slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: CKPHA: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*

Bit 4: CKPOL: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: SLVSEL: Slave Selected Flag (read only).
This bit is set to logic ‘1’ whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic ‘0’ when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.

Bit 2: NSSIN: NSS Instantaneous Pin Input (read only).
This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.

Bit 1: SRMT: Shift Register Empty (Valid in Slave Mode, read only).
This bit will be set to logic ‘1’ when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic ‘0’ when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: RXBMT: Receive Buffer Empty (Valid in Slave Mode, read only).
This bit will be set to logic ‘1’ when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic ‘0’.
NOTE: RXBMT = 1 when in Master Mode.

***Note:** In slave mode, data on MOSI is sampled in the center of each data bit. In master mode, data on MISO is sampled one SYSCLK before the end of each data bit, to provide maximum settling time for the slave device. See Table 20.1 for timing parameters.

SFR Definition 20.2. SPI0CN: SPI0 Control

SFR Page: all pages
SFR Address: 0xF8 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
SPIF	WCOL	MODF	RXOVRN	NSSMD1	NSSMD0	TXBMT	SPIEN	00000110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: SPIF: SPI0 Interrupt Flag.
This bit is set to logic ‘1’ by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 6: WCOL: Write Collision Flag.
This bit is set to logic ‘1’ by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. It must be cleared by software.

Bit 5: MODF: Mode Fault Flag.
This bit is set to logic ‘1’ by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.

Bit 4: RXOVRN: Receive Overrun Flag (Slave Mode only).
This bit is set to logic ‘1’ by hardware (and generates a SPI0 interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. This bit is not automatically cleared by hardware. It must be cleared by software.

Bits 3–2: NSSMD1–NSSMD0: Slave Select Mode.
Selects between the following NSS operation modes:
(See Section 20.2 and Section 20.3).
00: 3-Wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin.
01: 4-Wire Slave or Multi-Master Mode (Default). NSS is always an input to the device.
1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.

Bit 1: TXBMT: Transmit Buffer Empty.
This bit will be set to logic ‘0’ when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic ‘1’, indicating that it is safe to write a new byte to the transmit buffer.

Bit 0: SPIEN: SPI0 Enable.
This bit enables/disables the SPI.
0: SPI disabled.
1: SPI enabled.

SFR Definition 20.3. SPI0CKR: SPI0 Clock Rate

SFR Page: all pages
SFR Address: 0xA2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SCR7–SCR0: SPI0 Clock Rate.
These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

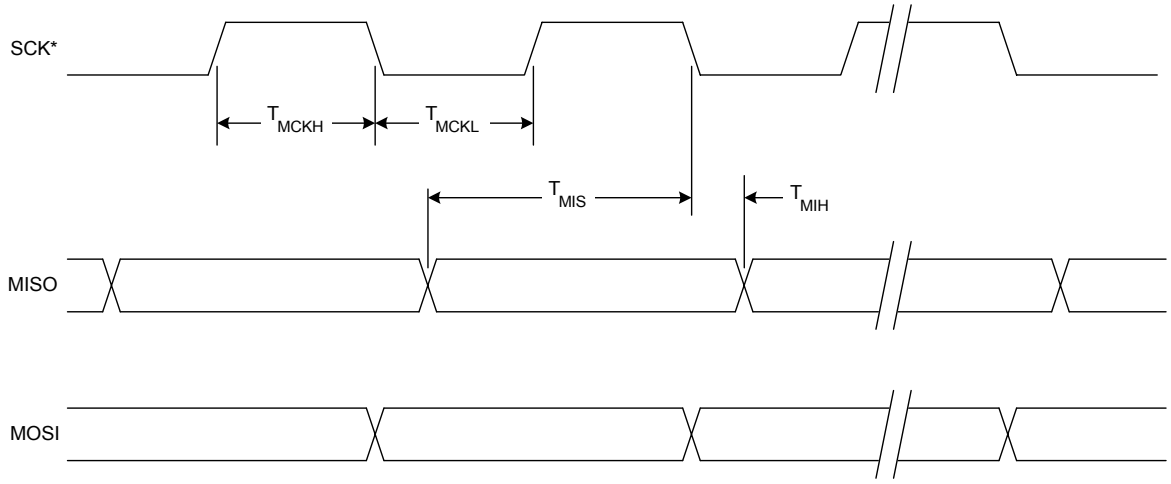
$$f_{SCK} = 200kHz$$

SFR Definition 20.4. SPI0DAT: SPI0 Data

SFR Page: all pages
SFR Address: 0xA3

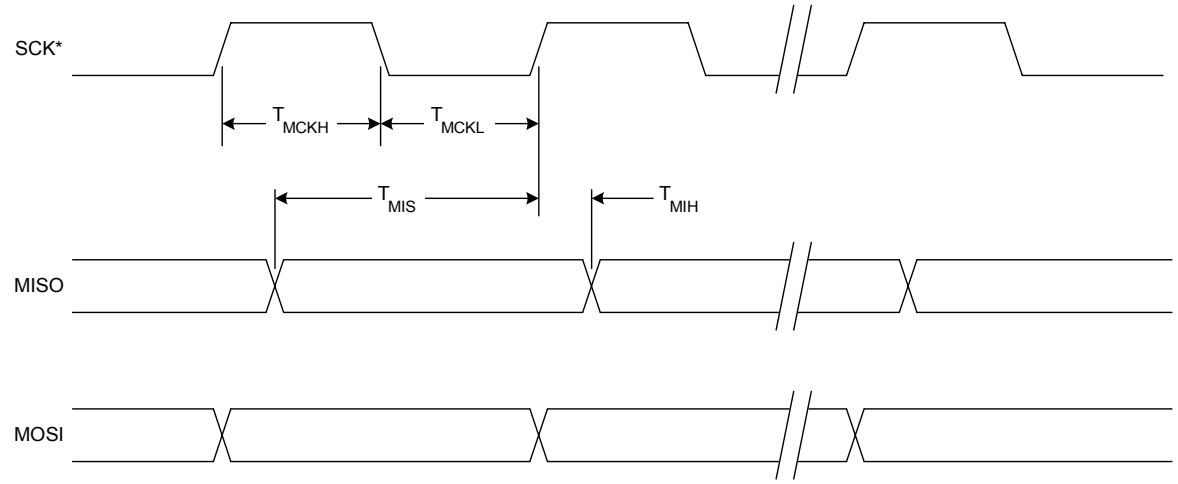
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits 7–0: SPI0DAT: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.



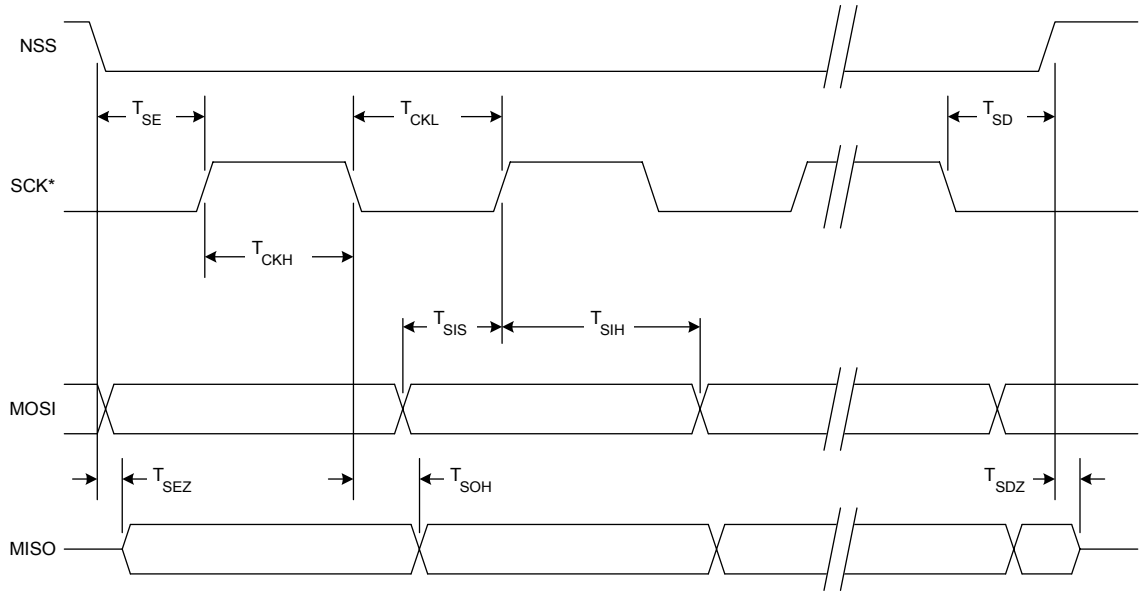
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.8. SPI Master Timing (CKPHA = 0)



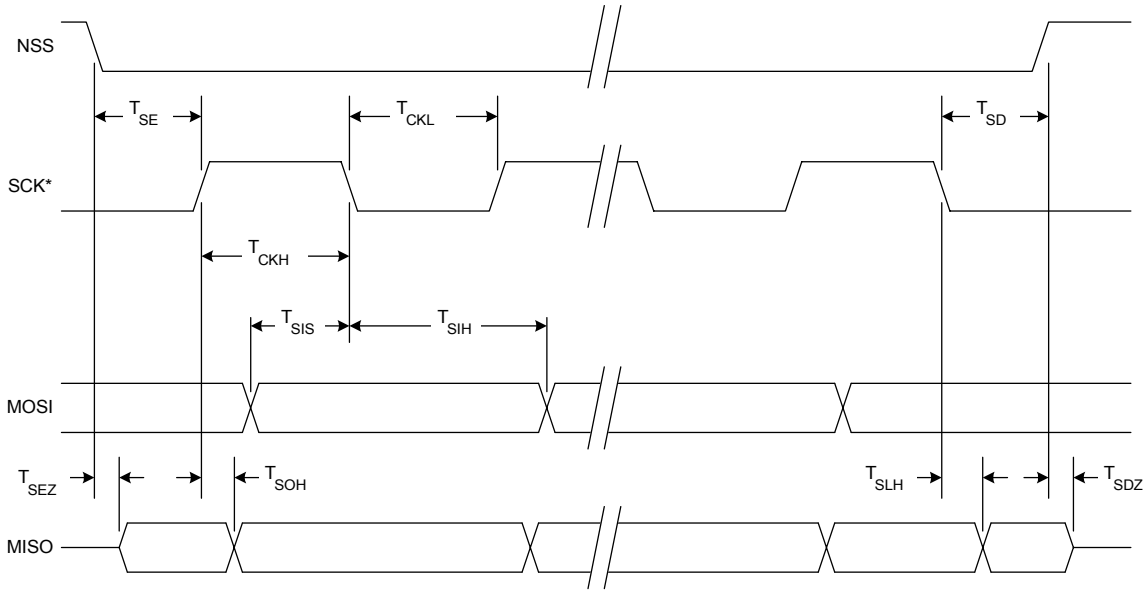
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.9. SPI Master Timing (CKPHA = 1)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.10. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.11. SPI Slave Timing (CKPHA = 1)

Table 20.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing* (See Figure 20.8 and Figure 20.9)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Shift Edge	1 x T_{SYSCLK} + 20	—	ns
T_{MIH}	SCK Shift Edge to MISO Change	0	—	ns
Slave Mode Timing* (See Figure 20.10 and Figure 20.11)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns
T_{SLH}	Last SCK Edge to MISO Change (CKPHA = 1 ONLY)	6 x T_{SYSCLK}	8 x T_{SYSCLK}	ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK).

21. Timers

Each MCU includes four counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and two are 16-bit auto-reload timer for use with the ADC, SMBus, or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 and Timer 3 offer 16-bit and split 8-bit timer functionality with auto-reload.

<table>
<tr><th>Timer 0 and Timer 1 Modes:</th><th>Timer 2 Modes:</th><th>Timer 3 Modes:</th></tr>
<tr><td>13-bit counter/timer</td><td rowspan="2">16-bit timer with auto-reload</td><td rowspan="2">16-bit timer with auto-reload</td></tr>
<tr><td>16-bit counter/timer</td></tr>
<tr><td>8-bit counter/timer with auto-reload</td><td rowspan="2">Two 8-bit timers with auto-reload</td><td rowspan="2">Two 8-bit timers with auto-reload</td></tr>
<tr><td>Two 8-bit counter/timers (Timer 0 only)</td></tr>
</table>

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 21.3 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it should be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

21.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section “10.4. Interrupt Register Descriptions” on page 109); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (Section 10.4). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

21.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic ‘1’, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to Section “17.1. Priority Crossbar Decoder” on page 185 for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 21.3).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic ‘0’ or the input signal /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 10.7). Setting GATE0 to ‘1’ allows the timer to be controlled by the external input signal /INT0 (see Section “10.4. Interrupt Register Descriptions” on page 109), facilitating pulse width measurements

TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal /INT1 is used with Timer 1; the /INT1 polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 10.7).

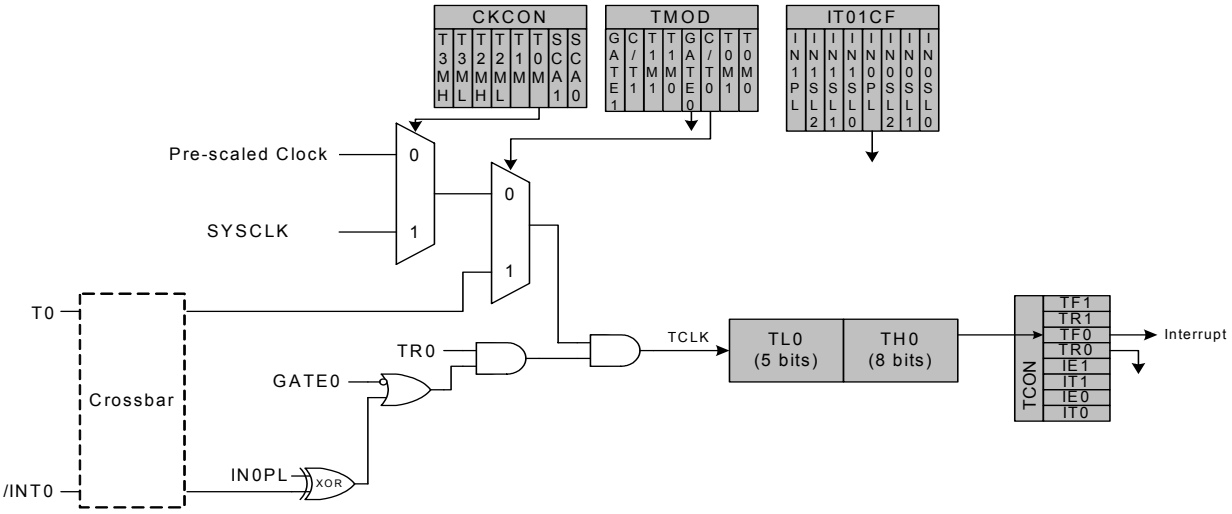


Figure 21.1. T0 Mode 0 Block Diagram

21.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

21.1.3. 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 (TCON.5) 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 (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic ‘0’ or when the input signal /INT0 is active as defined by bit IN0PL in register IT01CF (see Section “10.5. External Interrupts” on page 115 for details on the external input signals /INT0 and /INT1).

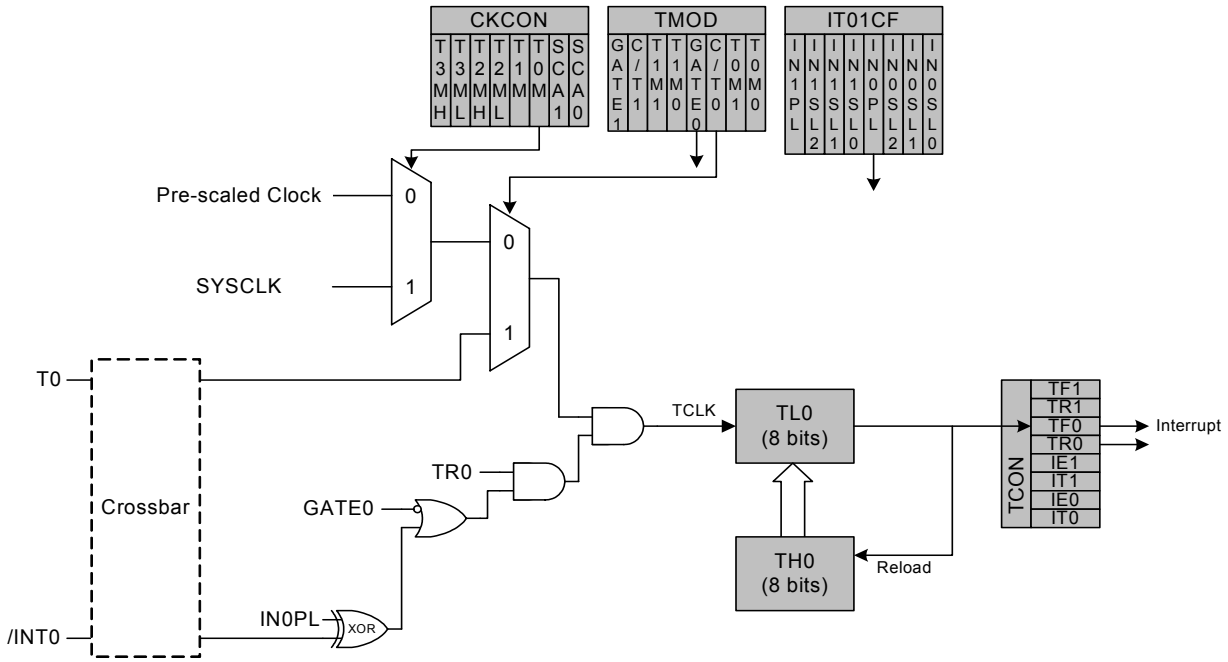


Figure 21.2. T0 Mode 2 Block Diagram

21.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and/or UART, and/or initiate ADC conversions. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

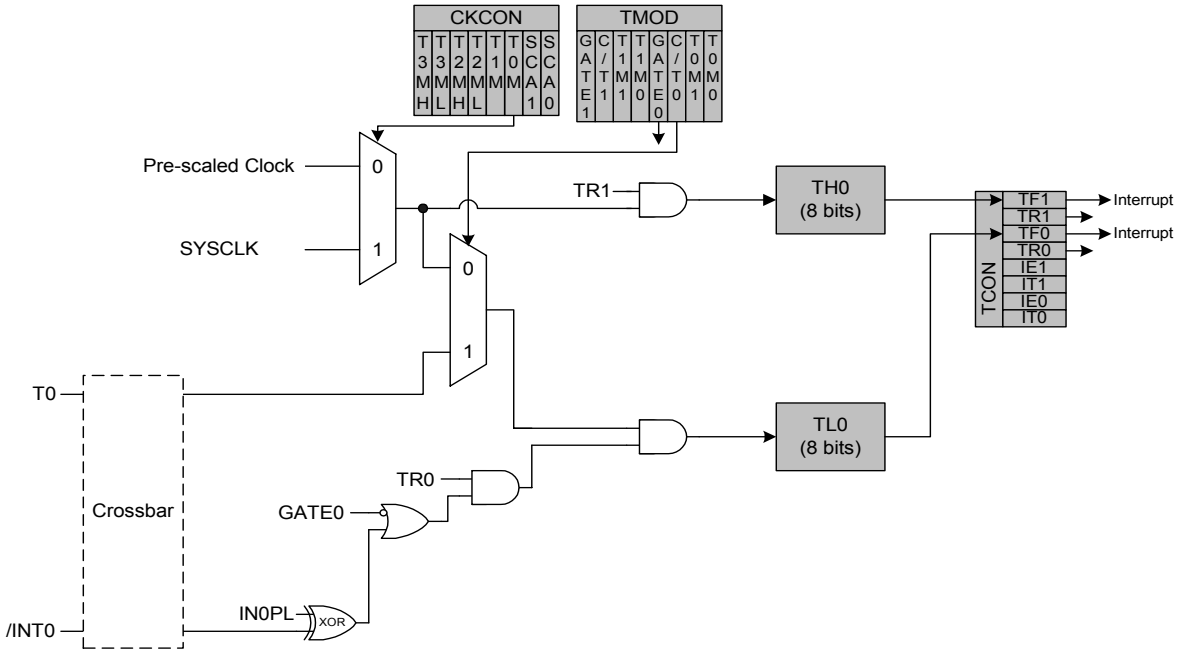


Figure 21.3. T0 Mode 3 Block Diagram

SFR Definition 21.1. TCON: Timer Control

SFR Page: all pages
SFR Address: 0x88 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: TF1: Timer 1 Overflow Flag.
Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

Bit 6: TR1: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit 5: TF0: Timer 0 Overflow Flag.
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

Bit 4: TR0: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit 3: IE1: External Interrupt 1.
This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. When IT1 = 0, this flag is set to ‘1’ when /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 10.7).

Bit 2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured /INT1 interrupt will be edge or level sensitive. /INT1 is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 10.7).
0: /INT1 is level triggered.
1: /INT1 is edge triggered.

Bit 1: IE0: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. When IT0 = 0, this flag is set to ‘1’ when /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 10.7).

Bit 0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured /INT0 interrupt will be edge or level sensitive. /INT0 is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 10.7).
0: /INT0 is level triggered.
1: /INT0 is edge triggered.

SFR Definition 21.2. TMOD: Timer Mode

SFR Page: all pages
SFR Address: 0x89

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 10.7).

Bit 6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits 5–4: T1M1–T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit 3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 10.7).

Bit 2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits 1–0: T0M1–T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

SFR Definition 21.3. CKCON: Clock Control

SFR Page: all pages
SFR Address: 0x8E

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: T3MH: Timer 3 High Byte Clock Select.
This bit selects the clock supplied to the Timer 3 high byte if Timer 3 is configured in split 8-bit timer mode. T3MH is ignored if Time 3 is in any other mode.
0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 high byte uses the system clock.

Bit 6: T3ML: Timer 3 Low Byte Clock Select.
This bit selects the clock supplied to Timer 3. If Timer 3 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN.
1: Timer 3 low byte uses the system clock.

Bit 5: T2MH: Timer 2 High Byte Clock Select.
This bit selects the clock supplied to the Timer 2 high byte if Timer 2 is configured in split 8-bit timer mode. T2MH is ignored if Timer 2 is in any other mode.
0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 high byte uses the system clock.

Bit 4: T2ML: Timer 2 Low Byte Clock Select.
This bit selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 low byte uses the system clock.

Bit 3: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic ‘1’.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.

Bit 2: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic ‘1’.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Counter/Timer 0 uses the system clock.

Bits 1–0: SCA1–SCA0: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48
1	1	External clock divided by 8

Note: External clock divided by 8 is synchronized with the system clock.

SFR Definition 21.4. TL0: Timer 0 Low Byte

SFR Page: all pages
SFR Address: 0x8A

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	

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 21.5. TL1: Timer 1 Low Byte

SFR Page: all pages
SFR Address: 0x8B

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	

Bits 7–0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

SFR Definition 21.6. TH0: Timer 0 High Byte

SFR Page: all pages
SFR Address: 0x8C

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	

Bits 7–0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 21.7. TH1: Timer 1 High Byte

SFR Page: all pages
SFR Address: 0x8D

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:

Bits 7–0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

21.2. 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.

21.2.1. 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 21.4, 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.

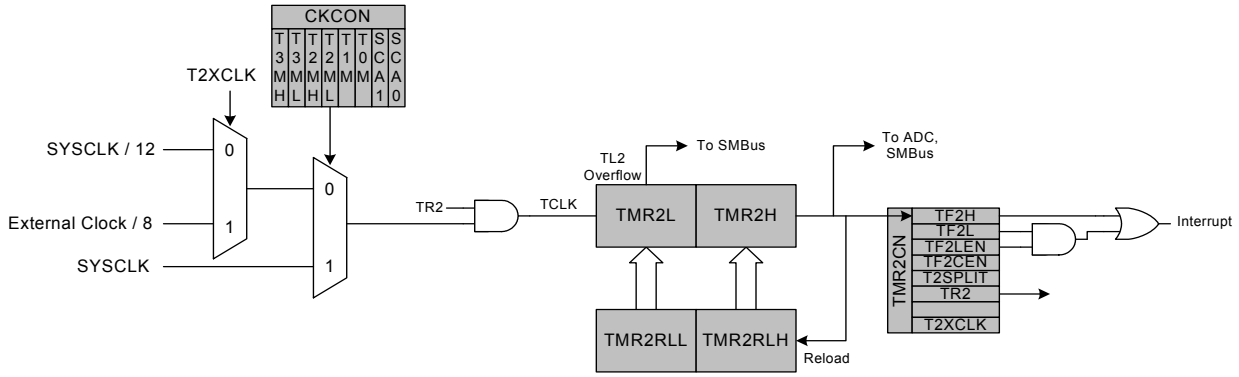


Figure 21.4. Timer 2 16-Bit Mode Block Diagram

21.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 21.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L 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 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

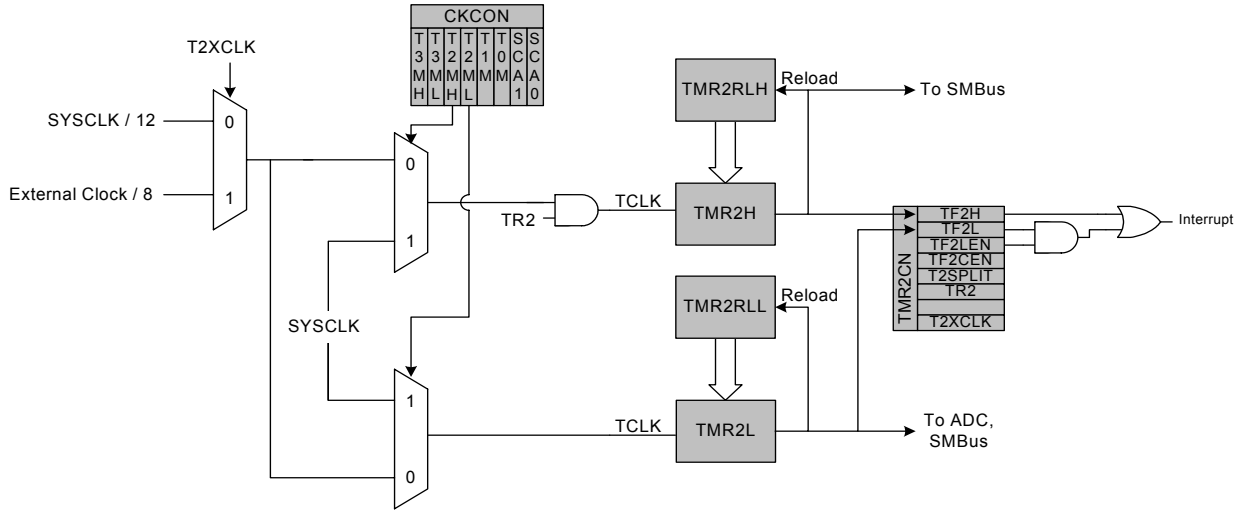


Figure 21.5. Timer 2 8-Bit Mode Block Diagram

SFR Definition 21.8. TMR2CN: Timer 2 Control

SFR Page: all pages
SFR Address: 0xC8
(bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	–	T2XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: TF2H: Timer 2 High Byte Overflow Flag.
Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. TF2H is not automatically cleared by hardware and must be cleared by software.

Bit 6: TF2L: Timer 2 Low Byte Overflow Flag.
Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF2LEN is set and Timer 2 interrupts are enabled. TF2L will set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.

Bit 5: TF2LEN: Timer 2 Low Byte Interrupt Enable.
This bit enables/disables Timer 2 Low Byte interrupts. If TF2LEN is set and Timer 2 interrupts are enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
0: Timer 2 Low Byte interrupts disabled.
1: Timer 2 Low Byte interrupts enabled.

Bit 4: TF2CEN: Timer 2 Low-Frequency Oscillator Capture Enable.
This bit enables/disables Timer 2 Low-Frequency Oscillator Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated on a falling edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL. See Section “16. Oscillators” on page 169 for more details.
0: Timer 2 Low-Frequency Oscillator Capture disabled.
1: Timer 2 Low-Frequency Oscillator Capture enabled.

Bit 3: T2SPLIT: Timer 2 Split Mode Enable.
When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload.
0: Timer 2 operates in 16-bit auto-reload mode.
1: Timer 2 operates as two 8-bit auto-reload timers.

Bit 2: TR2: Timer 2 Run Control.
This bit enables/disables Timer 2. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in this mode.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: T2XCLK: Timer 2 External Clock Select.
This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.
0: Timer 2 external clock selection is the system clock divided by 12.
1: Timer 2 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

SFR Definition 21.9. TMR2RLL: Timer 2 Reload Register Low Byte

SFR Page: all pages
SFR Address: 0xCA

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	

Bits 7–0: TMR2RLL: Timer 2 Reload Register Low Byte.
TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 21.10. TMR2RLH: Timer 2 Reload Register High Byte

SFR Page: all pages
SFR Address: 0xCB

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	

Bits 7–0: TMR2RLH: Timer 2 Reload Register High Byte.
The TMR2RLH holds the high byte of the reload value for Timer 2.

SFR Definition 21.11. TMR2L: Timer 2 Low Byte

SFR Page: all pages
SFR Address: 0xCC

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	

Bits 7–0: TMR2L: Timer 2 Low Byte.
In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

SFR Definition 21.12. TMR2H Timer 2 High Byte

SFR Page: all pages
SFR Address: 0xCD

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	

Bits 7–0: TMR2H: Timer 2 High Byte.
In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

21.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR3CN.3) defines the Timer 3 operation mode.

Timer 3 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 3 (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.

21.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 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 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 21.6, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

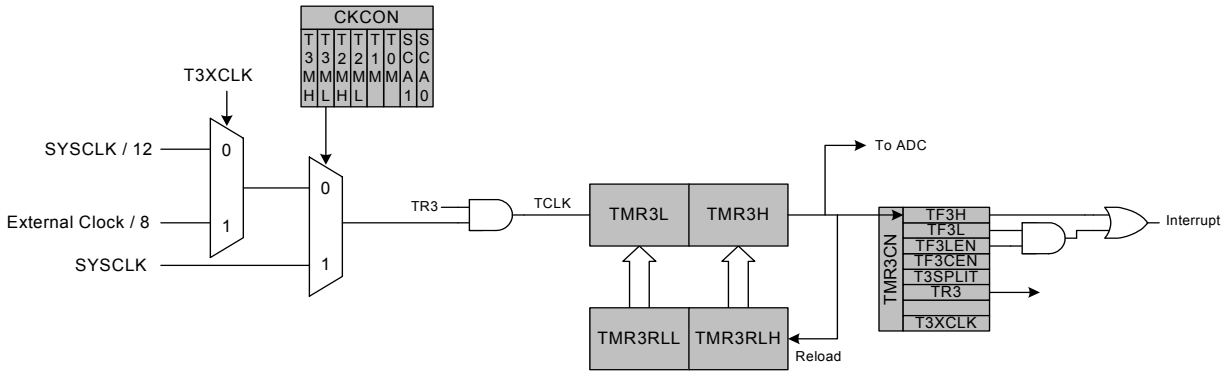


Figure 21.6. Timer 3 16-Bit Mode Block Diagram

21.3.2. 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 21.7. 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.

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	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.

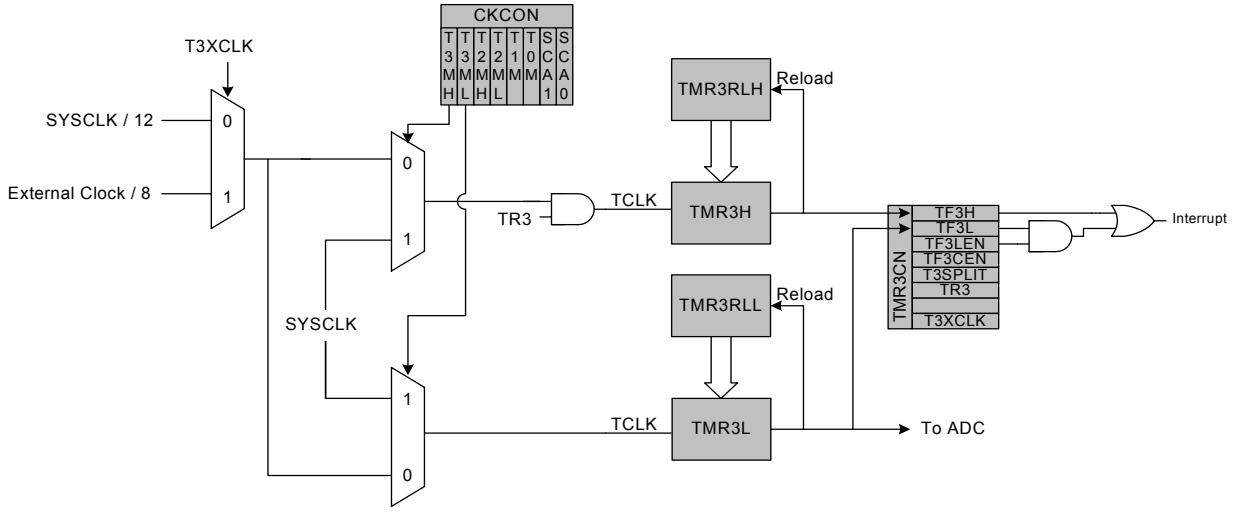


Figure 21.7. Timer 3 8-Bit Mode Block Diagram

SFR Definition 21.13. TMR3CN: Timer 3 Control

SFR Page: all pages
SFR Address: 0x91

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3	–	T3XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: TF3H: Timer 3 High Byte Overflow Flag.
Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. TF3H is not automatically cleared by hardware and must be cleared by software.

Bit 6: TF3L: Timer 3 Low Byte Overflow Flag.
Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF3LEN is set and Timer 3 interrupts are enabled. TF3L will set when the low byte overflows regardless of the Timer 3 mode. This bit is not automatically cleared by hardware.

Bit 5: TF3LEN: Timer 3 Low Byte Interrupt Enable.
This bit enables/disables Timer 3 Low Byte interrupts. If TF3LEN is set and Timer 3 interrupts are enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
0: Timer 3 Low Byte interrupts disabled.
1: Timer 3 Low Byte interrupts enabled.

Bit 4: TF3CEN: Timer 3 Low-Frequency Oscillator Capture Enable.
This bit enables/disables Timer 3 Low-Frequency Oscillator Capture Mode. If TF3CEN is set and Timer 3 interrupts are enabled, an interrupt will be generated on a rising edge of the low-frequency oscillator output, and the current 16-bit timer value in TMR3H:TMR3L will be copied to TMR3RLH:TMR3RLL. See Section “16. Oscillators” on page 169 for more details.
0: Timer 3 Low-Frequency Oscillator Capture disabled.
1: Timer 3 Low-Frequency Oscillator Capture enabled.

Bit 3: T3SPLIT: Timer 3 Split Mode Enable.
When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload.
0: Timer 3 operates in 16-bit auto-reload mode.
1: Timer 3 operates as two 8-bit auto-reload timers.

Bit 2: TR3: Timer 3 Run Control.
This bit enables/disables Timer 3. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in this mode.
0: Timer 3 disabled.
1: Timer 3 enabled.

Bit 1: UNUSED. Read = 0b. Write = don’t care.

Bit 0: T3XCLK: Timer 3 External Clock Select.
This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.
0: Timer 3 external clock selection is the system clock divided by 12.
1: Timer 3 external clock selection is the external clock divided by 8. Note that the external oscillator source divided by 8 is synchronized with the system clock.

SFR Definition 21.14. TMR3RLL: Timer 3 Reload Register Low Byte

SFR Page: all pages
SFR Address: 0x92

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	

Bits 7–0: TMR3RLL: Timer 3 Reload Register Low Byte.
TMR3RLL holds the low byte of the reload value for Timer 3.

SFR Definition 21.15. TMR3RLH: Timer 3 Reload Register High Byte

SFR Page: all pages
SFR Address: 0x93

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	

Bits 7–0: TMR3RLH: Timer 3 Reload Register High Byte.
The TMR3RLH holds the high byte of the reload value for Timer 3.

SFR Definition 21.16. TMR3L: Timer 3 Low Byte

SFR Page: all pages
SFR Address: 0x94

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	

Bits 7–0: TMR3L: Timer 3 Low Byte.
In 16-bit mode, the TMR3L register contains the low byte of the 16-bit Timer 3. In 8-bit mode, TMR3L contains the 8-bit low byte timer value.

SFR Definition 21.17. TMR3H Timer 3 High Byte

SFR Page: all pages
SFR Address: 0x95

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	

Bits 7–0: TMR3H: Timer 3 High Byte.
In 16-bit mode, the TMR3H register contains the high byte of the 16-bit Timer 3. In 8-bit mode, TMR3H contains the 8-bit high byte timer value.

22. Programmable Counter Array

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. PCA0 consists of a dedicated 16-bit counter/timer and six 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled (See Section “17.3. General Purpose Port I/O” on page 190). The counter/timer is driven by a programmable timebase that can select between six inputs as its source: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI line. Each capture/compare module may be configured to operate independently in one of six modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each is described in Section 22.2). The PCA is configured and controlled through the system controller's Special Function Registers. The basic PCA block diagram is shown in Figure 22.1.

Important Note: The PCA Module 5 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. **Access to certain PCA registers is restricted while WDT mode is enabled**. See Section 22.3 for details.

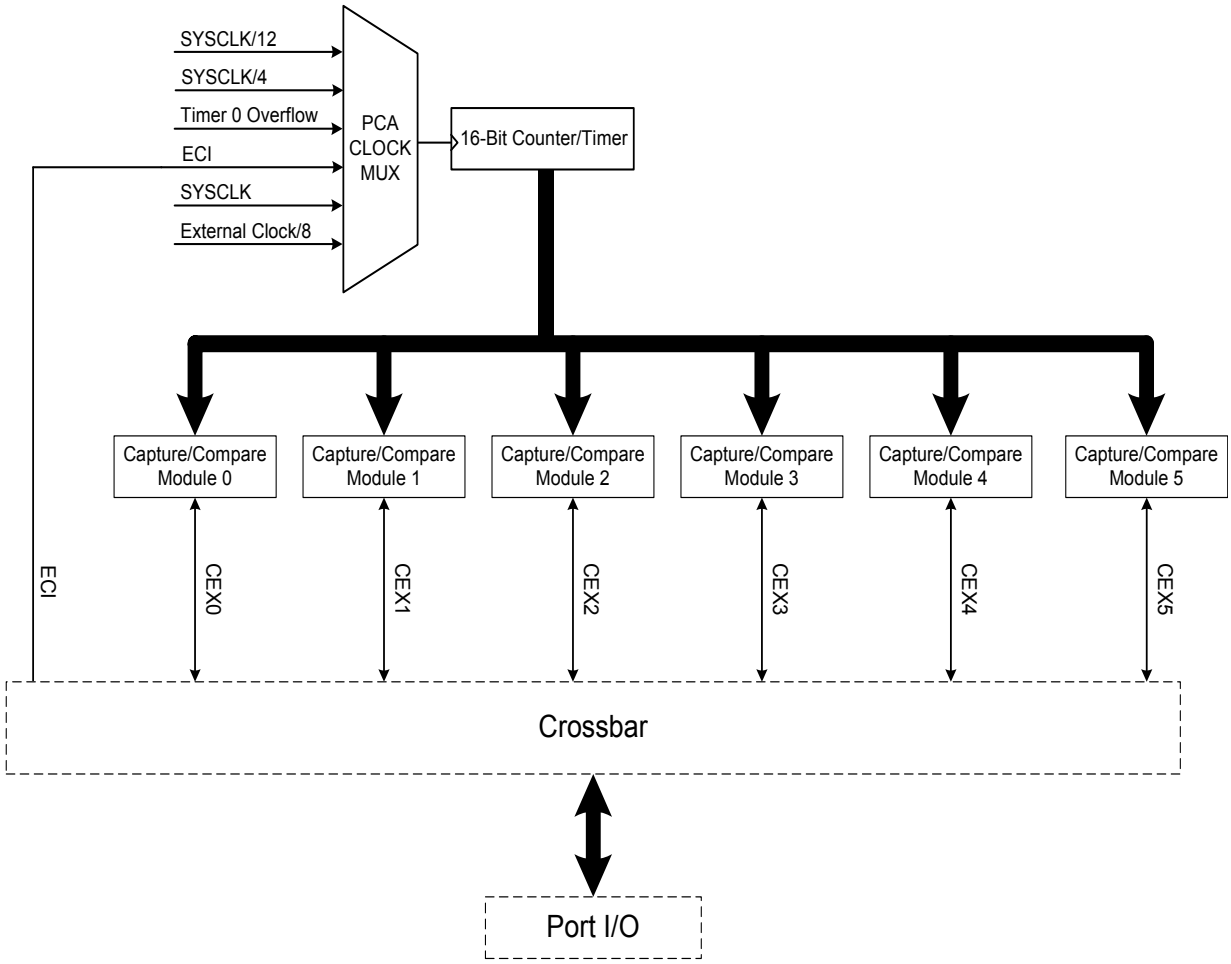


Figure 22.1. PCA Block Diagram

22.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter. Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2–CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 22.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic ‘1’ and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic ‘1’ enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic ‘1’). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 22.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8*
1	1	0	RESERVED
1	1	1	RESERVED
***Note:** External clock divided by 8 is synchronized with the system clock.			

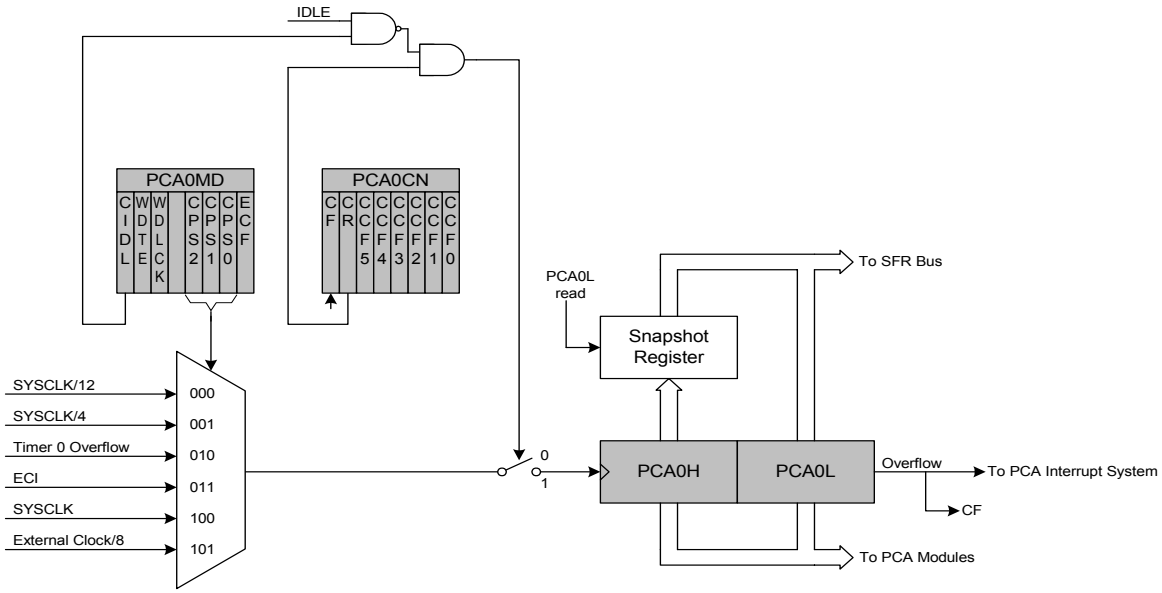


Figure 22.2. PCA Counter/Timer Block Diagram

22.2. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

Table 22.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA0 capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit (EIE1.3) to logic '1'. See Figure 22.3 for details on the PCA interrupt configuration.

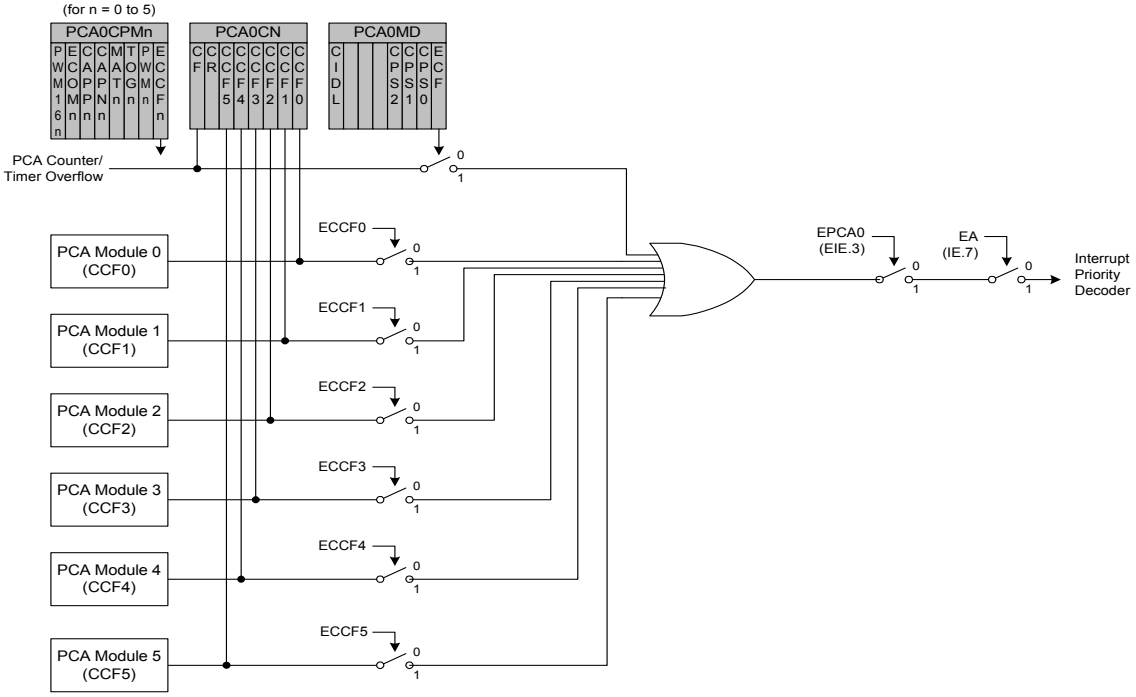


Figure 22.3. PCA Interrupt Block Diagram

Table 22.2. PCA0CPM Register Settings for PCA Capture/Compare Modules

PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Operation Mode
X	X	1	0	0	0	0	X	Capture triggered by positive edge on CEXn
X	X	0	1	0	0	0	X	Capture triggered by negative edge on CEXn
X	X	1	1	0	0	0	X	Capture triggered by transition on CEXn
X	1	0	0	1	0	0	X	Software Timer
X	1	0	0	1	1	0	X	High Speed Output
X	1	0	0	0	1	1	X	Frequency Output
0	1	0	0	0	0	1	0	8-Bit Pulse Width Modulator
1	1	0	0	0	0	1	0	16-Bit Pulse Width Modulator

X = Don't Care

22.2.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes PCA0 to capture the value of the PCA0 counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic '1' and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic '1', then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

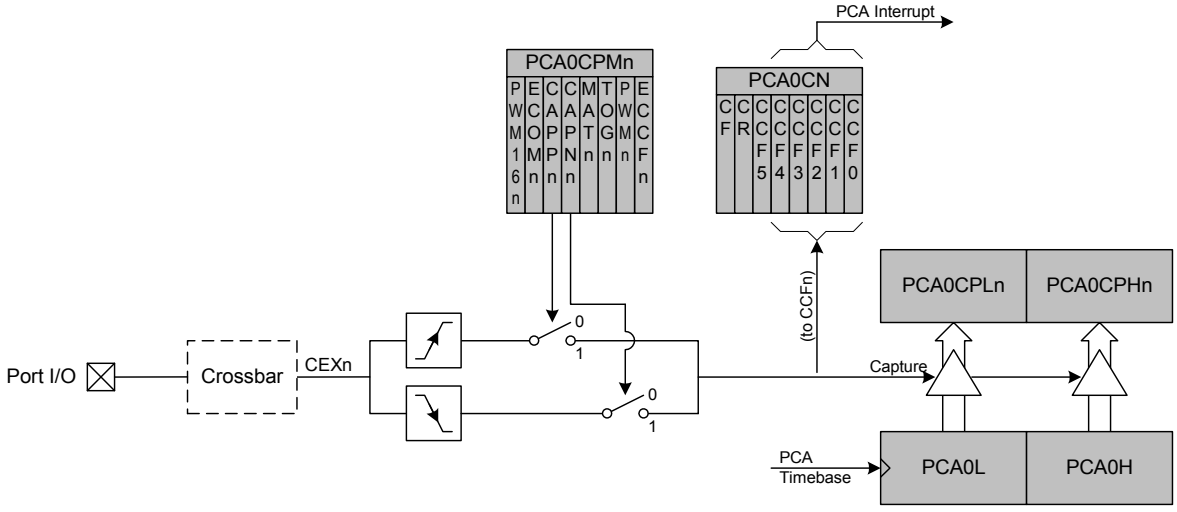


Figure 22.4. PCA Capture Mode Diagram

Note: The signal at CEXn must be high or low for at least 2 system clock cycles to be recognized by the hardware.

22.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA0 counter/timer is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic ‘1’ and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to ‘0’; writing to PCA0CPHn sets ECOMn to ‘1’.

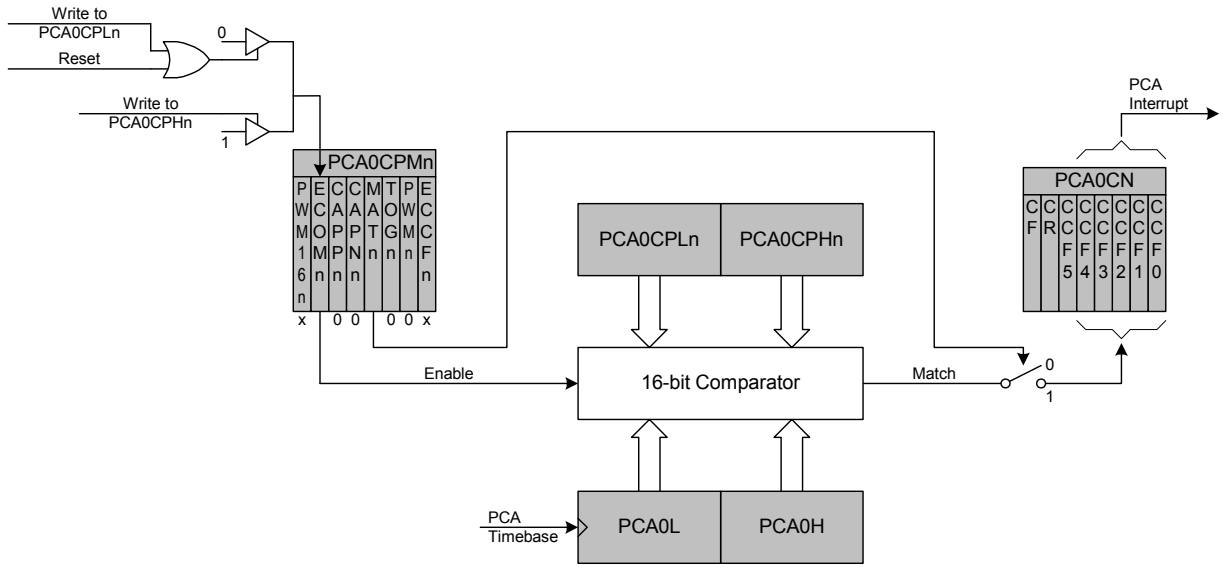


Figure 22.5. PCA Software Timer Mode Diagram

22.2.3. High Speed Output Mode

In High Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn) Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/ Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

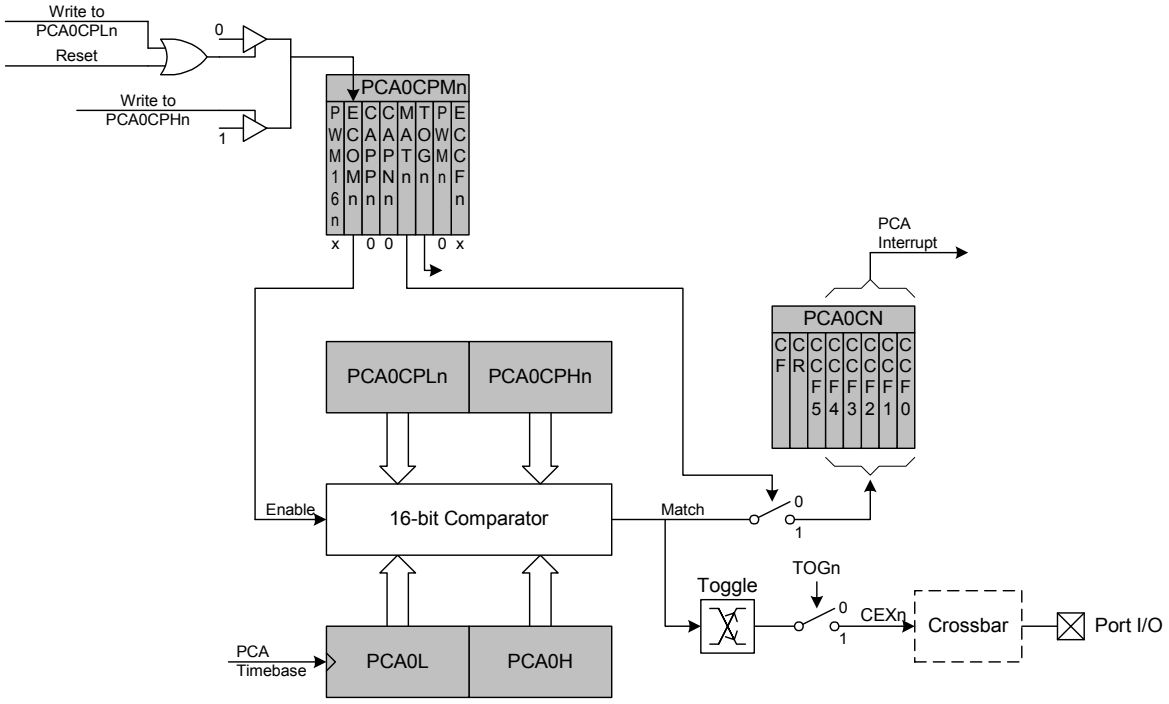


Figure 22.6. PCA High Speed Output Mode Diagram

Note: The initial state of the Toggle output is logic '1' and is initialized to this state when the module enters High Speed Output Mode.

22.2.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 22.1.

Equation 22.1. Square Wave Frequency Output

$$F_{sqr} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Where F_{PCA} is the frequency of the clock selected by the CPS2–0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA0 counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

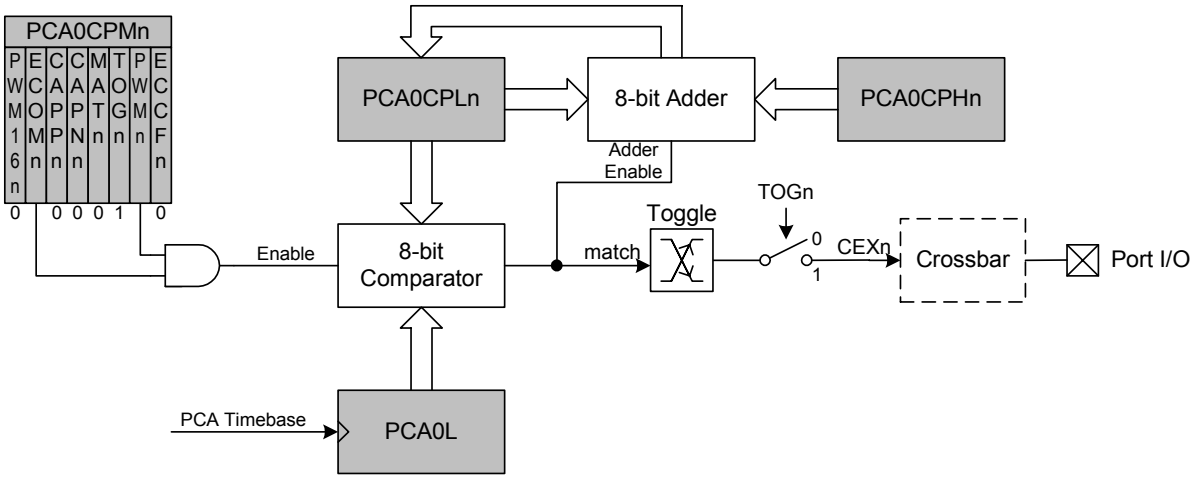


Figure 22.7. PCA Frequency Output Mode

22.2.5. 8-Bit Pulse Width Modulator Mode

Each module can be used independently to generate pulse width modulated (PWM) outputs on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA0 counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPLn capture/compare register. When the value in the low byte of the PCA0 counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be high. When the count value in PCA0L overflows, the CEXn output will be low (see Figure 22.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the counter/timer's high byte (PCA0H) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register enables 8-Bit Pulse Width Modulator mode. The duty cycle for 8-Bit PWM Mode is given by Equation 22.2.

Equation 22.2. 8-Bit PWM Duty Cycle

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Using Equation 22.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to ‘0’.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to ‘0’; writing to PCA0CPHn sets ECOMn to ‘1’.

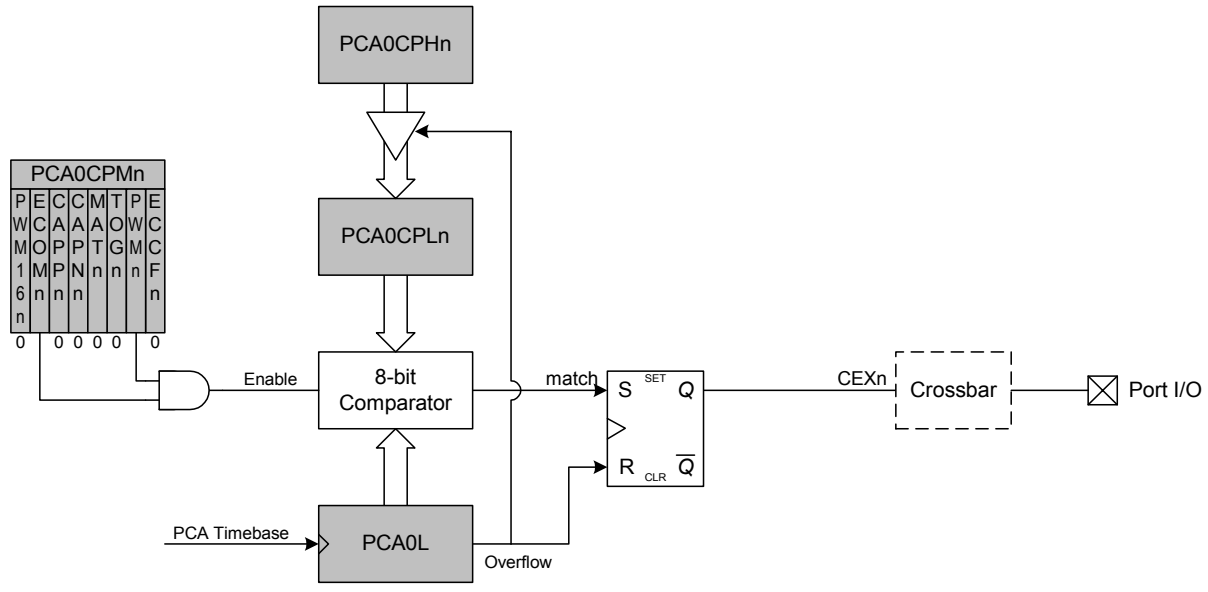


Figure 22.8. PCA 8-Bit PWM Mode Diagram

22.2.6. 16-Bit Pulse Width Modulator Mode

Each PCA0 module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA0 clocks for the low time of the PWM signal. When the PCA0 counter matches the module contents, the output on CEXn is asserted high; when the counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA0 CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, CCFn should also be set to logic '1' to enable match interrupts. The duty cycle for 16-Bit PWM Mode is given by Equation 22.3.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

Equation 22.3. 16-Bit PWM Duty Cycle

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

Using Equation 22.3, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to '0'.

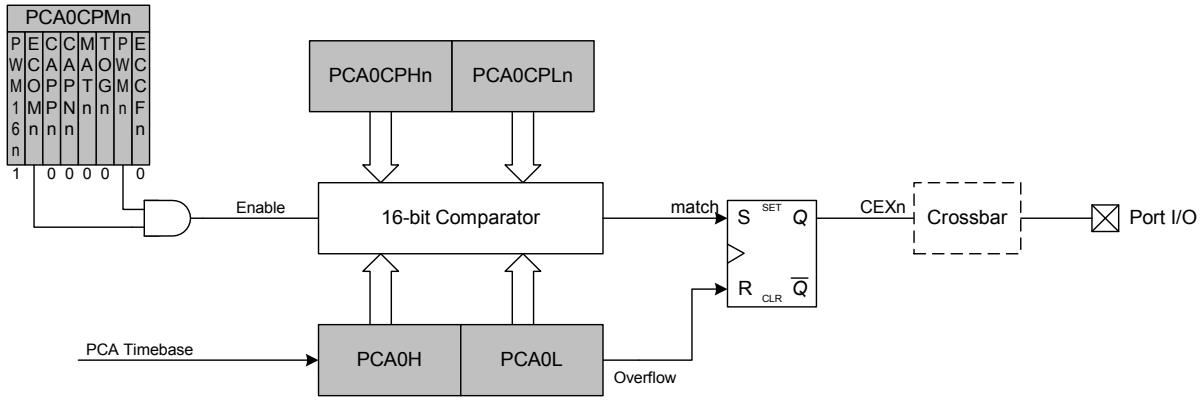


Figure 22.9. PCA 16-Bit PWM Mode

22.3. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 5. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH5) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 5 operates as a watchdog timer (WDT). The Module 5 high byte is compared to the PCA counter high byte; the Module 5 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.**

22.3.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2-CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 5 is forced into software timer mode.
- Writes to the Module 5 mode register (PCA0CPM5) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH5 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH5. Upon a PCA0CPH5 write, PCA0H plus the offset held in PCA0CPL5 is loaded into PCA0CPH5 (See Figure 22.10).

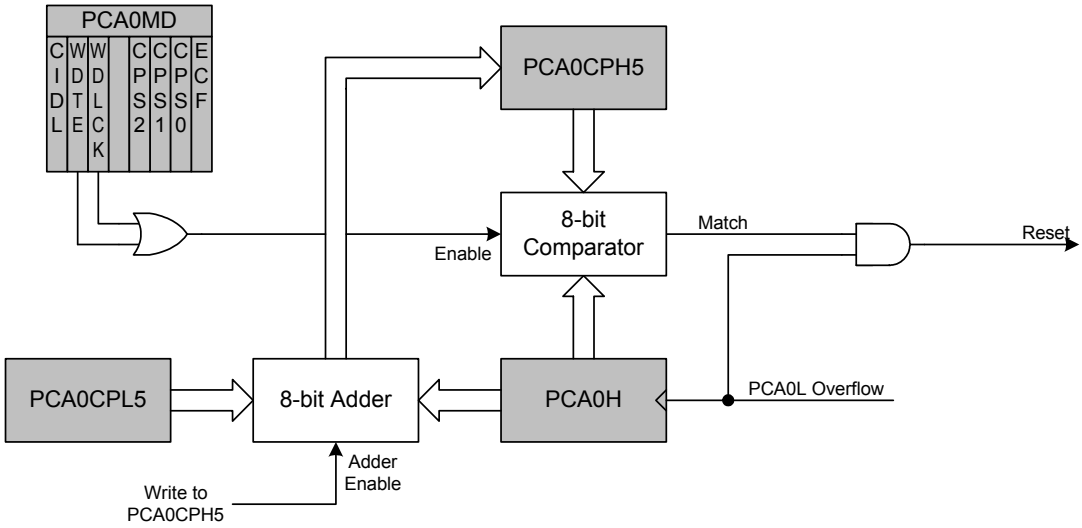


Figure 22.10. PCA Module 5 with Watchdog Timer Enabled

Note that the 8-bit offset held in PCA0CPH5 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 22.4, where PCA0L is the value of the PCA0L register at the time of the update.

Equation 22.4. Watchdog Timer Offset in PCA Clocks

$$Offset = (256 \times PCA0CPL5) + (256 - PCA0L)$$

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH5 and PCA0H. Software may force a WDT reset by writing a '1' to the CCF5 flag (PCA0CN.5) while the WDT is enabled.

22.3.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a '0' to the WDTE bit.
- Select the desired PCA clock source (with the CPS2-CPS0 bits).
- Load PCA0CPL5 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to '1'.
- Write a value to PCA0CPH5 to reload the WDT.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL5 defaults to 0x00. Using Equation 22.4, this results in a WDT timeout interval of 3072 system clock cycles. Table 22.3 lists some example timeout intervals for typical system clocks.

Table 22.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
18,432,000	255	42.7
18,432,000	128	21.5
18,432,000	32	5.5
11,059,200	255	71.1
11,059,200	128	35.8
11,059,200	32	9.2
3,062,500[2]	255	257
3,062,500[2]	128	129.5
3,062,500[2]	32	33.1
191,406	255	4109
191,406	128	2070
191,406	32	530
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:

1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal oscillator reset frequency.

22.4. Register Descriptions for PCA0

Following are detailed descriptions of the special function registers related to the operation of PCA0.

SFR Definition 22.1. PCA0CN: PCA Control

SFR Page: all pages
SFR Address: 0xD8 (bit addressable)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CF	CR	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CF: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA0 Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the CF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit 6: CR: PCA0 Counter/Timer Run Control.
This bit enables/disables the PCA0 Counter/Timer.
0: PCA0 Counter/Timer disabled.
1: PCA0 Counter/Timer enabled.

Bit 5: CCF5: PCA0 Module 5 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit 4: CCF4: PCA0 Module 4 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit 3: CCF3: PCA0 Module 3 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit 2: CCF2: PCA0 Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit 1: CCF1: PCA0 Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit 0: CCF0: PCA0 Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF interrupt is enabled, setting this bit causes the CPU to vector to the CCF interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

SFR Definition 22.2. PCA0MD: PCA0 Mode

SFR Page: all pages
SFR Address: 0xD9

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CIDL	WDTE	WDLCK	–	CPS2	CPS1	CPS0	ECF	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: CIDL: PCA0 Counter/Timer Idle Control.
Specifies PCA0 behavior when CPU is in Idle Mode.
0: PCA0 continues to function normally while the system controller is in Idle Mode.
1: PCA0 operation is suspended while the system controller is in Idle Mode.

Bit 6: WDTE: Watchdog Timer Enable
If this bit is set, PCA Module 5 is used as the watchdog timer.
0: Watchdog Timer disabled.
1: PCA Module 5 enabled as Watchdog Timer.

Bit 5: WDLCK: Watchdog Timer Lock
This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset.
0: Watchdog Timer Enable unlocked.
1: Watchdog Timer Enable locked.

Bit 4: UNUSED. Read = 0b, Write = don't care.

Bits 3–1: CPS2-CPS0: PCA0 Counter/Timer Pulse Select.
These bits select the timebase source for the PCA0 counter

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External clock divided by 8 (synchronized with system clock)
1	1	0	Reserved
1	1	1	Reserved
Note: External clock divided by 8 is synchronized with the system clock.			

Bit 0: ECF: PCA Counter/Timer Overflow Interrupt Enable.
This bit sets the masking of the PCA0 Counter/Timer Overflow (CF) interrupt.
0: Disable the CF interrupt.
1: Enable a PCA0 Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

Note: When the WDTE bit is set to '1', the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.

SFR Definition 22.3. PCA0CPMn: PCA0 Capture/Compare Mode

SFR Page: PCA0CPM0: all pages, PCA0CPM1: all pages, PCA0CPM2: all pages, PCA0CPM3: all pages, PCA0CPM4: all pages, PCA0CPM5: all pages

SFR Address: PCA0CPM0: 0xDA, PCA0CPM1: 0xDB, PCA0CPM2: 0xDC, PCA0CPM3: 0xDD, PCA0CPM4: 0xDE, PCA0CPM5: 0xDF

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bit 7: PWM16n: 16-bit Pulse Width Modulation Enable
This bit selects 16-bit mode when Pulse Width Modulation mode is enabled (PWMn = 1).
0: 8-bit PWM selected.
1: 16-bit PWM selected.

Bit 6: ECOMn: Comparator Function Enable.
This bit enables/disables the comparator function for PCA0 module n.
0: Disabled.
1: Enabled.

Bit 5: CAPPn: Capture Positive Function Enable.
This bit enables/disables the positive edge capture for PCA0 module n.
0: Disabled.
1: Enabled.

Bit 4: CAPNn: Capture Negative Function Enable.
This bit enables/disables the negative edge capture for PCA0 module n.
0: Disabled.
1: Enabled.

Bit 3: MATn: Match Function Enable.
This bit enables/disables the match function for PCA0 module n. When enabled, matches of the PCA0 counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic ‘1’.
0: Disabled.
1: Enabled.

Bit 2: TOGn: Toggle Function Enable.
This bit enables/disables the toggle function for PCA0 module n. When enabled, matches of the PCA0 counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic ‘1’, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit 1: PWMn: Pulse Width Modulation Mode Enable.
This bit enables/disables the PWM function for PCA0 module n. When enabled, a pulse width modulated signal is output on the CEXn pin. 8-bit PWM is used if PWM16n is logic ‘0’; 16-bit mode is used if PWM16n logic ‘1’. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit 0: ECCFn: Capture/Compare Flag Interrupt Enable.
This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt.
0: Disable CCFn interrupts.
1: Enable a Capture/Compare Flag interrupt request when CCFn is set.

SFR Definition 22.4. PCA0L: PCA0 Counter/Timer Low Byte

SFR Page: all pages
SFR Address: 0xF9

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	

Bits 7–0: PCA0L: PCA0 Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA0 Counter/Timer.

SFR Definition 22.5. PCA0H: PCA0 Counter/Timer High Byte

SFR Page: all pages
SFR Address: 0xFA

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	

Bits 7–0: PCA0H: PCA0 Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA0 Counter/Timer.

SFR Definition 22.6. PCA0CPLn: PCA0 Capture Module Low Byte

SFR Page: PCA0CPL0: all pages, PCA0CPL1: all pages, PCA0CPL2: all pages, PCA0CPL3: all pages, PCA0CPL4: all pages, PCA0CPL5: all pages
SFR Address: PCA0CPL0: 0xFB, PCA0CPL1: 0xE9, PCA0CPL2: 0xEB, PCA0CPL3: 0xED, PCA0CPL4: 0xFD, PCA0CPL5: 0xF5

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	

Bits 7–0: PCA0CPLn: PCA0 Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

SFR Definition 22.7. PCA0CPHn: PCA0 Capture Module High Byte

SFR Page:	**PCA0CPH0: all pages, PCA0CPH1: all pages, PCA0CPH2: all pages, PCA0CPH3: all pages, PCA0CPH4: all pages, PCA0CPH5: all pages**
SFR Address:	**PCA0CPH0: 0xFC, PCA0CPH1: 0xEA, PCA0CPH2: 0xEC, PCA0CPH3: 0xEE, PCA0CPH4: 0xFE, PCA0CPH5: 0xF6**

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	

Bits 7–0: PCA0CPHn: PCA0 Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

23. Revision Specific Behavior

This chapter contains behavioral differences between C8051F36x "REV A" and behavior as stated in the data sheet.

These deviations will be resolved in the next revision of the device.

23.1. Revision Identification

The Lot ID Code on the top side of the device package can be used for decoding device revision information. On C8051F36x devices the revision letter is the first letter of the Lot ID Code.

Figures 23.1, 23.2, and 23.3 show how to find the Lot ID Code on the top side of the device package.

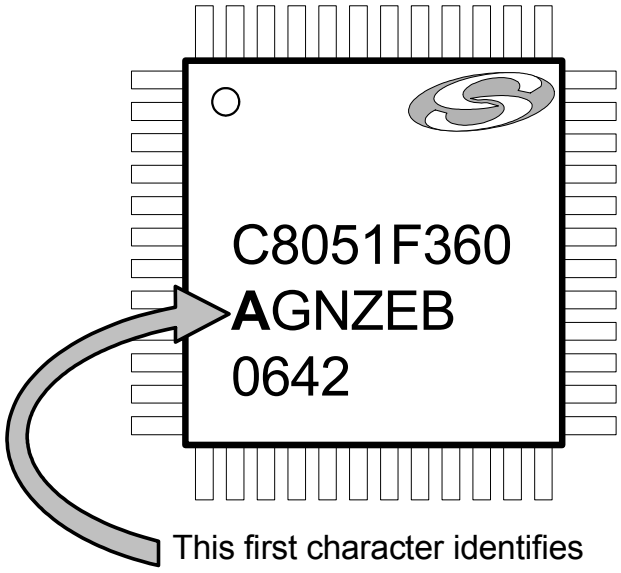


Figure 23.1. Device Package - TQFP 48

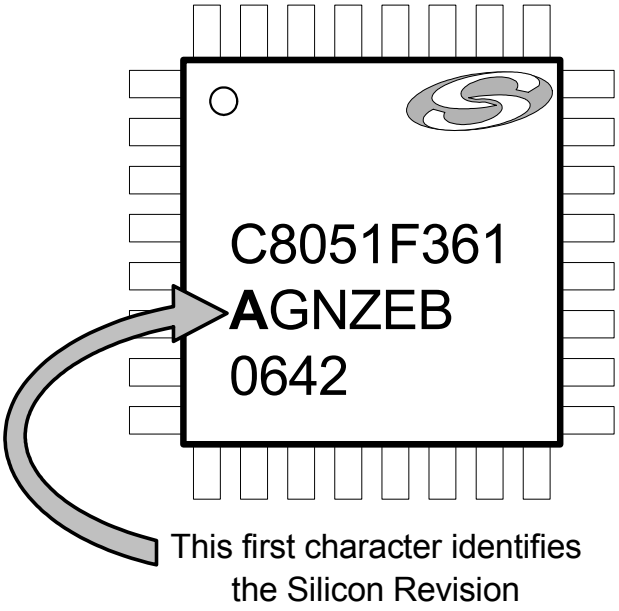


Figure 23.2. Device Package - LQFP 32

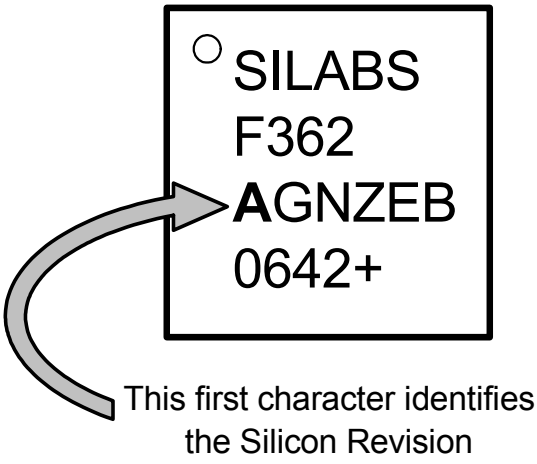


Figure 23.3. Device Package - QFN 28

23.2. C2D Port Pin Requirements

Problem

The C2D debugging port pin (shared with P4.6 for C8051F360/3 and P3.0 for C8051F361/2/4/5/6/7/8/9) behaves differently on "REV A" devices than specified in the data sheet.

On "REV A" devices, a C2D port pin that is pulled low by the associated port pin driver will disrupt debugging capability. In order to communicate with the device through the C2 interface, the value in the port latch associated C2D port pin must be '1'.

Workaround

To workaround this problem, add a strong pull-up resistor to the C2D port pin to ensure the pin will be high unless explicitly driven low. Furthermore, the port pin should be left in open-drain mode with a '1' in the appropriate port latch (PnMDOUT bit = '0', Pn bit = '1') when not in use. This will allow the debugging software to transfer data via the C2D pin as often as possible.

This behavior has been corrected on "REV B" of this device.

24. C2 Interface

C8051F36x devices include an on-chip Silicon Laboratories 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

24.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 24.1. C2ADD: C2 Address

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

Address	Description
0x00	Selects the Device ID register for Data Read instructions (DEVICEID)
0x01	Selects the Revision ID register for Data Read instructions (REVID)
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions (FPCTL)
0xBF	Selects the C2 Flash Programming Data register for Data Read/Write instructions (FPDAT)

C2 Register Definition 24.2. DEVICEID: C2 Device ID

C2 Address: 0x00

Reset Value: 00010010

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit device ID: 0x12.

C2 Register Definition 24.3. REVID: C2 Revision ID

C2 Address: 0x01

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit revision ID: 0x00 (Revision A) or 0x01 (Revision B).

C2 Register Definition 24.4. FPCTL: C2 Flash Programming Control

C2 Address: 0x02

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0 FPCTL: Flash Programming Control Register.
This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 24.5. FPDAT: C2 Flash Programming Data

C2 Address: 0xB4

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: FPDAT: C2 Flash Programming Data Register.
This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.

Code	Command
0x06	Flash Block Read
0x07	Flash Block Write
0x08	Flash Page Erase
0x03	Device Erase

24.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming functions may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (/RST) and C2D (P4.6 on C8051F360/3 devices, P3.0 on C8051F361/2/4/5/6/7/8/9 devices) pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 24.1.

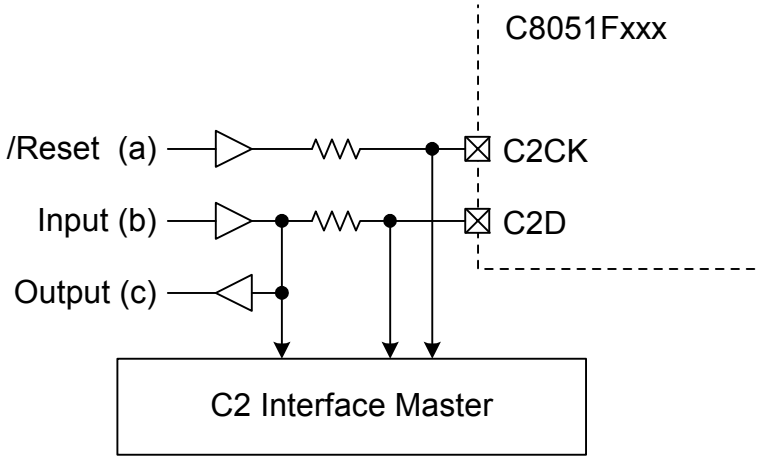


Figure 24.1. Typical C2 Pin Sharing

The configuration in Figure 24.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The /RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

DOCUMENT CHANGE LIST

Revision 0.1 to Revision 0.2

- Updated specification tables with most recently available characterization data.
- Fixed an error with the SYSCLK specification in Table 3.1, “Global Electrical Characteristics,” on page 33.
- Corrected the name of the PMAT bit in SFR Definition 10.2. IP: Interrupt Priority.
- Corrected the reset value for SFR Definition 22.2. PCA0MD: PCA0 Mode.

Revision 0.2 to Revision 1.0

- Updated specification tables with characterization data.
- Fixed Table 1.1, “Product Selection Guide,” on page 19 to reflect the correct number of Port I/O pins for the C8051F361/2/4/5.
- Updated Section “10. Interrupt Handler” on page 107.
  - Added note describing EA change behavior when followed by single cycle instruction.
- Updated SFR Definition 11.1
  - Changed the MAC0SC (MAC0CF.5) bit description to correctly refer to the MAC0SD bit.
- Updated SFR Definition 15.2.
  - Changed the EMI0CF description to properly describe the 1k XRAM boundaries.
- Added Table 16.2, “Internal Low Frequency Oscillator Electrical Characteristics,” on page 172.
- Updated SFR Definition 16.9:
  - Specified that the undefined states for PLLLP3–0 are RESERVED.
- Added Table 19.7 and Table 19.8 on page 233 for UART Baud Rates when using the PLL.
- Updated Table 22.1, “PCA Timebase Input Options,” on page 265:
  - Specified that the undefined states of CPS2–0 are RESERVED.
- Added Revision B to “Revision Specific Behavior” on page 281.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: MCUinfo@silabs.com
Internet: www.silabs.com