C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

Full Speed USB Flash MCU Family

Analog Peripherals
- **10-Bit ADC (C8051F340/1/2/3/4/5/6/7/A/B only)**
  - Up to 200 ksps
  - Built-in analog multiplexer with single-ended and differential mode
  - VREF from external pin, internal reference, or V_{DD}
  - Built-in temperature sensor
  - External conversion start input option
- **Two comparators**
- **Internal voltage reference (C8051F340/1/2/3/4/5/6/7/A/B only)**
- **Brown-out detector and POR Circuitry**

USB Function Controller
- USB specification 2.0 compliant
- Full speed (12 Mbps) or low speed (1.5 Mbps) operation
- Integrated clock recovery; no external crystal required for full speed or low speed
- Supports eight flexible endpoints
- 1 kB USB buffer memory
- Integrated transceiver; no external resistors required

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

Voltage Supply Input: 2.7 to 5.25 V
- Voltages from 3.6 to 5.25 V supported using On-Chip Voltage Regulator

HIgh Speed 8051 µC Core
- Pipelined instruction architecture; executes 70% of Instructions in 1 or 2 system clocks
- 48 MIPS and 25 MIPS versions available.
- Expanded interrupt handler

Memory
- 4352 or 2304 Bytes RAM
- 64 or 32 kB Flash; In-system programmable in 512-byte sectors

Digital Peripherals
- 40/25 Port I/O; All 5 V tolerant with high sink current
- Hardware enhanced SPI™, SMBus™, and one or two enhanced UART serial ports
- Four general purpose 16-bit counter/timers
- 16-bit programmable counter array (PCA) with five capture/compare modules
- External Memory Interface (EMIF)

Clock Sources
- Internal Oscillator: ±0.25% accuracy with clock recovery enabled. Supports all USB and UART modes
- External Oscillator: Crystal, RC, C, or clock (1 or 2 Pin modes)
- Low Frequency (80 kHz) Internal Oscillator
- Can switch between clock sources on-the-fly

Packages
- 48-pin TQFP (C8051F340/1/4/5/8/C)
- 32-pin LQFP (C8051F342/3/6/7/9/A/B/D)
- 5x5 mm 32-pin QFN (C8051F342/3/6/7/9/A/B)

Temperature Range: –40 to +85 °C

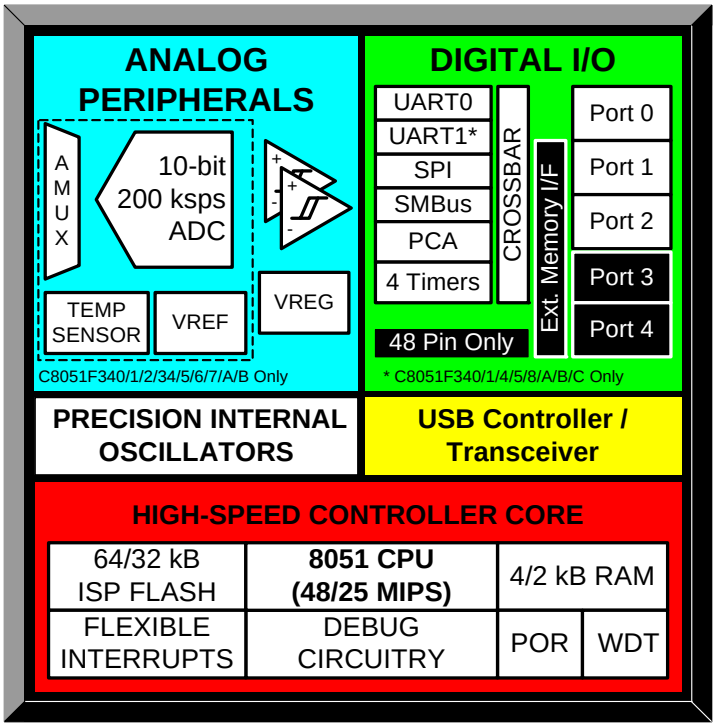


Table of Contents

List of Figures

List of Tables

List of Registers

1. System Overview

C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D 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 48 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- Universal Serial Bus (USB) Function Controller with eight flexible endpoint pipes, integrated transceiver, and 1 kB FIFO RAM
- Supply Voltage Regulator
- True 10-bit 200 ksps differential / single-ended ADC with analog multiplexer
- On-chip Voltage Reference and Temperature Sensor
- On-chip Voltage Comparators (2)
- Precision internal calibrated 12 MHz internal oscillator and 4x clock multiplier
- Internal low-frequency oscillator for additional power savings
- Up to 64 kB of on-chip Flash memory
- Up to 4352 Bytes of on-chip RAM (256 + 4 kB)
- External Memory Interface (EMIF) available on 48-pin versions.
- SMBus/I2C, up to 2 UARTs, and Enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- Programmable Counter/Timer Array (PCA) with five capture/compare modules and Watchdog Timer function
- On-chip Power-On Reset, V_{DD} Monitor, and Missing Clock Detector
- Up to 40 Port I/O (5 V tolerant)

With on-chip Power-On Reset, V_{DD} monitor, Voltage Regulator, Watchdog Timer, and clock oscillator, C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed 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 2.7–5.25 V operation over the industrial temperature range (–40 to +85 °C). For voltages above 3.6 V, the on-chip Voltage Regulator must be used. A minimum of 3.0 V is required for USB communication. The Port I/O and $\overline{RST}$ pins are tolerant of input signals up to 5 V. C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D devices are available in 48-pin TQFP, 32-pin LQFP, or 32-pin QFN packages. See Table 1.1, “Product Selection Guide,” on page 18 for feature and package choices.

Table 1.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	Flash Memory (Bytes)	RAM	Calibrated Internal Oscillator	Low Frequency Oscillator	USB with 1k Endpoint RAM	Supply Voltage Regulator	SMBus/I2C	Enhanced SPI	UARTs	Timers (16-bit)	Programmable Counter Array	Digital Port I/Os	External Memory Interface (EMIF)	10-bit 200 ksps ADC	Temperature Sensor	Voltage Reference	Analog Comparators	Package
C8051F340-GQ	48	64k	4352	✓	✓	✓	✓	✓	✓	2	4	✓	40	✓	✓	✓	✓	2	TQFP48
C8051F341-GQ	48	32k	2304	✓	✓	✓	✓	✓	✓	2	4	✓	40	✓	✓	✓	✓	2	TQFP48
C8051F342-GQ	48	64k	4352	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	LQFP32
C8051F342-GM	48	64k	4352	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	QFN32
C8051F343-GQ	48	32k	2304	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	LQFP32
C8051F343-GM	48	32k	2304	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	QFN32
C8051F344-GQ	25	64k	4352	✓	✓	✓	✓	✓	✓	2	4	✓	40	✓	✓	✓	✓	2	TQFP48
C8051F345-GQ	25	32k	2304	✓	✓	✓	✓	✓	✓	2	4	✓	40	✓	✓	✓	✓	2	TQFP48
C8051F346-GQ	25	64k	4352	✓	—	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	LQFP32
C8051F346-GM	25	64k	4352	✓	—	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	QFN32
C8051F347-GQ	25	32k	2304	✓	—	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	LQFP32
C8051F347-GM	25	32k	2304	✓	—	✓	✓	✓	✓	1	4	✓	25	—	✓	✓	✓	2	QFN32
C8051F348-GQ	25	32k	2304	✓	✓	✓	✓	✓	✓	2	4	✓	40	✓	—	—	—	2	TQFP48
C8051F349-GQ	25	32k	2304	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	—	—	—	2	LQFP32
C8051F349-GM	25	32k	2304	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	—	—	—	2	QFN32
C8051F34A-GQ	48	64k	4352	✓	✓	✓	✓	✓	✓	2	4	✓	25	—	✓	✓	✓	2	LQFP32
C8051F34A-GM	48	64k	4352	✓	✓	✓	✓	✓	✓	2	4	✓	25	—	✓	✓	✓	2	QFN32
C8051F34B-GQ	48	32k	2304	✓	✓	✓	✓	✓	✓	2	4	✓	25	—	✓	✓	✓	2	LQFP32
C8051F34B-GM	48	32k	2304	✓	✓	✓	✓	✓	✓	2	4	✓	25	—	✓	✓	✓	2	QFN32
C8051F34C-GQ	48	64k	4352	✓	✓	✓	✓	✓	✓	2	4	✓	40	✓	—	—	—	2	TQFP48
C8051F34D-GQ	48	64k	4352	✓	✓	✓	✓	✓	✓	1	4	✓	25	—	—	—	—	2	LQFP32

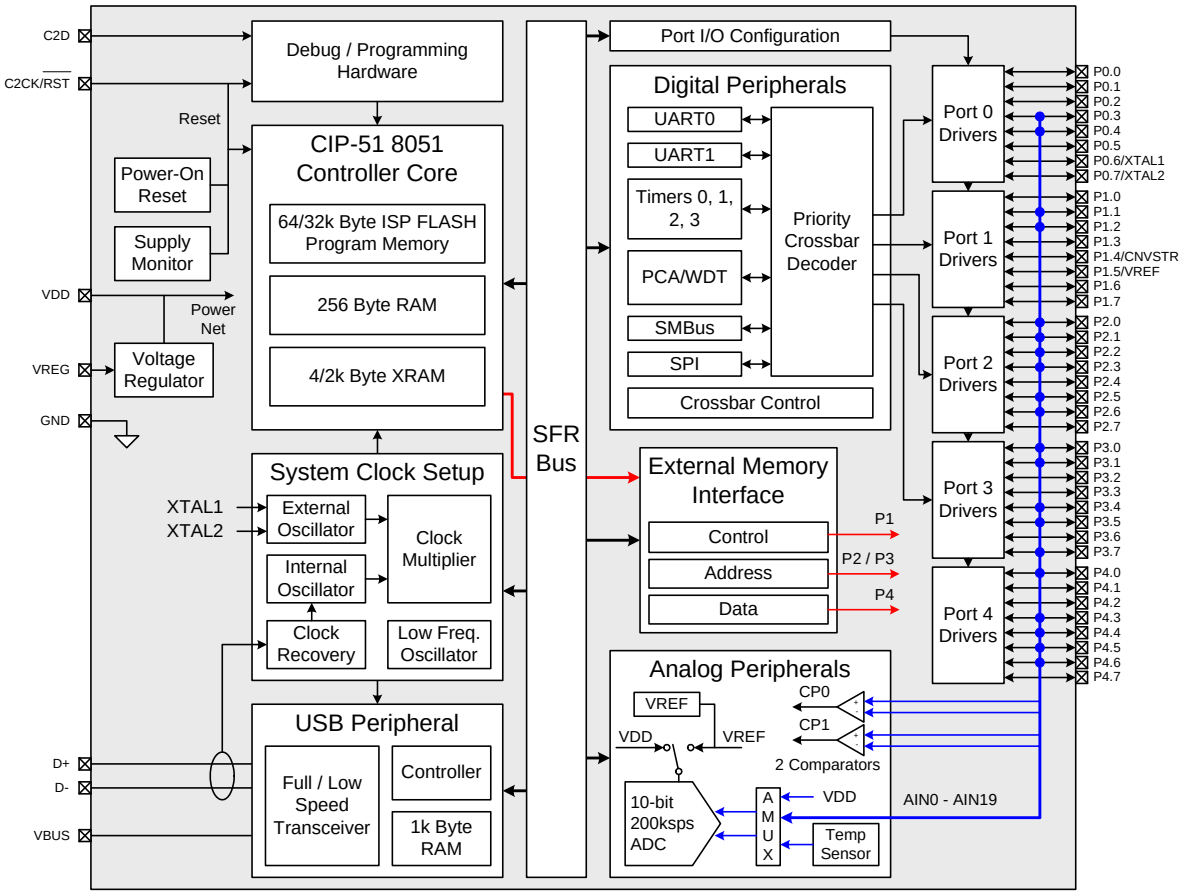
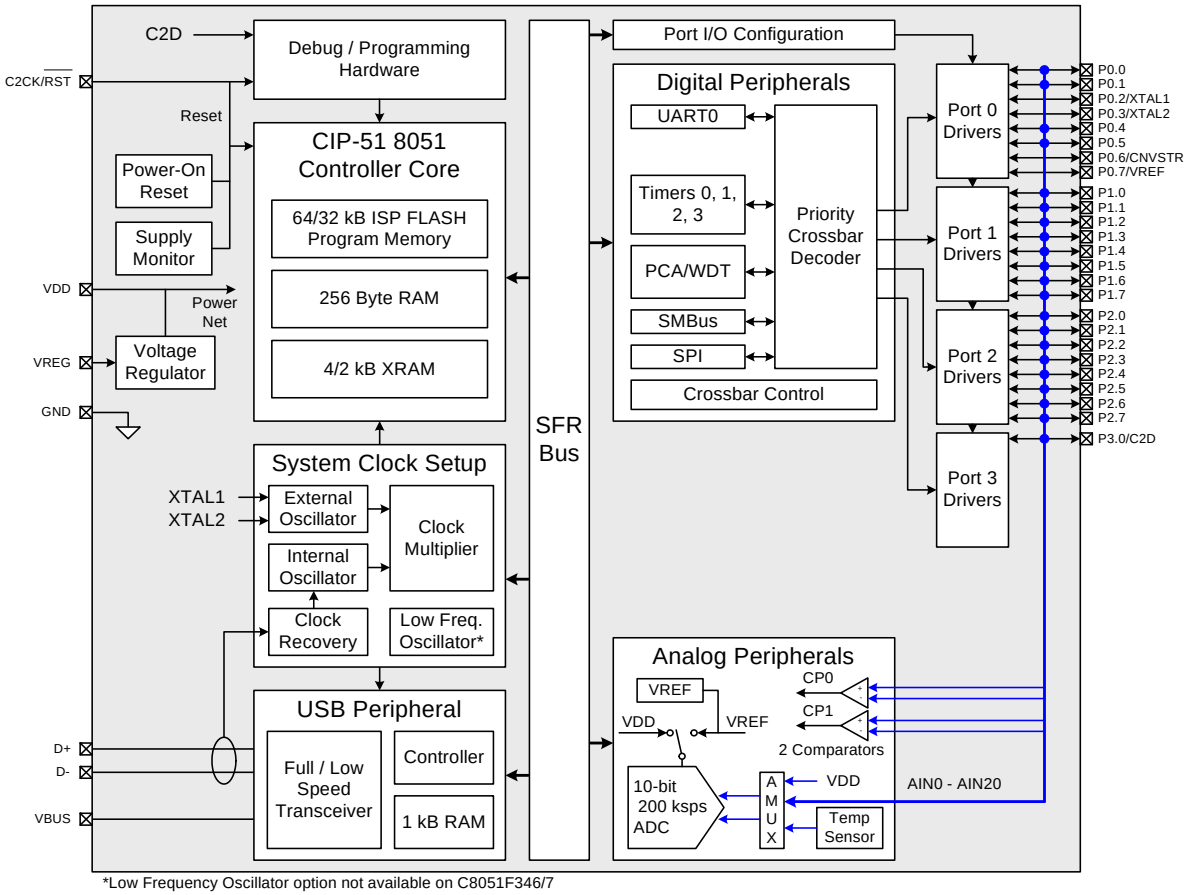


Figure 1.1. C8051F340/1/4/5 Block Diagram



*Low Frequency Oscillator option not available on C8051F346/7

Figure 1.2. C8051F342/3/6/7 Block Diagram

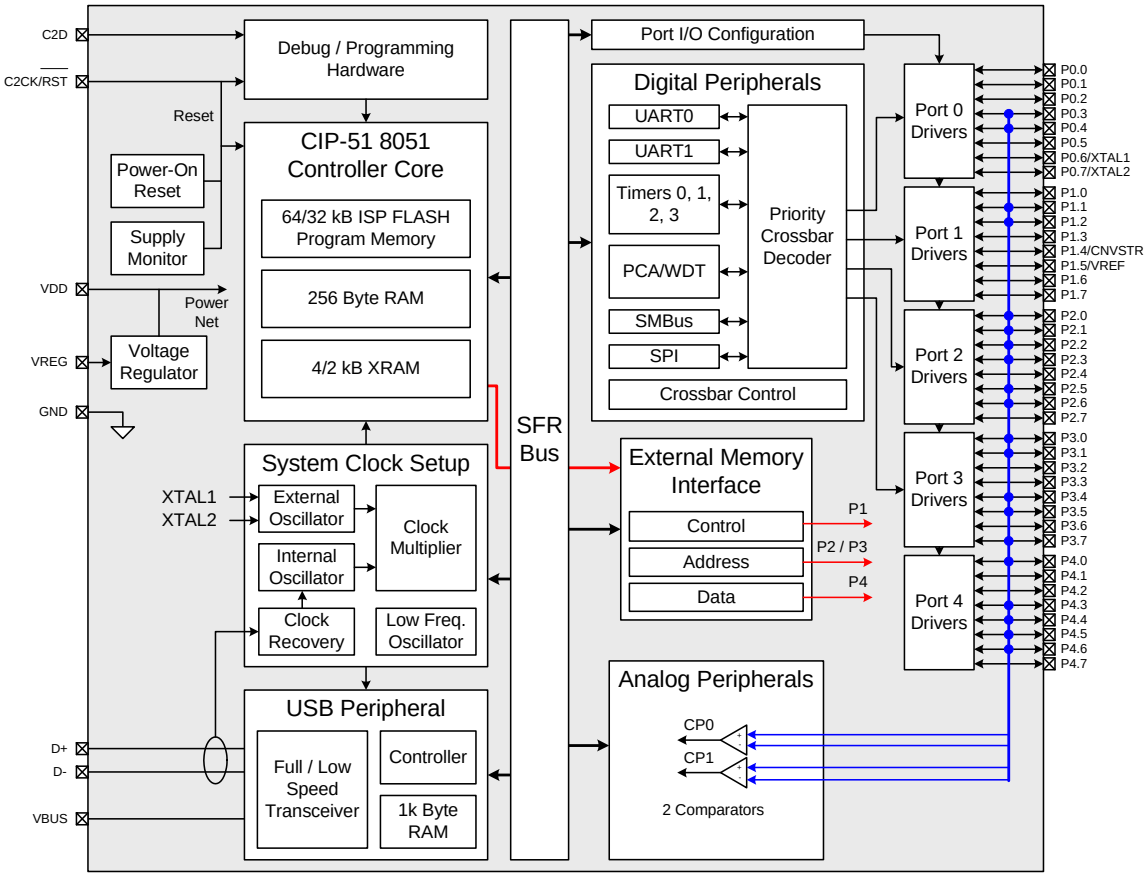


Figure 1.3. C8051F348/C Block Diagram

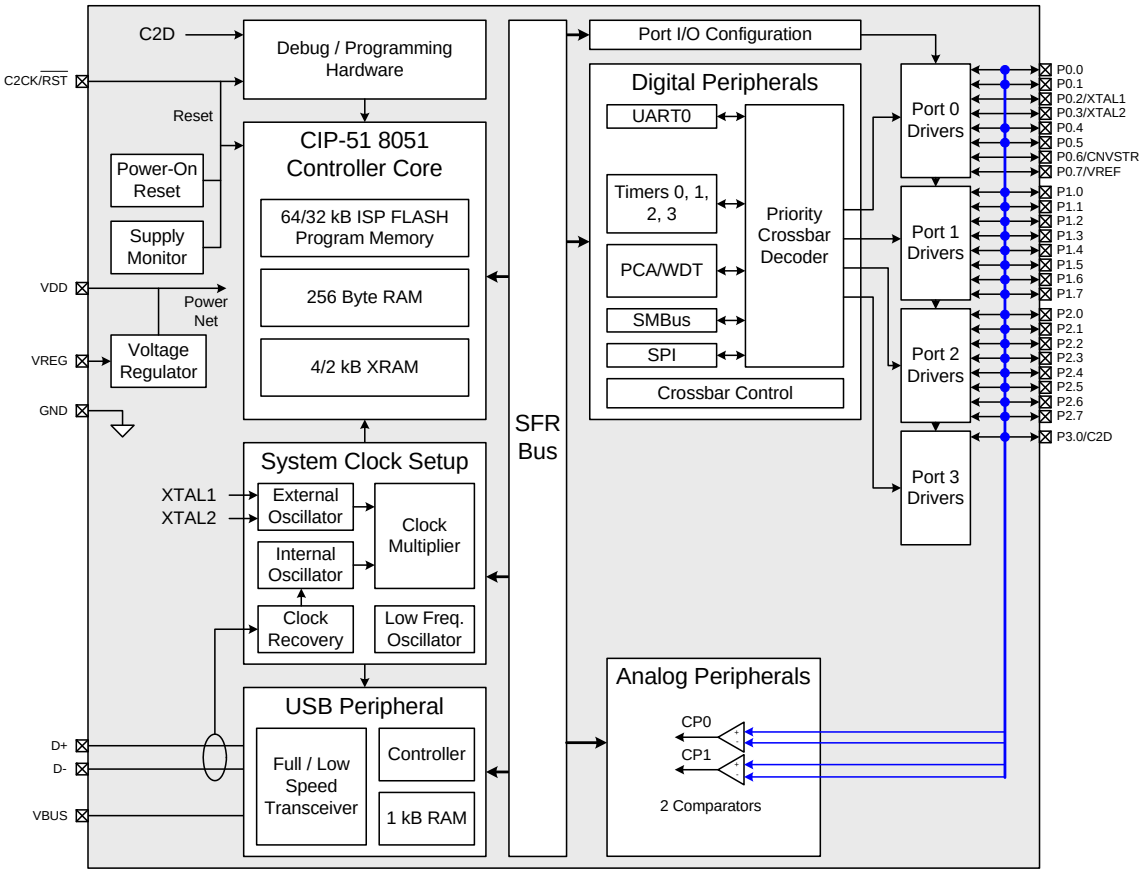


Figure 1.4. C8051F349/D Block Diagram

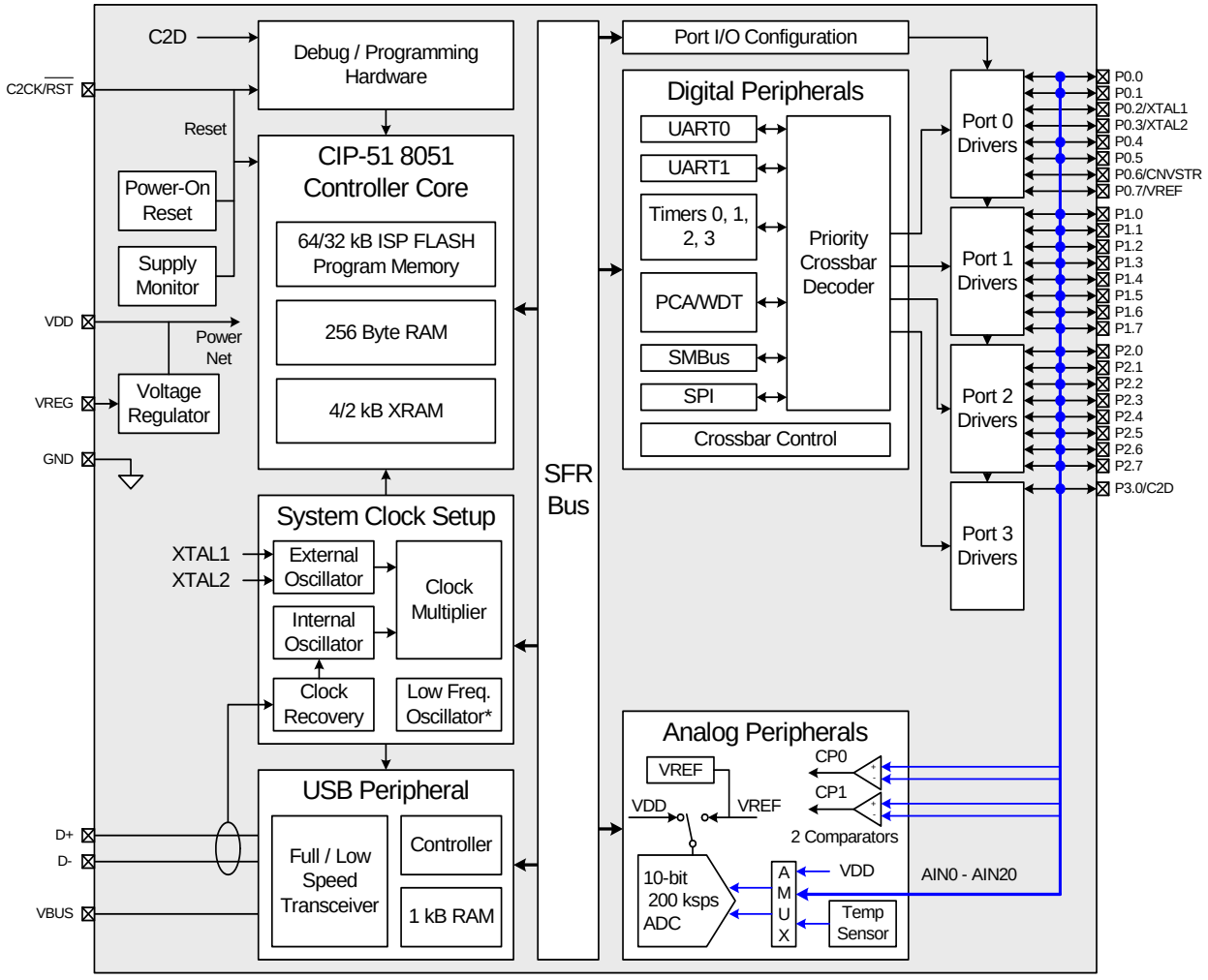


Figure 1.5. C8051F34A/B Block Diagram

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} and 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 DC Electrical Characteristics

Table 3.1. Global DC Electrical Characteristics

–40 to +85 °C, 25 MHz System Clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Digital Supply Voltage[1]		VRST	3.3	3.6	V
Digital Supply RAM Data Retention Voltage			1.5		V
SYSCLK (System Clock)[2]	C8051F340/1/2/3/A/B/C/D C8051F344/5/6/7/8/9	0 0		48 25	MHz
Specified Operating Temperature Range		–40		+85	°C
Digital Supply Current - CPU Active (Normal Mode, accessing Flash)					
I_{DD}[3]	V_{DD} = 3.3 V, SYSCLK = 48 MHz V_{DD} = 3.3 V, SYSCLK = 24 MHz V_{DD} = 3.3 V, SYSCLK = 1 MHz V_{DD} = 3.3 V, SYSCLK = 80 kHz V_{DD} = 3.6 V, SYSCLK = 48 MHz V_{DD} = 3.6 V, SYSCLK = 24 MHz		25.9 13.9 0.69 55 29.7 15.9	28.5 15.7 32.3 18	mA mA mA µA mA mA
I_{DD} Supply Sensitivity[3,4]	SYSCLK = 1 MHz, relative to V_{DD} = 3.3 V SYSCLK = 24 MHz, relative to V_{DD} = 3.3 V		47 46		%/V %/V
I_{DD} Frequency Sensitivity[3,5]	V_{DD} = 3.3 V, SYSCLK ≤ 30 MHz, T = 25 ºC V_{DD} = 3.3 V, SYSCLK > 30 MHz, T = 25 ºC V_{DD} = 3.6 V, SYSCLK ≤ 30 MHz, T = 25 ºC V_{DD} = 3.6 V, SYSCLK > 30 MHz, T = 25 ºC		0.69 0.44 0.80 0.50		mA/MHz mA/MHz mA/MHz mA/MHz
Digital Supply Current - CPU Inactive (Idle Mode, not accessing Flash)					
I_{DD}[3]	V_{DD} = 3.3 V, SYSCLK = 48 MHz V_{DD} = 3.3 V, SYSCLK = 24 MHz V_{DD} = 3.3 V, SYSCLK = 1 MHz V_{DD} = 3.3 V, SYSCLK = 80 kHz V_{DD} = 3.6 V, SYSCLK = 48 MHz V_{DD} = 3.6 V, SYSCLK = 24 MHz		16.6 8.25 0.44 35 18.6 9.26	18.75 9.34 20.9 10.5	mA mA mA µA mA mA
I_{DD} Supply Sensitivity[3,4]	SYSCLK = 1 MHz, relative to V_{DD} = 3.3 V SYSCLK = 24 MHz, relative to V_{DD} = 3.3 V		41 39		%/V %/V

Table 3.1. Global DC Electrical Characteristics (Continued)

–40 to +85 °C, 25 MHz System Clock unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
I_{DD} Frequency Sensitivity[3,6]	V_{DD} = 3.3 V, SYSCLK ≤ 1 MHz, T = 25 ºC		0.44		mA/MHz
	V_{DD} = 3.3 V, SYSCLK > 1 MHz, T = 25 ºC		0.32		mA/MHz
	V_{DD} = 3.6 V, SYSCLK ≤ 1 MHz, T = 25 ºC		0.49		mA/MHz
	V_{DD} = 3.6 V, SYSCLK > 1 MHz, T = 25 ºC		0.36		mA/MHz
Digital Supply Current (Stop Mode, shutdown)	Oscillator not running, V_{DD} monitor disabled		< 0.1		µA
Digital Supply Current for USB Module (USB Active Mode)	V_{DD} = 3.3 V, USB Clock = 48 MHz		8.69		mA
	V_{DD} = 3.6 V, USB Clock = 48 MHz		9.59		mA
Digital Supply Current for USB Module (USB Suspend Mode)	Oscillator not running V_{DD} monitor disabled		< 0.1		µA

Notes:

1. USB Requires 3.0 V Minimum Supply Voltage.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Based on device characterization of data; Not production tested.
4. Active and Inactive I_{DD} at voltages and frequencies other than those specified can be calculated using the I_{DD} Supply Sensitivity. For example, if the V_{DD} is 3.0 V instead of 3.3 V at 24 MHz: I_{DD} = 13.9 mA typical at 3.3 V and SYSCLK = 24 MHz. From this, I_{DD} = 13.9 mA + 0.46 x (3.0 V – 3.3 V) = 13.76 mA at 3.0 V and SYSCLK = 24 MHz.
5. I_{DD} can be estimated for frequencies ≤ 30 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} for > 30 MHz, the estimate should be the current at 24 MHz (or 48 MHz) minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 3.3 V; SYSCLK = 35 MHz, I_{DD} = 13.9 mA – (24 MHz – 35 MHz) x 0.44 mA/MHz = 18.74 mA.
6. Idle I_{DD} can be estimated for frequencies ≤ 1 MHz by 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 24 MHz (or 48 MHz) minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 3.3 V; SYSCLK = 5 MHz, Idle I_{DD} = 8.25 mA – (24 MHz – 5 MHz) x 0.32 mA/MHz = 2.17 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

Table Title	Page No.
ADC0 Electrical Characteristics	56
Voltage Reference Electrical Characteristics	58
Comparator Electrical Characteristics	68
Voltage Regulator Electrical Specifications	69
Reset Electrical Characteristics	106
Flash Electrical Characteristics	109
AC Parameters for External Memory Interface	130
Oscillator Electrical Characteristics	141
Port I/O DC Electrical Characteristics	158
USB Transceiver Electrical Characteristics	187

4. Pinout and Package Definitions

Table 4.1. Pin Definitions for the C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D

Name	Pin Numbers		Type	Description
	48-pin	32-pin		
V_{DD}	10	6	Power In Power Out	2.7–3.6 V Power Supply Voltage Input. 3.3 V Voltage Regulator Output. See **Section 8**.
GND	7	3		Ground.
$\overline{RST}$/ C2CK	13	9	D I/O 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 15 µs. See **Section 11**. Clock signal for the C2 Debug Interface.
C2D	14	—	D I/O	Bi-directional data signal for the C2 Debug Interface.
P3.0 / C2D	—	10	D I/O D I/O	Port 3.0. See **Section 15** for a complete description of Port 3. Bi-directional data signal for the C2 Debug Interface.
REGIN	11	7	Power In	5 V Regulator Input. This pin is the input to the on-chip voltage regulator.
VBUS	12	8	D In	VBUS Sense Input. This pin should be connected to the VBUS signal of a USB network. A 5 V signal on this pin indicates a USB network connection.
D+	8	4	D I/O	USB D+.
D-	9	5	D I/O	USB D–.
P0.0	6	2	D I/O or A In	Port 0.0. See **Section 15** for a complete description of Port 0.
P0.1	5	1	D I/O or A In	Port 0.1.
P0.2	4	32	D I/O or A In	Port 0.2.
P0.3	3	31	D I/O or A In	Port 0.3.
P0.4	2	30	D I/O or A In	Port 0.4.
P0.5	1	29	D I/O or A In	Port 0.5.
P0.6	48	28	D I/O or A In	Port 0.6.
P0.7	47	27	D I/O or A In	Port 0.7.

Table 4.1. Pin Definitions for the C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D (Continued)

Name	Pin Numbers		Type	Description
	48-pin	32-pin		
P1.0	46	26	D I/O or A In	Port 1.0. See **Section 15** for a complete description of Port 1.
P1.1	45	25	D I/O or A In	Port 1.1.
P1.2	44	24	D I/O or A In	Port 1.2.
P1.3	43	23	D I/O or A In	Port 1.3.
P1.4	42	22	D I/O or A In	Port 1.4.
P1.5	41	21	D I/O or A In	Port 1.5.
P1.6	40	20	D I/O or A In	Port 1.6.
P1.7	39	19	D I/O or A In	Port 1.7.
P2.0	38	18	D I/O or A In	Port 2.0. See **Section 15** for a complete description of Port 2.
P2.1	37	17	D I/O or A In	Port 2.1.
P2.2	36	16	D I/O or A In	Port 2.2.
P2.3	35	15	D I/O or A In	Port 2.3.
P2.4	34	14	D I/O or A In	Port 2.4.
P2.5	33	13	D I/O or A In	Port 2.5.
P2.6	32	12	D I/O or A In	Port 2.6.
P2.7	31	11	D I/O or A In	Port 2.7.
P3.0	30	—	D I/O or A In	Port 3.0. See **Section 15** for a complete description of Port 3.
P3.1	29	—	D I/O or A In	Port 3.1.
P3.2	28	—	D I/O or A In	Port 3.2.

Table 4.1. Pin Definitions for the C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D (Continued)

Name	Pin Numbers		Type	Description
	48-pin	32-pin		
P3.3	27	—	D I/O or A In	Port 3.3.
P3.4	26	—	D I/O or A In	Port 3.4.
P3.5	25	—	D I/O or A In	Port 3.5.
P3.6	24	—	D I/O or A In	Port 3.6.
P3.7	23	—	D I/O or A In	Port 3.7.
P4.0	22	—	D I/O or A In	Port 4.0. See **Section 15** for a complete description of Port 4.
P4.1	21	—	D I/O or A In	Port 4.1.
P4.2	20	—	D I/O or A In	Port 4.2.
P4.3	19	—	D I/O or A In	Port 4.3.
P4.4	18	—	D I/O or A In	Port 4.4.
P4.5	17	—	D I/O or A In	Port 4.5.
P4.6	16	—	D I/O or A In	Port 4.6.
P4.7	15	—	D I/O or A In	Port 4.7.

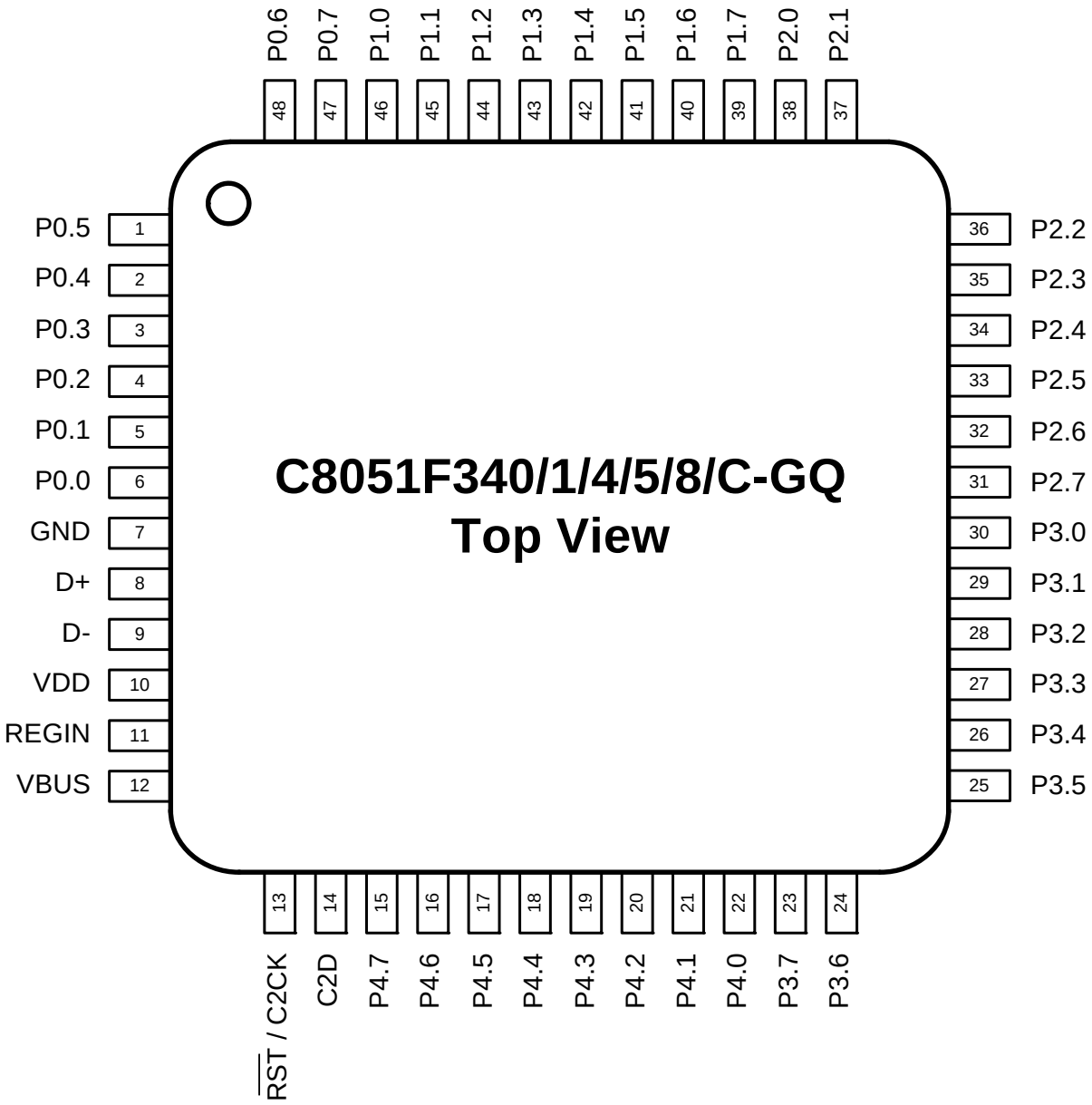


Figure 4.1. TQFP-48 Pinout Diagram (Top View)

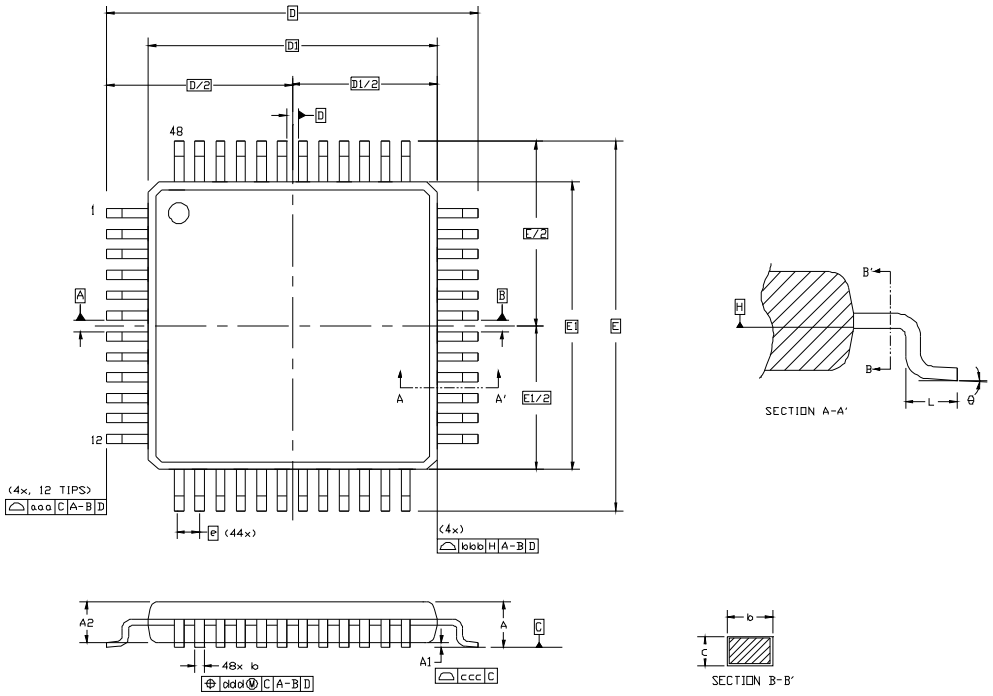


Figure 4.2. TQFP-48 Package Diagram

Table 4.2. TQFP-48 Package Dimensions

Dimension	Min	Nom	Max
A	—	—	1.20
A1	0.05	—	0.15
A2	0.95	1.00	1.05
b	0.17	0.22	0.27
c	0.09	—	0.20
D	9.00 BSC		
D1	7.00 BSC		
e	0.50 BSC		
E	9.00 BSC		
E1	7.00 BSC		
L	0.45	0.60	0.75
aaa	0.20		
bbb	0.20		
ccc	0.08		
ddd	0.08		
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. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

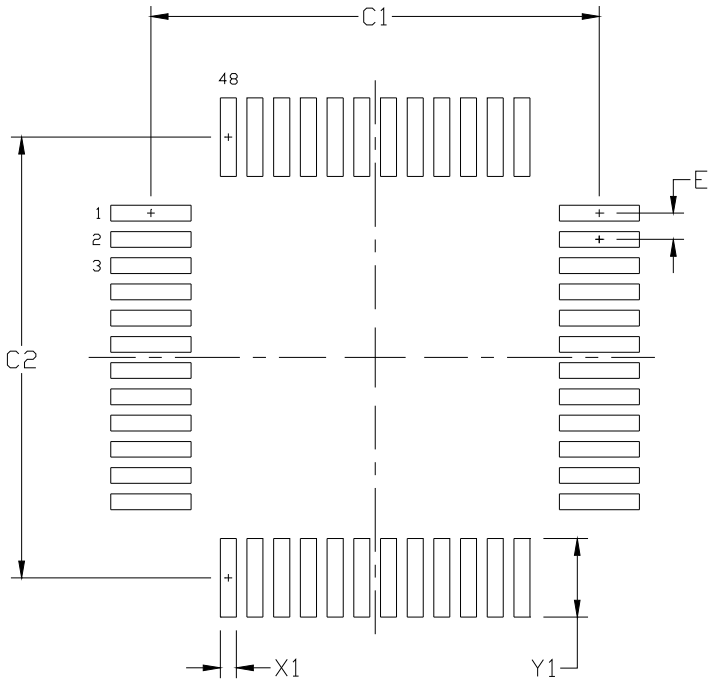


Figure 4.3. TQFP-48 Recommended PCB Land Pattern

Table 4.3. TQFP-48 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	8.30	8.40
C2	8.30	8.40
E	0.50 BSC	
X1	0.20	0.30
Y1	1.40	1.50

Notes:

General:

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

Solder Mask Design:

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

Stencil Design:

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

Card Assembly:

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

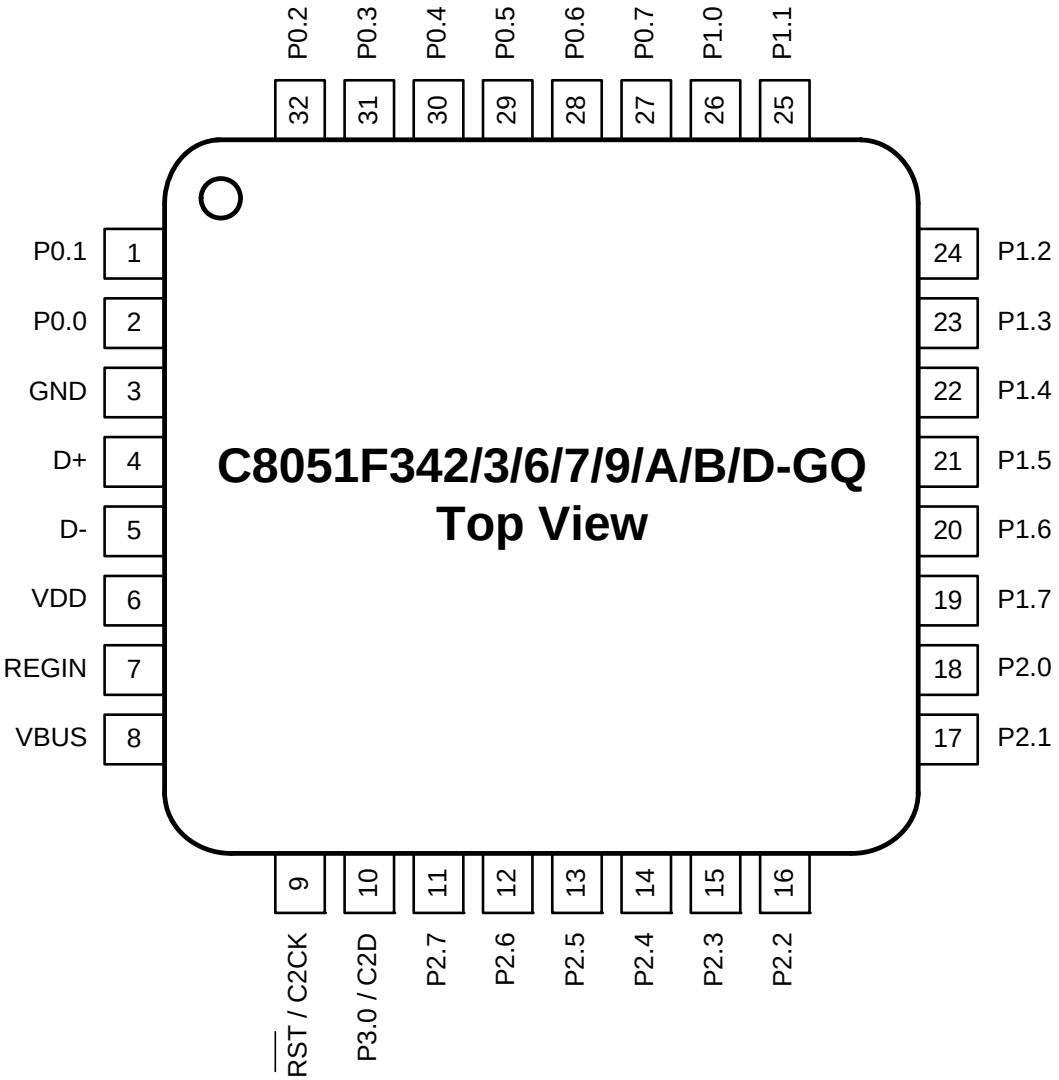


Figure 4.4. LQFP-32 Pinout Diagram (Top View)

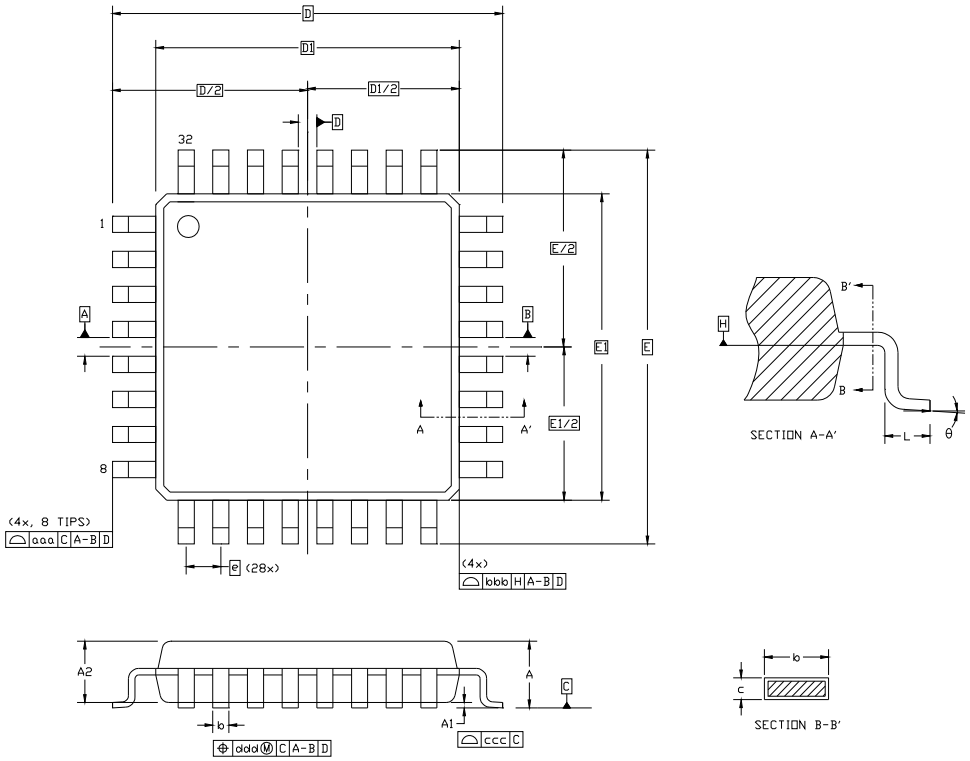


Figure 4.5. LQFP-32 Package Diagram

Table 4.4. 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		
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. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

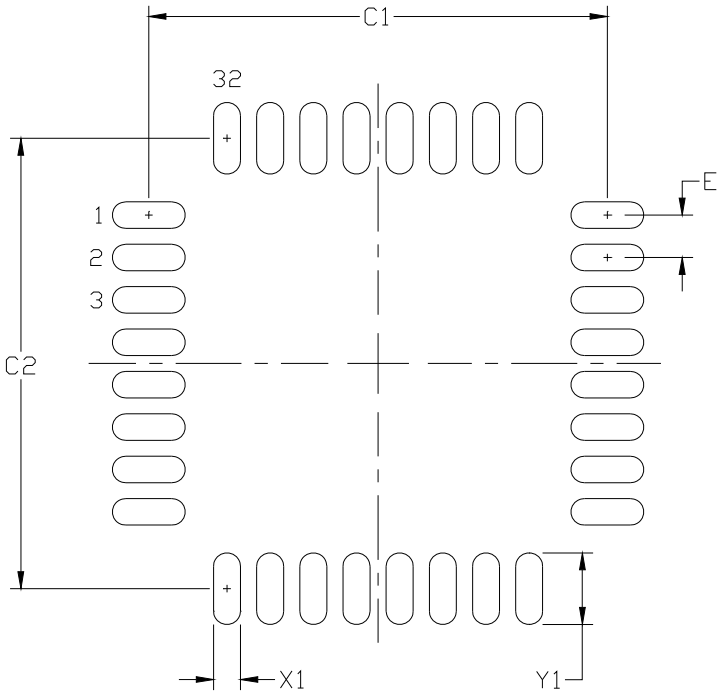


Figure 4.6. LQFP-32 Recommended PCB Land Pattern

Table 4.5. LQFP-32 PCB Land Pattern Dimensions

Dimension	Min	Max
C1	8.40	8.50
C2	8.40	8.50
E	0.80 BSC	
X1	0.40	0.50
Y1	1.25	1.35

Notes:

General:

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

Solder Mask Design:

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

Stencil Design:

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

Card Assembly:

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

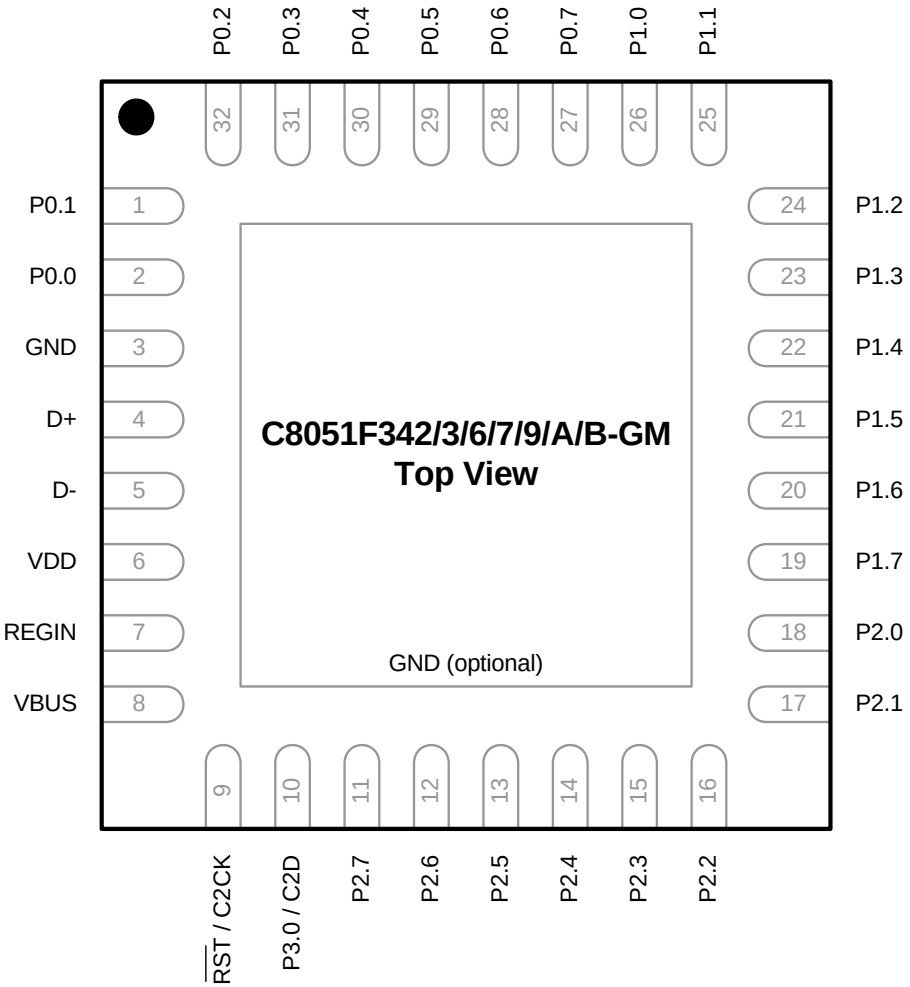


Figure 4.7. QFN-32 Pinout Diagram (Top View)

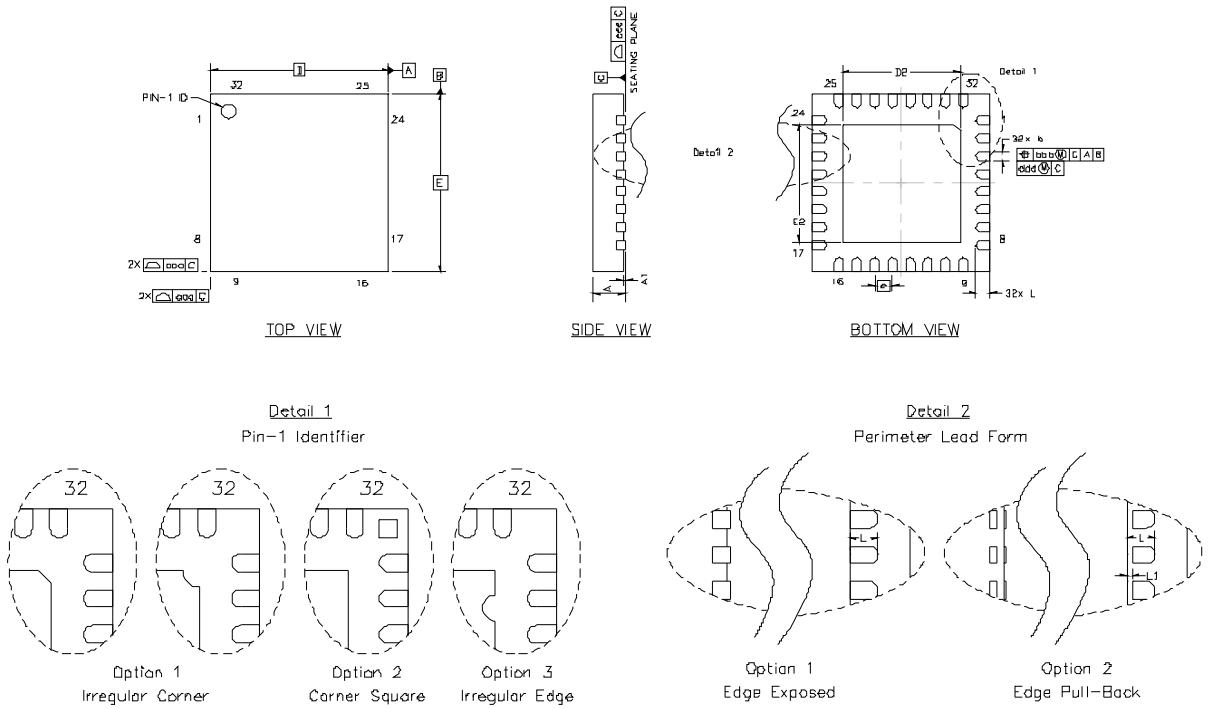


Figure 4.8. QFN-32 Package Drawing

Table 4.6. QFN-32 Package Dimensions

Dimension	Min	Nom	Max
A	0.80	0.9	1.00
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D		5.00 BSC	
D2	3.20	3.30	3.40
e		0.50 BSC	
E		5.00 BSC	
E2	3.20	3.30	3.40
L	0.30	0.40	0.50

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 the JEDEC Solid State Outline MO-220, variation VHHD except for custom features D2, E2, and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

Table 4.6. QFN-32 Package Dimensions (Continued)

Dimension	Min	Nom	Max
L1	0.00	—	0.15
aaa	—	—	0.15
bbb	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

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 the JEDEC Solid State Outline MO-220, variation VHHD except for custom features D2, E2, and L which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

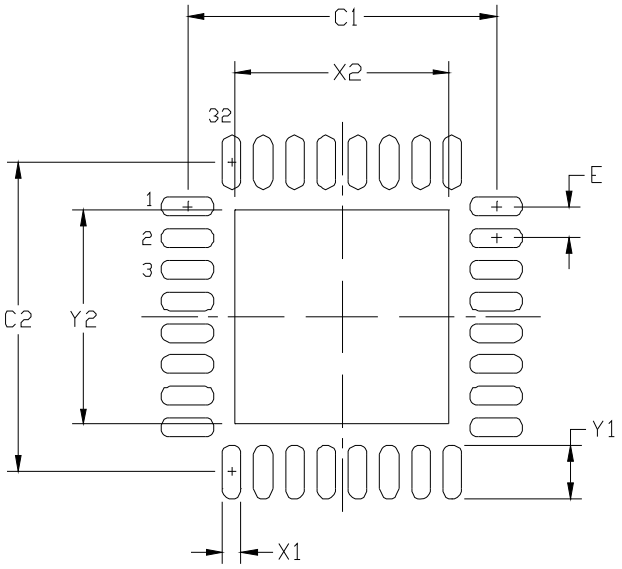


Figure 4.9. QFN-32 Recommended PCB Land Pattern

Table 4.7. QFN-32 PCB Land Pattern Dimesions

Dimension	Min	Max
C1	4.80	4.90
C2	4.80	4.90
E	0.50 BSC	
X1	0.20	0.30

Dimension	Min	Max
X2	3.20	3.40
Y1	0.75	0.85
Y2	3.20	3.40

Notes:

General:

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

Solder Mask Design:

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

Stencil Design:

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pins.
7. A 3x3 array of 1.0 mm openings on a 1.2mm pitch should be used for the center pad to assure the proper paste volume.

Card Assembly:

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

5. 10-Bit ADC (ADC0, C8051F340/1/2/3/4/5/6/7/A/B Only)

The ADC0 subsystem for the C8051F34x devices consists of two analog multiplexers (referred to collectively as AMUX0), 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 configured 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 voltages at port pins, the Temperature Sensor output, or V_{DD} with respect to a port pin, VREF, or GND. The connection options for AMUX0 are detailed in SFR Definition 5.1 and SFR Definition 5.2. 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.

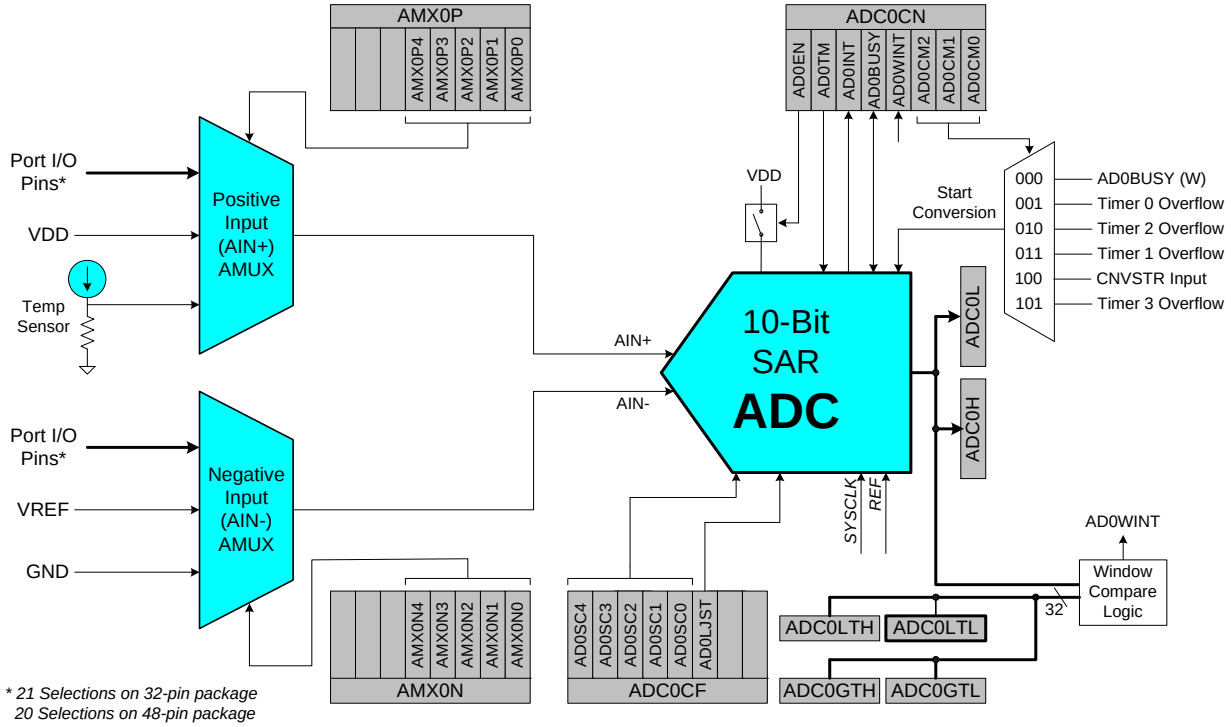


Figure 5.1. ADC0 Functional Block Diagram

5.1. Analog Multiplexer

AMUX0 selects the positive and negative inputs to the ADC. The positive input (AIN+) can be connected to individual Port pins, the on-chip temperature sensor, or the positive power supply (V_{DD}). The negative input (AIN-) can be connected to individual Port pins, VREF, or GND. **When GND is selected as the negative input, ADC0 operates in Single-ended Mode; at 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 x 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 (Single-Ended)	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 x 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 (Differential)	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). See **Section "15. Port Input/ Output" on page 142** for more Port I/O configuration details.

5.2. Temperature Sensor

The 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. Values for the Offset and Slope parameters can be found in Table 5.1.

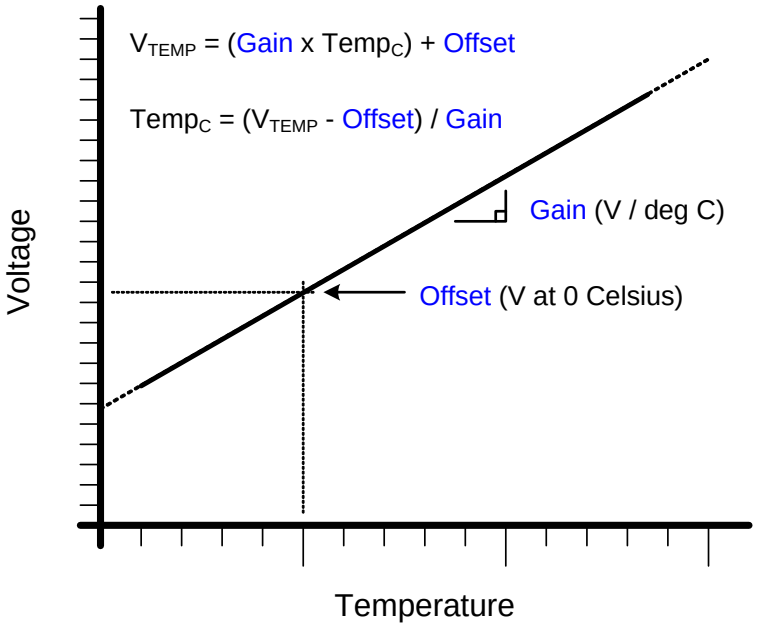


Figure 5.2. Temperature Sensor Transfer Function

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, offset and/or gain calibration is recommended. Typically a 1-point (offset) 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 characteristics, and store this value 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.**

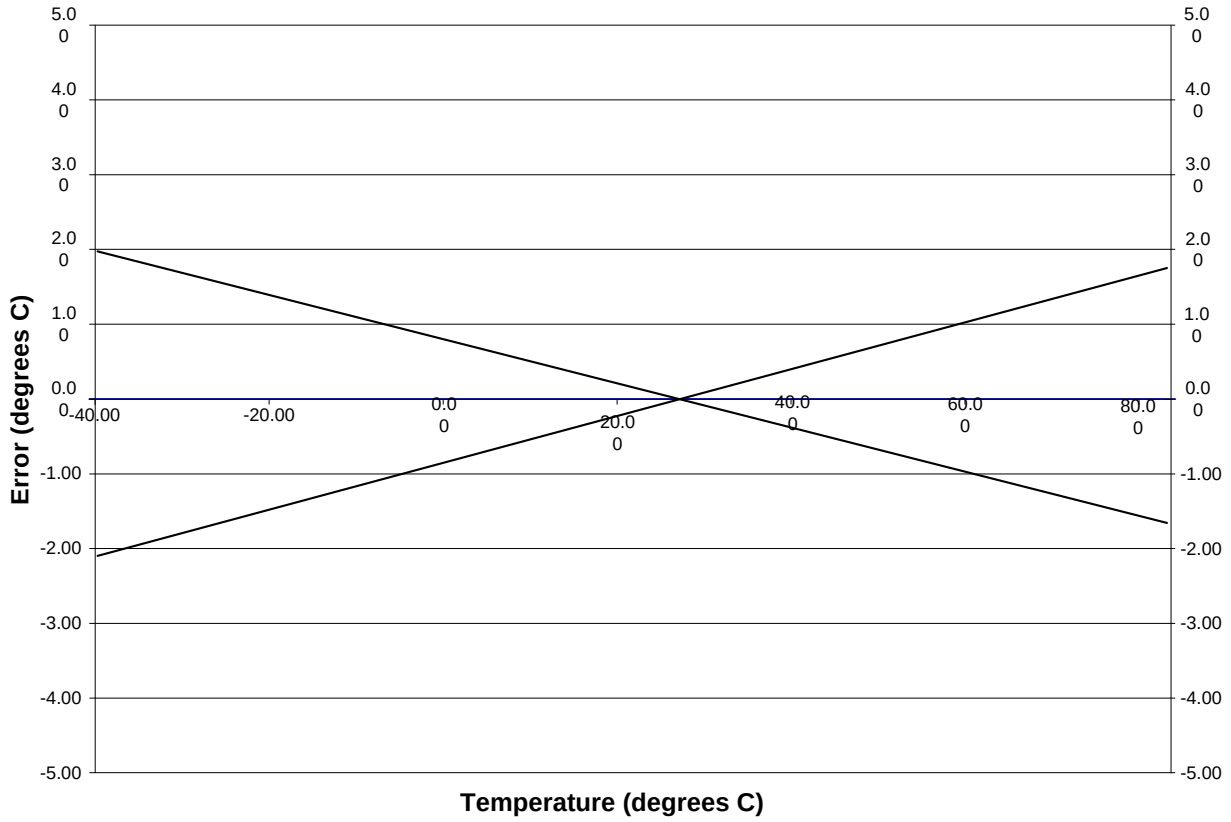


Figure 5.3. Temperature Sensor Error with 1-Point Calibration (VREF = 2.40 V)

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 ≤ AD0SC ≤ 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 235** for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as a Port pin. When the CNVSTR input is used as the ADC0 conversion source, the associated Port pin should be skipped by the Digital Crossbar. To configure the Crossbar to skip a pin, set the corresponding bit in the PnSKIP register to ‘1’. See **Section “15. Port Input/Output” on page 142** for details on Port I/O configuration.

5.3.2. Tracking Modes

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 47**.

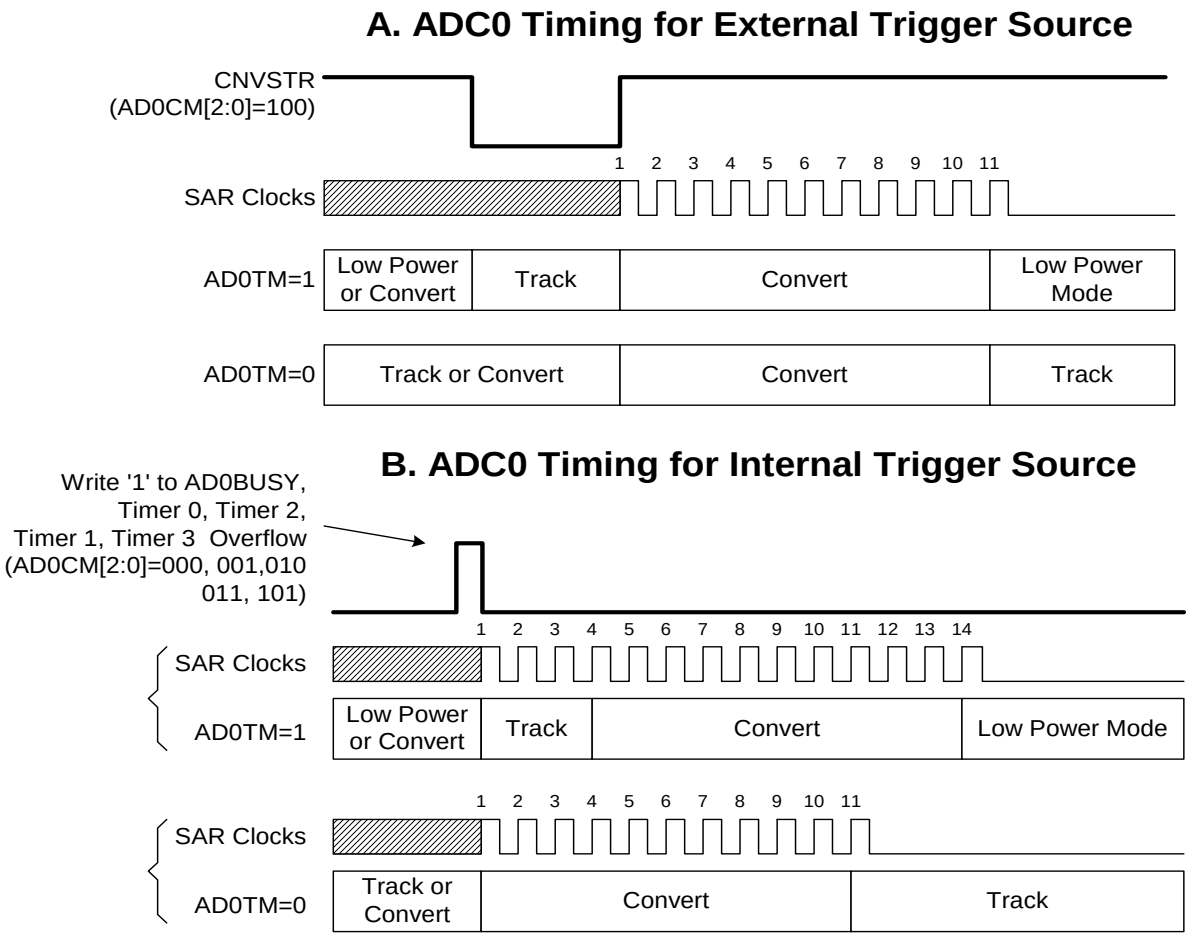


Figure 5.4. 10-Bit ADC Track and Conversion Example Timing

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. Note that 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.

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

Equation 5.1. ADC0 Settling Time Requirements

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).

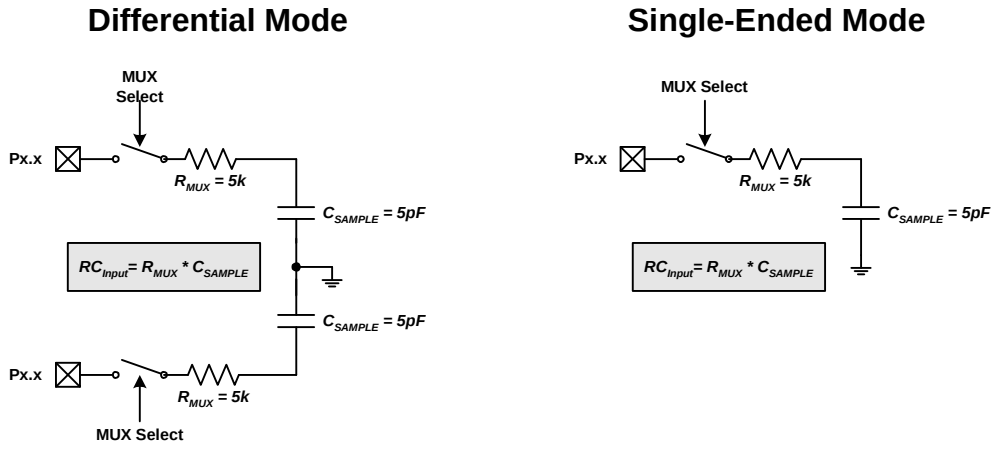


Figure 5.5. ADC0 Equivalent Input Circuits

SFR Definition 5.1. AMX0P: AMUX0 Positive Channel Select

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	SFR Address: 0xBB

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bits4–0: AMX0P4–0: AMUX0 Positive Input Selection

AMX0P4-0	ADC0 Positive Input (32-pin Package)	ADC0 Positive Input (48-pin Package)
00000	P1.0	P2.0
00001	P1.1	P2.1
00010	P1.2	P2.2
00011	P1.3	P2.3
00100	P1.4	P2.5
00101	P1.5	P2.6
00110	P1.6	P3.0
00111	P1.7	P3.1
01000	P2.0	P3.4
01001	P2.1	P3.5
01010	P2.2	P3.7
01011	P2.3	P4.0
01100	P2.4	P4.3
01101	P2.5	P4.4
01110	P2.6	P4.5
01111	P2.7	P4.6
10000	P3.0	RESERVED
10001	P0.0	P0.3
10010	P0.1	P0.4
10011	P0.4	P1.1
10100	P0.5	P1.2
10101 - 11101	RESERVED	RESERVED
11110	Temp Sensor	Temp Sensor
11111	V_{DD}	V_{DD}

SFR Definition 5.2. AMX0N: AMUX0 Negative Channel Select

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	SFR Address:
								0xBA

Bits7–5: UNUSED. Read = 000b; Write = don't care.
Bits4–0: 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 (32-pin Package)	ADC0 Negative Input (48-pin Package)
00000	P1.0	P2.0
00001	P1.1	P2.1
00010	P1.2	P2.2
00011	P1.3	P2.3
00100	P1.4	P2.5
00101	P1.5	P2.6
00110	P1.6	P3.0
00111	P1.7	P3.1
01000	P2.0	P3.4
01001	P2.1	P3.5
01010	P2.2	P3.7
01011	P2.3	P4.0
01100	P2.4	P4.3
01101	P2.5	P4.4
01110	P2.6	P4.5
01111	P2.7	P4.6
10000	P3.0	RESERVED
10001	P0.0	P0.3
10010	P0.1	P0.4
10011	P0.4	P1.1
10100	P0.5	P1.2
10101 - 11101	RESERVED	RESERVED
11110	VREF	VREF
11111	GND (Single-Ended Mode)	GND (Single-Ended Mode)

SFR Definition 5.3. ADC0CF: ADC0 Configuration

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	SFR Address: 0xBC

Bits7–3: 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.

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

Bit2: AD0LJST: ADC0 Left Justify Select.
0: Data in ADC0H:ADC0L registers are right-justified.
1: Data in ADC0H:ADC0L registers are left-justified.

Bits1–0: UNUSED. Read = 00b; Write = don't care.

SFR Definition 5.4. ADC0H: ADC0 Data Word MSB

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: 0xBE

Bits7–0: 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

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: 0xBD

Bits7–0: 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

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	SFR Address:
							(bit addressable)	0xE8

Bit7: AD0EN: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit6: 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).

Bit5: 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.

Bit4: 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

Bit3: 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.

Bits2–0: 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 conversion results 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.

The Window Detector registers must be written with the same format (left/right justified, signed/unsigned) as that of the current ADC configuration (left/right justified, single-ended/differential).

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

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	SFR Address:
								0xC4

Bits7–0: High byte of ADC0 Greater-Than Data Word.

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

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	SFR Address:
								0xC3

Bits7–0: Low byte of ADC0 Greater-Than Data Word.

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

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:
								0xC6

Bits7–0: High byte of ADC0 Less-Than Data Word.

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

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:
								0xC5

Bits7–0: 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 equivalent ADC0GT and ADC0LT register settings.

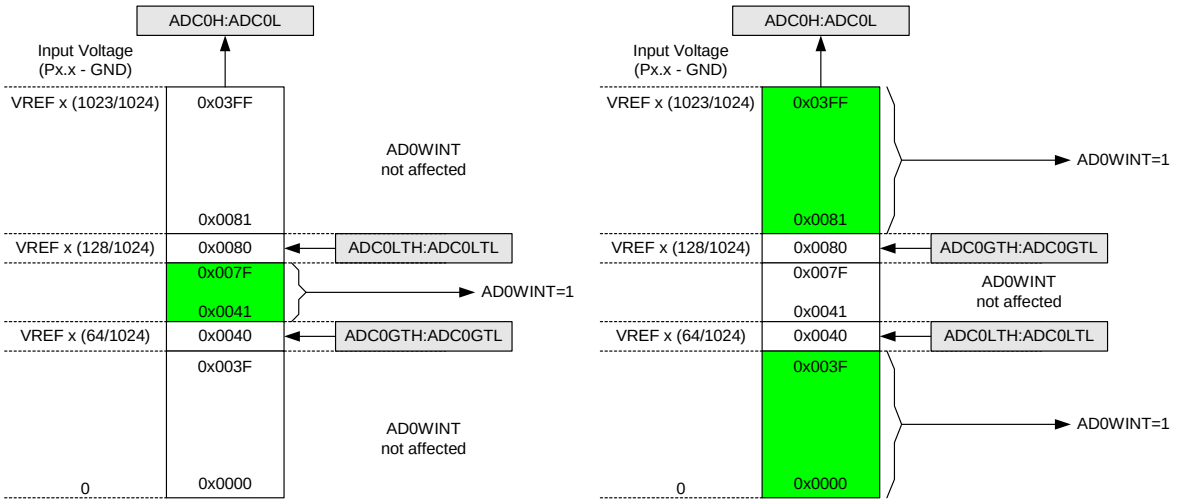


Figure 5.6. ADC Window Compare Example: Right-Justified Single-Ended Data

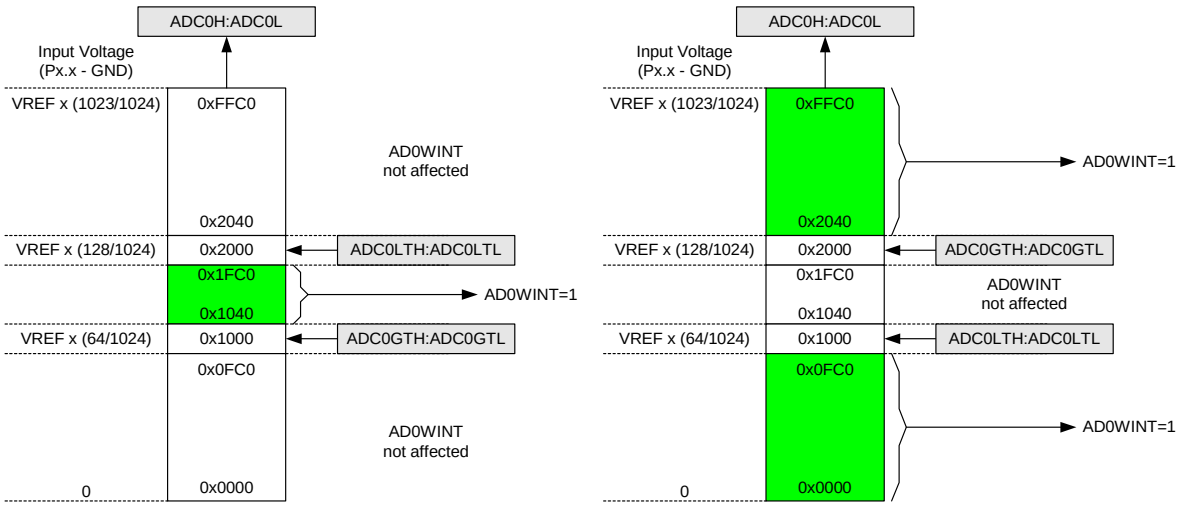


Figure 5.7. ADC Window Compare Example: Left-Justified Single-Ended Data

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 equivalent ADC0GT and ADC0LT register settings.

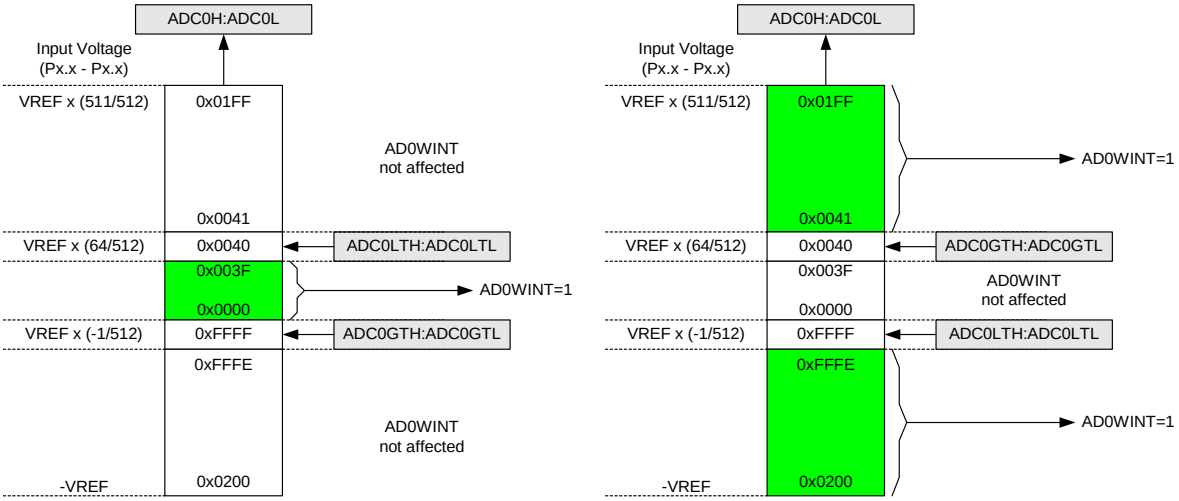


Figure 5.8. ADC Window Compare Example: Right-Justified Differential Data

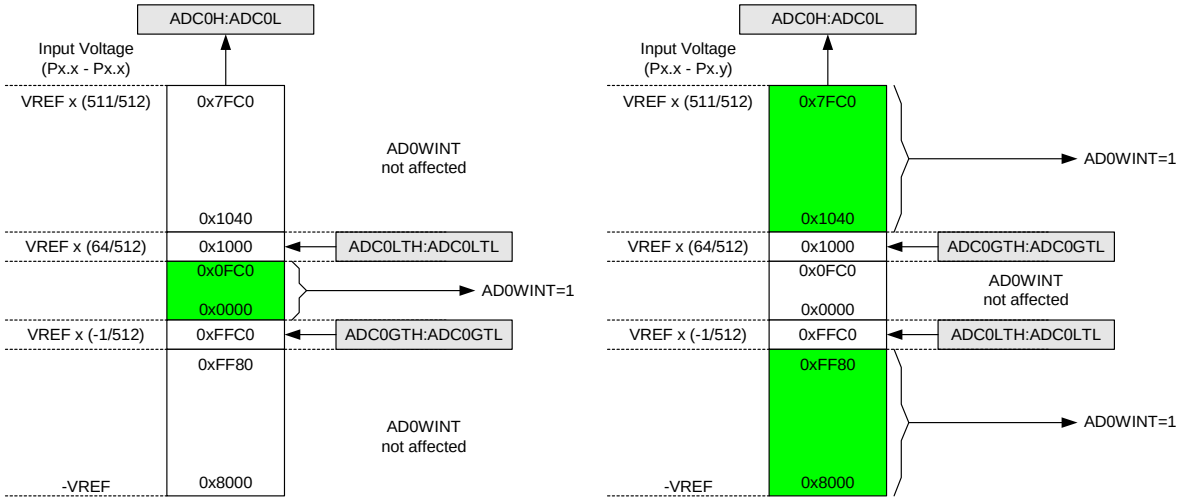


Figure 5.9. ADC Window Compare Example: Left-Justified Differential Data

Table 5.1. ADC0 Electrical Characteristics

V_{DD} = 3.0 V, VREF = 2.40 V, –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		–15	0	+15	LSB
Full Scale Error		–15	–1	+15	LSB
Offset Temperature Coefficient			10		ppm/°C
Dynamic Performance (10 kHz sine-wave Single-ended input, 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion		51	52.5		dB
Total Harmonic Distortion	Up to the 5th harmonic		–67		dB
Spurious-Free Dynamic Range			78		dB
Conversion Rate					
SAR Conversion Clock				3	MHz
Conversion Time in SAR Clocks		10			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[1]			±0.1		°C
Gain			2.86		mV/°C
Gain Error[2]			±33.5		µV/ºC
Offset[1]	(Temp = 0 °C)		776		mV
Offset Error[2]			±8.51		mV
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Operating Mode, 200 ksps		400	900	µA
Power Supply Rejection			±0.3		mV/V

Notes:

1. Includes ADC offset, gain, and linearity variations.
2. Represents one standard deviation from the mean.

6. Voltage Reference (C8051F340/1/2/3/4/5/6/7/A/B Only)

The Voltage reference MUX on C8051F34x devices is configurable to use an externally connected voltage reference, the on-chip reference voltage generator, or the power supply voltage V_{DD} (see Figure 6.1). The REFSL bit in the Reference Control register (REF0CN) selects the reference source. For the internal reference or an external source, REFSL should be set to '0'; For V_{DD} as the reference source, REFSL should be set to '1'.

The BIASE bit enables the internal ADC bias generator, which is used by the ADC and Internal Oscillator. This enable is forced to logic 1 when either of the aforementioned peripherals is enabled. The ADC bias generator may be enabled manually by writing a '1' to the BIASE bit in register REF0CN; see SFR Definition 6.1 for REF0CN register details. The Reference bias generator (see Figure 6.1) is used by the Internal Voltage Reference, Temperature Sensor, and Clock Multiplier. The Reference bias is automatically enabled when any of the aforementioned peripherals are enabled. The electrical specifications for the voltage reference and bias circuits are given in Table 6.1.

Important Note About the VREF Pin: The VREF pin, when not using the on-chip voltage reference or an external precision reference, can be configured as a GPIO Port pin. When using an external voltage reference or the on-chip reference, the VREF pin should be configured as analog pin and skipped by the Digital Crossbar. To configure the VREF pin for analog mode, set the corresponding bit in the PnMDIN register to '0'. To configure the Crossbar to skip the VREF pin, set the corresponding bit in register PnSKIP to '1'. Refer to **Section "15. Port Input/Output" on page 142** for complete Port I/O configuration details.

The temperature sensor connects to the ADC0 positive input multiplexer (see **Section "5.1. Analog Multiplexer" on page 42** for 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.

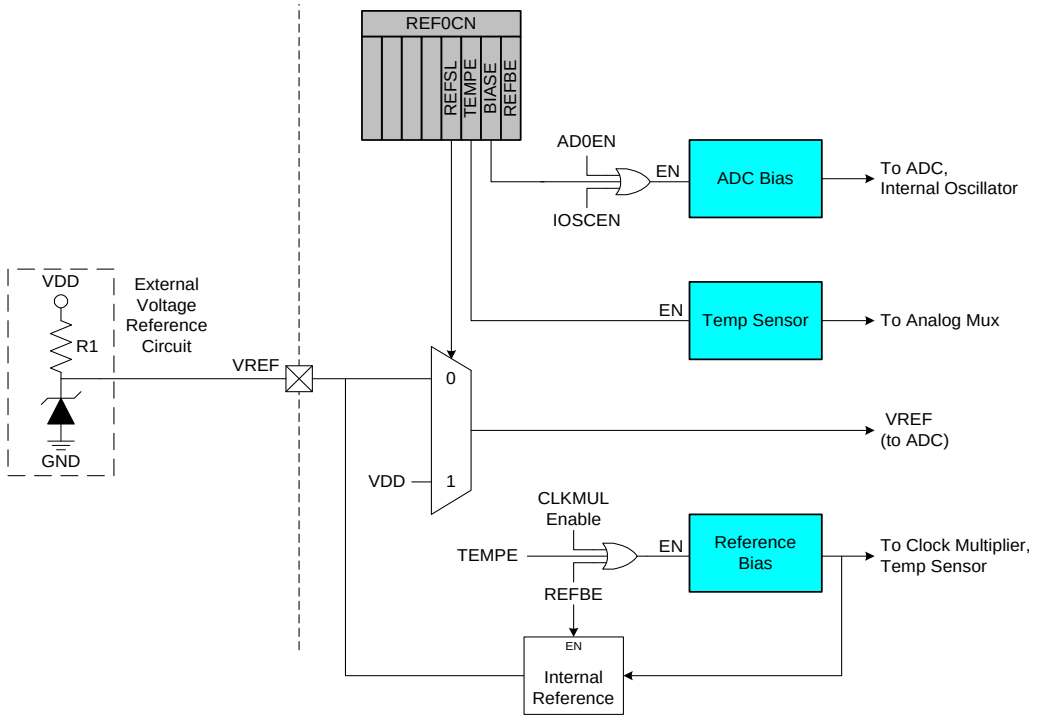


Figure 6.1. Voltage Reference Functional Block Diagram

SFR Definition 6.1. REF0CN: Reference Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	REFSL	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD1

Bits7–3: UNUSED. Read = 00000b; Write = don't care.
Bit3: 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.
Bit2: TEMPE: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.
Bit1: BIASE: Internal Analog Bias Generator Enable Bit.
0: Internal Bias Generator off.
1: Internal Bias Generator on.
Bit0: 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 6.1. 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	2.38	2.44	2.50	V
VREF Short-Circuit Current				10	mA
VREF Temperature Coefficient			15		ppm/°C
Load Regulation	Load = 0 to 200 µA to GND		1.5		ppm/µA
VREF Turn-on Time 1	4.7 µF tantalum, 0.1 µF ceramic bypass		2		ms
VREF Turn-on Time 2	0.1 µF ceramic bypass		20		µs
VREF Turn-on Time 3	no bypass cap		10		µs
Power Supply Rejection			140		ppm/V
External Reference (REFBE = 0)					
Input Voltage Range		0		V_{DD}	V
Input Current	Sample Rate = 200 ksps; VREF = 3.0 V		12		µA
Bias Generators					
ADC Bias Generator	BIASE = '1'		100		µA
Reference Bias Generator			40		µA

7. Comparators

C8051F34x devices include two on-chip programmable voltage Comparators. A block diagram of the comparators is shown in Figure 7.1, where “n” is the comparator number (0 or 1). The two Comparators operate identically with the following exceptions: (1) Their input selections differ, and (2) Comparator0 can be used as a reset source. For input selection details, refer to SFR Definition 7.2 and SFR Definition 7.5.

Each Comparator offers 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, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous signal is 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 “15.2. Port I/O Initialization” on page 147**). Comparator0 may also be used as a reset source (see **Section “11.5. Comparator0 Reset” on page 103**).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 7.2). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input. The Comparator1 inputs are selected in the CPT1MX register (SFR Definition 7.5). The CMX1P1-CMX1P0 bits select the Comparator1 positive input; the CMX1N1-CMX1N0 bits select the Comparator1 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 “15.3. General Purpose Port I/O” on page 150**).

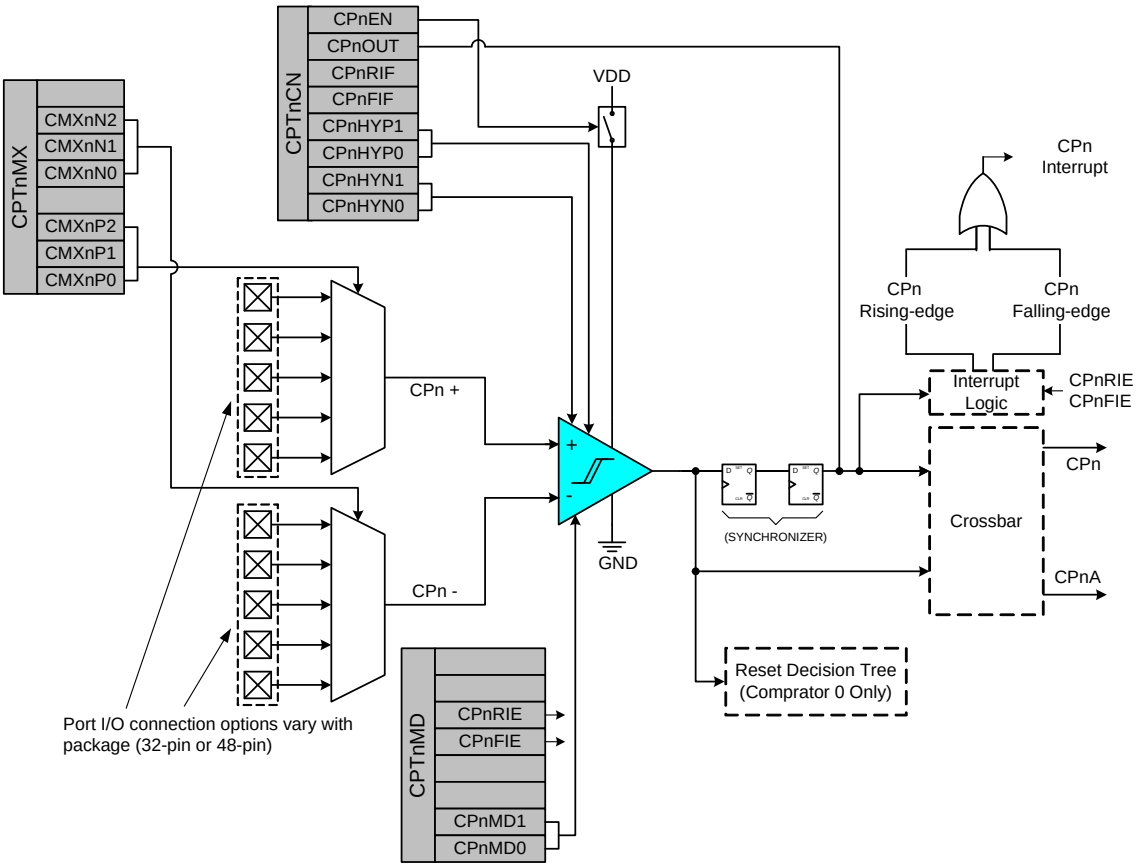


Figure 7.1. Comparator Functional Block Diagram

Comparator outputs can be polled in software, used as an interrupt source, and/or routed to a Port pin. When routed to a Port pin, Comparator outputs are available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and supply current falls to less than 100 nA. See **Section "15.1. Priority Crossbar Decoder" on page 144** 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 7.1.

Comparator response time may be configured in software via the CPTnMD registers (see SFR Definition 7.3 and SFR Definition 7.6). Selecting a longer response time reduces the Comparator supply current. See Table 7.1 for complete timing and supply current specifications.

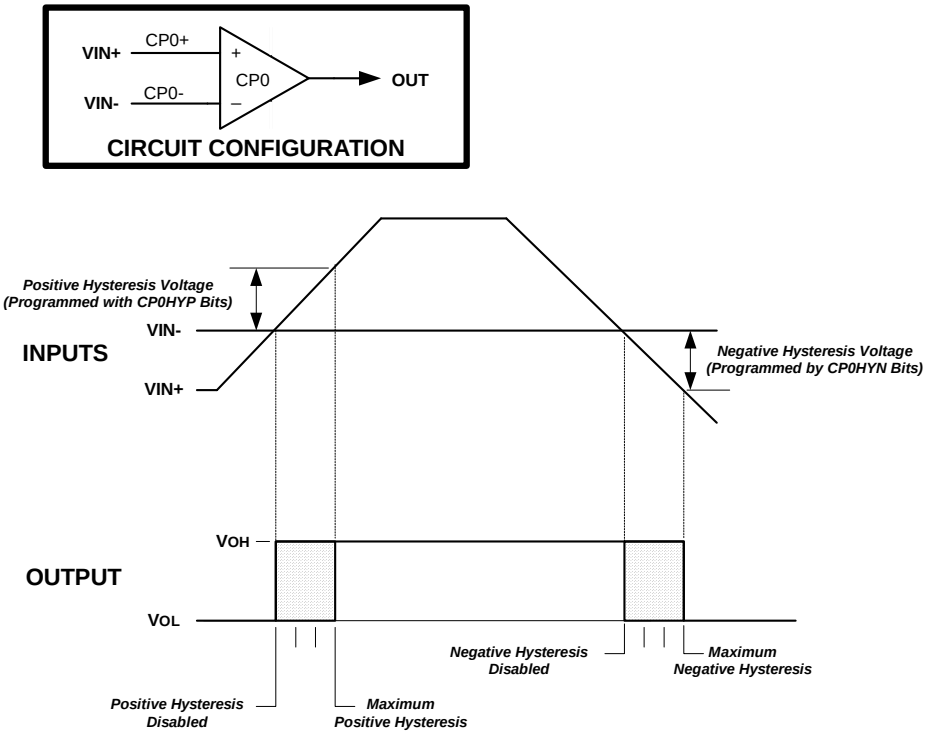


Figure 7.2. Comparator Hysteresis Plot

Comparator hysteresis is programmed using Bits3-0 in the Comparator Control Register CPTnCN (shown in SFR Definition 7.1 and SFR Definition 7.4). The amount of negative hysteresis voltage is determined by the settings of the CPnHYN bits. As shown in Figure 7.2, various levels 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 CPnHYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see **Section "9.3. Interrupt Handler" on page 88**.) The CPnFIF flag is set to '1' upon a Comparator falling-edge, and the CPnRIF flag is set to '1' upon the Comparator rising-edge. Once set, these bits remain set until cleared by software. The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CPnEN bit to '1', and is disabled by clearing this bit to '0'.

SFR Definition 7.1. CPT0CN: Comparator0 Control

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	SFR Address:
								0x9B

Bit7: CP0EN: Comparator0 Enable Bit.
0: Comparator0 Disabled.
1: Comparator0 Enabled.

Bit6: CP0OUT: Comparator0 Output State Flag.
0: Voltage on CP0+ < CP0–.
1: Voltage on CP0+ > CP0–.

Bit5: CP0RIF: Comparator0 Rising-Edge Flag.
0: No Comparator0 Rising Edge has occurred since this flag was last cleared.
1: Comparator0 Rising Edge has occurred.

Bit4: CP0FIF: Comparator0 Falling-Edge Flag.
0: No Comparator0 Falling-Edge has occurred since this flag was last cleared.
1: Comparator0 Falling-Edge Interrupt has occurred.

Bits3–2: 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.

Bits1–0: 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 7.2. CPT0MX: Comparator0 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	CMX0N2	CMX0N1	CMX0N0	-	CMX0P2	CMX0P1	CMX0P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9F

Bit7: UNUSED. Read = 0b, Write = don't care.
Bits6–4: CMX0N2–CMX0N0: Comparator0 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator0 negative input.

CMX0N1	CMX0N1	CMX0N0	Negative Input (32-pin Package)	Negative Input (48-pin Package)
0	0	0	P1.1	P2.1
0	0	1	P1.5	P2.6
0	1	0	P2.1	P3.5
0	1	1	P2.5	P4.4
1	0	0	P0.1	P0.4

Bit3: UNUSED. Read = 0b, Write = don't care.
Bits2–0: CMX0P2–CMX0P0: Comparator0 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator0 positive input.

CMX0P1	CMX0P1	CMX0P0	Positive Input (32-pin Package)	Positive Input (48-pin Package)
0	0	0	P1.0	P2.0
0	0	1	P1.4	P2.5
0	1	0	P2.0	P3.4
0	1	1	P2.4	P4.3
1	0	0	P0.0	P0.3

Note that the port pins used by the comparator depend on the package type (32-pin or 48-pin).

SFR Definition 7.3. CPT0MD: Comparator0 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP0RIE	CP0FIE	-	-	CP0MD1	CP0MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bits7–6: UNUSED. Read = 00b. Write = don't care.
Bit5: CP0RIE: Comparator0 Rising-Edge Interrupt Enable.
0: Comparator0 rising-edge interrupt disabled.
1: Comparator0 rising-edge interrupt enabled.
Bit4: CP0FIE: Comparator0 Falling-Edge Interrupt Enable.
0: Comparator0 falling-edge interrupt disabled.
1: Comparator0 falling-edge interrupt enabled.
Bits3–2: UNUSED. Read = 00b. Write = don't care.
Bits1–0: CP0MD1–CP0MD0: Comparator0 Mode Select
These bits select the response time for Comparator0.

Mode	CP0MD1	CP0MD0	CP0 Response Time*
0	0	0	Fastest Response
1	0	1	
2	1	0	
3	1	1	Lowest Power

* See Table 7.1 for response time parameters.

SFR Definition 7.4. CPT1CN: Comparator1 Control

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	SFR Address:
								0x9A

Bit7: CP1EN: Comparator1 Enable Bit.
0: Comparator1 Disabled.
1: Comparator1 Enabled.

Bit6: CP1OUT: Comparator1 Output State Flag.
0: Voltage on CP1+ < CP1–.
1: Voltage on CP1+ > CP1–.

Bit5: CP1RIF: Comparator1 Rising-Edge Flag.
0: No Comparator1 Rising Edge has occurred since this flag was last cleared.
1: Comparator1 Rising Edge has occurred.

Bit4: CP1FIF: Comparator1 Falling-Edge Flag.
0: No Comparator1 Falling-Edge has occurred since this flag was last cleared.
1: Comparator1 Falling-Edge has occurred.

Bits3–2: 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.

Bits1–0: 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 7.5. CPT1MX: Comparator1 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	CMX1N2	CMX1N1	CMX1N0	-	CMX1P2	CMX1P1	CMX1P0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9E

Bit7: UNUSED. Read = 0b, Write = don't care.
Bits6–4: CMX1N2–CMX1N0: Comparator1 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator1 negative input.

CMX1N2	CMX1N1	CMX1N0	Negative Input (32-pin Package)	Negative Input (48-pin Package)
0	0	0	P1.3	P2.3
0	0	1	P1.7	P3.1
0	1	0	P2.3	P4.0
0	1	1	P2.7	P4.6
1	0	0	P0.5	P1.2

Bit3: UNUSED. Read = 0b, Write = don't care.
Bits2–0: CMX1P1–CMX1P0: Comparator1 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator1 positive input.

CMX1P2	CMX1P1	CMX1P0	Positive Input (32-pin Package)	Positive Input (48-pin Package)
0	0	0	P1.2	P2.2
0	0	1	P1.6	P3.0
0	1	0	P2.2	P3.7
0	1	1	P2.6	P4.5
1	0	0	P0.4	P1.1

Note that the port pins used by the comparator depend on the package type (32-pin or 48-pin).

SFR Definition 7.6. CPT1MD: Comparator1 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP1RIE	CP1FIE	-	-	CP1MD1	CP1MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9C

Bits7–6: UNUSED. Read = 00b, Write = don't care.
Bit5: CP1RIE: Comparator1 Rising-Edge Interrupt Enable.
0: Comparator1 rising-edge interrupt disabled.
1: Comparator1 rising-edge interrupt enabled.
Bit4: CP1FIE: Comparator1 Falling-Edge Interrupt Enable.
0: Comparator1 falling-edge interrupt disabled.
1: Comparator1 falling-edge interrupt enabled.
Bits1–0: CP1MD1–CP1MD0: Comparator1 Mode Select.
These bits select the response time for Comparator1.

Mode	CP1MD1	CP1MD0	CP1 Response Time*
0	0	0	Fastest Response
1	0	1	
2	1	0	
3	1	1	Lowest Power

* See Table 7.1 for response time parameters.

Table 7.1. Comparator Electrical Characteristics

V_{DD} = 3.0 V, –40 to +85 °C unless otherwise noted.
All specifications apply to both Comparator0 and Comparator1 unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		100		ns
	CP0+ – CP0– = –100 mV		250		ns
Response Time: Mode 1, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		175		ns
	CP0+ – CP0– = –100 mV		500		ns
Response Time: Mode 2, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		320		ns
	CP0+ – CP0– = –100 mV		1100		ns
Response Time: Mode 3, Vcm* = 1.5 V	CP0+ – CP0– = 100 mV		1050		ns
	CP0+ – CP0– = –100 mV		5200		ns
Common-Mode Rejection Ratio			1.5	4	mV/V
Positive Hysteresis 1	CP0HYP1–0 = 00		0	1	mV
Positive Hysteresis 2	CP0HYP1–0 = 01	2	5	10	mV
Positive Hysteresis 3	CP0HYP1–0 = 10	7	10	20	mV
Positive Hysteresis 4	CP0HYP1–0 = 11	15	20	30	mV
Negative Hysteresis 1	CP0HYN1–0 = 00		0	1	mV
Negative Hysteresis 2	CP0HYN1–0 = 01	2	5	10	mV
Negative Hysteresis 3	CP0HYN1–0 = 10	7	10	20	mV
Negative Hysteresis 4	CP0HYN1–0 = 11	15	20	30	mV
Inverting or Non-Inverting Input Voltage Range		–0.25		V_{DD} + 0.25	V
Input Capacitance			3		pF
Input Bias Current			0.001		nA
Input Offset Voltage		–5		+5	mV
Power Supply					
Power Supply Rejection			0.1		mV/V
Power-up Time			10		µs
Supply Current at DC	Mode 0		7.6		µA
	Mode 1		3.2		µA
	Mode 2		1.3		µA
	Mode 3		0.4		µA

***Note:** Vcm is the common-mode voltage on CP0+ and CP0–.

8. Voltage Regulator (REG0)

C8051F34x devices include a voltage regulator (REG0). When enabled, the REG0 output appears on the V_{DD} pin and can be used to power external devices. REG0 can be enabled/disabled by software using bit REGEN in register REG0CN. See Table 8.1 for REG0 electrical characteristics.

Note that the VBUS signal must be connected to the VBUS pin when using the device in a USB network. The VBUS signal should only be connected to the REGIN pin when operating the device as a bus-powered function. REG0 configuration options are shown in Figure 8.1–Figure 8.4.

8.1. Regulator Mode Selection

REG0 offers a low power mode intended for use when the device is in suspend mode. In this low power mode, the REG0 output remains as specified; however the REG0 dynamic performance (response time) is degraded. See Table 8.1 for normal and low power mode supply current specifications. The REG0 mode selection is controlled via the REGMOD bit in register REG0CN.

8.2. VBUS Detection

When the USB Function Controller is used (see section **Section "16. Universal Serial Bus Controller (USB0)" on page 159**), the VBUS signal should be connected to the VBUS pin. The VBSTAT bit (register REG0CN) indicates the current logic level of the VBUS signal. If enabled, a VBUS interrupt will be generated when the VBUS signal matches the polarity selected by the VBPOL bit in register REG0CN. The VBUS interrupt is level-sensitive, and has no associated interrupt pending flag. The VBUS interrupt will be active as long as the VBUS signal matches the polarity selected by VBPOL. See Table 8.1 for VBUS input parameters.

Important Note: When USB is selected as a reset source, a system reset will be generated when the VBUS signal matches the polarity selected by the VBPOL bit. See **Section "11. Reset Sources" on page 100** for details on selecting USB as a reset source

Table 8.1. Voltage Regulator Electrical Specifications

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range[1]		2.7		5.25	V
Output Voltage (V_{DD})[2]	Output Current = 1 to 100 mA	3.0	3.3	3.6	V
Output Current[2]				100	mA
VBUS Detection Input Low Voltage				1.0	V
VBUS Detection Input High Voltage		3.0			V
Bias Current	Normal Mode (REGMOD = '0') Low Power Mode (REGMOD = '1')		65 35	111 61	µA
Dropout Voltage (V_{DO})[3]			1		mV/mA

Notes:
1. Input range specified for regulation. When an external regulator is used, should be tied to V_{DD}.
2. Output current is total regulator output, including any current required by the C8051F34x.
3. The minimum input voltage is 2.70 V or VDD + V_{DO} (max load), whichever is greater.

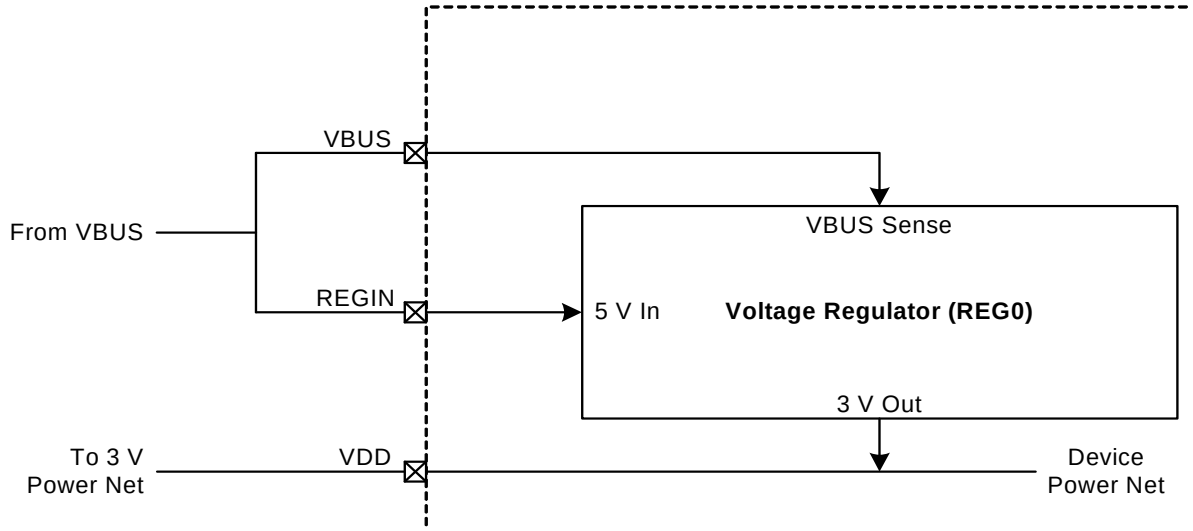


Figure 8.1. REG0 Configuration: USB Bus-Powered

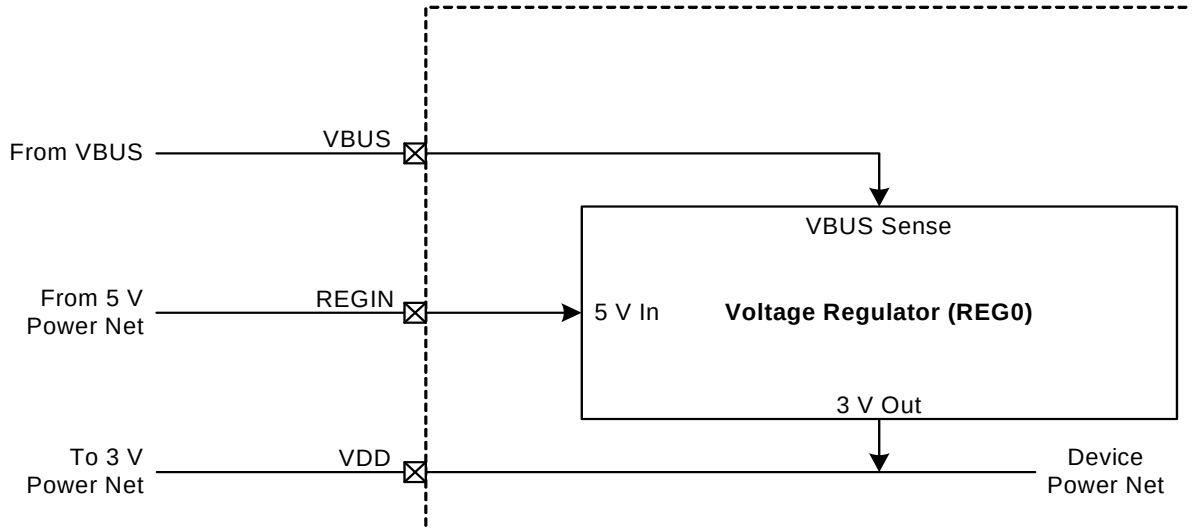


Figure 8.2. REG0 Configuration: USB Self-Powered

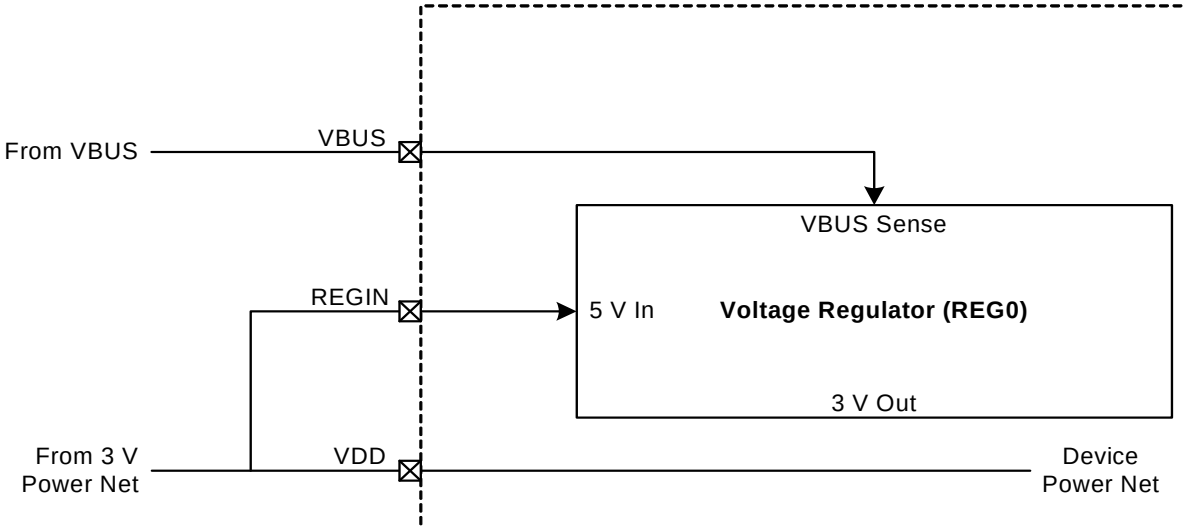


Figure 8.3. REG0 Configuration: USB Self-Powered, Regulator Disabled

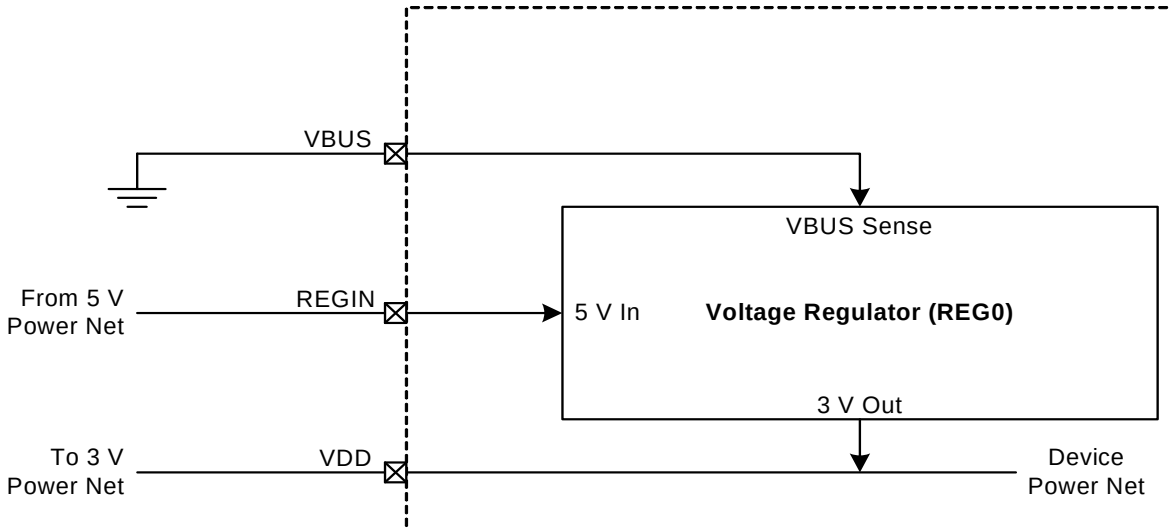


Figure 8.4. REG0 Configuration: No USB Connection

SFR Definition 8.1. REG0CN: Voltage Regulator Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
REGDIS	VBSTAT	VBPOL	REGMOD	Reserved	Reserved	Reserved	Reserved	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xC9

Bit7: REGDIS: Voltage Regulator Disable.
0: Voltage Regulator Enabled.
1: Voltage Regulator Disabled.

Bit6: VBSTAT: VBUS Signal Status.
0: VBUS signal currently absent (device not attached to USB network).
1: VBUS signal currently present (device attached to USB network).

Bit5: VBPOL: VBUS Interrupt Polarity Select.
This bit selects the VBUS interrupt polarity.
0: VBUS interrupt active when VBUS is low.
1: VBUS interrupt active when VBUS is high.

Bit4: REGMOD: Voltage Regulator Mode Select.
This bit selects the Voltage Regulator mode. When REGMOD is set to ‘1’, the voltage regulator operates in low power (suspend) mode.
0: USB0 Voltage Regulator in normal mode.
1: USB0 Voltage Regulator in low power mode.

Bits3–0: Reserved. Read = 0000b. Must Write = 0000b.

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 four 16-bit counter/timers (see description in **Section 21**), an enhanced full-duplex UART (see description in **Section 18**), an Enhanced SPI (see description in **Section 20**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (**Section 9.2.6**), and 25 Port I/O (see description in **Section 15**). The CIP-51 also includes on-chip debug hardware (see description in **Section 23**), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

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

- Fully Compatible with MCS-51 Instruction Set
- 0 to 48 MHz Clock Frequency
- 256 Bytes of Internal RAM
- 25 Port I/O
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

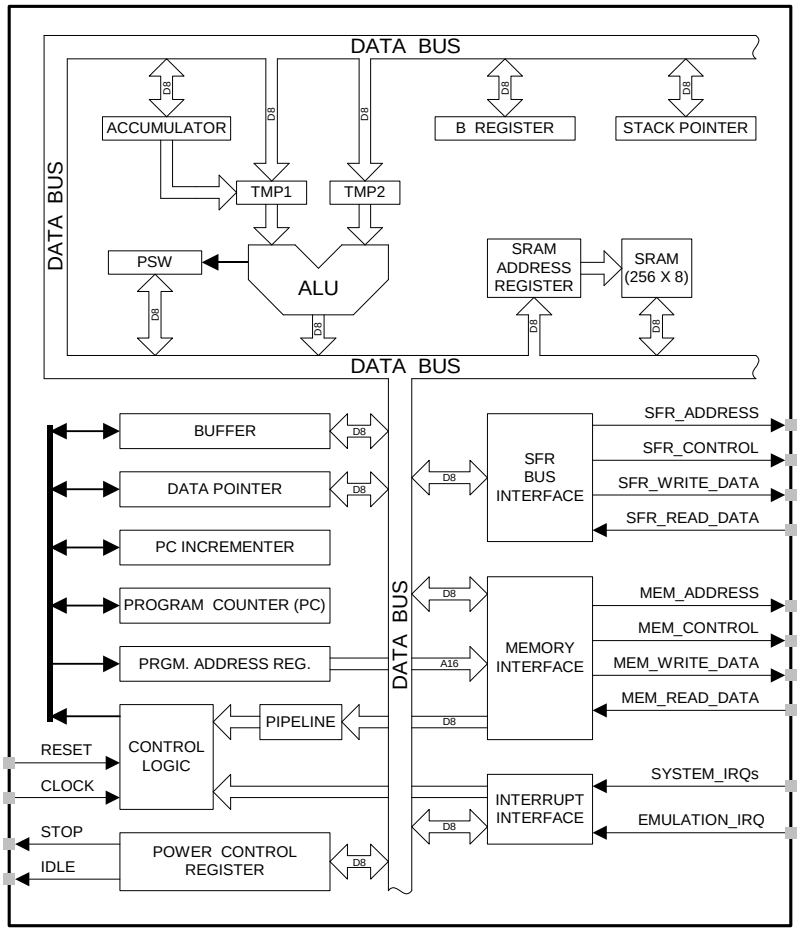


Figure 9.1. CIP-51 Block Diagram

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 maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that for execution time.

Clocks to Execute	1	2	2/4	3	3/5	4	5	4/6	6	8
Number of Instructions	26	50	5	10	7	5	2	1	2	1

Programming and Debugging Support
In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire Development Interface (C2). Note that the re-programmable Flash can also be read and changed a single byte at a time 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 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. C2 details can be found in **Section “23. C2 Interface” on page 271**.

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, debugger, and programmer. The IDE's debugger and programmer interface to the CIP-51 via the C2 interface to provide fast and efficient in-system device programming and debugging. An 8051 assembler, linker and evaluation ‘C’ compiler are included in the Development Kit. Many third party macro assemblers and C compilers are also available, which can be used directly with the IDE.

9.1. 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.

9.1.1. 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 two fewer clock cycles 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.

9.1.2. MOVX Instruction and Program Memory

In the CIP-51, the MOVX instruction serves three purposes: accessing on-chip XRAM, accessing off-chip data XRAM (only on C8051F340/1/4/5/8 devices), 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 "12. Flash Memory" on page 107**). The External Memory Interface (only on C8051F340/1/4/5/8 devices) provides a fast access interface to off-chip data XRAM (or memory-mapped peripherals) via the MOVX instruction. Refer to Section "**13. External Data Memory Interface and On-Chip XRAM**" on page **114**. 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
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

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

Mnemonic	Description	Bytes	Clock Cycles
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
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

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

Mnemonic	Description	Bytes	Clock Cycles
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	2/4
JNC rel	Jump if Carry is not set	2	2/4
JB bit, rel	Jump if direct bit is set	3	3/5
JNB bit, rel	Jump if direct bit is not set	3	3/5
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/5
Program Branching			
ACALL addr11	Absolute subroutine call	2	4
LCALL addr16	Long subroutine call	3	5
RET	Return from subroutine	1	6
RETI	Return from interrupt	1	6
AJMP addr11	Absolute jump	2	4
LJMP addr16	Long jump	3	5
SJMP rel	Short jump (relative address)	2	4
JMP @A+DPTR	Jump indirect relative to DPTR	1	4
JZ rel	Jump if A equals zero	2	2/4
JNZ rel	Jump if A does not equal zero	2	2/4
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/5
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/5
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/5
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/6
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/4
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/5
NOP	No operation	1	1

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 (two's 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 8K-byte program memory space.

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

9.2. 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. The CIP-51 memory organization is shown in Figure 9.2 and Figure 9.3.

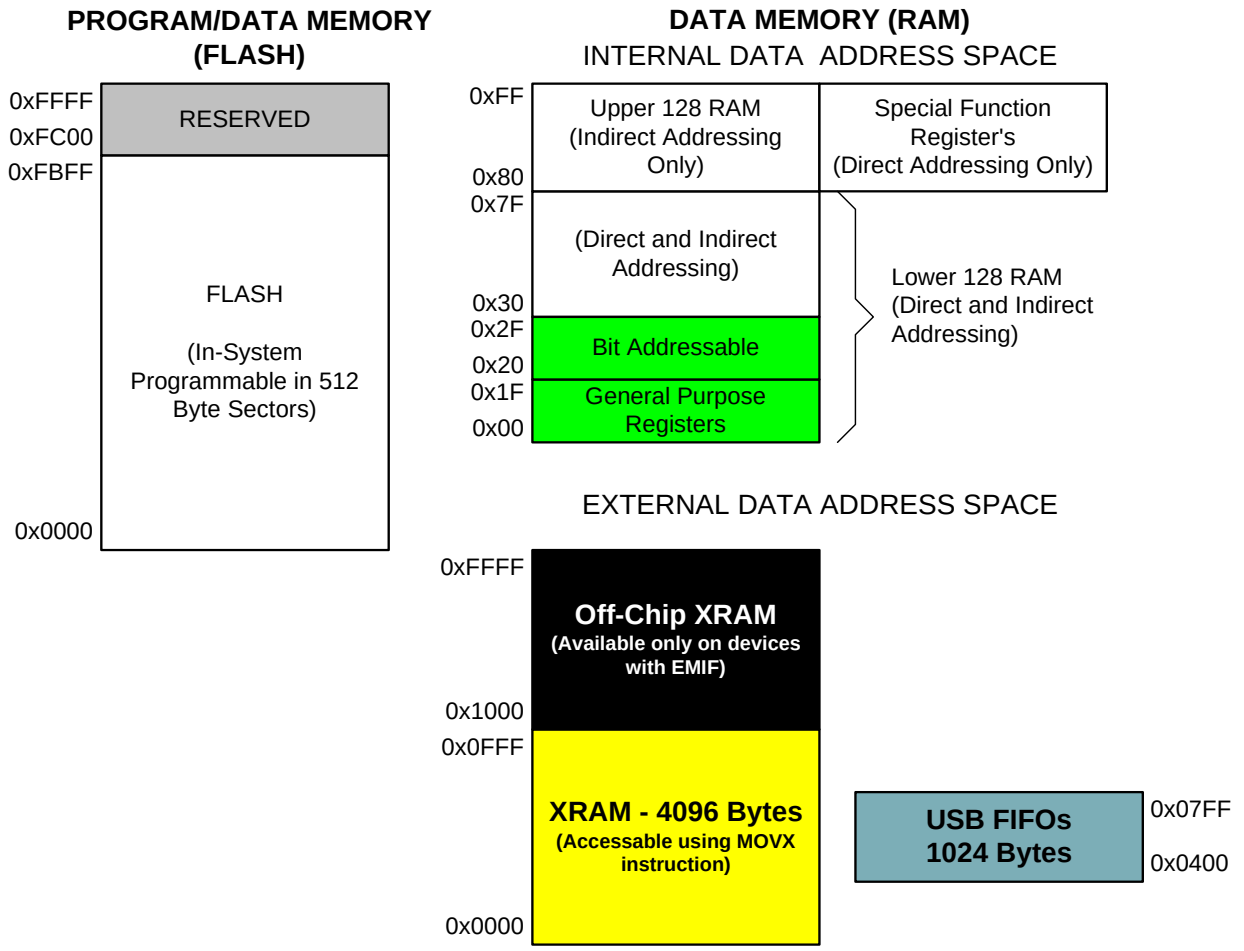


Figure 9.2. On-Chip Memory Map for 64 kB Devices

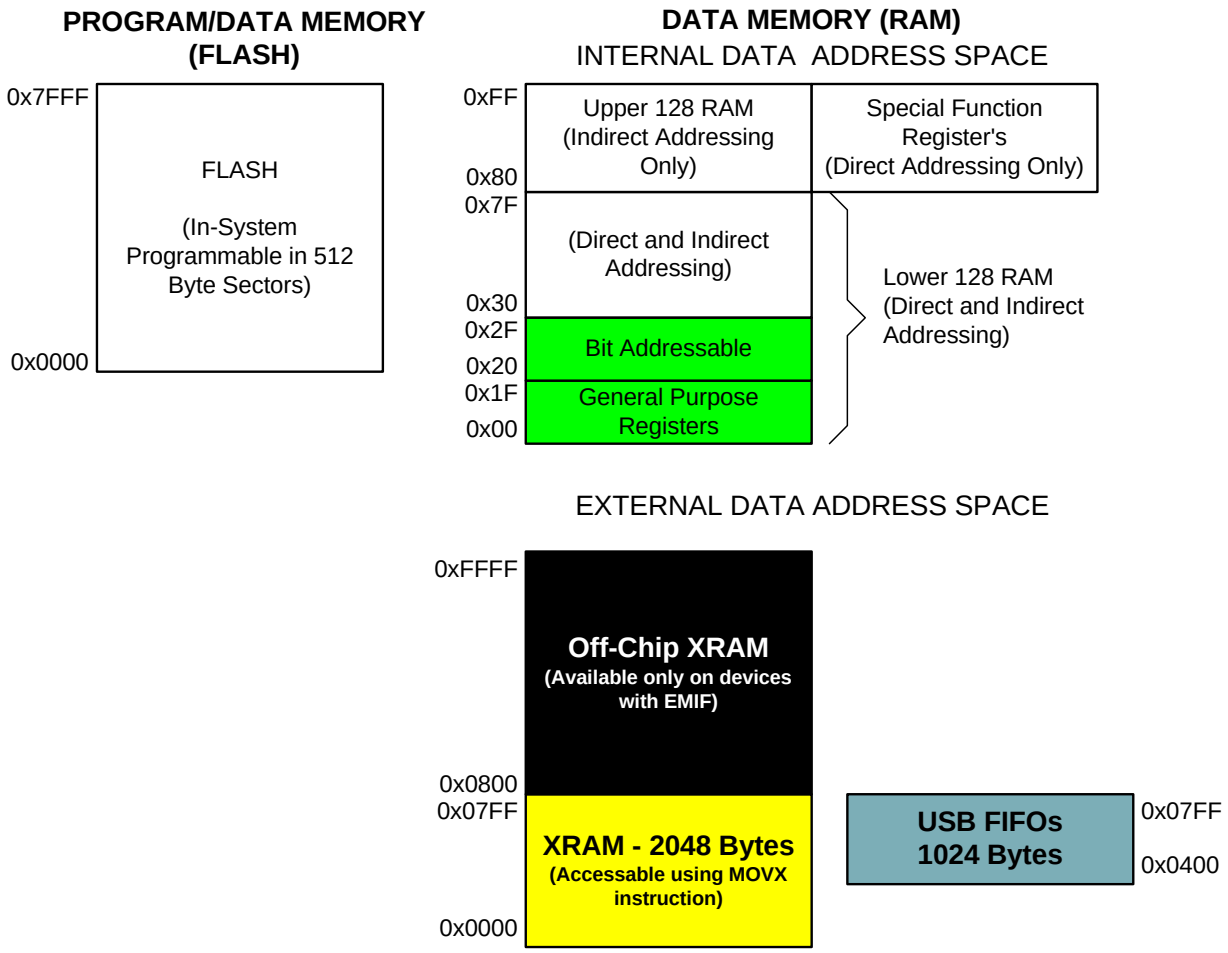


Figure 9.3. On-Chip Memory Map for 32 kB Devices

9.2.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F34x implements 64k or 32k bytes of this program memory space as in-system, re-programmable Flash memory. Note that on the 64k versions of the C8051F34x, addresses above 0xFBFF are reserved.

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 "12. Flash Memory" on page 107** for further details.

9.2.2. Data Memory

The CIP-51 includes 256 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 scratch pad 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 SFRs. 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.

9.2.3. 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 9.4). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

9.2.4. 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 bit7 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, 22h.3
```

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

9.2.5. 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, 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.

9.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs 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 SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs 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 datasheet, as indicated in Table 9.3, for a detailed description of each register.

Table 9.2. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0	B	P0MDIN	P1MDIN	P2MDIN	P3MDIN	P4MDIN	EIP1	EIP2
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF	SMOD1	EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	P3SKIP
D0	PSW	REF0CN	SCON1	SBUF1	P0SKIP	P1SKIP	P2SKIP	USB0XCN
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H	-	-
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P4
B8	IP	CLKMUL	AMX0N	AMX0P	ADC0CF	ADC0L	ADC0H	-
B0	P3	OSCXCN	OSCICN	OSCICL	SBRLL1	SBRLH1	FLSCL	FLKEY
A8	IE	CLKSEL	EMI0CN	-	SBCON1	-	P4MDOUT	PFE0CN
A0	P2	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT	P2MDOUT	P3MDOUT
98	SCON0	SBUF0	CPT1CN	CPT0CN	CPT1MD	CPT0MD	CPT1MX	CPT0MX
90	P1	TMR3CN	TMR3RLL	TMR3RLH	TMR3L	TMR3H	USB0ADR	USB0DAT
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	EMI0TC	EMI0CF	OSCLCN	PCON
	0(8) (bit addressable)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Table 9.3. Special Function Registers

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

Register	Address	Description	Page
ACC	0xE0	Accumulator	87
ADC0CF	0xBC	ADC0 Configuration	50
ADC0CN	0xE8	ADC0 Control	51
ADC0GTH	0xC4	ADC0 Greater-Than Compare High	52
ADC0GTL	0xC3	ADC0 Greater-Than Compare Low	52
ADC0H	0xBE	ADC0 High	50
ADC0L	0xBD	ADC0 Low	50
ADC0LTH	0xC6	ADC0 Less-Than Compare Word High	53
ADC0LTL	0xC5	ADC0 Less-Than Compare Word Low	53
AMX0N	0xBA	AMUX0 Negative Channel Select	49
AMX0P	0xBB	AMUX0 Positive Channel Select	48
B	0xF0	B Register	88
CKCON	0x8E	Clock Control	241
CLKMUL	0xB9	Clock Multiplier	138
CLKSEL	0xA9	Clock Select	140
CPT0CN	0x9B	Comparator0 Control	62
CPT0MD	0x9D	Comparator0 Mode Selection	64
CPT0MX	0x9F	Comparator0 MUX Selection	63
CPT1CN	0x9A	Comparator1 Control	65
CPT1MD	0x9C	Comparator1 Mode Selection	67
CPT1MX	0x9E	Comparator1 MUX Selection	66
DPH	0x83	Data Pointer High	86
DPL	0x82	Data Pointer Low	86
EIE1	0xE6	Extended Interrupt Enable 1	93
EIE2	0xE7	Extended Interrupt Enable 2	95
EIP1	0xF6	Extended Interrupt Priority 1	94
EIP2	0xF7	Extended Interrupt Priority 2	95
EMI0CN	0xAA	External Memory Interface Control	117
EMI0CF	0x85	External Memory Interface Configuration	118
EMI0TC	0x84	External Memory Interface Timing	123
FLKEY	0xB7	Flash Lock and Key	112
FLSCL	0xB6	Flash Scale	113
IE	0xA8	Interrupt Enable	91
IP	0xB8	Interrupt Priority	92
IT01CF	0xE4	INT0/INT1 Configuration	96
OSCICL	0xB3	Internal Oscillator Calibration	133
OSCICN	0xB2	Internal Oscillator Control	132
OSCLCN	0x86	Internal Low-Frequency Oscillator Control	134
OSCXCN	0xB1	External Oscillator Control	137
P0	0x80	Port 0 Latch	150
P0MDIN	0xF1	Port 0 Input Mode Configuration	150
P0MDOUT	0xA4	Port 0 Output Mode Configuration	151
P0SKIP	0xD4	Port 0 Skip	151
P1	0x90	Port 1 Latch	152

Table 9.3. Special Function Registers (Continued)

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

Register	Address	Description	Page
P1MDIN	0xF2	Port 1 Input Mode Configuration	152
P1MDOUT	0xA5	Port 1 Output Mode Configuration	152
P1SKIP	0xD5	Port 1 Skip	153
P2	0xA0	Port 2 Latch	153
P2MDIN	0xF3	Port 2 Input Mode Configuration	153
P2MDOUT	0xA6	Port 2 Output Mode Configuration	154
P2SKIP	0xD6	Port 2 Skip	154
P3	0xB0	Port 3 Latch	155
P3MDIN	0xF4	Port 3 Input Mode Configuration	155
P3MDOUT	0xA7	Port 3 Output Mode Configuration	155
P3SKIP	0xDF	Port 3Skip	156
P4	0xC7	Port 4 Latch	156
P4MDIN	0xF5	Port 4 Input Mode Configuration	157
P4MDOUT	0xAE	Port 4 Output Mode Configuration	157
PCA0CN	0xD8	PCA Control	266
PCA0CPH0	0xFC	PCA Capture 0 High	270
PCA0CPH1	0xEA	PCA Capture 1 High	270
PCA0CPH2	0xEC	PCA Capture 2 High	270
PCA0CPH3	0xEE	PCA Capture 3High	270
PCA0CPH4	0xFE	PCA Capture 4 High	270
PCA0CPL0	0xFB	PCA Capture 0 Low	269
PCA0CPL1	0xE9	PCA Capture 1 Low	269
PCA0CPL2	0xEB	PCA Capture 2 Low	269
PCA0CPL3	0xED	PCA Capture 3 Low	269
PCA0CPL4	0xFD	PCA Capture 4 Low	269
PCA0CPM0	0xDA	PCA Module 0 Mode Register	268
PCA0CPM1	0xDB	PCA Module 1 Mode Register	268
PCA0CPM2	0xDC	PCA Module 2 Mode Register	268
PCA0CPM3	0xDD	PCA Module 3 Mode Register	268
PCA0CPM4	0xDE	PCA Module 4 Mode Register	268
PCA0H	0xFA	PCA Counter High	269
PCA0L	0xF9	PCA Counter Low	269
PCA0MD	0xD9	PCA Mode	267
PCON	0x87	Power Control	98
PFE0CN	0xAF	Prefetch Engine Control	99
PSCTL	0x8F	Program Store R/W Control	112
PSW	0xD0	Program Status Word	87
REF0CN	0xD1	Voltage Reference Control	58
REG0CN	0xC9	Voltage Regulator Control	72
RSTSRC	0xEF	Reset Source Configuration/Status	105
SBCON1	0xAC	UART1 Baud Rate Generator Control	220
SBRLH1	0xB5	UART1 Baud Rate Generator High	221
SBRLL1	0xB4	UART1 Baud Rate Generator Low	221
SBUF1	0xD3	UART1 Data Buffer	220
SCON1	0xD2	UART1 Control	218

Table 9.3. Special Function Registers (Continued)

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

Register	Address	Description	Page
SBUF0	0x99	UART0 Data Buffer	211
SCON0	0x98	UART0 Control	210
SMB0CF	0xC1	SMBus Configuration	194
SMB0CN	0xC0	SMBus Control	196
SMB0DAT	0xC2	SMBus Data	198
SMOD1	0xE5	UART1 Mode	219
SP	0x81	Stack Pointer	86
SPI0CFG	0xA1	SPI Configuration	229
SPI0CKR	0xA2	SPI Clock Rate Control	231
SPI0CN	0xF8	SPI Control	230
SPI0DAT	0xA3	SPI Data	231
TCON	0x88	Timer/Counter Control	239
TH0	0x8C	Timer/Counter 0 High	242
TH1	0x8D	Timer/Counter 1 High	242
TL0	0x8A	Timer/Counter 0 Low	242
TL1	0x8B	Timer/Counter 1 Low	242
TMOD	0x89	Timer/Counter Mode	240
TMR2CN	0xC8	Timer/Counter 2 Control	247
TMR2H	0xCD	Timer/Counter 2 High	248
TMR2L	0xCC	Timer/Counter 2 Low	248
TMR2RLH	0xCB	Timer/Counter 2 Reload High	248
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	248
TMR3CN	0x91	Timer/Counter 3Control	253
TMR3H	0x95	Timer/Counter 3 High	254
TMR3L	0x94	Timer/Counter 3Low	254
TMR3RLH	0x93	Timer/Counter 3 Reload High	254
TMR3RLL	0x92	Timer/Counter 3 Reload Low	254
VDM0CN	0xFF	V_{DD} Monitor Control	102
USB0ADR	0x96	USB0 Indirect Address Register	163
USB0DAT	0x97	USB0 Data Register	164
USB0XCN	0xD7	USB0 Transceiver Control	161
XBR0	0xE1	Port I/O Crossbar Control 0	148
XBR1	0xE2	Port I/O Crossbar Control 1	149
XBR2	0xE3	Port I/O Crossbar Control 2	149
All Other Addresses		Reserved	

9.2.7. 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 l. 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 datasheet associated with their corresponding system function.

SFR Definition 9.1. DPL: Data Pointer Low Byte

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:
								0x82

Bits7–0: DPL: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed memory.

SFR Definition 9.2. DPH: Data Pointer High Byte

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:
								0x83

Bits7–0: DPH: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed memory.

SFR Definition 9.3. SP: Stack Pointer

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	SFR Address:
								0x81

Bits7–0: 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 9.4. PSW: Program Status Word

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	SFR Address:
							(bit addressable)	0xD0

Bit7: CY: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.

Bit6: AC: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.

Bit5: F0: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: 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

Bit2: 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.

Bit1: F1: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: PARITY: Parity Flag.
This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

SFR Definition 9.5. ACC: Accumulator

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	SFR Address:
							(bit addressable)	0xE0

Bits7–0: ACC: Accumulator.
This register is the accumulator for arithmetic operations.

SFR Definition 9.6. B: B Register

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	SFR Address:
							(bit addressable)	0xF0

Bits7–0: B: B Register.
This register serves as a second accumulator for certain arithmetic operations.

9.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting multiple interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs 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 an SFR (IE-EIE2). 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.

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.

9.3.1. MCU Interrupt Sources and Vectors

The MCU supports multiple 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 9.4 on page 90. Refer to the datasheet 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).

9.3.2. External Interrupts

The $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{\text{INT0}}$ Polarity) and IN1PL ($\overline{\text{INT1}}$ 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 235**) select level or edge sensitive. The following table 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 9.13). 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 "15.1. Priority Crossbar Decoder" on page 144** for complete details on configuring the Crossbar). In the typical configuration, the external interrupt pin should be skipped in the crossbar and configured as open-drain with the pin latch set to '1'.

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.

9.3.3. 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 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 9.4.

9.3.4. 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 6 system clock cycles: 1 clock cycle to detect the interrupt and 5 clock cycles to complete the LCALL to the ISR. 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) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 20 system clock cycles: 1 clock cycle to detect the interrupt, 6 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. 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.

Note that the CPU is stalled during Flash write/erase operations and USB FIFO MOVX accesses (see **Section "13.2. Accessing USB FIFO Space" on page 115**). Interrupt service latency will be increased for interrupts occurring while the CPU is stalled. The latency for these situations will be determined by the standard interrupt service procedure (as described above) and the amount of time the CPU is stalled.

Table 9.4. 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)
USB0	0x0043	8	Special	N	N	EUSB0 (EIE1.1)	PUSB0 (EIP1.1)
ADC0 Window Compare	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 Conversion Complete	0x0053	10	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.3)	PADC0 (EIP1.3)
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)
VBUS Level	0x007B	15	N/A	N/A	N/A	EVBUS (EIE2.0)	PVBUS (EIP2.0)
UART1	0x0083	16	RI1 (SCON1.0) TI1 (SCON1.1)	N	N	ES1 (EIE2.1)	PS1 (EIP2.1)

9.3.5. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described below. Refer to the datasheet 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 9.7. IE: Interrupt Enable

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	SFR Address:
							(bit addressable)	0xA8

Bit7: EA: Enable All Interrupts.
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.

Bit6: 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.

Bit5: 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.

Bit4: ES0: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit3: 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.

Bit2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the $\overline{\text{INT1}}$ input.

Bit1: 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.

Bit0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the $\overline{\text{INT0}}$ input.

SFR Definition 9.8. IP: Interrupt Priority

R/W	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	SFR Address:
							(bit addressable)	0xB8

Bit7: UNUSED. Read = 1, Write = don't care.

Bit6: 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.

Bit5: 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 interrupts set to high priority level.

Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupts set to high priority level.

Bit3: 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 interrupts set to high priority level.

Bit2: 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.

Bit1: 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.

Bit0: 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 9.9. EIE1: Extended Interrupt Enable 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	EUSB0	ESMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE6

Bit7: 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.

Bit6: 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.

Bit5: 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.

Bit4: 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.

Bit3: 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.

Bit2: EWADC0: Enable Window Comparison ADC0 Interrupt.
This bit sets the masking of ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison interrupt.
1: Enable interrupt requests generated by ADC0 Window Compare flag (AD0WINT).

Bit1: EUSB0: Enable USB0 Interrupt.
This bit sets the masking of the USB0 interrupt.
0: Disable all USB0 interrupts.
1: Enable interrupt requests generated by USB0.

Bit0: 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 9.10. EIP1: Extended Interrupt Priority 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PT3	PCP1	PCP0	PPCA0	PADC0	PWADC0	PUSB0	PSMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF6

Bit7: 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.

Bit6: 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.

Bit5: 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.

Bit4: 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.

Bit3: 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.

Bit2: PWADC0: ADC0 Window Comparator Interrupt Priority Control.
This bit sets the priority of the ADC0 Window interrupt.
0: ADC0 Window interrupt set to low priority level.
1: ADC0 Window interrupt set to high priority level.

Bit1: PUSB0: USB0 Interrupt Priority Control.
This bit sets the priority of the USB0 interrupt.
0: USB0 interrupt set to low priority level.
1: USB0 interrupt set to high priority level.

Bit0: 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 9.11. EIE2: Extended Interrupt Enable 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	ES1	EVBUS	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE7

Bits7–2: UNUSED. Read = 000000b. Write = don't care.
Bit1: ES1: Enable UART1 Interrupt.
This bit sets the masking of the UART1 interrupt.
0: Disable UART1 interrupt.
1: Enable UART1 interrupt.
Bit0: EVBUS: Enable VBUS Level Interrupt.
This bit sets the masking of the VBUS interrupt.
0: Disable all VBUS interrupts.
1: Enable interrupt requests generated by VBUS level sense.

SFR Definition 9.12. EIP2: Extended Interrupt Priority 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	PS1	PVBUS	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xF7

Bits7–2: UNUSED. Read = 000000b. Write = don't care.
Bit1: PS1: UART1 Interrupt Priority Control.
This bit sets the priority of the UART1 interrupt.
0: UART1 interrupt set to low priority level.
1: UART1 interrupts set to high priority level.
Bit0: PVBUS: VBUS Level Interrupt Priority Control.
This bit sets the priority of the VBUS interrupt.
0: VBUS interrupt set to low priority level.
1: VBUS interrupt set to high priority level.

SFR Definition 9.13. IT01CF: INT0/INT1 Configuration

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	SFR Address:
								0xE4

Note: Refer to SFR Definition 21.1 for INT0/1 edge- or level-sensitive interrupt selection.

Bit7: IN1PL: $\overline{\text{INT1}}$ Polarity
0: $\overline{\text{INT1}}$ input is active low.
1: $\overline{\text{INT1}}$ input is active high.

Bits6–4: IN1SL2–0: $\overline{\text{INT1}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT1}}$. Note that this pin assignment is independent of the Crossbar; $\overline{\text{INT1}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

IN1SL2–0	$\overline{\text{INT1}}$ 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

Bit3: IN0PL: $\overline{\text{INT0}}$ Polarity
0: $\overline{\text{INT0}}$ interrupt is active low.
1: $\overline{\text{INT0}}$ interrupt is active high.

Bits2–0: INT0SL2–0: $\overline{\text{INT0}}$ Port Pin Selection Bits
These bits select which Port pin is assigned to $\overline{\text{INT0}}$. Note that this pin assignment is independent of the Crossbar. $\overline{\text{INT0}}$ will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

IN0SL2–0	$\overline{\text{INT0}}$ 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

9.4. Power Management Modes

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

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

The internal oscillator can be placed in Suspend mode (see **Section “14. Oscillators” on page 131**). In Suspend mode, the internal oscillator is stopped until a non-idle USB event is detected, or the VBUS input signal matches the polarity selected by the VBPOL bit in register REG0CN (SFR Definition 8.1).

9.4.1. Idle Mode

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

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

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to **Section “11.6. PCA Watchdog Timer Reset” on page 103** for more information on the use and configuration of the WDT.

9.4.2. 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 execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may 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 in STOP mode for longer than the MCD timeout of 100 µsec.

SFR Definition 9.14. PCON: Power Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GF5	GF4	GF3	GF2	GF1	GF0	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x87

Bits7–2: GF5–GF0: General Purpose Flags 5–0.
These are general purpose flags for use under software control.

Bit1: STOP: Stop Mode Select.
Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0.
1: CPU goes into Stop mode (internal oscillator stopped).

Bit0: IDLE: Idle Mode Select.
Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0.
1: CPU goes into Idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and Analog Peripherals are still active.)

10. Prefetch Engine

The 48 MHz versions of the C8051F34x family of devices incorporate a 2-byte prefetch engine. Because the access time of the FLASH memory is 40 ns, and the minimum instruction time is roughly 20 ns, the prefetch engine is necessary for full-speed code execution. Instructions are read from FLASH memory two 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 allows instructions to be executed at full speed. When a code branch occurs, the processor may be stalled for up to two clock cycles while the next set of code bytes is retrieved from FLASH memory. The FLRT bit (FLSCL.4) determines how many clock cycles are used to read each set of two code bytes from FLASH. When operating from a system clock of 25 MHz or less, the FLRT bit should be set to '0' so that the prefetch engine takes only one clock cycle for each read. When operating with a system clock of greater than 25 MHz (up to 48 MHz), the FLRT bit should be set to '1', so that each prefetch code read lasts for two clock cycles.

SFR Definition 10.1. PFE0CN: Prefetch Engine Control

R	R	R/W	R	R	R	R	R/W	Reset Value
		PFEN					FLBWE	00100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xAF

Bits 7–6: Unused. Read = 00b; Write = Don't Care
Bit 5: PFEN: Prefetch Enable.
This bit enables the prefetch engine.
0: Prefetch engine is disabled.
1: Prefetch engine is enabled.
Bits 4–1: Unused. Read = 0000b; Write = Don't Care
Bit 0: FLBWE: FLASH 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 two.

11. 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 pull-ups 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 "14. Oscillators" on page 131** 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 264** details the use of the Watchdog Timer). Program execution begins at location 0x0000.

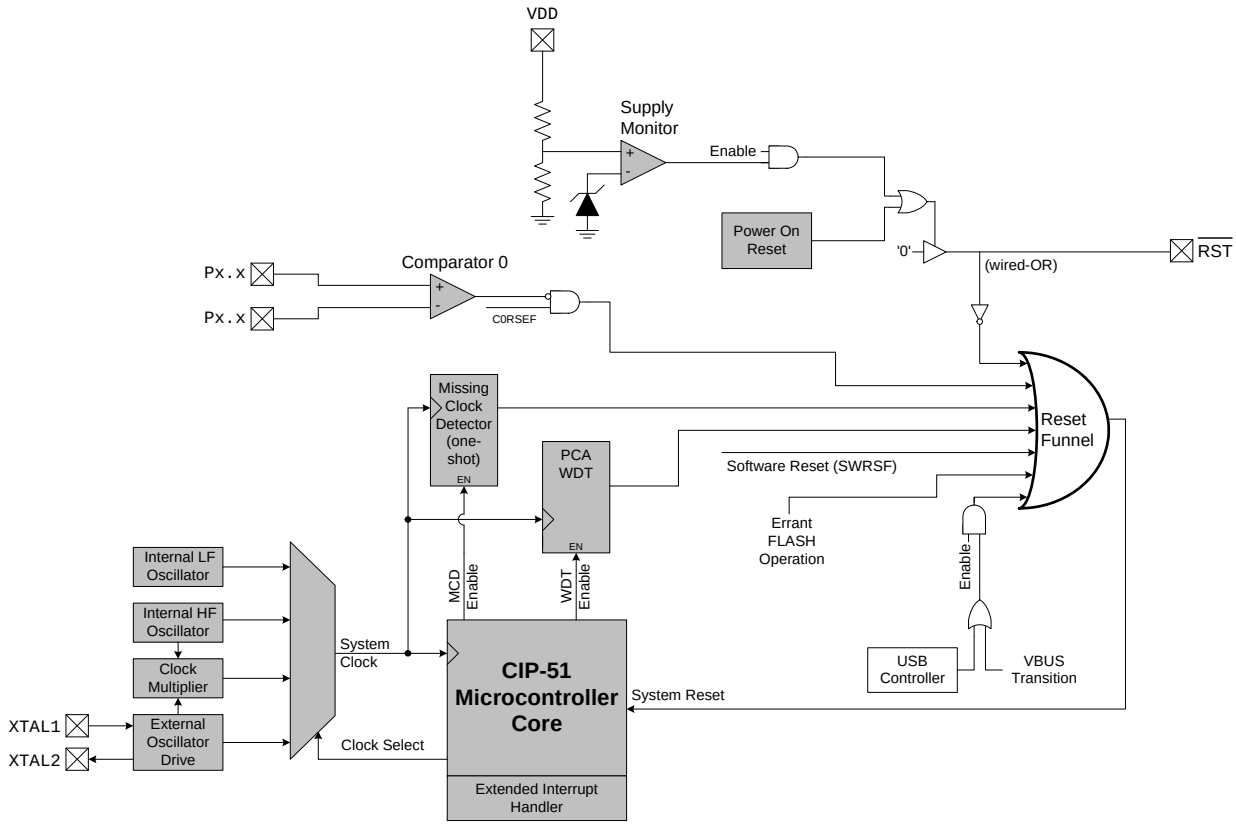


Figure 11.1. Reset Sources

11.1. 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 Power-On Reset delay ($T_{PORDelay}$) occurs before the device is released from reset; this delay is typically less than 0.3 ms. Figure 11.2. plots the power-on and V_{DD} monitor reset timing.

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.

Software can force a power-on reset by writing '1' to the PINRSF bit in register RSTSRC.

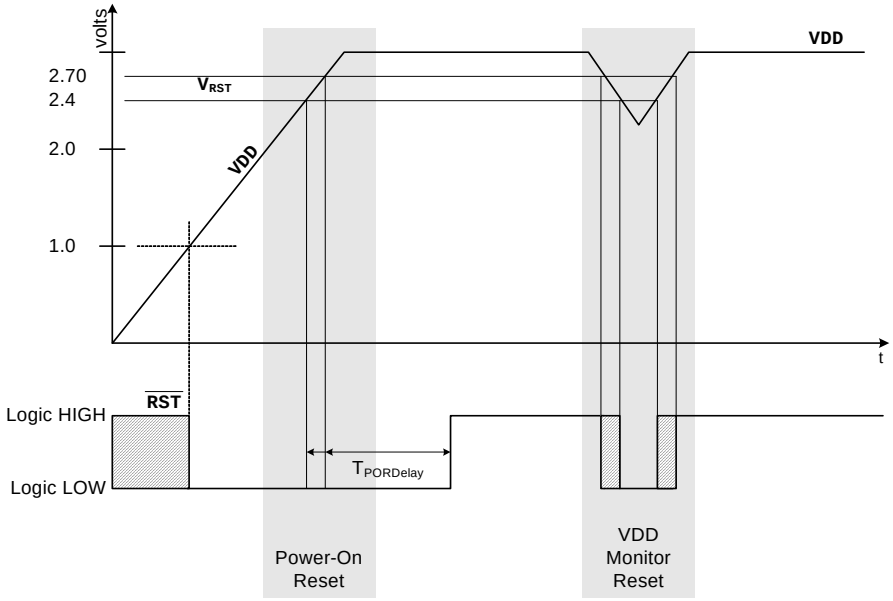


Figure 11.2. Power-On and V_{DD} Monitor Reset Timing

11.2. 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 11.2). 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 enabled and a software reset is performed, the V_{DD} monitor will still be enabled after the reset. It is strongly recommended that the V_{DD} monitor be left enabled at all times for any system that contains code to write to Flash memory.

Important Note: 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. In applications where this reset is undesirable, a delay can be implemented between enabling the V_{DD} monitor and selecting it as a reset source. The procedure for configuring the V_{DD} monitor as a reset source is shown below:

Step 1. Enable the V_{DD} monitor (VDM0CN.7 = '1').
Step 2. If desired, wait for the V_{DD} monitor to stabilize (see Table 11.1 for the V_{DD} Monitor turn-on time).
Step 3. Select the V_{DD} monitor as a reset source (RSTSRC.1 = '1').

See Figure 11.2 for V_{DD} monitor timing. See Table 11.1 for complete electrical characteristics of the V_{DD} monitor.

SFR Definition 11.1. VDM0CN: V_{DD} Monitor Control

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	SFR Address: 0xFF

Bit7: 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 11.2). 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 will generate a system reset.** See Table 11.1 for the minimum V_{DD} Monitor turn-on time. The V_{DD} Monitor is enabled following all POR resets.
0: V_{DD} Monitor Disabled.
1: V_{DD} Monitor Enabled.

Bit6: 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.

Bits5–0: Reserved. Read = Variable. Write = don't care.

11.3. External Reset

The external RST pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the RST pin generates a reset; an external pull-up and/or decoupling of the RST pin may be necessary to avoid erroneous noise-induced resets. See Table 11.1 for complete RST pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

11.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If more than 100 µs pass between rising edges on the system clock, 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 RST pin is unaffected by this reset.

11.5. 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-), a system reset is generated. 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 RST pin is unaffected by this reset.

11.6. 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 264**; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to ‘1’. The state of the RST pin is unaffected by this reset.

11.7. 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 is attempted above address 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Flash read is attempted above user code space. This occurs when a MOVC operation is attempted above address 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x7FFF (32 kB Flash devices) or 0xFBFF (64 kB Flash devices).
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see **Section “12.3. Security Options” on page 109**).
- A Flash Write or Erase is attempted when the V_{DD} monitor is not enabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the RST pin is unaffected by this reset.

11.8. 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.

11.9. USB Reset

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

1. RESET signaling is detected on the USB network. The USB Function Controller (USB0) must be enabled for RESET signaling to be detected. See **Section “16. Universal Serial Bus Controller (USB0)” on page 159** for information on the USB Function Controller.
2. The voltage on the VBUS pin matches the polarity selected by the VBPOL bit in register REG0CN. See **Section “8. Voltage Regulator (REG0)” on page 69** for details on the VBUS detection circuit.

The USBRSF bit will read ‘1’ following a USB reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

SFR Definition 11.2. RSTSRC: Reset Source

R/W	R	R/W	R/W	R	R/W	R/W	R	Reset Value
USBRSF	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xEF

Bit7: USBRSF: USB Reset Flag
0: **Read:** Last reset was not a USB reset; **Write:** USB resets disabled.
1: **Read:** Last reset was a USB reset; **Write:** USB resets enabled.

Bit6: FERROR: Flash Error Indicator.
0: Source of last reset was not a Flash read/write/erase error.
1: Source of last reset was a Flash read/write/erase error.

Bit5: 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).

Bit4: SWRSF: Software Reset Force and Flag.
0: **Read:** Source of last reset was not a write to the SWRSF bit; **Write:** No Effect.
1: **Read:** Source of last was a write to the SWRSF bit; **Write:** Forces a system reset.

Bit3: WDTRSF: Watchdog Timer Reset Flag.
0: Source of last reset was not a WDT timeout.
1: Source of last reset was a WDT timeout.

Bit2: MCDRSF: Missing Clock Detector Flag.
0: **Read:** Source of last reset was not a Missing Clock Detector timeout; **Write:** Missing Clock Detector disabled.
1: **Read:** Source of last reset was a Missing Clock Detector timeout; **Write:** Missing Clock Detector enabled; triggers a reset if a missing clock condition is detected.

Bit1: PORSF: Power-On / V_{DD} Monitor Reset Flag.
This bit is set anytime a power-on reset occurs. Writing this bit selects/deselects the V_{DD} monitor as a reset source. **Note: writing '1' to this bit before the V_{DD} monitor is enabled and stabilized can cause a system reset.** See register VDM0CN (SFR Definition 11.1).
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.

Bit0: PINRSF: HW Pin Reset Flag.
0: Source of last reset was not $\overline{\text{RST}}$ pin.
1: Source of last reset was $\overline{\text{RST}}$ pin.

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: USBRSF, C0RSEF, SWRSF, MCDRSF, PORSF.

Table 11.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{\text{RST}}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.7 to 3.6 V			0.6	V
$\overline{\text{RST}}$ Input High Voltage		0.7 x V_{DD}			V
$\overline{\text{RST}}$ Input Low Voltage				0.3 x V_{DD}	
$\overline{\text{RST}}$ Input Pull-Up Current	$\overline{\text{RST}}$ = 0.0 V		25	40	µA
V_{DD} POR Threshold (V_{RST})		2.40	2.55	2.70	V
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	220	500	µs
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	5.0			µs
Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset		15			µs
V_{DD} Monitor Turn-on Time		100			µs
V_{DD} Monitor Supply Current			20	50	µA

12. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation. Refer to Table 12.1 for complete Flash memory electrical characteristics.

12.1. 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 "23. C2 Interface" on page 271**.

To ensure the integrity of Flash contents, it is strongly recommended that the V_{DD} monitor be left enabled in any system which writes or erases Flash memory from code. It is also crucial to ensure that the FLRT bit in register FLSCL be set to '1' if a clock speed higher than 25 MHz is being used for the device.

12.1.1. 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 12.2.

12.1.2. Flash Erase Procedure

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) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY); and (2) Setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory). 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 must be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

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 the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE bit (register PSCTL).
Step 8. Clear the PSEE bit (register PSCTI).

12.1.3. Flash Write Procedure

Bytes in Flash memory can be written one byte at a time, or in groups of two. The FLBWE bit in register PFE0CN (SFR Definition 10.1) controls whether a single byte or a block of two bytes is written to Flash during a write operation. When FLBWE is cleared to ‘0’, the Flash will be written one byte at a time. When FLBWE is set to ‘1’, the Flash will be written in two-byte blocks. 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.During a single-byte write to Flash, bytes are written individually, and a Flash write will be performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is:

Step 1. Disable interrupts.
Step 2. Clear the FLBWE bit (register PFE0CN) to select single-byte write mode.
Step 3. Set the PSWE bit (register PSCTL).
Step 4. Clear the PSEE bit (register PSCTL).
Step 5. Write the first key code to FLKEY: 0xA5.
Step 6. Write the second key code to FLKEY: 0xF1.
Step 7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
Step 8. Clear the PSWE bit.
Step 9. Re-enable interrupts.

Steps 5-7 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. A Flash write block is two bytes long, from even addresses to odd addresses. Writes must be performed sequentially (i.e. addresses ending in 0b and 1b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 1b. If a byte in the block does not need to be updated in Flash, it should be written to 0xFF. The recommended procedure for writing Flash in blocks is:

Step 1. Disable interrupts.
Step 2. Set the FLBWE bit (register PFE0CN) to select block write mode.
Step 3. Set the PSWE bit (register PSCTL).
Step 4. Clear the PSEE bit (register PSCTL).
Step 5. Write the first key code to FLKEY: 0xA5.
Step 6. Write the second key code to FLKEY: 0xF1.
Step 7. Using the MOVX instruction, write the first data byte to the even block location (ending in 0b).
Step 8. Write the first key code to FLKEY: 0xA5.
Step 9. Write the second key code to FLKEY: 0xF1.
Step 10. Using the MOVX instruction, write the second data byte to the odd block location (ending in 1b).
Step 11. Clear the PSWE bit.
Step 12. Re-enable interrupts.

Steps 5–10 must be repeated for each block to be written.

Table 12.1. Flash Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F340/2/4/6/A/C/D*	65536*			Bytes
	C8051F341/3/5/7/8/9/B	32768			Bytes
Endurance		20k	100k		Erase/Write
Erase Cycle Time	25 MHz System Clock	10	15	20	ms
Write Cycle Time	25 MHz System Clock	40	55	70	µs

***Note:** 1024 bytes at location 0xFC00 to 0xFFFF are reserved.

12.2. 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 using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

12.3. 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* 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), 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 also locked when any other Flash pages are locked. See example below.

Security Lock Byte:	11111101b
1’s Complement:	00000010b
Flash pages locked:	3 (2 + Flash Lock Byte Page)
	First two pages of Flash: 0x0000 to 0x03FF
Addresses locked:	Flash Lock Byte Page: (0xFA00 to 0xFBFF for 64k devices; 0x7E00 to 0x7FFF for 32k devices)

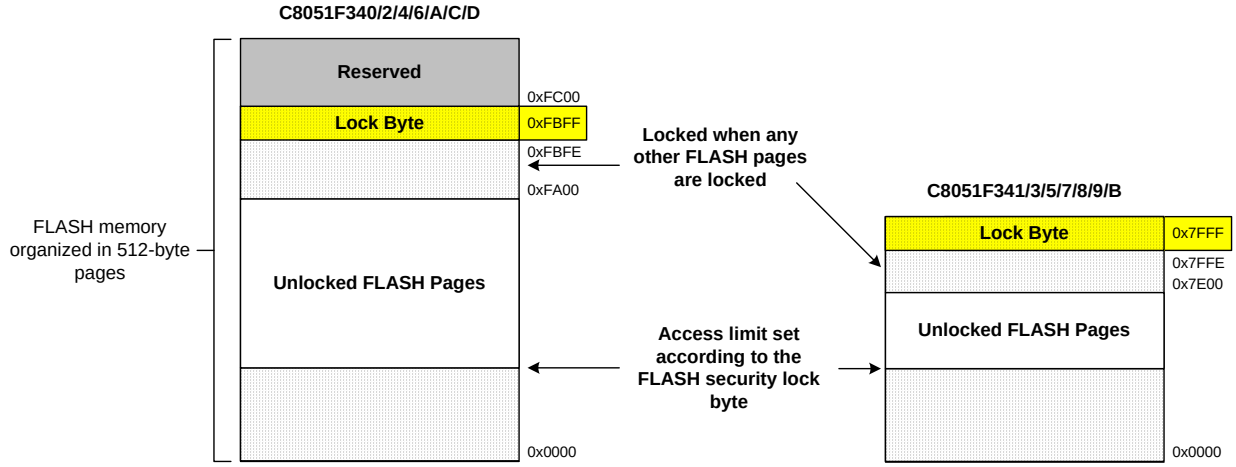


Figure 12.1. Flash Program Memory Map and Security Byte

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.

Accessing FLASH **from the C2 debug interface**:

1. Any unlocked page may be read, written, or erased.
2. Locked pages cannot be read, written, or erased.
3. The page containing the Lock Byte may be read, written, or erased if it is unlocked.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking FLASH pages (changing '0's to '1's in the Lock Byte) requires the C2 Device Erase command, which erases all FLASH pages including the page containing the Lock Byte and the Lock Byte itself.
7. The Reserved Area cannot be read, written, or erased.

Accessing FLASH **from user firmware** executing on an **unlocked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Locked pages cannot be read, written, or erased.
3. The page containing the Lock Byte cannot be erased. It may be read or written only if it is unlocked.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking FLASH pages (changing '0's to '1's in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a FLASH Error device reset.

Accessing FLASH **from user firmware** executing on a **locked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Any locked page except the page containing the Lock Byte may be read, written, or erased.
3. The page containing the Lock Byte cannot be erased. It may only be read or written.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing '1's to '0's in the Lock Byte) is not permitted.
6. Unlocking FLASH pages (changing '0's to '1's in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a FLASH Error device reset.

SFR Definition 12.1. PSCTL: Program Store R/W Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	Reserved	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8F

Bits7–3: Unused: Read = 00000b. Write = don't care.
Bit2: Reserved. Read = 0b. Must Write = 0b.
Bit1: PSEE: Program Store Erase Enable
Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), 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.
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.
Bit0: 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 Flash location should be erased before writing data.
0: Writes to Flash program memory disabled.
1: Writes to Flash program memory enabled; the MOVX write instruction targets Flash memory.

SFR Definition 12.2. FLKEY: Flash Lock and Key

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:
								0xB7

Bits–0: FLKEY: Flash Lock and Key Register
Write:
This register must be written to before Flash writes or erases can be performed. Flash remains locked until this register is written to with the following key codes: 0xA5, 0xF1. The timing of the writes does not matter, as long as the codes are written in order. The key codes must be written for each Flash write or erase operation. Flash will be locked until the next system reset if the wrong codes are written or if a Flash operation is attempted before the codes have been written correctly.
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.

SFR Definition 12.3. FLSCL: Flash Scale

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FOSE	Reserved	Reserved	FLRT	Reserved	Reserved	Reserved	Reserved	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB6

Bits7: FOSE: Flash One-shot Enable
This bit enables the Flash read one-shot. When the Flash one-shot disabled, the Flash sense amps are enabled for a full clock cycle during Flash reads. At system clock frequencies below 10 MHz, disabling the Flash one-shot will increase system power consumption.
0: Flash one-shot disabled.
1: Flash one-shot enabled.

Bits6–5: RESERVED. Read = 00b. Must Write 00b.

Bit 4: FLRT: FLASH Read Time.
This bit should be programmed to the smallest allowed value, according to the system clock speed.
0: SYSCLK <= 25 MHz.
1: SYSCLK <= 48 MHz.

Bits3–0: RESERVED. Read = 0000b. Must Write 0000b.

13. External Data Memory Interface and On-Chip XRAM

4k Bytes (C8051F340/2/4/6/A/C/D) or 2k Bytes (C8051F341/3/5/7/8/9/B) of RAM are included on-chip, and mapped into the external data memory space (XRAM). The 1k Bytes of USB FIFO space can also be mapped into XRAM address space for additional general-purpose data storage. Additionally, an External Memory Interface (EMIF) is available on the C8051F340/1/4/5/8/C 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 13.1). Note: the MOVX instruction can also be used for writing to the FLASH memory. See **Section "12. Flash Memory" on page 107** for details. The MOVX instruction accesses XRAM by default.

13.1. 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.

13.1.1. 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.

13.1.2. 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
```

13.2. Accessing USB FIFO Space

The C8051F34x devices include 1k of RAM which functions as USB FIFO space. Figure 13.1 shows an expanded view of the FIFO space and user XRAM. FIFO space is normally accessed via USB FIFO registers; see **Section "16.5. FIFO Management" on page 167** for more information on accessing these FIFOs. The MOVX instruction should not be used to load or modify USB data in the FIFO space.

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

To access the FIFO RAM directly using MOVX instructions, the following conditions must be met: (1) the USBFAE bit in register EMI0CF must be set to '1', and (2) the USB clock must be greater than or equal to twice the SYSCLK (USBCLK $\geq$ 2 x SYSCLK). When this bit is set, the USB FIFO space is mapped into XRAM space at addresses 0x0400 to 0x07FF. The normal XRAM (on-chip or external) at the same addresses cannot be accessed when the USBFAE bit is set to '1'.

Important Note: The USB clock must be active when accessing FIFO space.

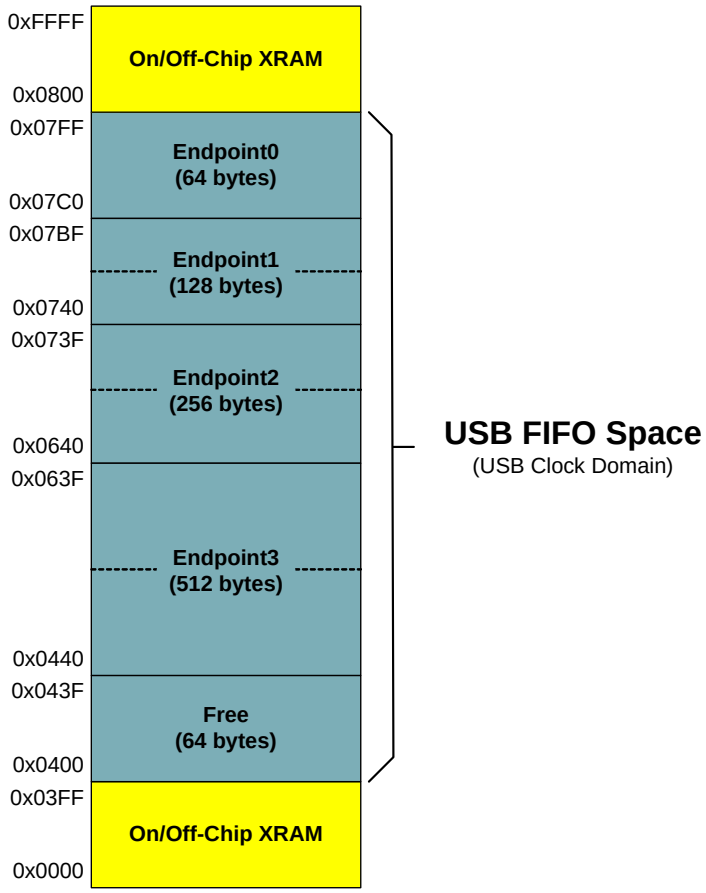


Figure 13.1. USB FIFO Space and XRAM Memory Map with USBFAE set to '1'

13.3. 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 13.2.

13.4. Port Configuration

The External Memory Interface appears on Ports 4, 3, 2, and 1 when it is used for off-chip memory access. When the EMIF is used, the Crossbar should be configured to skip over the control lines P1.7 ($\overline{WR}$), P1.6 ($\overline{RD}$), and if multiplexed mode is selected P1.3 (ALE) using the P1SKIP register. For more information about configuring the Crossbar, see **Section “Figure 15.1. Port I/O Functional Block Diagram (Port 0 through Port 3)” on page 142**.

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 “15. Port Input/ Output” on page 142** 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.

SFR Definition 13.1. EMI0CN: External Memory Interface Control

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	

SFR Address: 0xAA

Bits7–0: 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 13.2. EMI0CF: External Memory Configuration

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

SFR Address: 0x85

Bit7: Unused. Read = 0b. Write = don't care.

Bit6: USBFAE: USB FIFO Access Enable.
0: USB FIFO RAM not available through MOVX instructions.
1: USB FIFO RAM available using MOVX instructions. The 1k of USB RAM will be mapped in XRAM space at addresses 0x0400 to 0x07FF. **The USB clock must be active and greater than or equal to twice the SYSCLK (USBCLK ≥ 2 x SYSCLK) to access this area with MOVX instructions.**

Bit5: Unused. Read = 0b. Write = don't care.

Bit4: 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).

Bits3–2: 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 on-chip XRAM boundary are directed on-chip. Accesses above the on-chip XRAM 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 on-chip XRAM boundary are directed on-chip. Accesses above the on-chip XRAM 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.

Bits1–0: 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.

13.5. 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.

13.5.1. 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 13.2.

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 $\overline{RD}$ or $\overline{WR}$ is asserted.

See **Section "13.7.2. Multiplexed Mode" on page 127** for more information.

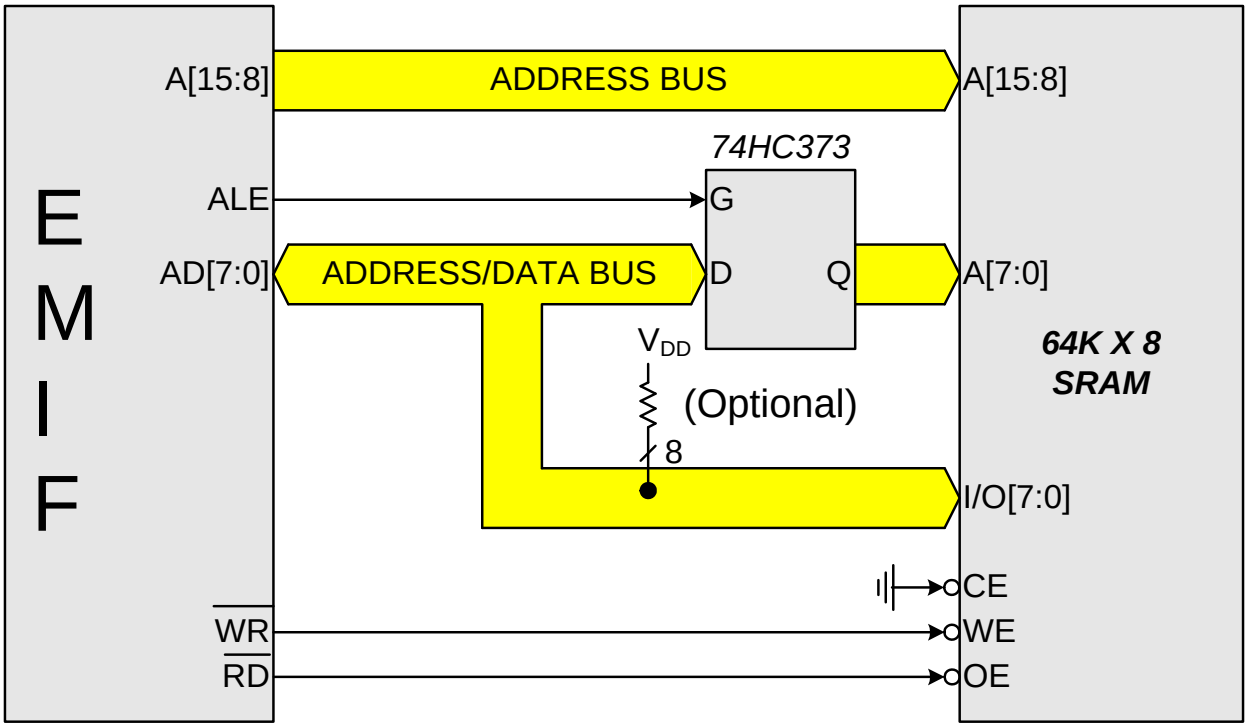


Figure 13.2. Multiplexed Configuration Example

13.5.2. 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 13.3. See **Section "13.7.1. Non-multiplexed Mode" on page 124** for more information about Non-multiplexed operation.

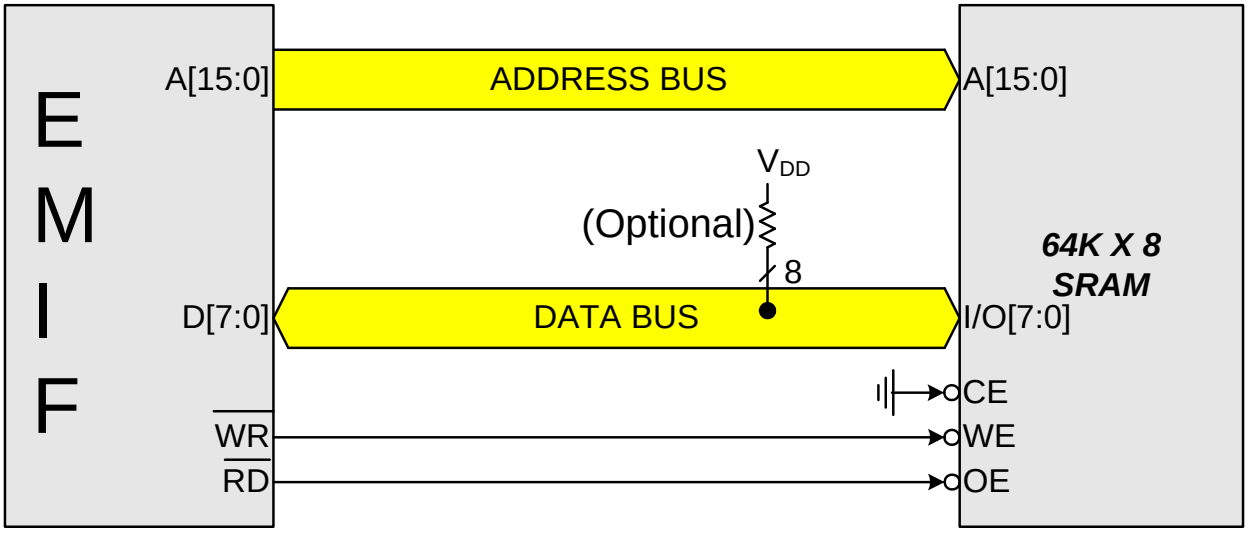


Figure 13.3. Non-multiplexed Configuration Example

13.6. Memory Mode Selection

The external data memory space can be configured in one of four modes, shown in Figure 13.4, based on the EMIF Mode bits in the EMI0CF register (SFR Definition 13.2). These modes are summarized below. More information about the different modes can be found in **Section "13.7. Timing" on page 122**.

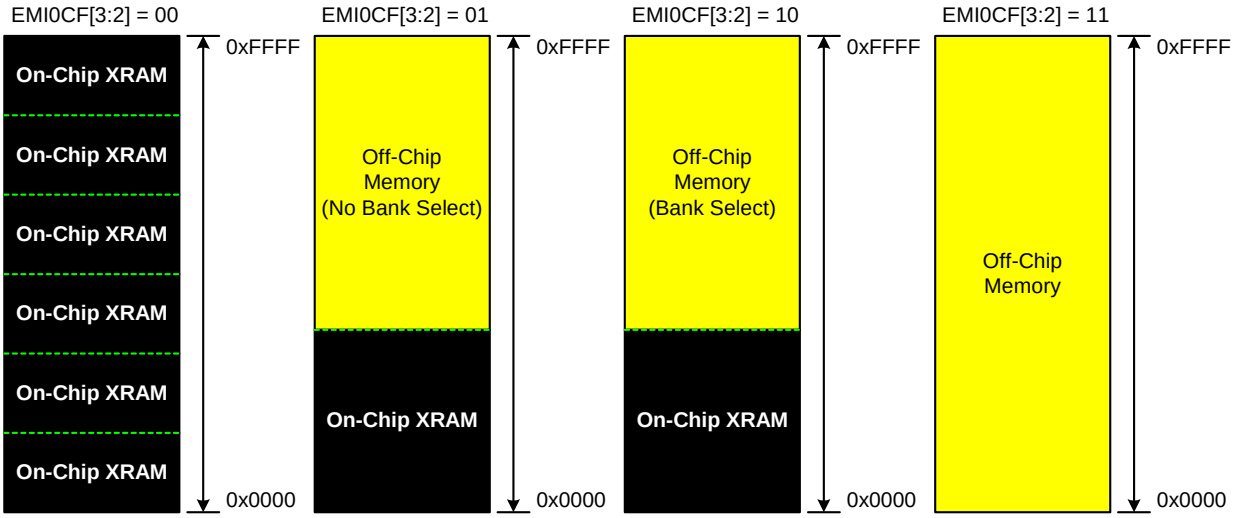


Figure 13.4. EMIF Operating Modes

13.6.1. 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 2k or 4k boundaries (depending on the RAM available on the device). As an example, the addresses 0x1000 and 0x2000 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.

13.6.2. 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.

13.6.3. 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.

13.6.4. 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.

13.7. Timing

The timing parameters of the External Memory Interface can be configured to enable connection to devices having different setup and hold time requirements. The Address Setup time, Address Hold time, $\overline{RD}$ and $\overline{WR}$ strobe widths, and in multiplexed mode, the width of the ALE pulse are all programmable in units of SYSCLK periods through EMI0TC, shown in SFR Definition 13.3, and EMI0CF[1:0].

The timing for an off-chip MOVX instruction can be calculated by adding 4 SYSCLK cycles to the timing parameters defined by the EMI0TC register. Assuming non-multiplexed operation, the minimum execution time for an off-chip XRAM operation is 5 SYSCLK cycles (1 SYSCLK for $\overline{RD}$ or $\overline{WR}$ pulse + 4 SYSCLKs). For multiplexed operations, the Address Latch Enable signal will require a minimum of 2 additional SYSCLK cycles. Therefore, the minimum execution time for an off-chip XRAM operation in multiplexed mode is 7 SYSCLK cycles (2 for $\overline{ALE}$ + 1 for $\overline{RD}$ or $\overline{WR}$ + 4). The programmable setup and hold times default to the maximum delay settings after a reset. Table 13.1 lists the AC parameters for the External Memory Interface, and Figure 13.5 through Figure 13.10 show the timing diagrams for the different External Memory Interface modes and MOVX operations.

SFR Definition 13.3. EMI0TC: External Memory Timing Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EAS1	EAS0	EWR3	EWR2	EWR1	EWR0	EAH1	EAH0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x84

Bits7–6: 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.

Bits5–2: EWR3–0: EMIF $\overline{WR}$ and $\overline{RD}$ Pulse-Width Control Bits.
0000: $\overline{WR}$ and $\overline{RD}$ pulse width = 1 SYSCLK cycle.
0001: $\overline{WR}$ and $\overline{RD}$ pulse width = 2 SYSCLK cycles.
0010: $\overline{WR}$ and $\overline{RD}$ pulse width = 3 SYSCLK cycles.
0011: $\overline{WR}$ and $\overline{RD}$ pulse width = 4 SYSCLK cycles.
0100: $\overline{WR}$ and $\overline{RD}$ pulse width = 5 SYSCLK cycles.
0101: $\overline{WR}$ and $\overline{RD}$ pulse width = 6 SYSCLK cycles.
0110: $\overline{WR}$ and $\overline{RD}$ pulse width = 7 SYSCLK cycles.
0111: $\overline{WR}$ and $\overline{RD}$ pulse width = 8 SYSCLK cycles.
1000: $\overline{WR}$ and $\overline{RD}$ pulse width = 9 SYSCLK cycles.
1001: $\overline{WR}$ and $\overline{RD}$ pulse width = 10 SYSCLK cycles.
1010: $\overline{WR}$ and $\overline{RD}$ pulse width = 11 SYSCLK cycles.
1011: $\overline{WR}$ and $\overline{RD}$ pulse width = 12 SYSCLK cycles.
1100: $\overline{WR}$ and $\overline{RD}$ pulse width = 13 SYSCLK cycles.
1101: $\overline{WR}$ and $\overline{RD}$ pulse width = 14 SYSCLK cycles.
1110: $\overline{WR}$ and $\overline{RD}$ pulse width = 15 SYSCLK cycles.
1111:$\overline{WR}$ and $\overline{RD}$ pulse width = 16 SYSCLK cycles.

Bits1–0: 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.

13.7.1. Non-multiplexed Mode

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

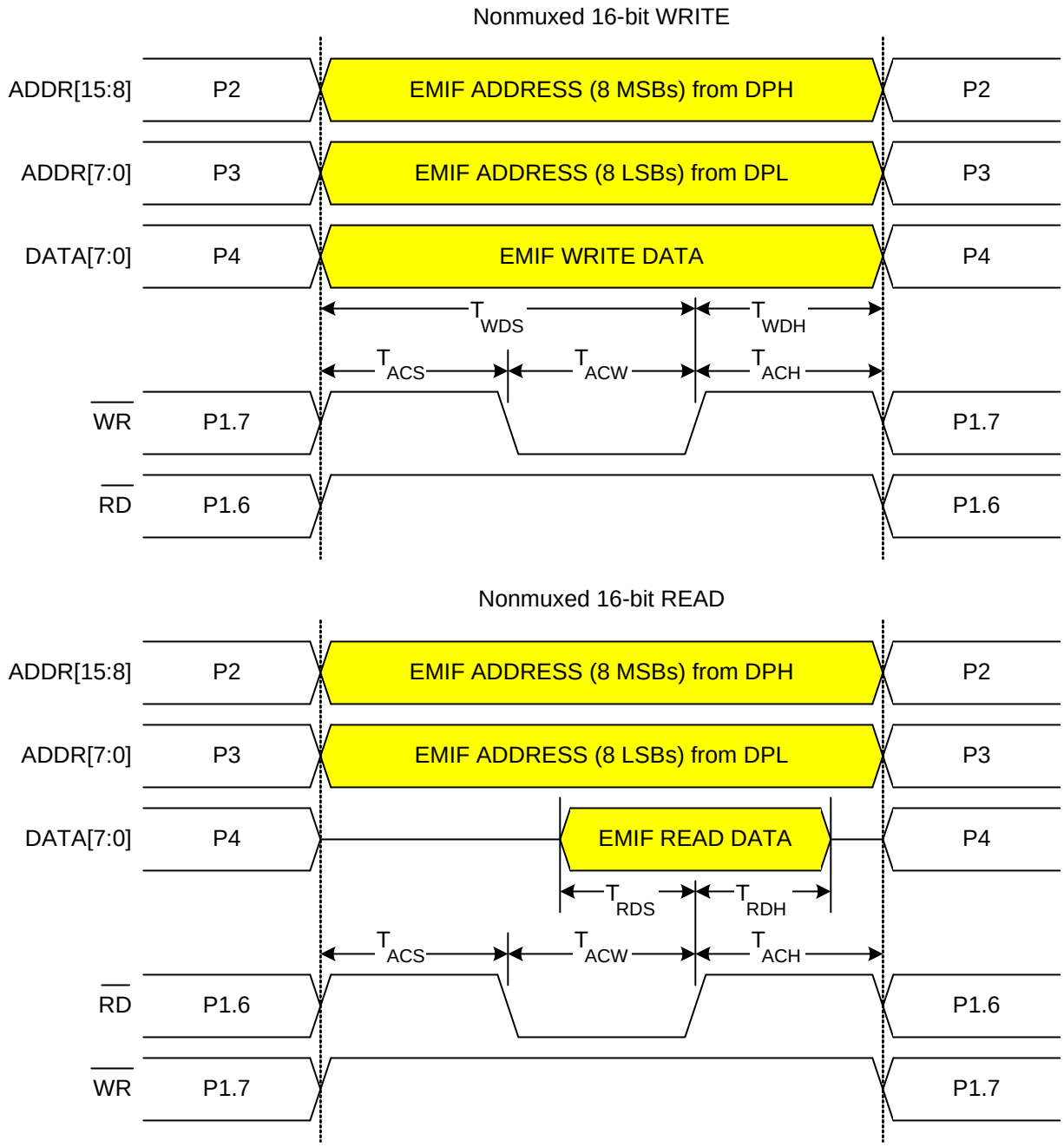


Figure 13.5. Non-multiplexed 16-bit MOVX Timing

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

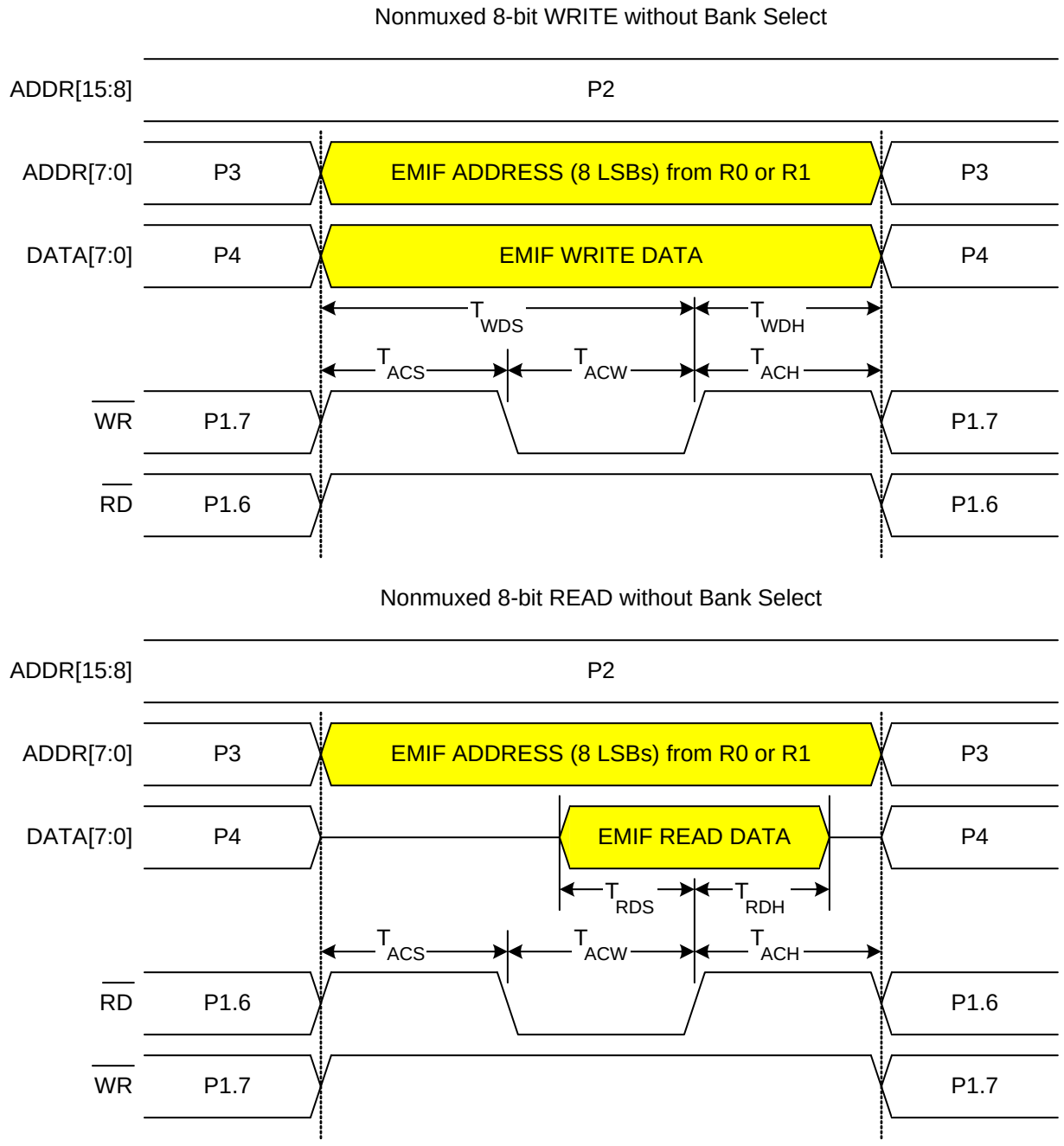


Figure 13.6. Non-multiplexed 8-bit MOVX without Bank Select Timing

13.7.1.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = ‘110’.

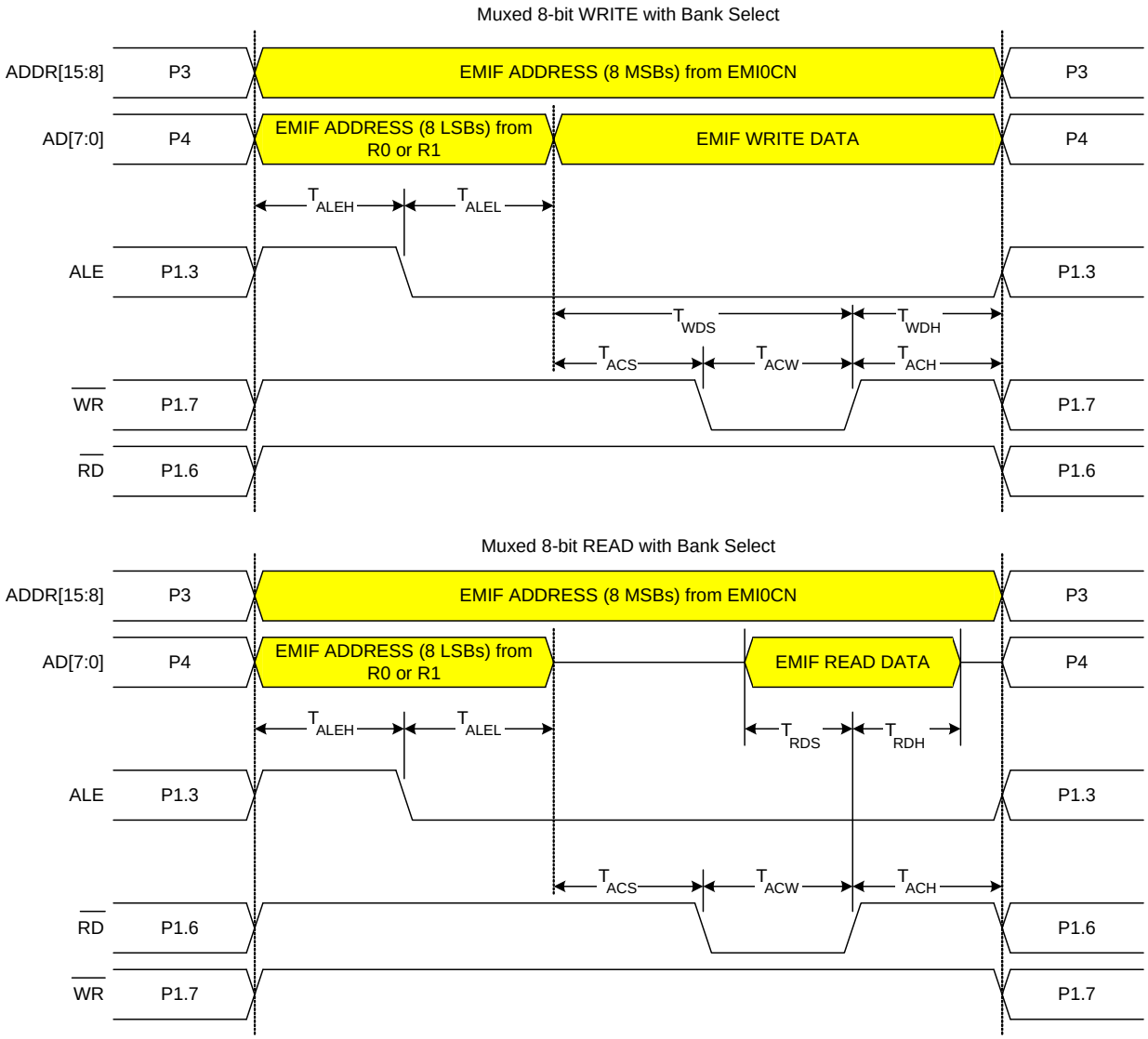


Figure 13.7. Non-multiplexed 8-bit MOVX with Bank Select Timing

13.7.2. Multiplexed Mode

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

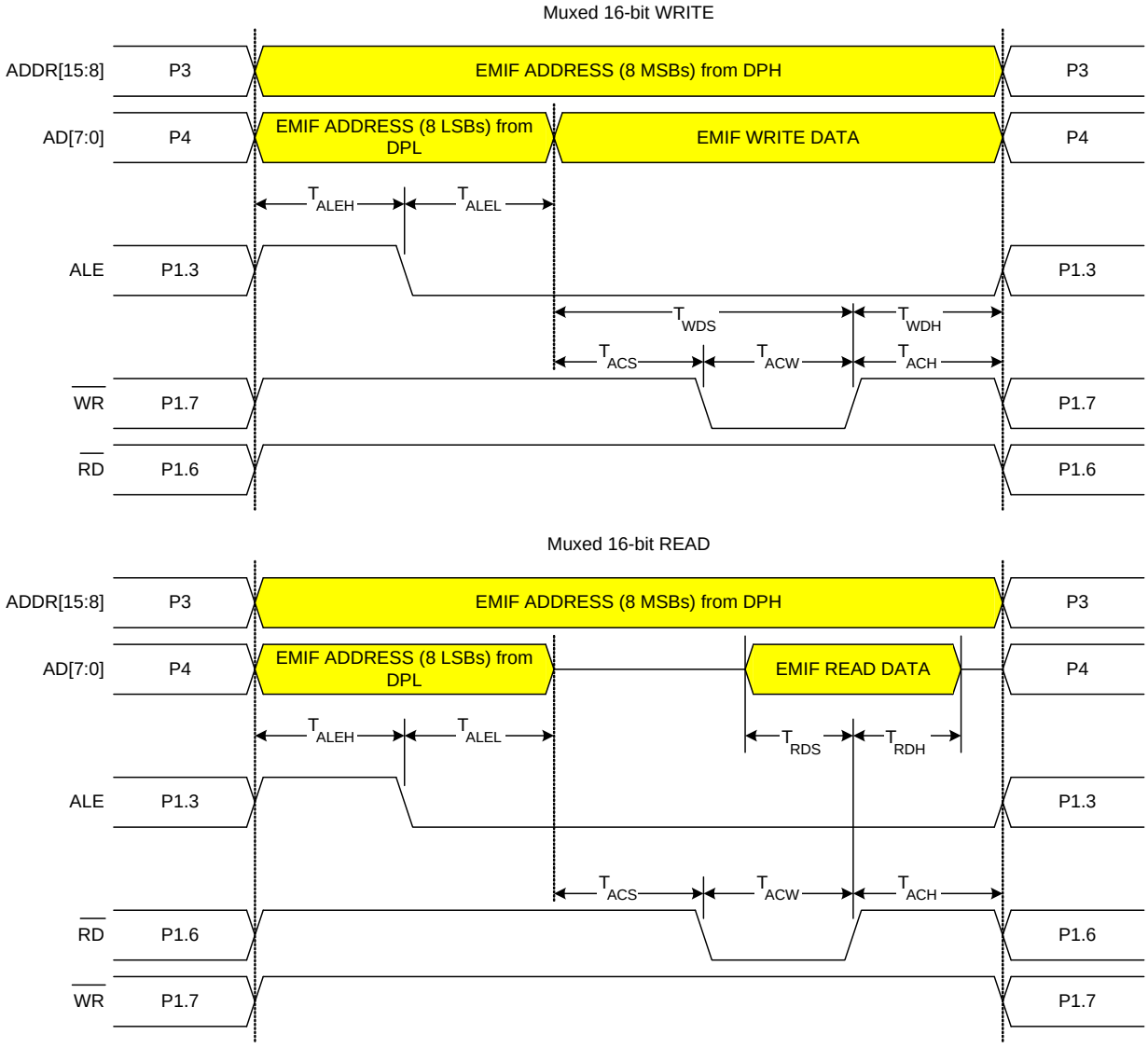


Figure 13.8. Multiplexed 16-bit MOVX Timing

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

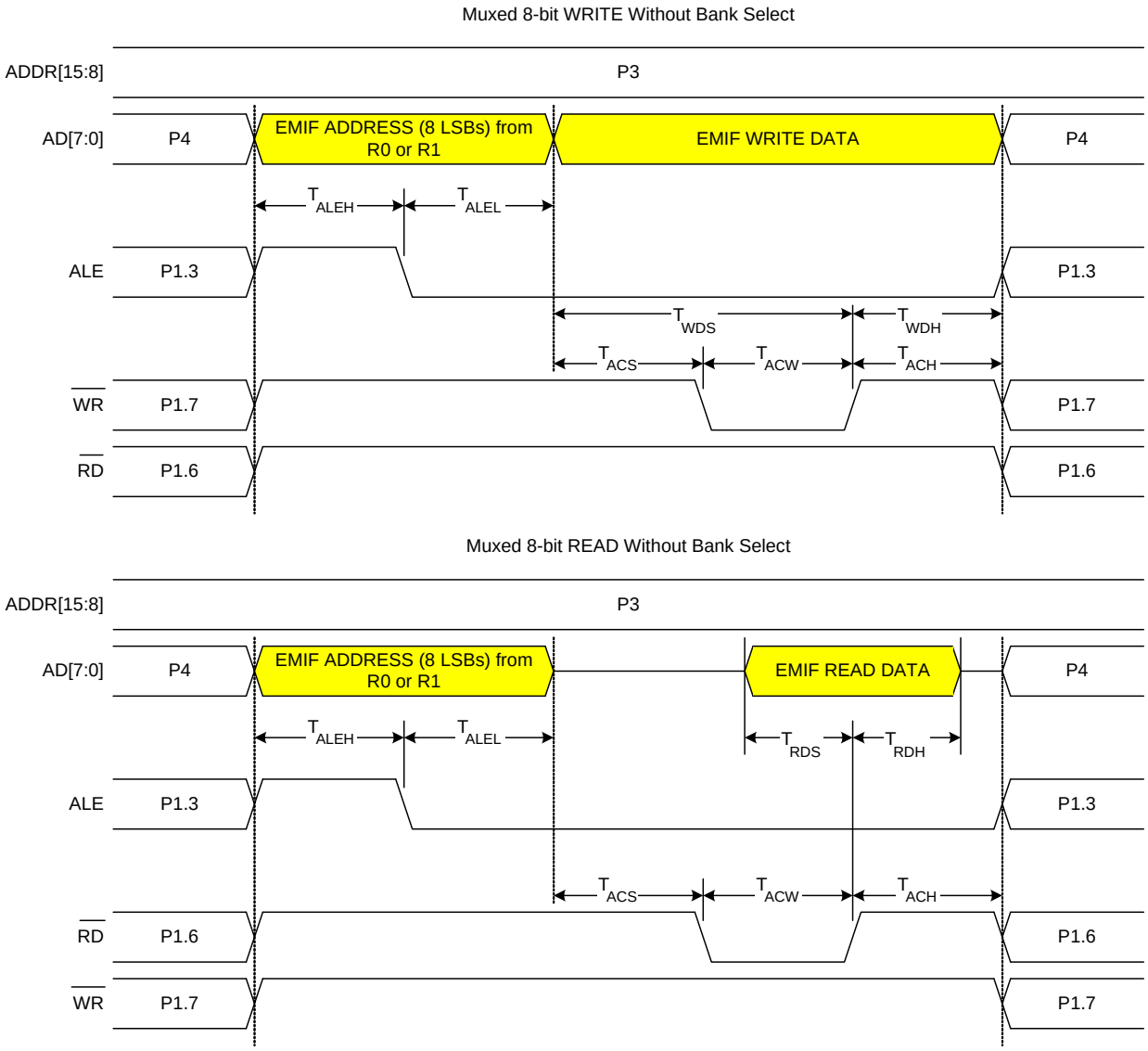


Figure 13.9. Multiplexed 8-bit MOVX without Bank Select Timing

13.7.2.3.8-bit MOVX with Bank Select: EMI0CF[4:2] = '010'.

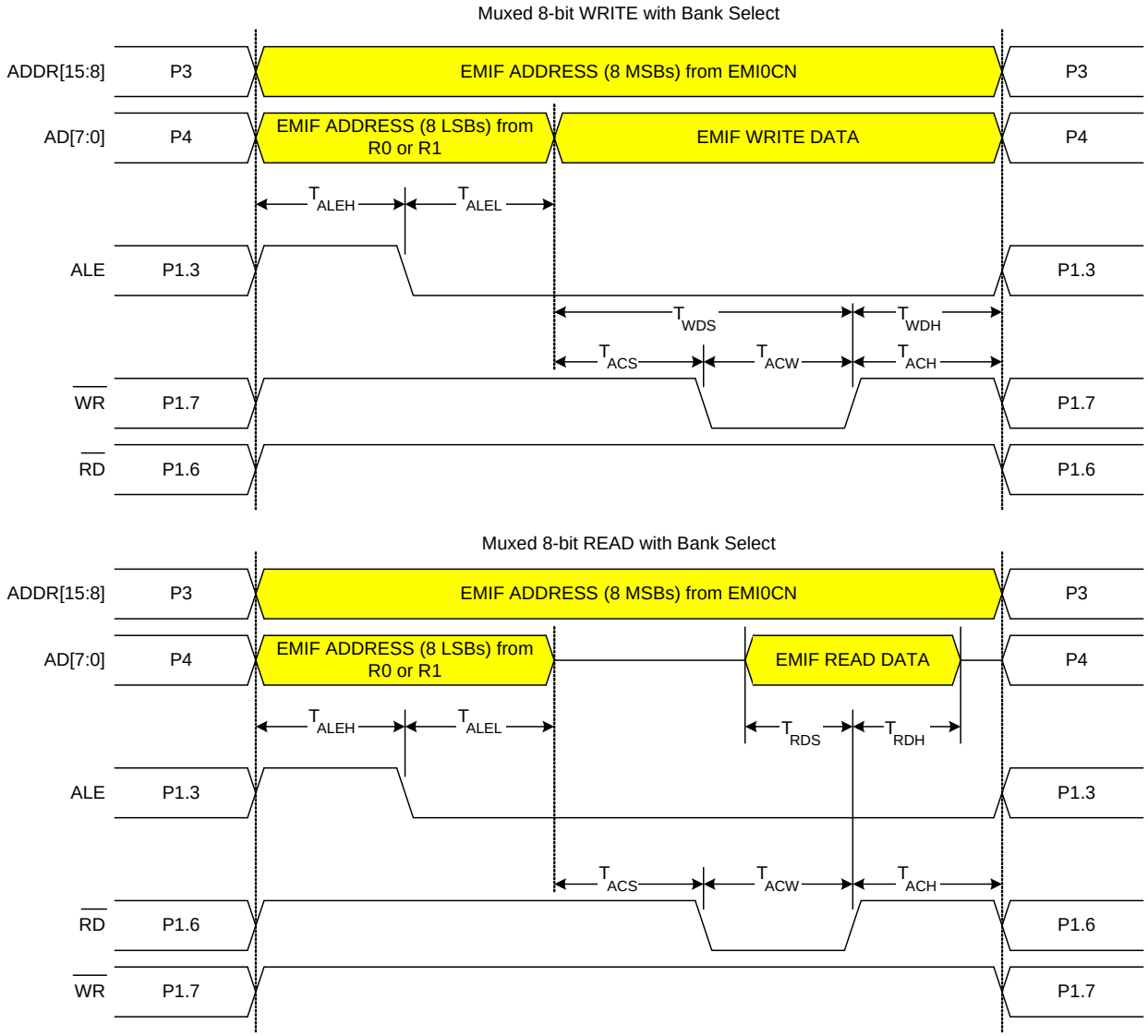


Figure 13.10. Multiplexed 8-bit MOVX with Bank Select Timing

Table 13.1. 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).

14. Oscillators

C8051F34x devices include a programmable internal high-frequency oscillator, a programmable internal low-frequency oscillator (C8051F340/1/2/3/4/5/8/9/A/B/C/D), an external oscillator drive circuit, and a 4x Clock Multiplier. The internal high-frequency and low-frequency oscillators can be enabled/disabled and adjusted using the special function registers, as shown in Figure 14.1. The system clock (SYSCLK) can be derived from either of the internal oscillators, the external oscillator circuit, or the 4x Clock Multiplier divided by 2. The USB clock (USBCLK) can be derived from the internal oscillator, external oscillator, or 4x Clock Multiplier. Oscillator electrical specifications are given in Table 14.1.

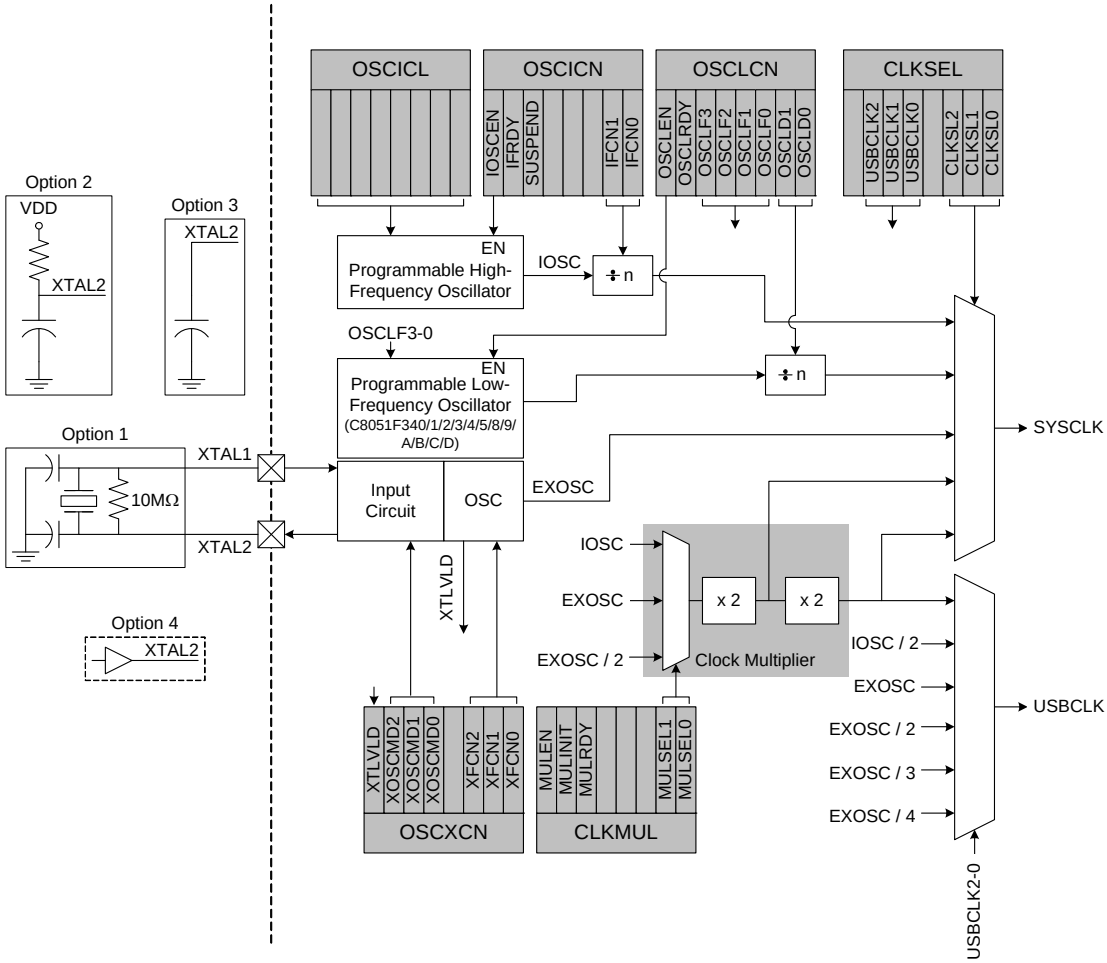


Figure 14.1. Oscillator Diagram

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

All C8051F34x devices include a programmable internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be programmed via the OSCICL register shown in SFR Definition 14.2. The OSCICL register is factory calibrated to obtain a 12 MHz internal oscillator frequency. Electrical specifications for the precision internal oscillator are given in Table 14.1 on page 141. 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.

14.1.1. Internal H-F Oscillator Suspend Mode

The internal high-frequency oscillator may be placed in Suspend mode by writing '1' to the SUSPEND bit in register OSCICN. In Suspend mode, the internal H-F oscillator is stopped until a non-idle USB event is detected (**Section 16**) or VBUS matches the polarity selected by the VBPOL bit in register REG0CN (**Section 8.2**). Note that the USB transceiver can still detect USB events when it is disabled.

SFR Definition 14.1. OSCICN: Internal H-F Oscillator Control

R/W	R	R/W	R	R/W	R/W	R/W	R/W	Reset Value
IOSCEN	IFRDY	SUSPEND	-	-	-	IFCN1	IFCN0	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

Bit7: IOSCEN: Internal H-F Oscillator Enable Bit.
0: Internal H-F Oscillator Disabled.
1: Internal H-F Oscillator Enabled.

Bit6: IFRDY: Internal H-F Oscillator Frequency Ready Flag.
0: Internal H-F Oscillator is not running at programmed frequency.
1: Internal H-F Oscillator is running at programmed frequency.

Bit5: SUSPEND: Force Suspend
Writing a '1' to this bit will force the internal H-F oscillator to be stopped. The oscillator will be re-started on the next non-idle USB event (i.e., RESUME signaling) or VBUS interrupt event (see SFR Definition 8.1).

Bits4–2: UNUSED. Read = 000b, Write = don't care.

Bits1–0: IFCN1–0: Internal H-F Oscillator Frequency Control.
00: SYSCLK derived from Internal H-F Oscillator divided by 8.
01: SYSCLK derived from Internal H-F Oscillator divided by 4.
10: SYSCLK derived from Internal H-F Oscillator divided by 2.
11: SYSCLK derived from Internal H-F Oscillator divided by 1.

SFR Definition 14.2. OSCICL: Internal H-F Oscillator Calibration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	OSCCAL					Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB3

Bits4–0: OSCCAL: Oscillator Calibration Value
These bits determine the internal H-F oscillator period. When set to 00000b, the oscillator operates at its fastest setting. When set to 11111b, the oscillator operates at is slowest setting. The contents of this register are factory calibrated to produce a 12 MHz internal oscillator frequency.

Note: The contents of this register are undefined when Clock Recovery is enabled. See Section "16.4. USB Clock Configuration" on page 166 for details on Clock Recovery.

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

The C8051F340/1/2/3/4/5/8/9/C/D devices include a programmable internal oscillator which operates at 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 14.3). Additionally, the OSCLF bits (OSCLCN5:2) can be used to adjust the oscillator's output frequency.

14.2.1. 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 period.

SFR Definition 14.3. OSCLCN: Internal L-F Oscillator Control

R/W	R	R/W	R	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	SFR Address:
								0x86

Bit7: OSCLEN: Internal L-F Oscillator Enable.
0: Internal L-F Oscillator Disabled.
1: Internal L-F Oscillator Enabled.

Bit6: OSCLRDY: Internal L-F Oscillator Ready Flag.
0: Internal L-F Oscillator frequency not stabilized.
1: Internal L-F Oscillator frequency stabilized.

Bits5–2: 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.

Bits1–0: OSCLD[1:0]: Internal L-F Oscillator Divider Select.
00: Divide by 8 selected.
01: Divide by 4selected.
10: Divide by 2 selected.
11: Divide by 1 selected.

14.3. 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 14.1. 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 14.1. 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 14.4)

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 P0.6 and P0.7 (C8051F340/1/4/5/8) or P0.2 and P0.3 (C8051F342/3/6/7/9/A/B) are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.7 (C8051F340/1/4/5/8) or P0.3 (C8051F342/3/6/7/9/A/B) is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see **Section "15.1. Priority Crossbar Decoder" on page 144** 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 "15.2. Port I/O Initialization" on page 147** for details on Port input mode selection.

14.3.1. Clocking Timers Directly Through the External Oscillator

The external oscillator source divided by eight is a clock option for the timers (**Section "21. Timers" on page 235**) and the Programmable Counter Array (PCA) (**Section "22. Programmable Counter Array (PCA0)" on page 255**). When the external oscillator is used to clock these peripherals, but is not used as the system clock, the external oscillator frequency must be less than or equal to the system clock frequency. In this configuration, the clock supplied to the peripheral (external oscillator / 8) is synchronized with the system clock; the jitter associated with this synchronization is limited to ±0.5 system clock cycles.

14.3.2. 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 14.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 14.4 (OSCXCN register). For example, a 12 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is first enabled, the oscillator amplitude detection circuit requires a settling time to achieve proper bias. Introducing a delay of 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. Enable the external oscillator.
Step 2. Wait at least 1 ms.
Step 3. Poll for XTLVLD => '1'.
Step 4. Switch the system clock to the external oscillator.

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.

14.3.3. 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 14.1, 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:

$$f = \frac{1.23(10^3)}{RC} = \frac{1.23(10^3)}{[246 \times 50]} = 0.1\ \text{MHz} = 100\ \text{kHz}$$

Referring to the table in SFR Definition 14.4, the required XFCN setting is 010b. Programming XFCN to a higher setting in RC mode will improve frequency accuracy at an increased external oscillator supply current.

14.3.4. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 14.1, 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 C = 50 pF:

$$f = \frac{KF}{(C \times V_{DD})} = \frac{KF}{(50\ \text{x}\ 3)\text{MHz}}$$

$$f = \frac{KF}{150\ \text{MHz}}$$

If a frequency of roughly 150 kHz is desired, select the K Factor from the table in SFR Definition 14.4 as KF = 22:

$$f = \frac{22}{150} = 0.146\ \text{MHz, or } 146\ \text{kHz}$$

Therefore, the XFCN value to use in this example is 011b.

SFR Definition 14.4. OSCXCN: External Oscillator Control

R	R/W	R/W	R/W	R	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	-	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xB1

Bit7: XTLVLD: Crystal Oscillator Valid Flag.
(Read only when XOSCMD = 11x.)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits6–4: 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.

Bit3: RESERVED. Read = 0, Write = don't care.

Bits2–0: XFCN2–0: External Oscillator Frequency Control Bits.
000-111: See table below:

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

CRYSTAL MODE (Circuit from Figure 14.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal or resonator frequency.

RC MODE (Circuit from Figure 14.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
$f = 1.23(10^3) / (R \times C)$, where
f = frequency of clock in MHz
C = capacitor value in pF
R = Pull-up resistor value in kΩ

C MODE (Circuit from Figure 14.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
$f = KF / (C \times V_{DD})$, where
f = frequency of clock in MHz
C = capacitor value the XTAL2 pin in pF
V_{DD} = Power Supply on MCU in volts

14.4. 4x Clock Multiplier

The 4x Clock Multiplier allows a 12 MHz oscillator to generate the 48 MHz clock required for Full Speed USB communication (see **Section "16.4. USB Clock Configuration" on page 166**). A divided version of the Multiplier output can also be used as the system clock. C8051F340/1/2/3 devices can use the 48 MHz Clock Multiplier output as system clock. See Table 3.1, "Global DC Electrical Characteristics," on page 25 for system clock frequency specifications. See **Section 14.5** for details on system clock and USB clock source selection.

The 4x Clock Multiplier is configured via the CLKMUL register. The procedure for configuring and enabling the 4x Clock Multiplier is as follows:

1. Reset the Multiplier by writing 0x00 to register CLKMUL.
2. Select the Multiplier input source via the MULSEL bits.
3. Enable the Multiplier with the MULEN bit (CLKMUL | = 0x80).
4. Delay for >5 µs.
5. Initialize the Multiplier with the MULINIT bit (CLKMUL | = 0xC0).
6. Poll for MULRDY => '1'.

Important Note: When using an external oscillator as the input to the 4x Clock Multiplier, the external source must be enabled and stable before the Multiplier is initialized. See Section 14.5 for details on selecting an external oscillator source.

SFR Definition 14.5. CLKMUL: Clock Multiplier Control

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
MULEN	MULINIT	MULRDY	-	-	-	MULSEL		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address
								0xB9

Bit7: MULEN: Clock Multiplier Enable
0: Clock Multiplier disabled.
1: Clock Multiplier enabled.

Bit6: MULINIT: Clock Multiplier Initialize
This bit should be a '0' when the Clock Multiplier is enabled. Once enabled, writing a '1' to this bit will initialize the Clock Multiplier. The MULRDY bit reads '1' when the Clock Multiplier is stabilized.

Bit5: MULRDY: Clock Multiplier Ready
This read-only bit indicates the status of the Clock Multiplier.
0: Clock Multiplier not ready.
1: Clock Multiplier ready (locked).

Bits4–2: Unused. Read = 000b; Write = don't care.

Bits1–0: MULSEL: Clock Multiplier Input Select
These bits select the clock supplied to the Clock Multiplier.

MULSEL	Selected Clock
00	Internal Oscillator
01	External Oscillator
10	External Oscillator / 2
11	RESERVED

14.5. System and USB Clock Selection

The internal oscillator requires little start-up time and may be selected as the system or USB clock immediately following the OSCICN write that enables the internal oscillator. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use. 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.

14.5.1. System Clock Selection

The CLKSL[1:0] bits in register CLKSEL select which oscillator source is used as the system clock. CLKSL[1:0] must be set to 01b for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals (timers, PCA, USB) when the internal oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal oscillator, external oscillator, and 4x Clock Multiplier so long as the selected oscillator is enabled and has settled. C8051F340/1/2/3 devices can use the 48 MHz Clock Multiplier output as system clock. See Table 3.1, "Global DC Electrical Characteristics," on page 25 for system clock frequency specifications. When operating with a system clock of greater than 25 MHz (up to 48 MHz), the FLRT bit (FLSCL.4) should be set to '1'. See **Section "10. Prefetch Engine" on page 99** for more details.

14.5.2. USB Clock Selection

The USBCLK[2:0] bits in register CLKSEL select which oscillator source is used as the USB clock. The USB clock may be derived from the 4x Clock Multiplier output, a divided version of the internal oscillator, or a divided version of the external oscillator. Note that the USB clock must be 48 MHz when operating USB0 as a Full Speed Function; the USB clock must be 6 MHz when operating USB0 as a Low Speed Function. See SFR Definition 14.6 for USB clock selection options.

Some example USB clock configurations for Full and Low Speed mode are given below:

Internal Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	Clock Multiplier	USBCLK = 000b
Clock Multiplier Input	Internal Oscillator*	MULSEL = 00b
Internal Oscillator	Divide by 1	IFCN = 11b
External Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	Clock Multiplier	USBCLK = 000b
Clock Multiplier Input	External Oscillator	MULSEL = 01b
External Oscillator	Crystal Oscillator Mode 12 MHz Crystal	XOSCMD = 110b XFCN = 111b

***Note:** Clock Recovery must be enabled for this configuration.

Internal Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	Internal Oscillator / 2	USBCLK = 001b
Internal Oscillator	Divide by 1	IFCN = 11b
External Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**

Internal Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	External Oscillator / 4	USBCLK = 101b
External Oscillator	Crystal Oscillator Mode 24 MHz Crystal	XOSCMD = 110b XFCN = 111b

SFR Definition 14.6. CLKSEL: Clock Select

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	USBCLK			-	CLKSL			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address
								0xA9

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

Bits6–4: USBCLK2–0: USB Clock Select

These bits select the clock supplied to USB0. When operating USB0 in full-speed mode, the selected clock should be 48 MHz. When operating USB0 in low-speed mode, the selected clock should be 6 MHz.

USBCLK	Selected Clock
000	4x Clock Multiplier
001	Internal Oscillator / 2
010	External Oscillator
011	External Oscillator / 2
100	External Oscillator / 3
101	External Oscillator / 4
110	RESERVED
111	RESERVED

Bit3: Unused. Read = 0b; Write = don't care.

Bits2–0: CLKSL2–0: System Clock Select

These bits select the system clock source. When operating from a system clock of 25 MHz or less, the FLRT bit should be set to '0'. When operating with a system clock of greater than 25 MHz (up to 48 MHz), the FLRT bit (FLSCL.4) should be set to '1'. See **Section "10. Prefetch Engine" on page 99** for more details.

CLKSL	Selected Clock
000	Internal Oscillator (as determined by the IFCN bits in register OSCICN)
001	External Oscillator
010	4x Clock Multiplier / 2
011*	4x Clock Multiplier*
100	Low-Frequency Oscillator
101-111	RESERVED
***Note:** This option is only available on 48 MHz devices.	

Table 14.1. Oscillator Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	Units
Internal High-Frequency Oscillator (Using Factory-Calibrated Settings)					
Oscillator Frequency	IFCN = 11b	11.82	12.00	12.18	MHz
Oscillator Supply Current (from V_{DD})	24 ºC, V_{DD} = 3.0 V, OSCICN.7 = 1	—	685	—	µA
Internal Low-Frequency Oscillator (Using Factory-Calibrated Settings)					
Oscillator Frequency	OSCLD = 11b	72	80	99	kHz
Oscillator Supply Current (from V_{DD})	24 ºC, V_{DD} = 3.0 V, OSCLCN.7 = 1	—	7.0	—	µA
External USB Clock Requirements					
USB Clock Frequency*	Full Speed Mode Low Speed Mode	47.88 5.91	48 6	48.12 6.09	MHz

***Note:** Applies only to external oscillator sources.

15. Port Input/Output

Digital and analog resources are available through 40 I/O pins (48-pin packages) or 25 I/O pins (32-pin packages). Port pins are organized as shown in Figure 15.1. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input; Port pins P0.0-P3.7 can be assigned to one of the internal digital resources as shown in Figure 15.3. 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 Priority Decoder (Figure 15.3 and Figure 15.4). The registers XBR0, XBR1, and XBR2 defined in SFR Definition 15.1, SFR Definition 15.2, and SFR Definition 15.3, are used to select internal digital functions.

All Port I/Os are 5 V tolerant (refer to Figure 15.2 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 15.1 on page 158.

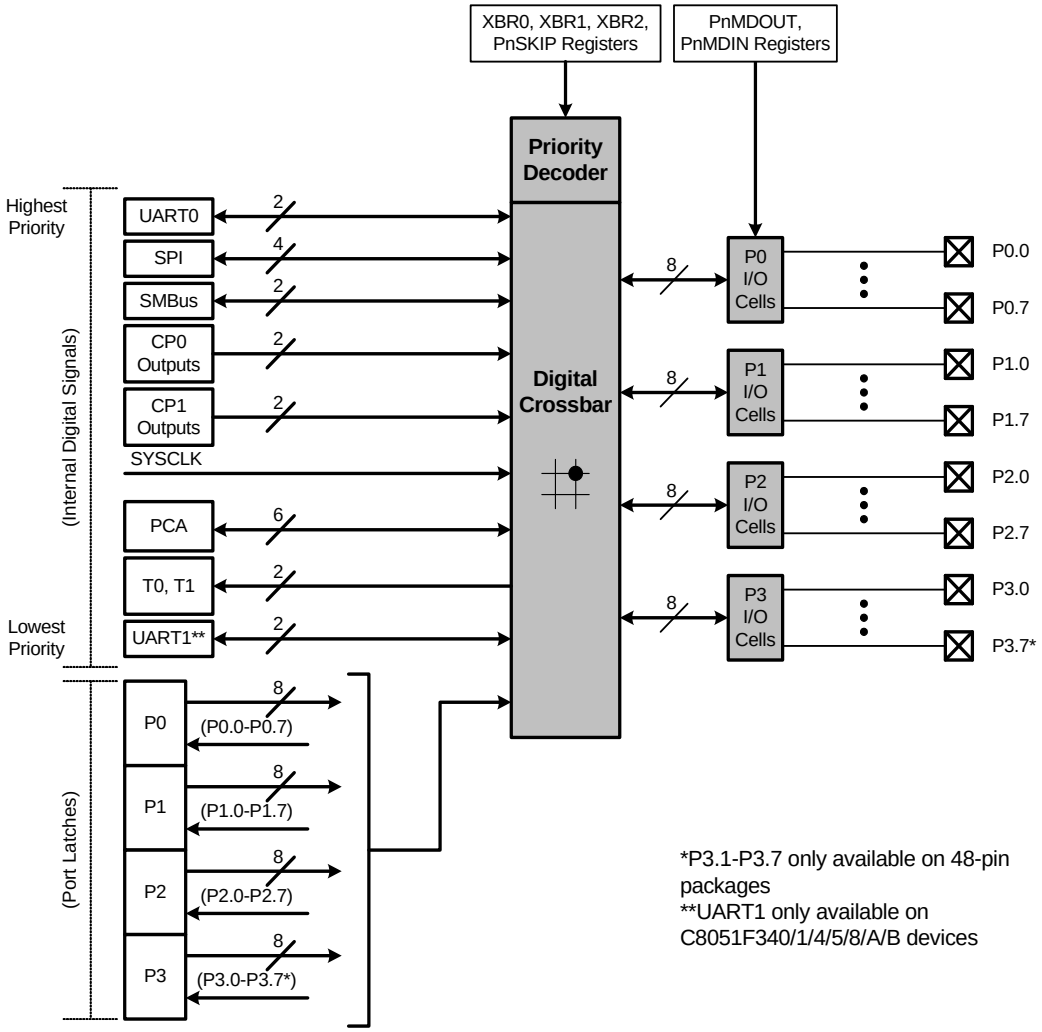


Figure 15.1. Port I/O Functional Block Diagram (Port 0 through Port 3)

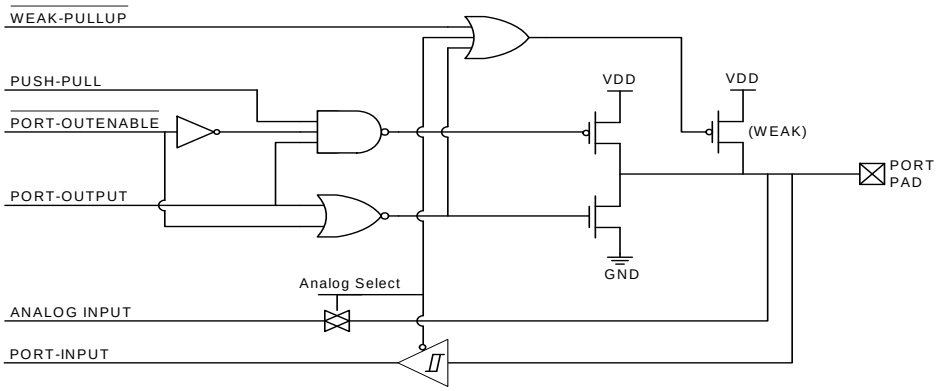


Figure 15.2. Port I/O Cell Block Diagram

15.1. Priority Crossbar Decoder

The Priority Crossbar Decoder (Figure 15.3) 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 is always at pins 4 and 5). 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 VREF signal, external oscillator pins (XTAL1, XTAL2), the ADC's external conversion start signal (CNVSTR), EMIF control signals, and any selected ADC or Comparator inputs. The PnSKIP registers may also be used to skip pins to be used as GPIO. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 15.3 shows all the possible pins available to each peripheral. Figure 15.4 shows the Crossbar Decoder priority with no Port pins skipped. Figure 15.5 shows a Crossbar example with pins P0.2, P0.3, and P1.0 skipped.

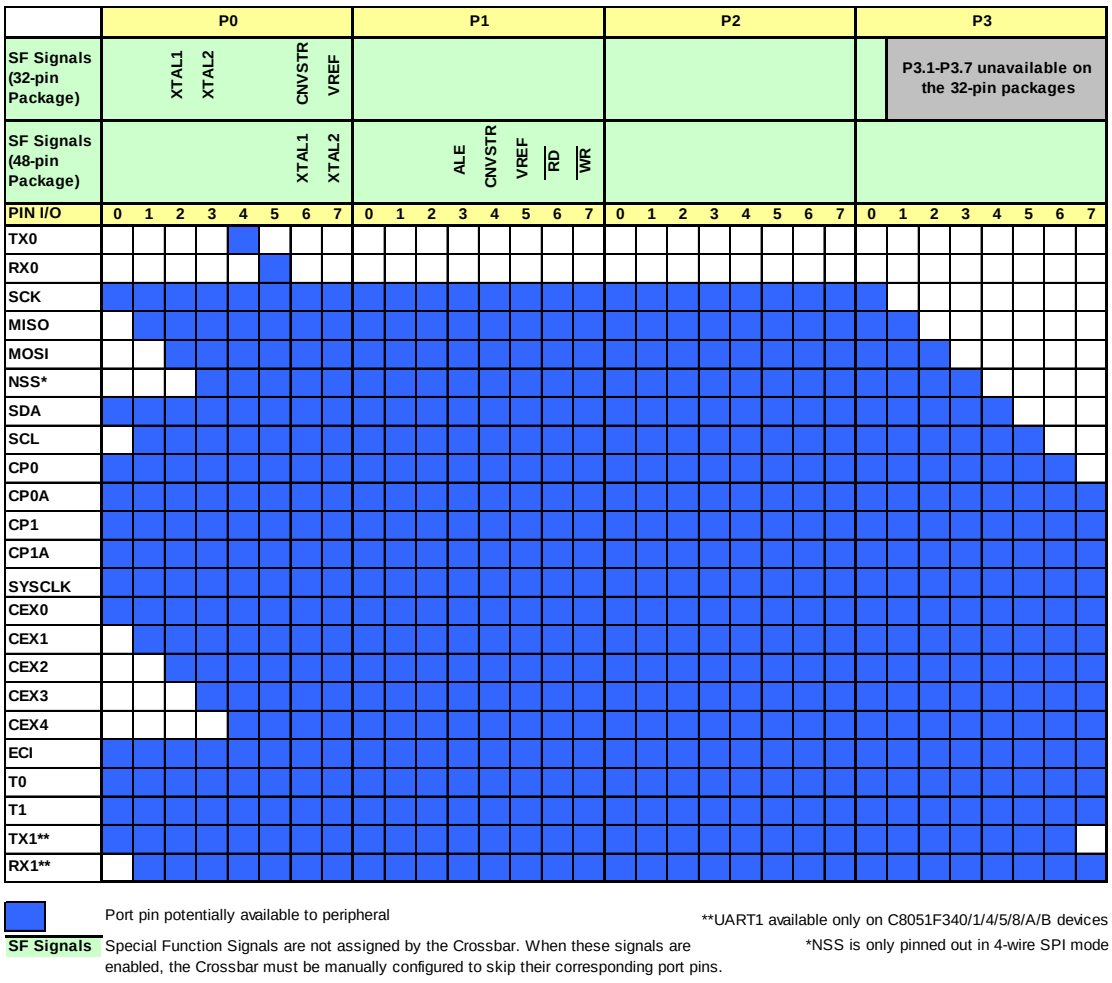


Figure 15.3. Peripheral Availability on Port I/O Pins

	P0								P1								P2								P3							
SF Signals (32-pin Package)							XTAL1	XTAL2							CNVSTR	VREF										P3.1-P3.7 unavailable on the 32-pin packages							
SF Signals (48-pin Package)							XTAL1	XTAL2						ALE	CNVSTR	VREF	RD	WR														
PIN I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
TX0					■																											
RX0						■																										
SCK	■																															
MISO		■																														
MOSI			■																													
NSS*				■		*NSS is only pinned out in 4-wire SPI mode																												
SDA							■																									
SCL								■																								
CP0																																
CP0A																																
CP1																																
CP1A																																
SYSCLK																																
CEX0									■																							
CEX1										■																						
CEX2											■																					
CEX3																																
CEX4																																
ECI																																
T0																																
T1																																
TX1**																	**UART1 available only on C8051F340/1/4/5/8/A/B devices														
RX1**																																
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	P0SKIP[0:7]								P1SKIP[0:7]								P2SKIP[0:7]								P3SKIP[0:7]							

■ Port pin assigned to peripheral by the Crossbar

SF Signals Special Function Signals are not assigned by the Crossbar. When these signals are

Example: XBR0 = 0x07
XBR1 = 0x43

Figure 15.4. Crossbar Priority Decoder in Example Configuration (No Pins Skipped)

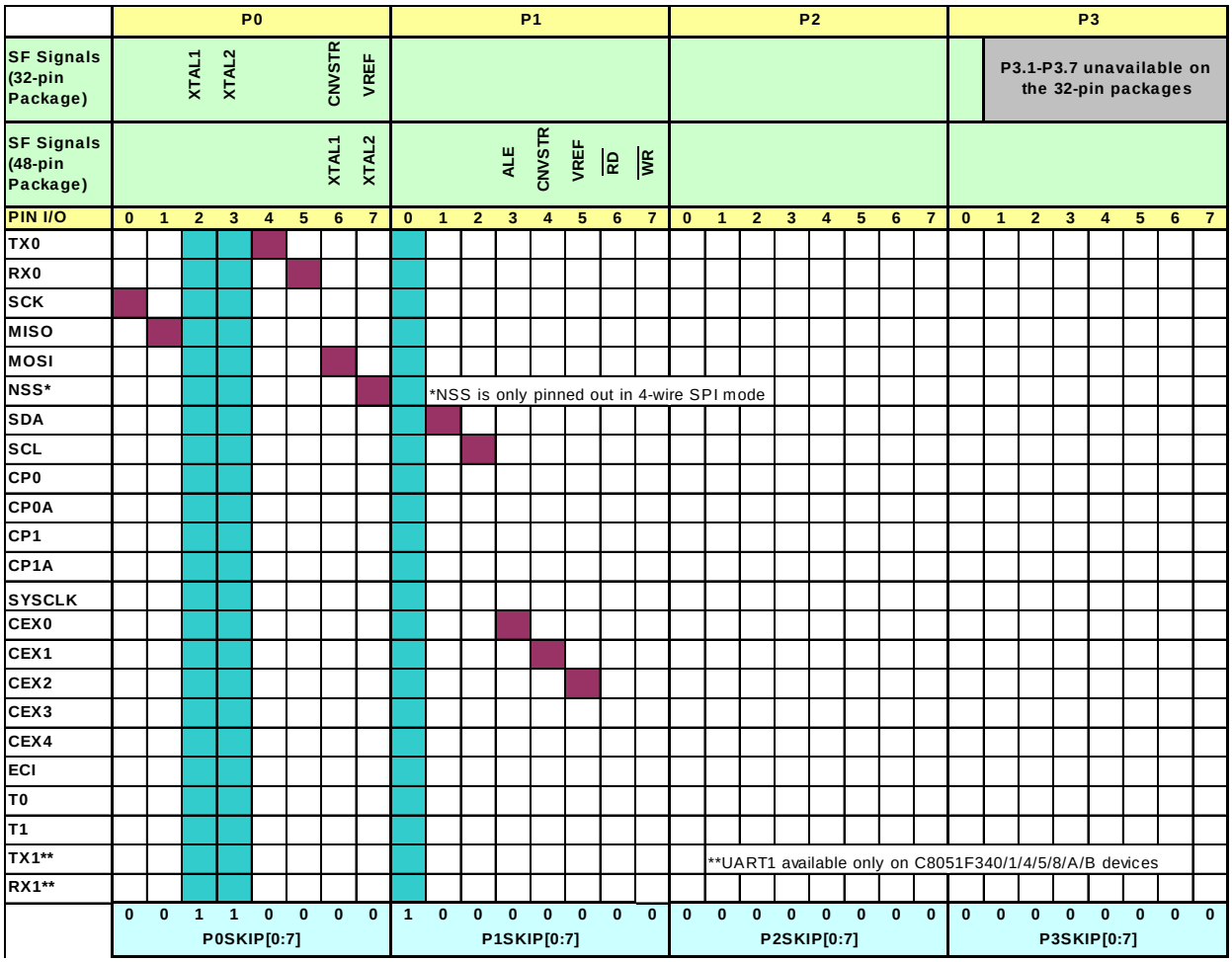


Figure 15.5. Crossbar Priority Decoder in Example Configuration (3 Pins Skipped)

Registers XBR0, XBR1, and XBR2 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 either UART is selected, the Crossbar assigns both pins associated with the UART (TX and RX). UART0 pin assignments are fixed for bootloading purposes: UART TX0 is always assigned to P0.4; UART RX0 is always assigned to P0.5. Standard Port I/Os appear contiguously after the prioritized functions have been assigned.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, 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.

15.2. 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 (XBR0, XBR1).
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 pull-up, 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. To configure a Port pin for digital input, write '0' to the corresponding bit in register PnMDOUT, and write '1' to the corresponding Port latch (register Pn).

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.

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 pull-up is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pull-up is turned off on an output that is driving a '0' 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.

Important Note: The Crossbar must be enabled to use Ports P0, P1, P2, and P3 as standard Port I/O in output mode. These Port output drivers are disabled while the Crossbar is disabled. Port 4 always functions as standard GPIO.

SFR Definition 15.1. XBR0: Port I/O Crossbar Register 0

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	SFR Address:
								0xE1

Bit7: CP1AE: Comparator1 Asynchronous Output Enable
0: Asynchronous CP1 unavailable at Port pin.
1: Asynchronous CP1 routed to Port pin.

Bit6: CP1E: Comparator1 Output Enable
0: CP1 unavailable at Port pin.
1: CP1 routed to Port pin.

Bit5: CP0AE: Comparator0 Asynchronous Output Enable
0: Asynchronous CP0 unavailable at Port pin.
1: Asynchronous CP0 routed to Port pin.

Bit4: CP0E: Comparator0 Output Enable
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.

Bit3: SYSCKE: /SYSCLK Output Enable
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK output routed to Port pin.

Bit2: SMB0E: SMBus I/O Enable
0: SMBus I/O unavailable at Port pins.
1: SMBus I/O routed to Port pins.

Bit1: SPI0E: SPI I/O Enable
0: SPI I/O unavailable at Port pins.
1: SPI I/O routed to Port pins.

Bit0: URT0E: UART0 I/O Output Enable
0: UART0 I/O unavailable at Port pins.
1: UART0 TX0, RX0 routed to Port pins P0.4 and P0.5.

SFR Definition 15.2. XBR1: Port I/O Crossbar Register 1

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	SFR Address:
								0xE2

Bit7: WEAKPUD: Port I/O Weak Pull-up Disable.
0: Weak Pull-ups enabled (except for Ports whose I/O are configured as analog input or push-pull output).
1: Weak Pull-ups disabled.

Bit6: XBARE: Crossbar Enable.
0: Crossbar disabled; all Port drivers disabled.
1: Crossbar enabled.

Bit5: T1E: T1 Enable
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.

Bit4: T0E: T0 Enable
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.

Bit3: ECIE: PCA0 External Counter Input Enable
0: ECI unavailable at Port pin.
1: ECI routed to Port pin.

Bits2–0: 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: Reserved.
111: Reserved.

SFR Definition 15.3. XBR2: Port I/O Crossbar Register 2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
							URT1E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xE3

Bits7–1: RESERVED: Always write to 0000000b

Bit0: URT1E: UART1 I/O Output Enable (C8051F340/1/4/5/8/A/B Only)
0: UART1 I/O unavailable at Port pins.
1: UART1 TX1, RX1 routed to Port pins.

15.3. 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 3-0 are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable. Port 4 (48-pin packages 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. 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 register (not the pin) is read, modified, and written back to the SFR.

SFR Definition 15.4. P0: Port0 Latch

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	SFR Address:
							(bit addressable)	0x80

Bits7–0: P0.[7:0]
Write - Output appears on I/O pins per Crossbar Registers (when XBARE = ‘1’).
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 15.5. P0MDIN: Port0 Input Mode

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	SFR Address:
								0xF1

Bits7–0: Analog Input Configuration Bits for P0.7–P0.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, 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 15.6. P0MDOUT: Port0 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xA4

Bits7–0: Output Configuration Bits for P0.7–P0.0 (respectively): ignored if corresponding bit in register P0MDIN is logic 0.
0: Corresponding P0.n Output is open-drain.
1: Corresponding P0.n Output is push-pull.

(Note: When SDA and SCL appear on any of the Port I/O, each are open-drain regardless of the value of P0MDOUT).

SFR Definition 15.7. P0SKIP: Port0 Skip

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xD4

Bits7–0: P0SKIP[7:0]: Port0 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (VREF input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P0.n pin is not skipped by the Crossbar.
1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 15.8. P1: Port1 Latch

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

Bits7–0: P1.[7:0]
Write - Output appears on I/O pins per Crossbar Registers (when XBARE = ‘1’).
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 15.9. P1MDIN: Port1 Input Mode

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	SFR Address:
								0xF2

Bits7–0: Analog Input Configuration Bits for P1.7–P1.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, 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 15.10. P1MDOUT: Port1 Output Mode

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:
								0xA5

Bits7–0: 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 15.11. P1SKIP: Port1 Skip

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:
								0xD5

Bits7–0: 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 (VREF 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 15.12. P2: Port2 Latch

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	SFR Address:
							(bit addressable)	0xA0

Bits7–0: P2.[7:0]
Write - Output appears on I/O pins per Crossbar Registers (when XBARE = ‘1’).
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 15.13. P2MDIN: Port2 Input Mode

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	SFR Address:
								0xF3

Bits7-0: Analog Input Configuration Bits for P2.7-P2.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, 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 15.14. P2MDOUT: Port2 Output Mode

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:
								0xA6

Bits7–0: 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 15.15. P2SKIP: Port2 Skip

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:
								0xD6

Bits7–0: 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 (VREF 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 15.16. P3: Port3 Latch

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	SFR Address:
							(bit addressable)	0xB0

Bits7–0: P3.[7:0]
Write - Output appears on I/O pins.
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.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.17. P3MDIN: Port3 Input Mode

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	SFR Address:
								0xF4

Bits7–0: Analog Input Configuration Bits for P3.7–P3.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, 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.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.18. P3MDOUT: Port3 Output Mode

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:
								0xA7

Bits7–0: 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.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.19. P3SKIP: Port3 Skip

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:
								0xDF

Bits7–0: P3SKIP[3:0]: 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 (VREF 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.

Note: P3.1–3.7 are only available on 48-pin devices.

SFR Definition 15.20. P4: Port4 Latch

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xC7

Bits7–0: P4.[7:0]
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P4MDOUT.n bit = 0).
Read - Always reads ‘0’ if selected as analog input in register P4MDIN. Directly reads Port pin when configured as digital input.
0: P4.n pin is logic low.
1: P4.n pin is logic high.

Note: P4 is only available on 48-pin devices.

SFR Definition 15.21. P4MDIN: Port4 Input Mode

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	SFR Address:
								0xF5

Bits7–0: Analog Input Configuration Bits for P4.7–P4.0 (respectively).
Port pins configured as analog inputs have their weak pull-up, digital driver, and digital receiver disabled.
0: Corresponding P4.n pin is configured as an analog input.
1: Corresponding P4.n pin is not configured as an analog input.

Note: P4 is only available on 48-pin devices.

SFR Definition 15.22. P4MDOUT: Port4 Output Mode

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:
								0xAE

Bits7–0: Output Configuration Bits for P4.7–P4.0 (respectively); ignored if corresponding bit in register P4MDIN is logic 0.
0: Corresponding P4.n Output is open-drain.
1: Corresponding P4.n Output is push-pull.

Note: P4 is only available on 48-pin devices.

Table 15.1. 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 Pull-up Off			±1	µA
	Weak Pull-up On, V_{IN} = 0 V		25	50	

16. Universal Serial Bus Controller (USB0)

C8051F34x devices include a complete Full/Low Speed USB function for USB peripheral implementations*. The USB Function Controller (USB0) consists of a Serial Interface Engine (SIE), USB Transceiver (including matching resistors and configurable pull-up resistors), 1k FIFO block, and clock recovery mechanism for crystal-less operation. No external components are required. The USB Function Controller and Transceiver is Universal Serial Bus Specification 2.0 compliant.

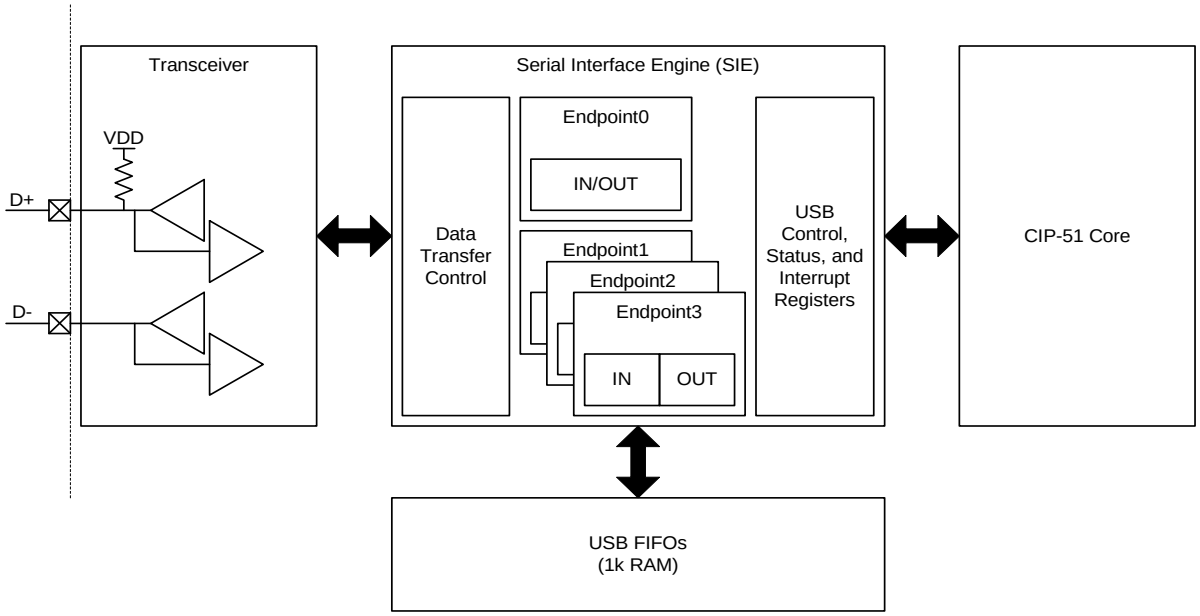


Figure 16.1. USB0 Block Diagram

Important Note: *This document assumes a comprehensive understanding of the USB Protocol. Terms and abbreviations used in this document are defined in the USB Specification. We encourage you to review the latest version of the USB Specification before proceeding.*

***Note:** The C8051F34x cannot be used as a USB Host device.

16.1. Endpoint Addressing

A total of eight endpoint pipes are available. The control endpoint (Endpoint0) always functions as a bi-directional IN/OUT endpoint. The other endpoints are implemented as three pairs of IN/OUT endpoint pipes:

Table 16.1. Endpoint Addressing Scheme

Endpoint	Associated Pipes	USB Protocol Address
Endpoint0	Endpoint0 IN	0x00
	Endpoint0 OUT	0x00
Endpoint1	Endpoint1 IN	0x81
	Endpoint1 OUT	0x01
Endpoint2	Endpoint2 IN	0x82
	Endpoint2 OUT	0x02
Endpoint3	Endpoint3 IN	0x83
	Endpoint3 OUT	0x03

16.2. USB Transceiver

The USB Transceiver is configured via the USB0XCN register shown in SFR Definition 16.1. This configuration includes Transceiver enable/disable, pull-up resistor enable/disable, and device speed selection (Full or Low Speed). When bit SPEED = '1', USB0 operates as a Full Speed USB function, and the on-chip pull-up resistor (if enabled) appears on the D+ pin. When bit SPEED = '0', USB0 operates as a Low Speed USB function, and the on-chip pull-up resistor (if enabled) appears on the D- pin. Bits4-0 of register USB0XCN can be used for Transceiver testing as described in SFR Definition 16.1. The pull-up resistor is enabled only when VBUS is present (see **Section "8.2. VBUS Detection" on page 69** for details on VBUS detection).

Important Note: The USB clock should be active before the Transceiver is enabled.

SFR Definition 16.1. USB0XCN: USB0 Transceiver Control

R/W	R/W	R/W	R/W	R/W	R	R	R	Reset Value
PREN	PHYEN	SPEED	PHYTST1	PHYTST0	DFREC	Dp	Dn	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD7

Bit7: PREN: Internal Pull-up Resistor Enable
The location of the pull-up resistor (D+ or D–) is determined by the SPEED bit.
0: Internal pull-up resistor disabled (device effectively detached from the USB network).
1: Internal pull-up resistor enabled when VBUS is present (device attached to the USB network).

Bit6: PHYEN: Physical Layer Enable
This bit enables/disables the USB0 physical layer transceiver.
0: Transceiver disabled (suspend).
1: Transceiver enabled (normal).

Bit5: SPEED: USB0 Speed Select
This bit selects the USB0 speed.
0: USB0 operates as a Low Speed device. If enabled, the internal pull-up resistor appears on the D– line.
1: USB0 operates as a Full Speed device. If enabled, the internal pull-up resistor appears on the D+ line.

Bits4–3: PHYTST1–0: Physical Layer Test
These bits can be used to test the USB0 transceiver.

PHYTST[1:0]	Mode	D+	D–
00b	Mode 0: Normal (non-test mode)	X	X
01b	Mode 1: Differential ‘1’ Forced	1	0
10b	Mode 2: Differential ‘0’ Forced	0	1
11b	Mode 3: Single-Ended ‘0’ Forced	0	0

Bit2: DFREC: Differential Receiver
The state of this bit indicates the current differential value present on the D+ and D– lines when PHYEN = ‘1’.
0: Differential ‘0’ signaling on the bus.
1: Differential ‘1’ signaling on the bus.

Bit1: Dp: D+ Signal Status
This bit indicates the current logic level of the D+ pin.
0: D+ signal currently at logic 0.
1: D+ signal currently at logic 1.

Bit0: Dn: D- Signal Status
This bit indicates the current logic level of the D– pin.
0: D– signal currently at logic 0.
1: D– signal currently at logic 1.

16.3. USB Register Access

The USB0 controller registers listed in Table 16.2 are accessed through two SFRs: USB0 Address (USB0ADR) and USB0 Data (USB0DAT). The USB0ADR register selects which USB register is targeted by reads/writes of the USB0DAT register. See Figure 16.2.

Endpoint control/status registers are accessed by first writing the USB register INDEX with the target endpoint number. Once the target endpoint number is written to the INDEX register, the control/status registers associated with the target endpoint may be accessed. See the “Indexed Registers” section of Table 16.2 for a list of endpoint control/status registers.

Important Note: The USB clock must be active when accessing USB registers.

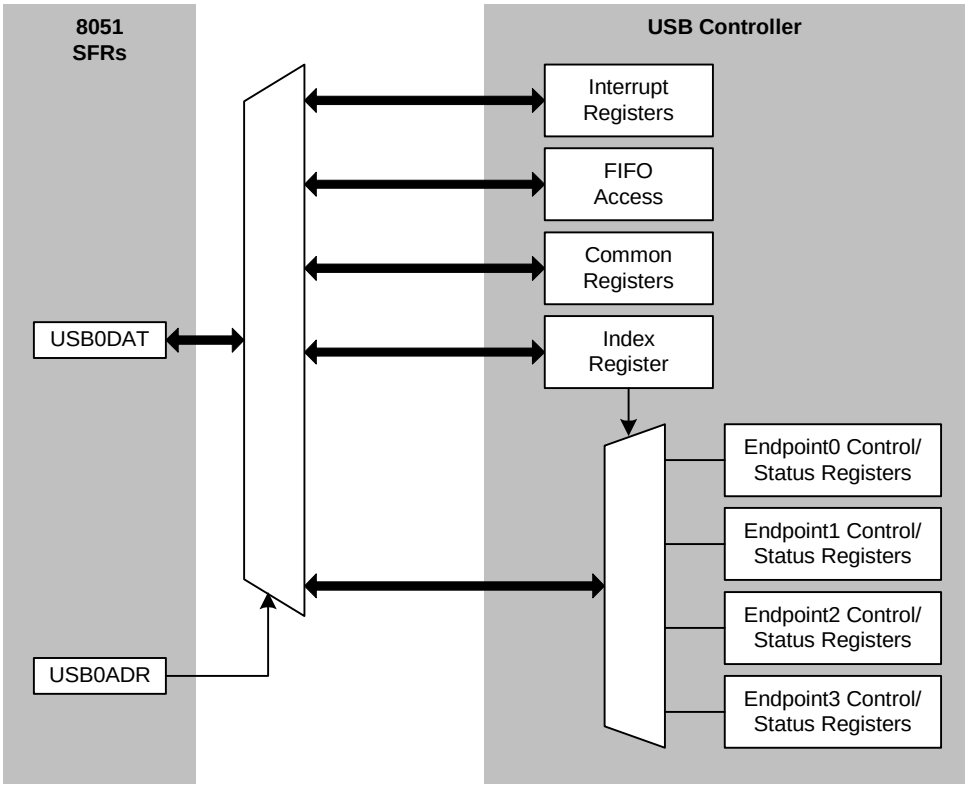


Figure 16.2. USB0 Register Access Scheme

SFR Definition 16.2. USB0ADR: USB0 Indirect Address

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
BUSY	AUTORD	USBADDR						00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x96

Bits7: BUSY: USB0 Register Read Busy Flag
This bit is used during indirect USB0 register accesses. Software should write '1' to this bit to initiate a read of the USB0 register targeted by the USBADDR bits (USB0ADR.[5-0]). The target address and BUSY bit may be written in the same write to USB0ADR. After BUSY is set to '1', hardware will clear BUSY when the targeted register data is ready in the USB0DAT register. Software should check BUSY for '0' before writing to USB0DAT.
Write:
0: No effect.
1: A USB0 indirect register read is initiated at the address specified by the USBADDR bits.
Read:
0: USB0DAT register data is valid.
1: USB0 is busy accessing an indirect register; USB0DAT register data is invalid.

Bit6: AUTORD: USB0 Register Auto-read Flag
This bit is used for block FIFO reads.
0: BUSY must be written manually for each USB0 indirect register read.
1: The next indirect register read will automatically be initiated when software reads USB0DAT (USBADDR bits will not be changed).

Bits5–0: USBADDR: USB0 Indirect Register Address
These bits hold a 6-bit address used to indirectly access the USB0 core registers. Table 16.2 lists the USB0 core registers and their indirect addresses. Reads and writes to USB0DAT will target the register indicated by the USBADDR bits.

SFR Definition 16.3. USB0DAT: USB0 Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
USB0DAT								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x97

This SFR is used to indirectly read and write USB0 registers.

Write Procedure:
1. Poll for BUSY (USB 0ADR.7) => ‘0’.
2. Load the target USB0 register address into the USBADDR bits in register USB0ADR.
3. Write data to USB0DAT.
4. Repeat (Step 2 may be skipped when writing to the same USB0 register).

Read Procedure:
1. Poll for BUSY (USB 0ADR.7) => ‘0’.
2. Load the target USB0 register address into the USBADDR bits in register USB0ADR.
3. Write ‘1’ to the BUSY bit in register USB0ADR (steps 2 and 3 can be performed in the same write).
4. Poll for BUSY (USB 0ADR.7) => ‘0’.
5. Read data from USB0DAT.
6. Repeat from Step 2 (Step 2 may be skipped when reading the same USB0 register; Step 3 may be skipped when the AUTORD bit (USB0ADR.6) is logic 1).

Table 16.2. USB0 Controller Registers

USB Register Name	USB Register Address	Description	Page Number
Interrupt Registers			
IN1INT	0x02	Endpoint0 and Endpoints1-3 IN Interrupt Flags	173
OUT1INT	0x04	Endpoints1-3 OUT Interrupt Flags	173
CMINT	0x06	Common USB Interrupt Flags	174
IN1IE	0x07	Endpoint0 and Endpoints1-3 IN Interrupt Enables	175
OUT1IE	0x09	Endpoints1-3 OUT Interrupt Enables	175
CMIE	0x0B	Common USB Interrupt Enables	176
Common Registers			
FADDR	0x00	Function Address	169
POWER	0x01	Power Management	171
FRAMEL	0x0C	Frame Number Low Byte	172
FRAMEH	0x0D	Frame Number High Byte	172
INDEX	0x0E	Endpoint Index Selection	165
CLKREC	0x0F	Clock Recovery Control	166
FIFOn	0x20–0x23	Endpoints0-3 FIFOs	168
Indexed Registers			
E0CSR	0x11	Endpoint0 Control / Status	179
EINCSRL		Endpoint IN Control / Status Low Byte	182
EINCSRH	0x12	Endpoint IN Control / Status High Byte	183
EOUTCSRL	0x14	Endpoint OUT Control / Status Low Byte	185
EOUTCSRH	0x15	Endpoint OUT Control / Status High Byte	186
E0CNT	0x16	Number of Received Bytes in Endpoint0 FIFO	180
EOUTCNTL		Endpoint OUT Packet Count Low Byte	186
EOUTCNTH	0x17	Endpoint OUT Packet Count High Byte	186

USB Register Definition 16.4. INDEX: USB0 Endpoint Index

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	EPSEL				00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0E

Bits7–4: Unused. Read = 0000b; Write = don't care.
Bits3–0: EPSEL: Endpoint Select
These bits select which endpoint is targeted when indexed USB0 registers are accessed.

INDEX	Target Endpoint
0x0	0
0x1	1
0x2	2
0x3	3
0x4–0xF	Reserved

16.4. USB Clock Configuration

USB0 is capable of communication as a Full or Low Speed USB function. Communication speed is selected via the SPEED bit in SFR USB0XCN. When operating as a Low Speed function, the USB0 clock must be 6 MHz. When operating as a Full Speed function, the USB0 clock must be 48 MHz. Clock options are described in **Section "14. Oscillators" on page 131**. The USB0 clock is selected via SFR CLKSEL (see SFR Definition 14.6).

Clock Recovery circuitry uses the incoming USB data stream to adjust the internal oscillator; this allows the internal oscillator (and 4x Clock Multiplier) to meet the requirements for USB clock tolerance. Clock Recovery should be used in the following configurations:

Communication Speed	USB Clock	4x Clock Multiplier Input
Full Speed	4x Clock Multiplier	Internal Oscillator
Low Speed	Internal Oscillator / 2	N/A

When operating USB0 as a Low Speed function with Clock Recovery, software must write '1' to the CRLOW bit to enable Low Speed Clock Recovery. Clock Recovery is typically not necessary in Low Speed mode.

Single Step Mode can be used to help the Clock Recovery circuitry to lock when high noise levels are present on the USB network. This mode is not required (or recommended) in typical USB environments.

USB Register Definition 16.5. CLKREC: Clock Recovery Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CRE	CRSSEN	CRLOW	Reserved					00001001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0F

Bit7: CRE: Clock Recovery Enable.
This bit enables/disables the USB clock recovery feature.
0: Clock recovery disabled.
1: Clock recovery enabled.

Bit6: CRSSEN: Clock Recovery Single Step.
This bit forces the oscillator calibration into 'single-step' mode during clock recovery.
0: Normal calibration mode.
1: Single step mode.

Bit5: CRLOW: Low Speed Clock Recovery Mode.
This bit must be set to '1' if clock recovery is used when operating as a Low Speed USB device.
0: Full Speed Mode.
1: Low Speed Mode.

Bits4–0: Reserved. Read = Variable. Must Write = 01001b.

Note: The USB transceiver must be enabled before enabling Clock Recovery.

16.5. FIFO Management

1024 bytes of on-chip XRAM are used as FIFO space for USB0. This FIFO space is split between Endpoints0-3 as shown in Figure 16.3. FIFO space allocated for Endpoints1-3 is configurable as IN, OUT, or both (Split Mode: half IN, half OUT).

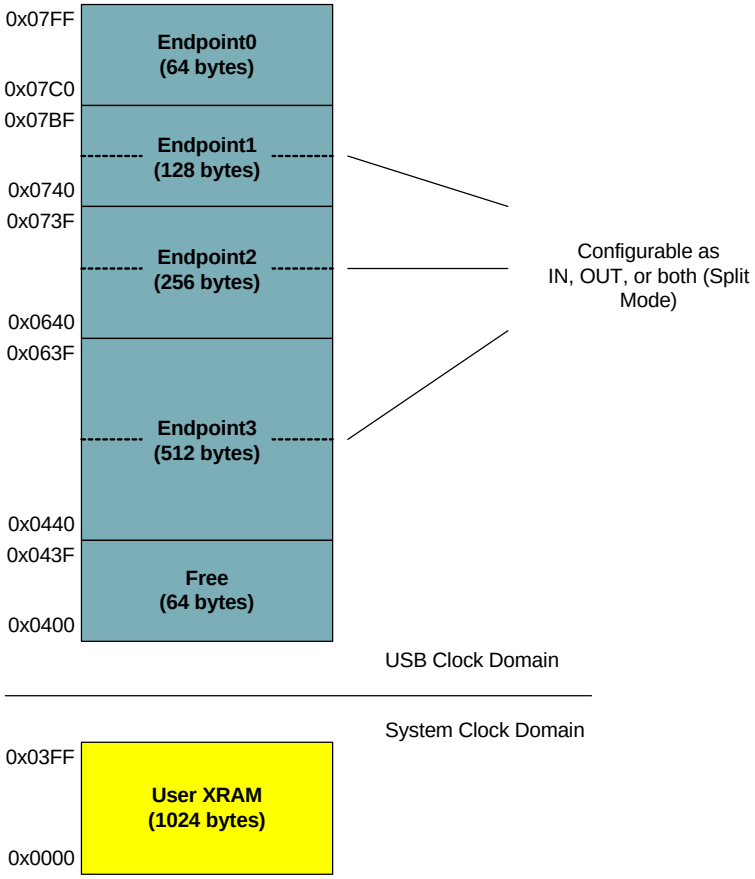


Figure 16.3. USB FIFO Allocation

16.5.1. FIFO Split Mode

The FIFO space for Endpoints1-3 can be split such that the upper half of the FIFO space is used by the IN endpoint, and the lower half is used by the OUT endpoint. For example: if the Endpoint3 FIFO is configured for Split Mode, the upper 256 bytes (0x0540 to 0x063F) are used by Endpoint3 IN and the lower 256 bytes (0x0440 to 0x053F) are used by Endpoint3 OUT.

If an endpoint FIFO is not configured for Split Mode, that endpoint IN/OUT pair's FIFOs are combined to form a single IN *or* OUT FIFO. In this case only one direction of the endpoint IN/OUT pair may be used at a time. The endpoint direction (IN/OUT) is determined by the DIRSEL bit in the corresponding endpoint's EINCSRH register (see SFR Definition 16.20).

16.5.2. FIFO Double Buffering

FIFO slots for Endpoints1-3 can be configured for double-buffered mode. In this mode, the maximum packet size is halved and the FIFO may contain two packets at a time. This mode is available for Endpoints1-3. When an endpoint is configured for Split Mode, double buffering may be enabled for the IN Endpoint and/or the OUT endpoint. When Split Mode is not enabled, double-buffering may be enabled for the entire endpoint FIFO. See Table 16.3 for a list of maximum packet sizes for each FIFO configuration.

Table 16.3. FIFO Configurations

Endpoint Number	Split Mode Enabled?	Maximum IN Packet Size (Double Buffer Disabled / Enabled)	Maximum OUT Packet Size (Double Buffer Disabled / Enabled)
0	N/A	64	
1	N	128 / 64	
	Y	64 / 32	64 / 32
2	N	256 / 128	
	Y	128 / 64	128 / 64
3	N	512 / 256	
	Y	256 / 128	256 / 128

16.5.1. FIFO Access

Each endpoint FIFO is accessed through a corresponding FIFOn register. A read of an endpoint FIFOn register unloads one byte from the FIFO; a write of an endpoint FIFOn register loads one byte into the endpoint FIFO. When an endpoint FIFO is configured for Split Mode, a read of the endpoint FIFOn register unloads one byte from the OUT endpoint FIFO; a write of the endpoint FIFOn register loads one byte into the IN endpoint FIFO.

USB Register Definition 16.6. FIFOn: USB0 Endpoint FIFO Access

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
FIFODATA								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x20 - 0x23

USB Addresses 0x20–0x23 provide access to the 4 pairs of endpoint FIFOs:

IN/OUT Endpoint FIFO	USB Address
0	0x20
1	0x21
2	0x22
3	0x23

Writing to the FIFO address loads data into the IN FIFO for the corresponding endpoint. Reading from the FIFO address unloads data from the OUT FIFO for the corresponding endpoint.

16.6. Function Addressing

The FADDR register holds the current USB0 function address. Software should write the host-assigned 7-bit function address to the FADDR register when received as part of a SET_ADDRESS command. A new address written to FADDR will not take effect (USB0 will not respond to the new address) until the end of the current transfer (typically following the status phase of the SET_ADDRESS command transfer). The UPDATE bit (FADDR.7) is set to '1' by hardware when software writes a new address to the FADDR register. Hardware clears the UPDATE bit when the new address takes effect as described above.

USB Register Definition 16.7. FADDR: USB0 Function Address

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Update	Function Address							00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x00

Bit7: Update: Function Address Update
Set to '1' when software writes the FADDR register. USB0 clears this bit to '0' when the new address takes effect.
0: The last address written to FADDR is in effect.
1: The last address written to FADDR is not yet in effect.

Bits6–0: Function Address
Holds the 7-bit function address for USB0. This address should be written by software when the SET_ADDRESS standard device request is received on Endpoint0. The new address takes effect when the device request completes.

16.7. Function Configuration and Control

The USB register POWER (SFR Definition 16.8) is used to configure and control USB0 at the device level (enable/disable, Reset/Suspend/Resume handling, etc.).

USB Reset: The USBRST bit (POWER.3) is set to '1' by hardware when Reset signaling is detected on the bus. Upon this detection, the following occur:

1. The USB0 Address is reset (FADDR = 0x00).
2. Endpoint FIFOs are flushed.
3. Control/status registers are reset to 0x00 (E0CSR, EINCSRL, EINCSRH, EOUTCSRL, EOUTCSRH).
4. USB register INDEX is reset to 0x00.
5. All USB interrupts (excluding the Suspend interrupt) are enabled and their corresponding flags cleared.
6. A USB Reset interrupt is generated if enabled.

Writing a '1' to the USBRST bit will generate an asynchronous USB0 reset. All USB registers are reset to their default values following this asynchronous reset.

Suspend Mode: With Suspend Detection enabled (SUSEN = '1'), USB0 will enter Suspend Mode when Suspend signaling is detected on the bus. An interrupt will be generated if enabled (SUSINTE = '1'). The Suspend Interrupt Service Routine (ISR) should perform application-specific configuration tasks such as disabling appropriate peripherals and/or configuring clock sources for low power modes. See **Section**

“14. Oscillators” on page 131 for more details on internal oscillator configuration, including the Suspend mode feature of the internal oscillator.

USB0 exits Suspend mode when any of the following occur: (1) Resume signaling is detected or generated, (2) Reset signaling is detected, or (3) a device or USB reset occurs. If suspended, the internal oscillator will exit Suspend mode upon any of the above listed events.

Resume Signaling: USB0 will exit Suspend mode if Resume signaling is detected on the bus. A Resume interrupt will be generated upon detection if enabled (RESINTE = ‘1’). Software may force a Remote Wakeup by writing ‘1’ to the RESUME bit (POWER.2). When forcing a Remote Wakeup, software should write RESUME = ‘0’ to end Resume signaling 10-15 ms after the Remote Wakeup is initiated (RESUME = ‘1’).

ISO Update: When software writes ‘1’ to the ISOUP bit (POWER.7), the ISO Update function is enabled. With ISO Update enabled, new packets written to an ISO IN endpoint will not be transmitted until a new Start-Of-Frame (SOF) is received. If the ISO IN endpoint receives an IN token before a SOF, USB0 will transmit a zero-length packet. When ISOUP = ‘1’, ISO Update is enabled for all ISO endpoints.

USB Enable: USB0 is disabled following a Power-On-Reset (POR). USB0 is enabled by clearing the USBINH bit (POWER.4). Once written to ‘0’, the USBINH can only be set to ‘1’ by one of the following: (1) a Power-On-Reset (POR), or (2) an asynchronous USB0 reset generated by writing ‘1’ to the USBRST bit (POWER.3).

Software should perform all USB0 configuration before enabling USB0. The configuration sequence should be performed as follows:

Step 1. Select and enable the USB clock source.
Step 2. Reset USB0 by writing USBRST= ‘1’.
Step 3. Configure and enable the USB Transceiver.
Step 4. Perform any USB0 function configuration (interrupts, Suspend detect).
Step 5. Enable USB0 by writing USBINH = ‘0’.

USB Register Definition 16.8. POWER: USB0 Power

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
ISOUD	-	-	USBINH	USBRST	RESUME	SUSMD	SUSEN	00010000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x01

Bit7: ISOUD: ISO Update
This bit affects all IN Isochronous endpoints.
0: When software writes INPRDY = ‘1’, USB0 will send the packet when the next IN token is received.
1: When software writes INPRDY = ‘1’, USB0 will wait for a SOF token before sending the packet. If an IN token is received before a SOF token, USB0 will send a zero-length data packet.

Bits6–5: Unused. Read = 00b. Write = don’t care.

Bit4: USBINH: USB0 Inhibit
This bit is set to ‘1’ following a power-on reset (POR) or an asynchronous USB0 reset (see Bit3: RESET). Software should clear this bit after all USB0 and transceiver initialization is complete. Software cannot set this bit to ‘1’.
0: USB0 enabled.
1: USB0 inhibited. All USB traffic is ignored.

Bit3: USBRST: Reset Detect
Writing ‘1’ to this bit forces an asynchronous USB0 reset. Reading this bit provides bus reset status information.
Read:
0: Reset signaling is not present on the bus.
1: Reset signaling detected on the bus.

Bit2: RESUME: Force Resume
Software can force resume signaling on the bus to wake USB0 from suspend mode. Writing a ‘1’ to this bit while in Suspend mode (SUSMD = ‘1’) forces USB0 to generate Resume signaling on the bus (a remote Wakeup event). Software should write RESUME = ‘0’ after 10 ms to15 ms to end the Resume signaling. An interrupt is generated, and hardware clears SUSMD, when software writes RESUME = ‘0’.

Bit1: SUSMD: Suspend Mode
Set to ‘1’ by hardware when USB0 enters suspend mode. Cleared by hardware when software writes RESUME = ‘0’ (following a remote wakeup) or reads the CMINT register after detection of Resume signaling on the bus.
0: USB0 not in suspend mode.
1: USB0 in suspend mode.

Bit0: SUSEN: Suspend Detection Enable
0: Suspend detection disabled. USB0 will ignore suspend signaling on the bus.
1: Suspend detection enabled. USB0 will enter suspend mode if it detects suspend signaling on the bus.

USB Register Definition 16.9. FRAMEL: USB0 Frame Number Low

R	R	R	R	R	R	R	R	Reset Value
Frame Number Low								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0C

Bits7-0: Frame Number Low
This register contains bits7-0 of the last received frame number.

USB Register Definition 16.10. FRAMEH: USB0 Frame Number High

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	-	Frame Number High			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0D

Bits7-3: Unused. Read = 0. Write = don't care.
Bits2-0: Frame Number High Byte
This register contains bits10-8 of the last received frame number.

16.8. Interrupts

The read-only USB0 interrupt flags are located in the USB registers shown in USB Register Definition 16.11 through USB Register Definition 16.13. The associated interrupt enable bits are located in the USB registers shown in USB Register Definition 16.14 through USB Register Definition 16.16. A USB0 interrupt is generated when any of the USB interrupt flags is set to '1'. The USB0 interrupt is enabled via the EIE1 SFR (see **Section "9.3. Interrupt Handler" on page 88**).

Important Note: Reading a USB interrupt flag register resets all flags in that register to '0'.

USB Register Definition 16.11. IN1INT: USB0 IN Endpoint Interrupt

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	IN3	IN2	IN1	EP0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x02

Bits7–4: Unused. Read = 0000b. Write = don't care.
Bit3: IN3: IN Endpoint 3 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 3 interrupt inactive.
1: IN Endpoint 3 interrupt active.
Bit2: IN2: IN Endpoint 2 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 2 interrupt inactive.
1: IN Endpoint 2 interrupt active.
Bit1: IN1: IN Endpoint 1 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 1 interrupt inactive.
1: IN Endpoint 1 interrupt active.
Bit0: EP0: Endpoint 0 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: Endpoint 0 interrupt inactive.
1: Endpoint 0 interrupt active.

USB Register Definition 16.12. OUT1INT: USB0 Out Endpoint Interrupt

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	OUT3	OUT2	OUT1	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x04

Bits7–4: Unused. Read = 0000b. Write = don't care.
Bit3: OUT3: OUT Endpoint 3 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 3 interrupt inactive.
1: OUT Endpoint 3 interrupt active.
Bit2: OUT2: OUT Endpoint 2 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 2 interrupt inactive.
1: OUT Endpoint 2 interrupt active.
Bit1: OUT1: OUT Endpoint 1 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 1 interrupt inactive.
1: OUT Endpoint 1 interrupt active.
Bit0: Unused. Read = 0; Write = don't care.

USB Register Definition 16.13. CMINT: USB0 Common Interrupt

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	SOF	RSTINT	RSUINT	SUSINT	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x06

Bits7–4: Unused. Read = 0000b; Write = don't care.

Bit3: SOF: Start of Frame Interrupt
Set by hardware when a SOF token is received. This interrupt event is synthesized by hardware: an interrupt will be generated when hardware expects to receive a SOF event, even if the actual SOF signal is missed or corrupted.
This bit is cleared when software reads the CMINT register.
0: SOF interrupt inactive.
1: SOF interrupt active.

Bit2: RSTINT: Reset Interrupt-pending Flag
Set by hardware when Reset signaling is detected on the bus.
This bit is cleared when software reads the CMINT register.
0: Reset interrupt inactive.
1: Reset interrupt active.

Bit1: RSUINT: Resume Interrupt-pending Flag
Set by hardware when Resume signaling is detected on the bus while USB0 is in suspend mode.
This bit is cleared when software reads the CMINT register.
0: Resume interrupt inactive.
1: Resume interrupt active.

Bit0: SUSINT: Suspend Interrupt-pending Flag
When Suspend detection is enabled (bit SUSEN in register POWER), this bit is set by hardware when Suspend signaling is detected on the bus. This bit is cleared when software reads the CMINT register.
0: Suspend interrupt inactive.
1: Suspend interrupt active.

USB Register Definition 16.14. IN1IE: USB0 IN Endpoint Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	IN3E	IN2E	IN1E	EP0E	00001111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x07

Bits7–4: Unused. Read = 0000b. Write = don't care.
Bit3: IN3E: IN Endpoint 3 Interrupt Enable
0: IN Endpoint 3 interrupt disabled.
1: IN Endpoint 3 interrupt enabled.
Bit2: IN2E: IN Endpoint 2 Interrupt Enable
0: IN Endpoint 2 interrupt disabled.
1: IN Endpoint 2 interrupt enabled.
Bit1: IN1E: IN Endpoint 1 Interrupt Enable
0: IN Endpoint 1 interrupt disabled.
1: IN Endpoint 1 interrupt enabled.
Bit0: EP0E: Endpoint 0 Interrupt Enable
0: Endpoint 0 interrupt disabled.
1: Endpoint 0 interrupt enabled.

USB Register Definition 16.15. OUT1IE: USB0 Out Endpoint Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	OUT3E	OUT2E	OUT1E	-	00001110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x09

Bits7–4: Unused. Read = 0000b. Write = don't care.
Bit3: OUT3E: OUT Endpoint 3 Interrupt Enable
0: OUT Endpoint 3 interrupt disabled.
1: OUT Endpoint 3 interrupt enabled.
Bit2: OUT2E: OUT Endpoint 2 Interrupt Enable
0: OUT Endpoint 2 interrupt disabled.
1: OUT Endpoint 2 interrupt enabled.
Bit1: OUT1E: OUT Endpoint 1 Interrupt Enable
0: OUT Endpoint 1 interrupt disabled.
1: OUT Endpoint 1 interrupt enabled.
Bit0: Unused. Read = 0; Write = don't' care.

USB Register Definition 16.16. CMIE: USB0 Common Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	SOFE	RSTINTE	RSUINTE	SUSINTE	00000110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0B

Bits7–4: Unused. Read = 0000b; Write = don't care.
Bit3: SOFE: Start of Frame Interrupt Enable
0: SOF interrupt disabled.
1: SOF interrupt enabled.
Bit2: RSTINTE: Reset Interrupt Enable
0: Reset interrupt disabled.
1: Reset interrupt enabled.
Bit1: RSUINTE: Resume Interrupt Enable
0: Resume interrupt disabled.
1: Resume interrupt enabled.
Bit0: SUSINTE: Suspend Interrupt Enable
0: Suspend interrupt disabled.
1: Suspend interrupt enabled.

16.9. The Serial Interface Engine

The Serial Interface Engine (SIE) performs all low level USB protocol tasks, interrupting the processor when data has successfully been transmitted or received. When receiving data, the SIE will interrupt the processor when a complete data packet has been received; appropriate handshaking signals are automatically generated by the SIE. When transmitting data, the SIE will interrupt the processor when a complete data packet has been transmitted and the appropriate handshake signal has been received.

The SIE will not interrupt the processor when corrupted/erroneous packets are received.

16.10. Endpoint0

Endpoint0 is managed through the USB register E0CSR (USB Register Definition 16.17). The INDEX register must be loaded with 0x00 to access the E0CSR register.

An Endpoint0 interrupt is generated when:

1. A data packet (OUT or SETUP) has been received and loaded into the Endpoint0 FIFO. The OPRDY bit (E0CSR.0) is set to '1' by hardware.
2. An IN data packet has successfully been unloaded from the Endpoint0 FIFO and transmitted to the host; INPRDY is reset to '0' by hardware.
3. An IN transaction is completed (this interrupt generated during the status stage of the transaction).
4. Hardware sets the STSTL bit (E0CSR.2) after a control transaction ended due to a protocol violation.
5. Hardware sets the SUEND bit (E0CSR.4) because a control transfer ended before firmware sets the DATAEND bit (E0CSR.3).

The E0CNT register (USB Register Definition 16.18) holds the number of received data bytes in the Endpoint0 FIFO.

Hardware will automatically detect protocol errors and send a STALL condition in response. Firmware may force a STALL condition to abort the current transfer. When a STALL condition is generated, the STSTL bit will be set to '1' and an interrupt generated. The following conditions will cause hardware to generate a STALL condition:

1. The host sends an OUT token during a OUT data phase after the DATAEND bit has been set to '1'.
2. The host sends an IN token during an IN data phase after the DATAEND bit has been set to '1'.
3. The host sends a packet that exceeds the maximum packet size for Endpoint0.
4. The host sends a non-zero length DATA1 packet during the status phase of an IN transaction.
5. Firmware sets the SDSTL bit (E0CSR.5) to '1'.

16.10.1.Endpoint0 SETUP Transactions

All control transfers must begin with a SETUP packet. SETUP packets are similar to OUT packets, containing an 8-byte data field sent by the host. Any SETUP packet containing a command field of anything other than 8 bytes will be automatically rejected by USB0. An Endpoint0 interrupt is generated when the data from a SETUP packet is loaded into the Endpoint0 FIFO. Software should unload the command from the Endpoint0 FIFO, decode the command, perform any necessary tasks, and set the SOPRDY bit to indicate that it has serviced the OUT packet.

16.10.2.Endpoint0 IN Transactions

When a SETUP request is received that requires USB0 to transmit data to the host, one or more IN requests will be sent by the host. For the first IN transaction, firmware should load an IN packet into the Endpoint0 FIFO, and set the INPRDY bit (E0CSR.1). An interrupt will be generated when an IN packet is transmitted successfully. Note that no interrupt will be generated if an IN request is received before firmware has loaded a packet into the Endpoint0 FIFO. If the requested data exceeds the maximum packet size for Endpoint0 (as reported to the host), the data should be split into multiple packets; each packet should be of the maximum packet size excluding the last (residual) packet. If the requested data is an integer multiple of the maximum packet size for Endpoint0, the last data packet should be a zero-length packet signaling the end of the transfer. Firmware should set the DATAEND bit to '1' after loading into the Endpoint0 FIFO the last data packet for a transfer.

Upon reception of the first IN token for a particular control transfer, Endpoint0 is said to be in Transmit Mode. In this mode, only IN tokens should be sent by the host to Endpoint0. The SUEND bit (E0CSR.4) is set to '1' if a SETUP or OUT token is received while Endpoint0 is in Transmit Mode.

Endpoint0 will remain in Transmit Mode until any of the following occur:

1. USB0 receives an Endpoint0 SETUP or OUT token.
2. Firmware sends a packet less than the maximum Endpoint0 packet size.
3. Firmware sends a zero-length packet.

Firmware should set the DATAEND bit (E0CSR.3) to '1' when performing (2) and (3) above.

The SIE will transmit a NAK in response to an IN token if there is no packet ready in the IN FIFO (INPRDY = '0').

16.10.3.Endpoint0 OUT Transactions

When a SETUP request is received that requires the host to transmit data to USB0, one or more OUT requests will be sent by the host. When an OUT packet is successfully received by USB0, hardware will set the OPRDY bit (E0CSR.0) to ‘1’ and generate an Endpoint0 interrupt. Following this interrupt, firmware should unload the OUT packet from the Endpoint0 FIFO and set the SOPRDY bit (E0CSR.6) to ‘1’.

If the amount of data required for the transfer exceeds the maximum packet size for Endpoint0, the data will be split into multiple packets. If the requested data is an integer multiple of the maximum packet size for Endpoint0 (as reported to the host), the host will send a zero-length data packet signaling the end of the transfer.

Upon reception of the first OUT token for a particular control transfer, Endpoint0 is said to be in Receive Mode. In this mode, only OUT tokens should be sent by the host to Endpoint0. The SUEND bit (E0CSR.4) is set to ‘1’ if a SETUP or IN token is received while Endpoint0 is in Receive Mode.

Endpoint0 will remain in Receive mode until:

1. The SIE receives a SETUP or IN token.
2. The host sends a packet less than the maximum Endpoint0 packet size.
3. The host sends a zero-length packet.

Firmware should set the DATAEND bit (E0CSR.3) to ‘1’ when the expected amount of data has been received. The SIE will transmit a STALL condition if the host sends an OUT packet after the DATAEND bit has been set by firmware. An interrupt will be generated with the STSTL bit (E0CSR.2) set to ‘1’ after the STALL is transmitted.

USB Register Definition 16.17. E0CSR: USB0 Endpoint0 Control

R/W	R/W	R/W	R	R/W	R/W	R/W	R	Reset Value
SSUEND	SOPRDY	SDSTL	SUEND	DATAEND	STSTL	INPRDY	OPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x11

Bit7: SSUEND: Serviced Setup End
Write: Software should set this bit to ‘1’ after servicing a Setup End (bit SUEND) event. Hardware clears the SUEND bit when software writes ‘1’ to SSUEND.
Read: This bit always reads ‘0’.

Bit6: SOPRDY: Serviced OPRDY
Write: Software should write ‘1’ to this bit after servicing a received Endpoint0 packet. The OPRDY bit will be cleared by a write of ‘1’ to SOPRDY.
Read: This bit always reads ‘0’.

Bit5: SDSTL: Send Stall
Software can write ‘1’ to this bit to terminate the current transfer (due to an error condition, unexpected transfer request, etc.). Hardware will clear this bit to ‘0’ when the STALL handshake is transmitted.

Bit4: SUEND: Setup End
Hardware sets this read-only bit to ‘1’ when a control transaction ends before software has written ‘1’ to the DATAEND bit. Hardware clears this bit when software writes ‘1’ to SSUEND.

Bit3: DATAEND: Data End
Software should write ‘1’ to this bit:
1. When writing ‘1’ to INPRDY for the last outgoing data packet.
2. When writing ‘1’ to INPRDY for a zero-length data packet.
3. When writing ‘1’ to SOPRDY after servicing the last incoming data packet.

This bit is automatically cleared by hardware.

Bit2: STSTL: Sent Stall
Hardware sets this bit to ‘1’ after transmitting a STALL handshake signal. This flag must be cleared by software.

Bit1: INPRDY: IN Packet Ready
Software should write ‘1’ to this bit after loading a data packet into the Endpoint0 FIFO for transmit. Hardware clears this bit and generates an interrupt under either of the following conditions:
1. The packet is transmitted.
2. The packet is overwritten by an incoming SETUP packet.
3. The packet is overwritten by an incoming OUT packet.

Bit0: OPRDY: OUT Packet Ready
Hardware sets this read-only bit and generates an interrupt when a data packet has been received. This bit is cleared only when software writes ‘1’ to the SOPRDY bit.

USB Register Definition 16.18. E0CNT: USB0 Endpoint 0 Data Count

R	R	R	R	R	R	R	R	Reset Value
-	E0CNT							00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x16

Bit7: Unused. Read = 0; Write = don't care.
Bits6–0: E0CNT: Endpoint 0 Data Count
This 7-bit number indicates the number of received data bytes in the Endpoint 0 FIFO. This number is only valid while bit OPRDY is a '1'.

16.11. Configuring Endpoints1-3

Endpoints1-3 are configured and controlled through their own sets of the following control/status registers: IN registers EINCSRL and EINCSRH, and OUT registers EOUTCSRL and EOUTCSRH. Only one set of endpoint control/status registers is mapped into the USB register address space at a time, defined by the contents of the INDEX register (USB Register Definition 16.4).

Endpoints1-3 can be configured as IN, OUT, or both IN/OUT (Split Mode) as described in **Section 16.5.1**. The endpoint mode (Split/Normal) is selected via the SPLIT bit in register EINCSRH.

When SPLIT = '1', the corresponding endpoint FIFO is split, and both IN and OUT pipes are available.

When SPLIT = '0', the corresponding endpoint functions as either IN or OUT; the endpoint direction is selected by the DIRSEL bit in register EINCSRH.

16.12. Controlling Endpoints1-3 IN

Endpoints1-3 IN are managed via USB registers EINCSRL and EINCSRH. All IN endpoints can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing '1' to the ISO bit in register EINCSRH. Bulk and Interrupt transfers are handled identically by hardware.

An Endpoint1-3 IN interrupt is generated by any of the following conditions:

1. An IN packet is successfully transferred to the host.
2. Software writes '1' to the FLUSH bit (EINCSRL.3) when the target FIFO is not empty.
3. Hardware generates a STALL condition.

16.12.1.Endpoints1-3 IN Interrupt or Bulk Mode

When the ISO bit (EINCSRH.6) = '0' the target endpoint operates in Bulk or Interrupt Mode. Once an endpoint has been configured to operate in Bulk/Interrupt IN mode (typically following an Endpoint0 SET_INTERFACE command), firmware should load an IN packet into the endpoint IN FIFO and set the INPRDY bit (EINCSRL.0). Upon reception of an IN token, hardware will transmit the data, clear the INPRDY bit, and generate an interrupt.

Writing ‘1’ to INPRDY without writing any data to the endpoint FIFO will cause a zero-length packet to be transmitted upon reception of the next IN token.

A Bulk or Interrupt pipe can be shut down (or Halted) by writing ‘1’ to the SDSTL bit (EINCSRL.4). While SDSTL = ‘1’, hardware will respond to all IN requests with a STALL condition. Each time hardware generates a STALL condition, an interrupt will be generated and the STSTL bit (EINCSRL.5) set to ‘1’. The STSTL bit must be reset to ‘0’ by firmware.

Hardware will automatically reset INPRDY to ‘0’ when a packet slot is open in the endpoint FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for firmware to load two packets into the IN FIFO at a time. In this case, hardware will reset INPRDY to ‘0’ immediately after firmware loads the first packet into the FIFO and sets INPRDY to ‘1’. An interrupt will not be generated in this case; an interrupt will only be generated when a data packet is transmitted.

When firmware writes ‘1’ to the FCDT bit (EINCSRH.3), the data toggle for each IN packet will be toggled continuously, regardless of the handshake received from the host. This feature is typically used by Interrupt endpoints functioning as rate feedback communication for Isochronous endpoints. When FCDT = ‘0’, the data toggle bit will only be toggled when an ACK is sent from the host in response to an IN packet.

16.12.2.Endpoints1-3 IN Isochronous Mode

When the ISO bit (EINCSRH.6) is set to ‘1’, the target endpoint operates in Isochronous (ISO) mode. Once an endpoint has been configured for ISO IN mode, the host will send one IN token (data request) per frame; the location of data within each frame may vary. Because of this, it is recommended that double buffering be enabled for ISO IN endpoints.

Hardware will automatically reset INPRDY (EINCSRL.0) to ‘0’ when a packet slot is open in the endpoint FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for firmware to load two packets into the IN FIFO at a time. In this case, hardware will reset INPRDY to ‘0’ immediately after firmware loads the first packet into the FIFO and sets INPRDY to ‘1’. An interrupt will not be generated in this case; an interrupt will only be generated when a data packet is transmitted.

If there is not a data packet ready in the endpoint FIFO when USB0 receives an IN token from the host, USB0 will transmit a zero-length data packet and set the UNDRUN bit (EINCSRL.2) to ‘1’.

The ISO Update feature (see **Section 16.7**) can be useful in starting a double buffered ISO IN endpoint. If the host has already set up the ISO IN pipe (has begun transmitting IN tokens) when firmware writes the first data packet to the endpoint FIFO, the next IN token may arrive and the first data packet sent before firmware has written the second (double buffered) data packet to the FIFO. The ISO Update feature ensures that any data packet written to the endpoint FIFO will not be transmitted during the current frame; the packet will only be sent after a SOF signal has been received.

USB Register Definition 16.19. EINCSRL: USB0 IN Endpoint Control Low Byte

R	W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	CLRDT	STSTL	SDSTL	FLUSH	UNDRUN	FIFONE	INPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x11

Bit7: Unused. Read = 0; Write = don't care.

Bit6: CLRDT: Clear Data Toggle.
Write: Software should write '1' to this bit to reset the IN Endpoint data toggle to '0'.
Read: This bit always reads '0'.

Bit5: STSTL: Sent Stall
Hardware sets this bit to '1' when a STALL handshake signal is transmitted. The FIFO is flushed, and the INPRDY bit cleared. This flag must be cleared by software.

Bit4: SDSTL: Send Stall.
Software should write '1' to this bit to generate a STALL handshake in response to an IN token. Software should write '0' to this bit to terminate the STALL signal. This bit has no effect in ISO mode.

Bit3: FLUSH: FIFO Flush.
Writing a '1' to this bit flushes the next packet to be transmitted from the IN Endpoint FIFO. The FIFO pointer is reset and the INPRDY bit is cleared. If the FIFO contains multiple packets, software must write '1' to FLUSH for each packet. Hardware resets the FLUSH bit to '0' when the FIFO flush is complete.

Bit2: UNDRUN: Data Underrun.
The function of this bit depends on the IN Endpoint mode:
Isochronous: Set when a zero-length packet is sent after an IN token is received while bit INPRDY = '0'.
Interrupt/Bulk: This bit is not used in these modes and will always read a '0'.
This bit must be cleared by software.

Bit1: FIFONE: FIFO Not Empty.
0: The IN Endpoint FIFO is empty.
1. The IN Endpoint FIFO contains one or more packets.

Bit0: INPRDY: In Packet Ready.
Software should write '1' to this bit after loading a data packet into the IN Endpoint FIFO. Hardware clears INPRDY due to any of the following:
1. A data packet is transmitted.
2. Double buffering is enabled (DBIEN = '1') and there is an open FIFO packet slot.
3. If the endpoint is in Isochronous Mode (ISO = '1') and ISOUD = '1', INPRDY will read '0' until the next SOF is received.
An interrupt (if enabled) will be generated when hardware clears INPRDY as a result of a packet being transmitted.

USB Register Definition 16.20. EINCSRH: USB0 IN Endpoint Control High Byte

R/W	R/W	R/W	R	R/W	R/W	R	R	Reset Value
DBIEN	ISO	DIRSEL	-	FCDT	SPLIT	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x12

Bit7: DBIEN: IN Endpoint Double-buffer Enable.
0: Double-buffering disabled for the selected IN endpoint.
1: Double-buffering enabled for the selected IN endpoint.

Bit6: ISO: Isochronous Transfer Enable.
This bit enables/disables isochronous transfers on the current endpoint.
0: Endpoint configured for bulk/interrupt transfers.
1: Endpoint configured for isochronous transfers.

Bit5: DIRSEL: Endpoint Direction Select.
This bit is valid only when the selected FIFO is not split (SPLIT = '0').
0: Endpoint direction selected as OUT.
1: Endpoint direction selected as IN.

Bit4: Unused. Read = '0'. Write = don't care.

Bit3: FCDT: Force Data Toggle.
0: Endpoint data toggle switches only when an ACK is received following a data packet transmission.
1: Endpoint data toggle forced to switch after every data packet is transmitted, regardless of ACK reception.

Bit2: SPLIT: FIFO Split Enable.
When SPLIT = '1', the selected endpoint FIFO is split. The upper half of the selected FIFO is used by the IN endpoint; the lower half of the selected FIFO is used by the OUT endpoint.

Bits1–0: Unused. Read = 00b; Write = don't care.

16.13. Controlling Endpoints1-3 OUT

Endpoints1-3 OUT are managed via USB registers EOUTCSRL and EOUTCSRH. All OUT endpoints can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing '1' to the ISO bit in register EOUTCSRH. Bulk and Interrupt transfers are handled identically by hardware.

An Endpoint1-3 OUT interrupt may be generated by the following:

1. Hardware sets the OPRDY bit (EINCSRL.0) to '1'.
2. Hardware generates a STALL condition.

16.13.1.Endpoints1-3 OUT Interrupt or Bulk Mode

When the ISO bit (EOUTCSRH.6) = '0' the target endpoint operates in Bulk or Interrupt mode. Once an endpoint has been configured to operate in Bulk/Interrupt OUT mode (typically following an Endpoint0 SET_INTERFACE command), hardware will set the OPRDY bit (EOUTCSRL.0) to '1' and generate an interrupt upon reception of an OUT token and data packet. The number of bytes in the current OUT data packet (the packet ready to be unloaded from the FIFO) is given in the EOUTCNTH and EOUTCNTL registers. In response to this interrupt, firmware should unload the data packet from the OUT FIFO and reset the OPRDY bit to '0'.

A Bulk or Interrupt pipe can be shut down (or Halted) by writing ‘1’ to the SDSTL bit (EOUTCSRL.5). While SDSTL = ‘1’, hardware will respond to all OUT requests with a STALL condition. Each time hardware generates a STALL condition, an interrupt will be generated and the STSTL bit (EOUTCSRL.6) set to ‘1’. The STSTL bit must be reset to ‘0’ by firmware.

Hardware will automatically set OPRDY when a packet is ready in the OUT FIFO. Note that if double buffering is enabled for the target endpoint, it is possible for two packets to be ready in the OUT FIFO at a time. In this case, hardware will set OPRDY to ‘1’ immediately after firmware unloads the first packet and resets OPRDY to ‘0’. A second interrupt will be generated in this case.

16.13.2.Endpoints1-3 OUT Isochronous Mode

When the ISO bit (EOUTCSRH.6) is set to ‘1’, the target endpoint operates in Isochronous (ISO) mode. Once an endpoint has been configured for ISO OUT mode, the host will send exactly one data per USB frame; the location of the data packet within each frame may vary, however. Because of this, it is recommended that double buffering be enabled for ISO OUT endpoints.

Each time a data packet is received, hardware will load the received data packet into the endpoint FIFO, set the OPRDY bit (EOUTCSRL.0) to ‘1’, and generate an interrupt (if enabled). Firmware would typically use this interrupt to unload the data packet from the endpoint FIFO and reset the OPRDY bit to ‘0’.

If a data packet is received when there is no room in the endpoint FIFO, an interrupt will be generated and the OVRUN bit (EOUTCSRL.2) set to ‘1’. If USB0 receives an ISO data packet with a CRC error, the data packet will be loaded into the endpoint FIFO, OPRDY will be set to ‘1’, an interrupt (if enabled) will be generated, and the DATAERR bit (EOUTCSRL.3) will be set to ‘1’. Software should check the DATAERR bit each time a data packet is unloaded from an ISO OUT endpoint FIFO.

USB Register Definition 16.21. EOUTCSRL: USB0 OUT Endpoint Control Low Byte

W	R/W	R/W	R/W	R	R/W	R	R/W	Reset Value
CLRDT	STSTL	SDSTL	FLUSH	DATERR	OVRUN	FIFOFUL	OPRDY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x14

Bit7: CLRDT: Clear Data Toggle
Write: Software should write '1' to this bit to reset the OUT endpoint data toggle to '0'.
Read: This bit always reads '0'.

Bit6: STSTL: Sent Stall
Hardware sets this bit to '1' when a STALL handshake signal is transmitted. This flag must be cleared by software.

Bit5: SDSTL: Send Stall
Software should write '1' to this bit to generate a STALL handshake. Software should write '0' to this bit to terminate the STALL signal. This bit has no effect in ISO mode.

Bit4: FLUSH: FIFO Flush
Writing a '1' to this bit flushes the next packet to be read from the OUT endpoint FIFO. The FIFO pointer is reset and the OPRDY bit is cleared. If the FIFO contains multiple packets, software must write '1' to FLUSH for each packet. Hardware resets the FLUSH bit to '0' when the FIFO flush is complete.

Note: If data for the current packet has already been read from the FIFO, the FLUSH bit should not be used to flush the packet. Instead, the entire data packet should be read from the FIFO manually.

Bit3: DATERR: Data Error
In ISO mode, this bit is set by hardware if a received packet has a CRC or bit-stuffing error. It is cleared when software clears OPRDY. This bit is only valid in ISO mode.

Bit2: OVRUN: Data Overrun
This bit is set by hardware when an incoming data packet cannot be loaded into the OUT endpoint FIFO. This bit is only valid in ISO mode, and must be cleared by software.
0: No data overrun.
1: A data packet was lost because of a full FIFO since this flag was last cleared.

Bit1: FIFOFUL: OUT FIFO Full
This bit indicates the contents of the OUT FIFO. If double buffering is enabled for the endpoint (DBIEN = '1'), the FIFO is full when the FIFO contains two packets. If DBIEN = '0', the FIFO is full when the FIFO contains one packet.
0: OUT endpoint FIFO is not full.
1: OUT endpoint FIFO is full.

Bit0: OPRDY: OUT Packet Ready
Hardware sets this bit to '1' and generates an interrupt when a data packet is available. Software should clear this bit after each data packet is unloaded from the OUT endpoint FIFO.

USB Register Definition 16.22. EOUTCSRH: USB0 OUT Endpoint Control High Byte

R/W	R/W	R/W	R/W	R	R	R	R	Reset Value
DBOEN	ISO	-	-	-	-	-	-	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x15

Bit7: DBOEN: Double-buffer Enable
0: Double-buffering disabled for the selected OUT endpoint.
1: Double-buffering enabled for the selected OUT endpoint.
Bit6: ISO: Isochronous Transfer Enable
This bit enables/disables isochronous transfers on the current endpoint.
0: Endpoint configured for bulk/interrupt transfers.
1: Endpoint configured for isochronous transfers.
Bits5–0: Unused. Read = 000000b; Write = don't care.

USB Register Definition 16.23. EOUTCNTL: USB0 OUT Endpoint Count Low

R	R	R	R	R	R	R	R	Reset Value
EOCL								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x16

Bits7–0: EOCL: OUT Endpoint Count Low Byte
EOCL holds the lower 8-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = '1'.

USB Register Definition 16.24. EOUTCNTH: USB0 OUT Endpoint Count High

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	-	-	E0CH		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x17

Bits7–2: Unused. Read = 00000. Write = don't care.
Bits1–0: EOCH: OUT Endpoint Count High Byte
EOCH holds the upper 2-bits of the 10-bit number of data bytes in the last received packet in the current OUT endpoint FIFO. This number is only valid while OPRDY = '1'.

Table 16.4. USB Transceiver Electrical Characteristics

V_{DD} = 3.0 to 3.6 V, –40 to +85 °C unless otherwise specified

Parameters	Symbol	Conditions	Min	Typ	Max	Units
Transmitter						
Output High Voltage	V_{OH}		2.8			V
Output Low Voltage	V_{OL}				0.8	V
Output Crossover Point	V_{CRS}		1.3		2.0	V
Output Impedance	Z_{DRV}	Driving High Driving Low		38 38		Ω
Pull-up Resistance	R_{PU}	Full Speed (D+ Pull-up) Low Speed (D– Pull-up)	1.425	1.5	1.575	kΩ
Output Rise Time	T_R	Low Speed Full Speed	75 4		300 20	ns
Output Fall Time	T_F	Low Speed Full Speed	75 4		300 20	ns
Receiver						
Differential Input Sensitivity	V_{DI}	| (D+) – (D–) |	0.2			V
Differential Input Common Mode Range	V_{CM}		0.8		2.5	V
Input Leakage Current	I_L	Pullups Disabled		<1.0		µA

Note: Refer to the USB Specification for timing diagrams and symbol definitions.

17. 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 I2C 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 1/20th 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.

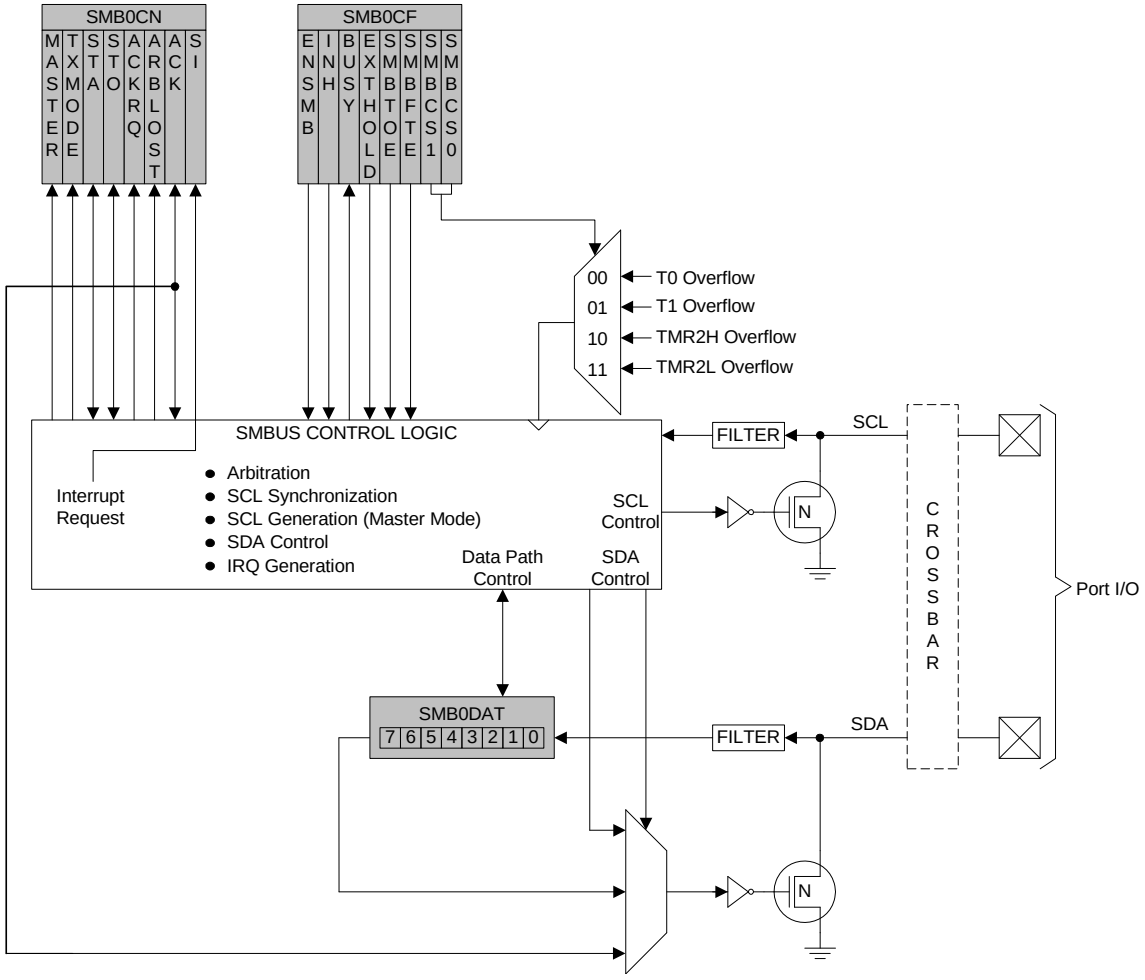


Figure 17.1. SMBus Block Diagram

17.1. Supporting Documents

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

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

17.2. SMBus Configuration

Figure 17.2 shows a typical SMBus configuration. The SMBus specification allows any recessive voltage between 3.0 V and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pull-up resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

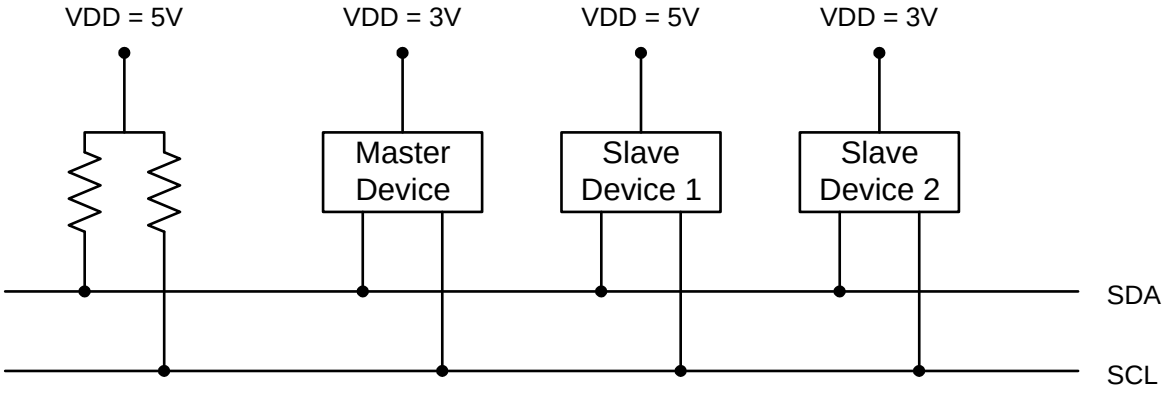


Figure 17.2. Typical SMBus Configuration

17.3. 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 17.3). 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 17.3 illustrates a typical SMBus transaction.

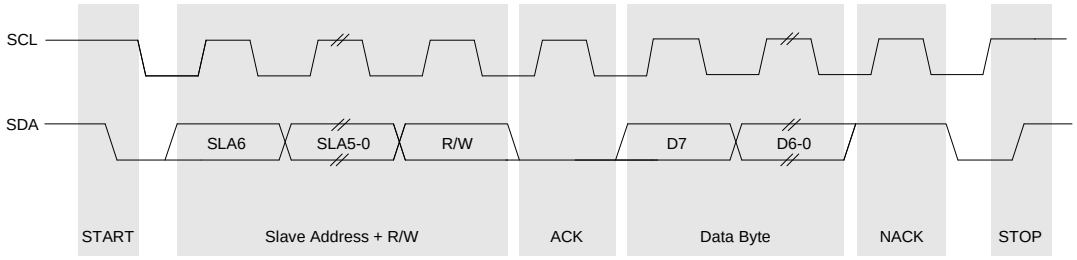


Figure 17.3. SMBus Transaction

17.3.1. 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 "17.3.4. SCL High (SMBus Free) Timeout" on page 191**). 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.

17.3.2. 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.

17.3.3. 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.

17.3.4. 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.

17.4. 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 “17.5. SMBus Transfer Modes” on page 198** 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 “17.4.2. SMB0CN Control Register” on page 195**; Table 17.4 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 "17.4.1. SMBus Configuration Register" on page 192**.

17.4.1. 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 17.1. 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 17.1. 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 235**.

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

Equation 17.1. 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 17.1. 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 17.2.

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

Equation 17.2. Typical SMBus Bit Rate

Figure 17.4 shows the typical SCL generation described by Equation 17.2. 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 17.1.

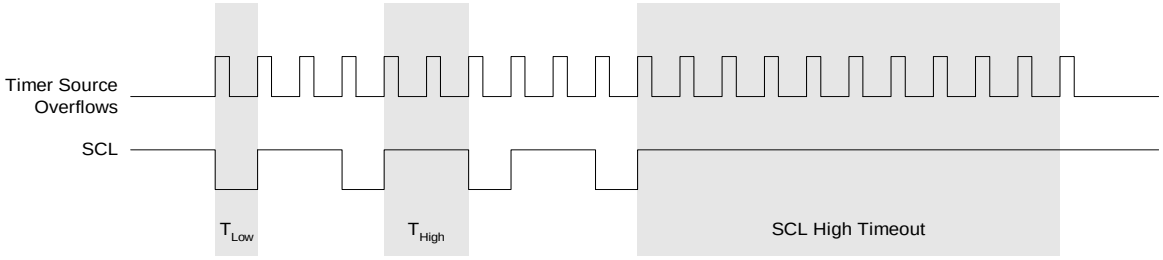


Figure 17.4. Typical SMBus SCL Generation

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 17.2 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 17.2. 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 "17.3.3. SCL Low Timeout" on page 191**). 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 17.4). 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 17.1. SMB0CF: SMBus Clock/Configuration

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS1	SMBCS0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC1

Bit7: ENSMB: SMBus Enable.
This bit enables/disables the SMBus interface. When enabled, the interface constantly monitors the SDA and SCL pins.
0: SMBus interface disabled.
1: SMBus interface enabled.

Bit6: INH: SMBus Slave Inhibit.
When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
0: SMBus Slave Mode enabled.
1: SMBus Slave Mode inhibited.

Bit5: BUSY: SMBus Busy Indicator.
This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.

Bit4: EXTHOLD: SMBus Setup and Hold Time Extension Enable.
This bit controls the SDA setup and hold times according to.
0: SDA Extended Setup and Hold Times disabled.
1: SDA Extended Setup and Hold Times enabled.

Bit3: SMBTOE: SMBus SCL Timeout Detection Enable.
This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.

Bit2: SMBFTE: SMBus Free Timeout Detection Enable.
When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.

Bits1–0: SMBCS1-SMBCS0: SMBus Clock Source Selection.
These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 17.1.

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

17.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information (see SFR Definition 17.2). 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 17.3 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 17.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 17.4 for SMBus status decoding using the SMB0CN register.

SFR Definition 17.2. SMB0CN: SMBus Control

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 Addressable

SFR Address: 0xC0

Bit7: 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.

Bit6: 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.

Bit5: 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.

Bit4: 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).

Bit3: 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.

Bit2: 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.

Bit1: 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).

Bit0: SI: SMBus Interrupt Flag.
This bit is set by hardware under the conditions listed in Table 17.3. SI must be cleared by software. While SI is set, SCL is held low and the SMBus is stalled.

Table 17.3. 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.

17.4.3. 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 17.3. SMB0DAT: SMBus Data

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: 0xC2

Bits7-0: 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.

17.5. 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.

17.5.1. 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 17.5 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.

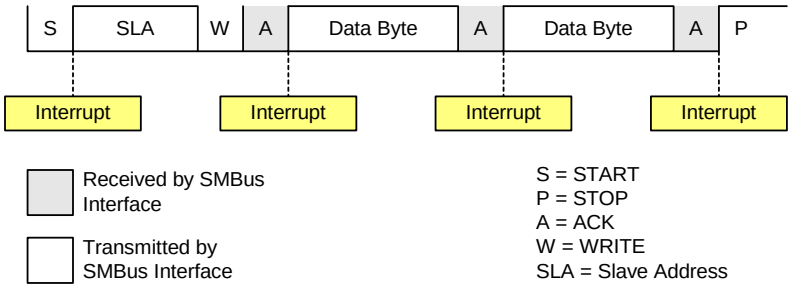


Figure 17.5. Typical Master Transmitter Sequence

17.5.2. 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. Note that the interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 17.6 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.

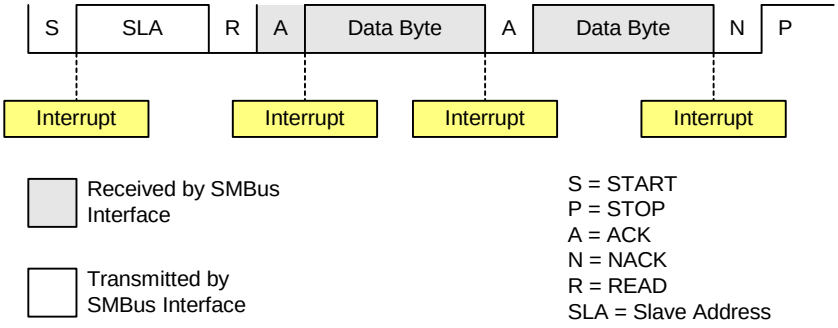


Figure 17.6. Typical Master Receiver Sequence

17.5.3. 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 17.7 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.

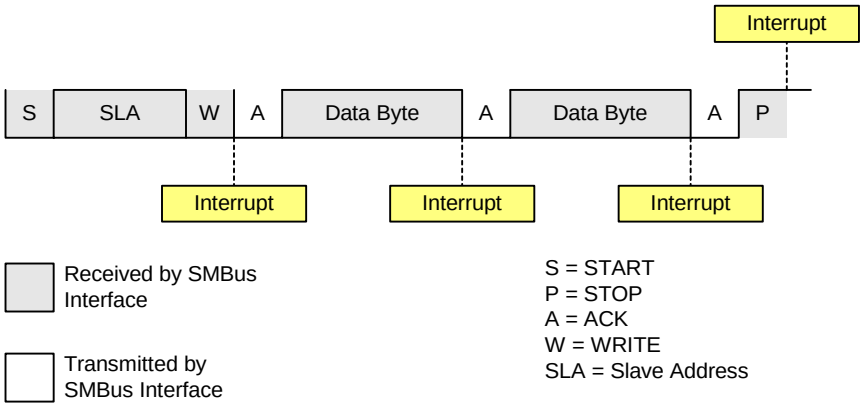


Figure 17.7. Typical Slave Receiver Sequence

17.5.4. 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 17.8 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.

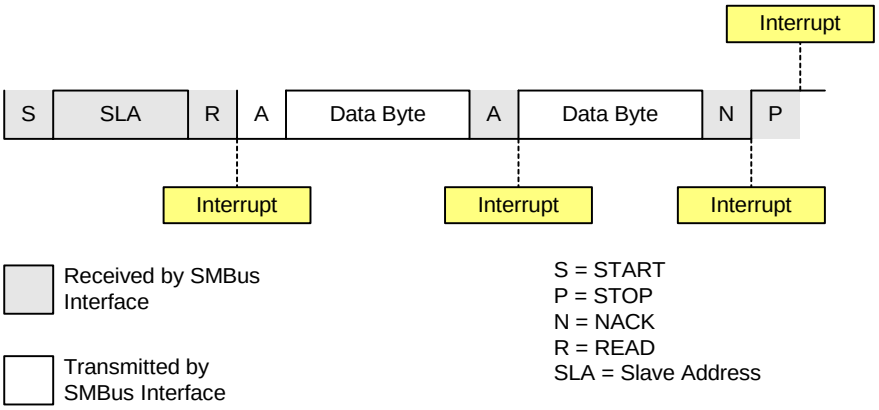


Figure 17.8. Typical Slave Transmitter Sequence

17.6. 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. Note that 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 17.4. 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
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

Table 17.4. SMBus Status Decoding (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STo	ACK
Slave 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	An illegal STOP or bus error was detected while a Slave Transmission was in progress.	Clear STO.	0	0	X
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
	0010	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 addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	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

18. 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 "18.1. Enhanced Baud Rate Generation" on page 206**). 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).

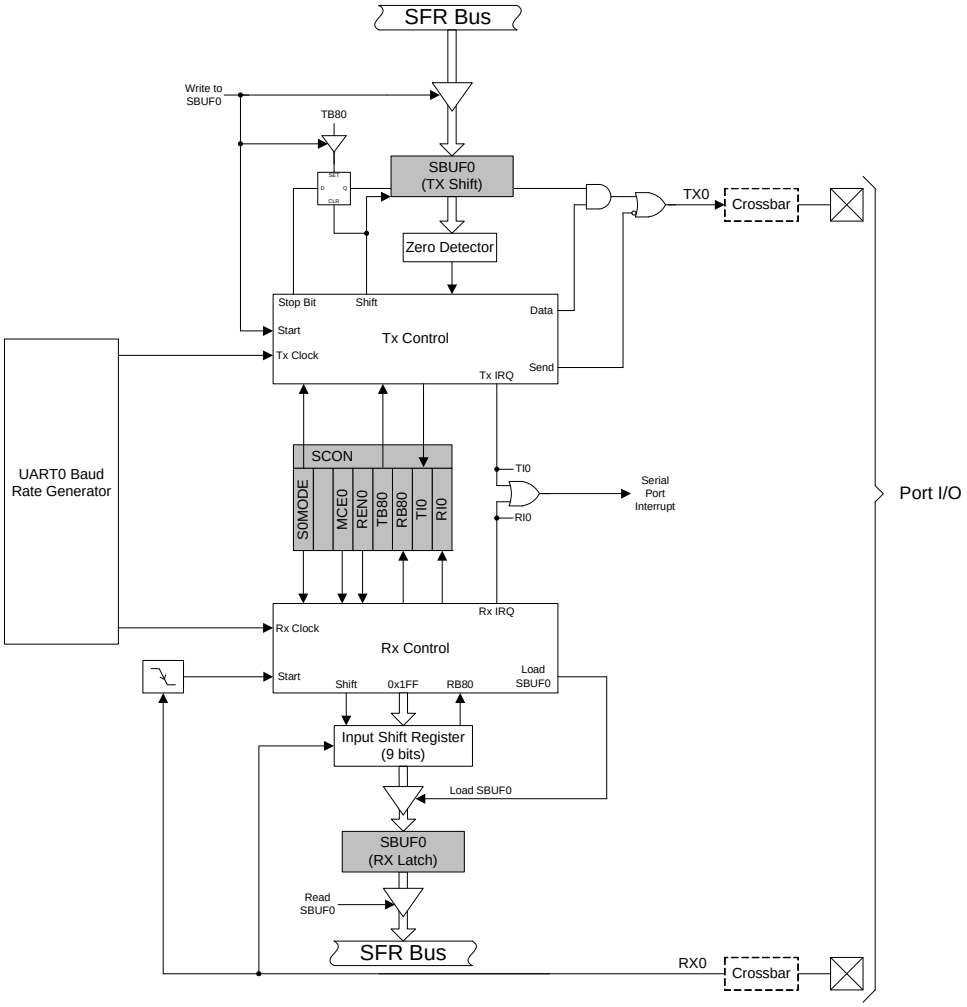


Figure 18.1. UART0 Block Diagram

18.1. 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 18.2), 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.

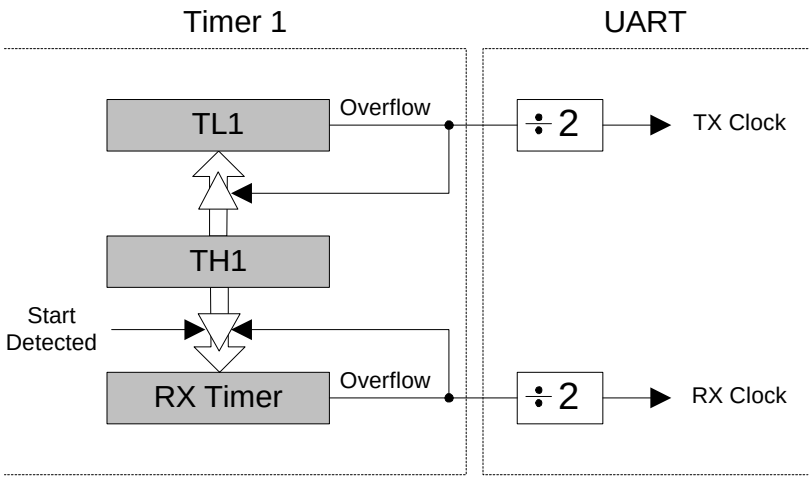


Figure 18.2. UART0 Baud Rate Logic

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 237**). 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 18.1.

$$UartBaudRate = \frac{T1_{CLK}}{(256 - T1H)} \times \frac{1}{2}$$

Equation 18.1. 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 235**. A quick reference for typical baud rates using the internal oscillator is given in Table 18.1. Note that the internal oscillator may still generate the system clock if an external oscillator is driving Timer 1.

18.2. 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.

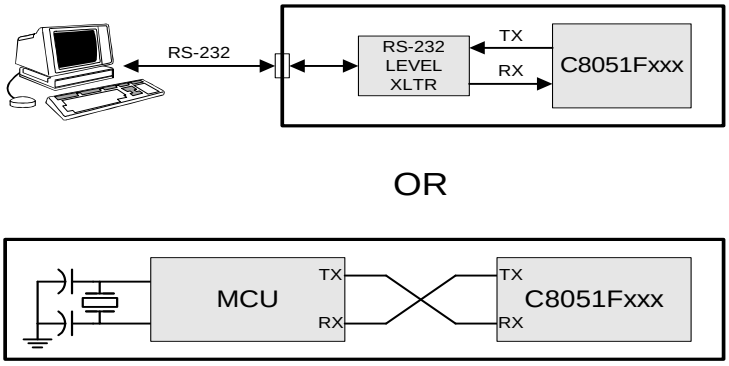


Figure 18.3. UART Interconnect Diagram

18.2.1. 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.

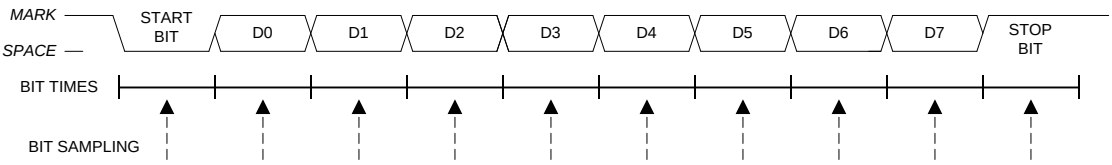


Figure 18.4. 8-Bit UART Timing Diagram

18.2.2. 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'.

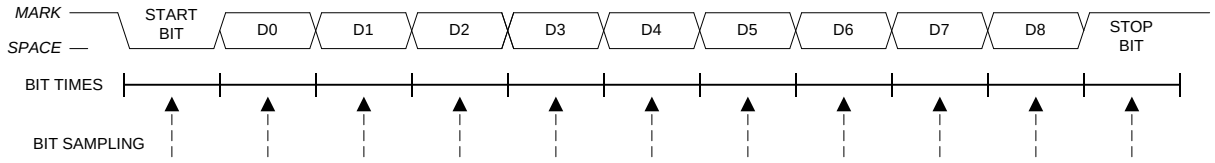


Figure 18.5. 9-Bit UART Timing Diagram

18.3. 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).

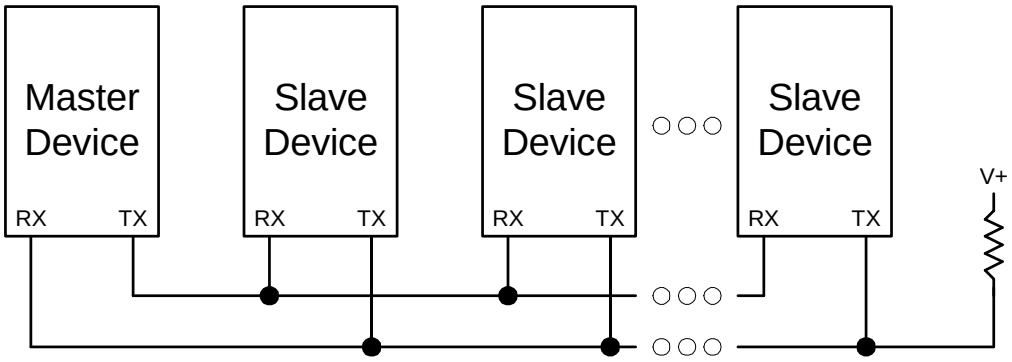


Figure 18.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 18.1. SCON0: Serial Port 0 Control

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 Addressable

SFR Address: 0x98

Bit7: 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.

Bit6: UNUSED. Read = 1b. Write = don't care.

Bit5: 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.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: 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.

Bit2: 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.

Bit1: 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.

Bit0: 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 18.2. SBUF0: Serial (UART0) Port Data Buffer

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: 0x99

Bits7–0: 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 18.1. Timer Settings for Standard Baud Rates Using the Internal Oscillator

	Target Baud Rate (bps)	Actual Baud Rate (bps)	Baud Rate Error	Oscillator Divide Factor	Timer Clock Source	SCA1-SCA0 (pre-scale select*	T1M*	Timer 1 Reload Value (hex)
SYSCLK = 12 MHz	230400	230769	0.16%	52	SYSCLK	XX	1	0xE6
	115200	115385	0.16%	104	SYSCLK	XX	1	0xCC
	57600	57692	0.16%	208	SYSCLK	XX	1	0x98
	28800	28846	0.16%	416	SYSCLK	XX	1	0x30
	14400	14423	0.16%	832	SYSCLK / 4	01	0	0x98
	9600	9615	0.16%	1248	SYSCLK / 4	01	0	0x64
	2400	2404	0.16%	4992	SYSCLK / 12	00	0	0x30
	1200	1202	0.16%	9984	SYSCLK / 48	10	0	0x98
SYSCLK = 24 MHz	230400	230769	0.16%	104	SYSCLK	XX	1	0xCC
	115200	115385	0.16%	208	SYSCLK	XX	1	0x98
	57600	57692	0.16%	416	SYSCLK	XX	1	0x30
	28800	28846	0.16%	832	SYSCLK / 4	01	0	0x98
	14400	14423	0.16%	1664	SYSCLK / 4	01	0	0x30
	9600	9615	0.16%	2496	SYSCLK / 12	00	0	0x98
	2400	2404	0.16%	9984	SYSCLK / 48	10	0	0x98
	1200	1202	0.16%	19968	SYSCLK / 48	10	0	0x30
SYSCLK = 48 MHz	230400	230769	0.16%	208	SYSCLK	XX	1	0x98
	115200	115385	0.16%	416	SYSCLK	XX	1	0x30
	57600	57692	0.16%	832	SYSCLK / 4	01	0	0x98
	28800	28846	0.16%	1664	SYSCLK / 4	01	0	0x30
	14400	14388	0.08%	3336	SYSCLK / 12	00	0	0x75
	9600	9615	0.16%	4992	SYSCLK / 12	00	0	0x30
	2400	2404	0.16%	19968	SYSCLK / 48	10	0	0x30

X = Don't care

***Note:** SCA1-SCA0 and T1M define the Timer Clock Source. Bit definitions for these values can be found in Section 21.1.

19. UART1 (C8051F340/1/4/5/8/A/B/C Only)

UART1 is an asynchronous, full duplex serial port offering a variety of data formatting options. A dedicated baud rate generator with a 16-bit timer and selectable prescaler is included, which can generate a wide range of baud rates (details in **Section "19.1. Baud Rate Generator" on page 214**). A received data FIFO allows UART1 to receive up to three data bytes before data is lost and an overflow occurs.

UART1 has six associated SFRs. Three are used for the Baud Rate Generator (SBCON1, SBRLH1, and SBRLL1), two are used for data formatting, control, and status functions (SCON1, SMOD1), and one is used to send and receive data (SBUF1). The single SBUF1 location provides access to both the transmit holding register and the receive FIFO. **Writes to SBUF1 always access the Transmit Holding Register. Reads of SBUF1 always access the first byte of the Receive FIFO; it is not possible to read data from the Transmit Holding Register.**

With UART1 interrupts enabled, an interrupt is generated each time a transmit is completed (TI1 is set in SCON1), or a data byte has been received (RI1 is set in SCON1). The UART1 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 UART1 interrupt (transmit complete or receive complete). Note that if additional bytes are available in the Receive FIFO, the RI1 bit cannot be cleared by software.

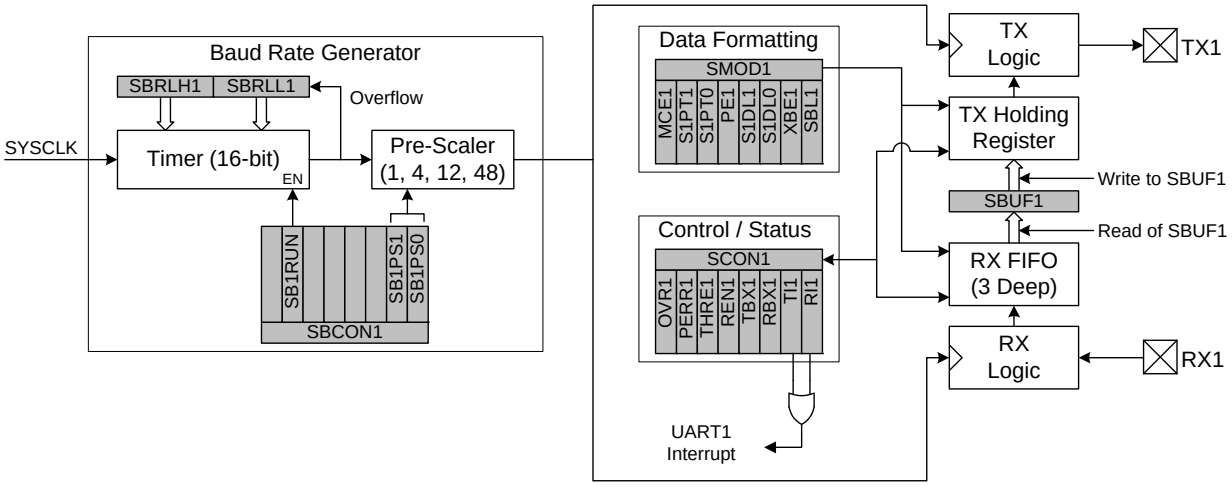


Figure 19.1. UART1 Block Diagram

19.1. Baud Rate Generator

The UART1 baud rate is generated by a dedicated 16-bit timer which runs from the controller's core clock (SYSCLK), and has prescaler options of 1, 4, 12, or 48. The timer and prescaler options combined allow for a wide selection of baud rates over many SYSCLK frequencies.

The baud rate generator is configured using three registers: SBCON1, SBRLH1, and SBRLL1. The UART1 Baud Rate Generator Control Register (SBCON1, SFR Definition 19.4) enables or disables the baud rate generator, and selects the prescaler value for the timer. The baud rate generator must be enabled for UART1 to function. Registers SBRLH1 and SBRLL1 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16. The baud rate for UART1 is defined in Equation 19.1.

$$\text{Baud Rate} = \frac{\text{SYSCLK}}{(65536 - (\text{SBRLH1:SBRLL1}))} \times \frac{1}{2} \times \frac{1}{\text{Prescaler}}$$

Equation 19.1. UART1 Baud Rate

A quick reference for typical baud rates and system clock frequencies is given in Table 19.1.

Table 19.1. Baud Rate Generator Settings for Standard Baud Rates

	Target Baud Rate (bps)	Actual Baud Rate (bps)	Baud Rate Error	Oscillator Divide Factor	SB1PS[1:0] (Prescaler Bits)	Reload Value in SBRLH1:SBRLL1
SYSCLK = 12 MHz	230400	230769	0.16%	52	11	0xFFE6
	115200	115385	0.16%	104	11	0xFFCC
	57600	57692	0.16%	208	11	0xFF98
	28800	28846	0.16%	416	11	0xFF30
	14400	14388	0.08%	834	11	0xFE5F
	9600	9600	0.0%	1250	11	0xFD8F
	2400	2400	0.0%	5000	11	0xF63C
	1200	1200	0.0%	10000	11	0xEC78
SYSCLK = 24 MHz	230400	230769	0.16%	104	11	0xFFCC
	115200	115385	0.16%	208	11	0xFF98
	57600	57692	0.16%	416	11	0xFF30
	28800	28777	0.08%	834	11	0xFE5F
	14400	14406	0.04%	1666	11	0xFCBF
	9600	9600	0.0%	2500	11	0xFB1E
	2400	2400	0.0%	10000	11	0xEC78
	1200	1200	0.0%	20000	11	0xD8F0
SYSCLK = 48 MHz	230400	230769	0.16%	208	11	0xFF98
	115200	115385	0.16%	416	11	0xFF30
	57600	57554	0.08%	834	11	0xFE5F
	28800	28812	0.04%	1666	11	0xFCBF
	14400	14397	0.02%	3334	11	0xF97D
	9600	9600	0.0%	5000	11	0xF63C
	2400	2400	0.0%	20000	11	0xD8F0
	1200	1200	0.0%	40000	11	0xB1E0

19.2. Data Format

UART1 has a number of available options for data formatting. Data transfers begin with a start bit (logic low), followed by the data bits (sent LSB-first), a parity or extra bit (if selected), and end with one or two stop bits (logic high). The data length is variable between 5 and 8 bits. A parity bit can be appended to the data, and automatically generated and detected by hardware for even, odd, mark, or space parity. The stop bit length is selectable between short (1 bit time) and long (1.5 or 2 bit times), and a multi-processor communication mode is available for implementing networked UART buses. All of the data formatting options can be configured using the SMOD1 register, shown in SFR Definition 19.2. Figure 19.2 shows the timing for a UART1 transaction without parity or an extra bit enabled. Figure 19.3 shows the timing for a UART1 transaction with parity enabled (PE1 = 1). Figure 19.4 is an example of a UART1 transaction when the extra bit is enabled (XBE1 = 1). Note that the extra bit feature is not available when parity is enabled, and the second stop bit is only an option for data lengths of 6, 7, or 8 bits.

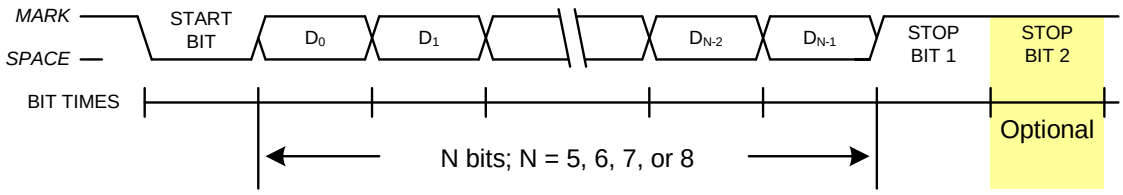


Figure 19.2. UART1 Timing Without Parity or Extra Bit

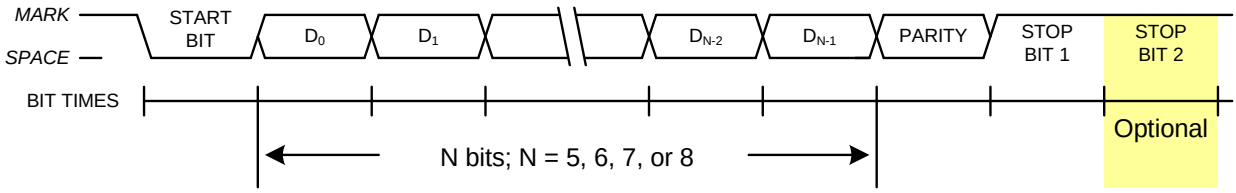


Figure 19.3. UART1 Timing With Parity

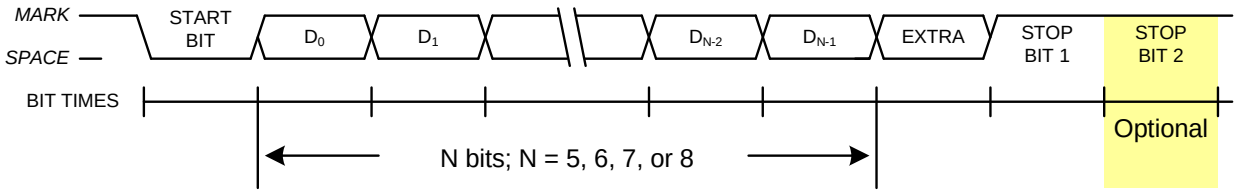


Figure 19.4. UART1 Timing With Extra Bit

19.3. Configuration and Operation

UART1 provides standard asynchronous, full duplex communication. It can operate in a point-to-point serial communications application, or as a node on a multi-processor serial interface. To operate in a point-to-point application, where there are only two devices on the serial bus, the MCE1 bit in SMOD1 should be cleared to '0'. For operation as part of a multi-processor communications bus, the MCE1 and XBE1 bits should both be set to '1'. In both types of applications, data is transmitted from the microcontroller on the TX1 pin, and received on the RX1 pin. The TX1 and RX1 pins are configured using the crossbar and the Port I/O registers, as detailed in **Section "15. Port Input/Output" on page 142**.

In typical UART communications, The transmit (TX) output of one device is connected to the receive (RX) input of the other device, either directly or through a bus transceiver, as shown in Figure 19.5.

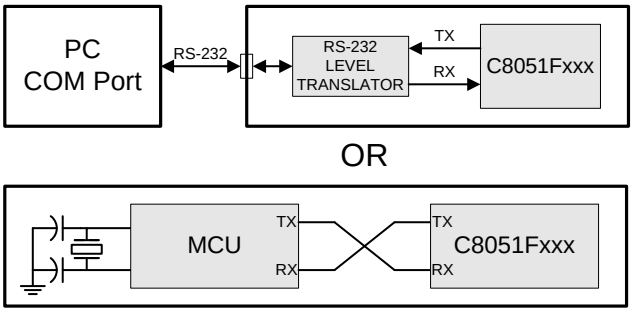


Figure 19.5. Typical UART Interconnect Diagram

19.3.1. Data Transmission

Data transmission is double-buffered, and begins when software writes a data byte to the SBUF1 register. Writing to SBUF1 places data in the Transmit Holding Register, and the Transmit Holding Register Empty flag (THRE1) will be cleared to '0'. If the UARTs shift register is empty (i.e., no transmission is in progress) the data will be placed in the shift register, and the THRE1 bit will be set to '1'. If a transmission is in progress, the data will remain in the Transmit Holding Register until the current transmission is complete. The TI1 Transmit Interrupt Flag (SCON1.1) will be set at the end of any transmission (the beginning of the stop-bit time). If enabled, an interrupt will occur when TI1 is set.

If the extra bit function is enabled (XBE1 = '1') and the parity function is disabled (PE1 = '0'), the value of the TBX1 (SCON1.3) bit will be sent in the extra bit position. When the parity function is enabled (PE1 = '1'), hardware will generate the parity bit according to the selected parity type (selected with S1PT[1:0]), and append it to the data field. Note: when parity is enabled, the extra bit function is not available.

19.3.2. Data Reception

Data reception can begin any time after the REN1 Receive Enable bit (SCON1.4) is set to logic 1. After the stop bit is received, the data byte will be stored in the receive FIFO if the following conditions are met: the receive FIFO (3 bytes deep) must not be full, and the stop bit(s) must be logic 1. In the event that the receive FIFO is full, the incoming byte will be lost, and a Receive FIFO Overrun Error will be generated (OVR1 in register SCON1 will be set to logic 1). If the stop bit(s) were logic 0, the incoming data will not be stored in the receive FIFO. If the reception conditions are met, the data is stored in the receive FIFO, and the RI1 flag will be set. Note: when MCE1 = '1', RI1 will only be set if the extra bit was equal to '1'. Data can be read from the receive FIFO by reading the SBUF1 register. The SBUF1 register represents the oldest byte in the FIFO. After SBUF1 is read, the next byte in the FIFO is immediately loaded into SBUF1, and

space is made available in the FIFO for another incoming byte. If enabled, an interrupt will occur when RI1 is set. RI1 can only be cleared to '0' by software when there is no more information in the FIFO. The recommended procedure to empty the FIFO contents is as follows:

1. Clear RI1 to '0'.
2. Read SBUF1.
3. Check RI1, and repeat at step 1 if RI1 is set to '1'.

If the extra bit function is enabled (XBE1 = ‘1’) and the parity function is disabled (PE1 = ‘0’), the extra bit for the oldest byte in the FIFO can be read from the RBX1 bit (SCON1.2). If the extra bit function is not enabled, the value of the stop bit for the oldest FIFO byte will be presented in RBX1. When the parity function is enabled (PE1 = ‘1’), hardware will check the received parity bit against the selected parity type (selected with S1PT[1:0]) when receiving data. If a byte with parity error is received, the PERR1 flag will be set to ‘1’. This flag must be cleared by software. Note: when parity is enabled, the extra bit function is not available.

19.3.3. Multiprocessor Communications

UART1 supports multiprocessor communication between a master processor and one or more slave processors by special use of the extra 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 extra bit is logic 1; in a data byte, the extra bit is always set to logic 0.

Setting the MCE1 bit (SMOD1.7) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the extra bit is logic 1 (RBX1 = 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 address. If the addresses match, the slave will clear its MCE1 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE1 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 MCE1 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).

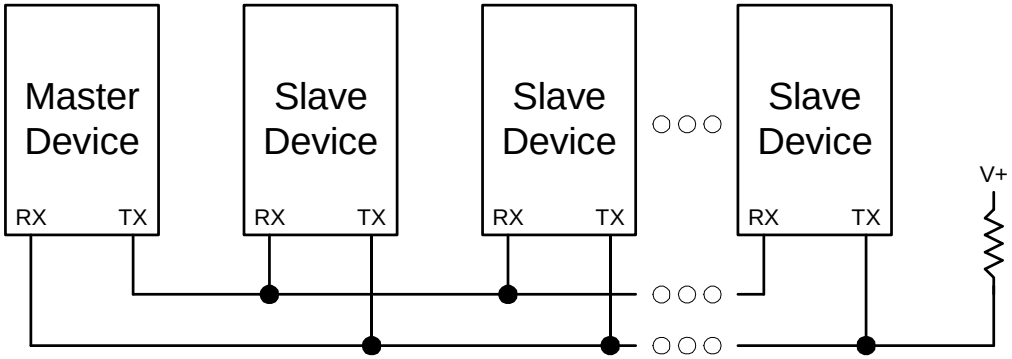


Figure 19.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 19.1. SCON1: UART1 Control

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
OVR1	PERR1	THRE1	REN1	TBX1	RBX1	TI1	RI1	00100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD2

Bit7: OVR1: Receive FIFO Overrun Flag.
This bit is used to indicate a receive FIFO overrun condition.
0: Receive FIFO Overrun has not occurred.
1: Receive FIFO Overrun has occurred (an incoming character was discarded due to a full FIFO).
This bit must be cleared to ‘0’ by software.

Bit6: PERR1: Parity Error Flag.
When parity is enabled, this bit is used to indicate that a parity error has occurred. It is set to ‘1’ when the parity of the oldest byte in the FIFO does not match the selected Parity Type.
0: Parity Error has not occurred.
1: Parity Error has occurred.
This bit must be cleared to ‘0’ by software.

Bit5: THRE1: Transmit Holding Register Empty Flag.
0: Transmit Holding Register not Empty - do not write to SBUF1.
1: Transmit Holding Register Empty - it is safe to write to SBUF1.

Bit4: REN1: Receive Enable.
This bit enables/disables the UART receiver. When disabled, bytes can still be read from the receive FIFO.
0: UART1 reception disabled.
1: UART1 reception enabled.

Bit3: TBX1: Extra Transmission Bit.
The logic level of this bit will be assigned to the extra transmission bit when XBE1 is set to ‘1’. This bit is not used when Parity is enabled.

Bit2: RBX1: Extra Receive Bit.
RBX1 is assigned the value of the extra bit when XBE1 is set to ‘1’. If XBE1 is cleared to ‘0’, RBX1 will be assigned the logic level of the first stop bit. This bit is not valid when Parity is enabled.

Bit1: TI1: Transmit Interrupt Flag.
Set to a ‘1’ by hardware after data has been transmitted, at the beginning of the STOP bit. When the UART1 interrupt is enabled, setting this bit causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI1: Receive Interrupt Flag.
Set to ‘1’ by hardware when a byte of data has been received by UART1 (set at the STOP bit sampling time). When the UART1 interrupt is enabled, setting this bit to ‘1’ causes the CPU to vector to the UART1 interrupt service routine. This bit must be cleared manually by software. Note that RI1 will remain set to '1' as long as there is still data in the UART FIFO. After the last byte has been shifted from the FIFO to SBUF1, RI1 can be cleared.

SFR Definition 19.2. SMOD1: UART1 Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
MCE1	S1PT1	S1PT0	PE1	S1DL1	S1DL0	XBE1	SBL1	00001100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE5

Bit7: MCE1: Multiprocessor Communication Enable.
0: RI will be activated if stop bit(s) are ‘1’.
1: RI will be activated if stop bit(s) and extra bit are ‘1’ (extra bit must be enabled using XBE1).
Note: This function is not available when hardware parity is enabled.

Bits6–5: S1PT[1:0]: Parity Type.
00: Odd
01: Even
10: Mark
11: Space

Bit4: PE1: Parity Enable.
This bit activates hardware parity generation and checking. The parity type is selected by bits S1PT1-0 when parity is enabled.
0: Hardware parity is disabled.
1: Hardware parity is enabled.

Bits3–2: S1DL[1:0]: Data Length.
00: 5-bit data
01: 6-bit data
10: 7-bit data
11: 8-bit data

Bit1: XBE1: Extra Bit Enable
When enabled, the value of TBX1 will be appended to the data field.
0: Extra Bit Disabled.
1: Extra Bit Enabled.

Bit0: SBL1: Stop Bit Length
0: Short - Stop bit is active for one bit time.
1: Long - Stop bit is active for two bit times (data length = 6, 7, or 8 bits), or 1.5 bit times (data length = 5 bits).

SFR Definition 19.3. SBUF1: UART1 Data Buffer

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: 0xD3

Bits7–0: SBUF1[7:0]: Serial Data Buffer Bits 7–0 (MSB-LSB)
This SFR is used to both send data from the UART and to read received data from the UART1 receive FIFO.
Write: Writing a byte to SBUF1 initiates the transmission. When data is written to SBUF1, it first goes to the Transmit Holding Register, where it is held for serial transmission. When the transmit shift register is available, data is transferred into the shift register, and SBUF1 may be written again.
Read: Reading SBUF1 retrieves data from the receive FIFO. When read, the oldest byte in the receive FIFO is returned, and removed from the FIFO. Up to three bytes may be held in the FIFO. If there are additional bytes available in the FIFO, the RI1 bit will remain at logic '1', even after being cleared by software.

SFR Definition 19.4. SBCON1: UART1 Baud Rate Generator Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
Reserved	SB1RUN	Reserved	Reserved	Reserved	Reserved	SB1PS1	SB1PS0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xAC

Bit7: RESERVED: Read = 0b; Must write 0b.
Bit6: SB1RUN: Baud Rate Generator Enable.
0: Baud Rate Generator is disabled. UART1 will not function.
1: Baud Rate Generator is enabled.
Bits5–2: RESERVED: Read = 0000b; Must write 0000b.
Bits1–0: SB1PS[1:0]: Baud Rate Prescaler Select.
00: Prescaler = 12
01: Prescaler = 4
10: Prescaler = 48
11: Prescaler = 1

SFR Definition 19.5. SBRLH1: UART1 Baud Rate Generator High Byte

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: 0xB5

Bits7–0: SBRLH1[7:0]: High Byte of reload value for UART1 Baud Rate Generator.

SFR Definition 19.6. SBRLL1: UART1 Baud Rate Generator Low Byte

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: 0xB4

Bits7–0: SBRLL1[7:0]: Low Byte of reload value for UART1 Baud Rate Generator.

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.

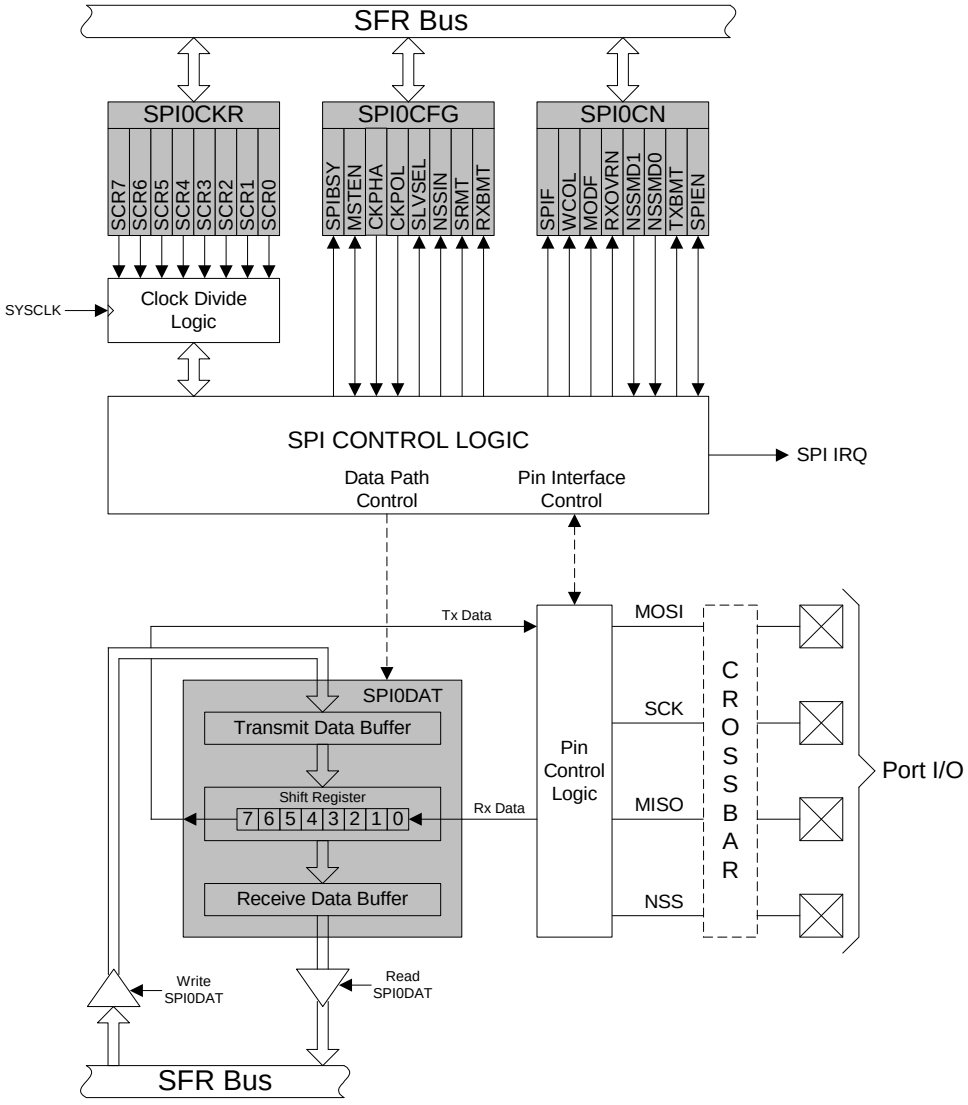


Figure 20.1. SPI Block Diagram

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 "**15. Port Input/Output**" on page **142** 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, SPI0CFG.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 (SPI0CFG.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.

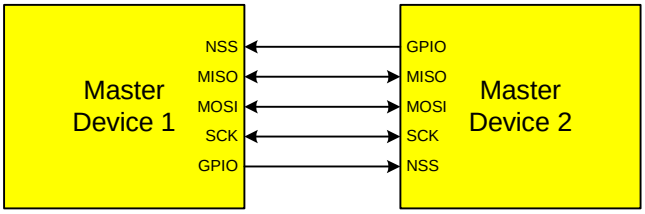


Figure 20.2. Multiple-Master Mode Connection Diagram

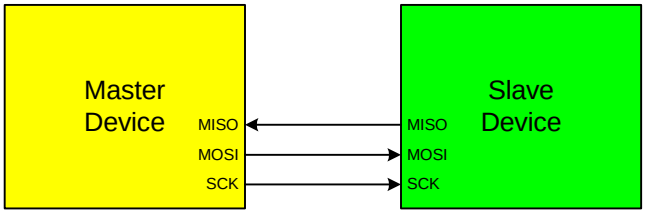


Figure 20.3. 3-Wire Single Master and Slave Mode Connection Diagram

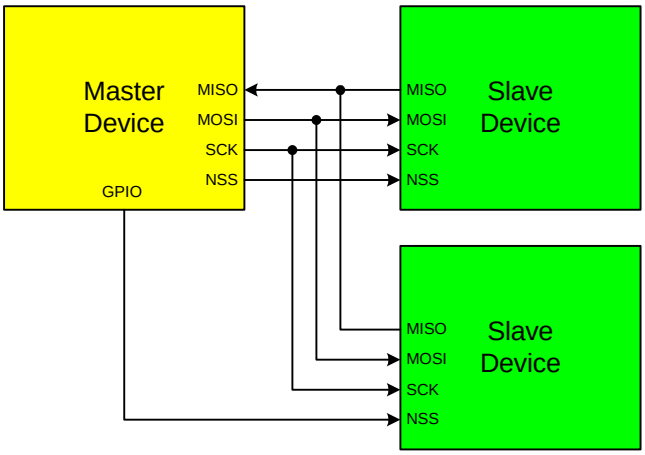


Figure 20.4. 4-Wire Single Master Mode and Slave Mode Connection Diagram

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:

Note that 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.

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.

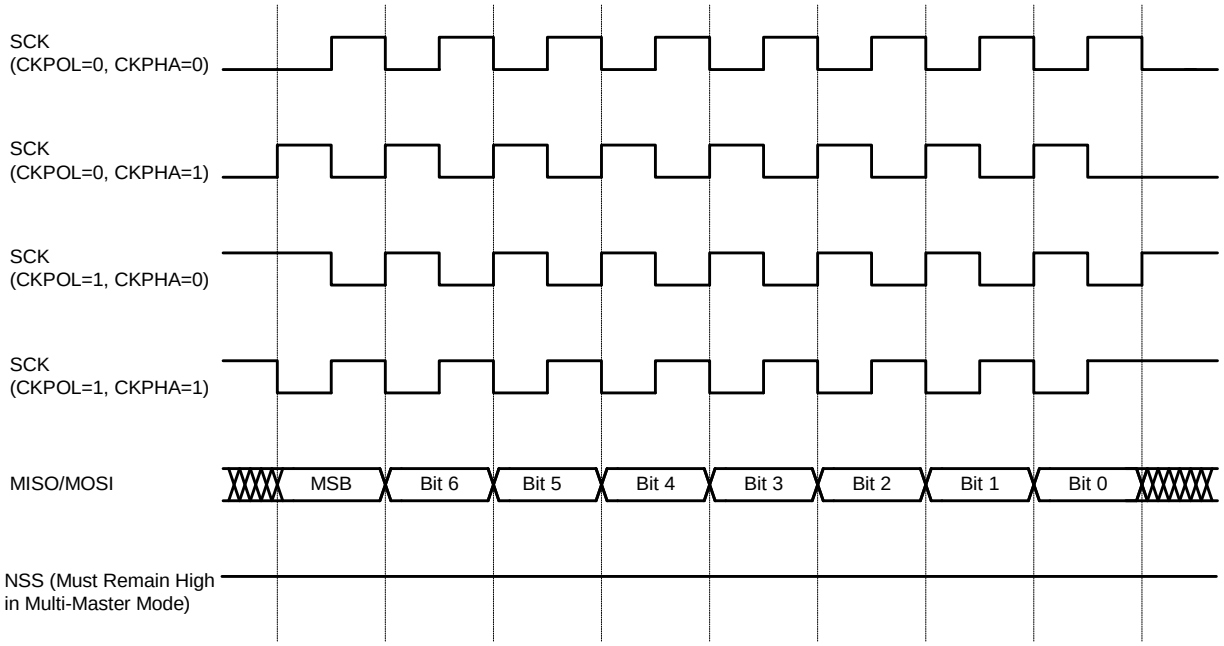


Figure 20.5. Master Mode Data/Clock Timing

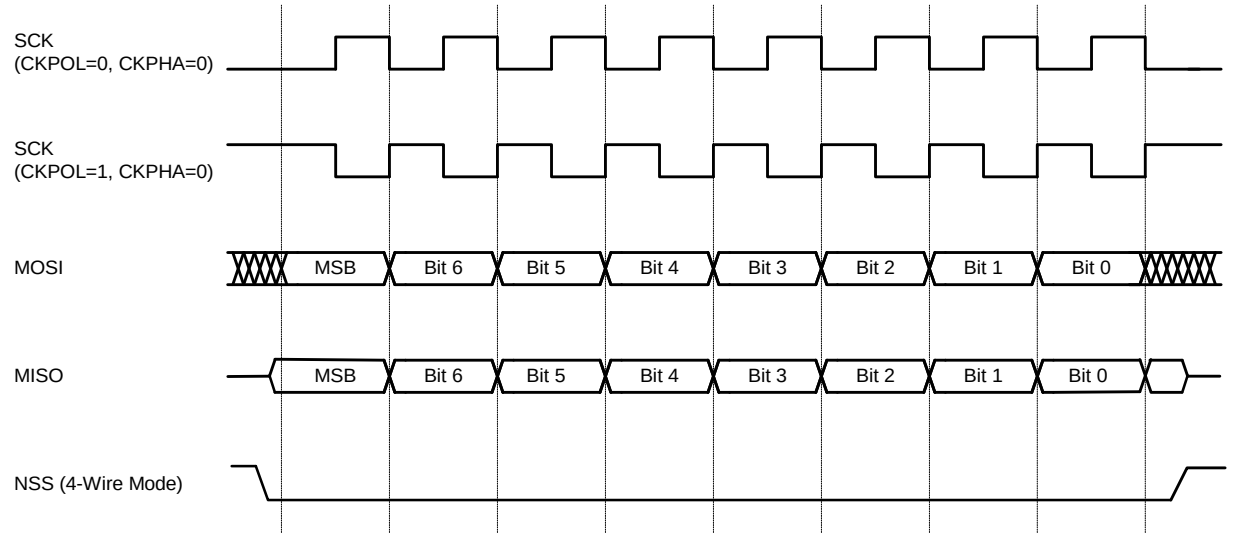


Figure 20.6. Slave Mode Data/Clock Timing (CKPHA = 0)

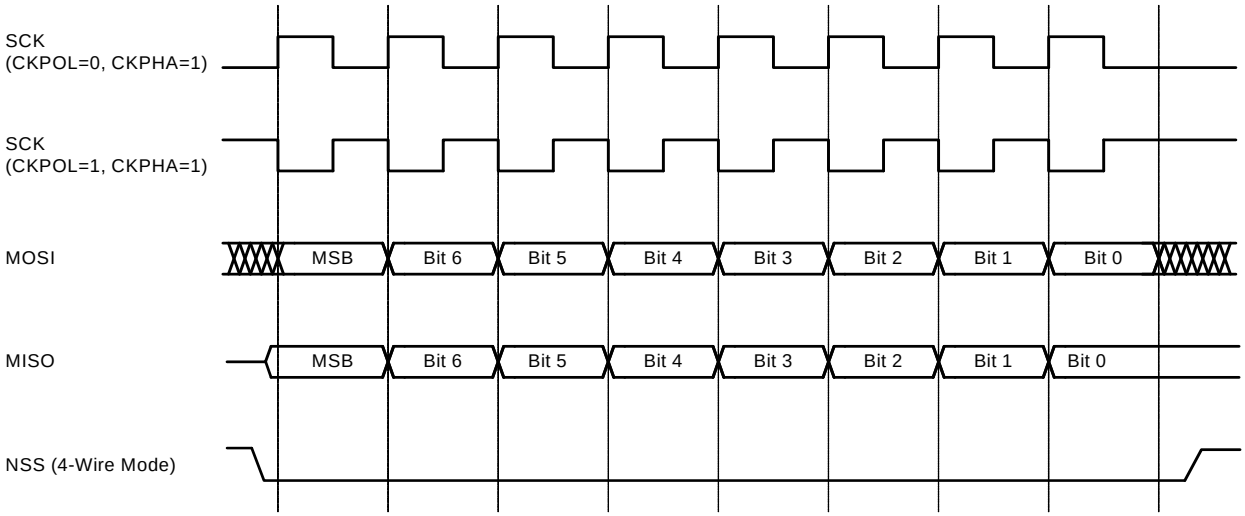


Figure 20.7. Slave Mode Data/Clock Timing (CKPHA = 1)

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

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	

SFR Address: 0xA1

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

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 Addressable

SFR Address: 0xF8

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 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. 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. SPI0 Master Mode Operation” on page 224** and **Section “20.3. SPI0 Slave Mode Operation” on page 226**).
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

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	

SFR Address: 0xA2

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,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

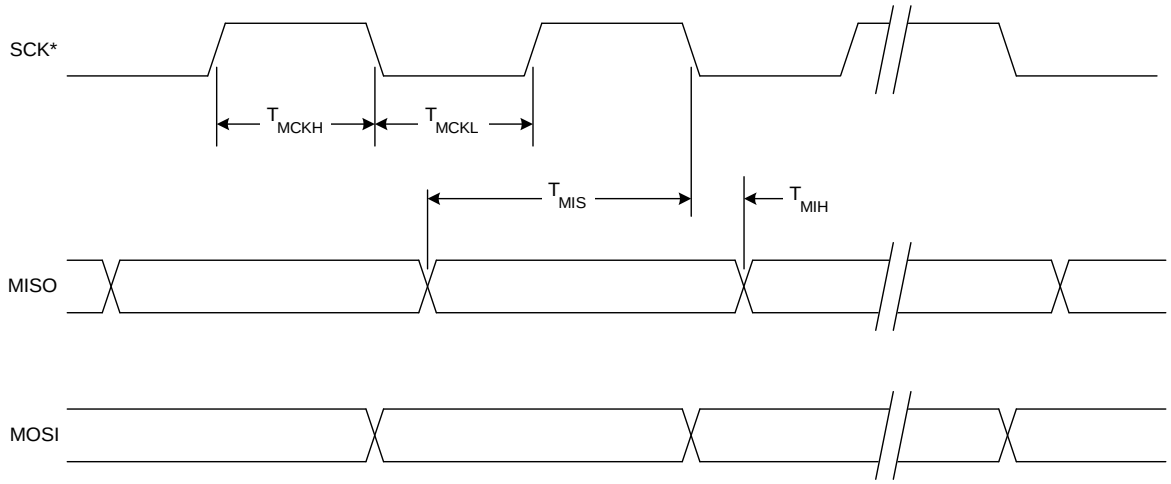
$$f_{SCK} = 200kHz$$

SFR Definition 20.4. SPI0DAT: SPI0 Data

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	

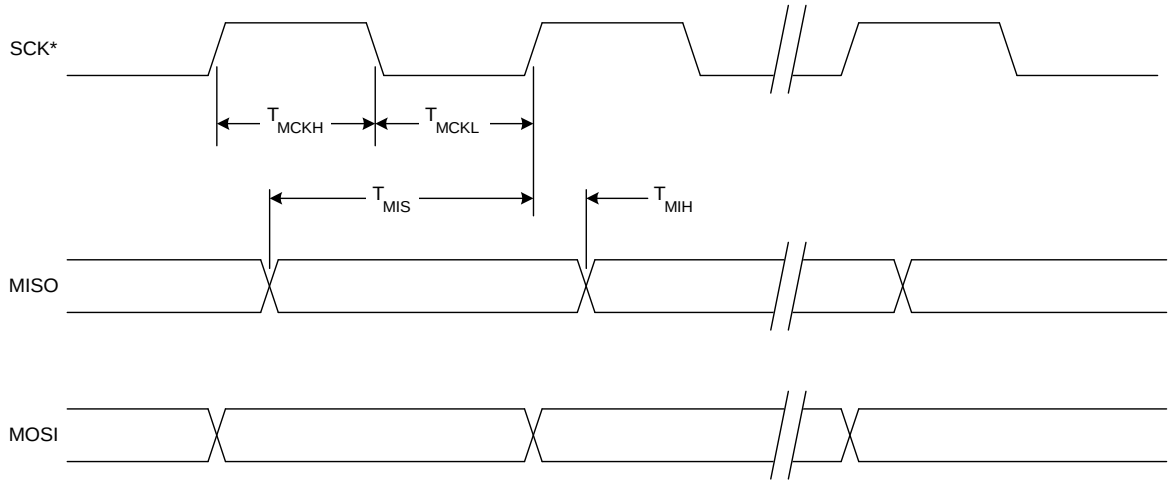
SFR Address: 0xA3

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.



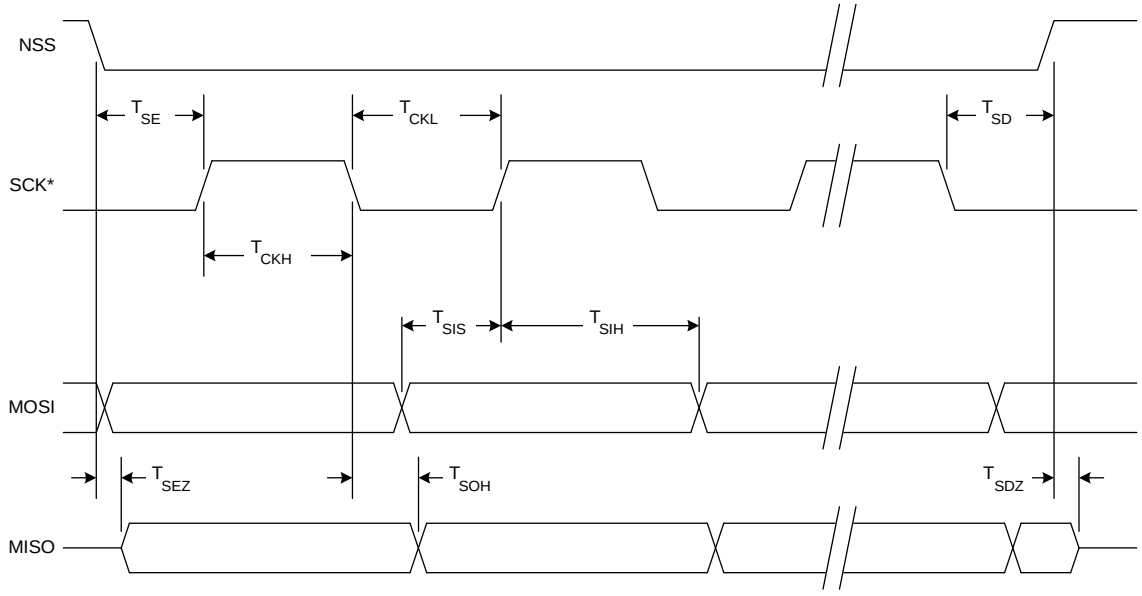
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 20.8. SPI Master Timing (CKPHA = 0)



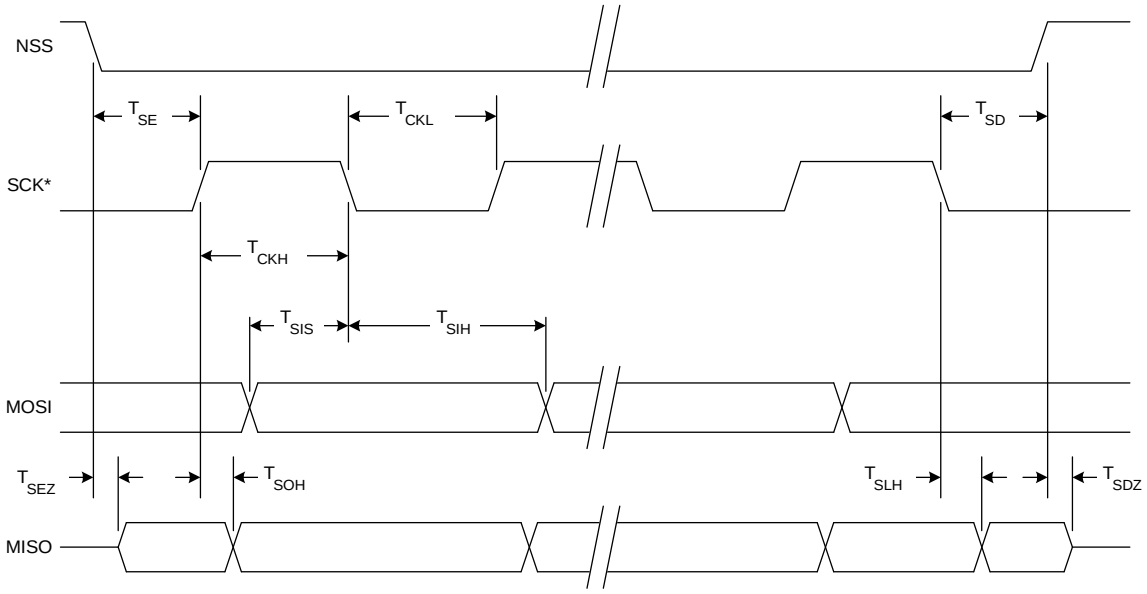
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

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, USB (frame measurements), Low-Frequency Oscillator (period measurements), 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.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

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's 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 “9.3.5. Interrupt Register Descriptions” on page 90**); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (**Section 9.3.5**). 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 "15.1. Priority Crossbar Decoder" on page 144** 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 INT01CF (see SFR Definition 9.13). Setting GATE0 to '1' allows the timer to be controlled by the external input signal INT0 (see **Section "9.3.5. Interrupt Register Descriptions" on page 90**), 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

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 INT01CF (see SFR Definition 9.13).

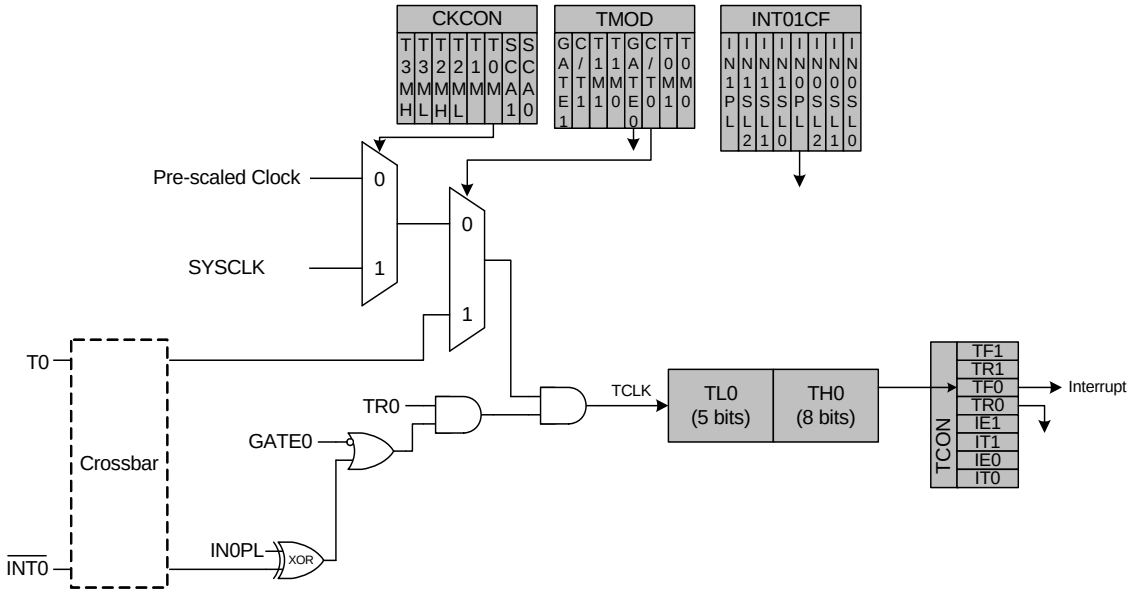


Figure 21.1. T0 Mode 0 Block Diagram

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 INT01CF (see **Section "9.3.2. External Interrupts" on page 88** for details on the external input signals INT0 and INT1).

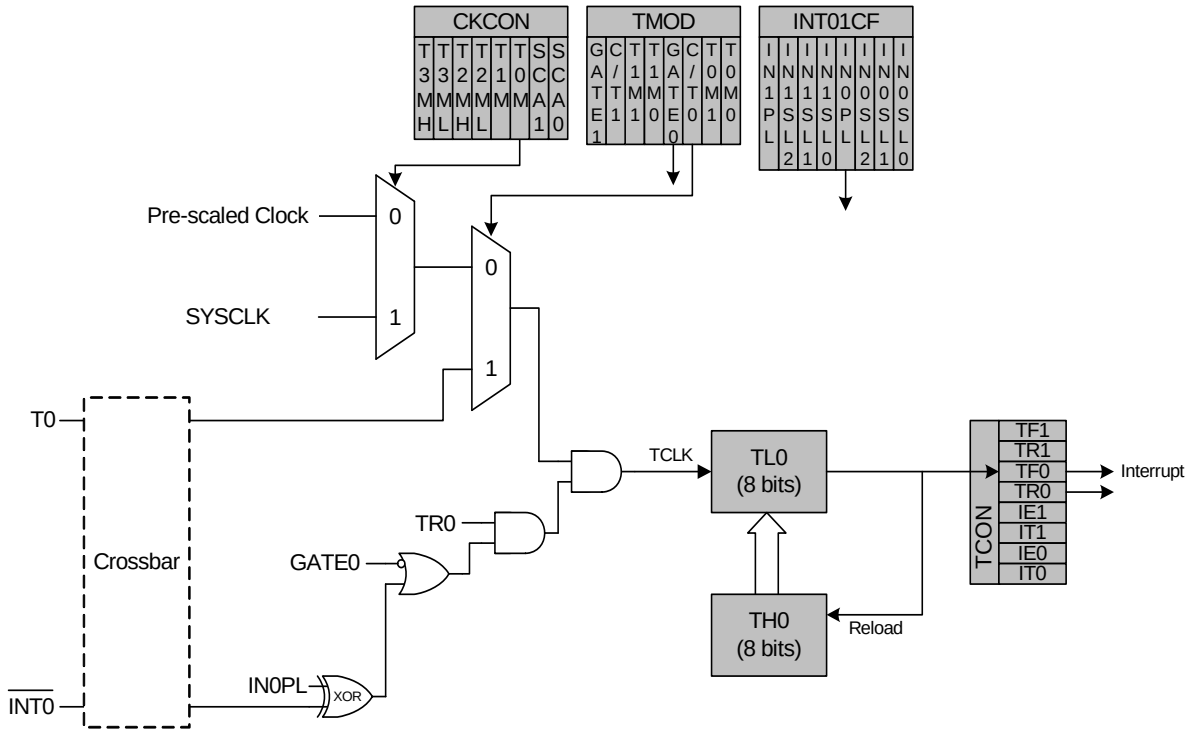


Figure 21.2. T0 Mode 2 Block Diagram

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.

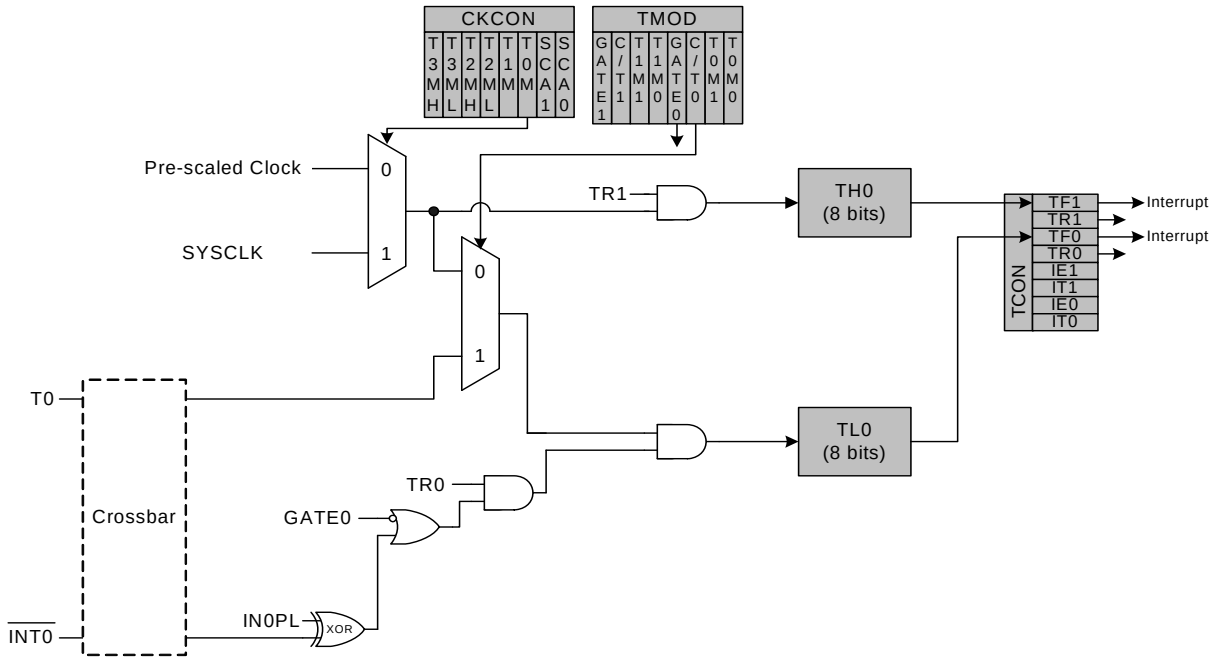


Figure 21.3. T0 Mode 3 Block Diagram

SFR Definition 21.1. TCON: Timer Control

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	SFR Address:
							(bit addressable)	0x88

Bit7: 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.

Bit6: TR1: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit5: 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.

Bit4: TR0: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit3: 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 $\overline{INT1}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured $\overline{INT1}$ interrupt will be edge or level sensitive. $\overline{INT1}$ is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 9.13).
0: $\overline{INT1}$ is level triggered.
1: $\overline{INT1}$ is edge triggered.

Bit1: 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 $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured $\overline{INT0}$ interrupt will be edge or level sensitive. $\overline{INT0}$ is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 9.13).
0: $\overline{INT0}$ is level triggered.
1: $\overline{INT0}$ is edge triggered.

SFR Definition 21.2. TMOD: Timer Mode

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	SFR Address: 0x89

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of $\overline{INT1}$ logic level.
1: Timer 1 enabled only when TR1 = 1 AND $\overline{INT1}$ is active as defined by bit IN1PL in register INT01CF (see SFR Definition 9.13).

Bit6: C/T1: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.3).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: 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

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of $\overline{INT0}$ logic level.
1: Timer 0 enabled only when TR0 = 1 AND $\overline{INT0}$ is active as defined by bit IN0PL in register INT01CF (see SFR Definition 9.13).

Bit2: C/T0: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.2).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: 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

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	SFR Address: 0x8E

Bit7: 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 Timer 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.

Bit6: 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.

Bit5: 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.

Bit4: 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.

Bit3: 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.

Bit2: 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.

Bits1–0: 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

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:
								0x8A

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

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:
								0x8B

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

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:
								0x8C

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

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:
								0x8D

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, (split) 8-bit auto-reload mode, USB Start-of-Frame (SOF) capture mode, or Low-Frequency Oscillator (LFO) Falling Edge capture mode. The Timer 2 operation mode is defined by the T2SPLIT (TMR2CN.3), T2CE (TMR2CN.4) bits, and T2CSS (TMR2CN.1) bits.

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 = ‘0’ and T2CE = ‘0’, 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, 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.

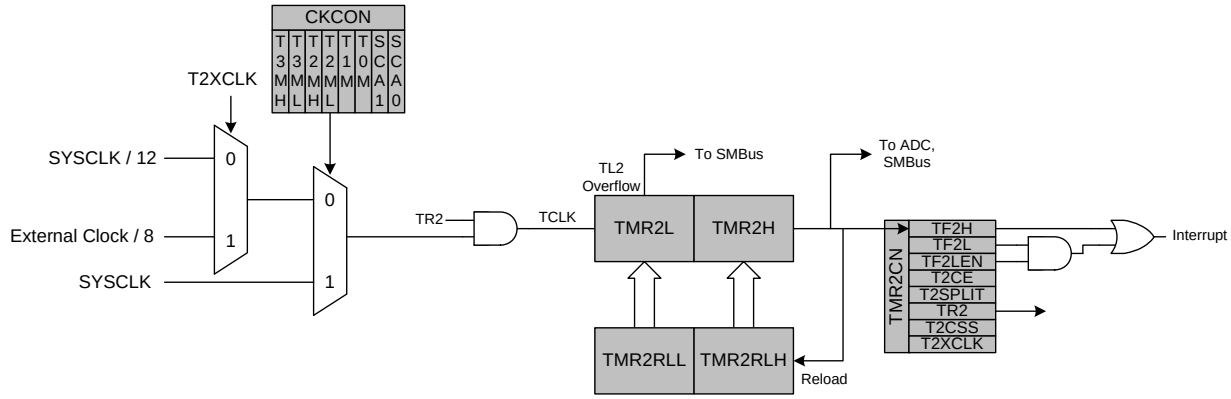


Figure 21.4. Timer 2 16-Bit Mode Block Diagram

21.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT = '1' and T2CE = '0', 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, 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.

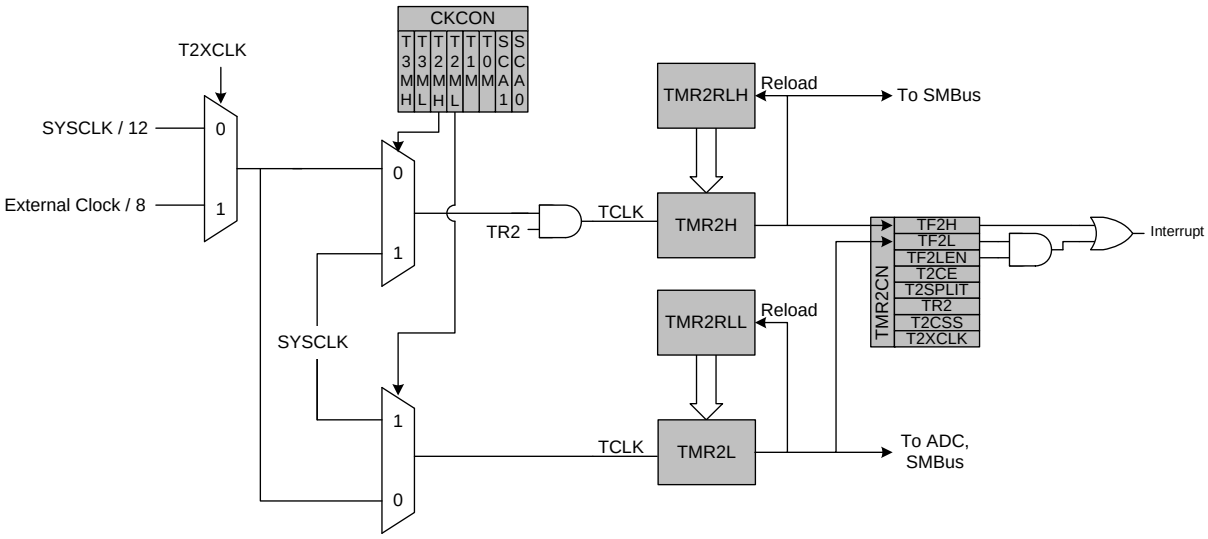


Figure 21.5. Timer 2 8-Bit Mode Block Diagram

21.2.3. Timer 2 Capture Modes: USB Start-of-Frame or LFO Falling Edge

When T2CE = ‘1’, Timer 2 will operate in one of two special capture modes. The capture event can be selected between a USB Start-of-Frame (SOF) capture, and a Low-Frequency Oscillator (LFO) Falling Edge capture, using the T2CSS bit. The USB SOF capture mode can be used to calibrate the system clock or external oscillator against the known USB host SOF clock. The LFO falling-edge capture mode can be used to calibrate the internal Low-Frequency Oscillator against the internal High-Frequency Oscillator or an external clock source. When T2SPLIT = ‘0’, Timer 2 counts up and overflows from 0xFFFF to 0x0000. Each time a capture event is received, the contents of the Timer 2 registers (TMR2H:TMR2L) are latched into the Timer 2 Reload registers (TMR2RLH:TMR2RLL). A Timer 2 interrupt is generated if enabled.

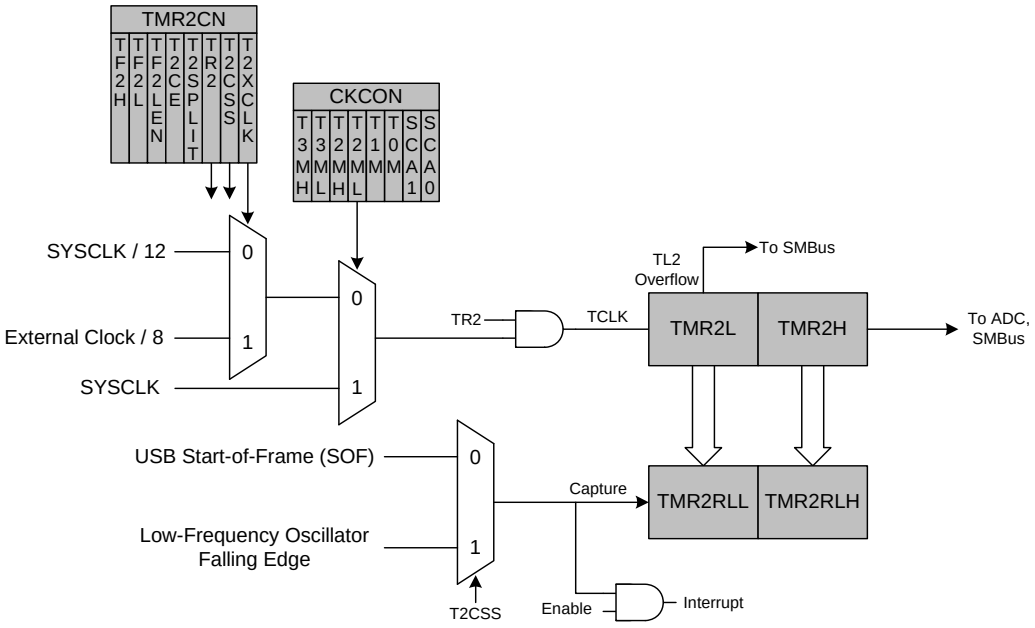


Figure 21.6. Timer 2 Capture Mode (T2SPLIT = ‘0’)

When T2SPLIT = ‘1’, the Timer 2 registers (TMR2H and TMR2L) act as two 8-bit counters. Each counter counts up independently and overflows from 0xFF to 0x00. Each time a capture event is received, the contents of the Timer 2 registers are latched into the Timer 2 Reload registers (TMR2RLH and TMR2RLL). A Timer 2 interrupt is generated if enabled.

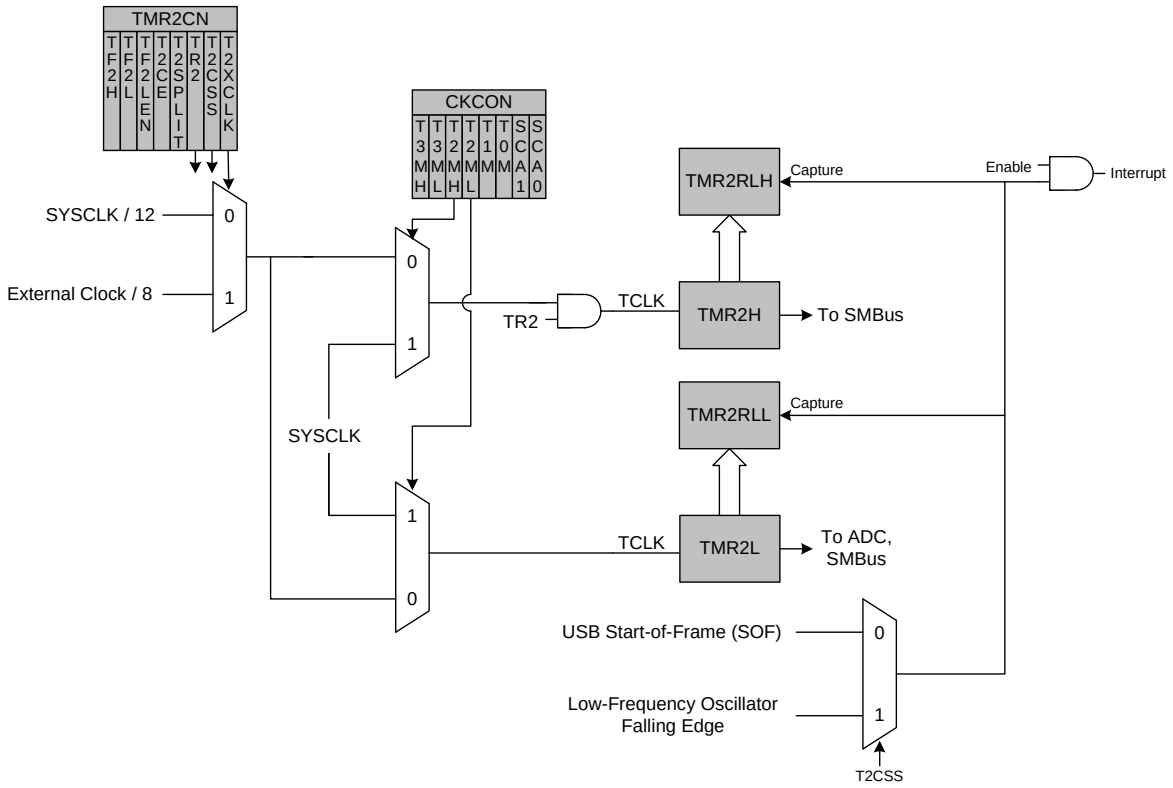


Figure 21.7. Timer 2 Capture Mode (T2SPLIT = ‘1’)

SFR Definition 21.8. TMR2CN: Timer 2 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF2H	TF2L	TF2LEN	T2CE	T2SPLIT	TR2	T2CSS	T2XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xC8

Bit7: 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.

Bit6: 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.

Bit5: 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.

Bit4: T2CE: Timer 2 Capture Enable
0: Capture function disabled.
1: Capture function enabled. The timer is in capture mode, with the capture event selected by bit T2CSS. Each time a capture event is received, the contents of the Timer 2 registers (TMR2H and TMR2L) are latched into the Timer 2 reload registers (TMR2RLH and TMR2RLH), and a Timer 2 interrupt is generated (if enabled).

Bit3: 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.

Bit2: 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.

Bit1: T2CSS: Timer 2 Capture Source Select.
This bit selects the source of a capture event when bit T2CE is set to ‘1’.
0: Capture source is USB SOF event.
1: Capture source is falling edge of Low-Frequency Oscillator.

Bit0: 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

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:
								0xCA

Bits 7–0: TMR2RLL: Timer 2 Reload Register Low Byte.
TMR2RLL holds the low byte of the reload value for Timer 2 when operating in auto-reload mode, or the captured value of the TMR2L register in capture mode.

SFR Definition 21.10. TMR2RLH: Timer 2 Reload Register High Byte

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:
								0xCB

Bits 7–0: TMR2RLH: Timer 2 Reload Register High Byte.
The TMR2RLH holds the high byte of the reload value for Timer 2 when operating in auto-reload mode, or the captured value of the TMR2H register in capture mode.

SFR Definition 21.11. TMR2L: Timer 2 Low Byte

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:
								0xCC

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

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:
								0xCD

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, (split) 8-bit auto-reload mode, USB Start-of-Frame (SOF) capture mode, or Low-Frequency Oscillator (LFO) Rising Edge capture mode. The Timer 3 operation mode is defined by the T3SPLIT (TMR3CN.3), T3CE (TMR3CN.4) bits, and T3CSS (TMR3CN.1) bits.

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 ‘0’ and T3CE = ‘0’, 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 TM3RLL) is loaded into the Timer 3 register as shown in Figure 21.4, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled, 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.

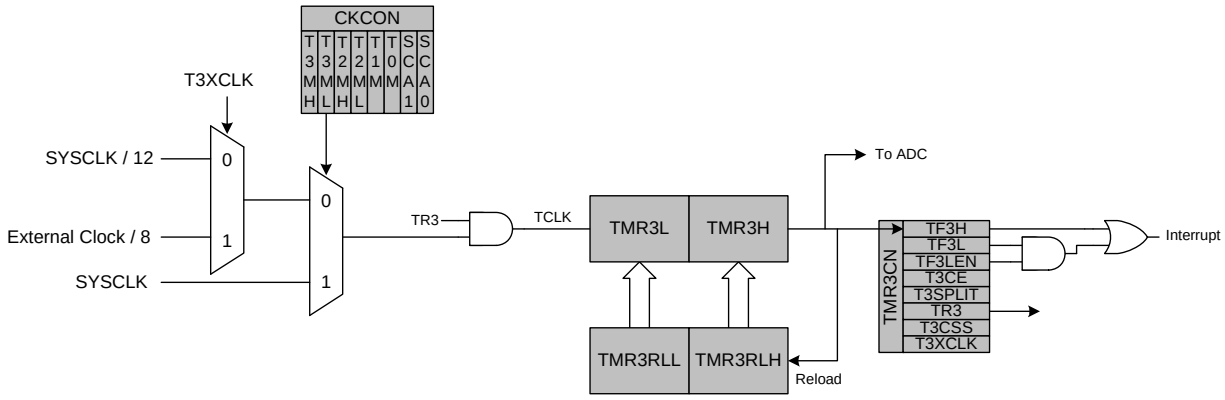


Figure 21.8. Timer 3 16-Bit Mode Block Diagram

21.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is ‘1’ and T3CE = ‘0’, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 21.5. 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.

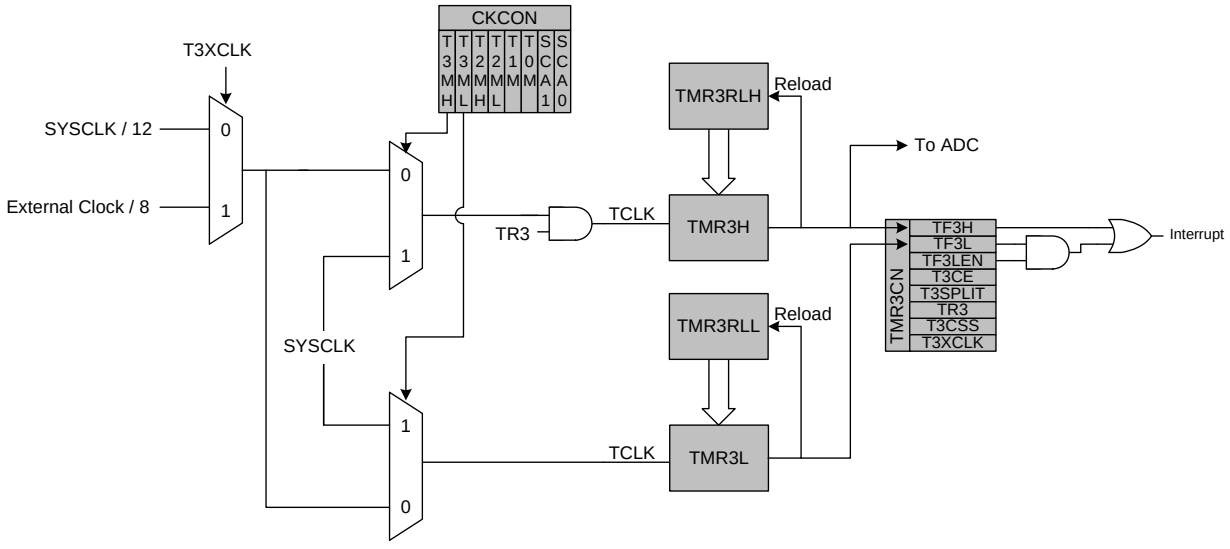


Figure 21.9. Timer 3 8-Bit Mode Block Diagram

21.3.3. USB Start-of-Frame Capture

When T3CE = ‘1’, Timer 3 will operate in one of two special capture modes. The capture event can be selected between a USB Start-of-Frame (SOF) capture, and a Low-Frequency Oscillator (LFO) Rising Edge capture, using the T3CSS bit. The USB SOF capture mode can be used to calibrate the system clock or external oscillator against the known USB host SOF clock. The LFO rising-edge capture mode can be used to calibrate the internal Low-Frequency Oscillator against the internal High-Frequency Oscillator or an external clock source. When T3SPLIT = ‘0’, Timer 3 counts up and overflows from 0xFFFF to 0x0000. Each time a capture event is received, the contents of the Timer 3 registers (TMR3H:TMR3L) are latched into the Timer 3 Reload registers (TMR3RLH:TMR3RLL). A Timer 3 interrupt is generated if enabled.

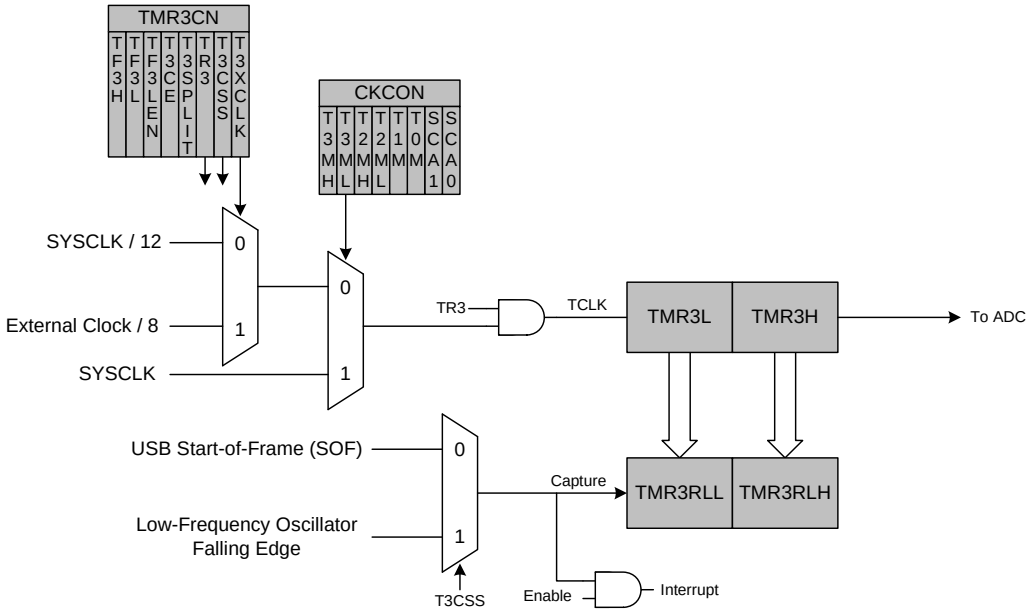


Figure 21.10. Timer 3 Capture Mode (T3SPLIT = ‘0’)

When T3SPLIT = ‘1’, the Timer 3 registers (TMR3H and TMR3L) act as two 8-bit counters. Each counter counts up independently and overflows from 0xFF to 0x00. Each time a capture event is received, the contents of the Timer 3 registers are latched into the Timer 3 Reload registers (TMR3RLH and TMR3RLL). A Timer 3 interrupt is generated if enabled.

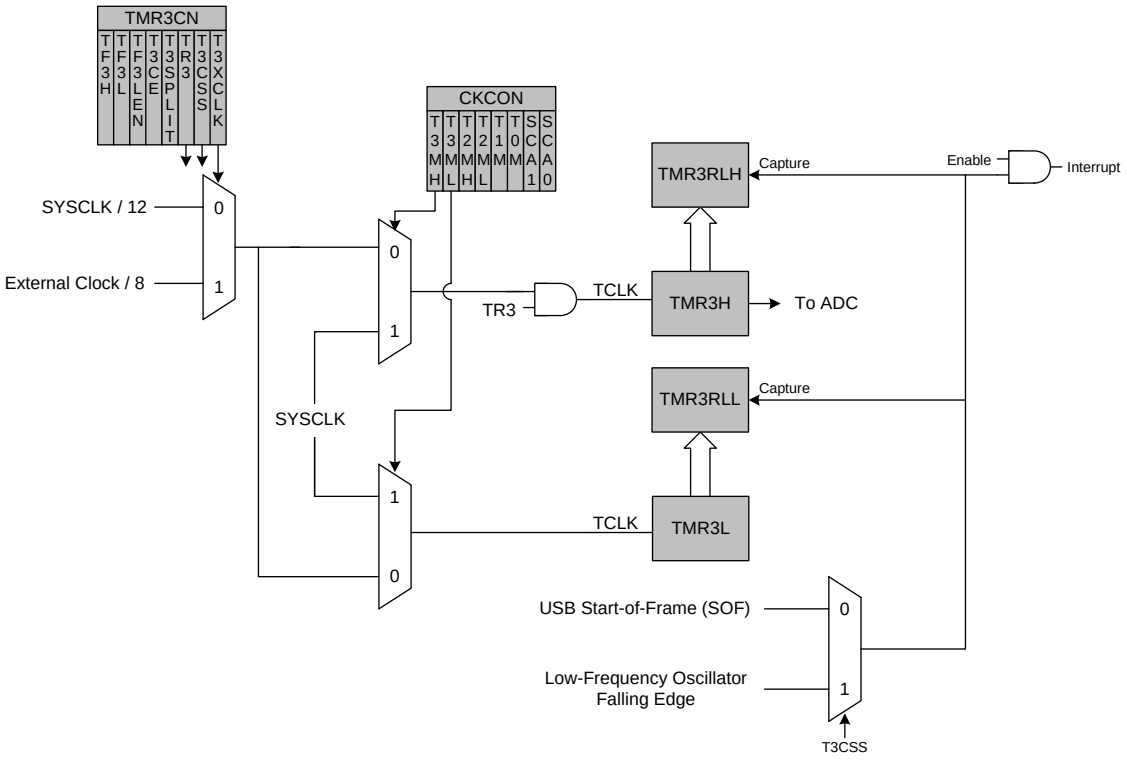


Figure 21.11. Timer 3 Capture Mode (T3SPLIT = ‘1’)

SFR Definition 21.13. TMR3CN: Timer 3 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF3H	TF3L	TF3LEN	T3CE	T3SPLIT	TR3	T3CSS	T3XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x91

Bit7: 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.

Bit6: 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.

Bit5: 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. This bit should be cleared when operating Timer 3 in 16-bit mode.
0: Timer 3 Low Byte interrupts disabled.
1: Timer 3 Low Byte interrupts enabled.

Bit4: T3CE: Timer 3 Capture Enable
0: Capture function disabled.
1: Capture function enabled. The timer is in capture mode, with the capture event selected by bit T3CSS. Each time a capture event is received, the contents of the Timer 3 registers (TMR3H and TMR3L) are latched into the Timer 3 reload registers (TMR3RLH and TMR3RLH), and a Timer 3 interrupt is generated (if enabled).

Bit3: 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.

Bit2: 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.

Bit1: T3CSS: Timer 3 Capture Source Select.
This bit selects the source of a capture event when bit T3CE is set to ‘1’.
0: Capture source is USB SOF event.
1: Capture source is rising edge of Low-Frequency Oscillator.

Bit0: 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

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:
								0x92

Bits 7–0: TMR3RLL: Timer 3 Reload Register Low Byte.
TMR3RLL holds the low byte of the reload value for Timer 3 when operating in auto-reload mode, or the captured value of the TMR3L register when operating in capture mode.

SFR Definition 21.15. TMR3RLH: Timer 3 Reload Register High Byte

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:
								0x93

Bits 7–0: TMR3RLH: Timer 3 Reload Register High Byte.
The TMR3RLH holds the high byte of the reload value for Timer 3 when operating in auto-reload mode, or the captured value of the TMR3H register when operating in capture mode.

SFR Definition 21.16. TMR3L: Timer 3 Low Byte

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:
								0x94

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

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:
								0x95

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 (PCA0)

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and five 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 "15.1. Priority Crossbar Decoder" on page 144** for details on configuring the Crossbar). The counter/timer is driven by a programmable timebase that can select between six sources: 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 input pin. 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 mode is described in **Section "22.2. Capture/Compare Modules" on page 257**). The external oscillator clock option is ideal for real-time clock (RTC) functionality, allowing the PCA to be clocked by a precision external oscillator while the internal oscillator drives the system clock. The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 22.1

Important Note: The PCA Module 4 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.

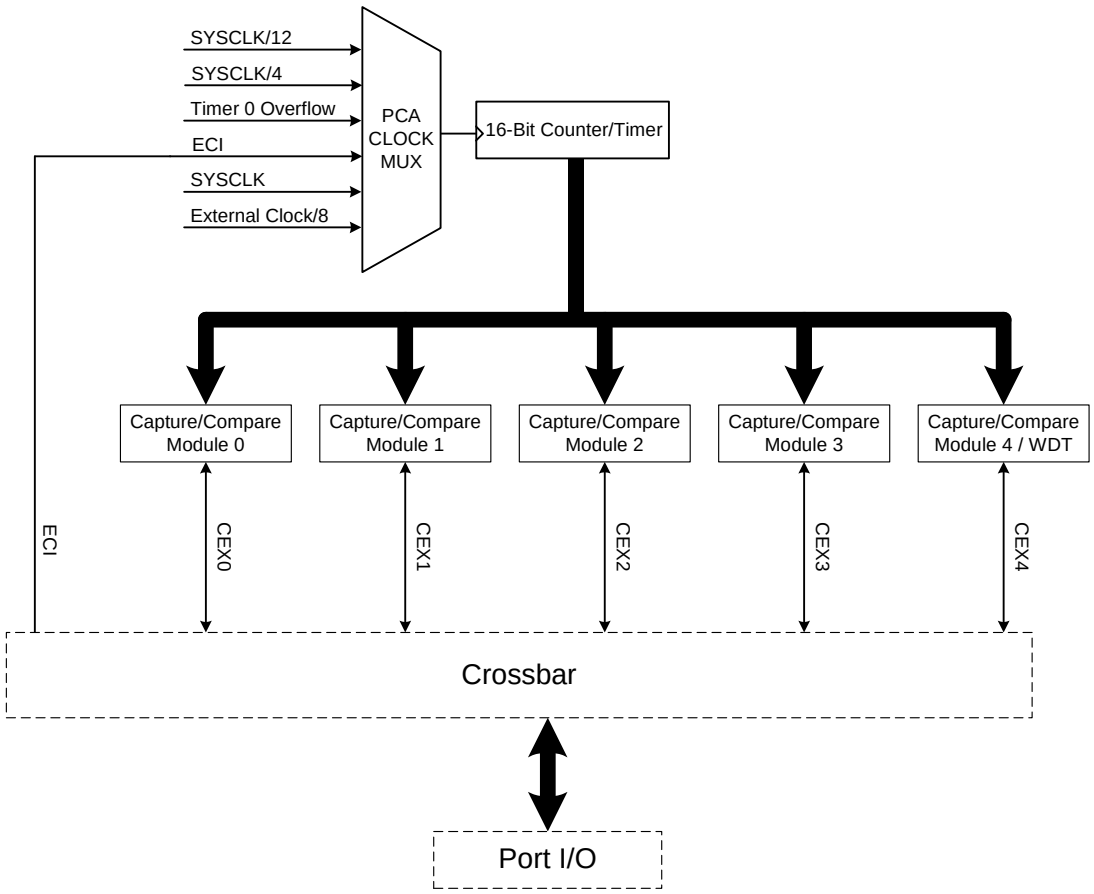


Figure 22.1. PCA Block Diagram

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*

***Note:** External oscillator source divided by 8 is synchronized with the system clock.

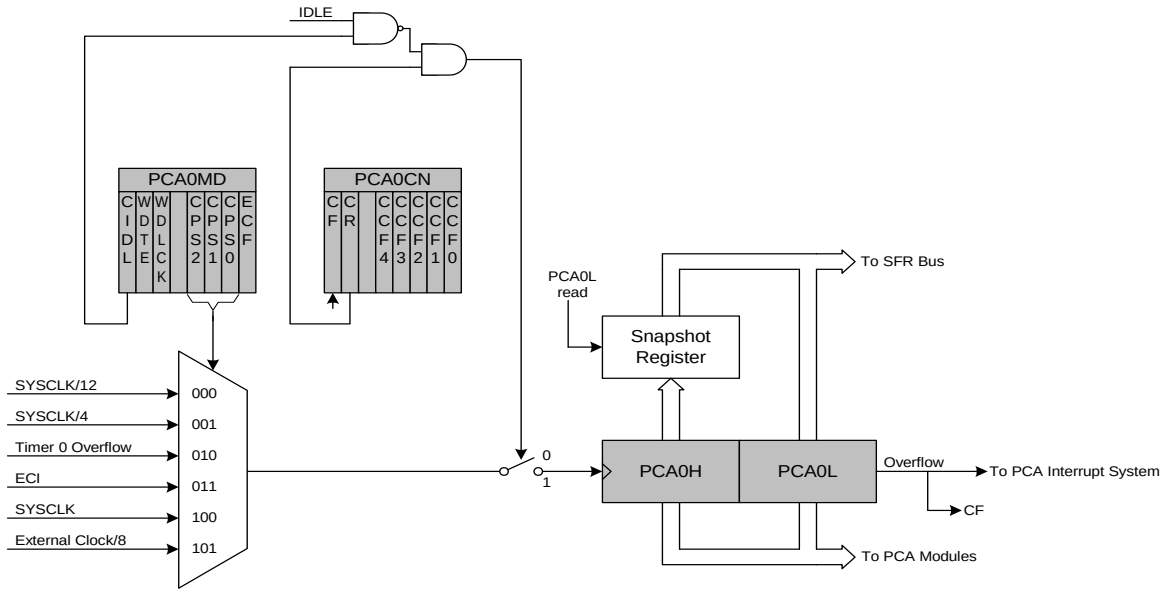


Figure 22.2. PCA Counter/Timer Block Diagram

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 PCA 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 and the EPCA0 bit to logic 1. See Figure 22.3 for details on the PCA interrupt configuration.

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	X	1	1	X	Frequency Output
0	1	0	0	X	0	1	X	8-Bit Pulse Width Modulator
1	1	0	0	X	0	1	X	16-Bit Pulse Width Modulator

X = Don't Care

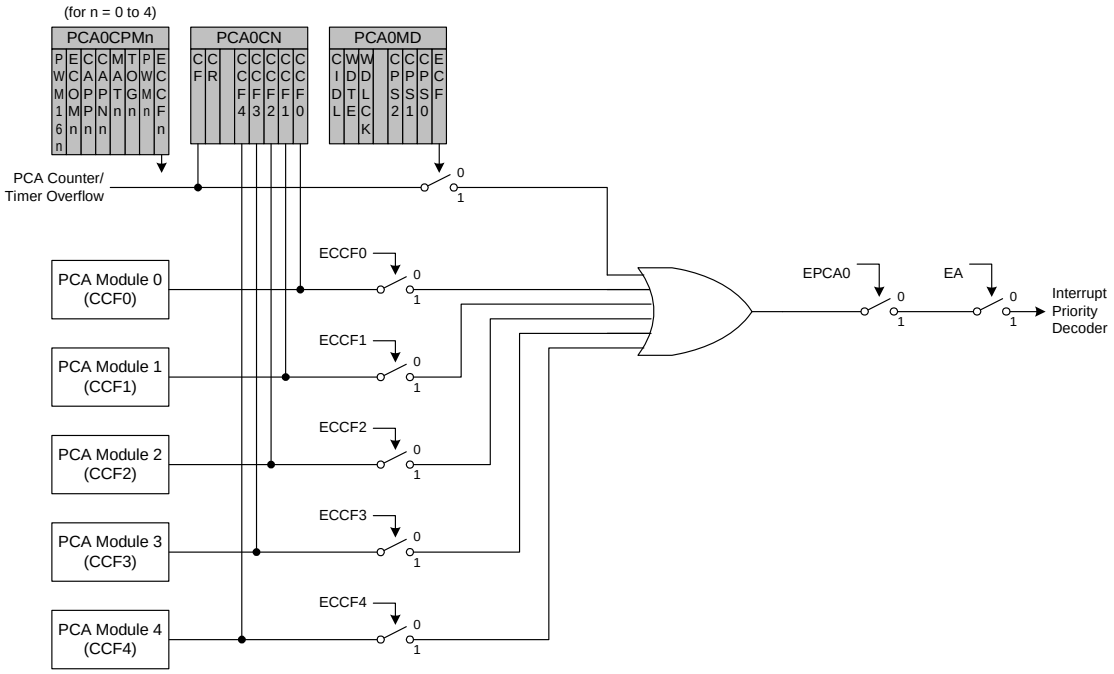


Figure 22.3. PCA Interrupt Block Diagram

22.2.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA 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.

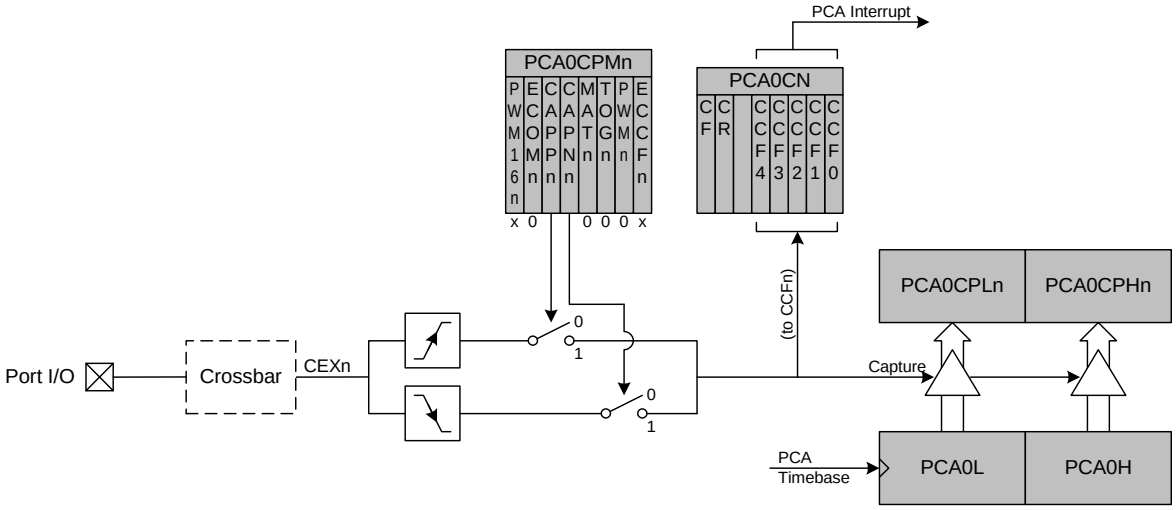


Figure 22.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain 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 PCA counter/timer value 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'.

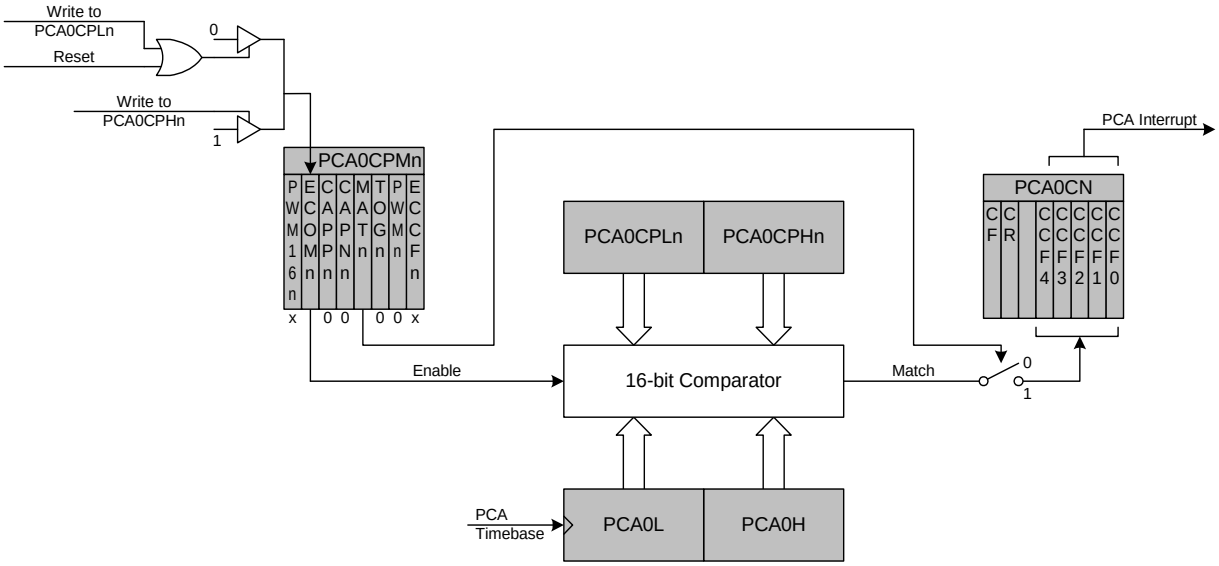


Figure 22.5. PCA Software Timer Mode Diagram

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'.

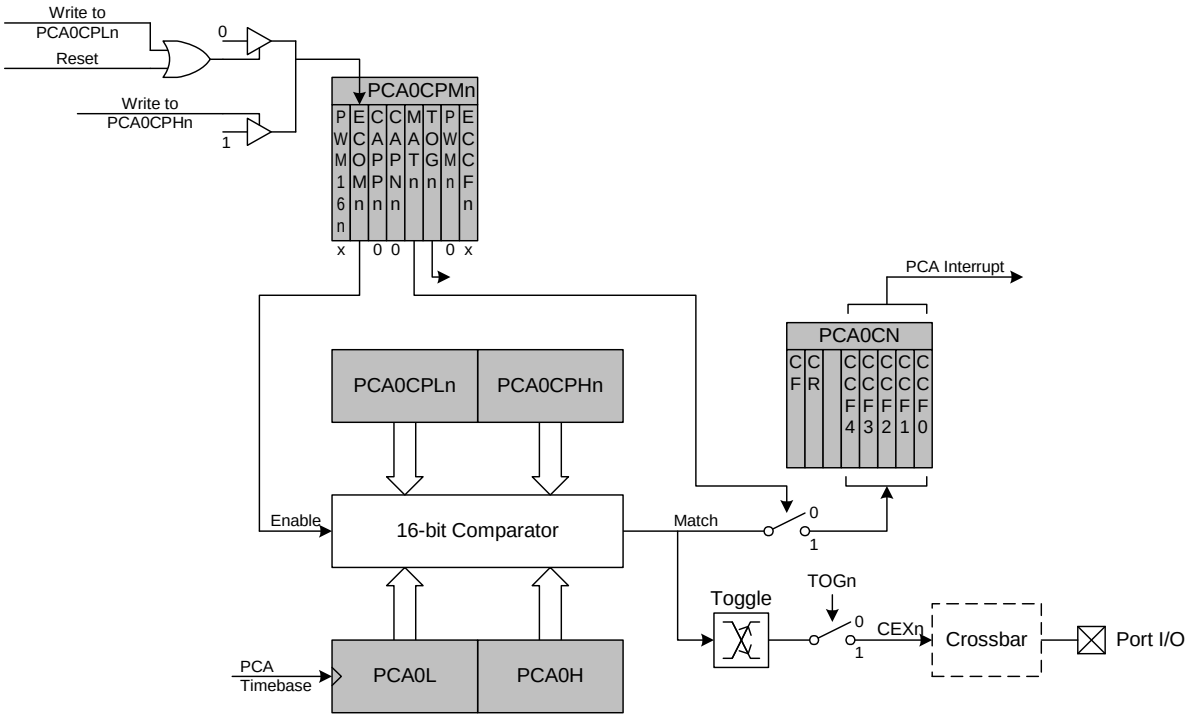


Figure 22.6. PCA High Speed Output Mode Diagram

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.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 22.1. Square Wave Frequency Output

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 PCA 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.

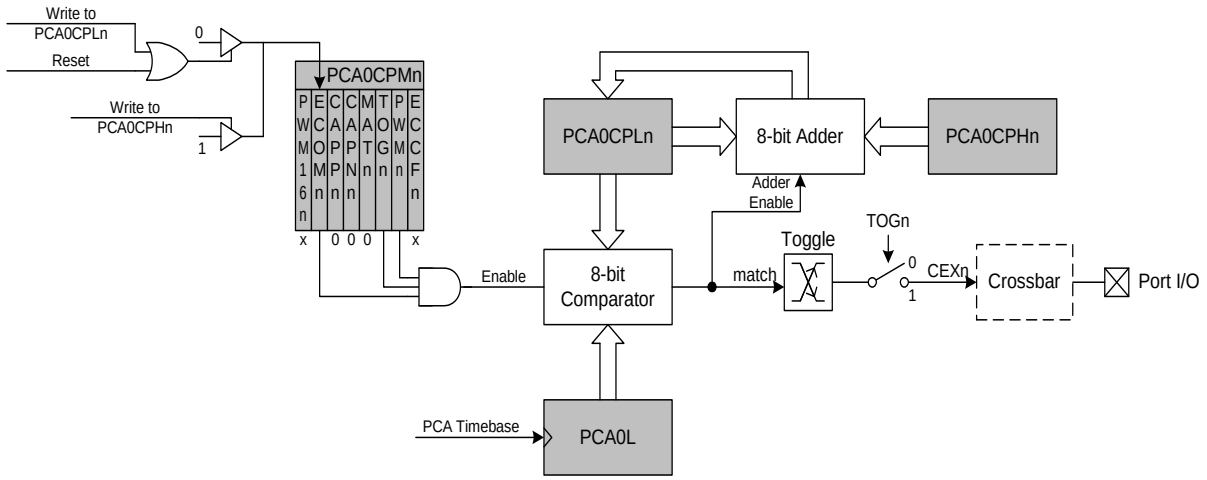


Figure 22.7. PCA Frequency Output Mode

22.2.5. 8-Bit Pulse Width Modulator Mode

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA 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 PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (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 module's capture/compare high byte (PCA0CPHn) 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.

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'.

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Equation 22.2. 8-Bit PWM Duty Cycle

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'.

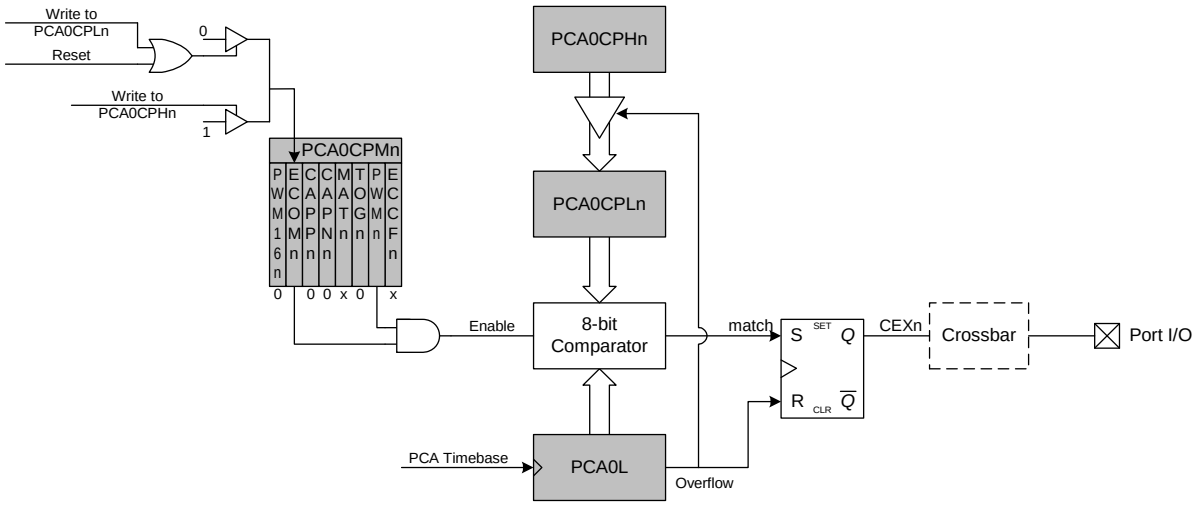


Figure 22.8. PCA 8-Bit PWM Mode Diagram

22.2.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA 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 PCA 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, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. 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'.

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 22.3. 16-Bit PWM Duty Cycle

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'.

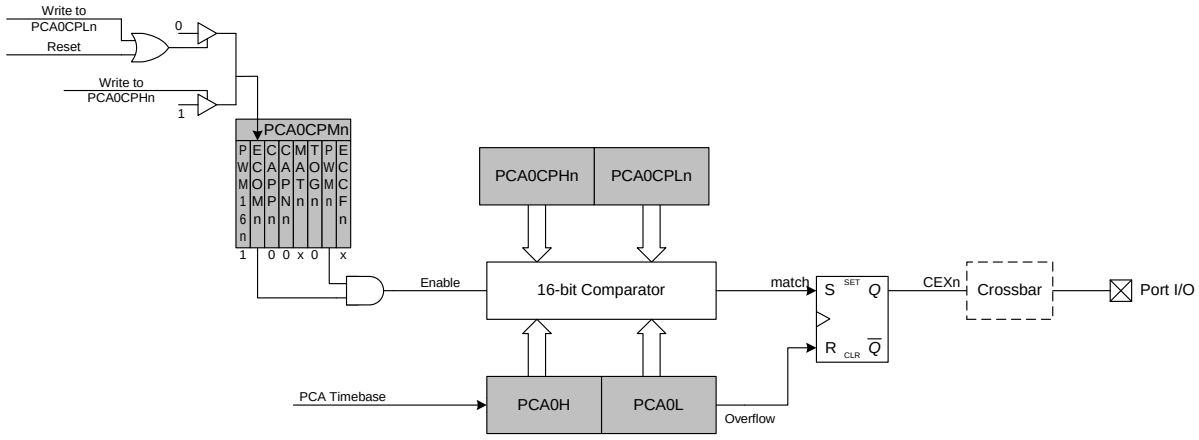


Figure 22.9. PCA 16-Bit PWM Mode

22.3. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 4. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH4) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE and/or WDLCK bits set to '1' in the PCA0MD register, Module 4 operates as a watchdog timer (WDT). The Module 4 high byte is compared to the PCA counter high byte; the Module 4 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 4 is forced into Watchdog Timer mode.
- Writes to the Module 4 mode register (PCA0CPM4) 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 PCA0CPH4 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 PCA0CPH4. Upon a PCA0CPH4 write, PCA0H plus the offset held in PCA0CPL4 is loaded into PCA0CPH4 (See Figure 22.10).

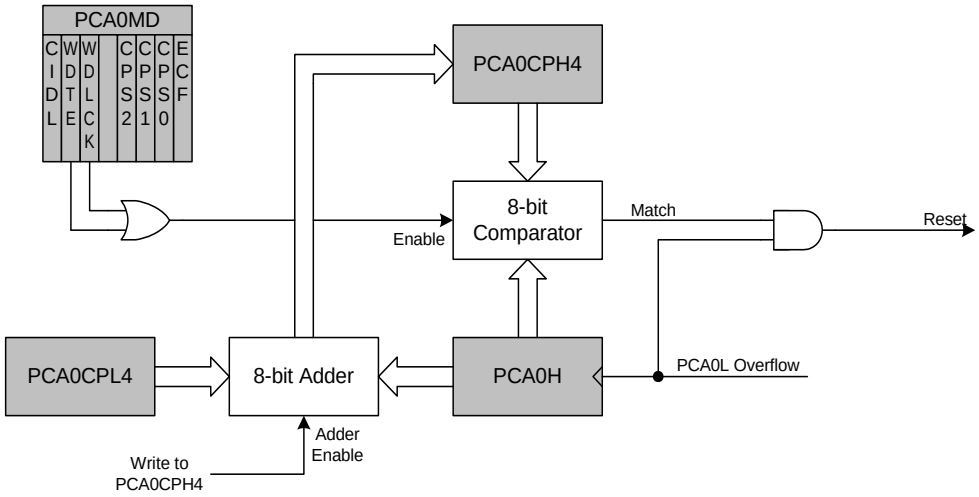


Figure 22.10. PCA Module 4 with Watchdog Timer Enabled

Note that the 8-bit offset held in PCA0CPH4 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.

$$Offset = (256 \times PCA0CPL4) + (256 - PCA0L)$$

Equation 22.4. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH4 and PCA0H. Software may force a WDT reset by writing a '1' to the CCF4 flag (PCA0CN.4) while the WDT is enabled.

22.3.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

1. Disable the WDT by writing a '0' to the WDTE bit.
2. Select the desired PCA clock source (with the CPS2-CPS0 bits).
3. Load PCA0CPL4 with the desired WDT update offset value.
4. Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
5. Enable the WDT by setting the WDTE bit to '1'.
6. (optional) Lock the WDT (prevent WDT disable until the next system reset) by setting the WDLCK bit to '1'.
7. Write a value to PCA0CPH4 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 PCA0CPL4 defaults to 0x00. Using Equation 22.4, this results in a WDT timeout interval of 256 PCA clocks. Table 22.3 lists some example timeout intervals for typical system clocks.

Table 22.3. Watchdog Timer Timeout Intervals[1]

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

Notes:

1. Assumes SYSCLK / 12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. System Clock reset frequency.

22.4. Register Descriptions for PCA

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

SFR Definition 22.1. PCA0CN: PCA Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xD8

Bit7: CF: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA 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 PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: CR: PCA Counter/Timer Run Control.
This bit enables/disables the PCA Counter/Timer.
0: PCA Counter/Timer disabled.
1: PCA Counter/Timer enabled.

Bit5: UNUSED. Read = 0b, Write = don't care.

Bit4: CCF4: PCA Module 4 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF4 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit3: CCF3: PCA Module 3 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF3 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit2: CCF2: PCA Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit1: CCF1: PCA Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit0: CCF0: PCA Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

SFR Definition 22.2. PCA0MD: PCA Mode

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	SFR Address:
								0xD9

Bit7: CIDL: PCA Counter/Timer Idle Control.
Specifies PCA behavior when CPU is in Idle Mode.
0: PCA continues to function normally while the system controller is in Idle Mode.
1: PCA operation is suspended while the system controller is in Idle Mode.

Bit6: WDTE: Watchdog Timer Enable
If this bit is set, PCA Module 4 is used as the watchdog timer.
0: Watchdog Timer disabled.
1: PCA Module 4 enabled as Watchdog Timer.

Bit5: WDLCK: Watchdog Timer Lock
This bit enables and locks the Watchdog Timer. When WDLCK is set to '1', the Watchdog Timer may not be disabled until the next system reset.
0: Watchdog Timer unlocked.
1: Watchdog Timer enabled and locked.

Bit4: UNUSED. Read = 0b, Write = don't care.

Bits3–1: CPS2–CPS0: PCA Counter/Timer Pulse Select.
These bits select the timebase source for the PCA 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*
1	1	0	Reserved
1	1	1	Reserved

***Note:** External oscillator source divided by 8 is synchronized with the system clock.

Bit0: ECF: PCA Counter/Timer Overflow Interrupt Enable.
This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt.
0: Disable the CF interrupt.
1: Enable a PCA 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: PCA Capture/Compare Mode

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	SFR Address:
								0xDA, 0xDB, 0xDC, 0xDD, 0xDE

PCA0CPMn Address: PCA0CPM0 = 0xDA (n = 0), PCA0CPM1 = 0xDB (n = 1), PCA0CPM2 = 0xDC (n = 2), PCA0CPM3 = 0xDD (n = 3), PCA0CPM4 = 0xDE (n = 4)

Bit7: 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.

Bit6: ECOMn: Comparator Function Enable.
This bit enables/disables the comparator function for PCA module n.
0: Disabled.
1: Enabled.

Bit5: CAPPn: Capture Positive Function Enable.
This bit enables/disables the positive edge capture for PCA module n.
0: Disabled.
1: Enabled.

Bit4: CAPNn: Capture Negative Function Enable.
This bit enables/disables the negative edge capture for PCA module n.
0: Disabled.
1: Enabled.

Bit3: MATn: Match Function Enable.
This bit enables/disables the match function for PCA module n. When enabled, matches of the PCA 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.

Bit2: TOGn: Toggle Function Enable.
This bit enables/disables the toggle function for PCA module n. When enabled, matches of the PCA 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.

Bit1: PWMn: Pulse Width Modulation Mode Enable.
This bit enables/disables the PWM function for PCA module n. When enabled, a pulse width modulated signal is output on the CEXn pin. 8-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit0: 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: PCA Counter/Timer Low Byte

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: 0xF9

Bits 7–0: PCA0L: PCA Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

SFR Definition 22.5. PCA0H: PCA Counter/Timer High Byte

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: 0xFA

Bits 7–0: PCA0H: PCA Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer.

SFR Definition 22.6. PCA0CPLn: PCA Capture Module Low Byte

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: 0xFB, 0xE9, 0xEB, 0xED, 0xFD

PCA0CPLn Address: PCA0CPL0 = 0xFB (n = 0), PCA0CPL1 = 0xE9 (n = 1),
PCA0CPL2 = 0xEB (n = 2), PCA0CPL3 = 0xED (n = 3),
PCA0CPL4 = 0xFD (n = 4)

Bits7–0: PCA0CPLn: PCA Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

SFR Definition 22.7. PCA0CPHn: PCA Capture Module High Byte

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: 0xFC, 0xEA, 0xEC,0xEE, 0xFE

PCA0CPHn Address: PCA0CPH0 = 0xFC (n = 0), PCA0CPH1 = 0xEA (n = 1),
PCA0CPH2 = 0xEC (n = 2), PCA0CPH3 = 0xEE (n = 3),
PCA0CPH4 = 0xFE (n = 4)

Bits7–0: PCA0CPHn: PCA Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

23. C2 Interface

C8051F34x devices include an on-chip Silicon Labs 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.

23.1. 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 23.1. 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
0x01	Selects the Revision ID register for Data Read instructions
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
0xAD	Selects the C2 Flash Programming Data register for Data Read/Write instructions

C2 Register Definition 23.2. DEVICEID: C2 Device ID

Reset Value: 00001111

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit device ID: 0x0F (C8051F340/1/2/3/4/5/6/7/8/9/A/B/C/D).

C2 Register Definition 23.3. REVID: C2 Revision ID

Reset Value: Variable

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit revision ID.

C2 Register Definition 23.4. FPCTL: C2 Flash Programming Control

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 23.5. FPDAT: C2 Flash Programming Data

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

23.2. 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 (P3.0) pins. Note that the C2D pin is shared on the 32-pin packages only (C8051F342/3/6/7/9/A/B). In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 23.1.

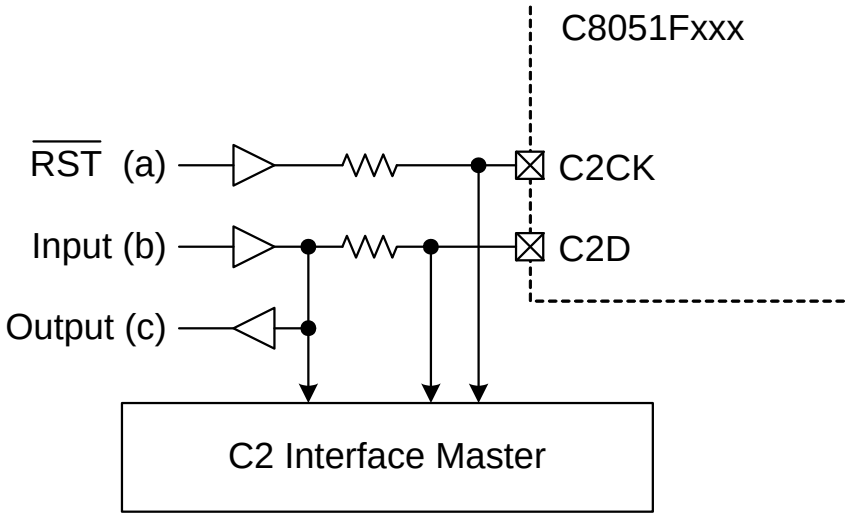


Figure 23.1. Typical C2 Pin Sharing

The configuration in Figure 23.1 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 0.5 to Revision 1.0

- Updated Table 3.1, “Global DC Electrical Characteristics,” on page 25.
- Updated Table 5.1, “ADC0 Electrical Characteristics,” on page 56.
- Various small text changes.
- Updated Table 8.1, “Voltage Regulator Electrical Specifications,” on page 69.
- Updated Flash security behavior.

Revision 1.0 to Revision 1.1

- Added two new part numbers C8051F348/9 and made associated changes.
- Corrected the entries "24 kHz" and "48 kHz" to "24 MHz" and "48 MHz" in the "Conditions" column of Table 3.1, "Global DC Electrical Characteristics," on page 38.
- Added note to configure external interrupt pin as open-drain with a “1” in the port latch in Section 9.3.2. "External Interrupts" on page 96.
- Various small text changes.
- Updated the figures in Section 15.1. "Priority Crossbar Decoder" and added a new figure to clarify crossbar capabilities.
- Corrected the description of the UNDRUN bit in USB Register Definition 16.19. "EINCSRL: USB0 IN Endpoint Control Low Byte" on page 198 to clarify that this bit works only in Isochronous Mode.
- Corrected the maximum SMBus speed from 1/10th to 1/20th of the system clock in Section 17. "SMBus" on page 205.
- Corrected the descriptions for the following states and the corresponding typical response options in Table 17.4. "SMBus Status Decoding" on page 221:
  - Slave Transmitter (Status Vector: 0101)
  - Slave Receiver (Status Vector: 0001)
- Corrected the bit location of MSTEN from SPI0CN.6 to SPI0CFG.6 in Section 20.2. "SPI0 Master Operation" on page 243.
- Corrected the description of the WCOL bit in SFR Definition 20.2. "SPI0CN: SPI0 Control" on page 249 to match the description in Section 20.4. "SPI0 Interrupt Sources" on page 245.
- Clarified the following parameters in Table 8.1, “Voltage Regulator Electrical Specifications,” on page 69:
  - VBUS Detection Input High and Low Voltages
  - Dropout Voltage
- Updated the package drawings with additional dimensions in Figure 4.2 and Table 4.2, “TQFP-48 Package Dimensions,” on page 32, and Figure 4.4 and Table 4.4, “LQFP-32 Package Dimensions,” on page 35.

Revision 1.1 to Revision 1.2

- Added two new part numbers C8051F34A/B and made associated changes.
- Corrected references to locations of T0M and T1M in the SFR definition of TMOD on page 240.
- Corrected instances of "8k" to "4k" in the SFR definition of EMI0CF on page 118.

Revision 1.2 to Revision 1.3

- Added QFN-32 package.

Revision 1.3 to Revision 1.4

- Added C8051F34C and C8051F34D devices.

NOTES:

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

Please visit the Silicon Labs Technical Support web page and register to submit a technical support request.