C8051F97x

Low Power Capacitive Sensing MCU with up to 32 kB of Flash

Low Power Consumption

- 200 µA/MHz in active mode (24.5 MHz clock)
- 2 µs wakeup time
- 55 nA sleep mode with brownout detector
- 280 nA sleep mode with LFO
- 600 nA sleep mode with external crystal

Capacitance Sense Interface

- Supports buttons, sliders, wheels, and proximity sensing
- Fast 40 µs per channel conversion time
- 16-bit resolution, up to 43 input channels
- Auto scan and wake-on-touch
- Auto-accumulate up to 64x samples

10-Bit Analog-to-Digital Converter

- Up to 43 external pin input channels, up to 300 ksps
- Internal VREF or external VREF supported

Clock Sources

- Internal oscillators: 24.5 MHz, ±2% accuracy supports UART operation; 20 MHz low power oscillator requires very little bias current.
- External oscillator: Crystal, RC, C, or CMOS Clock
- SmaRTClock oscillator: 32 kHz Crystal or internal LFO
- Can switch between clock sources on-the-fly; useful in implementing various power-saving modes

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

Unique Identifier

- 128-bit unique key for each device

High-Speed CIP-51 µC Core

- Efficient, pipelined instruction architecture
- Up to 25 MIPS throughput with 25 MHz clock
- Uses standard 8051 instruction set
- Expanded interrupt handler
- 1-cycle 16 x 16 MAC Engine
- 7-channel Direct Memory Access (DMA) module

Memory

- Up to 32 kB flash
- Flash is in-system programmable in 512-Byte sectors
- Up to 8 kB RAM

General-Purpose I/O

- Up to 43 pins with high sink current and programmable drive
- Crossbar-enabled

Timer/Counters and PWM

- 4 general purpose 16-bit timer/counters
- 16-bit Programmable Counter Array (PCA) with three channels of PWM, capture/compare, or frequency output capability, and watchdog timer

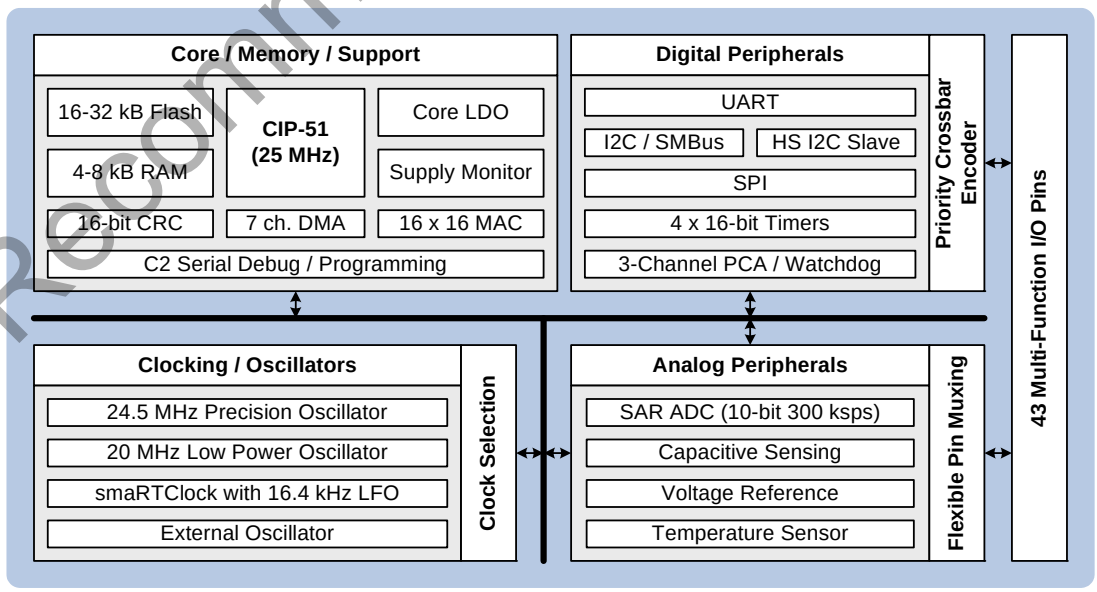
Supply Voltage: 1.8 to 3.6 V

- Built-in LDO regulator allows a high analog supply voltage and low digital core voltage
- 2 supply monitors (brownout detector) for sleep and active modes

Package Options

- 24-pin QFN (4x4 mm)
- 32-pin QFN (5x5 mm)
- 48-pin QFN (6x6 mm)

Temperature Ranges: –40 to +85 °C



1. Electrical Characteristics

Throughout the Electrical Characteristics chapter, "V_{DD}" refers to the Supply Voltage.

1.1. Electrical Characteristics

All electrical parameters in all tables are specified under the conditions listed in Table 1.1, unless stated otherwise.

Table 1.1. Recommended Operating Conditions

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Temperature Range	T_A		–40	25	85	°C
Supply Voltage	V_{DD}		1.8	3	3.6	V
***Note:** All minimum and maximum specifications are guaranteed and apply across the recommended operating conditions. Typical values apply at nominal supply voltages and an operating temperature of 25 °C unless otherwise noted.						

Table 1.2. Global Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	units
Supply Voltage (V_{DD})		1.8	3.0	3.6	V
RAM Data Retention Voltage[1]		—	1.4	—	V
SYSCLK (System Clock)[2]		0	—	25	MHz
T_{SYSH} (SYSCLK High Time)		18	—	—	ns
T_{SYSL} (SYSCLK Low Time)		18	—	—	ns
Specified Operating Temperature Range		–40	—	+85	°C

Notes:

1. Based on device characterization data; Not production tested.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. The values in this table are obtained with the CPU executing an "sjmp $" loop, which is the compiled form of a while(1) loop in C. See the power measurement code examples for more information.
4. Includes oscillator and regulator supply current.

Table 1.2. Global Electrical Characteristics (Continued)

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

Parameter	Conditions	Min	Typ	Max	units
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[3, 4]	V_{DD} = 1.8–3.6 V, Freq = 24.5 MHz (includes precision oscillator current)	—	5	6	mA
	V_{DD} = 1.8–3.6 V, Freq = 20 MHz (includes low power oscillator current)	—	4	—	mA
	V_{DD} = 1.8 V, Freq = 1 MHz (includes external CMOS oscillator / GPIO current)	—	420	—	µA
	V_{DD} = 3.6 V, Freq = 1 MHz (includes external CMOS oscillator / GPIO current)	—	440	—	µA
	V_{DD} = 1.8–3.6 V, Freq = 32.768 kHz (includes RTC current)	—	95	—	µA
I_{DD} Frequency Sensitivity[1, 3,	V_{DD} = 1.8-3.6 V, T = 25 °C, Freq < 14 MHz (Flash oneshot active)	—	230	—	µA/MHz
	V_{DD} = 1.8-3.6 V, T = 25 °C, Freq > 14 MHz (Flash oneshot bypassed)	—	130	—	µA/MHz

Notes:

1. Based on device characterization data; Not production tested.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. The values in this table are obtained with the CPU executing an “sjmp $” loop, which is the compiled form of a while(1) loop in C. See the power measurement code examples for more information.
4. Includes oscillator and regulator supply current.

Table 1.2. Global Electrical Characteristics (Continued)

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

Parameter	Conditions	Min	Typ	Max	units
Digital Supply Current—CPU Inactive (Idle Mode, not fetching instructions from Flash)					
I_{DD}[4]	V_{DD} = 1.8–3.6 V, Freq = 24.5 MHz (includes precision oscillator current)	—	3.3	4.5	mA
	V_{DD} = 1.8–3.6 V, Freq = 20 MHz (includes low power oscillator current)	—	2.5	—	mA
	V_{DD} = 1.8–3.6 V, Freq = 32.768 kHz (includes RTC current)	—	90	—	µA
I_{DD} Frequency Sensitivity[1]	V_{DD} = 1.8-3.6 V, T = 25 °C	—	110	—	µA/MHz

Notes:

1. Based on device characterization data; Not production tested.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. The values in this table are obtained with the CPU executing an “sjmp $” loop, which is the compiled form of a while(1) loop in C. See the power measurement code examples for more information.
4. Includes oscillator and regulator supply current.

Table 1.2. Global Electrical Characteristics (Continued)

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

Parameter	Conditions	Min	Typ	Max	units
Digital Supply Current—Suspend and Sleep Mode					
Digital Supply Current (Suspend Mode)	V_{DD} = 1.8–3.6 V	—	80	—	µA
Digital Supply Current (Sleep Mode, RTC 32 kHz Crystal Running, includes RTC and VDDMON)	V_{DD} = 1.8 V, T = 25 °C	—	0.6	—	µA
	V_{DD} = 3.3 V, T = 25 °C	—	0.7	—	µA
	V_{DD} = 3.6 V, T = 25 °C	—	0.8	—	µA
	V_{DD} = 1.8 V, T = 85 °C	—	1	—	µA
	V_{DD} = 3.3 V, T = 85 °C	—	1.3	—	µA
	V_{DD} = 3.6 V, T = 85 °C	—	1.5	—	µA
Digital Supply Current (Sleep Mode, RTC Int LFO running, includes RTC and VDDMON)	V_{DD} = 1.8 V, T = 25 °C	—	0.28	—	µA
Digital Supply Current (Sleep Mode, includes VDDMON)	V_{DD} = 1.8 V, T = 25 °C	—	0.05	—	µA
	V_{DD} = 3.3 V, T = 25 °C	—	0.06	—	µA
	V_{DD} = 3.6 V, T = 25 °C	—	0.11	—	µA
	V_{DD} = 1.8 V, T = 85 °C	—	0.8	—	µA
	V_{DD} = 3.3 V, T = 85 °C	—	0.9	—	µA
	V_{DD} = 3.6 V, T = 85 °C	—	1.0	—	µA

Notes:

1. Based on device characterization data; Not production tested.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. The values in this table are obtained with the CPU executing an “sjmp $” loop, which is the compiled form of a while(1) loop in C. See the power measurement code examples for more information.
4. Includes oscillator and regulator supply current.

Table 1.3. Port I/O DC Electrical Characteristics

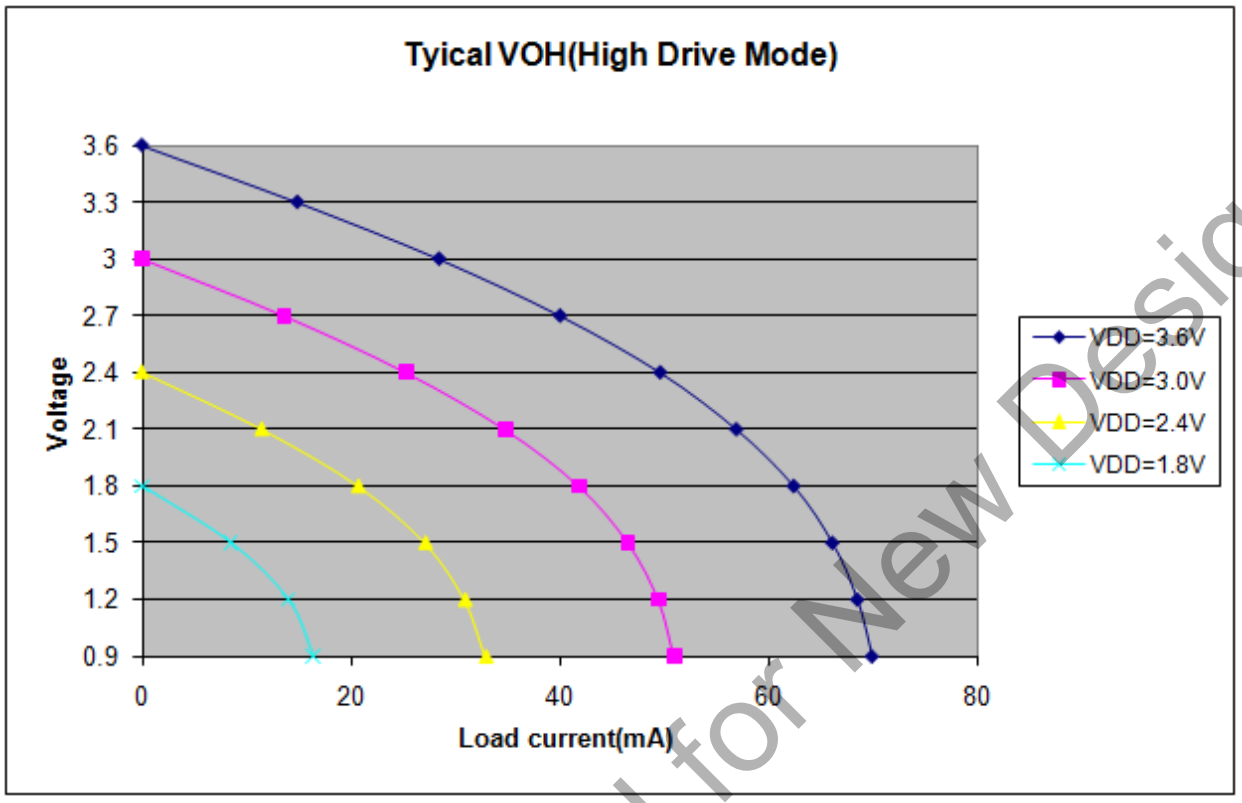
V_{DD} = 1.8 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Output High Voltage (High Drive Strength, PnDRV.n = 1)	I_{OH} = –3 mA, Port I/O push-pull	V_{DD} – 0.7	—	—	V
	I_{OH} = –10 µA	V_{DD} – 0.1	—	—	V
	I_{OH} = –10 mA	—	see chart	—	
Output High Voltage (Low Drive Strength, PnDRV.n = 0)	I_{OH} = –1 mA	V_{DD} – 0.7	—	—	V
	I_{OH} = –10 µA	V_{DD} – 0.1	—	—	V
	I_{OH} = –3 mA	—	see chart	—	V
Output Low Voltage (High Drive Strength, PnDRV.n = 1)	I_{OL} = 8.5 mA	—	—	0.7	V
	I_{OL} = 25 mA	—	see chart	—	V
Output Low Voltage (Low Drive Strength, PnDRV.n = 0)	I_{OL} = 1.4 mA	—	—	0.7	V
	I_{OL} = 4 mA	—	see chart	—	V
Input High Voltage	V_{DD} = 2.0 to 3.6 V	V_{DD} – 0.6	—	—	V
Input Low Voltage	V_{DD} = 2.0 to 3.6 V	—	—	0.6	V
Input Leakage Current	Weak Pull–up Off	—	—	0.5	µA
	Weak Pull-up On, V_{IN}=0 V, V_{DD} = 1.8 V	—	4	—	µA
	Weak Pull-up On, V_{IN} = 0 V, V_{DD} = 3.6 V	—	23	—	µA

Table 1.4. I^2C Slave Electrical Characteristics

Parameter	Condition	Min	Typ	Max	units
V_{DD} Range	Required to meet I^2C spec	1.8	—	3.6	V
Internal I^2C pull-ups	Required to meet I^2C spec	—	6	—	kΩ

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.



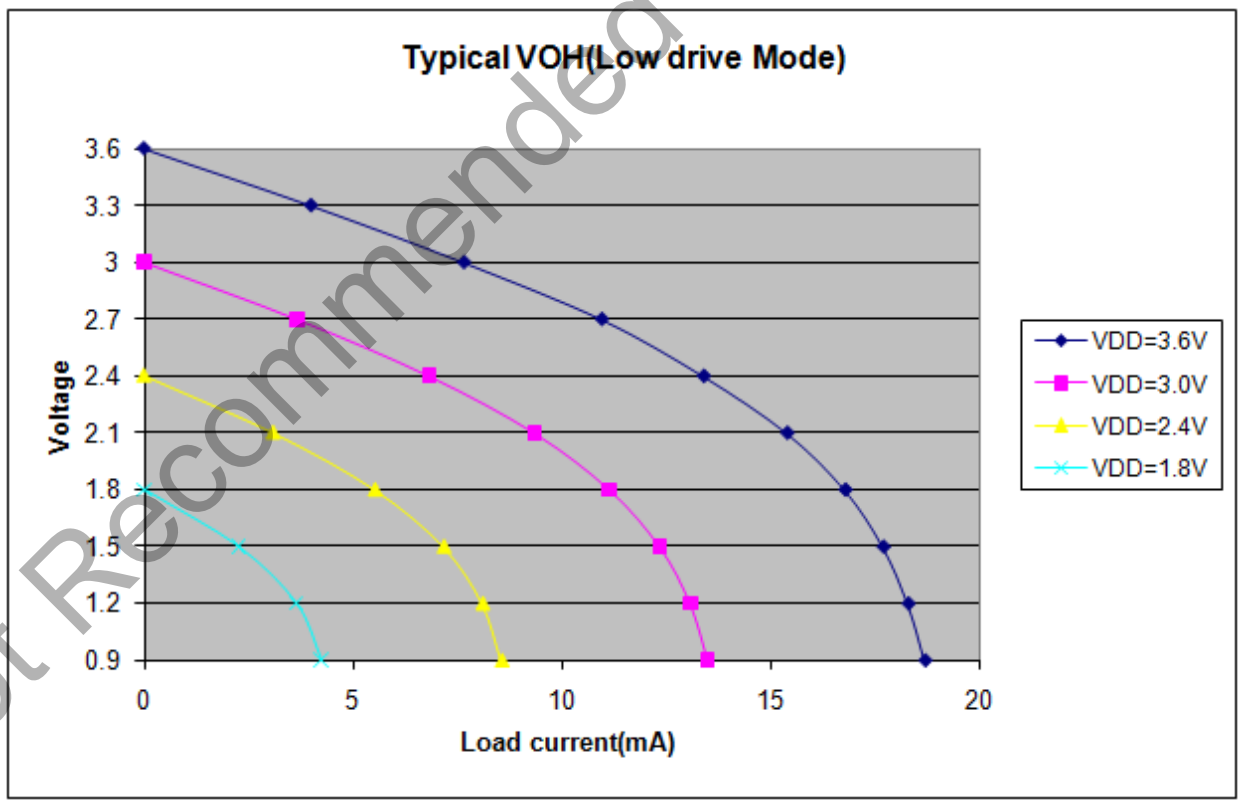


Figure 1.1. Typical VOH Curves, 1.8–3.6 V

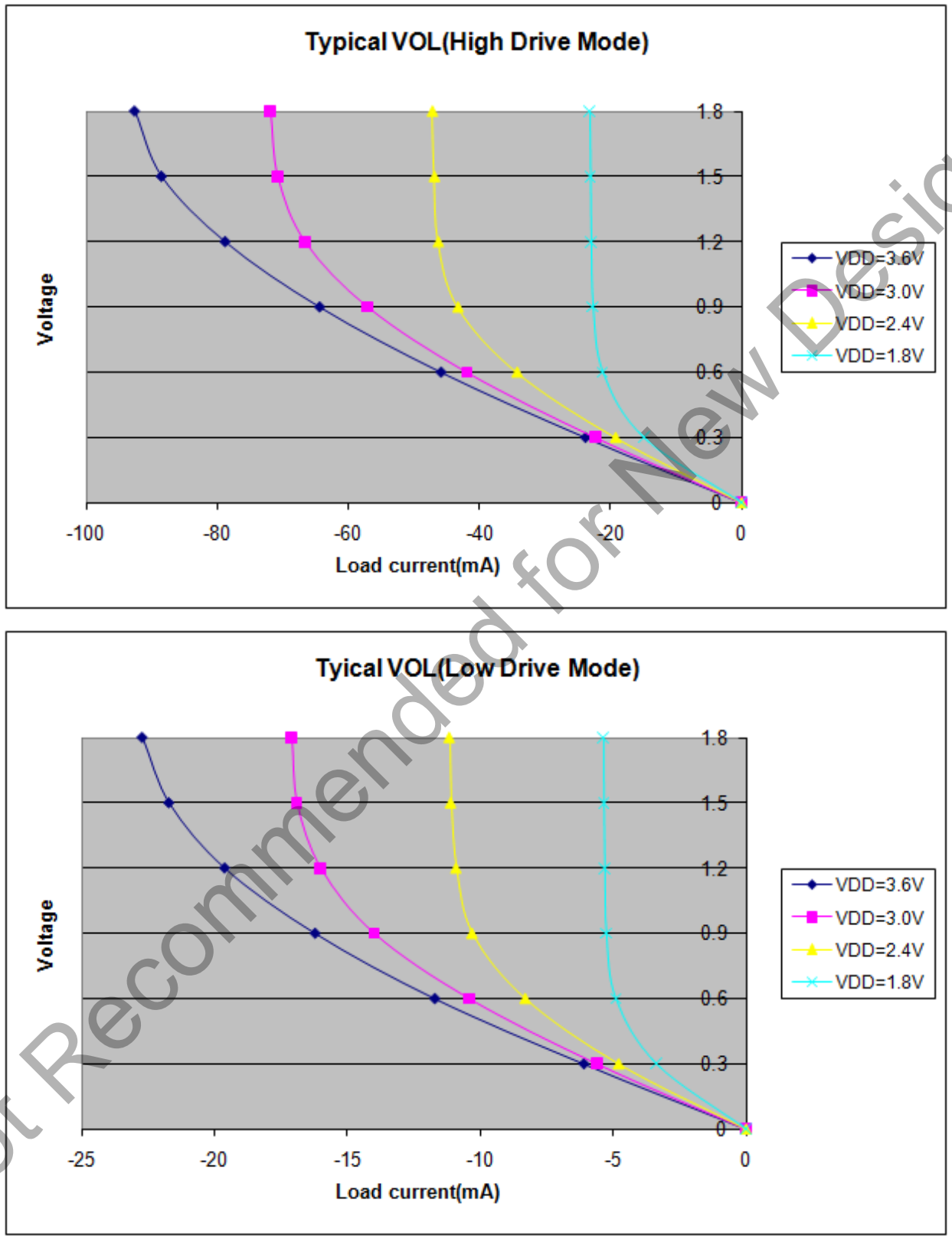


Figure 1.2. Typical VOL Curves, 1.8–3.6 V

Table 1.5. Reset Electrical Characteristics

V_{DD} = 1.8 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Unit
RST Output Low Voltage	I_{OL} = 1.4 mA	—	0.1	—	V
RST input High Voltage	V_{DD} = 2.0 to 3.6 V	V_{DD} – 0.6	—	—	V
RST input Low Voltage	V_{DD} = 2.0 to 3.6 V	—	—	0.6	V
RST Input Pull-up Current	RST = 0 V, V_{DD} = 3.6 V	—	22	—	µA
VDD Monitor Threshold (V_{RST})	Early Warning	—	1.85	—	V
VDD Monitor Threshold (V_{RST})	Reset Trigger (all modes ex. Sleep)	—	1.75	—	V
POR Monitor Threshold (V_{POR})	Initial Power-On	—	1.6	—	V
POR Monitor Threshold (V_{POR})	Brownout Condition	—	1.6	—	V
POR Monitor Threshold (V_{POR})	Recovery from Brownout	—	1.0	—	V
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	650	1000	µs
Minimum Sys Clock with Missing Clock Detector Enabled	System clock frequency which triggers a missing clock detector timeout	—	7	—	kHz
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	—	—	30	µs
Minimum RST Low Time to Generate System Reset		—	15	—	µs
VDD Monitor Turn-on Time		—	300	—	ns
VDD Monitor Supply Current		—	20	—	µA

Table 1.6. Power Management Electrical Specifications

V_{DD} = 1.8 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Idle Mode Wake-up Time		2	—	3	SYSCLKs
Suspend Mode Wake-up Time	Low power oscillator	—	400	—	ns
	Precision oscillator	—	1.3	—	µs
Sleep Mode Wake-up Time		—	2	—	µs

Table 1.7. Flash Electrical Characteristics

V_{DD} = 1.8 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	See ordering information for flash sizes of all C8051F97x devices	16384	—	32768	bytes
Endurance		20k	100k	—	
Erase Cycle Time		20	30	40	ms
Write Cycle Time		50	60	70	µs

Table 1.8. Internal Precision Oscillator Electrical Characteristics

V_{DD} = 1.8 to 3.6 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency	–40 to +85 °C, V_{DD} = 1.8–3.6 V	24	24.5	25	MHz
Oscillator Supply Current (from V_{DD})	25 °C; includes bias current of 90–100 µA	—	300	—	µA

Table 1.9. Internal Low-Power Oscillator Electrical Characteristics

V_{DD} = 1.8 to 3.6 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency (Low Power Oscillator)	–40 to +85 °C, V_{DD} = 1.8–3.6 V	18	20	22	MHz
Oscillator Supply Current (from V_{DD})	25 °C No separate bias current required	—	100	—	µA

Table 1.10. SmaRTClock Characteristics

V_{DD} = 1.8 to 3.6 V; T_A = –40 to +85 °C unless otherwise specified; Using factory-calibrated settings.

Parameter	Conditions	Min	Typ	Max	Units
Oscillator Frequency (LFO)		11	16.4	22	kHz

Table 1.11. ADC0 Electrical Characteristics

V_{DD} = 1.8 to 3.6 V V, VREF = 1.65 V (REFSL[1:0] = 11), –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		—	±<1	±2	LSB
Full Scale Error		—	±1	±2.5	LSB
Dynamic Performance (10 kHz sine-wave single-ended input, 1 dB below Full Scale, 300 ksps)					
Signal-to-Noise Plus Distortion		62	65	—	dB
Signal-to-Distortion		—	70	—	dB
Spurious-Free Dynamic Range		—	72	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	8.33	MHz
Conversion Time in SAR Clocks	10-bit Mode 8-bit Mode	13 11	— —	— —	clocks
Track/Hold Acquisition Time		1.5	—	—	µs
Throughput Rate		—	—	300	ksps
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ – GND)	0	—	VREF	V
Absolute Pin Voltage with respect to GND	Single Ended	0	—	V_{DD}	V
Sampling Capacitance	1x Gain 0.5x Gain	— —	28 26	— —	pF
Input Multiplexer Impedance		—	5	—	kΩ
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Conversion Mode (300 ksps) Tracking Mode (0 ksps)	— —	800 680	— —	µA
Power Supply Rejection	External VREF	—	76	—	dB

Table 1.12. Temperature Electrical Characteristics

Parameter	Condition	Min	Typ	Max	units
Linearity		—	±1	—	°C
Slope		—	3.4	—	mV/°C
Slope Error		—	40	—	µV/°C
Offset	Temp = 25 °C	—	1025	—	mV
Offset Error	Temp = 25 °C	—	18	—	mV
Supply Current		—	35	—	µA

Table 1.13. Voltage Reference Electrical Characteristics

V_{DD} = 1.8 to 3.6 V, –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal High-Speed Reference (REFSL[1:0] = 11)					
Output Voltage	–40 to +85 °C, V_{DD} = 1.8–3.6 V	1.60	1.65	1.70	V
Supply Current	Normal Power Mode Low Power Mode	— —	250 140	— —	µA
External Reference (REFSL[1:0] = 00, REFOE = 0)					
Input Voltage Range		0	—	V_{DD}	V
Input Current	Fs = 300 ksps; VREF = 3.0 V	—	5.25	—	µA

Table 1.14. CS0 Electrical Characteristics

V_{DD} = 1.8 to 3.6 V; T_A = –40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Single Conversion Time*	12-bit Mode 13-bit Mode (default) 14-bit Mode 16-bit Mode	20 21 23 26	25 27 29 33	40 42.5 45 50	µs
Power Supply Current		—	55	80	µA
***Note:** Conversion time is specified with the default configuration.					

1.2. Thermal Conditions

Table 1.15. Thermal Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Thermal Resistance Junction to Ambient	θ_{JA}	Still air, QFN 48 Still air, QFN 32 Still air, QFN 24	— — —	82 40.4 59.5	— — —	°C/W
Thermal Resistance Junction to Case	θ_{JC}	Still air, QFN 48 Still air, QFN 32 Still air, QFN 24	— — —	11.5 16.8 22.7	— — —	°C/W

1.3. Absolute Maximum Ratings

Stresses above those listed under Table 1.16 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.

For more information on the available quality and reliability data, see the Quality and Reliability Monitor Report at http://www.silabs.com/support/quality/pages/default.aspx.

Table 1.16. Absolute Maximum Limits

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	—	V_{DD} + 0.3	V
Voltage on V_{DD} with Respect to GND		–0.3	—	4.0	V
Maximum Total Current through V_{DD} or GND		—	—	500	mA
Maximum Current through $\overline{\text{RST}}$ or any Port Pin		—	—	100	mA
Maximum Total Current through all Port Pins		—	—	200	mA
Note: Exposure to maximum rating conditions for extended periods may affect device reliability.					

2. System Overview

The C8051F97x device family are fully integrated, mixed-signal system-on-a-chip MCUs. Highlighted features are listed below. Refer to Table 4.1 for specific product feature selection and part ordering numbers.

- **Core:**
  - Pipelined CIP-51 Core
  - Fully compatible with standard 8051 instruction set
  - 70% of instructions execute in 1-2 clock cycles
  - 25 MHz maximum operating frequency
- **Memory:**
  - 16-32 kB flash; in-system programmable in 512-byte sectors
  - 4352-8448 bytes RAM (including 256 bytes standard 8051 RAM and 4-8 kB on-chip XRAM)
- **Power:**
  - Ultra low power consumption in active and sleep modes.
  - Power-on reset circuit and brownout detectors
- **Capacitive Sensing:**
  - Supports button and slider elements
  - 40 us/channel conversion time
  - 16-bit resolution
  - Auto scan and wake-on-touch
  - Auto-accumulate up to 64 samples
- **I/O:**
  - Up to 43 total multifunction I/O pins:
  - Flexible peripheral crossbar for peripheral routing
- **Clock Sources:**
  - Precision internal oscillator: 24.5 MHz ±2%
  - Low Power Oscillator: 20 MHz ±20%
  - Low-frequency internal oscillator: 16.4 kHz
  - External crystal, RC, C, and CMOS options
- **Timers/Counters and PWM:**
  - 3-channel Programmable Counter Array (PCA) supporting PWM, capture/compare, frequency output modes, and watchdog timer function
  - 4x 16-bit general-purpose timers
- **Communications and Other Digital Peripherals:**
  - UART
  - SPI™
  - I^2C / SMBus™
  - High-Speed I^2C Slave
  - 16-bit CRC Unit, supporting automatic CRC of flash at 256-byte boundaries
- **Analog:**
  - 10-bit Analog-to-Digital Converter (ADC) (300 ksps)
- **Digital:**
  - 7-channel DMA
  - 16 x 16 Multiply and Accumulate hardware
- **Unique Identifier:**
  - 128-bit unique key for each device
- **On-Chip Debugging**

With on-chip power-on reset, voltage supply monitor, watchdog timer, and clock oscillator, the C8051F97x devices are truly standalone system-on-a-chip solutions. The flash memory is re-programmable in-circuit, providing non-volatile data storage and allowing field upgrades of the firmware.

The on-chip debugging interface (C2) 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, and run and halt commands. All analog and digital peripherals are fully functional while debugging.

Each device is specified for 1.8 to 3.6 V operation, and are available in 48-pin QFN, 32-pin QFN, or 24-pin QFN packages. All package options are lead-free and RoHS compliant. The device is available in two temperature grades: -40 to +85 °C. See Table 4.1 for ordering information. A block diagram is included in Figure 2.1.

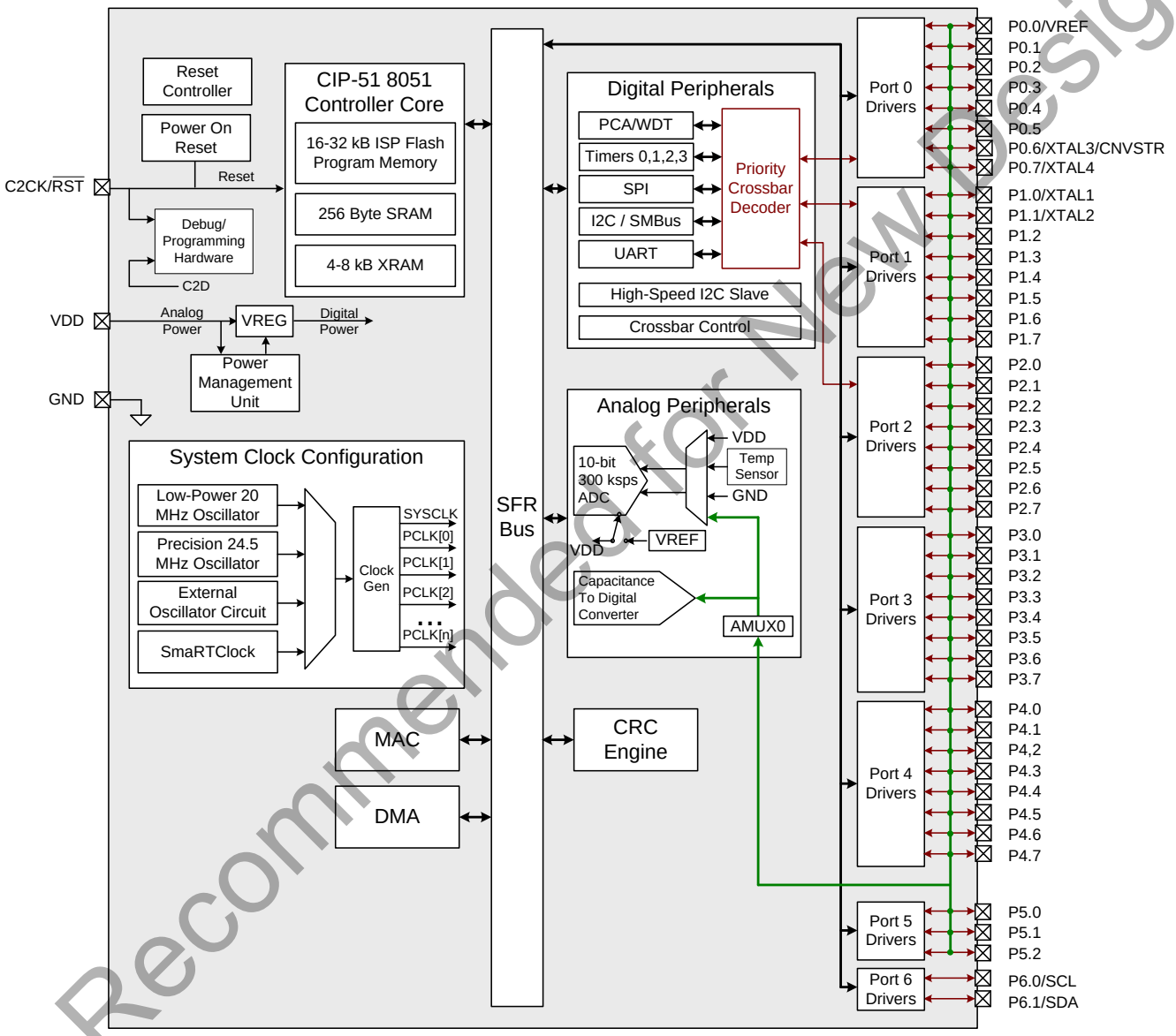


Figure 2.1. C8051F97x Family Block Diagram (QFN-48 Shown)

2.1. Power

2.1.1. Voltage Supply Monitor (VMON0)

The C8051F97x devices include a voltage supply monitor which allows devices to function in known, safe operating condition without the need for external hardware.

The supply monitor module includes the following features:

- Holds the device in reset if the main VDD supply drops below the VDD Reset threshold.

2.1.2. Device Power Modes

The C8051F97x devices feature seven low power modes in addition to normal operating mode, allowing the designer to save power when the core is not in use. All power modes are detailed in Table 2.1.

Table 2.1. Power Modes

Power Mode	Description	Wake-up Sources	Power and Performance
Normal Active	Device fully functional.	N/A	Excellent MIPS/mW
Low Power Active	Device fully functional except peripherals whose clocks are intentionally disabled.	N/A	Excellent. Clocks only enabled for peripherals that request for it.
Idle	All peripherals fully functional. Wake-up in 2 clock cycles.	Any Interrupt	Good No Code Execution
Low Power Idle	Similar to Idle mode, the CPU is halted. Clocks of unused peripherals can be intentionally gated. Wake-up in 2 clock cycles.	Any Interrupt	Very Good No Code Execution. Clocks only enabled for peripherals that request for it.
Stop	Legacy 8051 low power mode. A reset is required to wake up.	Any Reset	Good No Code Execution Precision Oscillator Disabled
Suspend	Similar to Stop mode, but very fast wake-up time and code resumes execution at the next instruction.	CS0, SmaRTClock, Port Match, I^2C Slave, $\overline{RST}$ pin	Very Good No Code Execution All Internal Oscillators Disabled System Clock Gated
Sleep	Ultra Low Power and flexible wake-up sources. Code resumes execution at the next instruction.	SmaRTClock, Port Match, I^2C Slave, $\overline{RST}$ pin	Excellent Power Supply Gated All Oscillators except SmaRTClock Disabled

2.1.2.1. Normal Mode

Normal mode encompasses the typical full-speed operation. The power consumption of the device in this mode will vary depending on the system clock speed and any analog peripherals that are enabled.

2.1.2.2. Idle Mode

Setting the IDLE bit in PCON causes the hardware 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 bit 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.

2.1.2.3. Stop Mode

Setting the Stop Mode Select bit in the PCON register causes the core to enter stop mode as soon as the instruction that sets the bit completes execution. In Stop mode, the precision internal oscillator and CPU are stopped; the state of the low power oscillator and 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.

2.1.3. Suspend Mode

Setting the Suspend Mode Select bit in the PMU0CF register causes the system clock to be gated off and all internal oscillators disabled. All digital logic (timers, communication peripherals, interrupts, CPU, etc.) stops functioning until one of the enabled wake-up sources occurs.

The following wake-up sources can be configured to wake the device from suspend mode:

- SmaRTClock oscillator fail
- SmaRTClock alarm
- Port Match event
- I2C0 address match
- CS0 end-of-conversion or end-of-scan

2.1.4. Sleep Mode

Setting the Sleep Mode Select bit in the PMU0CF register turns off the internal 1.8 V regulator (VREG0) and switches the power supply of all on-chip RAM to the VDD pin. Power to most digital logic on the chip is disconnected; only PMU0 and the SmaRTClock remain powered. All analog peripherals (ADC0, External Oscillator, etc.) should be disabled prior to entering sleep mode.

RAM and SFR register contents are preserved in sleep mode as long as the voltage on VDD does not fall below VPOR. The PC counter and all other volatile state information is preserved allowing the device to resume code execution upon waking up from sleep mode. The following wake-up sources can be configured to wake the device from sleep mode:

- SmaRTClock oscillator fail
- SmaRTClock alarm
- Port Match event
- I2C0 address match

2.1.5. Low Power Active Mode

Running in normal active mode can waste a significant amount of amount of power by clocking unused peripherals. Low power active mode in C8051F97x devices allows control of clocking activity in the clock tree, which enables firmware to shut off clocking to unused peripherals and save power. The CPU and all the analog and digital peripherals remain active except those whose clocks are turned off by firmware. The CPU will not be able to access SFR of peripherals on inactive branches of the clock tree.

Low power active mode is terminated when the CLKMODE register is programmed to 0x00 or a reset occurs. Systems that use all the peripherals and always stay in the active mode may not find improvement in power consumption in the low power active mode due overhead logic required for this implementation.

2.1.6. Low Power Idle Mode

In this mode of operation, the CPU is halted and clocks supplied to unused peripherals can be shut down to save power. All internal registers and memory maintain their original data. All the analog and digital peripherals remain active except those whose clocks are turned off by user. Modules that are capable of requesting for clocks can do so at any time.

Low power idle mode is terminated when an enabled interrupt is asserted or a reset occurs, but the interrupt only causes the device to switch from low power idle mode to low power active mode. To return to normal active mode, the CLKMODE register should be reset to 0x00. 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 low power idle mode is terminated by an internal or external reset, the CIP-51 performs a normal sequence and begins program execution at address 0x0000.

2.2. I/O

2.2.1. General Features

The C8051F97x ports have the following features:

- Push-pull or open-drain output modes and analog or digital modes.
- Port Match allows the device to recognize a change on a port pin value and wake from idle mode or generate an interrupt.
- Internal pull-up resistors can be globally enabled or disabled.
- Two external interrupts provide unique interrupt vectors for monitoring time-critical events.

2.2.2. Crossbar

The C8051F97x devices have a digital peripheral crossbar with the following features:

- Flexible peripheral assignment to port pins.
- Pins can be individually skipped to move peripherals as needed for design or layout considerations.

The crossbar has a fixed priority for each I/O function and assigns these functions to the port pins. When a digital resource is selected, the least-significant unassigned port pin is assigned to that resource. 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. This provides some flexibility when designing a system: pins involved with sensitive analog measurements can be moved away from digital I/O and peripherals can be moved around the chip as needed to ease layout constraints.

2.3. Clocking

The C8051F97x devices have three internal oscillators and the option to use an external crystal, RC, C, or CMOS oscillator as the system clock. A programmable divider allows the user to internally run the system clock at a slower rate than the selected oscillator if desired.

2.4. Counters/Timers and PWM

2.4.1. Programmable Counter Array (PCA0)

The C8051F97x devices include a three-channel, 16-bit Programmable Counter Array with the following features:

- 16-bit time base.
- Programmable clock divisor and clock source selection.
- Three independently-configurable channels.
- 8, 9, 10, 11 and 16-bit PWM modes (edge-aligned operation).
- Output polarity control.
- Frequency output mode.
- Capture on rising, falling or any edge.
- Compare function for arbitrary waveform generation.
- Software timer (internal compare) mode.
- Module 2 acts as a Watchdog Timer.

2.4.2. Timers (Timer 0, Timer 1, Timer 2, and Timer 3)

Timers include the following features:

- Timer 0 and Timer 1 are standard 8051 timers, supporting backwards-compatibility with firmware and hardware.
- Timer 2 and Timer 3 can each operate as 16-bit auto-reload or two independent 8-bit auto-reload timers, and include Comparator 0 or SmaRTClock clock capture capabilities.

2.5. Communications and other Digital Peripherals

2.5.1. Universal Asynchronous Receiver/Transmitter (UART0)

The UART uses two signals (TX and RX) and a predetermined fixed baud rate to provide asynchronous communications with other devices.

The UART module provides the following features:

- Asynchronous transmissions and receptions.
- Baud rates up to SYSCLK / 2 (transmit) or SYSCLK / 8 (receive).
- 8 or 9 bit data.
- Automatic start and stop generation.

2.5.2. Serial Peripheral Interface (SPI0)

SPI is a 3- or 4-wire communication interface that includes a clock, input data, output data, and an optional select signal.

The SPI module includes the following features:

- Supports 3- or 4-wire master or slave modes.
- Supports external clock frequencies up to SYSCLK / 2 in master mode and SYSCLK / 10 in slave mode.
- Support for all clock phase and polarity modes.
- 8-bit programmable clock rate.
- Support for multiple masters on the same data lines.

2.5.3. System Management Bus / I2C (SMBus0)

The SMBus interface is a two-wire, bidirectional serial bus compatible with both I2C and SMBus protocols. The two clock and data signals operate in open-drain mode with external pull-ups to support automatic bus arbitration.

Reads and writes to the interface are byte oriented with the SMBus interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/8th of the system clock as a master or slave, which can be faster than allowed by the SMBus / I2C specification, depending on the clock source 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.

The SMBus module includes the following features:

- Standard (up to 100 kbps) and Fast (400 kbps) transfer speeds.
- Support for master, slave, and multi-master modes.
- Hardware synchronization and arbitration for multi-master mode.
- Clock low extending (clock stretching) to interface with faster masters.
- Hardware support for 7-bit slave and general call address recognition.
- Firmware support for 10-bit slave address decoding.
- Ability to inhibit all slave states.
- Programmable data setup/hold times.

2.5.4. High-Speed I2C Slave (I2CSLAVE0)

The I2C Slave 0 interface is a 2-wire, bidirectional serial bus that is compatible with the I^2C Bus Specification 3.0. It is capable of transferring in high-speed mode (HS-mode) at speeds of up to 3.4 Mbps. Either the CPU or the DMA can write to the I^2C interface, and the I^2C interface can autonomously control the serial transfer of data. The interface also supports clock stretching for cases where the CPU may be temporarily prohibited from transmitting a byte or processing a received byte during an I^2C transaction. It can also operate in sleep mode without an active system clock and wake the CPU when a matching slave address is received.

It operates only as an I^2C slave device. The I2CSLAVE0 peripheral provides control of the SCL (serial clock) synchronization, SDA (serial data), SCL Clock stretching, I^2C arbitration logic, and low power mode operation.

The I2C Slave 0 module includes the following features:

- High-speed (up to 3.4 Mbps), fast (400 kbps), and standard (up to 100 kbps) transfer speeds.
- Support for slave mode only.
- Clock low extending (clock stretching) to interface with faster masters.
- Hardware support for 7-bit slave and general call address recognition.
- Can operate in sleep mode without an active system clock and wake the CPU after receiving a matching slave address.
- Internal pull-up resistors.

2.5.5. 16/32-bit CRC (CRC0)

The CRC module is designed to provide hardware calculations for flash memory verification and communications protocols. The CRC module supports the standard CCITT-16 16-bit polynomial (0x1021), and includes the following features:

- Support for four CCITT-16 polynomial.
- Byte-level bit reversal.
- Automatic CRC of flash contents on one or more 256-byte blocks.
- Initial seed selection of 0x0000 or 0xFFFF.

2.6. Analog Peripherals

2.6.1. 10-Bit Analog-to-Digital Converter (ADC0)

The ADC0 module on C8051F97x devices is a Successive Approximation Register (SAR) Analog to Digital Converter (ADC). The key features of the ADC module are:

- Single-ended 10-bit mode.
- Supports an output update rate of 300 ksps samples per second.
- Selectable asynchronous hardware conversion trigger.
- Output data window comparator allows automatic range checking.
- Support for Burst Mode, which produces one set of accumulated data per conversion-start trigger with programmable power-on settling and tracking time.
- Conversion complete and window compare interrupts supported.
- Flexible output data formatting.
- Includes an internal 1.65 V fast-settling reference with two levels and support for an external reference.

2.7. Digital Peripherals

2.7.1. Direct Memory Access (DMA0)

The DMA0 module on C8051F97x devices enables autonomous data movement between XRAM and select peripherals. The key features of the DMA module are:

- Supports 7 channels.
- Separate full-length and mid-point transfer interrupts for each channel.
- Options for big or little endian transfers.
- Directly supports the I2C Slave 0 and MAC0 modules.

2.7.2. Multiply and Accumulate (MAC0)

The C8051F97x devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. The MAC0 module has the following key features:

- Single cycle operation.
- Multiply-accumulate or multiply only.
- Support for integer or fractional operations.
- Support for signed or unsigned operations.
- Rounding with saturation.
- Auto-increment or constant A and/or B registers.
- Logical 1-bit or multiple bit shift of accumulator left or right.
- Negation of accumulator, A, and/or B registers.
- DMA support for repetitive operations on large arrays of data.
- Signed and unsigned alignment (right shift in bytes) of accumulator result.

2.8. 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:

- The core halts program execution.
- Module registers are initialized to their defined reset values unless the bits reset only with a power-on reset.
- External port pins are forced to a known state.
- Interrupts and timers are disabled.

All registers are reset to the predefined values noted in the register descriptions unless the bits only reset with a power-on reset. The contents of RAM are unaffected during a reset; any previously stored data is preserved as long as power is not lost.

The Port I/O latches are reset to 1 in open-drain mode. Weak pullups are enabled during and after the reset. For VDD Supply Monitor resets, the $\overline{\text{RST}}$ pin is driven low until the device exits the reset state.

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

2.9. Unique Identifier

Each device contains a 128-bit unique identifier (UID) at the last 16 bytes of XRAM. This value is reloaded after each device reset, and firmware can overwrite the memory area during operation, if desired.

2.10. On-Chip Debugging

The C8051F97x 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 bidirectional 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.

3. Pin Definitions

3.1. C8051F970/3 QFN-48 Pin Definitions

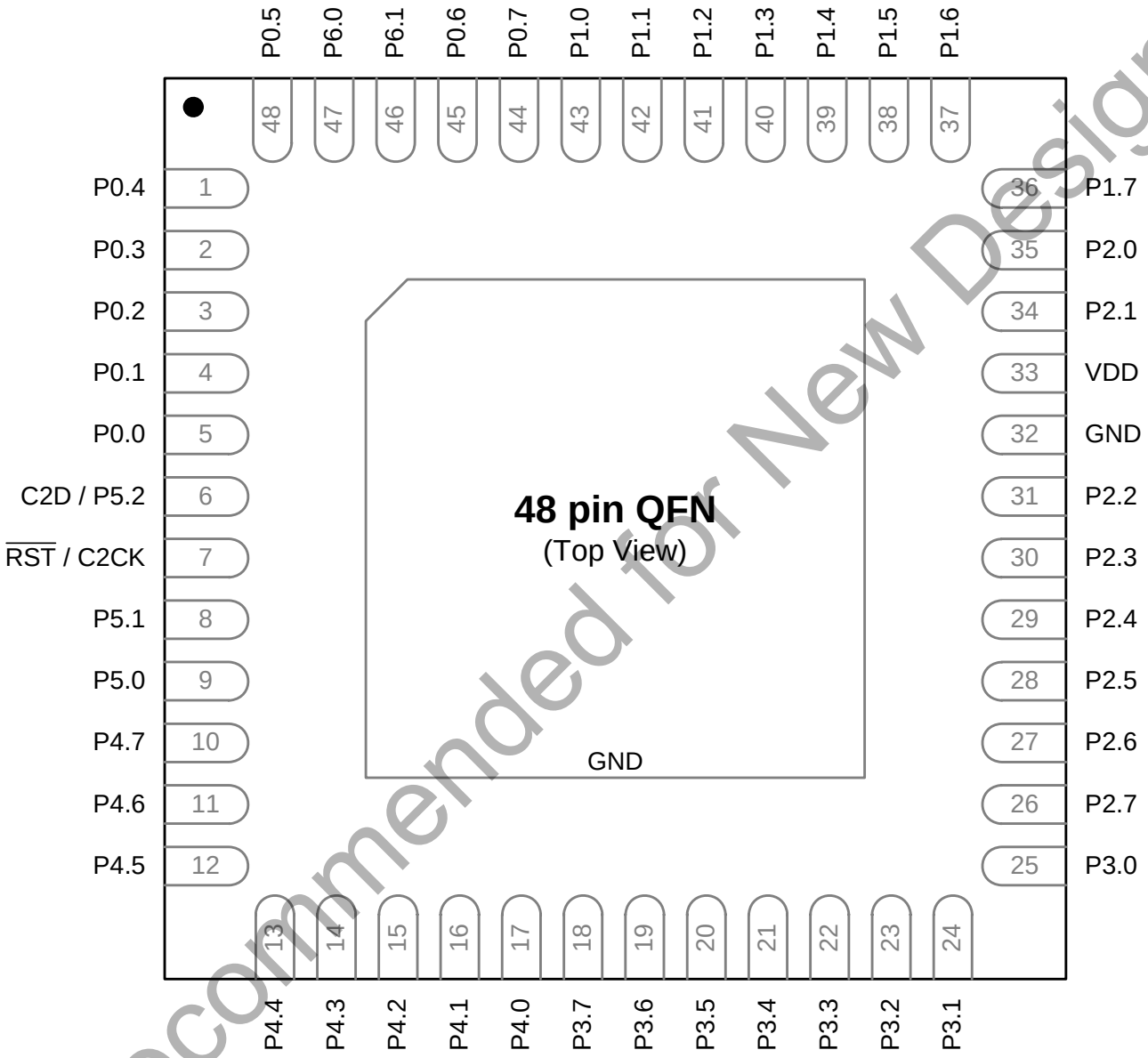


Figure 3.1. C8051F970/3-A-GM (QFN-48) Pinout

Table 3.1. Pin Definitions for C8051F970/3-A-GM (QFN-48)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
GND	Ground	32 / Bottom			
VDD	Power	33			
$\overline{\text{RST}}$ / C2CK	Active-low Reset / C2 Debug Clock	7			
P0.0	Standard I/O	5	Yes	P0MAT.0 INT0.0 INT1.0	AMUX0.0 CS0.0 VREF
P0.1	Standard I/O	4	Yes	P0MAT.1 INT0.1 INT1.1	AMUX0.1 CS0.1
P0.2	Standard I/O	3	Yes	P0MAT.2 INT0.2 INT1.2 RTC Osc. Output	AMUX0.2 CS0.2
P0.3	Standard I/O	2	Yes	P0MAT.3 INT0.3 INT1.3 Wake-up Request	AMUX0.3 CS0.3
P0.4	Standard I/O	1	Yes	P0MAT.4 INT0.4 INT1.4	AMUX0.4 CS0.4
P0.5	Standard I/O	48	Yes	P0MAT.5 INT0.5 INT1.5	AMUX0.5 CS0.5
P0.6	Standard I/O SmaRTClock Input	45	Yes	P0MAT.6 INT0.6 INT1.6 CNVSTR	AMUX0.6 CS0.6 XTAL3
P0.7	Standard I/O SmaRTClock Input	44	Yes	P0MAT.7 INT0.7 INT1.7	AMUX0.7 CS0.7 XTAL4
P1.0	Standard I/O External Oscillator Input	43	Yes	P1MAT.0	AMUX0.8 CS0.8 XTAL1

Table 3.1. Pin Definitions for C8051F970/3-A-GM (QFN-48) (Continued)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P1.1	Standard I/O External Oscillator Input	42	Yes	P1MAT.1	AMUX0.9 CS0.9 XTAL2
P1.2	Standard I/O	41	Yes	P1MAT.2	AMUX0.10 CS0.10
P1.3	Standard I/O	40	Yes	P1MAT.3	AMUX0.11 CS0.11
P1.4	Standard I/O	39	Yes	P1MAT.4	AMUX0.12 CS0.12
P1.5	Standard I/O	38	Yes	P1MAT.5	AMUX0.13 CS0.13
P1.6	Standard I/O	37	Yes	P1MAT.6	AMUX0.14 CS0.14
P1.7	Standard I/O	36	Yes	P1MAT.7	AMUX0.15 CS0.15
P2.0	Standard I/O	35	Yes	P2MAT.0	AMUX0.16 CS0.16
P2.1	Standard I/O	34	Yes	P2MAT.1	AMUX0.17 CS0.17
P2.2	Standard I/O	31	Yes	P2MAT.2	AMUX0.18 CS0.18
P2.3	Standard I/O	30	Yes	P2MAT.3	AMUX0.19 CS0.19
P2.4	Standard I/O	29	Yes	P2MAT.4	AMUX0.20 CS0.20
P2.5	Standard I/O	28	Yes	P2MAT.5	AMUX0.21 CS0.21
P2.6	Standard I/O	27	Yes	P2MAT.6	AMUX0.22 CS0.22
P2.7	Standard I/O	26	Yes	P2MAT.7	AMUX0.23 CS0.23
P3.0	Standard I/O	25			AMUX0.24 CS0.24

Table 3.1. Pin Definitions for C8051F970/3-A-GM (QFN-48) (Continued)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P3.1	Standard I/O	24			AMUX0.25 CS0.25
P3.2	Standard I/O	23			AMUX0.26 CS0.26
P3.3	Standard I/O	22			AMUX0.27 CS0.27
P3.4	Standard I/O	21			AMUX0.28 CS0.28
P3.5	Standard I/O	20			AMUX0.29 CS0.29
P3.6	Standard I/O	19			AMUX0.30 CS0.30
P3.7	Standard I/O	18			AMUX0.31 CS0.31
P4.0	Standard I/O	17			AMUX0.32 CS0.32
P4.1	Standard I/O	16			AMUX0.33 CS0.33
P4.2	Standard I/O	15			AMUX0.34 CS0.34
P4.3	Standard I/O	14			AMUX0.35 CS0.35
P4.4	Standard I/O	13			AMUX0.36 CS0.36
P4.5	Standard I/O	12			AMUX0.37 CS0.37
P4.6	Standard I/O	11			AMUX0.38 CS0.38
P4.7	Standard I/O	10			AMUX0.39 CS0.39
P5.0	Standard I/O	9			AMUX0.40 CS0.40
P5.1	Standard I/O	8			AMUX0.41 CS0.41

Table 3.1. Pin Definitions for C8051F970/3-A-GM (QFN-48) (Continued)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P5.2 / C2D	Standard I/O / C2 Debug Data	6			
P6.0	Standard I/O	47		I2CSLAVE_SCL	
P6.1	Standard I/O	46		I2CSLAVE_SDA	

3.2. C8051F971/4 QFN-32 Pin Definitions

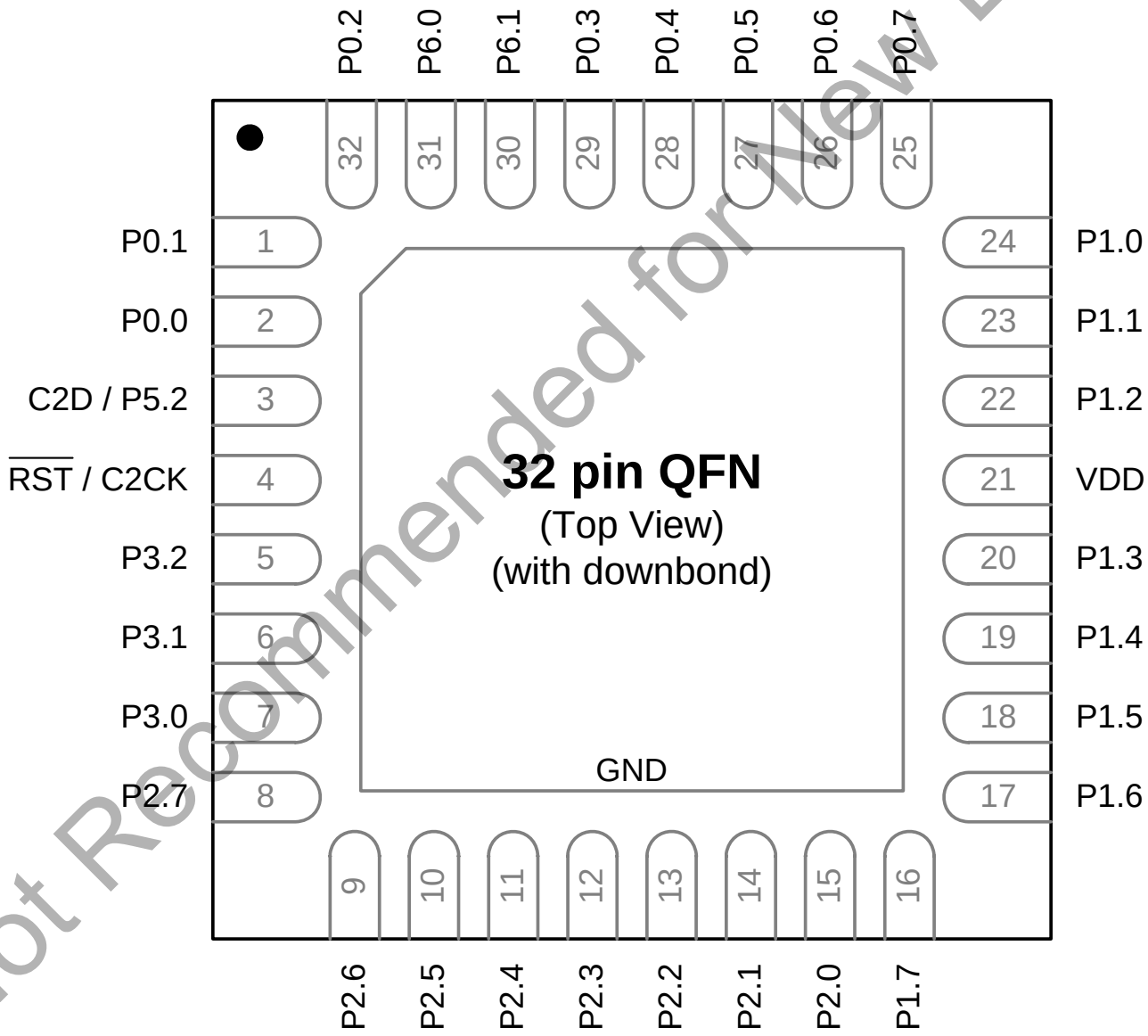


Figure 3.2. C8051F971/4-A-GM (QFN-32) Pinout

Table 3.2. Pin Definitions for C8051F971/4-A-GM (QFN-32)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
GND	Ground	Bottom			
VDD	Power	21			
$\overline{\text{RST}}$ / C2CK	Active-low Reset / C2 Debug Clock	4			
P0.0	Standard I/O	2	Yes	P0MAT.0 INT0.0 INT1.0	AMUX0.0 CS0.0 VREF
P0.1	Standard I/O	1	Yes	P0MAT.1 INT0.1 INT1.1	AMUX0.1 CS0.1
P0.2	Standard I/O	32	Yes	P0MAT.2 INT0.2 INT1.2	AMUX0.2 CS0.2
P0.3	Standard I/O	29	Yes	P0MAT.3 INT0.3 INT1.3	AMUX0.3 CS0.3 XTAL3
P0.4	Standard I/O	28	Yes	P0MAT.4 INT0.4 INT1.4	AMUX0.4 CS0.4 XTAL4
P0.5	Standard I/O	27	Yes	P0MAT.5 INT0.5 INT1.5	AMUX0.5 CS0.5 XTAL1
P0.6	Standard I/O SmaRTClock Input	26	Yes	P0MAT.6 INT0.6 INT1.6 CNVSTR	AMUX0.6 CS0.6 XTAL2
P0.7	Standard I/O SmaRTClock Input	25	Yes	P0MAT.7 INT0.7 INT1.7	AMUX0.7 CS0.7
P1.0	Standard I/O External Oscillator Input	24	Yes	P1MAT.0	AMUX0.8 CS0.8
P1.1	Standard I/O External Oscillator Input	23	Yes	P1MAT.1	AMUX0.9 CS0.9
P1.2	Standard I/O	22	Yes	P1MAT.2	AMUX0.10 CS0.10
P1.3	Standard I/O	20	Yes	P1MAT.3	AMUX0.11 CS0.11

Table 3.2. Pin Definitions for C8051F971/4-A-GM (QFN-32) (Continued)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P1.4	Standard I/O	19	Yes	P1MAT.4	AMUX0.12 CS0.12
P1.5	Standard I/O	18	Yes	P1MAT.5	AMUX0.13 CS0.13
P1.6	Standard I/O	17	Yes	P1MAT.6	AMUX0.14 CS0.14
P1.7	Standard I/O	16	Yes	P1MAT.7	AMUX0.15 CS0.15
P2.0	Standard I/O	15	Yes	P2MAT.0	AMUX0.16 CS0.16
P2.1	Standard I/O	14	Yes	P2MAT.1	AMUX0.17 CS0.17
P2.2	Standard I/O	13	Yes	P2MAT.2	AMUX0.18 CS0.18
P2.3	Standard I/O	12	Yes	P2MAT.3	AMUX0.19 CS0.19
P2.4	Standard I/O	11	Yes	P2MAT.4	AMUX0.20 CS0.20
P2.5	Standard I/O	10	Yes	P2MAT.5	AMUX0.21 CS0.21
P2.6	Standard I/O	9	Yes	P2MAT.6	AMUX0.22 CS0.22
P2.7	Standard I/O	8	Yes	P2MAT.7	AMUX0.23 CS0.23
P3.0	Standard I/O	7			AMUX0.24 CS0.24
P3.1	Standard I/O	6			AMUX0.25 CS0.25
P3.2	Standard I/O	5			AMUX0.26 CS0.26
P5.2 / C2D	Standard I/O / C2 Debug Data	3			
P6.0	Standard I/O	31	Yes	I2CSLAVE_SCL	
P6.1	Standard I/O	30	Yes	I2CSLAVE_SDA	

3.3. C8051F972/5 QFN-24 Pin Definitions

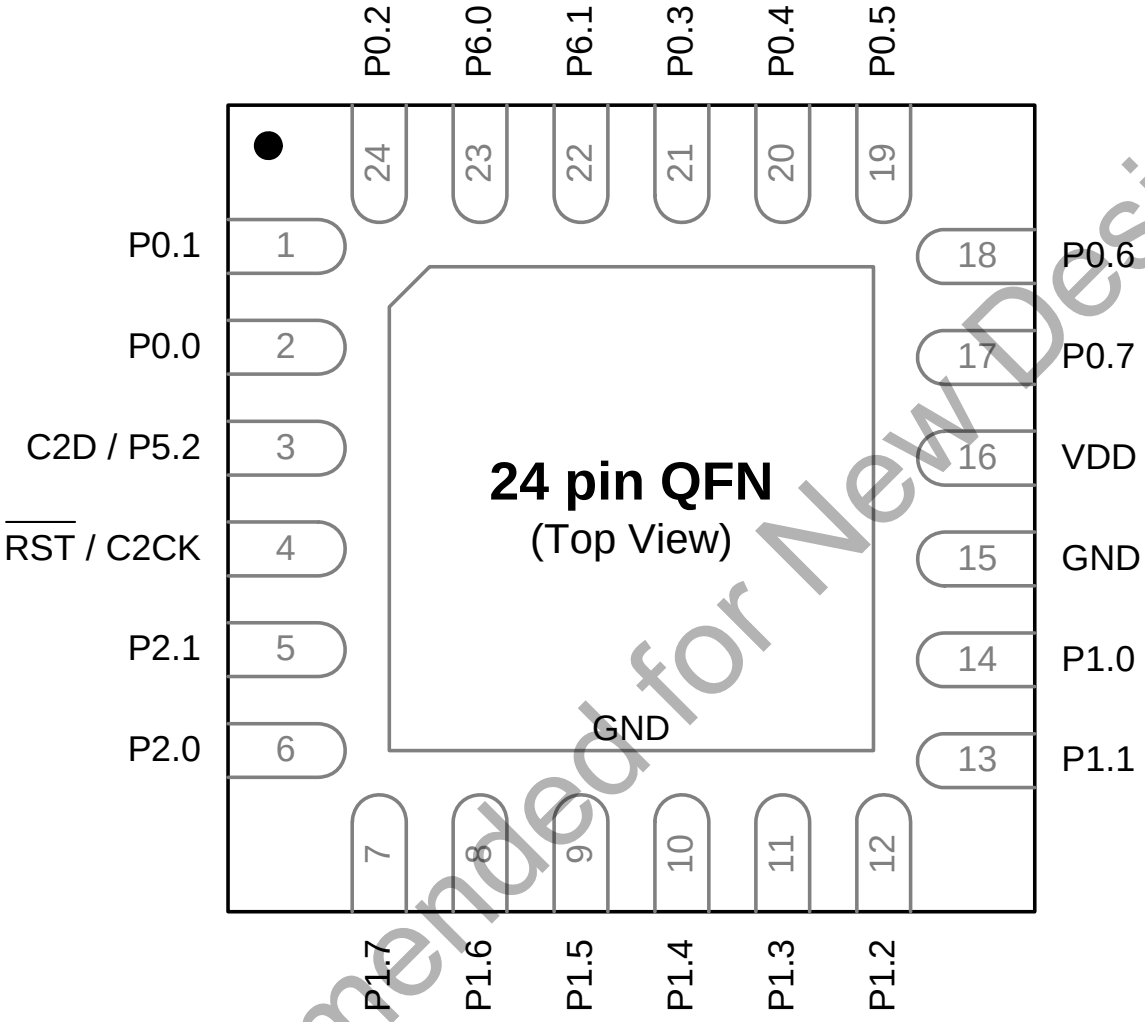


Figure 3.3. C8051F972/5-A-GM (QFN-24) Pinout

Table 3.3. Pin Definitions for C8051F972/5-A-GM (QFN-24)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
GND	Ground	15 / Bottom			
VDD	Power	16			
$\overline{\text{RST}}$ / C2CK	Active-low Reset / C2 Debug Clock	4			
P0.0	Standard I/O	2	Yes	P0MAT.0 INT0.0 INT1.0	AMUX0.0 CS0.0 VREF
P0.1	Standard I/O	1	Yes	P0MAT.1 INT0.1 INT1.1	AMUX0.1 CS0.1
P0.2	Standard I/O	24	Yes	P0MAT.2 INT0.2 INT1.2	AMUX0.2 CS0.2
P0.3	Standard I/O	21	Yes	P0MAT.3 EXTCLK INT0.3 INT1.3	AMUX0.3 CS0.3
P0.4	Standard I/O	20	Yes	P0MAT.4 INT0.4 INT1.4	AMUX0.4 CS0.4
P0.5	Standard I/O	19	Yes	P0MAT.5 INT0.5 INT1.5	AMUX0.5 CS0.5
P0.6	Standard I/O	18	Yes	P0MAT.6 INT0.6 INT1.6 CNVSTR	AMUX0.6 CS0.6
P0.7	Standard I/O	17	Yes	P0MAT.7 INT0.7 INT1.7	AMUX0.7 CS0.7
P1.0	Standard I/O	14	Yes	P1MAT.0	AMUX0.8 CS0.8
P1.1	Standard I/O	13	Yes	P1MAT.1	AMUX0.9 CS0.9

Table 3.3. Pin Definitions for C8051F972/5-A-GM (QFN-24) (Continued)

Pin Name	Type	Pin Numbers	Crossbar Capability	Additional Digital Functions	Analog Functions
P1.2	Standard I/O	12	Yes	P1MAT.2	AMUX0.10 CS0.10
P1.3	Standard I/O	11	Yes	P1MAT.3	AMUX0.11 CS0.11
P1.4	Standard I/O	10	Yes	P1MAT.4	AMUX0.12 CS0.12
P1.5	Standard I/O	9	Yes	P1MAT.5	AMUX0.13 CS0.13
P1.6	Standard I/O	8	Yes	P1MAT.6	AMUX0.14 CS0.14
P1.7	Standard I/O	7	Yes	P1MAT.7	AMUX0.15 CS0.15
P2.0	Standard I/O	6	Yes	P2MAT.0	AMUX0.16 CS0.16
P2.1	Standard I/O	5	Yes	P2MAT.1	AMUX0.17 CS0.17
P5.2 / C2D	Standard I/O / C2 Debug Data	3			
P6.0	Standard I/O	23	Yes	I2CSLAVE_SCL	
P6.1	Standard I/O	22	Yes	I2CSLAVE_SDA	

4. Ordering Information

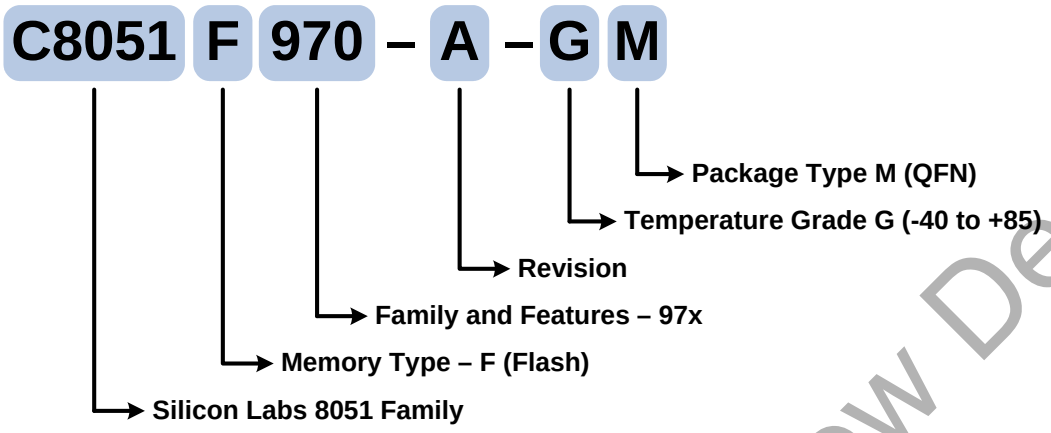


Figure 4.1. C8051F97x Part Numbering

All C8051F97x family members have the following features:

- CIP-51 Core running up to 25 MHz
- Two Internal Oscillators (24.5 MHz and 80 kHz)
- I2C/SMBus
- SPI
- UART
- 3-Channel Programmable Counter Array (PWM, Clock Generation, Capture/Compare)
- 4 Timers
- Capacitance-to-Digital (CS0) module
- Real Time Clock
- 16-bit CRC Unit

In addition to these features, each part number in the C8051F97x family has a set of features that vary across the product line. The product selection guide in Table 4.1 shows the features available on each family member.

Table 4.1. Product Selection Guide

Ordering Part Number	Flash Memory (kB)	RAM (Bytes)	Digital Port I/Os (Total)	Number of ADC0 Channels	Pb-Free (RoHS Compliant)	Temperature Range	Package
C8051F970-A-GM	32	8kB	43	43	✓	–40 to 85 °C	QFN-48
C8051F971-A-GM	32	8kB	28	28	✓	–40 to 85 °C	QFN-32
C8051F972-A-GM	32	8kB	19	19	✓	–40 to 85 °C	QFN-24
C8051F973-A-GM	16	4kB	43	43	✓	–40 to 85 °C	QFN-48
C8051F974-A-GM	16	4kB	28	28	✓	–40 to 85 °C	QFN-32
C8051F975-A-GM	16	4kB	19	19	✓	–40 to 85 °C	QFN-24

5. QFN-48 Package Specifications

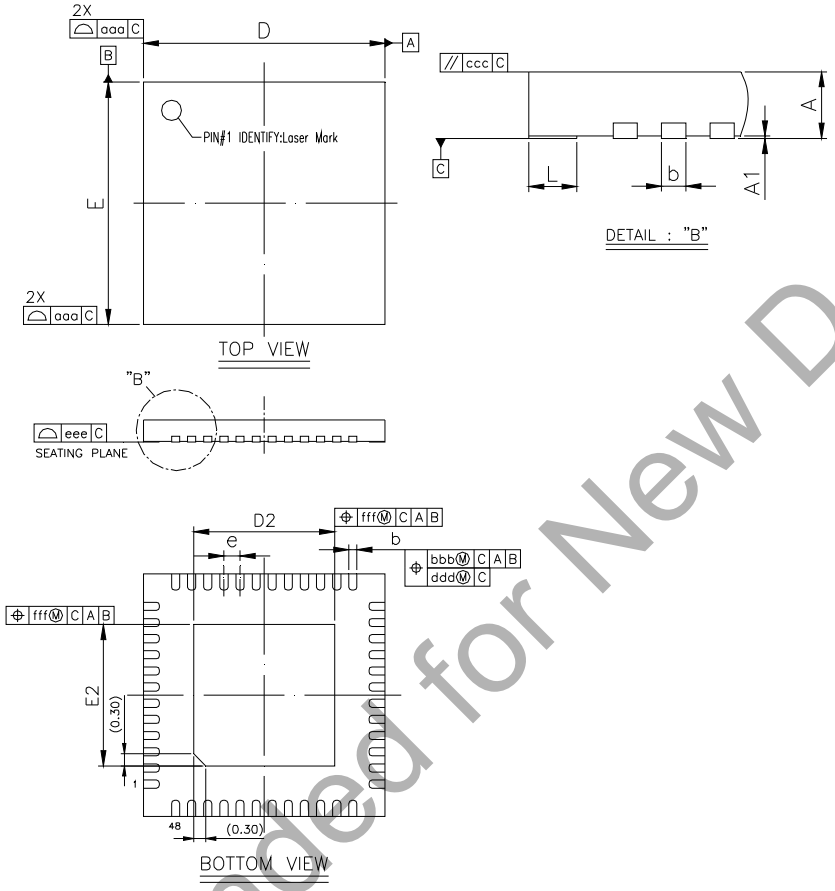


Figure 5.1. QFN-48 Package Drawing (6x6 mm)

Table 5.1. QFN-48 Package Dimensions

Dimension	Min	Typ	Max
A	0.50	0.55	0.60
A1	0.00	0.02	0.05
b	0.15	0.20	0.25
D		6.00 BSC	
D2	3.35	3.50	3.65
e		0.40 BSC	
E		6.00 BSC	

Dimension	Min	Typ	Max
D2	3.35	3.50	3.65
L	0.30	0.40	0.50
aaa		0.10	
bbb		0.07	
ccc		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 JEDEC outline MO-220.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

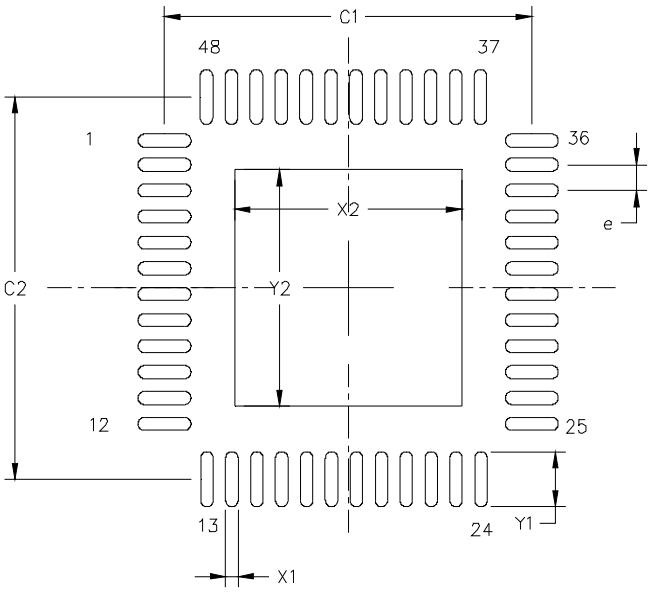


Figure 5.2. QFN-48 Landing Diagram

Table 5.2. QFN-48 Landing Diagram Dimensions

Dimension	Max	Dimension	Max
e	0.40 BSC	X2	3.65
C1	5.90	Y1	0.85
C2	5.90	Y2	3.65
X1	0.20		

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimension and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on IPC-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

5. 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

6. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
7. The stencil thickness should be 0.125 mm (5 mils).
8. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
9. A 3x3 array of 0.90 mm square openings on 1.15 mm pitch should be used for the center ground pad.

Card Assembly

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

5.1. QFN-48 Package Marking

The first row of the package marking is the part number, including the revision. The second row is the 6-digit trace code indicating assembly information. The last row indicates year (two digits) and workweek (two digits) when the device was packaged.

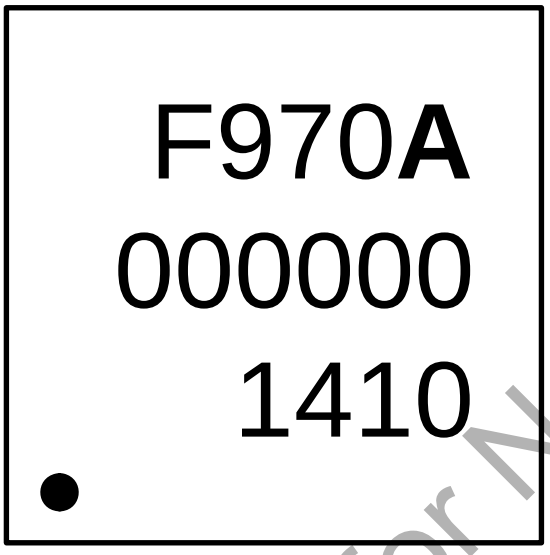


Figure 5.3. QFN-48 Package Marking

6. QFN-32 Package Specifications

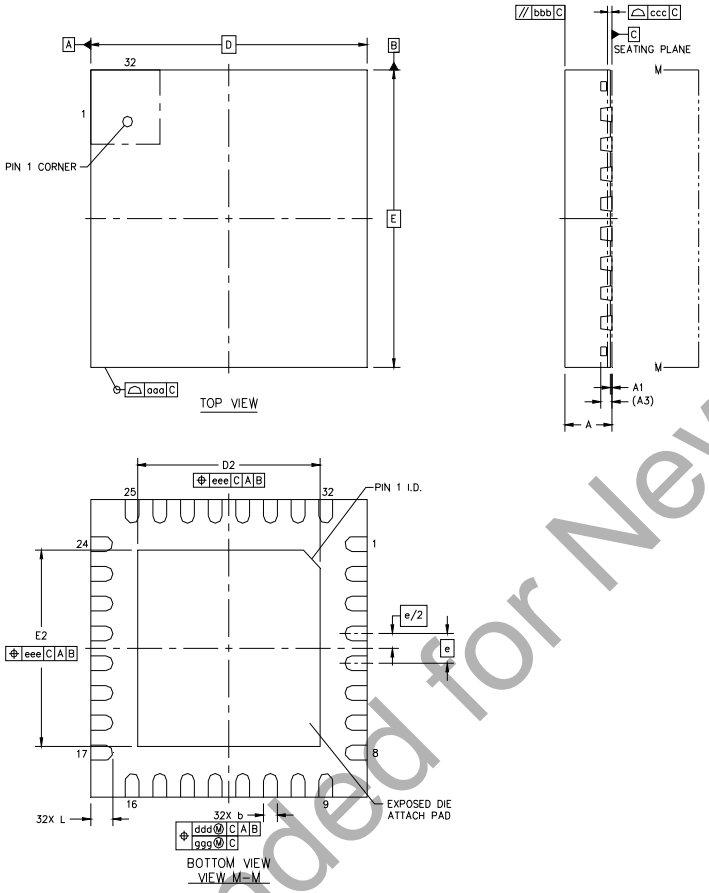


Figure 6.1. QFN-32 Package Drawing

Table 6.1. QFN-32 Package Dimensions

Dimension	Min	Typ	Max
A	0.80	0.85	0.90
A1	0.00	0.02	0.05
b	0.20	0.25	0.30
A3		0.203 REF	
D		5.00 BSC	
D2	3.20	3.30	3.40
e		0.50 BSC	
E		5.00 BSC	

Dimension	Min	Typ	Max
E2	3.20	3.30	3.40
L	0.35	0.40	0.45
aaa	—	—	0.10
bbb	—	—	0.10
ccc	—	—	0.08
ddd	—	—	0.05
eee	—	—	0.08
ggg	—	—	0.05

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

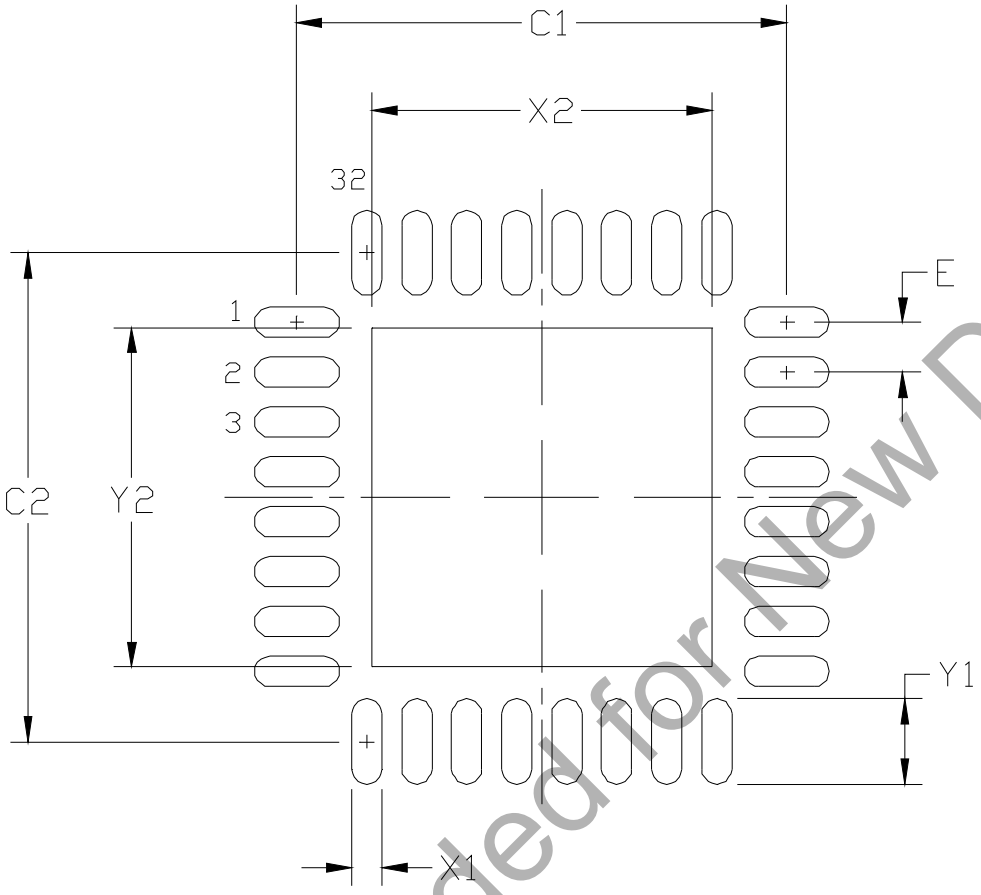


Figure 6.2. Typical QFN-32 Landing Diagram

Table 6.2. QFN-32 PCB Land Pattern

Dimension	MIN	MAX
C1	4.80	4.90
C2	4.80	4.90
E	0.50 BSC	
X1	0.20	0.30
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 60 m minimum, all the way around the pad.

Stencil Design

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

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.

6.1. QFN-32 Package Marking

The first row of the package marking is the part number, including the revision. The second row is the 6-digit trace code indicating assembly information. The last row indicates year (two digits) and workweek (two digits) when the device was packaged.

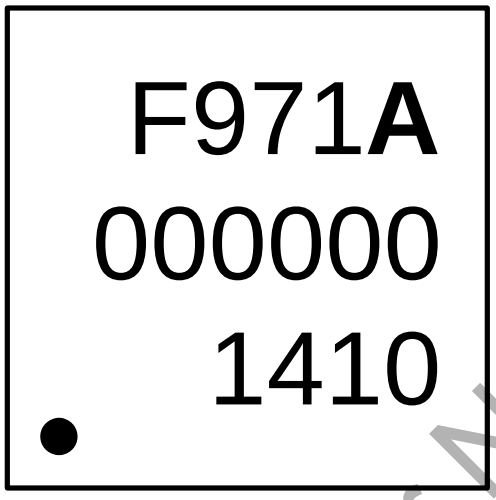


Figure 6.3. QFN-32 Package Marking

7. QFN-24 Package Specifications

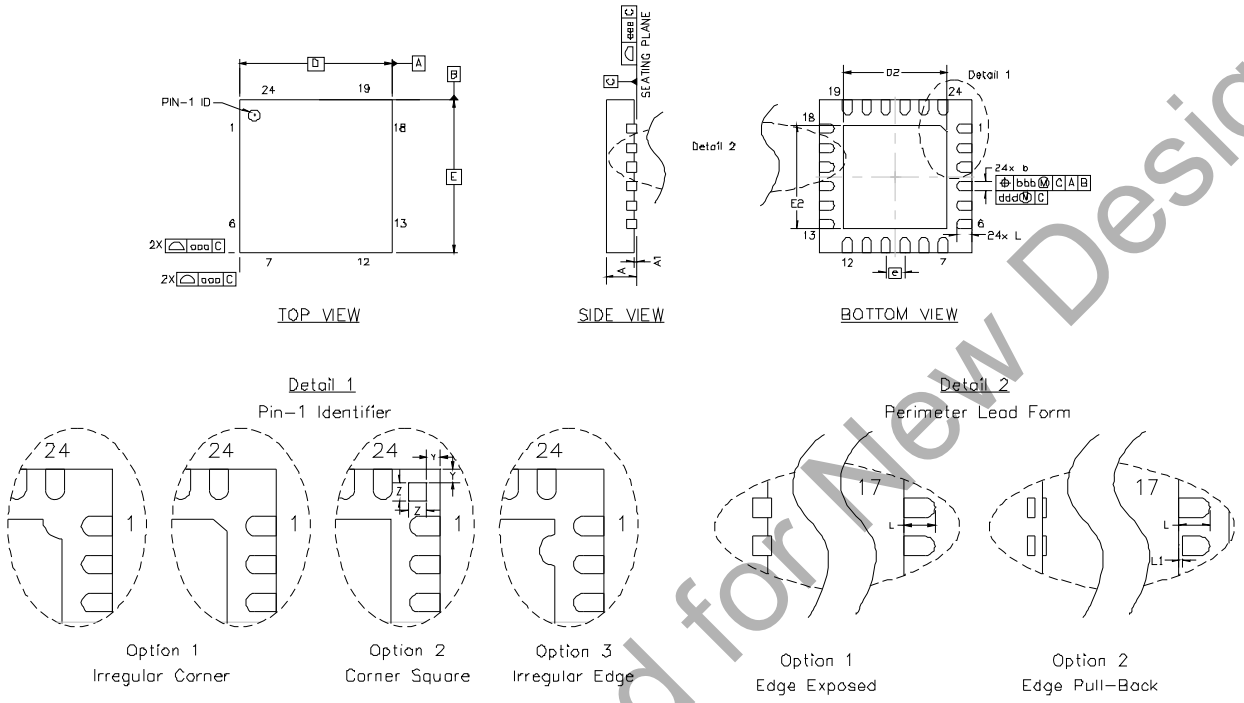


Figure 7.1. QFN-24 Package Drawing

Table 7.1. QFN-24 Package Dimensions

Dimension	Min	Typ	Max
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D		4.00 BSC	
D2	2.55	2.70	2.80
e		0.50 BSC	
E		4.00 BSC	
E2	2.55	2.70	2.80

Dimension	Min	Typ	Max
L	0.30	0.40	0.50
L1	0.00	—	0.15
aaa	—	—	0.15
bbb	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08
Z	—	0.24	—
Y	—	0.18	—

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to the JEDEC Solid State Outline MO-220, variation WGGD except for custom features D2, E2, Z, Y, 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.

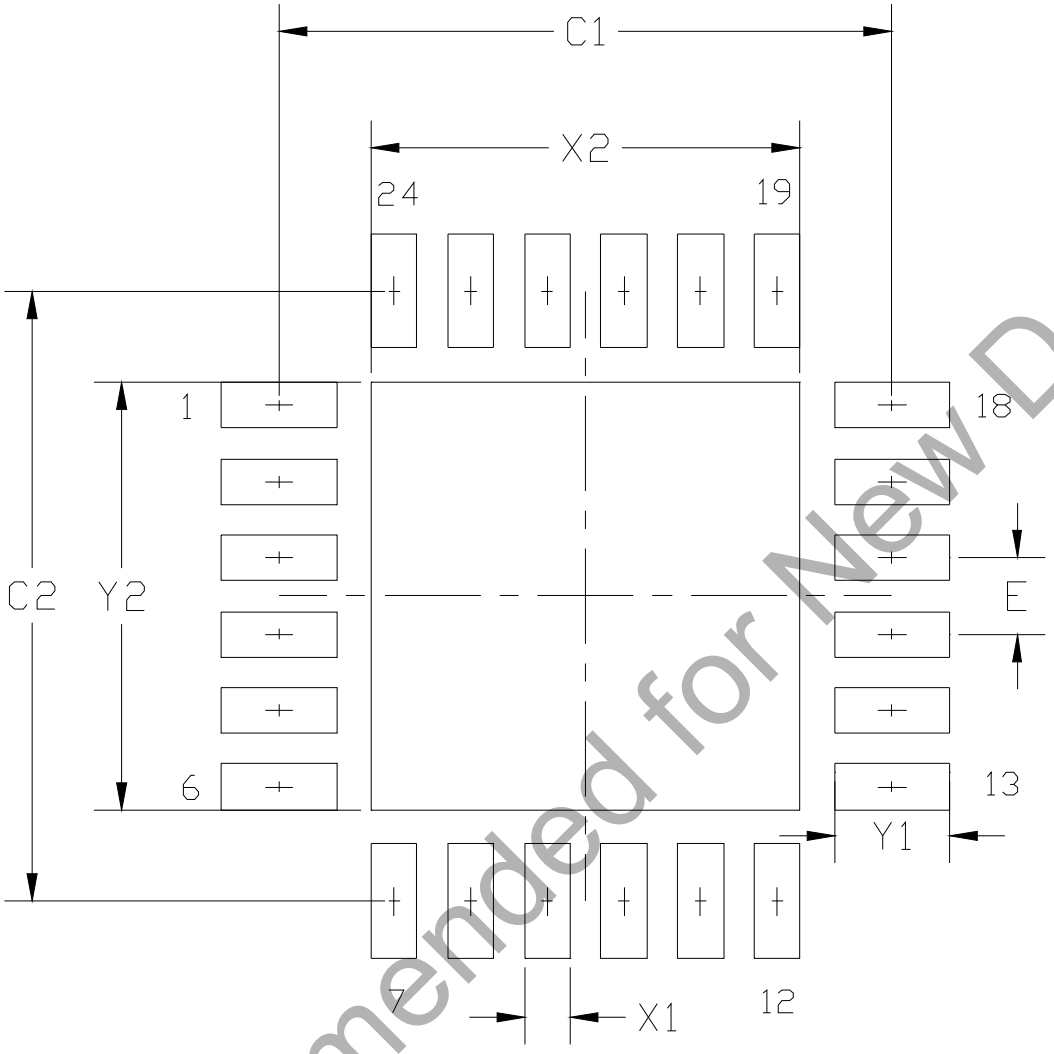


Figure 7.2. Typical QFN-24 Landing Diagram

Table 7.2. QFN-24 PCB Land Pattern

Dimension	MIN	MAX
C1	3.90	4.00
C2	3.90	4.00
E	0.50 BSC	
X1	0.20	0.30
X2	2.70	2.80
Y1	0.65	0.75
Y2	2.70	2.80

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 perimeter pads.
7. A 2x2 array of 1.10 mm x 1.10 mm openings on 1.30 mm pitch should be used for the center ground pad.

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.

7.1. QFN-24 Package Marking

The first row of the package marking is the part number, including the revision. The second row is the 6-digit trace code indicating assembly information. The last row indicates year (two digits) and workweek (two digits) when the device was packaged.

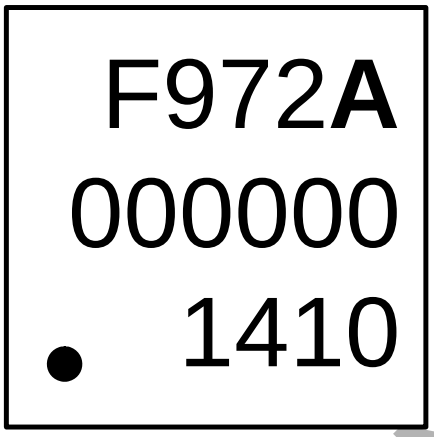


Figure 7.3. QFN-24 Package Marking

8. 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 memory organization of the C8051F97x device family is shown in Figure 8.1.

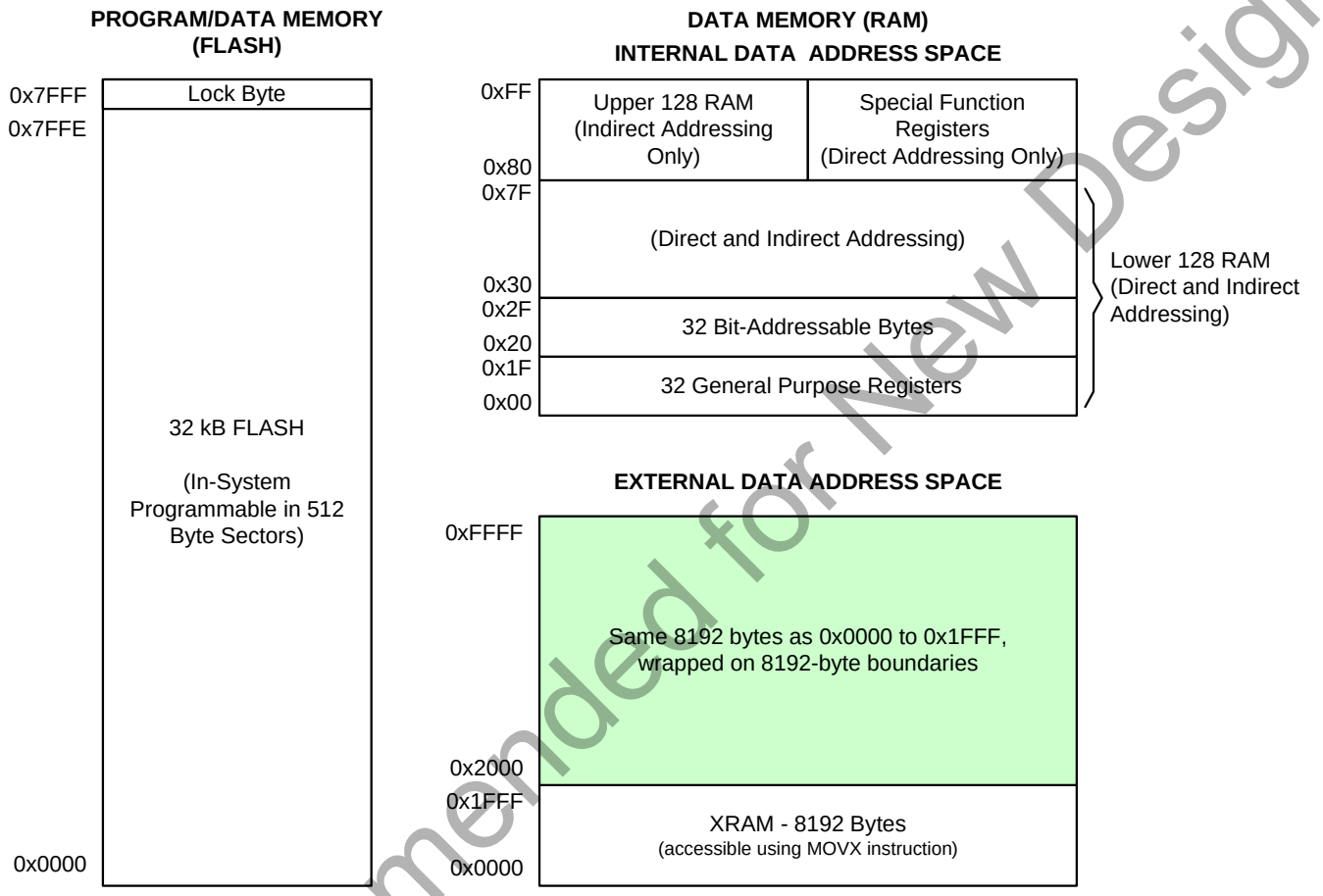


Figure 8.1. C8051F97x Memory Map (32 kB Flash Version Shown)

8.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F97x family implements 32 kB, or 16 kB of this program memory space as in-system, re-programmable flash memory. The last address in the flash block (0x7FFF on 32 kB devices and 0x3FFF on 16 kB devices) serves as a security lock byte for the device, and provides read, write and erase protection. Addresses above the lock byte within the 64 kB address space are reserved.

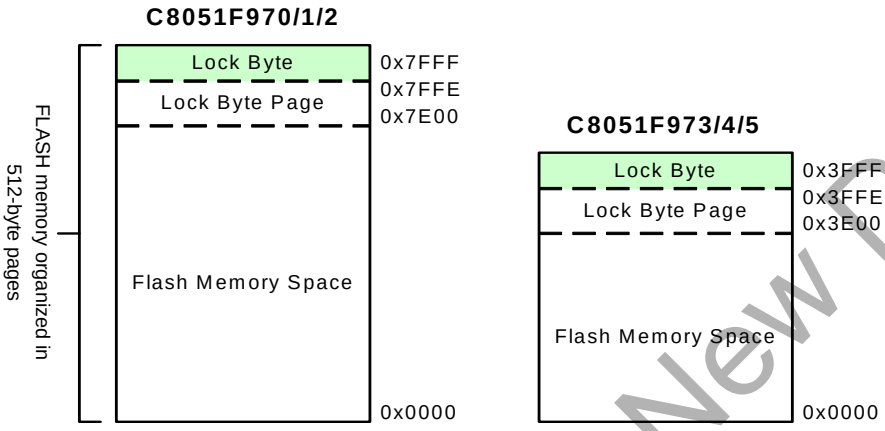


Figure 8.2. Flash Program Memory Map

8.1.1. MOVX Instruction and Program Memory

The MOVX instruction in an 8051 device is typically used to access external data memory. On the C8051F97x devices, the MOVX instruction is normally used to read and write on-chip XRAM, but can be re-configured to write and erase on-chip flash memory space. MOVC instructions are always used to read flash memory, while MOVX write instructions are used to erase and write flash. This flash access feature provides a mechanism for the C8051F97x to update program code and use the program memory space for non-volatile data storage. Refer to Section “10. Flash Memory” on page 65 for further details.

8.2. Data Memory

The C8051F97x device family includes up to 512 bytes of RAM data memory. 256 bytes of this memory is mapped into the internal RAM space of the 8051. On devices with 512 bytes total RAM, 256 additional bytes of memory are available as on-chip “external” memory. The data memory map is shown in Figure 8.1 for reference.

8.2.1. Internal RAM

There are 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and 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 8.1 illustrates the data memory organization of the C8051F97x.

8.2.1.1. 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 (PSW) register, RS0 and RS1, select the active register bank. This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

8.2.1.2. 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, 22.3h
```

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

8.2.1.3. 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) 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.

8.2.2. External RAM

C8051F97x devices have 4 kB or 8 kB of on-chip RAM mapped into the external data memory space. All of these address locations may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode. Note: the 16-bit MOVX instruction is also used for writes to the flash memory. See Section “10. Flash Memory” on page 65 for details. The MOVX instruction accesses XRAM by default.

For a 16-bit MOVX operation (@DPTR), the upper 8 bits of the 16-bit external data memory address word are “don't cares”. As a result, addresses 0x0000 through 0x00FF are mapped modulo style over the entire 64 k external data memory address range. For example, the XRAM byte at address 0x0000 is shadowed at addresses 0x0100, 0x0200, 0x0300, 0x0400, etc.

The C8051F97x devices implement the upper four bytes of external RAM as a 128-bit Unique Identifier. More information can be found in “Device Identification and Unique Identifier” on page 76.

8.2.3. 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.

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.

9. Special Function Register Memory Map

This section details the special function register memory map for the C8051F97x devices.

Table 9.1. C8051F97x SFR Memory Map

	Address								Page
	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)	
F8	SPI0CN I2C0STAT	PCA0L —	PCA0H —	PCA0CPL0 —	PCA0CPH0 —	CS0THL —	CS0THH —	VDM0CN P5MDOUT	0 F
F0	**B**	— P4MDIN	— P5MDIN	**SFRNEXT**	SMB0ADR P0MAT	SMB0ADM P1MAT	PCLKEN EIP1	CLKMODE EIP2	0 F
E8	ADC0CN DMA0INT	PCA0CPL1 —	PCA0CPH1 —	PCA0CPL2 —	PCA0CPH2 P0MDIN	CS0DL P1MDIN	CS0DH P2MDIN	RSTSRC P3MDIN	0 F
E0	**ACC**	P3 DEVICEID	P4 REVID	P5 —	P6 —	**FLWR**	**EIE1**	**EIE2**	0 F
D8	PCA0CN DMA0NCF	PCA0MD P0MDOUT	PCA0CPM0 —	PCA0CPM1 —	PCA0CPM2 P1MDOUT	CS0SS P3MDOUT	CS0SE IT01CF	PCA0PWM P3MDOUT	0 F
D0	**PSW**	REF0CN —	ADC0L MAC0ACC0	ADC0H MAC0ACC1	ADC0MX MAC0ACC2	— MAC0ACC3	— MAC0OVF	— MAC0ITER	0 F
C8	**TMR2CN**	REG0CN DMA0NBAL	TMR2RLL DMA0NBAH	TMR2RLH DMA0NAOL	TMR2L DMA0NAOH	TMR2H DMA0NSZL	PMU0FL DMA0NSZH	PMU0MD MAC0STA	0 F
C0	SMB0CN MAC0CF0	SMB0CF MAC0INTE	SMB0DAT —	ADC0GTL P4MDOUT	ADC0GTH MAC0CF1	ADC0LTL MAC0CF2	ADC0LTH P1SKIP	— P2SKIP	0 F
B8	**IP**	EMI0CN P4DRV	ADC0AC —	**ADC0PWR**	**ADC0TK**	CS0MD1 —	CS0MD2 —	CS0MD3 —	0 F
B0	CS0CN —	OSCXCN —	OSCICN —	**SFRLAST**	—	PMU0CF —	FLSCL P0SKIP	**FLKEY**	0 F
A8	**IE**	CLKSEL	CS0CF MAC0AL	CS0MX MAC0AH	RTC0ADR I2C0CN	RTC0DAT I2C0SLAD	MAC0BL	OSCICL MAC0BH	0 F
A0	**P2**	SPI0CFG —	SPI0CKR —	SPI0DAT —	AMUX0P3 I2C0DOUT	AMUX0P4 I2C0DIN	AMUX0P5 SFRPGCN	**SFRPAGE**	0 F
98	**SCON0**	SBUF0 P0DRV	AMUX0P0 P1DRV	AMUX0P1 P2DRV	AMUX0P2 P3DRV	CRC0CNT P5DRV	CRC0AUTO —	CRC0FLIP —	0 F
90	**P1**	TMR3CN DMA0BUSY	TMR3RLL DMA0EN	TMR3RLH —	TMR3L DMA0SEL	TMR3H XBR0	CS0PM XBR1	ADC0CF P6MDIN	0 F
88	TCON DMA0MINT	TMOD —	TL0 —	TL1 P0MASK	TH0 P1MASK	TH1 TOFFL	CKCON TOFFH	**PSCTL**	0 F
80	**P0**	**SP**	**DPL**	**DPH**	CRC0CN P2MASK	CRC0IN P2MAT	CRC0DAT —	**PCON**	0 F

denotes bit-addressable SFRs, **bold** indicates SFRs on all pages

Table 9.2. Special Function Registers

Register	Address	SFR Page	Register Description	Page
ACC	0xE0	All pages	Accumulator	180
ADC0AC	0xBA	0x0	ADC0 Accumulator Configuration	116
ADC0CF	0x97	0x0	ADC0 Configuration	115
ADC0CN	0xE8	0x0	ADC0 Control	114
ADC0GTH	0xC4	0x0	ADC0 Greater-Than High Byte	121
ADC0GTL	0xC3	0x0	ADC0 Greater-Than Low Byte	122
ADC0H	0xD3	0x0	ADC0 Data Word High Byte	119
ADC0L	0xD2	0x0	ADC0 Data Word Low Byte	120
ADC0LTH	0xC6	0x0	ADC0 Less-Than High Byte	123
ADC0LTL	0xC5	0x0	ADC0 Less-Than Low Byte	124
ADC0MX	0xD4	0x0	ADC0 Multiplexer Selection	125
ADC0PWR	0xBB	All pages	ADC0 Power Control	117
ADC0TK	0xBC	All pages	ADC0 Burst Mode Track Time	118
AMUX0P0	0x9A	0x0	Port 0 Analog Multiplexer Control	158
AMUX0P1	0x9B	0x0	Port 1 Analog Multiplexer Control	159
AMUX0P2	0x9C	0x0	Port 2 Analog Multiplexer Control	160
AMUX0P3	0xA4	0x0	Port 3 Analog Multiplexer Control	161
AMUX0P4	0xA5	0x0	Port 4 Analog Multiplexer Control	162
AMUX0P5	0xA6	0x0	Port 5 Analog Multiplexer Control	163
B	0xF0	0xF	B Register	181
CKCON	0x8E	0x0	Clock Control	400
CLKMODE	0xF7	0x0	Clock Mode	249
CLKSEL	0xA9	0x0	Clock Select	247
CRC0AUTO	0x9E	0x0	CRC0 Automatic Control	238
CRC0CN	0x84	0x0	CRC0 Control	235
CRC0CNT	0x9D	0x0	CRC0 Automatic Flash Sector Count	239
CRC0DAT	0x86	0x0	CRC0 Data Output	237
CRC0FLIP	0x9F	0x0	CRC0 Bit Flip	240

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
CRC0IN	0x85	0x0	CRC0 Data Input	236
CS0CF	0xAA	0x0	Capacitive Sense 0 Configuration	144
CS0CN	0xB0	0x0	Capacitive Sense 0 Control	143
CS0DH	0xEE	0x0	Capacitive Sense 0 Data High Byte	145
CS0DL	0xED	0x0	Capacitive Sense 0 Data Low Byte	146
CS0MD1	0xBD	0x0	Capacitive Sense 0 Mode 1	151
CS0MD2	0xBE	0x0	Capacitive Sense 0 Mode 2	152
CS0MD3	0xBF	0x0	Capacitive Sense 0 Mode 3	153
CS0MX	0xAB	0x0	Capacitive Sense 0 Mux Channel Select	155
CS0PM	0x96	0x0	Capacitive Sense 0 Pin Monitor	154
CS0SE	0xDE	0x0	Capacitive Sense 0 Auto Scan End Channel	148
CS0SS	0xDD	0x0	Capacitive Sense 0 Auto Scan Start Channel	147
CS0THH	0xFE	0x0	Capacitive Sense 0 Comparator Threshold High Byte	149
CS0THL	0xFD	0x0	Capacitive Sense 0 Comparator Threshold Low Byte	150
DEVICEID	0xE1	0xF	Device Identification	77
DMA0BUSY	0x91	0xF	DMA0 Busy	193
DMA0EN	0x92	0xF	DMA0 Channel Enable	190
DMA0INT	0xE8	0xF	DMA0 Full-Length Interrupt Flags	191
DMA0MINT	0x88	0xF	DMA0 Mid-Point Interrupt Flags	192
DMA0NAOH	0xCC	0xF	Memory Address Offset High	198
DMA0NAOL	0xCB	0xF	Memory Address Offset Low	199
DMA0NBAH	0xCA	0xF	Memory Base Address High	196
DMA0NBAL	0xC9	0xF	Memory Base Address Low	197
DMA0NCF	0xD8	0xF	DMA0 Channel Configuration	195
DMA0NSZH	0xCE	0xF	Memory Transfer Size High	200
DMA0NSZL	0xCD	0xF	Memory Transfer Size Low	201
DMA0SEL	0x94	0xF	DMA0 Channel Select	194
DPH	0x83	All pages	Data Pointer High	178
DPL	0x82	All pages	Data Pointer Low	177

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
EIE1	0xE6	All pages	Extended Interrupt Enable 1	85
EIE2	0xE7	All pages	Extended Interrupt Enable 2	89
EIP1	0xF6	0xF	Extended Interrupt Priority 1	87
EIP2	0xF7	0xF	Extended Interrupt Priority 2	90
EMI0CN	0xB9	0x0	External Memory Interface Control	75
FLKEY	0xB7	All pages	Flash Lock and Key	71
FLSCL	0xB6	0x0	Flash Scale	72
FLWR	0xE5	All pages	Flash Write Only	73
I2C0CN	0xAC	0xF	I2C0 Control	379
I2C0DIN	0xA5	0xF	I2C0 Received Data	374
I2C0DOUT	0xA4	0xF	I2C0 Transmit Data	375
I2C0SLAD	0xAD	0xF	I2C0 Slave Address	376
I2C0STAT	0xF8	0xF	I2C0 Status	377
IE	0xA8	All pages	Interrupt Enable	82
IP	0xB8	All pages	Interrupt Priority	84
IT01CF	0xDE	0xF	INT0 / INT1 Configuration	92
MAC0ACC0	0xD2	0xF	Accumulator Byte 0	230
MAC0ACC1	0xD3	0xF	Accumulator Byte 1	229
MAC0ACC2	0xD4	0xF	Accumulator Byte 2	228
MAC0ACC3	0xD5	0xF	Accumulator Byte 3	227
MAC0AH	0xAB	0xF	Operand A High Byte	222
MAC0AL	0xAA	0xF	Operand A Low Byte	223
MAC0BH	0xAF	0xF	Operand B High Byte	224
MAC0BL	0xAE	0xF	Operand B Low Byte	225
MAC0CF0	0xC0	0xF	MAC0 Configuration 0	215
MAC0CF1	0xC4	0xF	MAC0 Configuration 1	217
MAC0CF2	0xC5	0xF	MAC0 Configuration 2	219
MAC0INTE	0xC1	0xF	MAC0 Interrupt Enable	221
MAC0ITER	0xD7	0xF	Iteration Counter	231

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
MAC0OVF	0xD6	0xF	Accumulator Overflow Byte	226
MAC0STA	0xCF	0xF	MAC0 Status	213
OSCICL	0xAF	0x0	High Frequency Oscillator Calibration	250
OSCICN	0xB2	0x0	High Frequency Oscillator Control	251
OSCXCN	0xB1	0x0	External Oscillator Control	252
P0	0x80	All pages	Port 0 Pin Latch	289
P0DRV	0x99	0xF	Port 0 Drive Strength	293
P0MASK	0x8B	0xF	Port 0 Mask	287
P0MAT	0xF4	0xF	Port 0 Match	288
P0MDIN	0xEC	0xF	Port 0 Input Mode	290
P0MDOUT	0xD9	0xF	Port 0 Output Mode	291
P0SKIP	0xB6	0xF	Port 0 Skip	292
P1	0x90	All pages	Port 1 Pin Latch	296
P1DRV	0x9A	0xF	Port 1 Drive Strength	300
P1MASK	0x8C	0xF	Port 1 Mask	294
P1MAT	0xF5	0xF	Port 1 Match	295
P1MDIN	0xED	0xF	Port 1 Input Mode	297
P1MDOUT	0xDC	0xF	Port 1 Output Mode	298
P1SKIP	0xC6	0xF	Port 1 Skip	299
P2	0xA0	All pages	Port 2 Pin Latch	303
P2DRV	0x9B	0xF	Port 2 Drive Strength	307
P2MASK	0x84	0xF	Port 2 Mask	301
P2MAT	0x85	0xF	Port 2 Match	302
P2MDIN	0xEE	0xF	Port 2 Input Mode	304
P2MDOUT	0xDD	0xF	Port 2 Output Mode	305
P2SKIP	0xC7	0xF	Port 2 Skip	306
P3	0xE1	0x0	Port 3 Pin Latch	308
P3DRV	0x9C	0xF	Port 3 Drive Strength	311
P3MDIN	0xEF	0xF	Port 3 Input Mode	309

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
P3MDOUT	0xDF	0xF	Port 3 Output Mode	310
P4	0xE2	0x0	Port 4 Pin Latch	312
P4DRV	0xB9	0xF	Port 4 Drive Strength	315
P4MDIN	0xF1	0xF	Port 4 Input Mode	313
P4MDOUT	0xC3	0xF	Port 4 Output Mode	314
P5	0xE3	0x0	Port 5 Pin Latch	316
P5DRV	0x9D	0xF	Port 5 Drive Strength	319
P5MDIN	0xF2	0xF	Port 5 Input Mode	317
P5MDOUT	0xFF	0xF	Port 5 Output Mode	318
P6	0xE4	0x0	Port 6 Pin Latch	320
P6MDIN	0x97	0xF	Port 6 Input Mode	320
PCA0CN	0xD8	0x0	PCA Control	433
PCA0CPH0	0xFC	0x0	PCA Capture Module High Byte 0	440
PCA0CPH1	0xEA	0x0	PCA Capture Module High Byte 1	443
PCA0CPH2	0xEC	0x0	PCA Capture Module High Byte 2	446
PCA0CPL0	0xFB	0x0	PCA Capture Module Low Byte 0	439
PCA0CPL1	0xE9	0x0	PCA Capture Module Low Byte 1	442
PCA0CPL2	0xEB	0x0	PCA Capture Module Low Byte 2	445
PCA0CPM0	0xDA	0x0	PCA Capture/Compare Mode 0	438
PCA0CPM1	0xDB	0x0	PCA Capture/Compare Mode 1	441
PCA0CPM2	0xDC	0x0	PCA Capture/Compare Mode 1	444
PCA0H	0xFA	0x0	PCA Counter/Timer Low Byte	437
PCA0L	0xF9	0x0	PCA Counter/Timer High Byte	436
PCA0MD	0xD9	0x0	PCA Mode	434
PCA0PWM	0xDF	0x0	PCA PWM Configuration	435
PCLKEN	0xF6	0x0	Low Power Peripheral Clock Enable	248
PCON	0x87	All pages	Power Control	100
PMU0CF	0xB5	0x0	Power Management Unit Configuration	101
PMU0FL	0xCE	0x0	Power Management Unit Flag	102

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
PMU0MD	0xCF	0x0	Power Management Unit Mode	103
PSCTL	0x8F	All pages	Program Store Control	70
PSW	0xD0	All pages	Program Status Word	182
REF0CN	0xD1	0x0	Voltage Reference Control	127
REG0CN	0xC9	0x0	Voltage Regulator Control	93
REVID	0xE2	0xF	Revision Identification	78
RSTSRC	0xEF	0x0	Reset Source	326
RTC0ADR	0xAC	0x0	RTC Address	264
RTC0DAT	0xAD	0x0	RTC Data	265
SBUF0	0x99	0x0	UART0 Serial Port Data Buffer	388
SCON0	0x98	All pages	UART0 Serial Port Control	386
SFRLAST	0xB3	All pages	SFR Page Last	186
SFRNEXT	0xF3	All pages	SFR Page Next	185
SFRPAGE	0xA7	All pages	SFR Page	183
SFRPGCN	0xA6	0xF	SFR Page Control	184
SMB0ADM	0xF5	0x0	SMBus0 Slave Address Mask	362
SMB0ADR	0xF4	0x0	SMBus0 Slave Address	361
SMB0CF	0xC1	0x0	SMBus0 Configuration	358
SMB0CN	0xC0	0x0	SMBus0 Control	359
SMB0DAT	0xC2	0x0	SMBus0 Data	363
SP	0x81	All pages	Stack Pointer	179
SPI0CFG	0xA1	0x0	SPI0 Configuration	338
SPI0CKR	0xA2	0x0	SPI0 Clock Control	340
SPI0CN	0xF8	0x0	SPI0 Control	339
SPI0DAT	0xA3	0x0	SPI0 Data	341
TCON	0x88	0x0	Timer 0/1 Control	402
TH0	0x8C	0x0	Timer 0 High Byte	406
TH1	0x8D	0x0	Timer 1 High Byte	407
TL0	0x8A	0x0	Timer 0 Low Byte	404

Table 9.2. Special Function Registers (Continued)

Register	Address	SFR Page	Register Description	Page
TL1	0x8B	0x0	Timer 1 Low Byte	405
TMOD	0x89	0x0	Timer 0/1 Mode	403
TMR2CN	0xC8	All pages	Timer 2 Control	408
TMR2H	0xCD	0x0	Timer 2 High Byte	413
TMR2L	0xCC	0x0	Timer 2 Low Byte	412
TMR2RLH	0xCB	0x0	Timer 2 Reload High Byte	411
TMR2RLL	0xCA	0x0	Timer 2 Reload Low Byte	410
TMR3CN	0x91	0x0	Timer 3 Control	414
TMR3H	0x95	0x0	Timer 3 High Byte	419
TMR3L	0x94	0x0	Timer 3 Low Byte	418
TMR3RLH	0x93	0x0	Timer 3 Reload High Byte	417
TMR3RLL	0x92	0x0	Timer 3 Reload Low Byte	416
TOFFH	0x8E	0xF	Temperature Sensor Offset High	128
TOFFL	0x8D	0xF	Temperature Sensor Offset Low	129
VDM0CN	0xFF	0x0	Supply Monitor Control	327
XBR0	0x95	0xF	Port I/O Crossbar 0	285
XBR1	0x96	0xF	Port I/O Crossbar 1	286

10. Flash Memory

On-chip, re-programmable flash memory is included for program code and non-volatile data storage. The flash memory is organized in 512-byte pages. It can be erased and written through the C2 interface or from firmware by overloading the MOVX instruction. Any individual byte in flash memory must only be written once between page erase operations.

10.1. 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 in flash user space offers protection of the flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. See Section "8. Memory Organization" on page 54 for the location of the security byte. 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 unlocked when no other flash pages are locked (all bits of the Lock Byte are '1') and locked when any other flash pages are locked (any bit of the Lock Byte is '0').** An example is shown in Figure 10.1.

Security Lock Byte:	11111011b
1s Complement:	00000100b
Flash pages locked:	5 (First four flash pages + Lock Byte Page)

Figure 10.1. Security Byte Decoding

The level of flash security depends on the flash access method. The three flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 10.1 summarizes the flash security features of the C8051F97x devices.

Table 10.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware Executing from:	
		Unlocked Page	Locked Page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	Flash Error Reset	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	N/A
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	N/A
Read contents of Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	Permitted	N/A
Erase page containing Lock Byte—Unlock all pages (if any page is locked)	C2 Device Erase Only	Flash Error Reset	Flash Error Reset
Lock additional pages (change 1s to 0s in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Unlock individual pages (change 0s to 1s in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Read, Write or Erase Reserved Area	Not Permitted	Flash Error Reset	Flash Error Reset

Notes:
1. C2 Device Erase erases all flash pages including the page containing the Lock Byte.
2. Flash Error Reset is not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is “1” after reset).
3. All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).
4. Locking any flash page also locks the page containing the Lock Byte.
5. Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.
6. If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.

10.2. Programming the Flash Memory

Writes to flash memory clear bits from logic 1 to logic 0, and can be performed on single byte locations. Flash erasures set bits back to logic 1, and occur only on full pages. 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.

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.

To ensure the integrity of flash contents, it is strongly recommended that the on-chip supply monitor be enabled in any system that includes code that writes and/or erases flash memory from software.

10.2.1. 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.

10.2.2. 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) setting the PSWE Program Store Write Enable bit in the PSCTL register to logic 1 (this directs the MOVX writes to target flash memory); and (2) Writing the flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

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

1. Disable interrupts (recommended).
2. Set the PSEE bit (register PSCTL).
3. Set the PSWE bit (register PSCTL).
4. Write the first key code to FLKEY: 0xA5.
5. Write the second key code to FLKEY: 0xF1.
6. Using the MOVX instruction, write a data byte to any location within the page to be erased.
7. Clear the PSWE and PSEE bits.

10.2.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. Erase the flash page containing the target location, as described in Section 10.2.2.
3. Set the PSWE bit (register PSCTL).
4. Clear the PSEE bit (register PSCTL).
5. Write the first key code to FLKEY: 0xA5.
6. Write the second key code to FLKEY: 0xF1.
7. Using the MOVX instruction, write a single data byte to the desired location within the desired page.
8. Clear the PSWE bit.

Steps 5–7 must be repeated for each byte to be written. After flash writes are complete, PSWE should be cleared so that MOVX instructions do not target program memory.

10.3. 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.

10.4. Flash Write and Erase Guidelines

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

To help prevent the accidental modification of flash by firmware, hardware restricts flash writes and erasures when the supply monitor is not active and selected as a reset source. As the monitor is enabled and selected as a reset source by default, it is recommended that systems writing or erasing flash simply maintain the default state.

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

10.4.1. Voltage Supply Maintenance and the Supply Monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum supply rise time specification is met. If the system cannot meet this rise time specification, then add an external supply brownout circuit to the $\overline{\text{RST}}$ pin of the device that holds the device in reset until the voltage supply reaches the lower limit, and re-asserts $\overline{\text{RST}}$ if the supply drops below the low supply limit.
3. Do not disable the supply monitor. If the supply monitor must be disabled in the system, firmware should be added to the startup routine to enable the on-chip supply monitor and enable the supply monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the reset vector. For C-based systems, this may involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the supply monitor and enabling the supply monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash From Firmware", available from the Silicon Laboratories web site. **Note that the supply monitor must be enabled and enabled as a reset source when writing or erasing flash memory. A flash error reset will occur if either condition is not met.**
4. As an added precaution if the supply monitor is ever disabled, explicitly enable the supply monitor and enable the supply monitor as a reset source inside the functions that write and erase flash memory. The supply monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a '1'. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

10.4.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (in register PSCTL) is set to a 1. There should be exactly one routine in code that sets PSWE to a '1' to write flash bytes and one routine in code that sets PSWE and PSEE both to a '1' to erase flash pages.
8. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can

be found in “AN201: Writing to Flash From Firmware", available from the Silicon Laboratories web site.

9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to 0. Any interrupts posted during the flash write or erase operation will be serviced in priority order after the flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase flash memory to ensure that a routine called with an illegal address does not result in modification of the flash.

10.4.3. System Clock

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

Additional flash recommendations and example code can be found in “AN201: Writing to Flash From Firmware", available from the Silicon Laboratories website.

10.5. Flash Control Registers

Register 10.1. PSCTL: Program Store Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved						PSEE	PSWE
Type	R						RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0x8F								

Table 10.2. PSCTL Register Bit Descriptions

Bit	Name	Function
7:2	Reserved	Must write reset value.
1	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.
0	PSWE	**Program Store Write Enable.** Setting this bit allows writing a byte of data to the flash program memory using the MOVX write instruction. The 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.

Register 10.2. FLKEY: Flash Lock and Key

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

Table 10.3. FLKEY Register Bit Descriptions

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

Register 10.3. FLSCL: Flash Scale

Bit	7	6	5	4	3	2	1	0
Name	Reserved	BYPASS	Reserved					
Type	R	RW	R					
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xB6								

Table 10.4. FLSCL Register Bit Descriptions

Bit	Name	Function
7	Reserved	Must write reset value.
6	BYPASS	**Flash Read Timing One-Shot Bypass.** 0: The one-shot determines the flash read time. This setting should be used for operating frequencies less than 14 MHz. 1: The system clock determines the flash read time. This setting should be used for frequencies greater than 14 MHz.
5:0	Reserved	Must write reset value.

Note: Operations which clear the BYPASS bit do not need to be immediately followed by a benign 3-byte instruction. For code compatibility with C8051F930/31/20/21 devices, a benign 3-byte instruction whose third byte is a don't care should follow the clear operation. See the C8051F93x-C8051F92x data sheet for more details.

Register 10.4. FLWR: Flash Write Only

Bit	7	6	5	4	3	2	1	0
Name	FLWR							
Type	W							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xE5								

Table 10.5. FLWR Register Bit Descriptions

Bit	Name	Function
7:0	FLWR	**Flash Write Only.** All writes to this register have no effect on system operation.

11. On-Chip XRAM

The C8051F97x MCUs include on-chip RAM mapped into the external data memory space (XRAM). The external memory space may be accessed using the external move instruction (MOVX) with the target address specified in either the data pointer (DPTR), or with the target address low byte in R0 or R1. On C8051F97x devices, the target address high byte is a don't care.

When using the MOVX instruction to access on-chip RAM, no additional initialization is required and the MOVX instruction execution time is as specified in the CIP-51 chapter.

Important Note: MOVX write operations can be configured to target Flash memory, instead of XRAM. See Section "10. Flash Memory" on page 65 for more details. The MOVX instruction accesses XRAM by default.

11.1. 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.

11.1.1. 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.

11.1.2. 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
```

11.2. External Memory Interface Registers

Register 11.1. EMI0CN: External Memory Interface Control

Bit	7	6	5	4	3	2	1	0
Name	PGSEL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xB9								

Bit	Name	Function
7:0	PGSEL	**XRAM Page Select.** The XRAM Page Select field provides 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

f

12. Device Identification and Unique Identifier

The C8051F97x has SFRs that identify the device family, derivative, and revision. These SFRs can be read by firmware at runtime to determine the capabilities of the MCU that is executing code. This allows the same firmware image to run on MCUs with different memory sizes and peripherals, and dynamically change functionality to suit the capabilities of that MCU.

In addition to the device identification registers, a 128-bit unique identifier (UID) is preprogrammed into all devices. The UID resides in the last sixteen bytes of XRAM. The UID can be read by firmware using MOVX instructions and through the debug port.

Firmware can overwrite the UID during normal operation, and the bytes in memory will be automatically reinitialized with the UID value after any device reset. Firmware using this area of memory should always initialize the memory to a known value, as any previous data stored at these locations will be overwritten and not retained through a reset.

Table 12.1. UID Implementation Information

Device	External Memory (XRAM) Addresses
C8051F970 C8051F971 C8051F972	**(MSB)** 0x1FFF, 0x1FFE, 0x1FFD, 0x1FFC, 0x1FFB, 0x1FFA, 0x1FF9, 0x1FF8, 0x1FF7, 0x1FF6, 0x1FF5, 0x1FF4, 0x1FF3, 0x1FF2, 0x1FF1, 0x1FF0 **(LSB)**
C8051F973 C8051F974 C8051F975	**(MSB)** 0x0FFF, 0x0FFE, 0x0FFD, 0x0FFC, 0x0FFB, 0x0FFA, 0x0FF9, 0x0FF8, 0x0FF7, 0x0FF6, 0x0FF5, 0x0FF4, 0x0FF3, 0x0FF2, 0x0FF1, 0x0FF0 **(LSB)**

12.1. Device Identification Registers

Register 12.1. DEVICEID: Device Identification

Bit	7	6	5	4	3	2	1	0
Name	DEVICEID							
Type	R							
Reset	0	0	1	0	1	0	0	1
SFR Page = 0xF; SFR Address: 0xE1								

Table 12.2. DEVICEID Register Bit Descriptions

Bit	Name	Function
7:0	DEVICEID	**Device ID.** This read-only register returns the 8-bit device ID: 0x29 (C8051F97x).

Register 12.2. REVID: Revision Identifcation

Bit	7	6	5	4	3	2	1	0
Name	REVID							
Type	R							
Reset	X	X	X	X	X	X	X	X
SFR Page = 0xF; SFR Address: 0xE2								

Table 12.3. REVID Register Bit Descriptions

Bit	Name	Function
7:0	REVID	**Revision ID.** This read-only register returns the 8-bit revision ID. 00000000: Reserved. 00000001: Revision A 00000010-11111111: Reserved.

13. Interrupts

The C8051F97x includes an extended interrupt system supporting multiple interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

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

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE and EIE1). However, interrupts must first be globally enabled by setting the EA bit in the IE register 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.

13.1. MCU Interrupt Sources and Vectors

The C8051F97x MCUs support interrupt sources for each peripheral on the device. 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 13.1. 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).

13.1.1. 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 EIP1) 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 13.1.

13.1.2. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. 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 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 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. If more than one interrupt is pending when the CPU exits an ISR, the CPU will service the next highest priority interrupt that is pending.

Table 13.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit Addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0 (/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1 (/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART0	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
SMB0	0x003B	7	SI (SMB0CN.0)	Y	N	ESMB0 (EIE1.0)	PSMB0 (EIP1.0)
SmaRTClock Alarm	0x0043	8	ALRM (RTC0CN.2)*	N	N	ERTC0 (EIE1.1)	PRTC0 (EIP1.1)
ADC0 Window Comparator	0x004B	9	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.2)	PWADC0 (EIP1.2)
ADC0 End of Conversion	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 (EIE1.4)
DMA0 Midpoint Operation Complete	0x0063	12	CHn_MINT (DMA0MINT.n)	Y	N	EDMA0M (EIE1.5)	PDMA0M (EIP1.5)
DMA0 Endpoint Operation Complete	0x006B	13	CHn_INT (DMA0INT.n)	Y	N	EDMA0 (EIE1.6)	PDMA0 (EIP1.6)
Timer 3 Overflow	0x0073	14	TF3H (TMR3CN.7) TF3L (TMR3CN.6)	N	N	ET3 (EIE1.7)	PT3 (EIP1.7)
Port Match	0x0083	16	None			EMAT (EIE2.1)	PMAT (EIP2.1)

Table 13.1. Interrupt Summary (Continued)

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit Addressable?	Cleared by HW?	Enable Flag	Priority Control
SmaRTClock Oscillator Fail	0x008B	17	OSCFAIL (RTC0CN.5)*	N	N	ERTC0F (EIE2.2)	PRTC0F (EIP2.2)
MAC0	0x0093	18	MAC0INT (MAC0STA.6)	N	N	EMAC0 (EIE2.3)	PMAC0 (EIP2.3)
CS0 Conversion Complete	0x009B	19	CS0INT (CS0CN.5)	Y	N	ECSCPT (EIE2.4)	PCSCPT (EIP2.4)
CS0 Digital Comparator	0x00A3	20	CS0CMPF (CS0CN.0)	Y	N	ECSDC (EIE2.5)	PCSDC (EIP2.5)
CS0 End of Scan	0x00AB	21	CS0EOS (CS0CN.6)	Y	N	ECSEOS (EIE2.6)	PCSEOS (EIP2.6)
I2C Slave 0	0x00B3	22	I2C0INT (I2C0STAT.5)	Y	N	EI2C0 (EIE2.7)	PI2C0 (EIP2.7)

***Note:** Indicates a register located in an indirect memory space.

13.2. Interrupt Control Registers

Register 13.1. IE: Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xA8 (bit-addressable)								

Table 13.2. IE Register Bit Descriptions

Bit	Name	Function
7	EA	**All Interrupts Enable.** Globally enables/disables all interrupts and overrides individual interrupt mask settings. 0: Disable all interrupt sources. 1: Enable each interrupt according to its individual mask setting.
6	ESPI0	**SPI0 Interrupt Enable.** This bit sets the masking of the SPI0 interrupts. 0: Disable all SPI0 interrupts. 1: Enable interrupt requests generated by SPI0.
5	ET2	**Timer 2 Interrupt Enable.** 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.
4	ES0	**UART0 Interrupt Enable.** This bit sets the masking of the UART0 interrupt. 0: Disable UART0 interrupt. 1: Enable UART0 interrupt.
3	ET1	**Timer 1 Interrupt Enable.** 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.
2	EX1	**External Interrupt 1 Enable.** 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.
1	ET0	**Timer 0 Interrupt Enable.** 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.

Table 13.2. IE Register Bit Descriptions

Bit	Name	Function
0	EX0	**External Interrupt 0 Enable.** 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.

Register 13.2. IP: Interrupt Priority

Bit	7	6	5	4	3	2	1	0
Name	Reserved	PSPI0	PT2	PS0	PT1	PX1	PT0	PX0
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	1	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xB8 (bit-addressable)								

Table 13.3. IP Register Bit Descriptions

Bit	Name	Function
7	Reserved	Must write reset value.
6	PSPI0	**Serial Peripheral Interface (SPI0) Interrupt Priority Control.** This bit sets the priority of the SPI0 interrupt. 0: SPI0 interrupt set to low priority level. 1: SPI0 interrupt set to high priority level.
5	PT2	**Timer 2 Interrupt Priority Control.** This bit sets the priority of the Timer 2 interrupt. 0: Timer 2 interrupt set to low priority level. 1: Timer 2 interrupt set to high priority level.
4	PS0	**UART0 Interrupt Priority Control.** This bit sets the priority of the UART0 interrupt. 0: UART0 interrupt set to low priority level. 1: UART0 interrupt set to high priority level.
3	PT1	**Timer 1 Interrupt Priority Control.** This bit sets the priority of the Timer 1 interrupt. 0: Timer 1 interrupt set to low priority level. 1: Timer 1 interrupt set to high priority level.
2	PX1	**External Interrupt 1 Priority Control.** This bit sets the priority of the External Interrupt 1 interrupt. 0: External Interrupt 1 set to low priority level. 1: External Interrupt 1 set to high priority level.
1	PT0	**Timer 0 Interrupt Priority Control.** This bit sets the priority of the Timer 0 interrupt. 0: Timer 0 interrupt set to low priority level. 1: Timer 0 interrupt set to high priority level.
0	PX0	**External Interrupt 0 Priority Control.** This bit sets the priority of the External Interrupt 0 interrupt. 0: External Interrupt 0 set to low priority level. 1: External Interrupt 0 set to high priority level.

Register 13.3. EIE1: Extended Interrupt Enable 1

Bit	7	6	5	4	3	2	1	0
Name	ET3	EDMA0	EDMA0M	EPCA0	EADC0	EWADC0	ERTC0A	ESMB0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xE6								

Table 13.4. EIE1 Register Bit Descriptions

Bit	Name	Function
7	ET3	**Timer 3 Interrupt Enable.** 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.
6	EDMA0	**DMA0 Interrupt Enable.** This bit sets the masking of the DMA0 Interrupt. 0: Disable DMA0 interrupts. 1: Enable interrupt requests generated by DMA0.
5	EDMA0M	**DMA0 Mid-Point Interrupt Enable.** This bit sets the masking of the DMA0 Mid-Point interrupt. 0: Disable DMA0M interrupts. 1: Enable interrupt requests generated by the DMA0M flags.
4	EPCA0	**Programmable Counter Array (PCA0) Interrupt Enable.** This bit sets the masking of the PCA0 interrupts. 0: Disable all PCA0 interrupts. 1: Enable interrupt requests generated by PCA0.
3	EADC0	**ADC0 Conversion Complete Interrupt Enable.** This bit sets the masking of the ADC0 Conversion Complete interrupt. 0: Disable ADC0 Conversion Complete interrupt. 1: Enable interrupt requests generated by the ADINT flag.
2	EWADC0	**Window Comparison ADC0 Interrupt Enable.** 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 (ADWINT).
1	ERTC0A	**RTC Alarm Interrupts Enable.** This bit sets the masking of the RTC Alarm interrupt. 0: Disable RTC Alarm interrupts. 1: Enable interrupt requests generated by a RTC Alarm.

Table 13.4. EIE1 Register Bit Descriptions

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

Register 13.4. EIP1: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PT3	PDMA0	PDMA0M	PPCA0	PADC0	PWADC0	PRTC0A	PSMB0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xF6								

Table 13.5. EIP1 Register Bit Descriptions

Bit	Name	Function
7	PT3	**Timer 3 Interrupt Priority Control.** This bit sets the priority of the Timer 3 interrupt. 0: Timer 3 interrupts set to low priority level. 1: Timer 3 interrupts set to high priority level.
6	PDMA0	**DMA0 Interrupt Priority Control.** This bit sets the priority of the DMA0 interrupt. 0: DMA0 interrupts set to low priority level. 1: DMA0 interrupts set to high priority level.
5	PDMA0M	**DMA0 Mid-Point Interrupt Priority Control.** This bit sets the masking of the DMA0 Mid-Point interrupt. 0: Mid-point DMA0 interrupts set to low priority level. 1: Mid-point DMA0 interrupts set to high priority level.
4	PPCA0	**Programmable Counter Array (PCA0) Interrupt Priority Control.** This bit sets the priority of the PCA0 interrupt. 0: PCA0 interrupt set to low priority level. 1: PCA0 interrupt set to high priority level.
3	PADC0	**ADC0 Conversion Complete Interrupt Priority Control.** This bit sets the priority of the ADC0 Conversion Complete interrupt. 0: ADC0 Conversion Complete interrupt set to low priority level. 1: ADC0 Conversion Complete interrupt set to high priority level.
2	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.
1	PRTC0A	**RTC Alarm Interrupt Priority Control.** This bit sets the priority of the RTC Alarm interrupt. 0: RTC Alarm interrupt set to low priority level. 1: RTC Alarm interrupt set to high priority level.

Table 13.5. EIP1 Register Bit Descriptions

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

Register 13.5. EIE2: Extended Interrupt Enable 2

Bit	7	6	5	4	3	2	1	0
Name	EI2C0	ECSEOS	ECSDC	ECSCPT	EMAC0	ERTC0F	EMAT	Reserved
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xE7								

Table 13.6. EIE2 Register Bit Descriptions

Bit	Name	Function
7	EI2C0	**I2C0 Slave Interrupt Enable.** 0: Disable I2C0 Slave event interrupt. 1: Enable interrupt requests generated by the I2C0 Slave.
6	ECSEOS	**Capacitive Sense End of Scan Interrupt Enable.** This bit sets the masking of the Capacitive Sense End of Scan interrupt. 0: Disable Capacitive Sense End of Scan interrupt. 1: Enable interrupt requests generated by the Capacitive Sense End of Scan.
5	ECSDC	**Capacitive Sense Digital Comparator Interrupt Enable.** This bit sets the masking of the Capacitive Sense Digital Comparator interrupt. 0: Disable Capacitive Sense Digital Comparator interrupt. 1: Enable interrupt requests generated by the Capacitive Sense Digital Comparator.
4	ECSCPT	**Capacitive Sense Conversion Complete Interrupt Enable.** This bit sets the masking of the Capacitive Sense Conversion Complete interrupt. 0: Disable Capacitive Sense Conversion Complete interrupt. 1: Enable interrupt requests generated by CS0INT.
3	EMAC0	**MAC0 Interrupt Enable.** 0: Disable MAC0 event interrupt. 1: Enable interrupt requests generated by MAC0.
2	ERTC0F	**RTC Oscillator Fail Interrupt Enable.** This bit sets the masking of the RTC Alarm interrupt. 0: Disable RTC Alarm interrupts. 1: Enable interrupt requests generated by the RTC Alarm.
1	EMAT	**Port Match Interrupts Enable.** This bit sets the masking of the Port Match Event interrupt. 0: Disable all Port Match interrupts. 1: Enable interrupt requests generated by a Port Match.
0	Reserved	Must write reset value.

Register 13.6. EIP2: Extended Interrupt Priority 1

Bit	7	6	5	4	3	2	1	0
Name	PI2C0	PCSEOS	PCSDC	PCSCPT	PMAC0	PRTC0F	PMAT	Reserved
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xF7								

Table 13.7. EIP2 Register Bit Descriptions

Bit	Name	Function
7	PI2C0	**I2C0 Slave Interrupt Priority Control.** 0: I2C0 Slave interrupts set to low priority level. 1: I2C0 Slave interrupts set to high priority level.
6	PCSEOS	**Capacitive Sense End of Scan Interrupt Priority Control.** This bit sets the priority of the Capacitive Sense End of Scan interrupt. 0: Capacitive Sense End of Scan interrupt set to low priority level. 1: Capacitive Sense End of Scan interrupt set to high priority level.
5	PCSDC	**Capacitive Sense Digital Comparator Interrupt Priority Control.** This bit sets the priority of the Capacitive Sense Digital Comparator interrupt. 0: Capacitive Sense Digital Comparator interrupt set to low priority level. 1: Capacitive Sense Digital Comparator interrupt set to high priority level.
4	PCSCPT	**Capacitive Sense Conversion Complete Interrupt Priority Control.** This bit sets the priority of the Capacitive Sense Conversion Complete interrupt. 0: Capacitive Sense Conversion Complete interrupt set to low priority level. 1: Capacitive Sense Conversion Complete interrupt set to high priority level.
3	PMAC0	**MAC0 Interrupt Priority Control.** 0: MAC0 interrupts set to low priority level. 1: MAC0 interrupts set to high priority level.
2	PRTC0F	**RTC Oscillator Fail Interrupt Priority Control.** This bit sets the priority of the RTC Alarm interrupt. 0: RTC Alarm interrupt set to low priority level. 1: RTC Alarm interrupt set to high priority level.
1	PMAT	**Port Match Interrupt Priority Control.** This bit sets the priority of the Port Match Event interrupt. 0: Port Match interrupt set to low priority level. 1: Port Match interrupt set to high priority level.
0	Reserved	Must write reset value.

14. External Interrupts ($\overline{INT0}$ and $\overline{INT1}$)

The C8051F97x device family includes two external digital interrupt sources ($\overline{INT0}$ and $\overline{INT1}$), with dedicated interrupt sources (up to 16 additional I/O interrupts are available through the port match function). As is the case on a standard 8051 architecture, certain controls for these two interrupt sources are available in the Timer0/1 registers. Extensions to these controls which provide additional functionality on C8051F97x devices are available in the IT01CF register. $\overline{INT0}$ and $\overline{INT1}$ are configurable as active high or low, edge or level sensitive. The IN0PL and IN1PL bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{INT0}$ 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	$\overline{INT1}$ 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

$\overline{INT0}$ and $\overline{INT1}$ are assigned to port pins as defined in the IT01CF register. Note that $\overline{INT0}$ and $\overline{INT1}$ Port pin assignments are independent of any crossbar assignments. $\overline{INT0}$ and $\overline{INT1}$ 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 $\overline{INT0}$ and/or $\overline{INT1}$, configure the crossbar to skip the selected pin(s).

IE0 and IE1 in the TCON register serve as the interrupt-pending flags for the $\overline{INT0}$ and $\overline{INT1}$ external interrupts, respectively. If an $\overline{INT0}$ or $\overline{INT1}$ 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.

14.1. External Interrupt Control Registers

Register 14.1. IT01CF: INT0/INT1 Configuration

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL			IN0PL	IN0SL		
Type	RW	RW			RW	RW		
Reset	0	0	0	0	0	0	0	1
SFR Page = 0xF; SFR Address: 0xDE								

Table 14.1. IT01CF Register Bit Descriptions

Bit	Name	Function
7	IN1PL	**INT1 Polarity.** 0: INT1 input is active low. 1: INT1 input is active high.
6:4	IN1SL	**INT1 Port Pin Selection.** These bits select which Port pin is assigned to INT1. This pin assignment is independent of the Crossbar; INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0. 001: Select P0.1. 010: Select P0.2. 011: Select P0.3. 100: Select P0.4. 101: Select P0.5. 110: Select P0.6. 111: Select P0.7.
3	IN0PL	**INT0 Polarity.** 0: INT0 input is active low. 1: INT0 input is active high.
2:0	IN0SL	**INT0 Port Pin Selection.** These bits select which Port pin is assigned to INT0. This pin assignment is independent of the Crossbar; INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin. 000: Select P0.0. 001: Select P0.1. 010: Select P0.2. 011: Select P0.3. 100: Select P0.4. 101: Select P0.5. 110: Select P0.6. 111: Select P0.7.

15. Voltage Regulator (VREG0)

C8051F97x devices include an internal voltage regulator (VREG0) to regulate the internal core supply to 1.8 V from a VDD supply of 1.8 to 3.6 V. Electrical characteristics for the on-chip regulator are specified in the Electrical Specifications chapter.

The REG0CN register allows the Precision Oscillator Bias to be disabled, reducing supply current in all non-Sleep power modes. This bias should only be disabled when the precision oscillator is not being used.

The internal regulator (VREG0) is disabled when the device enters Sleep Mode and remains enabled when the device enters Suspend Mode. See Section “16. Power Management” on page 94 for complete details about low power modes.

15.1. Voltage Regulator Control Registers

Register 15.1. REG0CN: Voltage Regulator Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved			OSCBIAS	Reserved			
Type	R			RW	R			
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xC9								

Bit	Name	Function
7:5	Reserved	Must write reset value.
4	OSCBIAS	**High Frequency Oscillator Bias.** When set to 1, the bias used by the precision High Frequency Oscillator is forced on. If the precision oscillator is not being used, this bit may be cleared to 0 to reduce supply current in all non-Sleep power modes.
3:0	Reserved	Must write reset value.

16. Power Management

C8051F97x devices support seven power modes: normal active, low power active, idle, low power idle, stop, suspend, and sleep. The power management unit (PMU0) allows the device to enter and wake-up from the available power modes. A brief description of each power mode is provided in Table 16.1. Detailed descriptions of each mode can be found in the following sections.

Table 16.1. Power Modes

Power Mode	Description	Wake-up Sources	Power and Performance
Normal Active	Device fully functional.	N/A	Excellent MIPS/mW
Low Power Active	Device fully functional except peripherals whose clocks are intentionally disabled.	N/A	Excellent. Clocks only enabled for peripherals that request for it.
Idle	All peripherals fully functional. Wake-up in 2 clock cycles.	Any Interrupt	Good No Code Execution
Low Power Idle	Similar to Idle mode, the CPU is halted. Clocks of unused peripherals can be intentionally gated. Wake-up in 2 clock cycles.	Any Interrupt	Very Good No Code Execution. Clocks only enabled for peripherals that request for it.
Stop	Legacy 8051 low power mode. A reset is required to wake up.	Any Reset	Good No Code Execution Precision Oscillator Disabled
Suspend	Similar to Stop mode, but very fast wake-up time and code resumes execution at the next instruction.	CS0 comparator threshold event, SmaRTClock, Port Match, I^2C Slave, $\overline{RST}$ pin	Very Good No Code Execution All Internal Oscillators Disabled System Clock Gated
Sleep	Ultra Low Power and flexible wake-up sources. Code resumes execution at the next instruction.	SmaRTClock, Port Match, I^2C Slave, $\overline{RST}$ pin	Excellent Power Supply Gated All Oscillators except SmaRTClock Disabled

In battery powered systems, the system should spend as much time as possible in sleep mode in order to preserve battery life. When a task with a fixed number of clock cycles needs to be performed, the device should switch to normal active mode or low power Active mode, finish the task as quickly as possible, and return to sleep mode. Idle mode, low power idle mode and suspend mode provide a very fast wake-up time; however, the power savings in these modes will not be as much as in sleep mode. Stop mode is included for legacy reasons; the system will be more power efficient and easier to wake up when idle, suspend, or sleep modes are used.

Although switching power modes is an integral part of power management, clock gating and enabling/disabling individual peripherals as needed will help lower power consumption in all power modes. Each analog peripheral can be disabled when not in use or placed in a low power mode. Digital peripherals such as timers or serial buses draw little power whenever they are not in use. Digital peripherals draw no power in sleep mode.

In low power modes (low power active mode and low power idle mode) there are two mechanisms to provide clocks to peripherals:

- CPU turns on the clocks of peripherals through SFR registers. Examples of such peripherals are Timers, PCA0, ADC0, etc.
- Peripherals with clock request capability can request for the clocks when needed. Examples of such peripherals are DMA0, I2C Slave, etc.

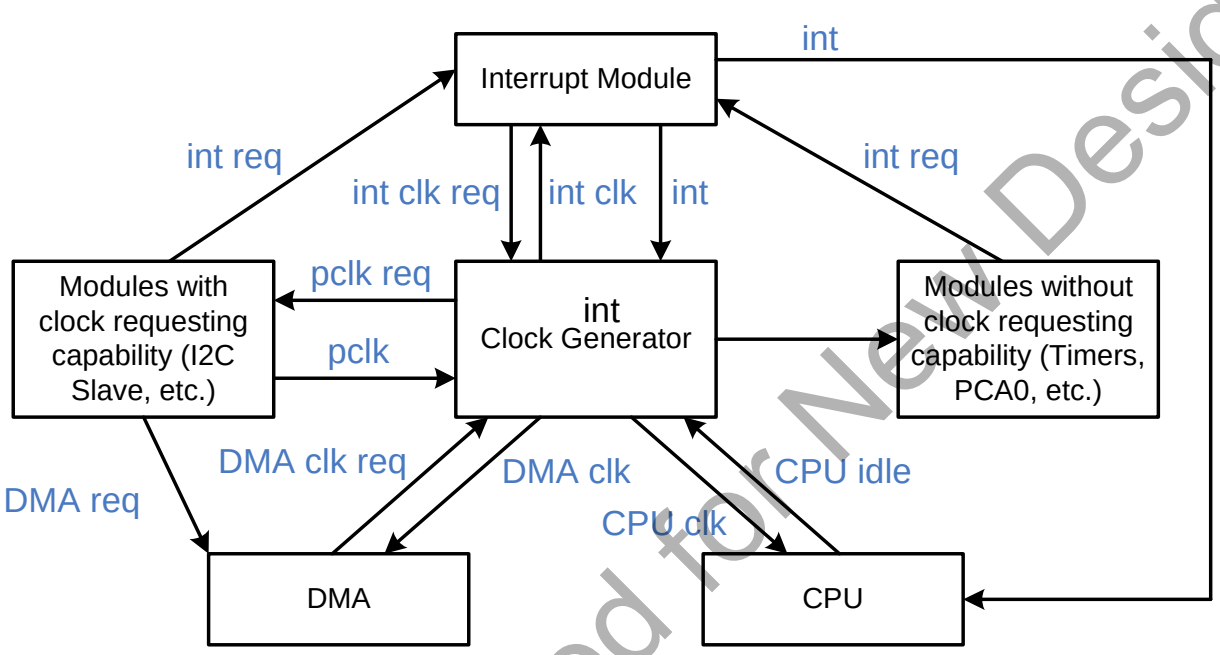


Figure 16.1. C8051F97x Module Clocking in Low Power Modes

Modules that are capable of requesting for clocks can do so at any time. These modules will use the requested clock to synchronize with outgoing DMA request or interrupt request. On the other hand, modules with no capability of requesting for clocks will always receive clock from clock generation module if their clock sources are enabled through t he PCLKEN register.

16.1. Normal Active Mode

The MCU is fully functional in normal active mode. Figure 16.2shows the on-chip power distribution to various peripherals. There are two supply voltages powering various sections of the chip: V_{DD} and the 1.8 V internal core supply. All analog peripherals are directly powered from the V_{DD} pin. All digital peripherals and the CIP-51 core are powered from the 1.8 V internal core supply. RAM, PMU0 and the SmaRTClock are always powered directly from the V_{DD} pin in sleep mode and powered from the core supply in all other power modes.

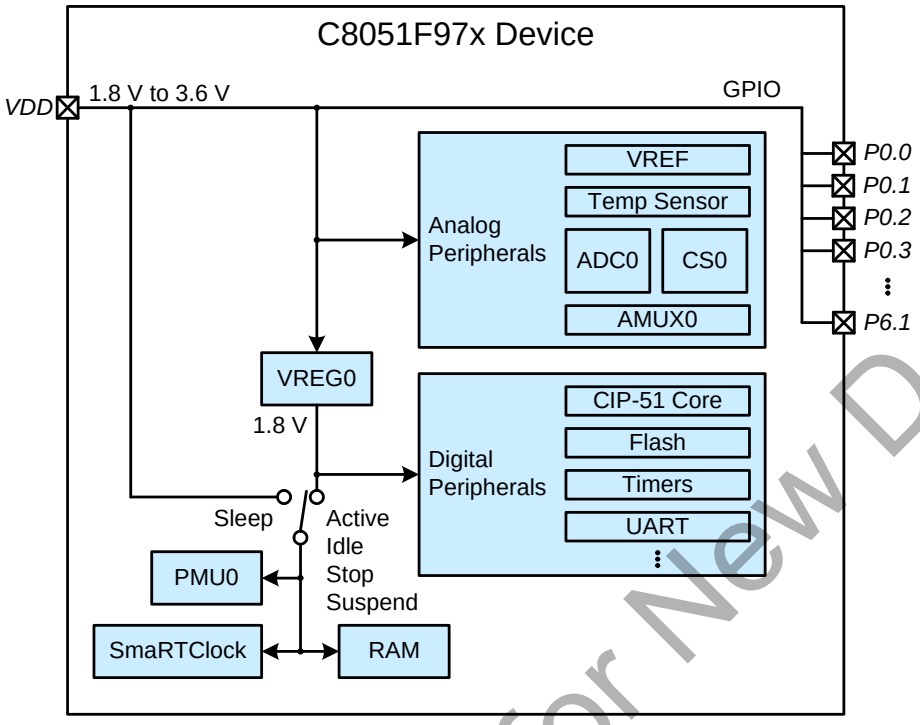


Figure 16.2. C8051F97x Power Distribution

16.2. 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.

Note: To ensure the MCU enters a low power state upon entry into Idle mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6).

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 "27.6. PCA Watchdog Timer Reset" on page 325 for more information on the use and configuration of the WDT.

16.3. 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 precision internal oscillator and CPU are stopped; the state of the low power oscillator and 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.

Stop mode is a legacy 8051 power mode; it will not result in optimal power savings. Sleep or suspend mode will provide more power savings if the MCU needs to be inactive for a long period of time.

Note: To ensure the MCU enters a low power state upon entry into stop mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6).

16.4. Suspend Mode

Setting the Suspend Mode Select bit (PMU0CF.6) causes the system clock to be gated off and all internal oscillators disabled. All digital logic (timers, communication peripherals, interrupts, CPU, etc.) stops functioning until one of the enabled wake-up sources occurs.

The following wake-up sources can be configured to wake the device from suspend mode:

- SmaRTClock oscillator fail
- SmaRTClock alarm
- Port Match event
- I2C0 address match
- CS0 comparator threshold event

Note: Upon wake-up from suspend mode, PMU0 requires two system clocks in order to update the PMU0CF wake-up flags. All flags will read back a value of '0' during the first two system clocks following a wake-up from suspend mode. The state of the wake-up source's interrupt indicator bit is not valid until 6 clock cycles after the device returns from suspend mode. If firmware needs to check a wake-up source's interrupt flag, firmware should insert instructions to wait 6 clock cycles between the call to enter suspend mode and the instruction that polls the interrupt flag.

In addition, a noise glitch on $\overline{\text{RST}}$ that is not long enough to reset the device will cause the device to exit suspend. In order for the MCU to respond to the pin reset event, software must not place the device back into suspend mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the $\overline{\text{RST}}$ pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into suspend mode. A 4.7 kΩ pullup resistor to V_{DD} is recommend for $\overline{\text{RST}}$ to prevent noise glitches from waking the device.

16.5. Sleep Mode

Setting the Sleep Mode Select bit (PMU0CF.6) turns off the internal 1.8 V regulator (VREG0) and switches the power supply of all on-chip RAM to the V_{DD} pin (see Figure 16.2). Power to most digital logic on the chip is disconnected; only PMU0 and the SmaRTClock remain powered. All analog peripherals (ADC0, External Oscillator, etc.) should be disabled prior to entering sleep mode.

GPIO pins configured as digital outputs will retain their output state during sleep mode. They will maintain the same current drive capability in sleep mode as they have in normal active mode.

GPIO pins configured as digital inputs can be used during sleep mode as wakeup sources using the port match feature. They will maintain the same input level specifications in sleep mode as they have in normal active mode.

C8051F97x devices support a wakeup request for external devices. Upon exit from sleep mode, the wake-up request signal is driven low, allowing other devices in the system to wake up from their low power modes. The wakeup request signal is low when the MCU is awake and high when the MCU is asleep.

RAM and SFR register contents are preserved in sleep mode as long as the voltage on V_{DD} does not fall below VPOR. The PC counter and all other volatile state information is preserved allowing the device to resume code execution upon waking up from sleep mode. The following wake-up sources can be configured to wake the device from sleep mode:

- SmaRTClock Oscillator Fail
- SmaRTClock Alarm
- Port Match Event
- I2C0 Address Match

The V_{DD} supply monitor can be disabled to save power by writing 1 to the MONDIS (PMU0MD.5) bit. When the V_{DD} supply monitor is disabled, all reset sources will trigger a full POR and will re-enable the V_{DD} supply monitor.

In addition, any falling edge on $\overline{\text{RST}}$ (due to a pin reset or a noise glitch) will cause the device to exit sleep mode. In order for the MCU to respond to the pin reset event, software must not place the device back into sleep mode for a period of 15 µs. The PMU0CF register may be checked to determine if the wake-up was due to a falling edge on the $\overline{\text{RST}}$ pin. If the wake-up source is not due to a falling edge on $\overline{\text{RST}}$, there is no time restriction on how soon software may place the device back into sleep mode. A 4.7 kΩ pullup resistor to V_{DD} is recommend for $\overline{\text{RST}}$ to prevent noise glitches from waking the device.

Note: Entering Sleep mode may cause the MCU to disconnect from the debug adapter while debugging.

16.6. Low Power Active Mode

Running in normal active mode can waste a significant amount of amount of power by clocking unused peripherals. Low power active mode in C8051F97x devices allows control of clocking activity in the clock tree, which enables firmware to shut off clocking to unused peripherals and save power.

Setting bit 1 and 2 of CLKMODE register causes the CIP-51 to enter low power active mode as soon as the instruction that sets the bits completes execution. CPU, all the analog and digital peripherals remain active except those whose clocks are turned off by user. **The CPU will not be able to access SFR of peripherals on inactive branches of the clock tree**. Refer to Figure 16.1 for more information on how modules receive or request for clocks in low power modes. **It is important that firmware configures bit 4 to 7 of PCLKEN register on page 248 to ensure desired clock gating of peripherals during low power active mode**. For example, if UART is supposed to be active during low power active mode, according to the PCLKEN register description bit 4 should be set. However, that is not sufficient in this case because Timer 1 is needed for UART Baud rate generation. As a consequence, bit 7 should be set as well for proper UART operation.

Low power active mode is terminated when the CLKMODE register is programmed to 0x00 or a reset occurs. Systems that use all the peripherals and always stay in the active mode may not find improvement in power consumption in the low power active mode due overhead logic required for this implementation.

16.7. Low Power Idle Mode

In this mode of operation, the CPU is halted and clocks supplied to unused peripherals can be shut down to save power. Clocks of these peripherals can be enabled or disabled by programming bits 0 to 3 of PCLKEN register accordingly on page 248. The device should be put to low power mode by setting bits 1 and 2 of CLKMODE register. The last step necessary to put device in low power idle mode is to set the Idle Mode Select bit (PCON.0) to 1. The CPU will be halted and the device will enter low-power idle mode as soon as the instruction that sets the Idle Mode Select bit completes execution. As a result, configuration of peripherals clock gating through PCLKEN register and CLKMODE register should be properly prepared before the Idle Mode Select bit in PCON is set. All internal registers and memory maintain their original data. All the analog and digital peripherals remain active except those whose clocks are turned off by user. Modules that are capable of requesting for clocks can do so at any time. Refer to Figure 16.1 for more information on how modules receive or request for clocks in low power modes.

Note: To ensure the MCU enters a low power state upon entry into Low Power Idle mode, the one-shot circuit should be enabled by clearing the BYPASS bit (FLSCL.6).

Low power 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. But the interrupt only causes the device to switch from low power idle mode to low power active mode. To return to normal active mode, the CLKMODE register should be reset to 0x00. 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 low power idle mode is terminated by an internal or external reset, the CIP-51 performs a normal 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 low power 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 low power 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 low power idle mode indefinitely, waiting for an external stimulus to wake up the system. Refer to "27.6. PCA Watchdog Timer Reset" on page 325 for more information on the use and configuration of the WDT.

16.8. Configuring Wakeup Sources

Before placing the device in a low power mode, one or more wakeup sources should be enabled so that the device does not remain in the low power mode indefinitely. For idle and low power idle modes, this includes enabling any interrupt. For stop mode, this includes enabling any reset source or relying on the $\overline{RST}$ pin to reset the device.

No wake-up source is needed for low power active mode. Firmware needs to set the CLKMODE register to 0x00 to return to normal active mode

Wake-up sources for suspend and sleep modes are configured through the PMU0CF register. Wake-up sources are enabled by writing 1 to the corresponding wake-up source enable bit. Wake-up sources must be re-enabled each time the device is placed in suspend or sleep mode, in the same write that places the device in the low power mode.

The reset pin is always enabled as a wake-up source. On the falling edge of $\overline{RST}$ the device will be awaken from sleep mode. The device must remain awake for more than 15 µs in order for the reset to take place.

16.9. Determining the Event that Caused the Last Wakeup

When waking from Idle or Low Power Idle mode, the CPU will vector to the interrupt which caused it to wake up. When waking from stop mode, the RSTSRC register may be read to determine the cause of the last reset.

Upon exit from suspend or sleep mode, the wake-up flags in the PMU0CF and PMU0FL registers can be read to determine the event which caused the device to wake up. After waking up, the wakeup flags will continue to be updated if any of the wake-up events occur. Wakeup flags are always updated, even if they are not enabled as wake-up sources.

All wakeup flags enabled as wakeup sources in PMU0CF and PMU0FL must be cleared before the device can enter suspend or sleep mode. After clearing the wakeup flags, each of the enabled wakeup events should be checked in the individual peripherals to ensure that a wakeup event did not occur while the wakeup flags were being cleared.

16.10. Power Control Registers

Register 16.1. PCON: Power Control

Bit	7	6	5	4	3	2	1	0
Name	GF5	GF4	GF3	GF2	GF1	GF0	STOP	IDLE
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0x87								

Bit	Name	Function
7	GF5	**General Purpose Flag 5.** This flag is a general purpose flag for use under firmware control.
6	GF4	**General Purpose Flag 4.** This flag is a general purpose flag for use under firmware control.
5	GF3	**General Purpose Flag 3.** This flag is a general purpose flag for use under firmware control.
4	GF2	**General Purpose Flag 2.** This flag is a general purpose flag for use under firmware control.
3	GF1	**General Purpose Flag 1.** This flag is a general purpose flag for use under firmware control.
2	GF0	**General Purpose Flag 0.** This flag is a general purpose flag for use under firmware control.
1	STOP	**Stop Mode Select.** Setting this bit will place the CIP-51 in Stop mode. This bit will always be read as 0.
0	IDLE	**Idle Mode Select.** Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0.

Register 16.2. PMU0CF: Power Management Unit Configuration

Bit	7	6	5	4	3	2	1	0
Name	SLEEP	SUSPEND	CLEAR	RSTWK	RTCFWK	RTCAWK	PMATWK	Reserved
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xB5								

Bit	Name	Function
7	SLEEP	**Sleep Mode Select.** Writing a 1 to this bit places the device in Sleep mode.
6	SUSPEND	**Suspend Mode Select.** Writing a 1 to this bit places the device in Suspend mode.
5	CLEAR	**Wake-up Flag Clear.** Writing a 1 to this bit clears all wake-up flags.
4	RSTWK	**Reset Pin Wake-up Flag.** This bit is set to 1 if a glitch has been detected on $\overline{\text{RST}}$.
3	RTCFWK	**RTC Oscillator Fail Wake-up Source Enable and Flag.** Read: Hardware sets this bit to 1 if the RTC oscillator failed. Write: Write this bit to 1 to enable wake-up on an RTC oscillator failure.
2	RTCAWK	**RTC Alarm Wake-up Source Enable and Flag.** Read: Hardware sets this bit to 1 if the RTC Alarm occured. Write: Write this bit to 1 to enable wake-up on an RTC Alarm.
1	PMATWK	**Port Match Wake-up Source Enable and Flag.** Read: Hardware sets this bit to 1 if Port Match event occured. Write: Write this bit to 1 to enable wake-up on a Port Match event.
0	Reserved	Must write reset value.

Notes:

1. Read-modify-write operations (ORL, ANL, etc.) should not be used on this register. Wake-up sources must be re-enabled each time the SLEEP or SUSPEND bits are written to 1.
2. The Low Power Internal Oscillator cannot be disabled and the MCU cannot be placed in Suspend or Sleep Mode if any wake-up flags are set to 1. Software should clear all wake-up sources after each reset and after each wake-up from Suspend or Sleep Modes.
3. PMU0 requires two system clocks to update the wake-up source flags after waking from Suspend mode. The wake-up source flags will read 0 during the first two system clocks following the wake from Suspend mode.

Register 16.3. PMU0FL: Power Management Unit Flag

Bit	7	6	5	4	3	2	1	0
Name	Reserved						I2CWK	CS0WK
Type	RW						RW	RW
Reset	0	0	0	0	0	0	X	0
SFR Page = 0x0; SFR Address: 0xCE								

Bit	Name	Function
7:2	Reserved	Must write reset value.
1	I2CWK	**I2C0 Slave Wake-up Source Enable and Flag.** Read: Hardware sets this bit to 1 if an I2C0 Slave event occured. Write: Write this bit to 1 to enable wake-up on an I2C0 Slave event.
0	CS0WK	**CS0 Wake-up Source Enable and Flag.** Read: Hardware sets this bit to 1 if a Capacitive Sensing comparator threshold event occured. Write: Write this bit to 1 to enable wake-up on a Capacitive Sensing comparator threshold event.

Notes:

1. The Low Power Internal Oscillator cannot be disabled and the MCU cannot be placed in Suspend or Sleep mode if any wake-up flags are set to 1. Software should clear all wake-up sources after each reset and after each wake-up from Suspend or Sleep modes.
2. PMU0 requires two system clocks to update the wake-up source flags after waking from Suspend mode. The wake-up source flags will read 0 during the first two system clocks following the wake from Suspend mode.

Register 16.4. PMU0MD: Power Management Unit Mode

Bit	7	6	5	4	3	2	1	0
Name	RTCOE	WAKEOE	Reserved					
Type	RW	RW	RW					
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xCF								

Bit	Name	Function
7	RTCOE	**Buffered RTC Output Enable.** Enables the buffered RTC oscillator output on P0.2. 0: Disable the buffered RTC output. 1: Enable the buffered RTC output.
6	WAKEOE	**Wakeup Request Output Enable.** Enables the Sleep Mode wake-up request signal on P0.3. 0: Disable the wake-up request signal. 1: Enable the wake-up request signal.
5:0	Reserved	Must write reset value.

17. Analog-to-Digital Converter (ADC0)

The ADC0 module on C8051F97x devices is a 300 ksps, 10-bit successive-approximation-register (SAR) ADC with integrated track-and-hold and programmable window detector. ADC0 also has an autonomous low power Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. It also has a 16-bit accumulator that can automatically oversample and average the ADC results.

The ADC is fully configurable under software control via Special Function Registers. The ADC0 operates in single-ended mode and may be configured to measure various different signals using the analog multiplexer described in “19. Analog Multiplexer (AMUX0)” on page 157. The ADC includes a temperature sensor described in “17.8. Temperature Sensor” on page 113, and the voltage reference for the ADC is described in “17.7. Voltage Reference” on page 112.

Important Note: The CS0 module and the ADC0 module cannot both be enabled at the same time. Even if one module is enabled and not running, the other module must not be enabled.

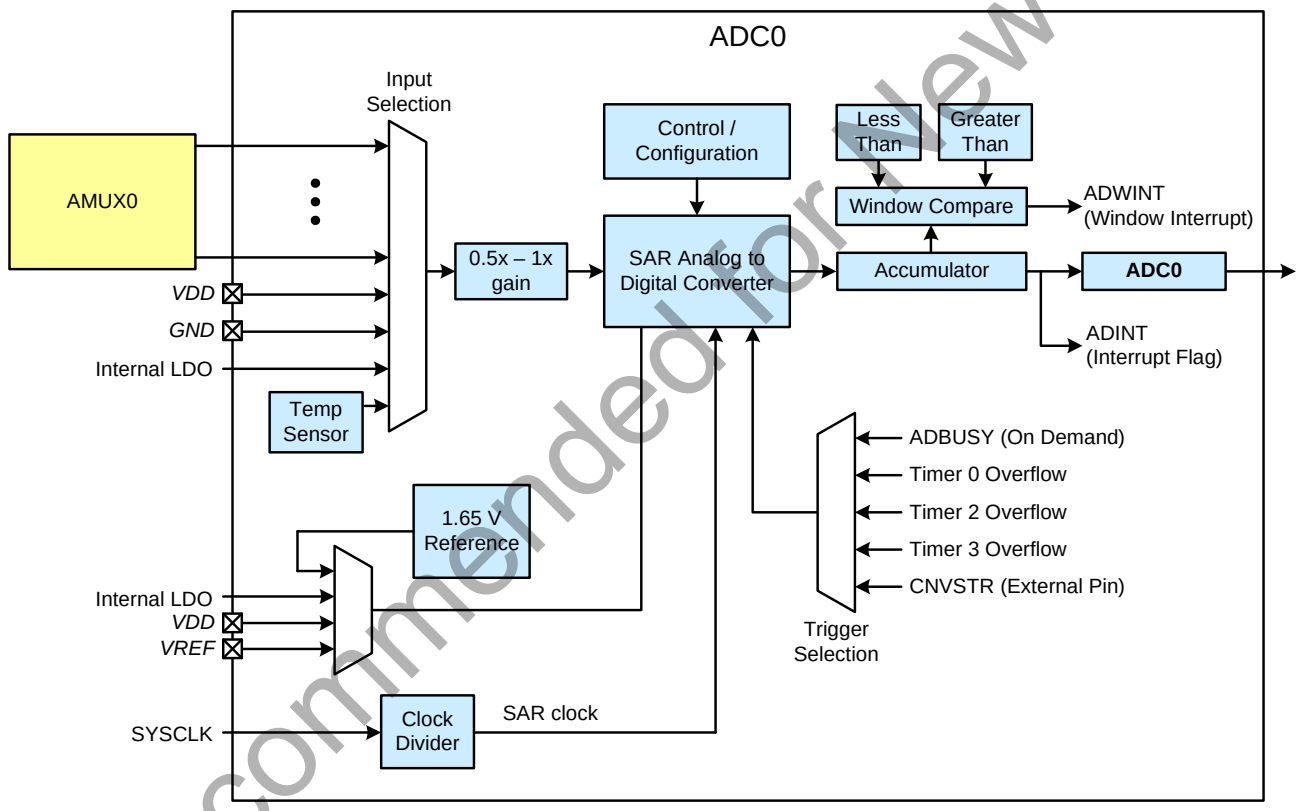


Figure 17.1. ADC0 Functional Block Diagram

17.1. ADC0 Analog Multiplexer

The ADC0 module has an analog multiplexer that selects the positive inputs to the single-ended ADC0. Any of the following may be selected as the positive input: pins from the analog pad selector (AMUX0), the on-chip temperature sensor, internal regulated digital supply voltage (output of VREG0), VDD, or GND. The ADC0 input channels are selected in the ADC0MX register. If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.

Important Note: Only one AMUX0 input should be enabled at a time when using ADC0 with AMUX0 (ADC0MX = 0).

Table 17.1. ADC0 Input Multiplexer Channels

ADC0MX setting	Signal Name	QFN-48 Pin Name	QFN-32 Pin Name	QFN-24 Pin Name
00000	ADC0.0	AMUX Input Selector		
00001 – 11010	Reserved	Reserved		
11011	TEMP	Temperature Sensor Output		
11100	VDD	VDD Supply Voltage		
11101	LDO	Internal LDO regulator output		
11110	Reserved	Reserved		
11111	GND	Ground		

Important Note about ADC0 Input Configuration: A port pin selected as ADC0 input should be configured as follows:

1. Set to analog mode input by clearing to 0 the corresponding bit in register PnMDIN.
2. Force the Priority Crossbar Decoder to skip the pin by setting 1 to the corresponding bit in register PnSKIP.
3. Disable the auto-ground for the pin by setting 1 to the corresponding bit in the port latch (Pn).
4. Enable the analog pad selection for the pin by setting 1 to the corresponding bit in the AMUX0Pn register.

See “26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)” on page 277 for more Port I/O configuration details.

17.2. Output Code Formatting

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 AD0SJST[2:0]. When the repeat count is set to 1, 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	Right-Justified ADC0H:ADC0L (AD0SJST = 000)	Left-Justified ADC0H:ADC0L (AD0SJST = 100)
VREF x 1023/1024	0x03FF	0xFFC0
VREF x 512/1024	0x0200	0x8000
VREF x 256/1024	0x0100	0x4000
0	0x0000	0x0000

When the repeat count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. Sets of 4, 8, 16, 32, or 64 consecutive samples can be accumulated and represented in unsigned integer format. The repeat count can be selected using the AD0RPT bits in the ADC0AC register. When a repeat count higher than 1, the ADC output must be right-justified (AD0SJST = 0xx); unused bits in the ADC0H and ADC0L registers are set to 0. The example below shows the right-justified result for various input voltages and repeat counts. Notice that accumulating 2^n samples is equivalent to left-shifting by *n* bit positions when all samples returned from the ADC have the same value.

Input Voltage	Repeat Count = 4	Repeat Count = 16	Repeat Count = 64
V_{REF} x 1023/1024	0x0FFC	0x3FF0	0xFFC0
V_{REF} x 512/1024	0x0800	0x2000	0x8000
V_{REF} x 511/1024	0x07FC	0x1FF0	0x7FC0
0	0x0000	0x0000	0x0000

The AD0SJST bits can be used to format the contents of the 16-bit accumulator. The accumulated result can be shifted right by 1, 2, or 3 bit positions. Based on the principles of oversampling and averaging, the effective ADC resolution increases by 1 bit each time the oversampling rate is increased by a factor of 4. The example below shows how to increase the effective ADC resolution by 1, 2, and 3 bits to obtain an effective ADC resolution of 11-bit, 12-bit, or 13-bit respectively without CPU intervention.

Input Voltage	Repeat Count = 4 Shift Right = 1 11-Bit Result	Repeat Count = 16 Shift Right = 2 12-Bit Result	Repeat Count = 64 Shift Right = 3 13-Bit Result
V_{REF} x 1023/1024	0x07F7	0x0FFC	0x1FF8
V_{REF} x 512/1024	0x0400	0x0800	0x1000
V_{REF} x 511/1024	0x03FE	0x04FC	0x0FF8
0	0x0000	0x0000	0x0000

17.3. Modes of Operation

ADC0 has a maximum conversion speed of 300 ksps. The ADC0 conversion clock (SARCLK) is a divided version of the system clock when burst mode is disabled (BURSTEN = 0), or a divided version of the low power oscillator when burst mode is enabled (BURSEN = 1). The clock divide value is determined by the AD0SC bits in the ADC0CF register.

17.3.1. 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 3 overflow
5. A rising edge on the CNVSTR input signal (pin P0.6)

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). 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. 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 “32. Timers (Timer0, Timer1, Timer2, and Timer3)” on page 389 for timer configuration.

Important Note About Using CNVSTR: The CNVSTR input pin also functions as Port pin P0.6. When the CNVSTR input is used as the ADC0 conversion source, Port pin P0.6 should be skipped by the Digital Crossbar. To configure the Crossbar to skip P0.6, set to 1 Bit 6 in register P0SKIP. See “26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)” on page 277 for details on Port I/O configuration.

17.3.2. Tracking Modes

Each ADC0 conversion must be preceded by a minimum tracking time in order for the converted result to be accurate. The minimum tracking time is given in Table 1.11. The AD0TM bit in register ADC0CN controls the ADC0 track-and-hold mode. In its default state when Burst Mode is disabled, 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 17.2). 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 “17.3.4. Settling Time Requirements” on page 110.

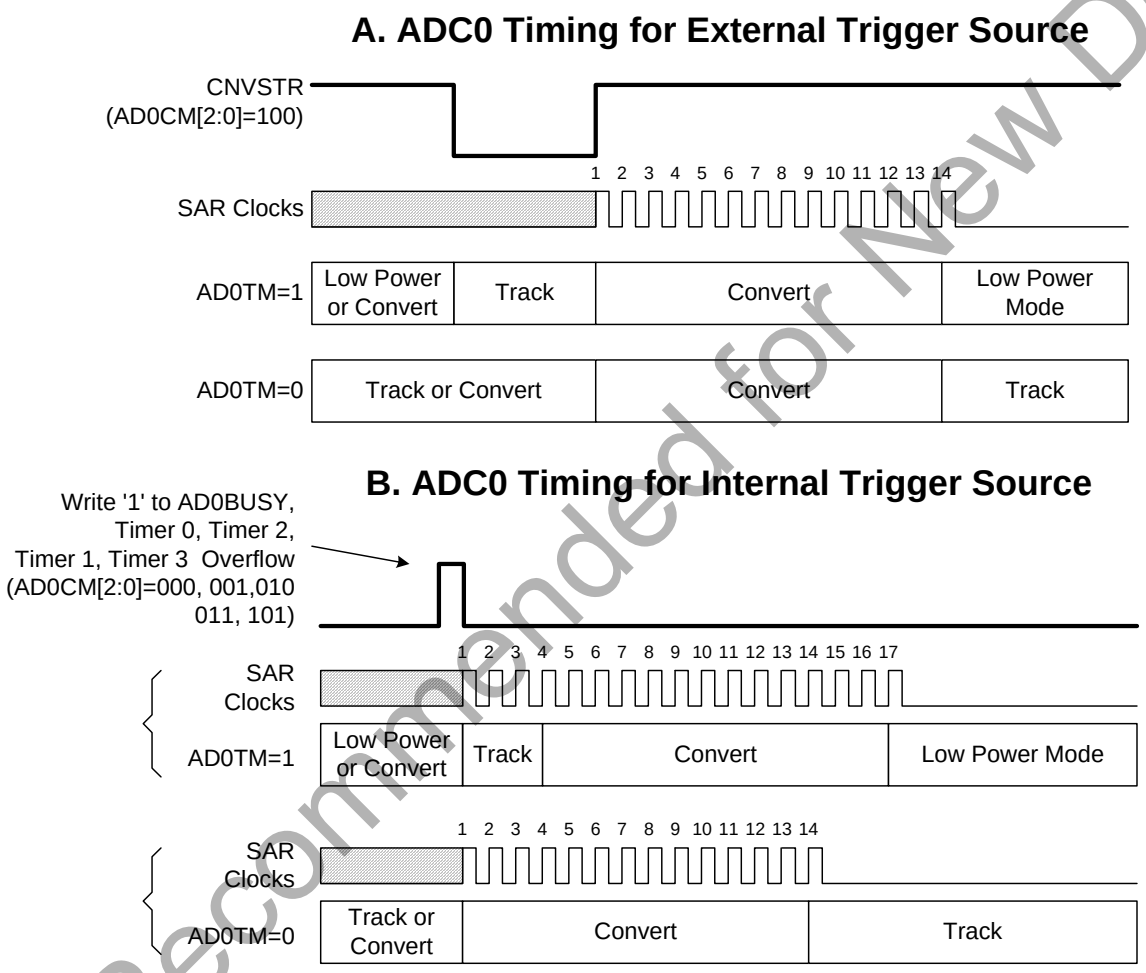


Figure 17.2. 10-Bit ADC Track and Conversion Example Timing (BURSTEN = 0)

17.3.3. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a low power state, accumulates 1, 4, 8, 16, 32, or 64 using an internal Burst Mode clock (approximately 20 MHz), then re-enters a low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a low power state within a single system clock cycle, even if the system clock is slow (e.g. 32.768 kHz), or suspended.

Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e., the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 17.3 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

In Burst Mode, tracking is determined by the settings in AD0PWR and AD0TK. The default settings for these registers will work in most applications without modification; however, settling time requirements may need adjustment in some applications. Refer to “17.3.4. Settling Time Requirements” on page 110 for more details.

Notes:

- Setting AD0TM to 1 will insert an additional 3 SAR clocks of tracking before each conversion, regardless of the settings of AD0PWR and AD0TK.
- When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals. Burst Mode must not be enabled together with the Capacitive Sense (CS0) module.

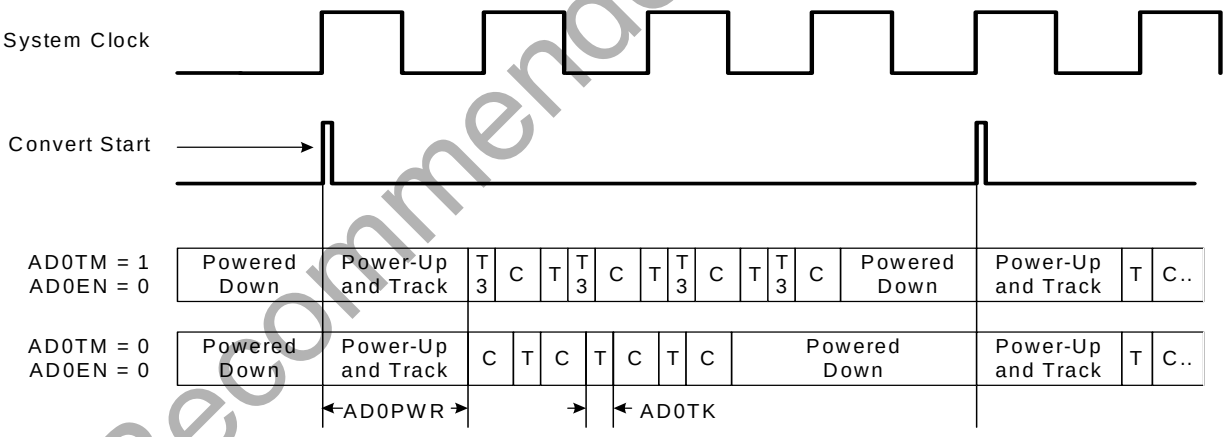


Figure 17.3. Burst Mode Tracking Example with Repeat Count Set to 4

17.3.4. Settling Time Requirements

A minimum amount of tracking time is required before each conversion can be performed, to allow the sampling capacitor voltage to settle. 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 many applications, these three SAR clocks will meet the minimum tracking time requirements, and higher values for the external source impedance will increase the required tracking time.

Figure 17.4 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 17.1. When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 1.11 for ADC0 minimum settling time requirements as well as the mux impedance and sampling capacitor values.

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

Where

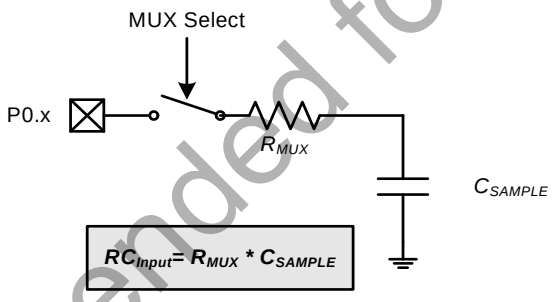
SA is the settling accuracy, given as a fraction of an LSB (e.g., 0.25 to settle within 1/4 LSB)

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)

Equation 17.1. ADC0 Settling Time Requirements



Note: The value of CSAMPLE depends on the PGA Gain. See Table 1.11 for details.

Figure 17.4. ADC0 Equivalent Input Circuits

17.3.5. Gain Setting

The ADC has gain settings of 1x and 0.5x. In 1x mode, the full scale reading of the ADC is determined directly by V_{REF}. In 0.5x mode, the full-scale reading of the ADC occurs when the input voltage is V_{REF} x 2. The 0.5x gain setting can be useful to obtain a higher input Voltage range when using a small V_{REF} voltage, or to measure input voltages that are between V_{REF} and V_{DD}. Gain settings for the ADC are controlled by the AMP0GN bit in register ADC0CF.

17.4. 8-Bit Mode

Setting the ADC08BE bit in register ADC0CF to 1 will put the ADC in 8-bit mode. In 8-bit mode, only the 8 MSBs of data are converted, allowing the conversion to be completed in two fewer SAR clock cycles than a 10-bit conversion. This can result in an overall lower power consumption since the system can spend more time in a low power mode. The two LSBs of a conversion are always 00 in this mode, and the ADC0L register will always read back 0x00.

17.5. Low Power Mode

The SAR converter provides a low power mode that allows a significant reduction in operating current when operating at low SAR clock frequencies. Low power mode is enabled by setting the AD0LPM bit (ADC0PWR.7) to 1. In general, low power mode is recommended when operating with SAR conversion clock frequency at 4 MHz or less. See “1. Electrical Characteristics” on page 10 for details on power consumption and the maximum clock frequencies allowed in each mode. Setting the Low Power Mode bit reduces the bias currents in both the SAR converter and in the High-Speed Voltage Reference.

17.6. Window Detector In Single-Ended Mode

Figure 17.5 shows two example window comparisons for right-justified data, with ADC0LTH:ADC0LTL = 0x0080 (128d) and ADC0GTH:ADC0GTL = 0x0040 (64d). 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 17.6 shows an example using left-justified data with the same comparison values.

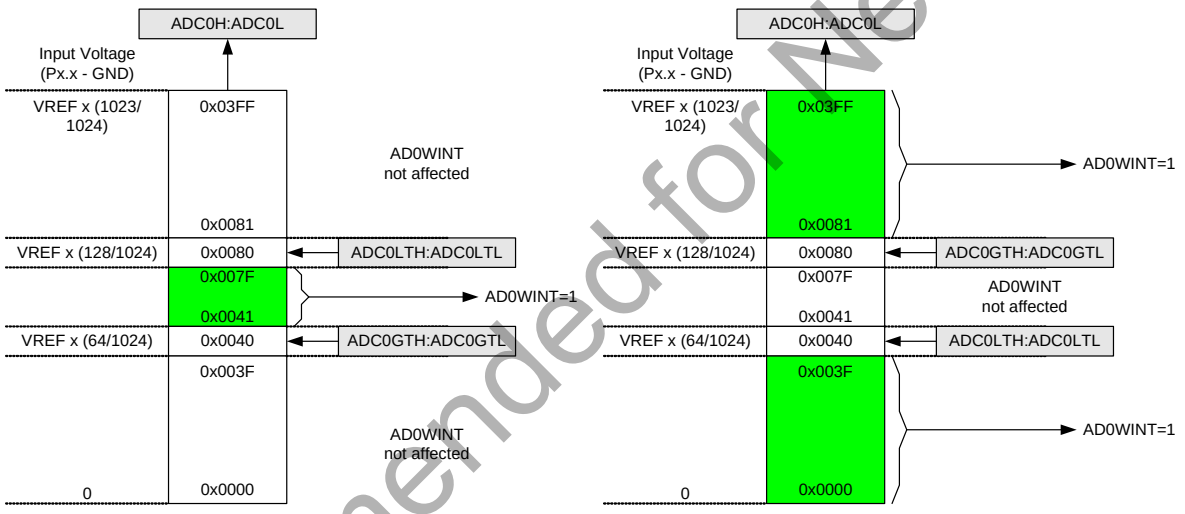


Figure 17.5. ADC Window Compare Example: Right-Justified Single-Ended Data

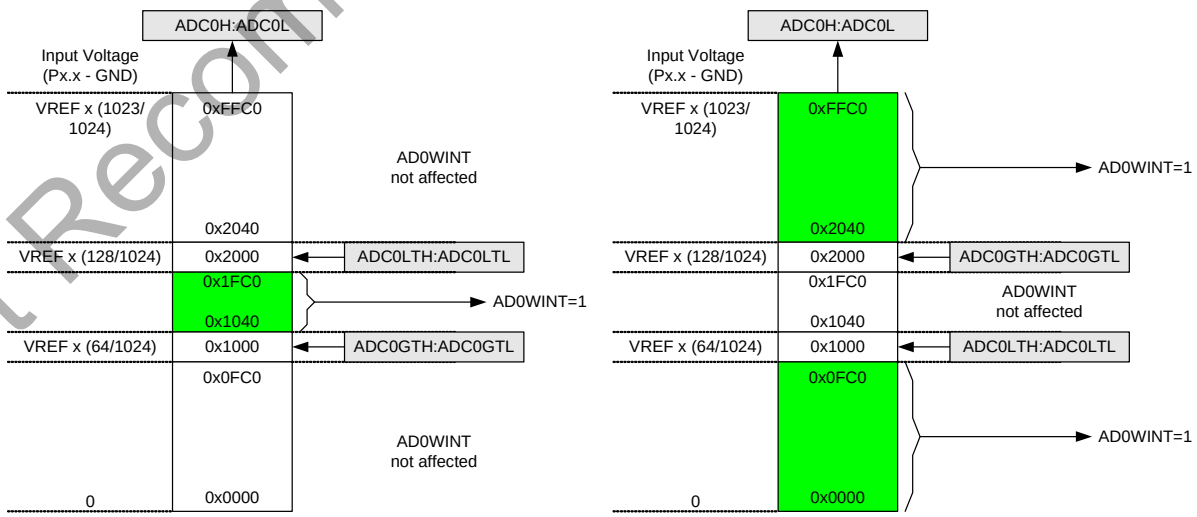


Figure 17.6. ADC Window Compare Example: Left-Justified Single-Ended Data

17.7. Voltage Reference

The voltage reference MUX is configurable to use an externally connected voltage reference, the internal voltage reference, or one of two power supply voltages (see Figure 17.7).

The voltage reference MUX is configured using the REF0CN register. Electrical specifications are can be found in the Electrical Specifications Chapter.

Important Note about the VREF Input: Port pin P0.0 is used as the external VREF input. When using an external voltage reference, P0.0/VREF should be configured as an analog input and skipped by the Priority Crossbar Decoder. Refer to “26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)” on page 277 for complete Port I/O configuration details. The external reference voltage must be within the range 0 ≤ VREF ≤ VDD.

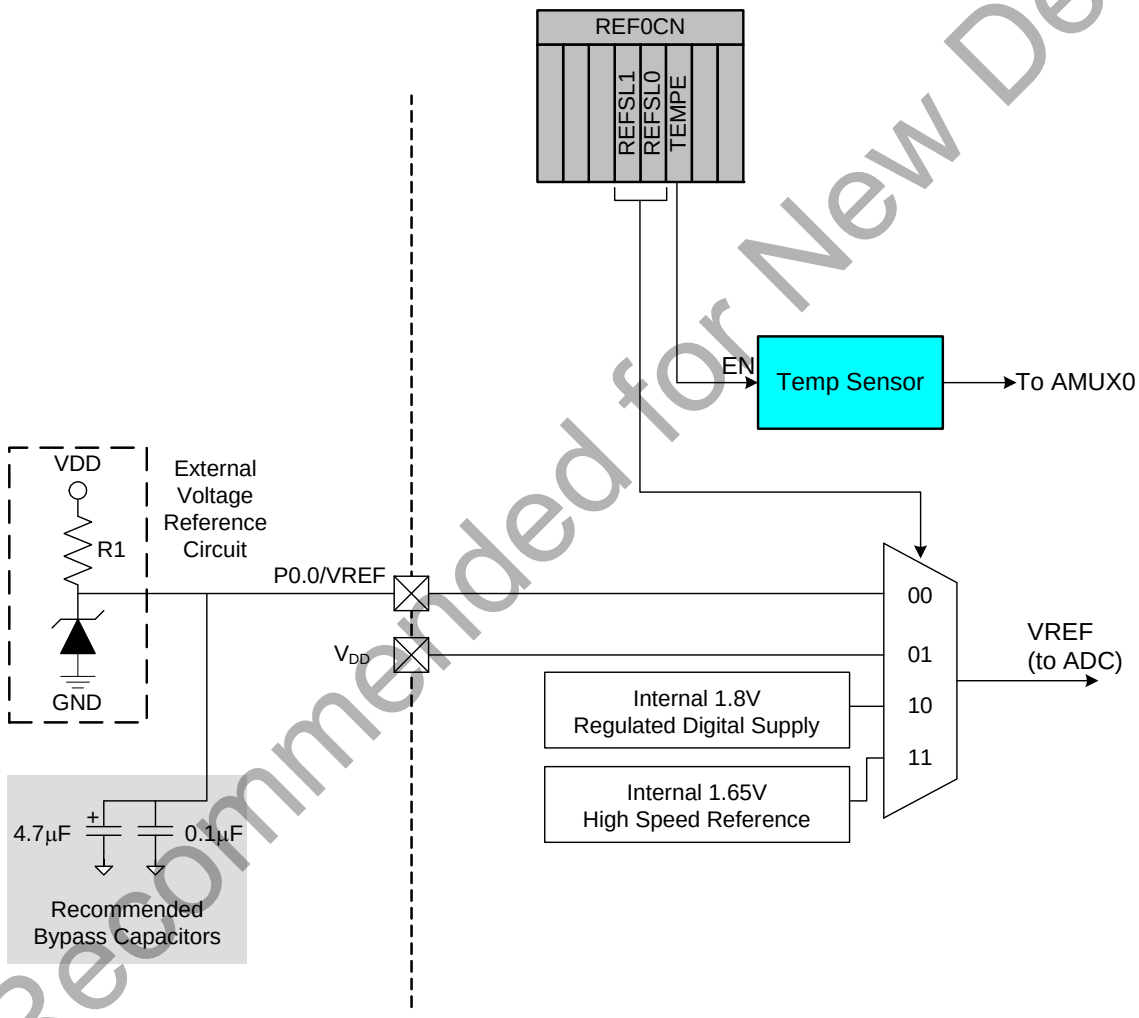


Figure 17.7. Voltage Reference Functional Block Diagram

17.7.1. External Voltage Reference

To use an external voltage reference, REFSL[1:0] should be set to 00. Bypass capacitors should be added as recommended by the manufacturer of the external voltage reference. If the manufacturer does not provide recommendations, a 4.7 µF in parallel with a 0.1 µF capacitor is recommended.

17.7.2. Internal Voltage Reference

For applications requiring the maximum number of port I/O pins, or very short VREF turn-on time, the 1.65 V high-speed reference will be the best internal reference option to choose. The high-speed internal reference is selected by setting REFSL to 11b. When selected, the high-speed internal reference will be automatically enabled/disabled on an as-needed basis by ADC0.

For applications with a non-varying power supply voltage, using the power supply as the voltage reference can provide ADC0 with added dynamic range at the cost of reduced power supply noise rejection. To use the 1.8 to 3.6 V power supply voltage (VDD) or the 1.8 V regulated digital supply voltage as the reference source, REFSL should be set to 01b or 10b, respectively.

17.8. Temperature Sensor

An on-chip temperature sensor is included, which can be directly accessed via the ADC multiplexer in single-ended configuration. To use the ADC to measure the temperature sensor, the ADC mux channel should select the temperature sensor. The temperature sensor transfer function is shown in Figure 17.8. The output voltage (V_{TEMP}) is the positive ADC input when the ADC multiplexer is set correctly. The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC measurements performed on the sensor will result in meaningless data. Refer to the electrical specification tables for the slope and offset parameters of the temperature sensor.

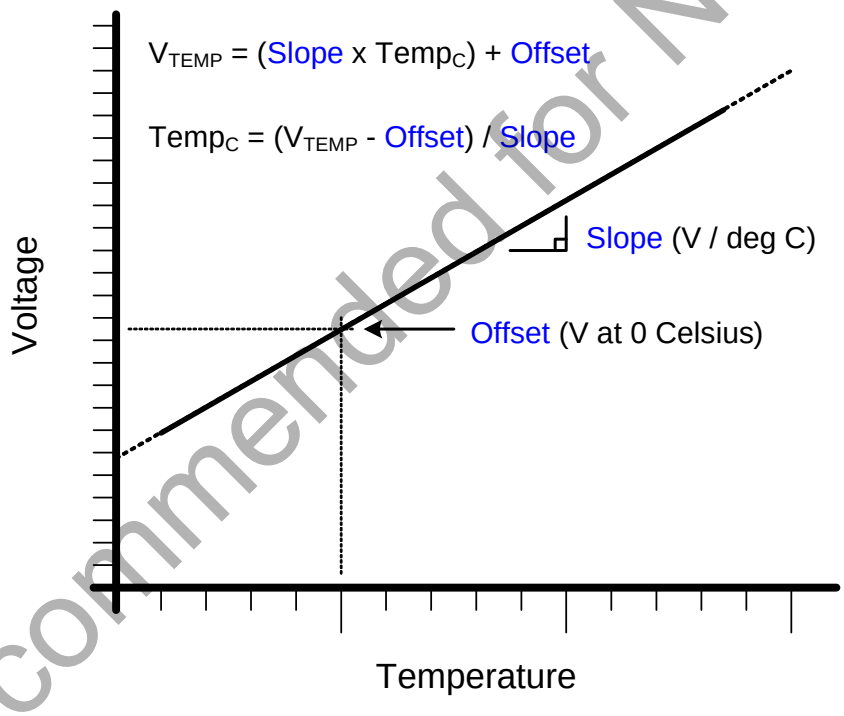


Figure 17.8. Temperature Sensor Transfer Function

17.8.1. Calibration

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

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

17.9. ADC Control Registers

Register 17.1. ADC0CN: ADC0 Control

Bit	7	6	5	4	3	2	1	0
Name	ADEN	ADBMEN	ADINT	ADBUSY	ADWINT	ADCM		
Type	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xE8 (bit-addressable)								

Table 17.2. ADC0CN Register Bit Descriptions

Bit	Name	Function
7	ADEN	**ADC Enable.** 0: Disable ADC0 (low-power shutdown). 1: Enable ADC0 (active and ready for data conversions).
6	ADBMEN	**Burst Mode Enable.** Important Note: Burst Mode must be disabled if CS0 is active. This is true regardless of the ADC0MX configuration. 0: ADC0 Burst Mode Disabled. 1: ADC0 Burst Mode Enabled.
5	ADINT	**Conversion Complete Interrupt Flag.** Set by hardware upon completion of a data conversion (ADBMEN=0), or a burst of conversions (ADBMEN=1). Can trigger an interrupt. Must be cleared by firmware.
4	ADBUSY	**ADC Busy.** Writing 1 to this bit initiates an ADC conversion when ADCM = 000. This bit should not be polled to indicate when a conversion is complete. Instead, the ADINT bit should be used when polling for conversion completion.
3	ADWINT	**Window Compare Interrupt Flag.** Set by hardware when the contents of ADC0H:ADC0L fall within the window specified by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL. Can trigger an interrupt. Must be cleared by firmware.
2:0	ADCM	**Start of Conversion Mode Select.** Specifies the ADC0 start of conversion source. All remaining bit combinations are reserved. 000: ADC0 conversion initiated on write of 1 to ADBUSY. 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 3. 100: ADC0 conversion initiated on rising edge of CNVSTR. 101-111: Reserved.

Register 17.2. ADC0CF: ADC0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	ADSC					AD8BE	ADTM	ADGN
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x97								

Table 17.3. ADC0CF Register Bit Descriptions

Bit	Name	Function
7:3	ADSC	**SAR Clock Divider.** This field sets the ADC clock divider value. It should be configured to be as close to the maximum SAR clock speed as the datasheet will allow. The SAR clock frequency is given by the following equation: $F_{CLKSAR} = \frac{F_{ADCCLK}}{ADSC + 1}$ F_{ADCCLK} is equal to the selected SYSCLK when ADBMEN is 0 and the high-frequency oscillator when ADBMEN is 1.
2	AD8BE	**8-Bit Mode Enable.** 0: ADC0 operates in 10-bit mode (normal operation). 1: ADC0 operates in 8-bit mode.
1	ADTM	**Track Mode.** Selects between Normal or Delayed Tracking Modes. 0: Normal Track Mode. When ADC0 is enabled, conversion begins immediately following the start-of-conversion signal. 1: Delayed Track Mode. When ADC0 is enabled, conversion begins 3 SAR clock cycles following the start-of-conversion signal. The ADC is allowed to track during this time.
0	ADGN	**Gain Control.** 0: The on-chip PGA gain is 0.5. 1: The on-chip PGA gain is 1.

Register 17.3. ADC0AC: ADC0 Accumulator Configuration

Bit	7	6	5	4	3	2	1	0
Name	Reserved	ADAE	ADSJST			ADRPT		
Type	RW	W	RW			RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xBA								

Table 17.4. ADC0AC Register Bit Descriptions

Bit	Name	Function
7	Reserved	Must write reset value.
6	ADAE	**Accumulate Enable.** Enables multiple conversions to be accumulated when burst mode is disabled. This bit is write-only and always reads as zero. 0: ADC0H:ADC0L contain the result of the latest conversion when Burst Mode is disabled. 1: ADC0H:ADC0L contain the accumulated conversion results when Burst Mode is disabled. Firmware must write 0x0000 to ADC0H:ADC0L to clear the accumulated result.
5:3	ADSJST	**Accumulator Shift and Justify.** Specifies the format of data read from ADC0H:ADC0L. All remaining bit combinations are reserved. 000: Right justified. No shifting applied. 001: Right justified. Shifted right by 1 bit. 010: Right justified. Shifted right by 2 bits. 011: Right justified. Shifted right by 3 bits. 100: Left justified. No shifting applied. 101-111: Reserved.
2:0	ADRPT	**Repeat Count.** Selects the number of conversions to perform and accumulate in Burst Mode. This bit field must be set to 000 if Burst Mode is disabled. 000: Perform and Accumulate 1 conversion. 001: Perform and Accumulate 4 conversions. 010: Perform and Accumulate 8 conversions. 011: Perform and Accumulate 16 conversions. 100: Perform and Accumulate 32 conversions. 101: Perform and Accumulate 64 conversions. 110-111: Reserved.

Register 17.4. ADC0PWR: ADC0 Power Control

Bit	7	6	5	4	3	2	1	0
Name	ADLPM	Reserved			ADPWR			
Type	RW	RW			RW			
Reset	0	0	0	0	1	1	1	1
SFR Page = ALL; SFR Address: 0xBB								

Table 17.5. ADC0PWR Register Bit Descriptions

Bit	Name	Function
7	ADLPM	**Low Power Mode Enable.** This bit can be used to reduce power to the ADC's internal common mode buffer. It can be set to 1 to reduce power when tracking times in the application are longer (slower sample rates). 0: Disable low power mode. 1: Enable low power mode (requires extended tracking time).
6:4	Reserved	Must write reset value.
3:0	ADPWR	**Burst Mode Power Up Time.** This field sets the time delay allowed for the ADC to power up from a low power state. When ADTM is set, an additional 3 SARCLKs are added to this time. $T_{PWRTIME} = \frac{8 \times ADPWR}{F_{HFOSC}}$

Register 17.5. ADC0TK: ADC0 Burst Mode Track Time

Bit	7	6	5	4	3	2	1	0
Name	Reserved		ADTK					
Type	R		RW					
Reset	0	0	0	1	1	1	1	0
SFR Page = ALL; SFR Address: 0xBC								

Table 17.6. ADC0TK Register Bit Descriptions

Bit	Name	Function
7:6	Reserved	Must write reset value.
5:0	ADTK	**Burst Mode Tracking Time.** This field sets the time delay between consecutive conversions performed in Burst Mode. When ADTM is set, an additional 3 SARCLKs are added to this time. $T_{BMTK} = \frac{64 - ADTK}{F_{HFOSC}}$ The Burst Mode track delay is not inserted prior to the first conversion. The required tracking time for the first conversion should be defined with the ADPWR field.

Register 17.6. ADC0H: ADC0 Data Word High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0H							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xD3								

Table 17.7. ADC0H Register Bit Descriptions

Bit	Name	Function
7:0	ADC0H	**Data Word High Byte.** When read, this register returns the most significant byte of the 16-bit ADC0 accumulator, formatted according to the settings in ADSJST. The register may also be written, to set the upper byte of the 16-bit ADC0 accumulator.
Note: If Accumulator shifting is enabled, the most significant bits of the value read will be zeros.		

Register 17.7. ADC0L: ADC0 Data Word Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0L							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xD2								

Table 17.8. ADC0L Register Bit Descriptions

Bit	Name	Function
7:0	ADC0L	**Data Word Low Byte.** When read, this register returns the least significant byte of the 16-bit ADC0 accumulator, formatted according to the settings in ADSJST. The register may also be written, to set the lower byte of the 16-bit ADC0 accumulator.
Note: If Accumulator shifting is enabled, the most significant bits of the value read will be zeros.		

Register 17.8. ADC0GTH: ADC0 Greater-Than High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0GTH							
Type	RW							
Reset	1	1	1	1	1	1	1	1
SFR Page = 0x0; SFR Address: 0xC4								

Table 17.9. ADC0GTH Register Bit Descriptions

Bit	Name	Function
7:0	ADC0GTH	**Greater-Than High Byte.** Most significant byte of the 16-bit greater-than window compare register.

Register 17.9. ADC0GTL: ADC0 Greater-Than Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0GTL							
Type	RW							
Reset	1	1	1	1	1	1	1	1
SFR Page = 0x0; SFR Address: 0xC3								

Table 17.10. ADC0GTL Register Bit Descriptions

Bit	Name	Function
7:0	ADC0GTL	**Greater-Than Low Byte.** Least significant byte of the 16-bit greater-than window compare register.
Note: In 8-bit mode, this register should be set to 0x00.		

Register 17.10. ADC0LTH: ADC0 Less-Than High Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0LTH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xC6								

Table 17.11. ADC0LTH Register Bit Descriptions

Bit	Name	Function
7:0	ADC0LTH	**Less-Than High Byte.** Most significant byte of the 16-bit less-than window compare register.

Register 17.11. ADC0LTL: ADC0 Less-Than Low Byte

Bit	7	6	5	4	3	2	1	0
Name	ADC0LTL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xC5								

Table 17.12. ADC0LTL Register Bit Descriptions

Bit	Name	Function
7:0	ADC0LTL	**Less-Than Low Byte.** Least significant byte of the 16-bit less-than window compare register.
Note: In 8-bit mode, this register should be set to 0x00.		

Register 17.12. ADC0MX: ADC0 Multiplexer Selection

Bit	7	6	5	4	3	2	1	0
Name	Reserved			ADC0MX				
Type	R			RW				
Reset	0	0	0	1	1	1	1	1
SFR Page = 0x0; SFR Address: 0xD4								

Table 17.13. ADC0MX Register Bit Descriptions

Bit	Name	Function
7:5	Reserved	Must write reset value.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

Table 17.13. ADC0MX Register Bit Descriptions

Bit	Name	Function
4:0	ADC0MX	**AMUX0 Positive Input Selection.** Selects the positive input channel for ADC0. For reserved bit combinations, no input is selected. 00000: Select channel ADC0.0. 00001: Select channel ADC0.1. 00010: Select channel ADC0.2. 00011: Select channel ADC0.3. 00100: Select channel ADC0.4. 00101: Select channel ADC0.5. 00110: Select channel ADC0.6. 00111: Select channel ADC0.7. 01000: Select channel ADC0.8. 01001: Select channel ADC0.9. 01010: Select channel ADC0.10. 01011: Select channel ADC0.11. 01100: Select channel ADC0.12. 01101: Select channel ADC0.13. 01110: Select channel ADC0.14. 01111: Select channel ADC0.15. 10000: Select channel ADC0.16. 10001: Select channel ADC0.17. 10010: Select channel ADC0.18. 10011: Select channel ADC0.19. 10100: Select channel ADC0.20. 10101: Select channel ADC0.21. 10110: Select channel ADC0.22. 10111: Select channel ADC0.23. 11000: Select channel ADC0.24. 11001: Select channel ADC0.25. 11010: Select channel ADC0.26. 11011: Temperature Sensor. 11100: VDD Supply Voltage. 11101: Internal LDO regulator output. 11110: Reserved. 11111: Ground.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

17.10. Voltage Reference Registers

Register 17.13. REF0CN: Voltage Reference Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved			REFSL		TEMPE	Reserved	
Type	RW			RW		RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xD1								

Table 17.14. REF0CN Register Bit Descriptions

Bit	Name	Function
7:5	Reserved	Must write reset value.
4:3	REFSL	**Voltage Reference Select.** Selects the ADC0 voltage reference. 00: The ADC0 voltage reference is the P0.0/VREF pin. 01: The ADC0 voltage reference is the VDD pin. 10: The ADC0 voltage reference is the internal 1.8 V digital supply voltage. 11: The ADC0 voltage reference is the internal 1.65 V high speed voltage reference.
2	TEMPE	**Temperature Sensor Enable.** Enables/Disables the internal temperature sensor. 0: Disable the Temperature Sensor. 1: Enable the Temperature Sensor.
1:0	Reserved	Must write reset value.

17.11. Temperature Sensor Registers

Register 17.14. TOFFH: Temperature Sensor Offset High

Bit	7	6	5	4	3	2	1	0
Name	TOFF							
Type	R							
Reset	X	X	X	X	X	X	X	X
SFR Page = 0xF; SFR Address: 0x8E								

Table 17.15. TOFFH Register Bit Descriptions

Bit	Name	Function
7:0	TOFF	**Temperature Sensor Offset High.** Most Significant Bits of the 10-bit temperature sensor offset measurement.

Register 17.15. TOFFL: Temperature Sensor Offset Low

Bit	7	6	5	4	3	2	1	0
Name	TOFF		Reserved					
Type	R		R					
Reset	X	X	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x8D								

Table 17.16. TOFFL Register Bit Descriptions

Bit	Name	Function
7:6	TOFF	**Temperature Sensor Offset Low.** Least Significant Bits of the 10-bit temperature sensor offset measurement.
5:0	Reserved	Must write reset value.

18. Capacitive Sense (CS0)

The Capacitive Sense subsystem uses a capacitance-to-digital circuit to determine the capacitance on a port pin. The module can take measurements from different port pins using the module's analog multiplexer. The module is enabled only when the CS0INT bit (CS0CN) is set to 1. Otherwise the module is in a low-power shutdown state. The module can be configured to take measurements on one port pin or a group of port pins, using auto-scan. A selectable gain circuit allows the designer to adjust the maximum allowable capacitance. An accumulator is also included, which can be configured to average multiple conversions on an input channel. Interrupts can be generated when CS0 completes a conversion or when the measured value crosses a threshold defined in CS0THH:L.

Important Note: The CS0 module and the ADC0 module cannot both be enabled at the same time. Even if one module is enabled and not running, the other module must not be enabled.

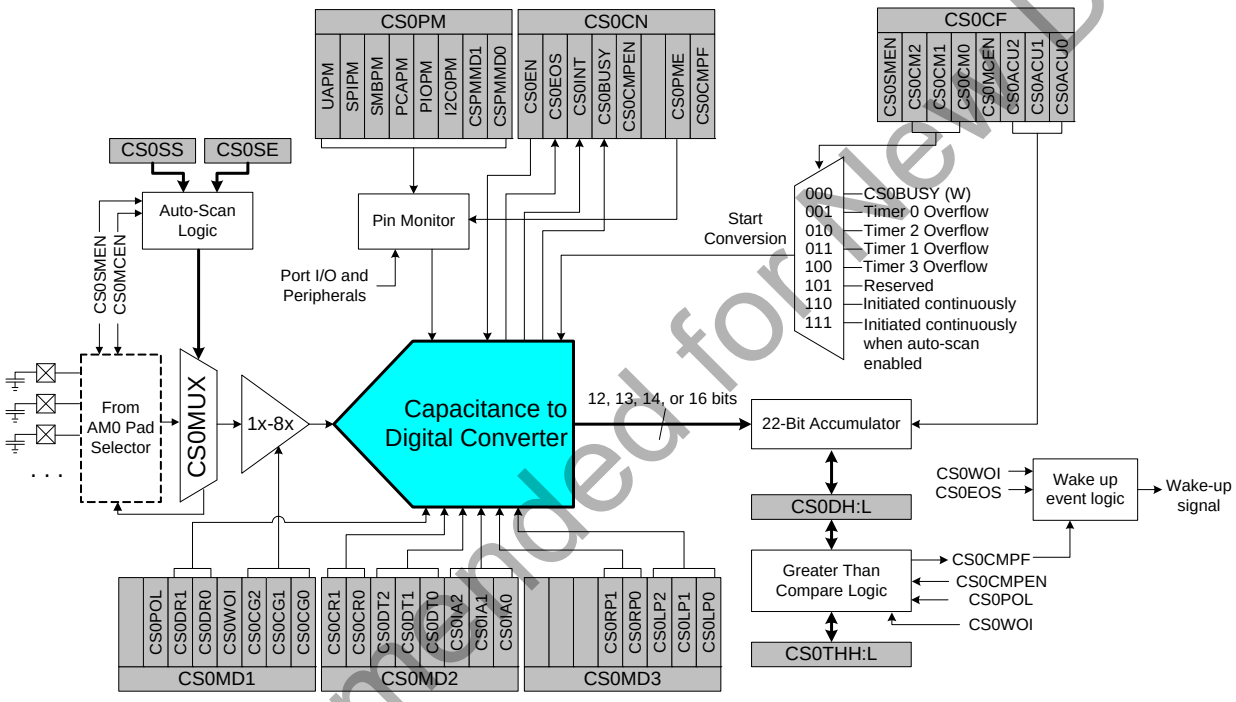


Figure 18.1. CS0 Block Diagram

18.1. Configuring Port Pins as Capacitive Sense Inputs

In order for a port pin to be measured by CS0, that port pin must be configured as an analog input (see “26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)”). Configuring the input multiplexer to a port pin not configured as an analog input will cause the capacitive sense comparator to output incorrect measurements.

18.2. Initializing the Capacitive Sensing Peripheral

The following procedure is recommended for properly initializing the CS0 peripheral:

1. Enable the CS0 block (CS0INT = 1) before performing any other initializations.
2. Initialize the Start of Conversion Mode Select bits (CS0CM[2:0]) to the desired mode.
3. Continue initializing all remaining CS0 registers.

18.3. Capacitive Sense Start-Of-Conversion Sources

A capacitive sense conversion can be initiated in one of seven ways, depending on the programmed state of the CS0 start of conversion bits (CS0CF6:4). Conversions may be initiated by one of the following:

1. Writing a 1 to the CS0BUSY bit of register CS0CN
2. Timer 0 overflow
3. Timer 2 overflow
4. Timer 1 overflow
5. Timer 3 overflow
6. Convert continuously
7. Convert continuously with auto-scan enabled

Conversions can be configured to be initiated continuously through one of two methods. CS0 can be configured to convert at a single channel continuously or it can be configured to convert continuously with auto-scan enabled. When configured to convert continuously, conversions will begin after the CS0BUSY bit in CS0CF has been set. An interrupt will be generated if CS0 conversion complete interrupts are enabled by setting the ECSCPT bit (EIE2.0).

The CS0 module uses a method of successive approximation to determine the value of an external capacitance. The number of bits the CS0 module converts is adjustable using the CS0CR bits in register CS0MD2. Conversions are 13 bits long by default, but they can be adjusted to 12, 13, 14, or 16 bits depending on the needs of the application. Unconverted bits will be set to 0. Shorter conversion lengths produce faster conversion rates, and vice-versa. Applications can take advantage of faster conversion rates when the unconverted bits fall below the noise floor.

Note: CS0 conversion complete interrupt behavior depends on the settings of the CS0 accumulator. If CS0 is configured to accumulate multiple conversions on an input channel, a CS0 conversion complete interrupt will be generated only after the last conversion completes.

18.4. CS0 Multiple Channel Enable

CS0 has the capability of measuring the total capacitance of multiple channels using a single conversion. When the multiple channel feature is enabled (CS0MCEN = 1), Channels selected by AMUX0Pn are internally shorted together and the combined node is selected as the CS0 input. This mode can be used to detect a capacitance change on multiple channels using a single conversion and is useful for implementing “wake-on-multiple channels”.

18.5. CS0 Gain Adjustment

The gain of the CS0 circuit can be adjusted in integer increments from 1x to 8x (8x is the default). High gain gives the best sensitivity and resolution for small capacitors, such as those typically implemented as touch-sensitive PCB features. To measure larger capacitance values, the gain should be lowered accordingly. The bits CS0CG[2:0] in register CS0MD set the gain value.

18.6. Wake from Suspend

CS0 has the capability of waking the device from a low power suspend mode upon detection of a “touch” using the digital comparator. This threshold comparator event can also be used as an end-of-conversion wake by setting the threshold registers to all zeros.

18.7. Using CS0 in Applications that Utilize Sleep Mode

To achieve maximum power efficiency, CS0 should be enabled only when taking a conversion and disabled at all other times. CS0 must be disabled by software prior to entering sleep mode.

18.8. Automatic Scanning (Method 1—CS0SMEN = 0)

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

Auto-scan is enabled by setting the CS0 start-of-conversion bits (CS0CF6:4) to 111b. After enabling auto-scan, the starting and ending channels should be set to appropriate values in CS0SS and CS0SE, respectively. Writing to CS0SS when auto-scan is enabled will cause the value written to CS0SS to be copied into CS0MX. After being enabled, writing a 1 to CS0BUSY will start auto-scan conversions. When auto-scan completes the number of conversions defined in the CS0 accumulator bits (CS0CF1:0), auto-scan configures CS0MX to the next sequential port pin configured as an analog input and begins a conversion on that channel. All other pins between CS0SS and CS0SE which are set as analog inputs are grounded during the conversion. This scan sequence continues until CS0MX reaches the ending input channel value defined in CS0SE. After one or more conversions have been taken at this channel, auto-scan configures CS0MX back to the starting input channel. For an example system configured to use auto-scan, please see Figure “18.2 Auto-Scan Example” on page 133.

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

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

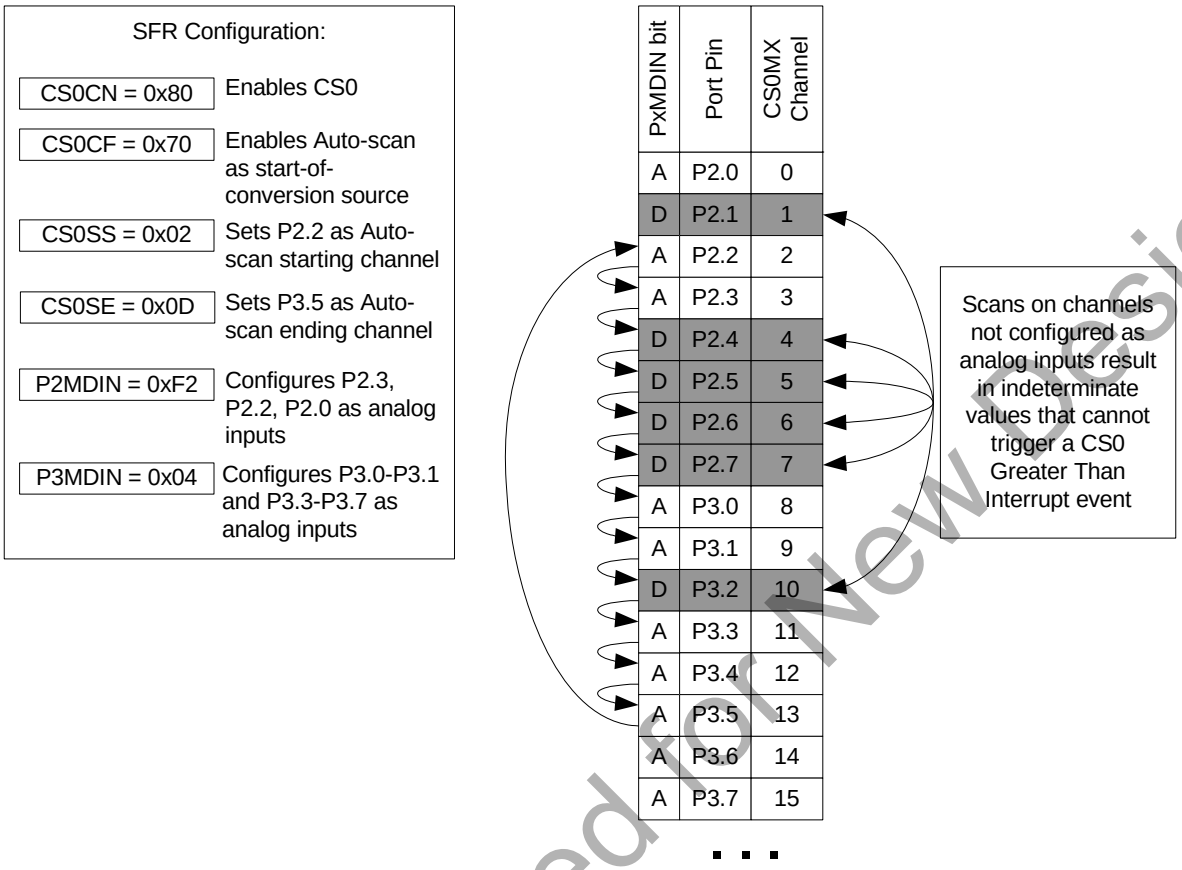


Figure 18.2. Auto-Scan Example

18.9. Automatic Scanning (Method 2—CS0SMEN = 1)

When CS0SMEN is enabled, CS0 uses an alternate autoscanning method that uses the contents of AMUX0Pn to determine which channels to include in the scan. This maximizes flexibility for application development and can result in more power efficient scanning. The following procedure can be used to configure the device for Automatic Scanning with CS0SMEN = 1.

1. Set the CS0SMEN bit to 1.
2. Select the start of conversion mode (CS0CM[2:0]) if not already configured. Mode 101b is the mode of choice for most systems.
3. Configure the AMUX0Pn registers to enable channels in the scan.
4. Configure the CS0THH:CS0THL digital comparator threshold and polarity.
5. Enable wake from suspend on end of scan (CS0WOI = 1) if this functionality is desired.
6. Set CS0SS to point to the first channel in the scan. Note: CS0SS uses the same bit mapping as the CS0MX register.
7. Issue a start of conversion (BUSY = 1).
8. Enable the CS0 Wakeup Source and place the device in Suspend mode (optional).

If using Mode 101b, scanning will stop once a “touch” has been detected using the digital comparator. The CS0MX register will contain the channel mux value of the channel that caused the interrupt. Setting the busy bit when servicing the interrupt will cause the scan to continue where it left off. Scanning will also stop after all channels have been sampled and no “touches” have been detected. If the CS0WOI bit is set, a wake from suspend event will be generated. Note: When automatic scanning is enabled, the contents of the CS0MX register are only valid when the digital comparator interrupt is set and BUSY = 0.

18.10. CS0 Comparator

The CS0 comparator compares the latest capacitive sense conversion result with the value stored in CS0THH:CS0THL. If the result is less than or equal to the stored value, the CS0CMPF bit(CS0CN:0) is set to 0. If the result is greater than the stored value, CS0CMPF is set to 1.

If the CS0 conversion accumulator is configured to accumulate multiple conversions, a comparison will not be made until the last conversion has been accumulated.

An interrupt will be generated if CS0 greater-than comparator interrupts are enabled by setting the ECSGRT bit (EIE2.1) when the comparator sets CS0CMPF to 1.

If auto-scan is running when the comparator sets the CS0CMPF bit, no further auto-scan initiated conversions will start until firmware sets CS0BUSY to 1.

A CS0 greater-than comparator event can wake a device from suspend mode. This feature is useful in systems configured to continuously sample one or more capacitive sense channels. The device will remain in the low-power suspend state until the captured value of one of the scanned channels causes a CS0 greater-than comparator event to occur. It is not necessary to have CS0 comparator interrupts enabled in order to wake a device from suspend with a greater-than event.

For a summary of behavior with different CS0 comparator, auto-scan, and auto accumulator settings, please see Table 18.1.

18.11. CS0 Conversion Accumulator

CS0 can be configured to accumulate multiple conversions on an input channel. The number of samples to be accumulated is configured using the CS0ACU2:0 bits (CS0CF2:0). The accumulator can accumulate 1, 4, 8, 16, 32, or 64 samples. After the defined number of samples have been accumulated, the result is divided by either 1, 4, 8, 16, 32, or 64 (depending on the CS0ACU[2:0] setting) and copied to the CS0DH:CS0DL SFRs.

Table 18.1. Operation with Auto-Scan and Accumulate

Auto-Scan Enabled	Accumulator Enabled	CS0 Conversion Complete Interrupt Behavior	CS0 Greater Than Interrupt Behavior	CS0MX Behavior
N	N	CS0INT Interrupt serviced after 1 conversion completes	Interrupt serviced after 1 conversion completes if value in CS0DH:CS0DL is greater than CS0THH:CS0THL	CS0MX unchanged.
N	Y	CS0INT Interrupt serviced after *M* conversions complete	Interrupt serviced after *M* conversions complete if value in CS0DH:CS0DL (post accumulate and divide) is greater than CS0THH:CS0THL	CS0MX unchanged.
Y	N	CS0INT Interrupt serviced after 1 conversion completes	Interrupt serviced after conversion completes if value in CS0DH:CS0DL is greater than CS0THH:CS0THL; Auto-Scan stopped	If greater-than comparator detects conversion value is greater than CS0THH:CS0THL, CS0MX is left unchanged; otherwise, CS0MX updates to the next channel (CS0MX + 1) and wraps back to CS0SS after passing CS0SE.
Y	Y	CS0INT Interrupt serviced after *M* conversions complete	Interrupt serviced after *M* conversions complete if value in CS0DH:CS0DL (post accumulate and divide) is greater than CS0THH:CS0THL; Auto-Scan stopped	If greater-than comparator detects conversion value is greater than CS0THH:CS0THL, CS0MX is left unchanged; otherwise, CS0MX updates to the next channel (CS0MX + 1) and wraps back to CS0SS after passing CS0SE.

Note: M = Accumulator setting (1x, 4x, 8x, 16x, 32x, 64x).

18.12. CS0 Pin Monitor

The CS0 module provides accurate conversions in all operating modes of the CPU, peripherals and I/O ports. Pin monitoring circuits are provided to eliminate possible interference from high-current output pin switching. The CS0 Pin Monitor register (CS0PM) controls the operation of these pin monitors.

Conversions in the CS0 module are immune to any change on digital inputs and immune to most output switching. Even high-speed serial data transmission will not affect CS0 operation as long as the output load is limited. Output changes that switch large loads such as LEDs and heavily-loaded communications lines can affect conversion accuracy. For this reason, the CS0 module includes pin monitoring circuits that will, if enabled, automatically adjust conversion timing if necessary to eliminate any effect from high-current output pin switching.

The pin monitor enable bit should be set for any output signal that is expected to drive a large load.

Example: The SMBus in a system is heavily loaded with multiple slaves and a long PCB route. Set the SMBus pin monitor enable, SMBPM = 1.

Example: Timer2 controls an LED on Port 1, pin 3 to provide variable dimming. Set the Port SFR write monitor enable, PIOPM = 1.

Example: The SPI bus is used to communicate to a nearby host. The pin monitor is not needed because the output is not heavily loaded, SPIPM remains = 0, the default reset state.

Pin monitors should not be enabled unless they are required. The pin monitor works by repeating any portion of a conversion that may have been corrupted by a change on an output pin. Setting pin monitor enables bits will slow CS0 conversions.

The frequency of CS0 retry operations can be limited by setting the CSPMMD bits. In the default (reset) state, all converter retry requests will be performed. This is the recommended setting for all applications. The number of retries per conversion can be limited to either two or four retries by changing CSPMMD. Limiting the number of retries per conversion ensures that even in circumstances where extremely frequent high-power output switching occurs, conversions will be completed, though there may be some loss of accuracy due to switching noise.

Activity of the pin monitor circuit can be detected by reading the Pin Monitor Event bit, CS0PME, in register CS0CN. This bit will be set if any CS0 converter retries have occurred. It remains set until cleared by software or a device reset.

Notes on pin monitor operation:

- When the system clock is active in the system, the pin monitor feature requires a system clock frequency of 5.5 MHz or higher to function correctly.
- When using CS0 as a wake-up source with pin monitoring enabled, the minimum active mode system clock frequency that can be used is 1.8 MHz. A CS0 wake-up event in a system with an active mode clock frequency below 1.8 MHz will cause CS0 to start another conversion instead of remaining stopped when the system wakes from its low power state.

18.13. Adjusting CS0 For Special Situations

There are several configuration options in the CS0 module designed to modify the operation of the circuit and address special situations. In particular, any circuit with more than 500 Ω of series impedance between the sensor and the device pin may require the adjustments detailed in this section for optimal performance. Typical applications which may require adjustments include the following:

- Touch panel sensors fabricated using a resistive conductor such as indium-tin-oxide (ITO).
- Circuits using a high-value series resistor to isolate the sensor element for high ESD protection.

Capacitive sensors created using PCB traces should generally require no fine tuning, and the default settings for CS0DT, CS0DR, CS0IA, CS0RP and CS0LP should be used.

18.13.1. Adjusting the CS0 Reset Timing

The CS0 module determines capacitance by discharging an external capacitor and then measuring how quickly that capacitor charges. In order to do this, the external capacitor must be fully discharged before every test. There are two timers inside the CS0 module which determine the timing for the reset (discharge) operation.

CS0 performs a two-stage discharge (double reset) of the external capacitor at the start of every bit conversion to improve performance in high-noise environments. In this method, most of the charge in the external capacitor is removed in a first reset stage through a low-resistance switch to ground. A second reset is then performed using a high-resistance switch to ground. This second reset removes any ambient noise energy that might have been captured in the external capacitor at the end of the first reset stage.

The lengths of both reset periods are independently adjustable. Longer periods are used when the external capacitor is separated from the CS0 module by a large resistor (more than 500 Ω) because that series resistor would slow the rate of discharge.

Determining the appropriate settings for CS0DT (the primary reset) and CS0DR (the secondary reset) are two of a series of related adjustments which must be made when using CS0 to measure capacitive loads in the presence of high resistance.

18.13.2. Adjusting Primary Reset Timing: CS0DT

Primary reset timing adjustment is performed to provide peak sensitivity for highly-resistive loads and peak linearity for capacitive loads linked thorough a distributed resistance (such as an ITO touch panel) while minimizing the required conversion time.

The adjustment for CS0DT should be performed while CS0DR and CS0IA bits are set at their maximum values (CS0DR = 11b, CS0IA = 001b).

1. Begin the adjustment with CS0DT set to maximum delay (CS0DT = 111b). Measure the untouched average CS0 result for the channel under test.

Note: When calibrating CS0 for use with an ITO panel, consider the use of an artificial finger: a small (¼" O.D.) washer (#2 regular, #4 narrow) wired through a 1000 pF capacitor to ground. Select a touch point at the on the farthest end of the longest row. Find the point where maximum response is returned from the CS0 conversion.

2. Record the average touched CS0 value with CS0DT = 111b. The touched value should be higher than the untouched value. The magnitude of the difference between the touched and untouched average CS0 values is the figure of merit for touch sensitivity.
3. Decrease the primary reset time CS0DT by one and repeat the touched and untouched CS0 measurements. Repeat this step until values have been recorded for all eight CS0DT settings. As the CS0DT setting decreases, the average sensitivity of the CS0 value may begin to decrease significantly.
4. CS0DT should be set high enough that there is not a significant decrease in sensitivity due to resistance. Select the CS0DT setting that occurred prior to the observed drop in CS0 touch sensitivity.

18.13.3. Adjusting Secondary Reset Timing: CS0DR

Adjustments for CS0IA and CS0DT should be set to their maximum value (CS0IA = 001b, CS0DT = 111b) while the CS0DR adjustments are being performed.

Because the only function of DR is to reduce the effect of environmental noise, establishing the proper DR adjustment can only be performed in a test environment with the highest expected level of ambient noise while connected to the sensor which is specific to the intended application.

Increasing the CS0DR adjustment does not increase the level of possible noise rejection, it only changes the amount of time that the CS0 module will wait for the secondary reset circuit to finish its noise-reduction operation. Resistive sensors require longer CS0DR operating periods, and their CS0DR settings will be necessarily higher. Higher settings for CS0DR cause the CS0 conversion process to slow substantially. The adjustment method is intended to find the lowest (fastest) CS0DR setting that delivers full function.

1. Begin the adjustment with CS0DR set to maximum delay (CS0DR = 11b). Record a series of CS0 output values for the sensor when it is being touched.

Note: If an ITO touch panel is being tested, this sensor touch should be performed at the sensor location at the end of the longest ITO trace. The series of CS0 output values should be large enough that a reliably repeatable determination of standard deviation can be made, a hundred samples or greater (but typically less than 10,000 samples).

2. For this test, the standard deviation of data in the series is the figure of merit used to define the level of noise received by the CS0 converter. The secondary reset circuit reduces noise. An increase in standard deviation indicates that the secondary reset circuit is no longer working optimally. The best adjustment point for CS0DR is the lowest setting for which there is an acceptably-low standard deviation.
3. Decrease the value of the CS0DR setting by one (from 11b to 10b). Record a new data set and determine its standard deviation. Repeat this process for CS0DR settings 01b and 00b. Compare the standard deviations calculated for the four CS0DR settings. Select the lowest CS0DR setting for which there is not a significant increase in standard deviation.

18.13.4. Adjusting CS0 Ramp Timing: CS0IA

In the presence of larger series resistors between the device pin and the capacitive sensor, it is necessary to also adjust the ramp time for the CS0 conversion. This is done by using CS0IA to modify the source current used to charge up the capacitive sensor. If this source current and the series impedance are both high, the CS0 module will “see” less of the capacitor on the other side of the impedance. Reducing the current allows the pin voltage to more directly reflect the voltage at the capacitive sensor.

The adjustment for CS0IA should be performed while CS0DR and CS0DT bits are set at their maximum values (CS0DR = 11b, CS0DT = 111b).

1. Begin the adjustment with CS0IA set to minimum current (CS0IA = 001b). Measure the untouched average CS0 result for the channel under test.

Note: When calibrating CS0 for use with an ITO panel, consider the use of an artificial finger: a small (¼” O.D.) washer (#2 regular, #4 narrow) wired through a 1000pF capacitor to ground. Select a touch point at the on the farthest end of the longest row. Find the point where maximum response is returned from the CS0 conversion.

2. Record the average touched CS0 value with CS0IA = 001b. The touched value should be higher than the untouched value. The magnitude of the difference between the touched and untouched average CS0 values is the figure of merit for touch sensitivity.
3. Increase the current control CS0IA by one and repeat the touched and untouched CS0 measurements. Repeat this step until values have been recorded for all eight CS0IA settings (including CS0IA = 000b, which is the highest current setting). As CS0IA is changed, the average sensitivity of the CS0 value may begin to decrease significantly.
4. CS0IA should be chosen such that there is not a significant decrease in sensitivity due to resistance. Select the CS0IA setting that occurred prior to the observed drop in CS0 touch sensitivity.

18.13.5. Low-Pass Filter Adjustments

A programmable active low-pass filter is provided to limit external noise interference in CS0 operation. The filter is programmable in two ways. The filter can be tailored for optimal performance with slow-rising signals by adjusting the low-pass filter ramping control, CS0RP. Another control, CS0LP, allows the user to adjust the filter's corner frequency. For most applications, the default settings for the filter controls (CS0LP = 000b, CS0RP = 00b) should be used.

18.13.6. Adjusting CS0 Ramp Timing: CS0RP

Determining the appropriate setting for CS0RP is one of the last in a series of related adjustments to be made for high-resistance loads. It requires that the adjustments for gain (CS0CG), output current (CS0IA), and the reset timing (CS0DT and CS0DR) have already been made. The adjustment values determined for those settings should be programmed into the CS0 module when performing the CS0RP adjustment.

Configure the CS0 module to perform continuously repeated capacitance sensing operations. Using an oscilloscope, measure the maximum rise time seen on the CS0 sensor pin. Subtract 200 ns. This is the CS0 ramp time for this channel.

18.13.7. Adjusting CS0LP for Non-Default CS0RP Settings

The default setting for the low-pass filter corner frequency (CS0LP = 000b) gives the best sensing response for all applications using default ramp timing (CS0RP = 00b). For applications with slower ramp timing, the corner frequency should always be modified to match the edge-rate of the input ramp. For all non-default settings of CS0RP (CS0RP = 01b, 10b or 11b), set CS0LP = 001b.

18.13.8. Other Options for Adjusting CS0LP

In some circumstances, it may be preferable to trade CS0 sensitivity for increased noise filtering. Decreasing the filter's corner frequency below the natural ramp rate of the converter will cause a lower capacitance value to be reported. The change in capacitance due to a touch event will also be attenuated. As a result, lowering the corner frequency will not necessarily increase the signal-to-noise ratio for capacitive touch events.

Although signal-to-noise is the figure of merit for this adjustment, there may be acceptable trade-offs in the adjustment of CS0LP which result in an overall lower SNR but better operation over a wide range of environmental conditions. Some applications may call for adaptive changes to the corner frequency based on measurements of input noise, trading sensitivity for noise rejection only when necessary. Because this optional adjustment requires a subjective trade-off between noise rejection and sensitivity, the ultimate determination of acceptable results for this adjustment will be determined by the end application.

When performing these tests, all other CS0 configuration registers should be properly adjusted for the channel under test. CS0LP operation can only be analyzed when the CS0 is otherwise optimally adjusted. CS0LP adjustments should only be performed during performance tuning for a specific application in a well-defined noise environment.

CS0LP settings adjust the CS0 response to environmental noise. As in the adjustment of CS0DR settings, CS0LP adjustment can only be performed in a test environment with the highest expected level of ambient noise while connected to the sensor which is specific to the intended application.

Higher settings for CS0LP cause the low-pass filter corner frequency to drop. Noise will be reduced and reported capacitance will be reduced. This adjustment process incrementally increases CS0LP settings to determine which, if any, of the settings provide a higher SNR. For this test, the optimum setting for CS0LP will provide higher SNR results for the system in this high-interference environment, although the same setting is likely to reduce SNR for the same system in a low-interference environment.

1. Begin the adjustment with CS0LP set to maximum corner frequency (CS0LP = 000b). Record a series of CS0 output values for the sensor when it is untouched and record another series of CS0 output values from the sensor when it is being touched. If an ITO touch panel is being tested, this sensor touch should be performed at a sensor location on the shortest ITO trace at a location closest to the panel connector. Calculation of SNR may be performed as described in application note AN367, "Understanding Capacitive Sensing Signal to Noise Ratios and Setting Reliable Thresholds".
2. Increase the value of the CS0LP setting by one (from 000b to 001b). Record a new data set and determine its SNR. Repeat this process for all remaining CS0LP settings.

3. This series of tests can be repeated in a variety of noise environments. Comparison of the resulting SNR tables can then be used to determine how CS0LP adjustments might be used to improve capacitive sensing in high-interference environments.

18.14. CS0 Analog Multiplexer

The CS0 input multiplexer can be controlled through two methods. The CS0MX register can be written to through firmware, or the register can be configured automatically using the auto-scan functionality (see Section “18.8. Automatic Scanning (Method 1—CS0SMEN = 0)” on page 132).

Table 18.2. CS0 Input Multiplexer Channels

ADC0MX Setting	Signal Name	QFN-48 Pin Name	QFN-32 Pin Name	QFN-24 Pin Name
000000	CS0.0	P0.0	P0.0	P0.0
000001	CS0.1	P0.1	P0.1	P0.1
000010	CS0.2	P0.2	P0.2	P0.2
000011	CS0.3	P0.3	P0.3	P0.3
000100	CS0.4	P0.4	P0.4	P0.4
000101	CS0.5	P0.5	P0.5	P0.5
000110	CS0.6	P0.6	P0.6	P0.6
000111	CS0.7	P0.7	P0.7	P0.7
001000	CS0.8	P1.0	P1.0	P1.0
001001	CS0.9	P1.1	P1.1	P1.1
001010	CS0.10	P1.2	P1.2	P1.2
001011	CS0.11	P1.3	P1.3	P1.3
001100	CS0.12	P1.4	P1.4	P1.4
001101	CS0.13	P1.5	P1.5	P1.5
001110	CS0.14	P1.6	P1.6	P1.6
001111	CS0.15	P1.7	P1.7	P1.7
010000	CS0.16	P2.0	P2.0	P2.0
010001	CS0.17	P2.1	P2.1	P2.1
010010	CS0.18	P2.2	P2.2	Reserved
010011	CS0.19	P2.3	P2.3	Reserved
010100	CS0.20	P2.4	P2.4	Reserved
010101	CS0.21	P2.5	P2.5	Reserved

Table 18.2. CS0 Input Multiplexer Channels (Continued)

ADC0MX Setting	Signal Name	QFN-48 Pin Name	QFN-32 Pin Name	QFN-24 Pin Name
010110	CS0.22	P2.6	P2.6	Reserved
010111	CS0.23	P2.7	P2.7	Reserved
011000	CS0.24	P3.0	P3.0	Reserved
011001	CS0.25	P3.1	P3.1	Reserved
011010	CS0.26	P3.2	P3.2	Reserved
011011	CS0.27	P3.3	Reserved	Reserved
011100	CS0.28	P3.4	Reserved	Reserved
011101	CS0.29	P3.5	Reserved	Reserved
011110	CS0.30	P3.6	Reserved	Reserved
011111	CS0.31	P3.7	Reserved	Reserved
100000	CS0.32	P4.0	Reserved	Reserved
100001	CS0.33	P4.1	Reserved	Reserved
100010	CS0.34	P4.2	Reserved	Reserved
100011	CS0.35	P4.3	Reserved	Reserved
100100	CS0.36	P4.4	Reserved	Reserved
100101	CS0.37	P4.5	Reserved	Reserved
100110	CS0.38	P4.6	Reserved	Reserved
100111	CS0.39	P4.7	Reserved	Reserved
101000	CS0.40	P5.0	Reserved	Reserved
101001	CS0.41	P5.1	Reserved	Reserved
101010	CS0.42	P5.2	P5.2	P5.2
101011-111111	Reserved	Reserved		

18.14.1. Pin Configuration for CS0 Measurements Method

A port pin selected as CS0 input should be configured as follows:

1. Set to analog mode input by clearing to 0 the corresponding bit in register PnMDIN.
2. Force the Priority Crossbar Decoder to skip the pin by setting 1 to the corresponding bit in register PnSKIP.
3. Enable or disable the auto-ground for the pin by clearing 0 or setting 1 to the corresponding bit in the port latch (Pn), respectively. Auto-grounding means that the pin will be grounded when CS0 measurement is not being performed on the pin.
4. Set to 1 the corresponding bits in AMUX0Pn that CS0 will be taking measurements on.
5. If only a single channel is to be sensed, setup the CS0 Multiplexer to select the appropriate pin for measurement. If automatic scanning is used, setup CS0SS and CS0SE registers. If multiple channels are to be binded, the CS0MC

Important Notes:

- When CS0 is active, ADC0 must not be enabled even if ADC0 is not going to perform an operation.
- Similarly, when ADC0 is active, CS0 should not be enabled.

See Section “26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)” on page 277 for more Port I/O configuration details.

18.15. Capacitive Sense Register

Register 18.1. CS0CN: Capacitive Sense 0 Control

Bit	7	6	5	4	3	2	1	0
Name	CSEN	CSEOS	CSINT	CSBUSY	CSCMPEN	Reserved	CSPME	CSCMPF
Type	RW	R	RW	RW	RW	R	R	R
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xB0 (bit-addressable)								

Bit	Name	Function
7	CSEN	**CS0 Enable.** 0: CS0 disabled and in low-power mode. 1: CS0 enabled and ready to convert.
6	CSEOS	**CS0 End of Scan Interrupt Flag.** This bit must be cleared by firmware. 0: CS0 has not completed a scan since the last time CS0EOS was cleared. 1: CS0 has completed a scan.
5	CSINT	**CS0 Interrupt Flag.** This bit must be cleared by firmware. 0: CS0 has not completed a data conversion since the last time CS0INT was cleared. 1: CS0 has completed a data conversion.
4	CSBUSY	**CS0 Busy.** Read: A 1 indicates a CS0 conversion is in progress. Write: Writing a 1 to this bit initiates a CS0 conversion if CS0CM[2:0] = 000b, 110b, or 111b.
3	CSCMPEN	**CS0 Digital Comparator Enable.** Enables the digital comparator, which compares accumulated CS0 conversion output to the value stored in CS0THH:CS0THL. 0: Disable CS0 digital comparator. 1: Enable CS0 digital comparator.
2	Reserved	Must write reset value.
1	CSPME	**CS0 Pin Monitor Event.** Set if any converter re-tries have occurred due to a pin monitor event. This bit must be cleared by firmware.
0	CSCMPF	**CS0 Digital Comparator Interrupt Flag.** 0: CS0 result is smaller than the value set by CS0THH and CS0THL since the last time CS0CMPF was cleared. 1: CS0 result is greater than the value set by CS0THH and CS0THL since the last time CS0CMPF was cleared.

Register 18.2. CS0CF: Capacitive Sense 0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	CS0SMEN	CS0CM			CS0MCEN	CS0ACU		
Type	RW	RW			R	RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xAA								

Bit	Name	Function
7	CS0SMEN	**CS0 Channel Scan Masking Enable.** 0: The AMUX0 register contents are ignored. 1: The AMUX0 registers determine which channels will be included in the scan.
6:4	CS0CM	**CS0 Start of Conversion Mode Select.** 000: Conversion initiated on every write of 1 to CS0BUSY. 001: Conversion initiated on overflow of Timer 0. 010: Conversion initiated on overflow of Timer 2. 011: Conversion initiated on overflow of Timer 1. 100: Conversion initiated on overflow of Timer 3. 101: When CS0SMEN is set to 1, the converter completes a Single Scan of the channels selected by the AMUX0 registers. This setting is invalid when CS0SMEN is cleared to 0. 110: Conversion initiated continuously on the channel selected by CS0MX after writing 1 to CS0BUSY. 111: When CS0SMEN is set to 1, the converter enters Auto Scan Mode and continuously scans the channels selected by the AMUX0 registers. When CS0SMEN is cleared to 0, the converter scans continuously on channels from CS0SS to CS0SE after firmware writes 1 to CS0BUSY.
3	CS0MCEN	**CS0 Multiple Channel Enable.** 0: Multiple channel feature is disabled. 1: Channels selected by the AMUX0 registers are internally shorted together and the combined node is selected as the CS0 input. This mode can be used to detect a capacitance change on multiple channels using a single conversion.
2:0	CS0ACU	**CS0 Accumulator Mode Select.** 000: Accumulate 1 sample. 001: Accumulate 4 samples. 010: Accumulate 8 samples. 011: Accumulate 16 samples 100: Accumulate 32 samples. 101: Accumulate 64 samples. 110-111: Reserved.

Register 18.3. CS0DH: Capacitive Sense 0 Data High Byte

Bit	7	6	5	4	3	2	1	0
Name	CS0DH							
Type	R							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xEE								

Bit	Name	Function
7:0	CS0DH	**CS0 Data High Byte.** Stores the high byte of the last completed 16-bit Capacitive Sense conversion.

Register 18.4. CS0DL: Capacitive Sense 0 Data Low Byte

Bit	7	6	5	4	3	2	1	0
Name	CS0DL							
Type	R							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xED								

Bit	Name	Function
7:0	CS0DL	**CS0 Data Low Byte.** Stores the low byte of the last completed 16-bit Capacitive Sense conversion.

Register 18.5. CS0SS: Capacitive Sense 0 Auto Scan Start Channel

Bit	7	6	5	4	3	2	1	0
Name	Reserved		CS0SS					
Type	R		RW					
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDD								

Bit	Name	Function
7:6	Reserved	Must write reset value.
5:0	CS0SS	**Starting Channel for Auto Scan.** Sets the first CS0 channel to be selected by the mux for Capacitive Sense conversion when Auto Scan is enabled and active. All channels detailed in CS0MX are possible choices for this register. When Auto Scan is enabled, a write to CS0SS will also update CS0MX.

Register 18.6. CS0SE: Capacitive Sense 0 Auto Scan End Channel

Bit	7	6	5	4	3	2	1	0
Name	Reserved		CS0SE					
Type	R		RW					
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDE								

Bit	Name	Function
7:6	Reserved	Must write reset value.
5:0	CS0SE	**Ending Channel for Auto Scan.** Sets the last CS0 channel to be selected by the mux for Capacitive Sense conversion when Auto Scan is enabled and active. All channels detailed in CS0MX are possible choices for this register.

Register 18.7. CS0THH: Capacitive Sense 0 Comparator Threshold High Byte

Bit	7	6	5	4	3	2	1	0
Name	CS0THH							
Type	RW							
Reset	X	X	X	X	X	X	X	X
SFR Page = 0x0; SFR Address: 0xFE								

Bit	Name	Function
7:0	CS0THH	**CS0 Comparator Threshold High Byte.** High byte of the 16-bit value compared to the Capacitive Sense conversion result.

Register 18.8. CS0THL: Capacitive Sense 0 Comparator Threshold Low Byte

Bit	7	6	5	4	3	2	1	0
Name	CS0THL							
Type	RW							
Reset	X	X	X	X	X	X	X	X
SFR Page = 0x0; SFR Address: 0xFD								

Bit	Name	Function
7:0	CS0THL	**CS0 Comparator Threshold Low Byte.** Low byte of the 16-bit value compared to the Capacitive Sense conversion result.

Register 18.9. CS0MD1: Capacitive Sense 0 Mode 1

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CS0POL	CS0DR		CS0WOI	CS0CG		
Type	RW	R	R		R	RW		
Reset	0	0	0	0	0	1	1	1
SFR Page = 0x0; SFR Address: 0xBD								

Bit	Name	Function
7	Reserved	Must write reset value.
6	CS0POL	**CS0 Digital Comparator Polarity Select.** 0: The digital comparator generates an interrupt if the conversion is greater than the threshold. 1: The digital comparator generates an interrupt if the conversion is less than or equal to the threshold.
5:4	CS0DR	**CS0 Double Reset Select.** These bits adjust the secondary CS0 reset time. For most touch-sensitive switches, the default (fastest) value is sufficient. 00: No additional time is used for secondary reset. 01: An additional 0.75 us is used for secondary reset. 10: An additional 1.5 us is used for secondary reset. 11: An additional 2.25 us is used for secondary reset.
3	CS0WOI	**CS0 Wake on Interrupt Configuration.** 0: Wake-up event generated on digital comparator interrupt only. 1: Wake-up event generated on end of scan or digital comparator interrupt.
2:0	CS0CG	**CS0 Capacitance Gain Select.** These bits select the gain applied to the capacitance measurement. Lower gain values increase the size of the capacitance that can be measured with the CS0 module. The capacitance gain is equivalent to CS0CG[2:0] + 1. 000: Gain = 1x 001: Gain = 2x 010: Gain = 3x 011: Gain = 4x 100: Gain = 5x 101: Gain = 6x 110: Gain = 7x 111: Gain = 8x

Register 18.10. CS0MD2: Capacitive Sense 0 Mode 2

Bit	7	6	5	4	3	2	1	0
Name	CS0CR		CS0DT			CS0IA		
Type	RW		RW			RW		
Reset	0	1	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xBE								

Bit	Name	Function
7:6	CS0CR	**CS0 Conversion Rate.** These bits control the conversion rate of the CS0 module. 00: Conversions last 12 internal CS0 clocks and are 12 bits in length. 01: Conversions last 13 internal CS0 clocks and are 13 bits in length. 10: Conversions last 14 internal CS0 clocks and are 14 bits in length. 11: Conversions last 16 internal CS0 clocks.and are 16 bits in length.
5:3	CS0DT	**CS0 Discharge Time.** These bits adjust the primary CS0 reset time. For most touch-sensitive switches, the default (fastest) value is sufficient. 000: Discharge time is 0.75 us (recommended for most switches). 001: Discharge time is 1.0 us. 010: Discharge time is 1.2 us. 011: Discharge time is 1.5 us. 100: Discharge time is 2 us. 101: Discharge time is 3 us. 110: Discharge time is 6 us. 111: Discharge time is 12 us.
2:0	CS0IA	**CS0 Output Current Adjustment.** These bits adjust the output current used to charge up the capacitive sensor element. For most touch-sensitive switches, the default (highest) current is sufficient. 000: Full current. 001: 1/8 current. 010: 1/4 current. 011: 3/8 current. 100: 1/2 current. 101: 5/8 current. 110: 3/4 current. 111: 7/8 current.

Register 18.11. CS0MD3: Capacitive Sense 0 Mode 3

Bit	7	6	5	4	3	2	1	0
Name	Reserved			CS0RP		CS0LP		
Type	RW			RW		RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xBF								

Bit	Name	Function
7:5	Reserved	Must write reset value.
4:3	CS0RP	**CS0 Ramp Selection.** These bits are used to compensate CS0 conversions for circuits requiring slower ramp times. For most touch-sensitive switches, the default (fastest) value is sufficient. 00: Ramp time is less than 1.5 us. 01: Ramp time is between 1.5 us and 3 us. 10: Ramp time is between 3 us and 6 us. 11: Ramp time is greater than 6 us.
2:0	CS0LP	**CS0 Low Pass Filter Selection.** These bits set the internal corner frequency of the CS0 low-pass filter. Higher values of CS0LP result in a lower internal corner frequency. For most touch-sensitive switches, the default setting of 000b should be used. If the CS0RP bits are adjusted from their default value, the CS0LP bits should normally be set to 001b. Settings higher than 001b will result in attenuated readings from the CS0 module and should be used only under special circumstances.

Register 18.12. CS0PM: Capacitive Sense 0 Pin Monitor

Bit	7	6	5	4	3	2	1	0
Name	UAPM	SPIPM	SMBPM	PCAPM	PIOPM	I2C0PM	CSPMMD	
Type	RW	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x96								

Bit	Name	Function
7	UAPM	**UART Pin Monitor Enable.** Enables monitoring of the UART TX pin.
6	SPIPM	**SPI Pin Monitor Enable.** Enables monitoring SPI output pins.
5	SMBPM	**SMBus Pin Monitor Enable.** Enables monitoring of the SMBus pins.
4	PCAPM	**PCA Pin Monitor Enable.** Enables monitoring of PCA output pins.
3	PIOPM	**Port I/O Pin Monitor Enable.** Enables monitoring of writes to the port latch registers.
2	I2C0PM	**I2C Slave Pin Monitor Enable.** Enables monitoring of the I2C Slave output pin.
1:0	CSPMMD	**CS0 Pin Monitor Mode.** Selects the operation to take when a monitored signal changes state. 00: Always retry bit cycles on a pin state change. 01: Retry up to twice on consecutive bit cycles. 10: Retry up to four times on consecutive bit cycles. 11: Reserved.

Register 18.13. CS0MX: Capacitive Sense 0 Mux Channel Select

Bit	7	6	5	4	3	2	1	0
Name	CS0UC	Reserved	CS0MX					
Type	RW	RW	RW					
Reset	1	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xAB								

Bit	Name	Function
7	CS0UC	**CS0 Connect.** Disconnects CS0 from all port pins, regardless of the selected channel. 0: CS0 connected to port pins. 1: CS0 disconnected from port pins.
6	Reserved	Must write reset value.

Bit	Name	Function
5:0	CS0MX	**CS0 Mux Channel Select.** Selects a single input channel for Capacitive Sense conversion. 000000: Select CS0.0. 000001: Select CS0.1. 000010: Select CS0.2. 000011: Select CS0.3. 000100: Select CS0.4. 000101: Select CS0.5. 000110: Select CS0.6. 000111: Select CS0.7. 001000: Select CS0.8. 001001: Select CS0.9. 001010: Select CS0.10. 001011: Select CS0.11. 001100: Select CS0.12. 001101: Select CS0.13. 001110: Select CS0.14. 001111: Select CS0.15. 010000: Select CS0.16. 010001: Select CS0.17. 010010: Select CS0.18. 010011: Select CS0.19. 010100: Select CS0.20. 010101: Select CS0.21. 010110: Select CS0.22. 010111: Select CS0.23. 011000: Select CS0.24. 011001: Select CS0.25. 011010: Select CS0.26. 011011: Select CS0.27. 011100: Select CS0.28. 011101: Select CS0.29. 011110: Select CS0.30. 011111: Select CS0.31. 100000: Select CS0.32. 100001: Select CS0.33. 100010: Select CS0.34. 100011: Select CS0.35. 100100: Select CS0.36. 100101: Select CS0.37. 100110: Select CS0.38. 100111: Select CS0.39. 101000: Select CS0.40. 101001: Select CS0.41. 101010: Select CS0.42. 101011-111111: Reserved.

19. Analog Multiplexer (AMUX0)

The analog multiplexer (AMUX0) module connects physical device pins with analog peripherals, like the ADC and Capacitive Sensing modules.

For the ADC, the AMUX0 module selects the single channel for ADC measurement. Only one AMUX0 input should be enabled at a time when using ADC0 with AMUX0 (ADC0MX = 0). See “Analog-to-Digital Converter (ADC0)” on page 104 for more information on configuring pins for use with the ADC.

For the CS0 module, the AMUX0 module selects all channels that should be measured together using channel binding or scanned individually using single- or auto-scan modes. See “Capacitive Sense (CS0)” on page 130 for more information on configuring pins for use with the CS0 module.

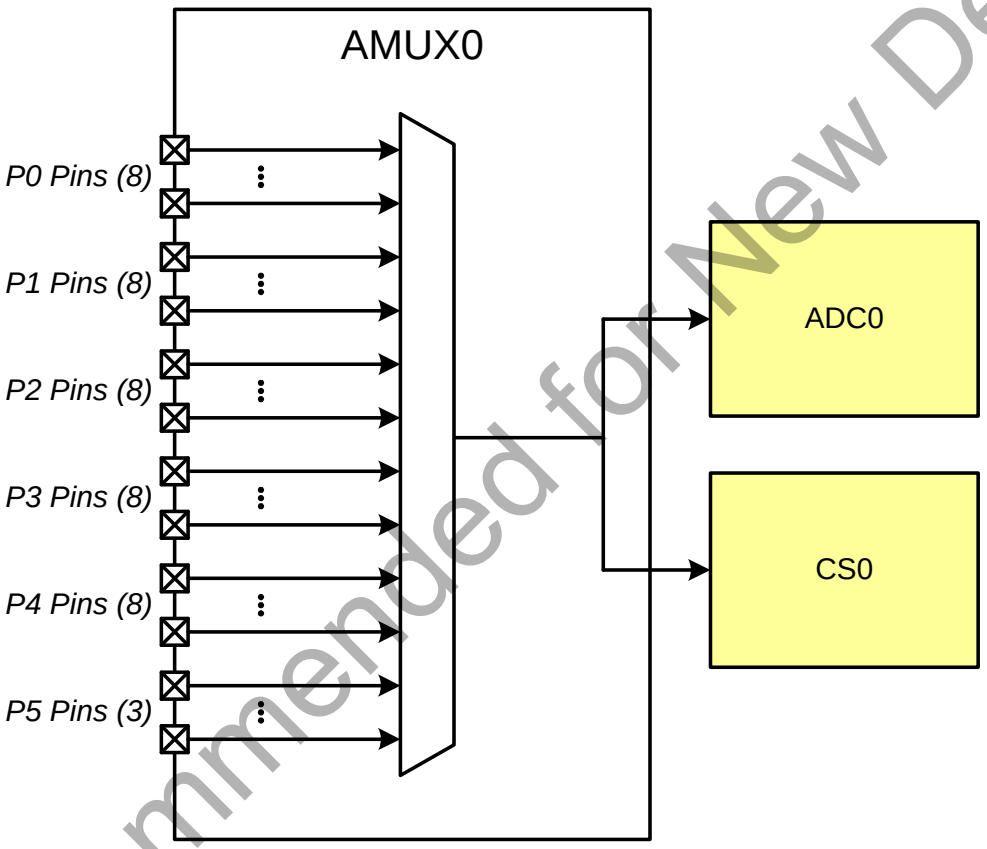


Figure 19.1. AMUX0 Block Diagram

19.1. AMUX Control Registers

Register 19.1. AMUX0P0: Port 0 Analog Multiplexer Control

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x9A								

Table 19.1. AMUX0P0 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Analog Multiplexor Control.** 0: P0.7 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.7 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
6	B6	**Port 0 Bit 6 Analog Multiplexor Control.** 0: P0.6 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.6 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
5	B5	**Port 0 Bit 5 Analog Multiplexor Control.** 0: P0.5 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.5 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
4	B4	**Port 0 Bit 4 Analog Multiplexor Control.** 0: P0.4 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.4 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
3	B3	**Port 0 Bit 3 Analog Multiplexor Control.** 0: P0.3 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.3 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
2	B2	**Port 0 Bit 2 Analog Multiplexor Control.** 0: P0.2 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.2 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
1	B1	**Port 0 Bit 1 Analog Multiplexor Control.** 0: P0.1 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.1 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
0	B0	**Port 0 Bit 0 Analog Multiplexor Control.** 0: P0.0 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P0.0 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

Register 19.2. AMUX0P1: Port 1 Analog Multiplexer Control

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x9B								

Table 19.2. AMUX0P1 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Analog Multiplexor Control.** 0: P1.7 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.7 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
6	B6	**Port 1 Bit 6 Analog Multiplexor Control.** 0: P1.6 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.6 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
5	B5	**Port 1 Bit 5 Analog Multiplexor Control.** 0: P1.5 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.5 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
4	B4	**Port 1 Bit 4 Analog Multiplexor Control.** 0: P1.4 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.4 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
3	B3	**Port 1 Bit 3 Analog Multiplexor Control.** 0: P1.3 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.3 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
2	B2	**Port 1 Bit 2 Analog Multiplexor Control.** 0: P1.2 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.2 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
1	B1	**Port 1 Bit 1 Analog Multiplexor Control.** 0: P1.1 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.1 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
0	B0	**Port 1 Bit 0 Analog Multiplexor Control.** 0: P1.0 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P1.0 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

Register 19.3. AMUX0P2: Port 2 Analog Multiplexer Control

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x9C								

Table 19.3. AMUX0P2 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Analog Multiplexor Control.** 0: P2.7 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.7 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
6	B6	**Port 2 Bit 6 Analog Multiplexor Control.** 0: P2.6 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.6 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
5	B5	**Port 2 Bit 5 Analog Multiplexor Control.** 0: P2.5 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.5 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
4	B4	**Port 2 Bit 4 Analog Multiplexor Control.** 0: P2.4 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.4 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
3	B3	**Port 2 Bit 3 Analog Multiplexor Control.** 0: P2.3 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.3 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
2	B2	**Port 2 Bit 2 Analog Multiplexor Control.** 0: P2.2 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.2 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
1	B1	**Port 2 Bit 1 Analog Multiplexor Control.** 0: P2.1 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.1 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
0	B0	**Port 2 Bit 0 Analog Multiplexor Control.** 0: P2.0 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P2.0 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

Register 19.4. AMUX0P3: Port 3 Analog Multiplexer Control

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xA4								

Table 19.4. AMUX0P3 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 3 Bit 7 Analog Multiplexor Control.** 0: P3.7 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.7 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
6	B6	**Port 3 Bit 6 Analog Multiplexor Control.** 0: P3.6 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.6 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
5	B5	**Port 3 Bit 5 Analog Multiplexor Control.** 0: P3.5 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.5 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
4	B4	**Port 3 Bit 4 Analog Multiplexor Control.** 0: P3.4 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.4 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
3	B3	**Port 3 Bit 3 Analog Multiplexor Control.** 0: P3.3 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.3 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
2	B2	**Port 3 Bit 2 Analog Multiplexor Control.** 0: P3.2 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.2 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
1	B1	**Port 3 Bit 1 Analog Multiplexor Control.** 0: P3.1 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.1 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
0	B0	**Port 3 Bit 0 Analog Multiplexor Control.** 0: P3.0 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P3.0 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

Register 19.5. AMUX0P4: Port 4 Analog Multiplexer Control

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xA5								

Table 19.5. AMUX0P4 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Analog Multiplexor Control.** 0: P4.7 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.7 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
6	B6	**Port 4 Bit 6 Analog Multiplexor Control.** 0: P4.6 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.6 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
5	B5	**Port 4 Bit 5 Analog Multiplexor Control.** 0: P4.5 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.5 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
4	B4	**Port 4 Bit 4 Analog Multiplexor Control.** 0: P4.4 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.4 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
3	B3	**Port 4 Bit 3 Analog Multiplexor Control.** 0: P4.3 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.3 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
2	B2	**Port 4 Bit 2 Analog Multiplexor Control.** 0: P4.2 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.2 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
1	B1	**Port 4 Bit 1 Analog Multiplexor Control.** 0: P4.1 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.1 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
0	B0	**Port 4 Bit 0 Analog Multiplexor Control.** 0: P4.0 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P4.0 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

Register 19.6. AMUX0P5: Port 5 Analog Multiplexer Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved					B2	B1	B0
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xA6								

Table 19.6. AMUX0P5 Register Bit Descriptions

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	B2	**Port 5 Bit 2 Analog Multiplexor Control.** 0: P5.2 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P5.2 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
1	B1	**Port 5 Bit 1 Analog Multiplexor Control.** 0: P5.1 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P5.1 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
0	B0	**Port 5 Bit 0 Analog Multiplexor Control.** 0: P5.0 pin is not connected by the Analog Multiplexer for CS0 or ADC0 measurements. 1: P5.0 pin is connected by the Analog Multiplexer for CS0 or ADC0 measurements.
Note: If any of the internal signals are selected as the ADC0 input channel, the AMUX0 registers must not select an external pin and all be cleared to 0.		

20. CIP-51 Microcontroller Core

The C8051F97x uses 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. The CIP-51 also includes on-chip debug hardware 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 20.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput with 25 MHz Clock
- 0 to 25 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

20.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. 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.

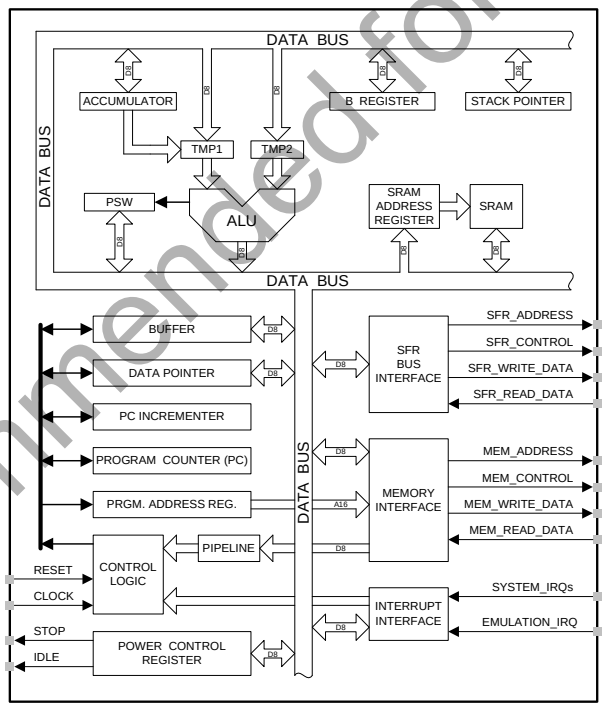


Figure 20.1. CIP-51 Block Diagram

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 require each execution time.

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

20.2. 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).

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

The CIP-51 is supported by development tools from Silicon Labs and third party vendors. Silicon Labs provides an integrated development environment (IDE) including editor, 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. Third party macro assemblers and C compilers are also available.

20.3. Instruction Set

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

20.3.1. Instruction and CPU Timing

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

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 20.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

Table 20.1. CIP-51 Instruction Set Summary

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

Table 20.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
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

Table 20.1. CIP-51 Instruction Set Summary

Mnemonic	Description	Bytes	Clock Cycles
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/3
JNC rel	Jump if Carry is not set	2	2/3
JB bit, rel	Jump if direct bit is set	3	3/4
JNB bit, rel	Jump if direct bit is not set	3	3/4
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/4
Program Branching			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	5
RETI	Return from interrupt	1	5
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative address)	2	3
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
JZ rel	Jump if A equals zero	2	2/3
JNZ rel	Jump if A does not equal zero	2	2/3
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/4
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/4
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/4
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/5
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/3
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/4
NOP	No operation	1	1

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 (twos 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 2 kB 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 8 kB program memory space.

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

20.4. SFR Paging

The CIP-51 features SFR paging, allowing the device to map many SFRs into the 0x80 to 0xFF memory address space. The SFR memory space has 256 pages. In this way, each memory location from 0x80 to 0xFF can access up to 256 SFRs. The C8051F97x family of devices utilizes two SFR pages: 0x00 and 0x0F. SFR pages are selected using the Special Function Register Page register, SFRPAGE. The procedure for reading and writing an SFR is as follows:

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

20.5. Interrupts and SFR Paging

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

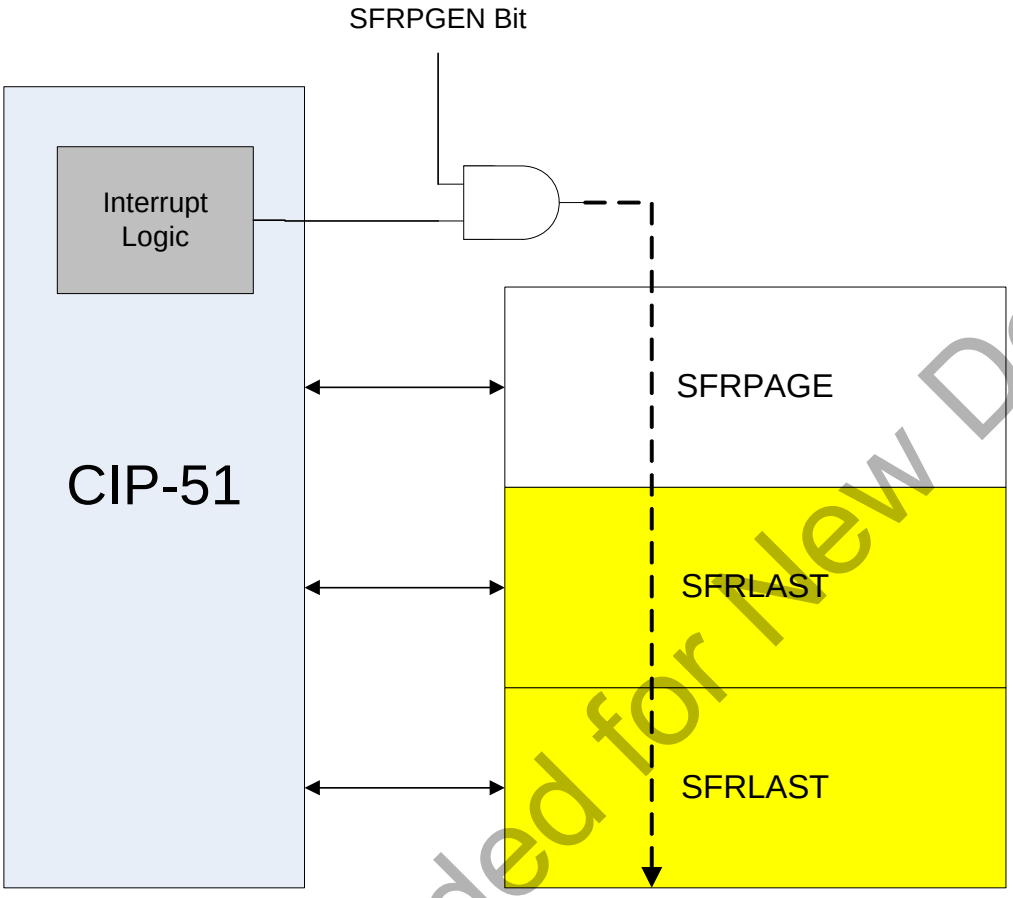


Figure 20.2. SFR Page Stack

Automatic hardware switching of the SFR page on interrupts may be enabled or disabled as desired using the SFR Automatic Page Control Enable Bit located in the SFR page Control Register (SFRPGCN). This function defaults to “enabled” upon reset. In this way, the auto-switching function will be enabled unless disabled in software.

A summary of the SFR locations (address and SFR page) are provided in Table 9.1 on page 57 and Table 9.2 on page 58. Each memory location in the map has an SFR page row, denoting the page in which that SFR resides. Certain SFRs are accessible from all SFR pages. For example, the Port I/O registers P0, P1, P2, and P3 are available on all pages regardless of the SFRPAGE register value.

20.6. SFR Page Stack Example

This example demonstrates the operation of the SFR page stack during interrupts. In this example, the SFR control register is left in the default enabled state (i.e., SFRPGEN = 1), and the CIP-51 is executing in-line code that is writing values to I2C0STAT I2C Slave 0 status register (located at address 0xF8 on SFR page 0x0F). The device is also using the SPI peripheral (SPI0) and the Programmable Counter Array (PCA0) peripheral to generate a PWM output. The PCA is timing a critical control function in its interrupt service routine, and so its associated ISR is set to high priority. At this point, the SFR page is set to access the I2C0STAT SFR (SFRPAGE = 0x0F). See Figure 20.3.

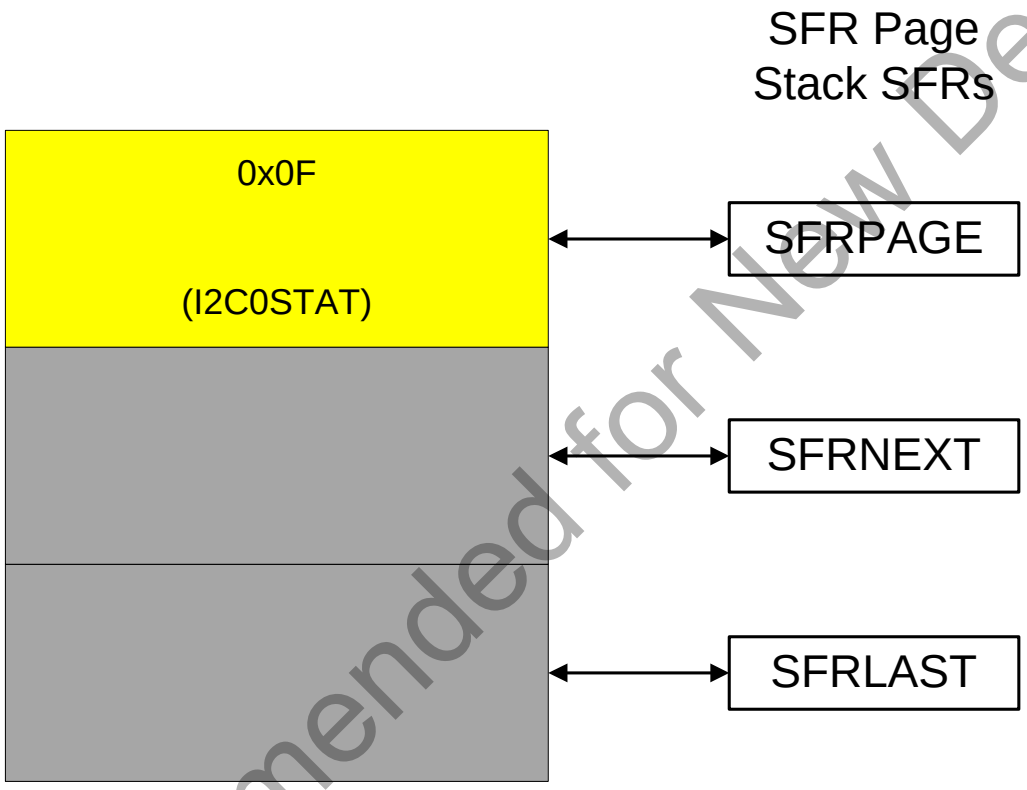


Figure 20.3. SFR Page Stack While Using SFR Page 0x0F To Access I2C0STAT

While CIP-51 executes in-line code (reading a value from I2C0STAT in this example), the SPI0 interrupt occurs. The CIP-51 vectors to the SPI0 ISR and pushes the current SFR page value (SFR page 0x0F) into SFRNEXT in the SFR page stack. The SFR page needed to access SPI0's SFRs is then automatically placed in the SFRPAGE register (SFR page 0x00). SFRPAGE is considered the top of the SFR page stack. Firmware can now access the SPI0 SFRs. Software may switch to any SFR page by writing a new value to the SFRPAGE register at any time during the SPI0 ISR to access SFRs that are not on SFR page 0x00. See Figure 20.4.

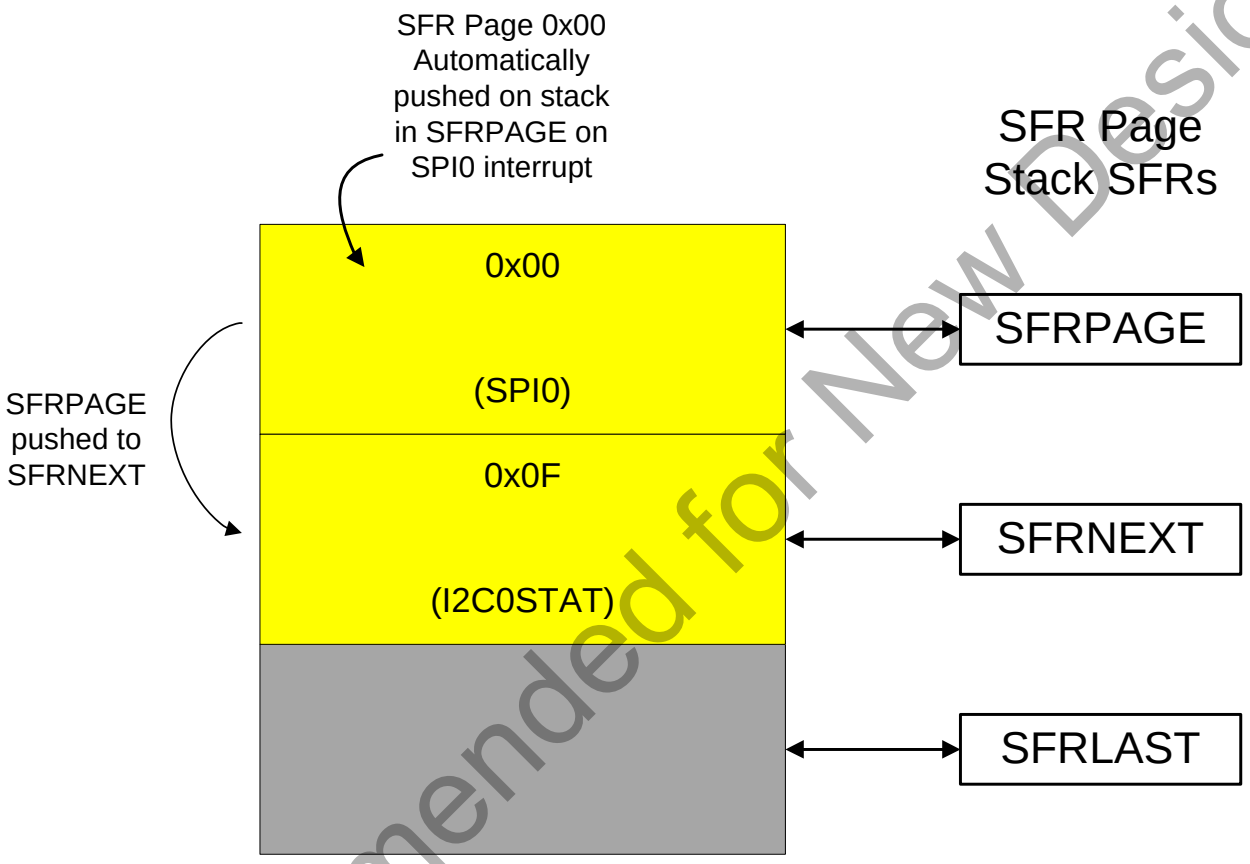


Figure 20.4. SFR Page Stack after SPI0 Interrupt Occurs

While in the SPI0 ISR, a PCA interrupt occurs. Recall the PCA interrupt is configured as a high priority interrupt, while the SPI0 interrupt is configured as a low priority interrupt. Thus, the CIP-51 will now vector to the high priority PCA ISR. Upon doing so, the CIP-51 will automatically place the SFR page needed to access the PCA's special function registers into the SFRPAGE register, SFR page 0x00. The value that was in the SFRPAGE register before the PCA interrupt (SFR page 0x00 for SPI0) is pushed down the stack into SFRNEXT. Likewise, the value that was in the SFRNEXT register before the PCA interrupt (in this case SFR page 0x0F for I2C0STAT) is pushed down to the SFRLAST register, the bottom of the stack. Note that a value stored in SFRLAST (via a previous software write to the SFRLAST register) will be overwritten. See Figure 20.5.

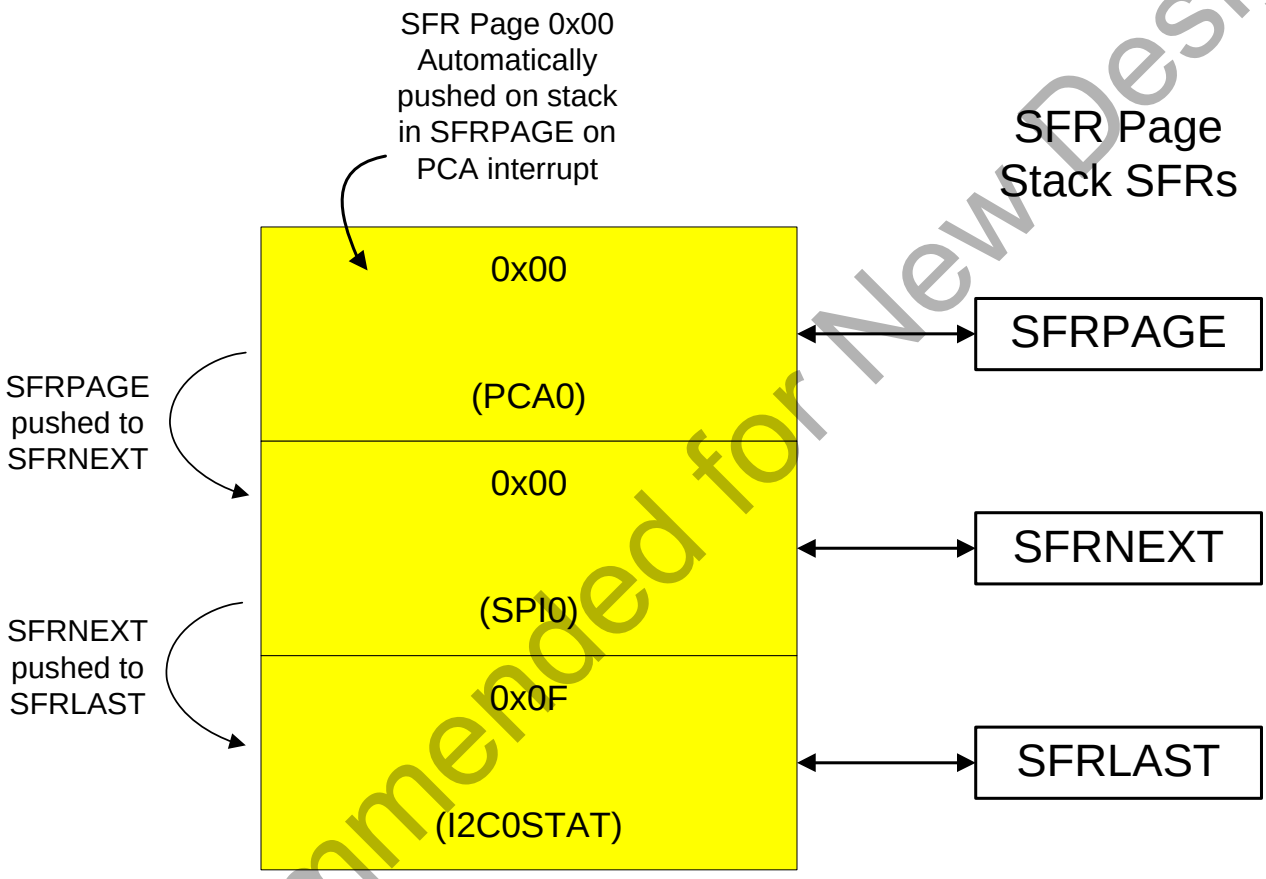


Figure 20.5. SFR Page Stack Upon PCA Interrupt Occurring During a SPI0 ISR

On exit from the PCA interrupt service routine, the CIP-51 will return to the SPI0 ISR. On execution of the RETI instruction, SFR page 0x00 used to access the PCA registers will be automatically popped off of the SFR page stack, and the contents of the SFRNEXT register will be moved to the SFRPAGE register. Firmware in the SPI0 ISR can continue to access SFRs as it did prior to the PCA interrupt. Likewise, the contents of SFRLAST are moved to the SFRNEXT register. Recall this was the SFR page value 0x0F being used to access I2C0STAT before the SPI0 interrupt occurred. See Figure 20.6.

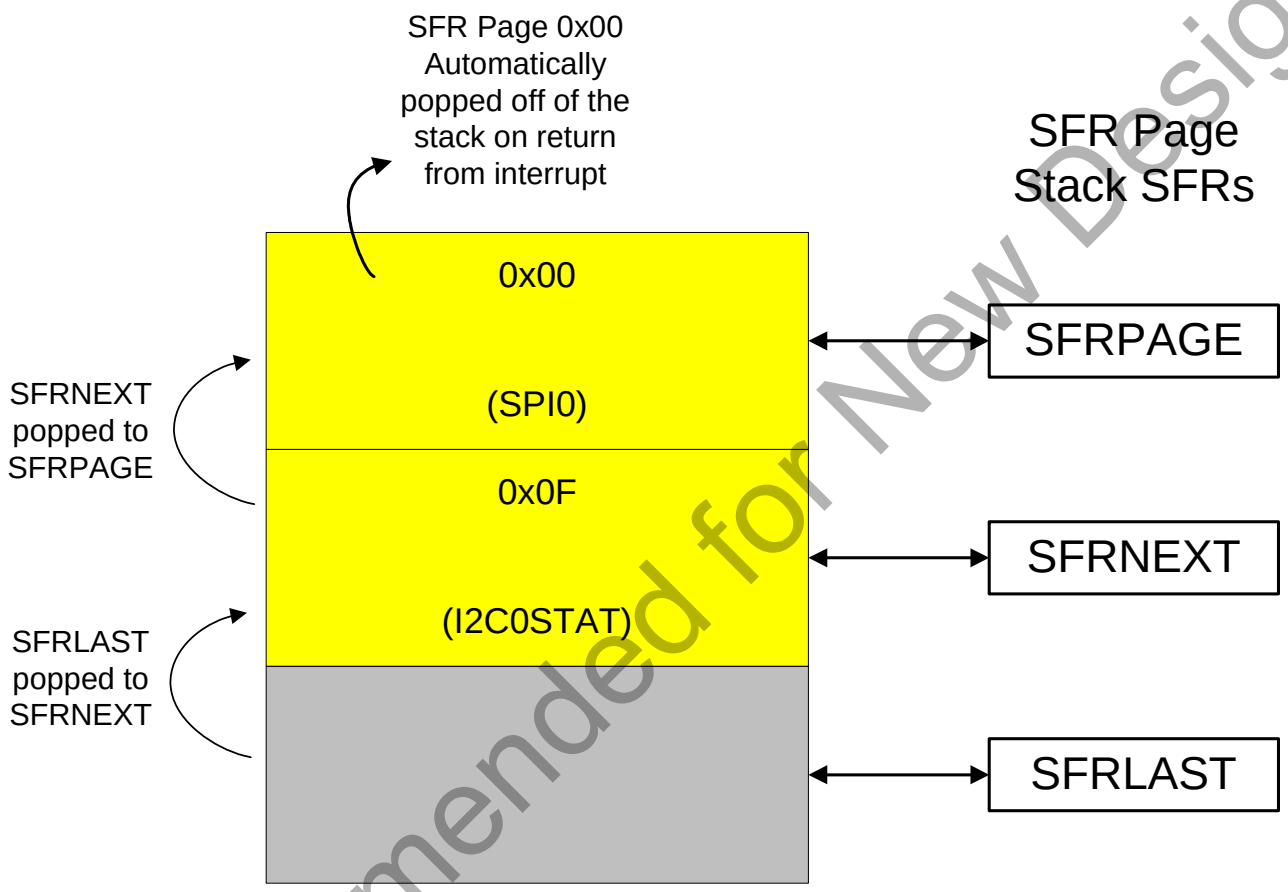


Figure 20.6. SFR Page Stack Upon Return From PCA Interrupt

On the execution of the RETI instruction in the SPI0 ISR, the value in SFRPAGE register is overwritten with the contents of SFRNEXT. The CIP-51 may now access the I2C0STAT register as it did prior to the interrupts occurring. See Figure 20.7.

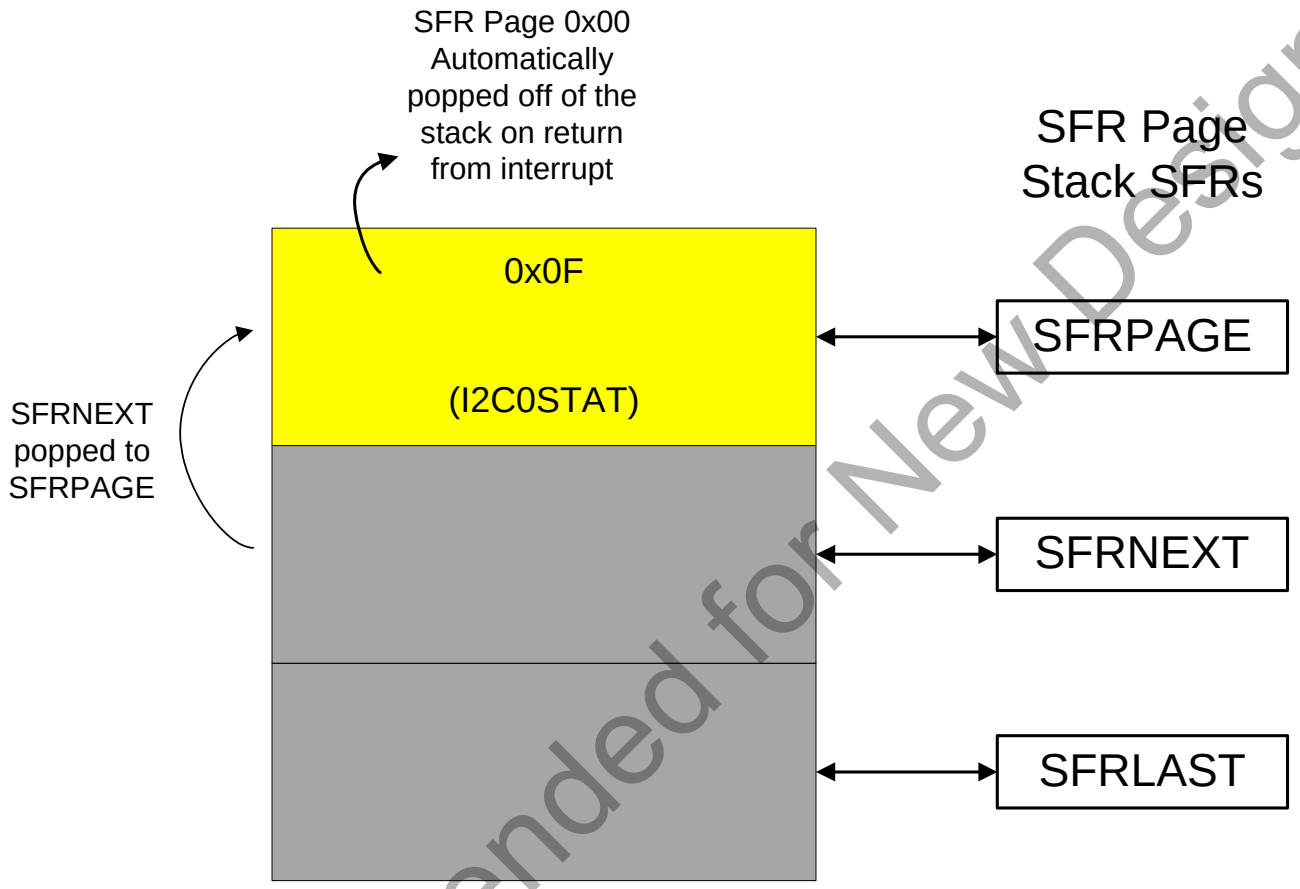


Figure 20.7. SFR Page Stack Upon Return From SPI0 Interrupt

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

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

20.7. CPU Core Registers

Register 20.1. DPL: Data Pointer Low

Bit	7	6	5	4	3	2	1	0
Name	DPL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0x82								

Table 20.2. DPL Register Bit Descriptions

Bit	Name	Function
7:0	DPL	**Data Pointer Low.** The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed flash memory or XRAM.

Register 20.2. DPH: Data Pointer High

Bit	7	6	5	4	3	2	1	0
Name	DPH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0x83								

Table 20.3. DPH Register Bit Descriptions

Bit	Name	Function
7:0	DPH	**Data Pointer High.** The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed flash memory or XRAM.

Register 20.3. SP: Stack Pointer

Bit	7	6	5	4	3	2	1	0
Name	SP							
Type	RW							
Reset	0	0	0	0	0	1	1	1
SFR Page = ALL; SFR Address: 0x81								

Table 20.4. SP Register Bit Descriptions

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

Register 20.4. ACC: Accumulator

Bit	7	6	5	4	3	2	1	0
Name	ACC							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xE0 (bit-addressable)								

Table 20.5. ACC Register Bit Descriptions

Bit	Name	Function
7:0	ACC	**Accumulator.** This register is the accumulator for arithmetic operations.

Register 20.5. B: B Register

Bit	7	6	5	4	3	2	1	0
Name	B							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xF0 (bit-addressable)								

Table 20.6. B Register Bit Descriptions

Bit	Name	Function
7:0	B	**B Register.** This register serves as a second accumulator for certain arithmetic operations.

Register 20.6. PSW: Program Status Word

Bit	7	6	5	4	3	2	1	0
Name	CY	AC	F0	RS		OV	F1	PARITY
Type	RW	RW	RW	RW		RW	RW	R
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xD0 (bit-addressable)								

Table 20.7. PSW Register Bit Descriptions

Bit	Name	Function
7	CY	**Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to logic 0 by all other arithmetic operations.
6	AC	**Auxiliary Carry Flag.** This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to logic 0 by all other arithmetic operations.
5	F0	**User Flag 0.** This is a bit-addressable, general purpose flag for use under firmware control.
4:3	RS	**Register Bank Select.** These bits select which register bank is used during register accesses. 00: Bank 0, Addresses 0x00-0x07 01: Bank 1, Addresses 0x08-0x0F 10: Bank 2, Addresses 0x10-0x17 11: Bank 3, Addresses 0x18-0x1F
2	OV	**Overflow Flag.** This bit is set to 1 under the following circumstances: 1. An ADD, ADDC, or SUBB instruction causes a sign-change overflow. 2. A MUL instruction results in an overflow (result is greater than 255). 3. 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.
1	F1	**User Flag 1.** This is a bit-addressable, general purpose flag for use under firmware control.
0	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.

Register 20.7. SFRPAGE: SFR Page

Bit	7	6	5	4	3	2	1	0
Name	SFRPAGE							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xA7								

Table 20.8. SFRPAGE Register Bit Descriptions

Bit	Name	Function
7:0	SFRPAGE	**SFR Page.** Specifies the SFR Page used when reading, writing, or modifying special function registers.

Register 20.8. SFRPGCN: SFR Page Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved							SFRPGEN
Type	R							RW
Reset	0	0	0	0	0	0	0	1
SFR Page = 0xF; SFR Address: 0xA6								

Table 20.9. SFRPGCN Register Bit Descriptions

Bit	Name	Function
7:1	Reserved	Must write reset value.
0	SFRPGEN	**SFR Automatic Page Control Enable.** Upon interrupt, the C8051 core will vector to the specified interrupt service routine and automatically switch the SFR page to the corresponding peripheral or function's SFR page. This bit is used to control this autopaging function. 0: Disable SFR automatic paging. The C8051 core will not automatically change to the appropriate SFR page (i.e., the SFR page that contains the SFRs for the peripheral/function that was the source of the interrupt). 1: Enable SFR automatic paging. Upon interrupt, the C8051 will switch the SFR page to the page that contains the SFRs for the peripheral or function that is the source of the interrupt.

Register 20.9. SFRNEXT: SFR Page Next

Bit	7	6	5	4	3	2	1	0
Name	SFRNEXT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xF3								

Table 20.10. SFRNEXT Register Bit Descriptions

Bit	Name	Function
7:0	SFRNEXT	**SFR Page Next.** This is the value that will go to the SFR Page register upon a return from interrupt. Write: Sets the SFR Page contained in the second byte of the SFR Stack. This will cause the SFRPAGE SFR to have this SFR page value upon a return from interrupt. Read: Returns the value of the SFR page contained in the second byte of the SFR stack. SFR page context is retained upon interrupts/return from interrupts in a 3-byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to push or pop. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

Register 20.10. SFRLAST: SFR Page Last

Bit	7	6	5	4	3	2	1	0
Name	SFRLAST							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xB3								

Table 20.11. SFRLAST Register Bit Descriptions

Bit	Name	Function
7:0	SFRLAST	**SFR Page Last.** This is the value that will go to the SFRNEXT register upon a return from interrupt. Write: Sets the SFR Page in the last entry of the SFR Stack. This will cause the SFRNEXT SFR to have this SFR page value upon a return from interrupt. Read: Returns the value of the SFR page contained in the last entry of the SFR stack. SFR page context is retained upon interrupts/return from interrupts in a 3-byte SFR Page Stack: SFRPAGE is the first entry, SFRNEXT is the second, and SFRLAST is the third entry. The SFR stack bytes may be used alter the context in the SFR Page Stack, and will not cause the stack to push or pop. Only interrupts and return from interrupts cause pushes and pops of the SFR Page Stack.

21. Direct Memory Access (DMA0)

An on-chip direct memory access (DMA0) is included on the C8051F97x devices. The DMA0 subsystem allows unattended variable-length data transfers between XRAM and peripheral SFR registers without CPU intervention. During DMA0 operation, the CPU is free to perform some other tasks. In order to save total system power consumption, the CPU and Flash can be powered down. DMA0 improves the system performance and efficiency with high data throughput peripherals.

DMA0 contains seven independent channels, common control registers, and a DMA0 engine (see Figure 21.1). Each channel includes a register that assigns a peripheral to the channel, a channel control register, and a set of SFRs that include XRAM address information and SFR address information used by the channel during a data transfer. The DMA0 architecture is described in detail in “21.1. DMA0 Architecture” .

The DMA0 in C8051F97x devices supports two peripherals: MAC0 and I2C0. Peripherals with DMA0 capability should be configured to work with the DMA0 through their own registers. The DMA0 provides up to seven channels, and each channel can be configured for one of eight possible data transfer functions:

- XRAM to MAC A registers (MAC0AH, MAC0AL).
- XRAM to MAC B registers (MAC0BH, MAC0BL).
- XRAM to MAC accumulator registers (MAC0ACCn).
- MAC accumulator registers (MAC0ACCn) to XRAM.
- I2C Slave 0 received data register (I2C0DIN) to XRAM.
- XRAM to I2C Slave 0 transmit data register (I2C0DOUT).

Functions 5 and 6 are mutually exclusive.

The DMA0 subsystem signals the MCU through a set of interrupt service routine flags. Interrupts can be generated when the DMA0 transfers half of the data length or full data length on any channel.

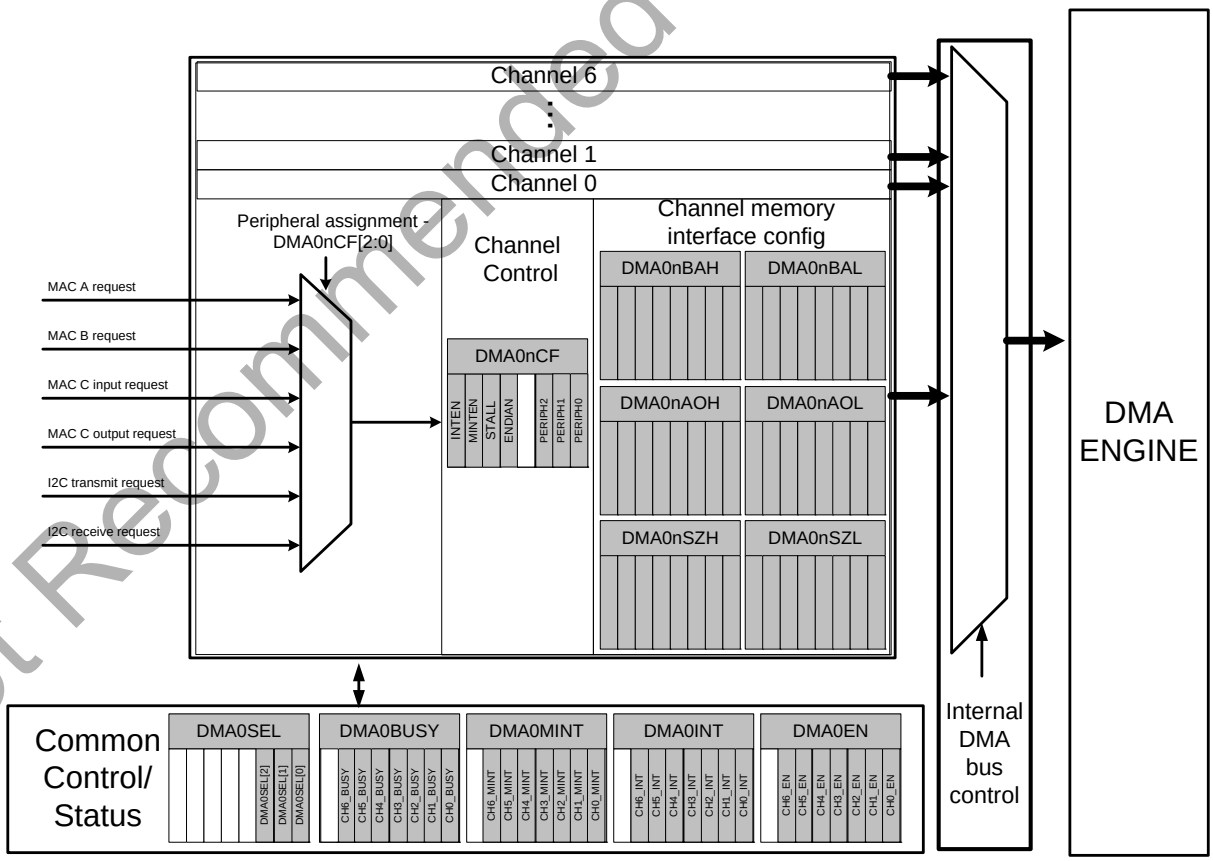


Figure 21.1. DMA0 Block Diagram

21.1. DMA0 Architecture

The first step in configuring a DMA0 channel is to select the desired channel for data transfer using DMA0SEL[2:0] bits (DMA0SEL). After setting the DMA0 channel, firmware can address channel-specific registers such as DMA0NCF, DMA0NBAH/L, DMA0NAOH/L, and DMA0NSZH/L. Once firmware selects a channel, the subsequent SFR configuration applies to the DMA0 transfer of that selected channel.

Each DMA0 channel consists of an SFR assigning the channel to a peripheral, a channel control register and a set of SFRs that describe XRAM and SFR addresses to be used during data transfer (Figure 21.1). The peripheral assignment bits of DMA0nCF select one of the eight data transfer functions. The selected channel can choose the desired function by writing to the PERIPH field in the DMA0NCF register.

The control register DMA0NCF of each channel configures the endianness of the data in XRAM, stall enable, full-length interrupt enable and mid-point interrupt enable. When a channel is stalled by setting the STALL bit (DMA0NCF.5), DMA0 transfers in progress will not be aborted, but new DMA0 transfers will be blocked until the stall status of the channel is reset. After the stall bit is set, software should poll the corresponding CHnBUSY bit to verify that there are no more DMA transfers for that channel.

The memory interface configuration SFRs of a channel define the linear region of XRAM involved in the transfer through a 12-bit base address register DMA0NBAH:L, a 10-bit address offset register DMA0NAOH/L and a 10-bit data transfer size DMA0NSZH:L. The effective memory address is the address involved in the current DMA0 transaction.

Effective Memory Address = Base Address + Address Offset

The address offset serves as byte counter. The address offset should be always less than data transfer length. The address offset increments by one after each byte transferred. For DMA0 configuration of any channel, address offsets of active channels should be reset to 0 before DMA0 transfers occur.

Data transfer size DMA0NSZH:L defines the maximum number of bytes for the DMA0 transfer of the selected channel. If the address offset reaches data transfer size, the full-length interrupt flag bit CHnI (DMA0INT) of the selected channel will be asserted. Similarly, the mid-point interrupt flag bit CHnMI is set when the address offset is equal to half of data transfer size if the transfer size is an even number or when the address offset is equal to half of the transfer size plus one if the transfer size is an odd number. Interrupt flags must be cleared by software so that the next DMA0 data transfer can proceed.

The DMA0 subsystem permits data transfer between SFR registers and XRAM. The DMA0 subsystem executes its task based on settings of a channel's control and memory interface configuration SFRs. When data is copied from XRAM to SFR registers, it takes two cycles for DMA0 to read from XRAM and the SFR write occurs in the second cycle. If more than one byte is involved, a pipeline is used. When data is copied from SFR registers to XRAM, the DMA0 only requires one cycle for one byte transaction.

The selected DMA0 channel for a peripheral should be enabled through the enable bits CHnEN (DMA0EN.n) to allow the DMA0 to transfer the data. When the DMA0 is transferring data on a channel, the busy status bit of the channel CHnBUSY (DMA0BUSY.n) is set. During the transaction, writes to DMA0NSZH:L, DMA0NBAH:L, and DMA0NAOH:L are disabled.

Besides reporting transaction status of a channel, DMA0BUSY can be used to force a DMA0 transfer on an already configured channel by setting the CHnBUSY bit (DMA0BUSY.n). This is useful for communication peripherals such as the I2C0. For example, after the I2C0 acknowledges the received slave address from I2C Master during a read transaction, a byte of data should be written to the I2C0DOUT register that will be transmitted to the I2C Master during the next I2C data transfer. Writing 1 to the CHn_BUSY bit (DMA0BUSY.n) of an enabled DMA0 channel forces a data transfer from the XRAM to I2C0DOUT.

21.2. DMA0 Arbitration

21.2.1. DMA0 Memory Access Arbitration

If both DMA0 and CPU attempt to access SFR register or XRAM at the same time, the CPU preempts the DMA0 module. DMA0 will be stalled until CPU completes its bus activity.

21.2.2. DMA0 channel arbitration

Multiple DMA0 channels can request transfer simultaneously, but only one DMA0 channel will be granted the bus to transfer the data. Channel 0 has the highest priority. DMA0 channels are serviced based on their priority. A higher priority channel is serviced first. Channel arbitration occurs at the end of the data transfer granularity (transaction boundary) of the DMA. When there is a DMA0 request at the transaction boundary from higher priority channel, lower priority ones will be stalled until the highest priority one completes its transaction. So, for 16-bit transfers like MAC0AH:L, the transaction boundary is at every 2 bytes.

21.3. DMA0 Operation in Low Power Modes

DMA0 remains functional in normal active, low power active, idle, low power idle modes but not in sleep or suspend mode. CPU will wait for DMA0 to complete all pending requests before it enters sleep mode. When the system wakes up from suspend or sleep mode to normal active mode, pending DMA0 interrupts will be serviced according to priority of channels. DMA0 stalls when CPU is in debug mode.

21.4. Transfer Configuration

The following steps are required to configure one of the DMA0 channels for operation:

1. Select the channel to be configured by writing DMA0SEL.
2. Specify the data transfer function by writing DMA0NCF. This register also specifies the endianness of the data in XRAM and enables full or mid-point interrupts.
3. Specify the base address in XRAM for the transfer by writing DMA0NBAH:L.
4. Specify the size of the transfer in bytes by writing DMA0NSZH:L.
5. Reset the address offset counter by writing 0 to DMA0NAOH:L.
6. Enable the DMA0 channel by writing 1 to the appropriate bit in DMA0EN.

21.5. DMA0 Registers

Register 21.1. DMA0EN: DMA0 Channel Enable

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x92								

Bit	Name	Function
7	Reserved	Must write reset value.
6	CH6EN	**Channel 6 Enable.** 0: Disable DMA0 channel 6. 1: Enable DMA0 channel 6.
5	CH5EN	**Channel 5 Enable.** 0: Disable DMA0 channel 5. 1: Enable DMA0 channel 5.
4	CH4EN	**Channel 4 Enable.** 0: Disable DMA0 channel 4. 1: Enable DMA0 channel 4.
3	CH3EN	**Channel 3 Enable.** 0: Disable DMA0 channel 3. 1: Enable DMA0 channel 3.
2	CH2EN	**Channel 2 Enable.** 0: Disable DMA0 channel 2. 1: Enable DMA0 channel 2.
1	CH1EN	**Channel 1 Enable.** 0: Disable DMA0 channel 1. 1: Enable DMA0 channel 1.
0	CH0EN	**Channel 0 Enable.** 0: Disable DMA0 channel 0. 1: Enable DMA0 channel 0.

Register 21.2. DMA0INT: DMA0 Full-Length Interrupt Flags

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CH6I	CH5I	CH4I	CH3I	CH2I	CH1I	CH0I
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xE8 (bit-addressable)								

Bit	Name	Function
7	Reserved	Must write reset value.
6	CH6I	**Channel 6 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 6.
5	CH5I	**Channel 5 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 5.
4	CH4I	**Channel 4 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 4.
3	CH3I	**Channel 3 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 3.
2	CH2I	**Channel 2 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 2.
1	CH1I	**Channel 1 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 1.
0	CH0I	**Channel 0 Full-Length Interrupt Flag.** 0: No Interrupt generated. 1: Full-length interrupt generated in channel 0.

Note: Channel full-length interrupt flags are set when the offset address DMA0NAOH/L equals to data transfer size DMA0NSZH/L minus 1 for the channel. Firmware must clear this flag. The full-length interrupt is enabled by setting the IEN bit in the DMA0NCF register with DMA0SEL configured for the corresponding channel.

Register 21.3. DMA0MINT: DMA0 Mid-Point Interrupt Flags

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CH6MI	CH5MI	CH4MI	CH3MI	CH2MI	CH1MI	CH0MI
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x88 (bit-addressable)								

Bit	Name	Function
7	Reserved	Must write reset value.
6	CH6MI	**Channel 6 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 6.
5	CH5MI	**Channel 5 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 5.
4	CH4MI	**Channel 4 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 4.
3	CH3MI	**Channel 3 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 3.
2	CH2MI	**Channel 2 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 2.
1	CH1MI	**Channel 1 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 1.
0	CH0MI	**Channel 0 Mid-Point Interrupt Flag.** 0: No mid-point Interrupt generated. 1: Mid-point Interrupt generated in channel 0.

Note: Mid-point Interrupt flag is set when the offset address DMA0NAOH/L equals to half of data transfer size DMA0NSZH/L if the transfer size is an even number or half of data transfer size DMA0NSZH/L plus one if the transfer size is an odd number. Firmware must clear this flag. The mid-point interrupt is enabled by setting the MIEN bit in the DMA0NCF register with DMA0SEL configured for the corresponding channel.

Register 21.4. DMA0BUSY: DMA0 Busy

Bit	7	6	5	4	3	2	1	0
Name	Reserved	CH6BUSY	CH5BUSY	CH4BUSY	CH3BUSY	CH2BUSY	CH1BUSY	CH0BUSY
Type	R	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x91								

Bit	Name	Function
7	Reserved	Must write reset value.
6	CH6BUSY	**Channel 6 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 6. Writing this bit to 1 forces a DMA0 transfer to start on channel 6.
5	CH5BUSY	**Channel 5 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 5. Writing this bit to 1 forces a DMA0 transfer to start on channel 5.
4	CH4BUSY	**Channel 4 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 4. Writing this bit to 1 forces a DMA0 transfer to start on channel 4.
3	CH3BUSY	**Channel 3 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 3. Writing this bit to 1 forces a DMA0 transfer to start on channel 3.
2	CH2BUSY	**Channel 2 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 2. Writing this bit to 1 forces a DMA0 transfer to start on channel2.
1	CH1BUSY	**Channel 1 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 1. Writing this bit to 1 forces a DMA0 transfer to start on channel 1.
0	CH0BUSY	**Channel 0 Busy.** This bit is set to 1 by hardware when a DMA0 transfer is in progress on channel 0. Writing this bit to 1 forces a DMA0 transfer to start on channel 0.

Register 21.5. DMA0SEL: DMA0 Channel Select

Bit	7	6	5	4	3	2	1	0
Name	Reserved					SELECT		
Type	R	RW				RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x94								

Bit	Name	Function
7:3	Reserved	Must write reset value.
2:0	SELECT	**Channel Select.** This field selects the channel and provides access to the channel's DMA0NCF, DMA0NBAL/H, DMA0NAOL/H, DMA0NSZL/H registers. 000: Select channel 0. 001: Select channel 1. 010: Select channel 2. 011: Select channel 3. 100: Select channel 4. 101: Select channel 5. 110: Select channel 6. 111: Reserved.

Register 21.6. DMA0NCF: DMA0 Channel Configuration

Bit	7	6	5	4	3	2	1	0
Name	IEN	MIEN	STALL	ENDIAN	Reserved	PERIPH		
Type	RW	RW	RW	RW	R	RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD8 (bit-addressable)								

Bit	Name	Function
7	IEN	**Full-Length Interrupt Enable.** This bit enables DMA0 full-length interrupt requests for the selected channel. 0: Disable the DMA0 full-length interrupt request of the selected channel. 1: Enable the DMA0 full-length interrupt request of the selected channel.
6	MIEN	**Mid-Point Interrupt Enable.** This bit enables DMA0 mid-point interrupt requests for the selected channel. 0: Disable the DMA0 mid-point interrupt request of the selected channel. 1: Enable the DMA0 mid-point interrupt request of the selected channel.
5	STALL	**Channel Stall.** Setting this bit stalls the DMA transfer on the selected channel. A stalled channel cannot initiate new DMA transfers. The DMA0 transfer of the stalled channel resumes where it was only when this bit is cleared by firmware. 0: The DMA transfer of the selected channel is not stalled. 1: The DMA transfer of the selected channel is stalled.
4	ENDIAN	**Data Transfer Endianness.** This bit sets the byte order of the XRAM data. 0: Data is written to and read from XRAM in little endian order. 1: Data is written to and read from XRAM in big endian order.
3	Reserved	Must write reset value.
2:0	PERIPH	**Peripheral Transfer Select.** This field selects the DMA0 transfer function for the selected channel. 000-001: Reserved. 010: The DMA channel transfers from XRAM to the MAC A register. 011: The DMA channel transfers from XRAM to the MAC B register. 100: The DMA channel transfers from XRAM to the MAC accumulator registers. 101: The DMA channel transfers from the MAC accumulator registers to XRAM. 110: The DMA channel transfers from the I2C Slave data register to XRAM. 111: The DMA channel transfers from XRAM to the I2C Slave data register.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.7. DMA0NBAH: Memory Base Address High

Bit	7	6	5	4	3	2	1	0
Name	Reserved				NBAH			
Type	R				RW			
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCA								

Bit	Name	Function
7:4	Reserved	Must write reset value.
3:0	NBAH	**Memory Base Address High.** This field sets high byte of the channel memory base address. This base address is the starting channel XRAM address if the channel's address offset DMA0NAO is reset to 0.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.8. DMA0NBAL: Memory Base Address Low

Bit	7	6	5	4	3	2	1	0
Name	NBAL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC9								

Bit	Name	Function
7:0	NBAL	**Memory Base Address Low.** This field sets low byte of the channel memory base address. This base address is the starting channel XRAM address if the channel's address offset DMA0NAO is reset to 0.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.9. DMA0NAOH: Memory Address Offset High

Bit	7	6	5	4	3	2	1	0
Name	Reserved						NAOH	
Type	R						RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCC								

Bit	Name	Function
7:2	Reserved	Must write reset value.
1:0	NAOH	**Memory Address Offset High.** This field is the high byte of the channel offset address. The base address added to the offset address creates the current channel XRAM address. The address offset auto-increments by one after one byte is transferred. When configuring a channel for a DMA transfer, the address offset should be reset to 0.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.10. DMA0NAOL: Memory Address Offset Low

Bit	7	6	5	4	3	2	1	0
Name	NAOL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCB								

Bit	Name	Function
7:0	NAOL	**Memory Address Offset Low.** This field is the high byte of the channel offset address. The base address added to the offset address creates the current channel XRAM address. The address offset auto-increments by one after one byte is transferred. When configuring a channel for a DMA transfer, the address offset should be reset to 0.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.11. DMA0NSZH: Memory Transfer Size High

Bit	7	6	5	4	3	2	1	0
Name	Reserved						NSZH	
Type	R						RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCE								

Bit	Name	Function
7:2	Reserved	Must write reset value.
1:0	NSZH	**Memory Transfer Size High.** This field sets the upper two bits of the number of DMA transfers for the selected channel.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

Register 21.12. DMA0NSZL: Memory Transfer Size Low

Bit	7	6	5	4	3	2	1	0
Name	NSZL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCD								

Bit	Name	Function
7:0	NSZL	**Memory Transfer Size Low.** This field sets the low byte of the number of DMA transfers for the selected channel.
Note: This register is a DMA channel indirect register. Select the desired channel first using the DMA0SEL register.		

22. Multiply and Accumulate (MAC0)

The C8051F97x devices include a multiply and accumulate engine which can be used to speed up many mathematical operations. MAC0 contains a 16-by-16 bit multiplier and a 40-bit adder, which can perform integer or fractional multiply-accumulate and multiply operations in a single SYSCLK cycle. Figure 22.1 shows a block diagram of the MAC0 subsystem and its associated SFRs.

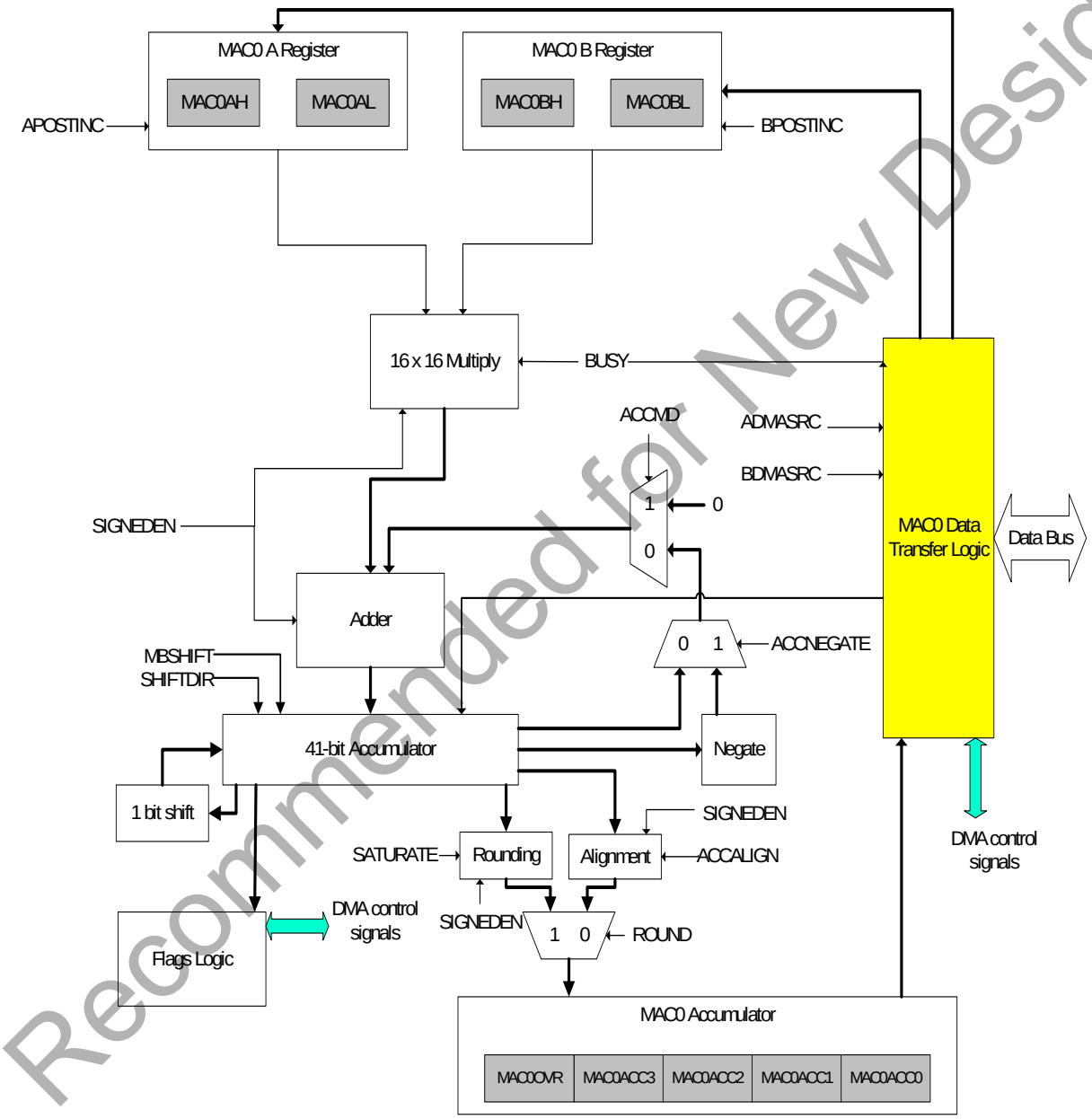


Figure 22.1. MAC0 Block Diagram

The module provides the following capabilities:

- Single cycle operation
- Multiply-accumulate or multiply only
- Support for integer or fractional operations
- Support for signed or unsigned operations
- Rounding with saturation
- Auto-increment or constant A and/or B registers
- Logical 1-bit or multiple bit shift of accumulator left or right
- Negation of accumulator, A, and/or B registers
- DMA support for repetitive operations on large arrays of data.
- Signed and unsigned alignment (right shift in bytes) of accumulator result

22.1. Special Function Registers

There are fifteen special function register (SFR) locations associated with MAC0. Six of these registers are related to configuration and operation, while the other nine are used to store multi-byte input and output data for MAC0. The configuration registers MAC0CF0, MAC0CF1, MAC0CF2 are used to configure and control MAC0. The status register MAC0STA contains flags to indicate overflow and interrupt conditions, MAC operation completion, as well as zero result. The 16-bit MAC0A (MAC0AH:MAC0AL) and MAC0B (MAC0BH:MAC0BL) registers are used as inputs to the multiplier. Figure 22.1 shows a 41-bit accumulator—but only 40 bits can be read by the MCU or DMA via the MAC0 Accumulator. The MAC0 Accumulator consists of five SFRs: MAC0OVF, MAC0ACC3, MAC0ACC2, MAC0ACC1, and MAC0ACC0. The primary result of a MAC0 operation is available from the Accumulator registers after the appropriate rounding, saturation or byte-realignment has been performed.

22.2. Integer and Fractional Math

MAC0 can perform math operations in unsigned or signed mode—this is controlled by the SIGNEDEN signed mode selection bit. MAC0 is capable of interpreting the 16-bit inputs stored in MAC0A and MAC0B as integers or as fractional numbers. When the FRACMD bit (MAC0CF0.4) is cleared to 0, the inputs are treated as 16-bit integer values. After a MAC operation, the internal 41-bit accumulator will contain the integer result. Figure 22.2 shows how integers are stored in the SFRs.

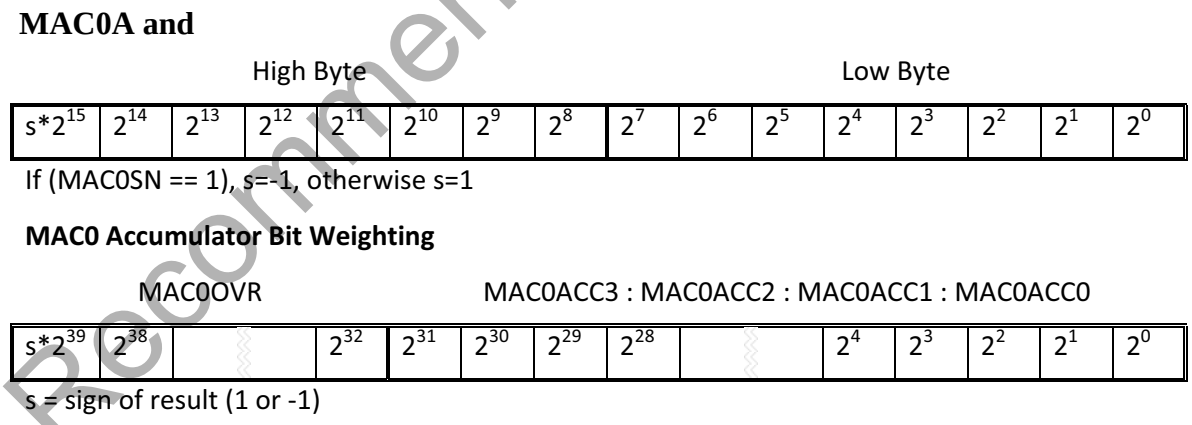


Figure 22.2. Integer Mode Data Representation

When the FRACMD bit is set to 1, the inputs are treated at 16-bit fractional values. The decimal point is located between bits 15 and 14 of the data word. After the operation, the accumulator will contain a 40-bit fractional value, with the decimal point located between bits 31 and 30. Figure 22.3 shows how fractional numbers are stored in the SFRs.

MAC0A and MAC0B Bit Weighting

	High Byte							Low Byte							
s*1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

If (MAC0SN == 1), s=-1, otherwise s=1

MAC0 Accumulator Bit Weighting (no rounding)

MAC0OVR				MAC0ACC3 : MAC0ACC2 : MAC0ACC1 : MAC0ACC0										
$s*2^{8}$	2^{7}		2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}		2^{-27}	2^{-28}	2^{-29}	2^{-30}	2^{-31}	

s = sign of result (1 or -1)

MAC0 Accumulator rounded result (MAC0RND=1) Bit Weighting

	MAC0ACC1							MAC0ACC0							
s*1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}

s = sign of result (1 or -1)

Figure 22.3. Fractional Mode Data Representation

22.3. Operating in Multiply and Accumulate Mode

MAC0 operates in multiply and accumulate (MAC) mode when the ACCMD bit (MAC0CF0.3) is cleared to 0. When operating in MAC mode, MAC0 performs a 16-by-16 bit multiply on the contents of the MAC0A and MAC0B registers and adds the result to the contents of the 41-bit accumulator. A MAC operation takes 1 SYSCLK cycle to complete. A rounded (and optionally, saturated) result is available when the MAC0RND bit is set.

If the CLRACC bit (MAC0CF0.5) is set to 1, the accumulator and all MAC0STA flags will be cleared during the next SYSCLK cycle. The CLRACC bit will clear itself to 0 when the clear operation has completed.

22.4. Operating in Multiply Only Mode

MAC0 operates in multiply only mode when the ACCMD bit (MAC0CF0.5) is set to 1. Multiply only mode is identical to Multiply and Accumulate mode, except that the multiplication result is added with a value of zero before being stored in the 41-bit accumulator (i.e. it overwrites the current accumulator contents). As in MAC mode, the rounded result can be read if the ROUND bit is set. Note that in Multiply Only mode, the HOVF flag is not affected.

22.5. MCU Mode Operation

MCU mode operation for the MAC is enabled when there is no DMA channel enabled for transferring data to or from any MAC0 register. When MCU mode operation is active, the MAC inputs and results are transferred to or from XRAM via software access to the SFRs. During MCU mode operation, a MAC operation is triggered by software writing 1 to the BUSY bit in the MAC0STA register. When the MAC operation has completed, hardware clears the BUSY bit to 0. The typical sequence of MCU mode operations is as follows:

1. Firmware disables all MAC-specific DMA channels.
2. Firmware initializes the control registers (MAC0CF0, MAC0CF1, MAC0CF2, MAC0ITER) appropriately.
3. Firmware writes all the operand values:
   - Update MAC0A, MAC0B
   - Setup accumulator either by clearing it (via setting the CLRACC bit) or writing directly to MAC0ACC0-3 and MAC0OVR
4. Firmware writes 1 to the BUSY bit.
5. MAC0 completes the MAC operation in 1 SYSCLK cycle, clears BUSY bit to 0, and sets both the MACINT and ACCRDY bits to 1.

22.6. DMA Mode Operation

DMA mode operation is a powerful mode that can be used process large array of data using the MAC0 module. Alternatively, it can be used to implement digital filters efficiently. DMA mode operation for the MAC0 is enabled when there is at least one DMA channel enabled for transferring data to or from any MAC0 register.

During DMA mode operation, the BUSY bit must be set to 1 to generate DMA requests. The complete flowchart of DMA mode operation is given in Figure 22.4.

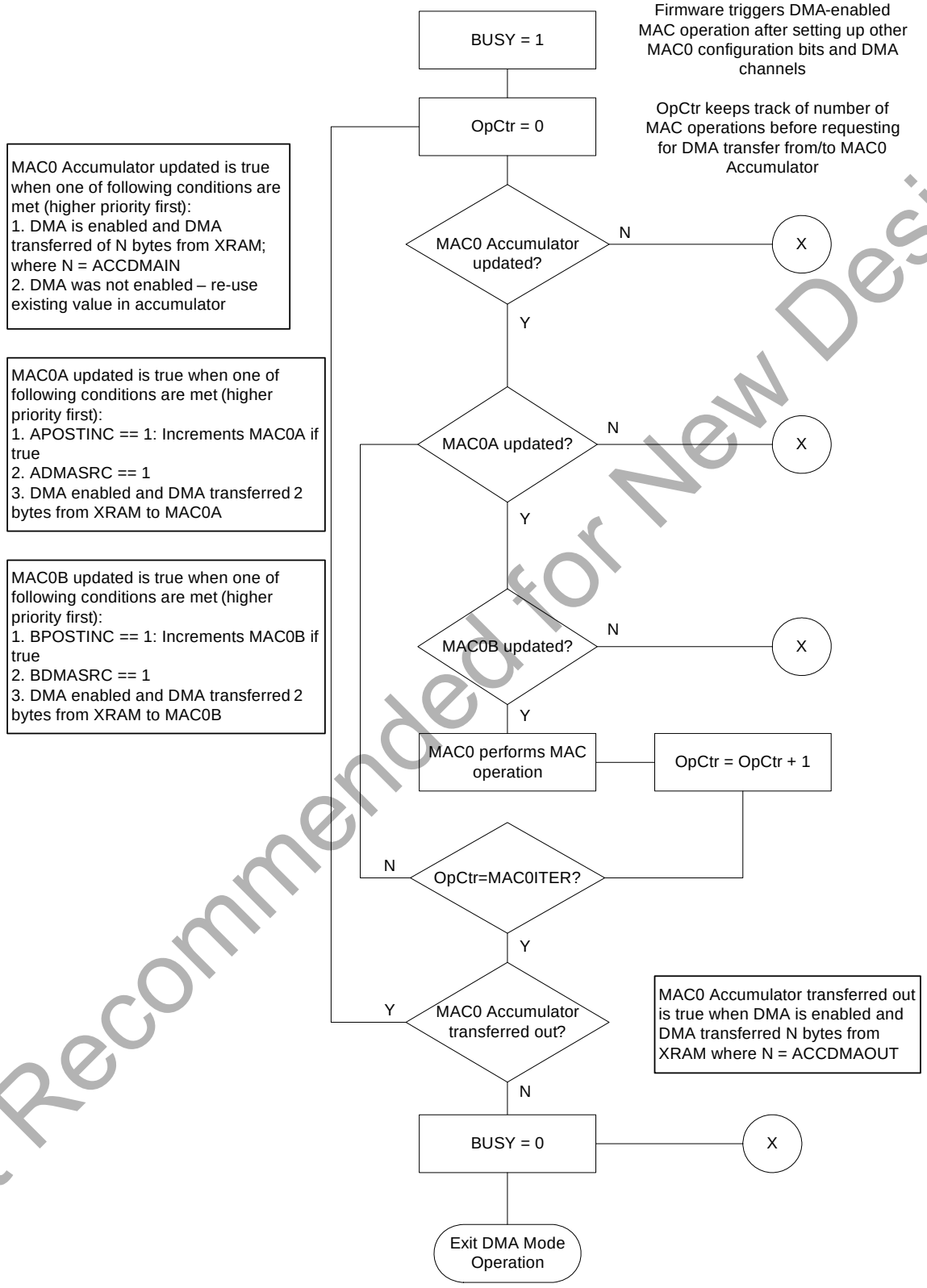


Figure 22.4. DMA Mode Operation Flow Chart

A few important points to note for DMA mode operation:

- Partial DMA transfers will terminate DMA mode operation. For example, a MAC0A DMA transaction requires 2 bytes, but if the DMA has only 1 byte left to transfer, the DMA mode operation will terminate after the single byte is transferred into MAC0A.
- Even if DMA transfers are enabled for MAC0A, no DMA request will be generated for MAC0A if either APOSTINC or ADMASRC bit is set to 1.
- If the ACCDMAIN field specifies a transfer length of less than 5 bytes, only the least significant bytes of the accumulator will receive data from the DMA. The upper bytes will be sign-extended in a way that is compatible with the SIGNEDEN setting as shown in Figure 22.5. The example below shows the result of the 41-bit accumulator after a transfer of a 16-bit number d[0:15] by DMA.
- To calculate the MAC output of two arrays, 3 DMA channels are needed: one each for MAC0A and MAC0B and one for an XRAM transfer to the ACC to set the ACC registers at the start of the calculation.

Unsigned DMA transfer of 2 bytes (b[0:15] into 41-bit accumulator (MAC0ACCIN_DMA = 1)

Bit position	40	39	38	37	17	16	15	14	13	2	1	0
	0	0	0	0	0	0	d15	d14	d13	d3	d1	d0

Signed DMA transfer of 2 bytes (b[0:15] into 41-bit accumulator (MAC0ACCIN_DMA = 1)

Bit position	40	39	38	37	17	16	15	14	13	2	1	0
	d15	d15	d15	d15	d15	d15	d15	d14	d13	d3	d1	d0

Figure 22.5. DMA Transfer into Accumulator in Sign and Unsigned Modes

22.7. Accumulator 1-Bit Shift Operations

MAC0 contains a 1-bit arithmetic shift function which can be used to shift the contents of the 41-bit accumulator left or right by one bit. The accumulator shift is initiated by writing a 1 to the SHIFTEN bit and takes one SYSCLK cycle to complete. In MCU mode, setting SHIFTEN bit will immediately shift one bit of the accumulator at the next system clock regardless of whether the MAC has finished the operation or not. In DMA mode, setting the SHIFTEN bit will shift the accumulator after the completion of the MAC operation. Hardware automatically clears the SHIFTEN bit to 0 after the shift has completed. The direction of the arithmetic shift is controlled by the SHIFTDIR bit. A 1-bit arithmetic shift does not affect any flag in the MAC0STA register.

The 1-bit shift examples for signed (SIGNEDEN = 1) and unsigned (SIGNEDEN = 0) modes are shown in Figure 22.6.

Original 41-bit accumulator data (before shifting)

Bit position:	40	39	38	37	7	6	5	4	3	2	1	0
	d40	d39	d38	d37	d7	d6	d5	d4	d3	d2	d1	d0

Signed mode 1-bit shift left result

Bit position	40	39	38	37	7	6	5	4	3	2	1	0
	d40	d38	d37	d36	d6	d5	d4	d3	d2	d1	d0	0

Signed mode 1-bit shift right result

Bit position	40	39	38	37	7	6	5	4	3	2	1	0
	d40	d40	d39	d38	d8	d7	d6	d5	d4	d3	d2	d1

Unsigned mode 1-bit shift left result

Bit position	40	39	38	37	7	6	5	4	3	2	1	0
	d39	d38	d37	d36	d6	d5	d4	d3	d2	d1	d0	0

Unsigned mode 1-bit shift right result

Bit position	40	39	38	37	7	6	5	4	3	2	1	0
	0	d40	d39	d38	d8	d7	d6	d5	d4	d3	d2	d1

Figure 22.6. 1-Bit Shift of 41-Bit Accumulator in Signed and Unsigned Modes

22.8. Multi-Bit Shift Accumulator Operation

The MAC0 also includes a multi-bit arithmetic shift (up to 4 bits) function that can operate in DMA mode only. The shift operation is only performed at the end of a MAC operation. The SHIFTDIR bit controls the direction of shift. To enable multi-bit shifting on the Accumulator, the SHIFTEN bit must be set.

- MBSHIFT = 0x00 : 1-bit shift
- MBSHIFT = 0x01 : 2-bit shift
- MBSHIFT = 0x02 : 3-bit shift
- MBSHIFT = 0x03 : 4-bit shift

22.9. Accumulator Alignment (Right Byte Shift)

Accumulator alignment (or right byte shifting) is one of two methods for DMA or MCU to access the 41-bit accumulator in different ways without changing the contents of the 41-bit accumulator. On power-up reset, this is the default option by which data is read from the MAC0 Accumulator. The other method of access is the rounding and saturation logic described on page 209 MAC0 provides 4 different byte alignment options:

- No align (no shifting)—this is the default on reset.
- SR1 (1 byte right shift)
- SR3 (3 byte right shift)
- SR4 (4 byte right shift)

The alignment is performed with the appropriate sign extension depending on the setting of SIGNEDEN. The examples in Table 22.1 and Table 22.2 show what the MCU or DMA would read from MAC0 Accumulator with the SR3 setting.

Table 22.1. Unsigned Mode (SIGNEDEN = 0)

Byte Position	41-Bit Accumulator Value	MAC0 Accumulator Value
ACC0	X0	X3
ACC1	X1	X4
ACC2	X2	0
ACC3	X3	0
ACCOVF	X4	0

Table 22.2. Signed Mode (SIGNEDEN = 1)

Byte Position	41-Bit Accumulator Value	MAC0 Accumulator Value (bit 40 of 41-bit acc = 0)	MAC0 Accumulator Value (bit 40 of 41-bit acc = 1)
ACC0	X0	X3	X3
ACC1	X1	X4	X4
ACC2	X2	0	0xff
ACC3	X3	0	0xff
ACCOVF	X4	0	0xff

Accumulator alignment does not affect any flag in the MAC0STA.

22.10. Rounding and Saturation

Rounding is another method for the MAC0 to access the 41-bit accumulator without changing its contents. A rounding engine is included, which can be used to provide a rounded result when operating on fractional numbers. MAC0 uses an unbiased rounding algorithm to round the data stored in bits 31–16 of the 41-bit accumulator, as shown in Table 22.3. The 41-bit accumulator is not affected by the rounding engine. Although rounding is primarily used for fractional data, it can be used in integer mode to obtain a 2-byte right-shifted rounded result. The upper 3 bytes are stuffed with the appropriate bits depending on the SIGNEDEN setting, just like in the Accumulator Alignment option. The rounding and saturation logic does not affect any flag in the MAC0STA register. Table 22.3 and Table 22.4 show the results for the signed and unsigned rounding.

Table 22.3. MAC0 Unsigned Rounding without Saturation (SIGNEDEN = 0, SATURATE= 0)

41-Bit ACC Bits 15–0 (ACC41[15:0])	41-Bit ACC Bits 31–16 (ACC41[31:16])	Lower 16 Bits of MAC0 Accumulator	Upper 24 Bits of MAC0 Accumulator
>0x8000	Any value	ACC41[31:16] + 1	All bits = 0
<0x8000	Any value	ACC41[31:16]	All bits = 0
== 0x8000	Odd (LSB == 1)	ACC41[31:16] + 1	All bits = 0
== 0x8000	Even (LSB == 0)	ACC41[31:16]	All bits = 0

Table 22.4. MAC0 Signed Rounding without Saturation (SIGNEDEN = 1, SATURATE = 0, Let X41 = [ACC41 + 0x8000])

41-Bit ACC Bits 15–0 (ACC41[15:0])	41-Bit ACC Bits 31–16 (ACC41[31:16])	Lower 15 Bits of MAC0 Accumulator	Upper 25 Bits of MAC0 Accumulator
>0x8000	Any value	X41[30:16]	All bits = X41[39]
<0x8000	Any value	X41[30:16]	All bits = X41[39]
== 0x8000	Odd (LSB == 1)	X41[30:16]	All bits = X41[39]
== 0x8000	Even (LSB == 0)	X41[30:16] - 1	All bits = X41[39]

When saturation bit SATURATE is set to 1, Table 22.5 and Table 22.6 show the saturation results.

Table 22.5. SIGNEDEN = 0, SATURATE = 1, Y40 = Unsaturated Rounded Result

(SOVF | HOVF) value	MAC0 Accumulator
0	Y40
1	0x000000FFFF

Table 22.6. SIGNEDEN = 1, SATURATE = 1, Z40 = Unsaturated Rounded Result

(SOVF | HOVF | (ACC41[31]^Z40[15])) Value	MAC0 Accumulator
0	Z40
1, ACC41[39] = 0	0x0000007FFF
1, ACC41[39] = 1	0xFFFFFF8000

22.11. Usage Examples

This section details some software examples for using MAC0.

22.11.1. Multiply and Accumulate in Fractional Mode

The example below implements the equation:

```
(0.5 x 0.25) + (0.5 x -0.25) = 0.125 – 0.125 = 0.0
```

```
SFRPAGE = MAC0_PAGE;    // Change Page register to MAC0 Page
MAC0CF0 = 0x30;         // Clear accumulator, select fractional mode
                        // Multiply and Accumulate mode
MAC0CF1 = 0x10;         // Select signed mode
MAC0CF2 = 0x00;         // Disable rounding, saturation and alignment logic
MAC0A = 0x4000;         // Load MAC0A with 0.5 decimal
MAC0B = 0x2000;         // Load MAC0B with 0.25 decimal
MAC0ITER = 1;           // Set to 1 iteration
MAC0STA = 1;            // Set BUSY to start first MAC operation
NOP();                  // NOP to allow MAC0 to complete operation
MAC0CF0 |= 0x02;        // Negate MAC0B
MAC0STA = 1;            // Set BUSY to start second MAC operation
NOP();                  // After this, MAC0 Accumulator should be zero and
                        // ZEROF flag should be set
```

22.11.2. Multiply Only in Integer Mode

The example below implements the equation:

```
4660 x -292 = -1360720
```

```
SFRPAGE = MAC0_PAGE;    // Change Page register to MAC0 Page
MAC0CF0 = 0x2a;         // Clear accumulator, Multiply Only mode
                        // Negate B
MAC0CF1 = 0x10;         // Select signed mode
MAC0CF2 = 0x00;         // Disable rounding, saturation and alignment logic
MAC0A = 4660;           // Load MAC0A with 4660 decimal
MAC0B = 292;            // Load MAC0B with 292 decimal (will be negated)
MAC0ITER = 1;           // Set to 1 iteration
MAC0STA = 1;            // Set BUSY to start MAC operation
NOP();                  // NOP to allow MAC0 to complete operation
                        // MAC0 Accumulator should contain 0xFFFFEB3CB0
```

22.11.3. Initializing Memory Block Using DMA0 and MAC0

This example demonstrates a sophisticated example of initializing a block of memory in XRAM with value 0x55.

```
// Memory block to initialize
volatile SEGMENT_VARIABLE(memblock[500], U8, SEG_XDATA);

SFRPAGE = DMA0_PAGE;    // Change Page register to DMA0 Page
DMA0SEL = 0;            // Select DMA channel 0
DMA0EN &= ~0x01;        // Disable DMA channel 0
DMA0INT &= ~0x01;       // Clear interrupt bit for channel 0
DMA0NCF = 0x05;         // Select MAC0 Accumulator to XRAM transfer
DMA0NCF |= (LSB << 4);  // Use LSB to spec endian bit for compiler independence
DMA0NBA = (U16)&memblock[0];  // XRAM base address
DMA0NAO = 0;                  // XRAM offset
DMA0NSZ = 500;                // Transfer 500 bytes
DMA0EN |= 0x01;         // Enable DMA channel 0

// No need to change page register as MAC0 and DMA0 registers are in same page.
MAC0CF0 = 0x28;         // Clear accumulator, Multiply Only mode
MAC0CF1 = 0x0A;         // Select unsigned mode, constant A and constant B
MAC0CF2 = 0x04;         // Disable rounding, saturation and alignment logic
                        // 2-byte DMA transaction from MAC0 Accumulator to XRAM
MAC0A = 1;              // Load MAC0A with 1 decimal
MAC0B = 0x5555;         // Load MAC0B with 0x5555 hexadecimal
MAC0ITER = 1;           // Set to 1 iteration
MAC0STA = 1;            // Set BUSY to start MAC operation

while (!(DMA0INT & 1));// Poll for DMA Channel 0 completion interrupt

// All 500 bytes in memblock[] will be initialized to 0x55
```

22.12. MAC0 Registers

Register 22.1. MAC0STA: MAC0 Status

Bit	7	6	5	4	3	2	1	0
Name	Reserved	MACINT	SCHANGE	ACCRDY	HOVF	ZEROF	SOVF	BUSY
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xCF								

Bit	Name	Function
7	Reserved	Must write reset value.
6	MACINT	**Interrupt Flag.** This flag is set when at least one of the following conditions are met: - Hardware overflow bit (HOVF) is set. - Soft overflow bit (SOVF) is set AND the SOVFIEN bit is set. - The ACCRDY bit is set when the MAC operates in MCU mode. Firmware must clear this bit.
5	SCHANGE	**Sign Change Event.** Hardware sets this bit when operating in signed mode (SIGNEDEN = 1) and the sign bit of the MAC0 accumulator result is opposite of the expected sign (SIGNEXP) bit. Firmware must clear this bit. This bit also clears automatically if firmware clears the accumulator by setting the CLRACC bit in the MAC0CF0 register.
4	ACCRDY	**ACC Ready Status Flag.** This bit is set only when the MAC operates in MCU mode and a MAC operation completes. This bit must be cleared by firmware.
3	HOVF	**Hard Overflow Flag.** This bit is set to 1 when the a MAC operation results in a hardware overflow. Hardware overflow for a signed operation (SIGNEDEN = 1) occurs when a MAC operation causes MAC0OVF to change from 0x7F to 0x80 or 0x80 to 0x7F. Hardware overflow for an unsigned operation (SIGNEDEN = 0) occurs when a MAC operation causes MAC0OVF to change from 0xFF to 0x00. Firmware must clear this bit. This bit also clears automatically if firmware clears the accumulator by setting the CLRACC bit in the MAC0CF0 register.
2	ZEROF	**Zero Flag.** This bit is set to 1 if a MAC operation results in a MAC accumulator value of zero. Firmware must clear this bit. This bit also clears automatically if firmware clears the accumulator by setting the CLRACC bit in the MAC0CF0 register.

Bit	Name	Function
1	SOVF	**Soft Overflow Flag.** Hardware sets this bit to 1 when a signed MAC operation causes an overflow into the sign bit (bit 31) of the accumulator or when an unsigned MAC operation causes an overflow into the MAC0OVF register. Firmware must clear this bit. This bit also clears automatically if firmware clears the accumulator by setting the CLRACC bit in the MAC0CF0 register.
0	BUSY	**Busy Flag.** Firmware can set this bit to trigger the MAC0 peripheral to start an operation in both MCU mode and DMA mode. In MCU mode, the MAC0 module will perform a single MAC operation using data that is already present in the registers. Hardware clears this bit when the operation completes. In DMA mode, the MAC0 module will perform the required number of MAC operations as defined by the DMA registers, fetching the data into and out of the MAC0 registers via DMA transfers. Hardware automatically clears this bit when when all of the DMA operations complete.

Register 22.2. MAC0CF0: MAC0 Configuration 0

Bit	7	6	5	4	3	2	1	0
Name	SHIFTEN	SHIFTDIR	CLRACC	FRACMD	ACCMD	ACCNE-GATE	BNEGATE	ANEGATE
Type	W	RW	W	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC0 (bit-addressable)								

Bit	Name	Function
7	SHIFTEN	**Accumulator Shift Control.** When this bit is set, hardware shifts the 40-bit MAC0 accumulator by 1 bit. This bit also enables multi-bit shifts in DMA mode. 0: Do not shift the accumulator by one bit in the SHIFTDIR direction. 1: Shift the accumulator by one bit in the SHIFTDIR direction.
6	SHIFTDIR	**Accumulator Shift Direction.** This bit controls the direction of the accumulator shift activated by the SHIFTEN bit. 0: The MAC0 accumulator will be shifted left. 1: The MAC0 accumulator will be shifted right.
5	CLRACC	**Clear Accumulator.** This bit works only in MCU mode only and is used to reset the accumulator before the next operation. When this bit is set, the MAC0 accumulator will be cleared to zero. In addition, all bits in the MAC0STA register except MAC0INT and ACCRDY are cleared. Hardware will automatically clear this bit when the operation completes. Firmware will always read this bit as 0.
4	FRACMD	**Fractional Mode.** 0: MAC0 operates in Integer Mode. 1: MAC0 operates in Fractional Mode.
3	ACCMD	**Accumulate Mode.** 0: Select multiply-and-accumulate (MAC) mode. 1: Select multiply-only mode.
2	ACCNE-GATE	**Negate Accumulator Input.** This bit controls whether hardware negates the accumulator before it is added with the multiplication result of A and B. If this bit is set to 1, the SIGNEDEN bit must also be set to 1, enabling signed arithmetic. 0: No change is applied to the accumulator before it is added to the multiplication result of A and B. 1: Hardware negates the accumulator before it is added to the multiplication result of A and B.

Bit	Name	Function
1	BNEGATE	**Negate MAC0 B input.** This bit controls whether hardware negates the MAC0 B input before the multiplication operation with the MAC0 A input. The contents of the MAC0 B register do not change if the input is negated. 0: No change is applied to the MAC0 B input before the multiply operation. 1: Hardware negates the MAC0 B input before the multiply operation.
0	ANEGATE	**Negate MAC0 A input.** This bit controls whether hardware negates the MAC0 A input before the multiplication operation with the MAC0 B input. The contents of the MAC0 A register do not change if the input is negated. 0: No change is applied to the MAC0 A input before the multiply operation. 1: Hardware negates the MAC0 A input before the multiply operation.

Register 22.3. MAC0CF1: MAC0 Configuration 1

Bit	7	6	5	4	3	2	1	0
Name	MBSHIFT		SIGNEXP	SIGNEDEN	MAC0BC	BPOSTINC	ADMASRC	APOSTINC
Type	RW		RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC4								

Bit	Name	Function
7:6	MBSHIFT	**Multi-Bit Accumulator Shift.** This field sets the number of bits hardware shifts the accumulator at the end of a MAC operation in DMA mode if SHIFTEN is set to 1. The direction of the shift is determined by the SHIFTDIR bit. 00: Shift the accumulator 1 bit. 01: Shift the accumulator by 2 bits. 10: Shift the accumulator by 3 bits. 11: Shift the accumulator by 4 bits.
5	SIGNEXP	**Expected Accumulator Sign.** This bit sets the expected sign of the accumulator after an operation. A signed operation that does not match the SIGNEXP bit will set the SCHANGE bit. This bit can only be used with signed arithmetic (SIGNEDEN = 1). 0: Set the expected sign to zero. 1: Set the expected sign to one.
4	SIGNEDEN	**Signed Mode Enable.** 0: MAC operations use unsigned arithmetic. 1: MAC operations use signed arithmetic.
3	MAC0BC	**B DMA Data Source Selection.** This bit controls the source of data for the MAC0 B register when operating in DMA mode. This bit is ignored when the MAC0 module operates in MCU mode. 0: Each MAC0 operation will request the DMA fetch data from XRAM for the MAC0 B register. 1: Each MAC0 operation will use existing data in MAC0 B register.
2	BPOSTINC	**B Post-Increment Enable.** This bit controls whether the MAC0 B register value is incremented after a MAC0 operation. This bit can be used in both MCU and DMA modes. 0: Do not change the MAC0 B register after a MAC0 operation. 1: Increment the MAC0 B register after a MAC0 operation.

Bit	Name	Function
1	ADMASRC	**A DMA Data Source Selection.** This bit controls the source of data for the MAC0 A register when operating in DMA mode. This bit is ignored when the MAC0 module operates in MCU mode. 0: Each MAC0 operation will request the DMA fetch data from XRAM for the MAC0 A register. 1: Each MAC0 operation will use existing data in MAC0 A register.
0	APOSTINC	**A Post-Increment Enable.** This bit controls whether the MAC0 A register value is incremented after a MAC0 operation. This bit can be used in both MCU and DMA modes. 0: Do not change the MAC0 A register after a MAC0 operation. 1: Increment the MAC0 A register after a MAC0 operation.

Register 22.4. MAC0CF2: MAC0 Configuration 2

Bit	7	6	5	4	3	2	1	0
Name	ROUND	SATU-RATE	ACCALIGN		ACCDMAOUT		ACCDMAIN	
Type	RW	RW	RW		RW		RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC5								

Bit	Name	Function
7	ROUND	**Rounding Enable.** This bit controls whether the MCU or DMA reads a rounded result from the MAC0 accumulator. 0: MAC0 accumulator does not contain a 16-bit rounded result. 1: MAC0 accumulator contains a 16-bit rounded result.
6	SATURATE	**Saturation Enable.** If ROUND is set to 1, this bit controls whether the rounded MAC0 accumulator result will saturate. The details of the saturation logic is defined in the section on Rounding and Saturation. 0: Rounded result will not saturate. 1: Rounded result will saturate.
5:4	ACCALIGN	**Accumulator Alignment.** This field controls how many bytes the hardware shifts the MAC0 accumulator result to the right after an operation. This setting can be used in both MCU and DMA modes. 00: Do not shift the accumulator output. 01: Shift the accumulator output right by 1 byte. 10: Shift the accumulator output right by 3 bytes. 11: Shift the accumulator output right by 4 bytes.
3:2	ACCD-MAOUT	**Accumulator DMA Input Count.** This field only effects DMA mode operations. This field specifies the number of bytes the MAC0 will request the DMA to transfer from the MAC0 accumulator to XRAM after each operation, starting from the least significant byte. 00: Request the DMA move 1 byte from the accumulator to XRAM. 01: Request the DMA move 2 bytes from the accumulator to XRAM. 10: Request the DMA move 4 bytes from the accumulator to XRAM. 11: Request the DMA move 5 bytes from the accumulator to XRAM.

Bit	Name	Function
1:0	ACCDMAIN	**Accumulator DMA Input Count.** This field only effects DMA mode operations. This field specifies the number of bytes the MAC0 will request the DMA to transfer from XRAM to the MAC0 accumulator before each operation, starting from the least significant byte. 00: Request the DMA move 1 byte from XRAM to the accumulator. 01: Request the DMA move 2 bytes from XRAM to the accumulator. 10: Request the DMA move 4 bytes from XRAM to the accumulator. 11: Request the DMA move 5 bytes from XRAM to the accumulator.

Register 22.5. MAC0INTE: MAC0 Interrupt Enable

Bit	7	6	5	4	3	2	1	0
Name	Reserved					SCHANGEIEN	ZEROFIEN	SOVFIEN
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC1								

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	SCHANGEIEN	**Sign Change Event Interrupt Enable.** Enables MAC0 interrupts if a sign change event (SCHANGE) occurs.
1	ZEROFIEN	**Zero Flag Interrupt Enable.** Enables MAC0 interrupts if the zero flag (ZEROF) is set.
0	SOVFIEN	**Soft Overflow Interrupt Enable.** Enables MAC0 interrupts if the soft overflow (SOVF) flag is set.

Register 22.6. MAC0AH: Operand A High Byte

Bit	7	6	5	4	3	2	1	0
Name	MAC0AH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xAB								

Bit	Name	Function
7:0	MAC0AH	**MAC0 A High Byte.** This field is the upper 8 bits of the MAC0 A input.

Register 22.7. MAC0AL: Operand A Low Byte

Bit	7	6	5	4	3	2	1	0
Name	MAC0AL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xAA								

Bit	Name	Function
7:0	MAC0AL	**MAC0 A Low Byte.** This field is the lower 8 bits of the MAC0 A input.

Register 22.8. MAC0BH: Operand B High Byte

Bit	7	6	5	4	3	2	1	0
Name	MAC0BH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xAF								

Bit	Name	Function
7:0	MAC0BH	**MAC0 B High Byte.** This field is the upper 8 bits of the MAC0 B input.

Register 22.9. MAC0BL: Operand B Low Byte

Bit	7	6	5	4	3	2	1	0
Name	MAC0BL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xAE								

Bit	Name	Function
7:0	MAC0BL	**MAC0 B Low Byte.** This field is the lower 8 bits of the MAC0 B input.

Register 22.10. MAC0OVF: Accumulator Overflow Byte

Bit	7	6	5	4	3	2	1	0
Name	MAC0OVF							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD6								

Bit	Name	Function
7:0	MAC0OVF	**MAC0 Accumulator Overflow Byte.** This field is the accumulator overflow byte, or bits [39:32] of the MAC0 operation accumulated result.

Register 22.11. MAC0ACC3: Accumulator Byte 3

Bit	7	6	5	4	3	2	1	0
Name	MAC0ACC3							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD5								

Bit	Name	Function
7:0	MAC0ACC3	**MAC0 Accumulator Byte 3.** This field is byte 3 of the accumulator, or bits [31:24] of the MAC0 operation accumulated result.

Register 22.12. MAC0ACC2: Accumulator Byte 2

Bit	7	6	5	4	3	2	1	0
Name	MAC0ACC2							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD4								

Bit	Name	Function
7:0	MAC0ACC2	**MAC0 Accumulator Byte 2.** This field is byte 2 of the accumulator, or bits [23:16] of the MAC0 operation accumulated result.

Register 22.13. MAC0ACC1: Accumulator Byte 1

Bit	7	6	5	4	3	2	1	0
Name	MAC0ACC1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD3								

Bit	Name	Function
7:0	MAC0ACC1	**MAC0 Accumulator Byte 1.** This field is byte 1 of the accumulator, or bits [15:8] of the MAC0 operation accumulated result.

Register 22.14. MAC0ACC0: Accumulator Byte 0

Bit	7	6	5	4	3	2	1	0
Name	MAC0ACC0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD2								

Bit	Name	Function
7:0	MAC0ACC0	**MAC0 Accumulator Byte 0.** This field is byte 0 of the accumulator, or bits [7:0] of the MAC0 operation accumulated result.

Register 22.15. MAC0ITER: Iteration Counter

Bit	7	6	5	4	3	2	1	0
Name	MAC0ITER							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD7								

Bit	Name	Function
7:0	MAC0ITER	**Iteration Counter.** This field specifies the number of MAC0 operations before requesting a DMA transfer from XRAM into the accumulator and from the accumulator into XRAM. A value of 0 in this field requests 256 iterations.

23. Cyclic Redundancy Check Unit (CRC0)

C8051F97x devices include a cyclic redundancy check unit (CRC0) that can perform a CRC using a 16-bit polynomial. CRC0 accepts a stream of 8-bit data written to the CRC0IN register. CRC0 posts the 16-bit result to an internal register. The internal result register may be accessed indirectly using the CRCPNT bits and CRC0DAT register, as shown in Figure 23.1. CRC0 also has a bit reverse register for quick data manipulation.

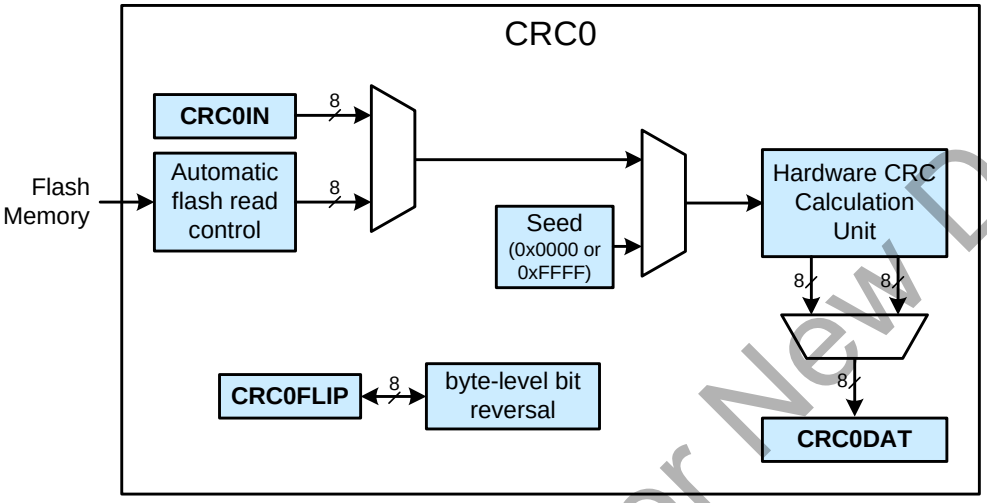


Figure 23.1. CRC0 Block Diagram

23.1. CRC Algorithm

The CRC unit generates a CRC result equivalent to the following algorithm:

1. XOR the input with the most-significant bits of the current CRC result. If this is the first iteration of the CRC unit, the current CRC result will be the set initial value (0x0000 or 0xFFFF).

2a. If the MSB of the CRC result is set, shift the CRC result and XOR the result with the selected polynomial.

2b. If the MSB of the CRC result is not set, shift the CRC result.

Repeat Steps 2a/2b for the number of input bits (8). The algorithm is also described in the following example.

The 16-bit CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
   unsigned char i;                    // loop counter

   #define POLY 0x1021

   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x8000) == 0x8000)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc << 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc << 1;
      }
   }

   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

Table 23.1 lists several input values and the associated outputs using the 16-bit CRC algorithm:

Table 23.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

23.2. Preparing for a CRC Calculation

To prepare CRC0 for a CRC calculation, software should set the initial value of the result. The polynomial used for the CRC computation is 0x1021. The CRC0 result may be initialized to one of two values: 0x0000 or 0xFFFF. The following steps can be used to initialize CRC0.

1. Select the initial result value (Set CRCVAL to 0 for 0x0000 or 1 for 0xFFFF).
2. Set the result to its initial value (Write 1 to CRCINIT).

23.3. Performing a CRC Calculation

Once CRC0 is initialized, the input data stream is sequentially written to CRC0IN, one byte at a time. The CRC0 result is automatically updated after each byte is written. The CRC engine may also be configured to automatically perform a CRC on one or more 256 byte blocks read from flash. The following steps can be used to automatically perform a CRC on flash memory.

1. Prepare CRC0 for a CRC calculation as shown above.
2. Write the index of the starting page to CRC0AUTO.
3. Set the AUTOEN bit to 1 in CRC0AUTO.
4. Write the number of 256 byte blocks to perform in the CRC calculation to CRCCNT.
5. Write any value to CRC0CN (or OR its contents with 0x00) to initiate the CRC calculation. The CPU will not execute code any additional code until the CRC operation completes. See the note in the CRC0CN register definition for more information on how to properly initiate a CRC calculation.
6. Clear the AUTOEN bit in CRC0AUTO.
7. Read the CRC result.

23.4. Accessing the CRC0 Result

The internal CRC0 result is 16 bits. The CRCPNT bits select the byte that is targeted by read and write operations on CRC0DAT and increment after each read or write. The calculation result will remain in the internal CR0 result register until it is set, overwritten, or additional data is written to CRC0IN.

23.5. CRC0 Bit Reverse Feature

CRC0 includes hardware to reverse the bit order of each bit in a byte as shown in Figure 23.2. Each byte of data written to CRC0FLIP is read back bit reversed. For example, if 0xC0 is written to CRC0FLIP, the data read back is 0x03. Bit reversal is a useful mathematical function used in algorithms such as the FFT.

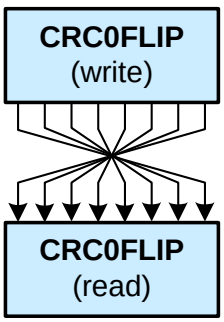


Figure 23.2. Bit Reversal

23.6. CRC Control Registers

Register 23.1. CRC0CN: CRC0 Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved				CRCINIT	CRCVAL	Reserved	CRCPNT
Type	R				RW	RW	R	RW
Reset	0	0	0	1	0	0	0	0
SFR Page = 0x0; SFR Address: 0x84								

Table 23.2. CRC0CN Register Bit Descriptions

Bit	Name	Function
7:4	Reserved	Must write reset value.
3	CRCINIT	**CRC Initialization Enable.** Writing a 1 to this bit initializes the entire CRC result based on CRCVAL.
2	CRCVAL	**CRC Initialization Value.** This bit selects the set value of the CRC result. 0: CRC result is set to 0x0000 on write of 1 to CRCINIT. 1: CRC result is set to 0xFFFF on write of 1 to CRCINIT.
1	Reserved	Must write reset value.
0	CRCPNT	**CRC Result Pointer.** Specifies the byte of the CRC result to be read/written on the next access to CRC0DAT. This bit will automatically toggle upon each read or write. 0: CRC0DAT accesses bits 7-0 of the 16-bit CRC result. 1: CRC0DAT accesses bits 15-8 of the 16-bit CRC result.

Note: Upon initiation of an automatic CRC calculation, the three cycles following a write to CRC0CN that initiate a CRC operation must only contain instructions which execute in the same number of cycles as the number of bytes in the instruction. An example of such an instruction is a 3-byte MOV that targets the CRC0FLIP register. When programming in C, the dummy value written to CRC0FLIP should be a non-zero value to prevent the compiler from generating a 2-byte MOV instruction.

Register 23.2. CRC0IN: CRC0 Data Input

Bit	7	6	5	4	3	2	1	0
Name	CRC0IN							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x85								

Table 23.3. CRC0IN Register Bit Descriptions

Bit	Name	Function
7:0	CRC0IN	**CRC Data Input.** Each write to CRC0IN results in the written data being computed into the existing CRC result according to the CRC algorithm.

Register 23.3. CRC0DAT: CRC0 Data Output

Bit	7	6	5	4	3	2	1	0
Name	CRC0DAT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x86								

Table 23.4. CRC0DAT Register Bit Descriptions

Bit	Name	Function
7:0	CRC0DAT	**CRC Data Output.** Each read or write performed on CRC0DAT targets the CRC result bits pointed to by the CRC0 Result Pointer (CRCPNT bits in CRC0CN).
Note: CRC0DAT may not be valid for one cycle after setting the CRCINIT bit in the CRC0CN register to 1. Any time CRCINIT is written to 1 by firmware, at least one instruction should be performed before reading CRC0DAT.		

Register 23.4. CRC0AUTO: CRC0 Automatic Control

Bit	7	6	5	4	3	2	1	0
Name	AUTOEN	CRCST						
Type	RW	RW						
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x9E								

Table 23.5. CRC0AUTO Register Bit Descriptions

Bit	Name	Function
7	AUTOEN	**Automatic CRC Calculation Enable.** When AUTOEN is set to 1, any write to CRC0CN will initiate an automatic CRC starting at flash sector CRCST and continuing for CRCCNT sectors.
6:0	CRCST	**Automatic CRC Calculation Starting Block.** These bits specify the flash block to start the automatic CRC calculation. The starting address of the first flash block included in the automatic CRC calculation is CRCST x block_size, where block_size is 256 bytes.

Register 23.5. CRC0CNT: CRC0 Automatic Flash Sector Count

Bit	7	6	5	4	3	2	1	0
Name	CRCDN	CRCCNT						
Type	RW	RW						
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x9D								

Table 23.6. CRC0CNT Register Bit Descriptions

Bit	Name	Function
7	CRCDN	**Automatic CRC Calculation Complete.** Set to 0 when a CRC calculation is in progress. Code execution is stopped during a CRC calculation; therefore, reads from firmware will always return 1.
6:0	CRCCNT	**Automatic CRC Calculation Block Count.** These bits specify the number of flash blocks to include in an automatic CRC calculation. The last address of the last flash block included in the automatic CRC calculation is (CRCST+CRCCNT) x Block Size - 1. The block size is 256 bytes.

Register 23.6. CRC0FLIP: CRC0 Bit Flip

Bit	7	6	5	4	3	2	1	0
Name	CRC0FLIP							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x9F								

Table 23.7. CRC0FLIP Register Bit Descriptions

Bit	Name	Function
7:0	CRC0FLIP	**CRC0 Bit Flip.** Any byte written to CRC0FLIP is read back in a bit-reversed order, i.e., the written LSB becomes the MSB. For example: If 0xC0 is written to CRC0FLIP, the data read back will be 0x03. If 0x05 is written to CRC0FLIP, the data read back will be 0xA0.

24. Clocking Sources

The C8051F97x devices can be clocked from the internal low-power 24.5 MHz oscillator, the internal low-power 20 MHz oscillator, the SmaRTClock real time clock oscillator, or externally by an external oscillator (not supported on devices in the QFN-24 package). An adjustable clock divider allows the selected clock source to be post-scaled by powers of 2, up to a factor of 128. By default, the system clock comes up as the 24.5 MHz oscillator divided by 8.

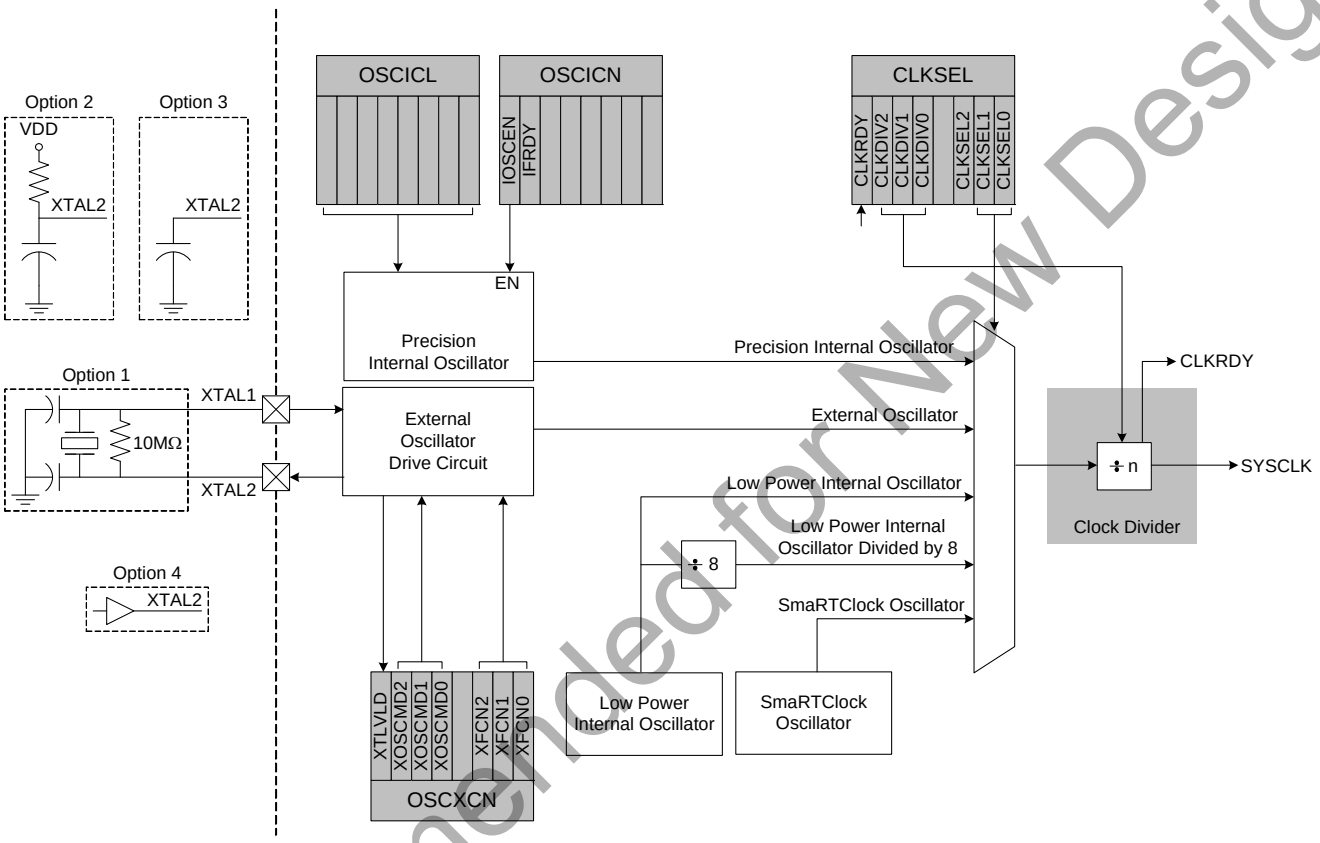


Figure 24.1. Clocking Sources Block Diagram

The proper way of changing the system clock when both the clock source and the clock divide value are being changed is as follows:

If switching from a fast “undivided” clock to a slower “undivided” clock:

1. Change the clock divide value.
2. Poll for CLKRDY > 1.
3. Change the clock source.

If switching from a slow “undivided” clock to a faster “undivided” clock:

1. Change the clock source.
2. Change the clock divide value.
3. Poll for CLKRDY > 1.

24.1. Programmable Precision Internal Oscillator

All C8051F97x devices include a programmable precision internal oscillator that may be selected as the system clock. OSCICL is factory calibrated to obtain a 24.5 MHz frequency. See Section “1. Electrical Characteristics” on page 10 for complete oscillator specifications.

The precision oscillator supports a spread spectrum mode which modulates the output frequency in order to reduce the EMI generated by the system. When enabled (SSE = 1), the oscillator output frequency is modulated by a stepped triangle wave whose frequency is equal to the oscillator frequency divided by 384 (63.8 kHz using the factory calibration). The deviation from the nominal oscillator frequency is +0%, –1.6%, and the step size is typically 0.26% of the nominal frequency. When using this mode, the typical average oscillator frequency is lowered from 24.5 MHz to 24.3 MHz.

24.2. Low Power Internal Oscillator

All C8051F97x devices include a low power internal oscillator that defaults as the system clock after a system reset. The low power internal oscillator frequency is 20 MHz ± 10% and is automatically enabled when selected as the system clock and disabled when not in use. See Section “1. Electrical Characteristics” on page 10 for complete oscillator specifications.

24.3. External Oscillator Drive Circuit

The C8051F970/1/3/4 devices include an external oscillator circuit that may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. Figure 24.1 shows a block diagram of the four external oscillator options. The external oscillator is enabled and configured using the OSCXCN register.

The external oscillator output may be selected as the system clock or used to clock some of the digital peripherals (e.g., Timers, PCA, etc.). See the data sheet chapters for each digital peripheral for details. See Section “1. Electrical Characteristics” on page 10 for complete oscillator specifications.

24.3.1. External Crystal Mode

If a crystal or ceramic resonator is used as the external oscillator, the crystal/resonator and a 10 MΩ resistor must be wired across the XTAL1 and XTAL2 pins as shown in Figure 24.1, Option 1. Appropriate loading capacitors should be added to XTAL1 and XTAL2, and both pins should be configured for analog I/O with the digital output drivers disabled.

Figure 24.2 shows the external oscillator circuit for a 20 MHz quartz crystal with a manufacturer recommended load capacitance of 12.5 pF. Loading capacitors are “in-series” as seen by the crystal and “in-parallel” with the stray capacitance of the XTAL1 and XTAL2 pins. The total value of the each loading capacitor and the stray capacitance of each XTAL pin should equal 12.5 pF x 2 = 25 pF. With a stray capacitance of 10 pF per pin, the 15 pF capacitors yield an equivalent series capacitance of 12.5 pF across the crystal.

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

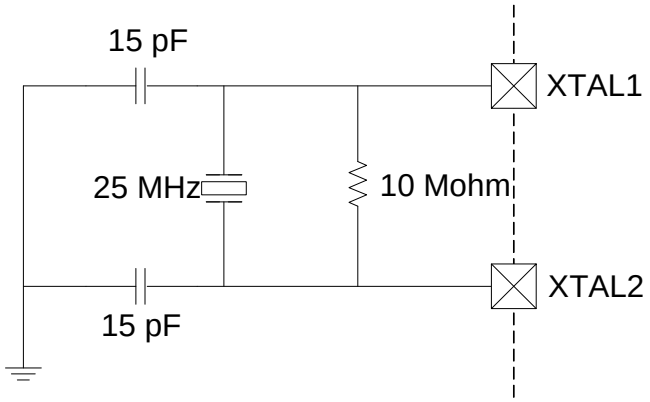


Figure 24.2. 25 MHz External Crystal Example

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

When using an external crystal, the external oscillator drive circuit must be configured by software for *Crystal Oscillator Mode* or *Crystal Oscillator Mode with divide by 2 stage*. The divide by 2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. The External Oscillator Frequency Control value (XFCN) must also be specified based on the crystal frequency. The selection should be based on Table 24.1. For example, a 25 MHz crystal requires an XFCN setting of 111b.

Table 24.1. Recommended XFCN Settings for Crystal Mode

XFCN	Crystal Frequency	Bias Current	Typical Supply Current (VDD = 2.4 V)
000	f ≤ 20 kHz	0.5 µA	3.0 µA, f = 32.768 kHz
001	20 kHz < f ≤ 58 kHz	1.5 µA	4.8 µA, f = 32.768 kHz
010	58 kHz < f ≤ 155 kHz	4.8 µA	9.6 µA, f = 32.768 kHz
011	155 kHz < f ≤ 415 kHz	14 µA	28 µA, f = 400 kHz
100	415 kHz < f ≤ 1.1 MHz	40 µA	71 µA, f = 400 kHz
101	1.1 MHz < f ≤ 3.1 MHz	120 µA	193 µA, f = 400 kHz
110	3.1 MHz < f ≤ 8.2 MHz	550 µA	940 µA, f = 8 MHz
111	8.2 MHz < f ≤ 25 MHz	2.6 mA	3.9 mA, f = 25 MHz

When the crystal oscillator is first enabled, the external oscillator valid detector allows software to determine when the external system clock has stabilized. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure for starting the crystal is as follows:

1. Configure XTAL1 and XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD => 1.
4. Switch the system clock to the external oscillator.

24.3.2. External RC Mode

If an RC network is used as the external oscillator, the circuit should be configured as shown in Figure 24.1, Option 2. The RC network should be added to XTAL2, and XTAL2 should be configured for analog I/O with the digital output drivers disabled. XTAL1 is not affected in RC mode.

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. The resistor should be no smaller than 10 kΩ. The oscillation frequency can be determined by the following equation:

$$f = \frac{1.23 \times 10^3}{R \times C}$$

Where

f = frequency of clock in MHz

R = pull-up resistor value in kΩ

V_{DD} = power supply voltage in Volts

C = capacitor value on the XTAL2 pin in pF

The equation above gives the oscillation frequency at the pin. There is an additional divide-by-2 stage internally to ensure the clock is properly conditioned to be used as SYSCLK or with internal peripherals.

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. For example, if the frequency desired is 100 kHz, let R = 246 kΩ, and C = 50 pF:

$$f = \frac{1.23 \times 10^3}{R \times C} = \frac{1.23 \times 10^3}{246 \times 50} = 100 \text{ kHz}$$

Where

f = frequency of clock in MHz

R = pull-up resistor value in kΩ

V_{DD} = power supply voltage in Volts

C = capacitor value on the XTAL2 pin in pF

Referencing Table 24.2, the recommended XFCN setting is 010.

Table 24.2. Recommended XFCN Settings for RC and C Modes

XFCN	Approximate Frequency Range (RC and C Mode)	K Factor (C Mode)	Typical Supply Current/ Actual Measured Frequency (C Mode, VDD = 2.4 V)
000	f ≤ 25 kHz	K Factor = 0.87	3.0 µA, f = 11 kHz, C = 33 pF
001	25 kHz < f ≤ 50 kHz	K Factor = 2.6	5.5 µA, f = 33 kHz, C = 33 pF
010	50 kHz < f ≤ 100 kHz	K Factor = 7.7	13 µA, f = 98 kHz, C = 33 pF
011	100 kHz < f ≤ 200 kHz	K Factor = 22	32 µA, f = 270 kHz, C = 33 pF
100	200 kHz < f ≤ 400 kHz	K Factor = 65	82 µA, f = 310 kHz, C = 46 pF
101	400 kHz < f ≤ 800 kHz	K Factor = 180	242 µA, f = 890 kHz, C = 46 pF
110	800 kHz < f ≤ 1.6 MHz	K Factor = 664	1.0 mA, f = 2.0 MHz, C = 46 pF
111	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590	4.6 mA, f = 6.8 MHz, C = 46 pF

When the RC oscillator is first enabled, the external oscillator valid detector allows software to determine when oscillation has stabilized. The recommended procedure for starting the RC oscillator is as follows:

1. Configure XTAL2 for analog I/O and disable the digital output drivers.
2. Configure and enable the external oscillator.
3. Poll for XTLVLD => 1.
4. Switch the system clock to the external oscillator.

24.3.3. External Capacitor Mode

If a capacitor is used as the external oscillator, the circuit should be configured as shown in Figure 24.1, Option 3. The capacitor should be added to XTAL2, and XTAL2 should be configured for analog I/O with the digital output drivers disabled. XTAL1 is not affected in RC mode.

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. The oscillation frequency and the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register can be determined by the following equation:

$$f = \frac{KF}{C \times V_{DD}}$$

Where

f = frequency of clock in MHz

R = pull-up resistor value in kΩ

V_{DD} = power supply voltage in Volts

C = capacitor value on the XTAL2 pin in pF

The equation above gives the oscillation frequency at the pin. There is an additional divide-by-2 stage internally to ensure the clock is properly conditioned to be used as SYSCLK or with internal peripherals.

Below is an example of selecting the capacitor and finding the frequency of oscillation Assume V_{DD} = 3.0 V and f = 150 kHz:

$$f = \frac{KF}{C \times V_{DD}}$$

$$0.150\ \text{MHz} = \frac{KF}{C \times 3.0}$$

Since a frequency of roughly 150 kHz is desired, select the K Factor from Table 24.2 as KF = 22:

$$0.150\ \text{MHz} = \frac{22}{C \times 3.0\ \text{V}}$$

$$C = \frac{22}{0.150\ \text{MHz} \times 3.0\ \text{V}}$$

$$C = 48.8\ \text{pF}$$

Therefore, the XFCN value to use in this example is 011 and C is approximately 50 pF.

The recommended startup procedure for C mode is the same as RC mode.

24.3.4. External CMOS Clock Mode

If an external CMOS clock is used as the external oscillator, the clock should be directly routed into XTAL2. The XTAL2 pin should be configured as a digital input. XTAL1 is not used in external CMOS clock mode.

The external oscillator valid detector will always return zero when the external oscillator is configured to External CMOS Clock mode.

24.4. Special Function Registers for Selecting and Configuring the System Clock

The clocking sources on C8051F97x devices are enabled and configured using the OSCICN, OSCICL, OSCXCN and the SmaRTClock internal registers. See Section “21. SmaRTClock (Real Time Clock)” on page 292 for SmaRTClock register descriptions. The system clock source for the MCU can be selected using the CLKSEL register. To minimize active mode current, the oneshot timer which sets Flash read time should by bypassed when the system clock is greater than 10 MHz. See the FLSCL register description for details.

The clock selected as the system clock can be divided by 1, 2, 4, 8, 16, 32, 64, or 128. When switching between two clock divide values, the transition may take up to 128 cycles of the undivided clock source. The CLKRDY flag can be polled to determine when the new clock divide value has been applied. The clock divider must be set to "divide by 1" when entering Suspend or Sleep Mode.

The system clock source may also be switched on-the-fly. The switchover takes effect after one clock period of the slower oscillator.

24.5. Clock Selection Control Registers

Register 24.1. CLKSEL: Clock Select

Bit	7	6	5	4	3	2	1	0
Name	CLKRDY	CLKDIV			Reserved	CLKSL		
Type	R	RW			R	RW		
Reset	0	0	1	1	0	1	0	0
SFR Page = 0x0; SFR Address: 0xA9								

Bit	Name	Function
7	CLKRDY	**System Clock Divider Clock Ready Flag.** 0: The selected clock divide setting has not been applied to the system clock. 1: The selected clock divide setting has been applied to the system clock.
6:4	CLKDIV	**Clock Source Divider.** This field controls the divider applied to the clock source selected by CLKSL. The output of this divider is the system clock (SYSCLK). 000: SYSCLK is equal to selected clock source divided by 1. 001: SYSCLK is equal to selected clock source divided by 2. 010: SYSCLK is equal to selected clock source divided by 4. 011: SYSCLK is equal to selected clock source divided by 8. 100: SYSCLK is equal to selected clock source divided by 16. 101: SYSCLK is equal to selected clock source divided by 32. 110: SYSCLK is equal to selected clock source divided by 64. 111: SYSCLK is equal to selected clock source divided by 128.
3	Reserved	Must write reset value.
2:0	CLKSL	**Clock Source Select.** Selects the oscillator to be used as the undivided system clock source. 000: Clock derived from the internal precision High-Frequency Oscillator. 001: Clock derived from the External Oscillator circuit. 010: Clock derived from the Internal Low Power Oscillator divided by 8. 011: Clock derived from the RTC. 100: Clock derived from the Internal Low Power Oscillator. 101-111: Reserved.

Register 24.2. PCLKEN: Low Power Peripheral Clock Enable

Bit	7	6	5	4	3	2	1	0
Name	PCLKEN7	PCLKEN6	PCLKEN5	PCLKEN4	PCLKEN3	PCLKEN2	PCLKEN1	PCLKEN0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xF6								

Bit	Name	Function
7	PCLKEN7	**Low Power Active Mode Peripheral Set 3 Enable.** 0: Disable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power active mode. 1: Enable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power active mode.
6	PCLKEN6	**Low Power Active Mode Peripheral Set 2 Enable.** 0: Disable clocks of PCA0, CS0, and ADC0 in low power active mode. 1: Enable clocks of PCA0, CS0, and ADC0 in low power active mode.
5	PCLKEN5	**Low Power Active Mode Peripheral Set 1 Enable.** 0: Disable clocks of MAC0 in low power active mode. 1: Enable clocks of MAC0 in low power active mode.
4	PCLKEN4	**Low Power Active Mode Peripheral Set 0 Enable.** 0: Disable clocks of UART0, Timer 3, SPI0, and I2C0 in low power active mode. 1: Enable clocks of UART0, Timer 3, SPI0, and I2C0 in low power active mode.
3	PCLKEN3	**Low Power Idle Mode Peripheral Set 3 Enable.** 0: Disable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power idle mode. 1: Enable clocks of Timer 0, 1, 2, CRC0, C2, RTC0, and Port Match in low power idle mode.
2	PCLKEN2	**Low Power Idle Mode Peripheral Set 2 Enable.** 0: Disable clocks of PCA0, CS0, and ADC0 in low power idle mode. 1: Enable clocks of PCA0, CS0, and ADC0 in low power idle mode.
1	PCLKEN1	**Low Power Idle Mode Peripheral Set 1 Enable.** 0: Disable clocks of MAC0 in low power idle mode. 1: Enable clocks of MAC0 in low power idle mode.
0	PCLKEN0	**Low Power Idle Mode Peripheral Set 0 Enable.** 0: Disable clocks of UART0, Timer 3, SPI0, and I2C0 in low power idle mode. 1: Enable clocks of UART0, Timer 3, SPI0, and I2C0 in low power idle mode.

Register 24.3. CLKMODE: Clock Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved					LPME	ECSR	FCAM
Type	R					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xF7								

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	LPME	**Low Power Mode Enable.** Setting this bit allows the device to enter low power active/idle mode. PCLKEN settings are only enabled when this bit is set.
1	ECSR	**Clock Request Enable.** When this bit is set, the source clocks will only be requested when an incoming request is active or a peripheral clock is enabled.
0	FCAM	**Force Clock Tree Enable.** When this bit is set, clock gating will only be performed in idle mode. This allows a user to have complete access to all SFRs in active mode, but still get the benefits of clock gating in idle mode.

24.6. High Frequency Oscillator Registers

Register 24.4. OSCICL: High Frequency Oscillator Calibration

Bit	7	6	5	4	3	2	1	0
Name	SSE	OSCICL						
Type	RW	RW						
Reset	0	X	X	X	X	X	X	X
SFR Page = 0x0; SFR Address: 0xAF								

Bit	Name	Function
7	SSE	**Spread Spectrum Enable.** 0: Spread Spectrum clock dithering disabled. 1: Spread Spectrum clock dithering enabled.
6:0	OSCICL	**Oscillator Calibration.** These bits determine the internal oscillator period. When set to 00000000b, the oscillator operates at its fastest setting. When set to 11111111b, the oscillator operates at its slowest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

Register 24.5. OSCICN: High Frequency Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	IOSCEN	IFRDY	Reserved					
Type	RW	R	RW					
Reset	0	0	X	X	X	X	X	X
SFR Page = 0x0; SFR Address: 0xB2								

Bit	Name	Function
7	IOSCEN	**High Frequency Oscillator Enable.** 0: High Frequency Oscillator disabled. 1: High Frequency Oscillator enabled.
6	IFRDY	**Internal Oscillator Frequency Ready Flag.** 0: High Frequency Oscillator is not running at its programmed frequency. 1: High Frequency Oscillator is running at its programmed frequency.
5:0	Reserved	Must write reset value.

Notes:
1. Read-modify-write operations such as ORL and ANL must be used to set or clear the enable bit of this register.
2. OSCBIAS (REG0CN.4) must be set to 1 before enabling the High Frequency Oscillator.

24.7. External Oscillator Registers

Register 24.6. OSCXCN: External Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	XCLKVLD	XOSCMD			Reserved	XFCN		
Type	R	RW			RW	RW		
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xB1								

Bit	Name	Function
7	XCLKVLD	**External Oscillator Valid Flag.** Provides External Oscillator status and is valid at all times for all modes of operation except External CMOS Clock Mode and External CMOS Clock Mode with divide by 2. In these modes, XCLKVLD always returns 0. 0: External Oscillator is unused or not yet stable. 1: External Oscillator is running and stable.
6:4	XOSCMD	**External Oscillator Mode.** 000: External Oscillator circuit disabled. 001: Reserved. 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.
3	Reserved	Must write reset value.
2:0	XFCN	**External Oscillator Frequency Control.** Controls the external oscillator bias current. The value selected for this field depends on the frequency range of the external oscillator.

25. SmaRTClock (Real Time Clock, RTC0)

C8051F97x devices include an ultra low power 32-bit SmaRTClock Peripheral (Real Time Clock) with alarm. The SmaRTClock has a dedicated 32 kHz oscillator that can be configured for use with or without a crystal. No external resistor or loading capacitors are required. The on-chip loading capacitors are programmable to 16 discrete levels allowing compatibility with a wide range of crystals. The SmaRTClock can operate directly from a 0.9–3.6 V battery voltage and remains operational even when the device goes into its lowest power down mode. C8051F97x devices also support an ultra low power internal LFO that reduces sleep mode current.

The SmaRTClock allows a maximum of 36 hour 32-bit independent time-keeping when used with a 32.768 kHz Watch Crystal. The SmaRTClock provides an Alarm and Missing SmaRTClock events, which could be used as reset or wakeup sources. See Section “27. Reset Sources and Supply Monitor” on page 322 and Section “16. Power Management” on page 94 for details on reset sources and low-power mode wake-up sources, respectively.

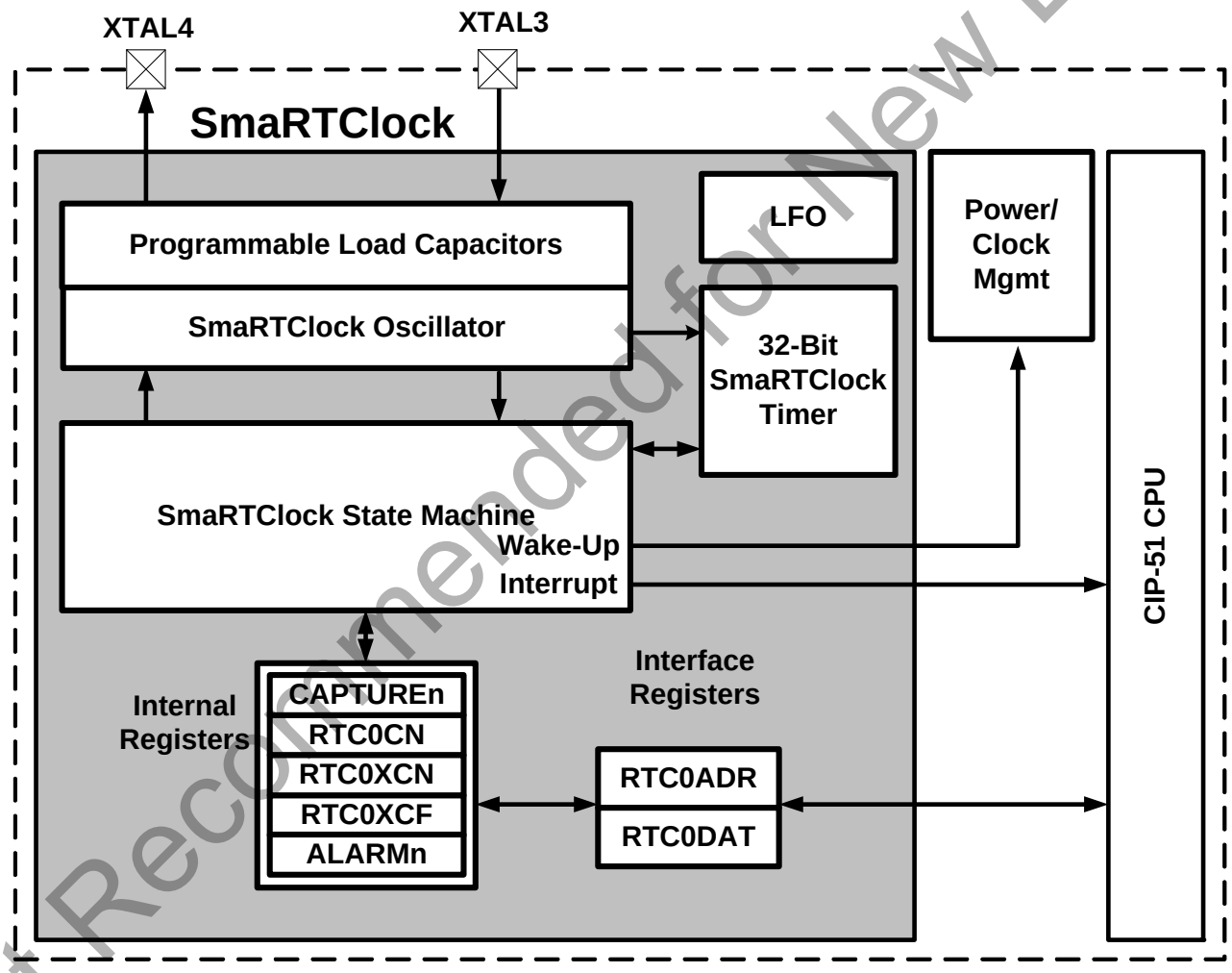


Figure 25.1. SmaRTClock Block Diagram

25.1. SmaRTClock Interface

The SmaRTClock Interface consists of two registers: RTC0ADR and RTC0DAT. These interface registers are located on the CIP-51's SFR map and provide access to the SmaRTClock internal registers listed in Table 25.1. The SmaRTClock internal registers can only be accessed indirectly through the SmaRTClock Interface.

Table 25.1. SmaRTClock Internal Registers

SmaRTClock Address	SmaRTClock Register	Register Name	Description
0x00–0x03	CAPTUREn	SmaRTClock Capture Registers	Four Registers used for setting the 32-bit SmaRTClock timer or reading its current value.
0x04	RTC0CN	SmaRTClock Control Register	Controls the operation of the SmaRTClock State Machine.
0x05	RTC0XCN	SmaRTClock Oscillator Control Register	Controls the operation of the SmaRTClock Oscillator.
0x06	RTC0XCF	SmaRTClock Oscillator Configuration Register	Controls the value of the progammable oscillator load capacitance and enables/disables AutoStep.
0x08–0x0B	ALARMn	SmaRTClock Alarm Registers	Four registers used for setting or reading the 32-bit SmaRTClock alarm value.

25.1.1. Using RTC0ADR and RTC0DAT to Access SmaRTClock Internal Registers

The SmaRTClock internal registers can be read and written using RTC0ADR and RTC0DAT. The RTC0ADR register selects the SmaRTClock internal register that will be targeted by subsequent reads or writes. Recommended instruction timing is provided in this section. If the recommended instruction timing is not followed, then BUSY (RTC0ADR.7) should be checked prior to each read or write operation to make sure the SmaRTClock Interface is not busy performing the previous read or write operation. A SmaRTClock Write operation is initiated by writing to the RTC0DAT register. Below is an example of writing to a SmaRTClock internal register.

1. Poll BUSY (RTC0ADR.7) until it returns 0 or follow recommended instruction timing.
2. Write 0x05 to RTC0ADR. This selects the internal RTC0XCN register at SmaRTClock Address 0x05.
3. Write 0x00 to RTC0DAT. This operation writes 0x00 to the internal RTC0CN register.

A SmaRTClock Read operation is initiated by setting the SmaRTClock Interface Busy bit. This transfers the contents of the internal register selected by RTC0ADR to RTC0DAT. The transferred data will remain in RTC0DAT until the next read or write operation. Below is an example of reading a SmaRTClock internal register.

1. Poll BUSY (RTC0ADR.7) until it returns 0 or follow recommended instruction timing.
2. Write 0x05 to RTC0ADR. This selects the internal RTC0XCN register at SmaRTClock Address 0x05.
3. Write 1 to BUSY. This initiates the transfer of data from RTC0XCN to RTC0DAT.
4. Poll BUSY (RTC0ADR.7) until it returns 0 or follow recommend instruction timing.
5. Read data from RTC0DAT. This data is a copy of the RTC0XCN register.

Note: The RTC0ADR and RTC0DAT registers will retain their state upon a device reset.

25.1.2. RTC0ADR Short Strobe Feature

Reads and writes to indirect SmaRTClock registers normally take 7 system clock cycles. To minimize the indirect register access time, the Short Strobe feature decreases the read and write access time to 6 system clocks. The Short Strobe feature is automatically enabled on reset and can be manually enabled/disabled using the SHORT (RTC0ADR.4) control bit.

Recommended Instruction Timing for a single register read with short strobe enabled:

```
mov RTC0ADR, #095h
nop
nop
```

```
nop
mov A, RTC0DAT
```

Recommended Instruction Timing for a single register write with short strobe enabled:

```
mov RTC0ADR, #0h
mov RTC0DAT, #000h
nop
```

25.1.3. SmaRTClock Interface Autoread Feature

When Autoread is enabled, each read from RTC0DAT initiates the next indirect read operation on the SmaRTClock internal register selected by RTC0ADR. Software should set the BUSY bit once at the beginning of each series of consecutive reads. Software should follow recommended instruction timing or check if the SmaRTClock Interface is busy prior to reading RTC0DAT. Autoread is enabled by setting AUTORD (RTC0ADR.6) to logic 1.

25.1.4. RTC0ADR Autoincrement Feature

For ease of reading and writing the 32-bit CAPTURE and ALARM values, RTC0ADR automatically increments after each read or write to a CAPTUREn or ALARMn register. This speeds up the process of setting an alarm or reading the current SmaRTClock timer value. Autoincrement is always enabled.

Recommended Instruction Timing for a multi-byte register read with short strobe and auto read enabled:

```
mov RTC0ADR, #0d0h
nop
nop
nop
mov A, RTC0DAT
nop
nop
mov A, RTC0DAT
nop
nop
mov A, RTC0DAT
nop
nop
mov A, RTC0DAT
```

Recommended Instruction Timing for a multi-byte register write with short strobe enabled:

```
mov RTC0ADR, #010h
mov RTC0DAT, #05h
nop
mov RTC0DAT, #06h
nop
mov RTC0DAT, #07h
nop
mov RTC0DAT, #08h
nop
```

25.2. SmaRTClock Clocking Sources

The SmaRTClock peripheral is clocked from its own timebase, independent of the system clock. The SmaRTClock timebase can be derived from the internal LFO or the SmaRTClock oscillator circuit, which has two modes of operation: Crystal Mode, and Self-Oscillate Mode. The oscillation frequency is 32.768 kHz in Crystal Mode and can be programmed in the range of 10 kHz to 40 kHz in Self-Oscillate Mode. The internal LFO frequency is 16.4 kHz ±20%. The frequency of the SmaRTClock oscillator can be measured with respect to another oscillator using an on-chip timer. See Section “32. Timers (Timer0, Timer1, Timer2, and Timer3)” on page 389 for more information on how this can be accomplished.

Note: The SmaRTClock timebase can be selected as the system clock and routed to a port pin. See Section “24. Clocking Sources” on page 241 for information on selecting the system clock source and Section “26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)” on page 277 for information on how to route the system clock to a port pin.

25.2.1. Using the SmaRTClock Oscillator with a Crystal

When using Crystal Mode, a 32.768 kHz crystal should be connected between XTAL3 and XTAL4. No other external components are required. The following steps show how to start the SmaRTClock crystal oscillator in software:

1. Configure the XTAL3 and XTAL4 pins for Analog I/O.
2. Set SmaRTClock to Crystal Mode (XMODE = 1).
3. Disable Automatic Gain Control (AGCINT) and enable Bias Doubling (BIASX2) for fast crystal startup.
4. Set the desired loading capacitance (RTC0XCF).
5. Enable power to the SmaRTClock oscillator circuit (RTC0INT = 1).
6. Wait 20 ms.
7. Poll the SmaRTClock Clock Valid Bit (CLKVLD) until the crystal oscillator stabilizes.
8. Poll the SmaRTClock Load Capacitance Ready Bit (LOADRDY) until the load capacitance reaches its programmed value.
9. Enable Automatic Gain Control (AGCINT) and disable Bias Doubling (BIASX2) for maximum power savings.
10. Enable the SmaRTClock missing clock detector.
11. Wait 2 ms.
12. Clear the PMU0CF wake-up source flags.

25.2.2. Using the SmaRTClock Oscillator in Self-Oscillate Mode

When using Self-Oscillate Mode, the XTAL3 and XTAL4 pins are internally shorted together. The following steps show how to configure SmaRTClock for use in Self-Oscillate Mode:

1. Set SmaRTClock to Self-Oscillate Mode (XMODE = 0).
2. Set the desired oscillation frequency:
   For oscillation at about 20 kHz, set BIASX2 = 0.
   For oscillation at about 40 kHz, set BIASX2 = 1.
3. The oscillator starts oscillating instantaneously.
4. Fine tune the oscillation frequency by adjusting the load capacitance (RTC0XCF).

25.2.3. Using the Low Frequency Oscillator (LFO)

The low frequency oscillator provides an ultra low power, on-chip clock source to the SmaRTClock. The typical frequency of oscillation is 16.4 kHz ±20%. No external components are required to use the LFO and the XTAL3 and XTAL4 pins do not need to be shorted together.

The following steps show how to configure SmaRTClock for use with the LFO:

1. Enable and select the Low Frequency Oscillator (LFOINT = 1).
2. The LFO starts oscillating instantaneously.

When the LFO is enabled, the SmaRTClock oscillator increments bit 1 of the 32-bit timer (instead of bit 0). This effectively multiplies the LFO frequency by 2, making the RTC timebase behave as if a 32.768 kHz crystal is connected at the output.

25.2.4. Programmable Load Capacitance

The programmable load capacitance has 16 values to support crystal oscillators with a wide range of recommended load capacitance. If Automatic Load Capacitance Stepping is enabled, the crystal load capacitors start at the smallest setting to allow a fast startup time, then slowly increase the capacitance until the final programmed value is reached. The final programmed loading capacitor value is specified using the LOADCAP bits in the RTC0XCF register. The LOADCAP setting specifies the amount of on-chip load capacitance and does not include any stray PCB capacitance. Once the final programmed loading capacitor value is reached, the LOADRDY flag will be set by hardware to logic 1.

When using the SmaRTClock oscillator in Self-Oscillate mode, the programmable load capacitance can be used to fine tune the oscillation frequency. In most cases, increasing the load capacitor value will result in a decrease in oscillation frequency.Table 25.2 shows the crystal load capacitance for various settings of LOADCAP.

Table 25.2. SmaRTClock Load Capacitance Settings

LOADCAP	Crystal Load Capacitance	Equivalent Capacitance seen on XTAL3 and XTAL4
0000	4.0 pF	8.0 pF
0001	4.5 pF	9.0 pF
0010	5.0 pF	10.0 pF
0011	5.5 pF	11.0 pF
0100	6.0 pF	12.0 pF
0101	6.5 pF	13.0 pF
0110	7.0 pF	14.0 pF
0111	7.5 pF	15.0 pF
1000	8.0 pF	16.0 pF
1001	8.5 pF	17.0 pF
1010	9.0 pF	18.0 pF
1011	9.5 pF	19.0 pF
1100	10.5 pF	21.0 pF
1101	11.5 pF	23.0 pF
1110	12.5 pF	25.0 pF
1111	13.5 pF	27.0 pF

25.2.5. Automatic Gain Control (Crystal Mode Only) and SmaRTClock Bias Doubling

Automatic Gain Control allows the SmaRTClock oscillator to trim the oscillation amplitude of a crystal in order to achieve the lowest possible power consumption. Automatic Gain Control automatically detects when the oscillation amplitude has reached a point where it safe to reduce the drive current, therefore, it may be enabled during crystal startup. It is recommended to enable Automatic Gain Control in most systems which use the SmaRTClock oscillator in Crystal Mode. The following are recommended crystal specifications and operating conditions when Automatic Gain Control is enabled:

- ESR < 50 kΩ
- Load Capacitance < 10 pF
- Supply Voltage < 3.0 V
- Temperature > –20 °C

When using Automatic Gain Control, it is recommended to perform an oscillation robustness test to ensure that the chosen crystal will oscillate under the worst case condition to which the system will be exposed. The worst case condition that should result in the least robust oscillation is at the following system conditions: lowest temperature, highest supply voltage, highest ESR, highest load capacitance, and lowest bias current (AGC enabled, Bias Double Disabled).

To perform the oscillation robustness test, the SmaRTClock oscillator should be enabled and selected as the system clock source. Next, the SYSCLK signal should be routed to a port pin configured as a push-pull digital output. The positive duty cycle of the output clock can be used as an indicator of oscillation robustness. As shown in Figure 25.2, duty cycles less than 55% indicate a robust oscillation. As the duty cycle approaches 60%, oscillation becomes less reliable and the risk of clock failure increases. Increasing the bias current (by disabling AGC) will always improve oscillation robustness and will reduce the output clock's duty cycle. This test should be performed at the worst case system conditions, as results at very low temperatures or high supply voltage will vary from results taken at room temperature or low supply voltage.

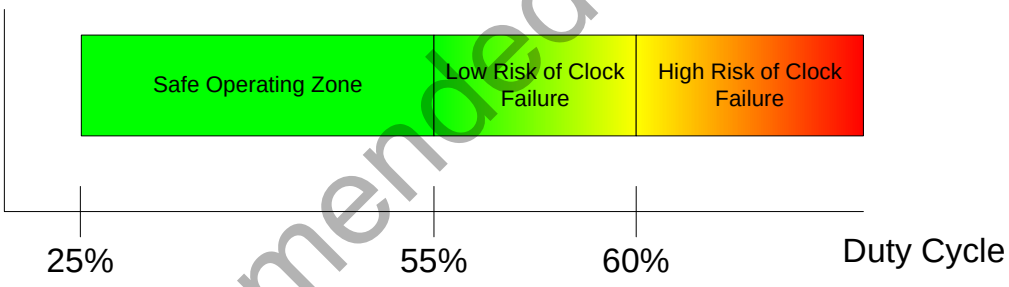


Figure 25.2. Interpreting Oscillation Robustness (Duty Cycle) Test Results

As an alternative to performing the oscillation robustness test, Automatic Gain Control may be disabled at the cost of increased power consumption (approximately 200 nA). Disabling Automatic Gain Control will provide the crystal oscillator with higher immunity against external factors which may lead to clock failure. Automatic Gain Control must be disabled if using the SmaRTClock oscillator in self-oscillate mode.

Table 25.3 shows a summary of the oscillator bias settings. The SmaRTClock Bias Doubling feature allows the self-oscillation frequency to be increased (almost doubled) and allows a higher crystal drive strength in crystal mode. High crystal drive strength is recommended when the crystal is exposed to poor environmental conditions such as excessive moisture. SmaRTClock Bias Doubling is enabled by setting BIASX2 (RTC0XCN.5) to 1.

Table 25.3. SmaRTClock Bias Settings

Mode	Setting	Power Consumption
Crystal	Bias Double Off, AGC On	Lowest 600 nA
	Bias Double Off, AGC Off	Low 800 nA
	Bias Double On, AGC On	High
	Bias Double On, AGC Off	Highest
Self-Oscillate	Bias Double Off	Low
	Bias Double On	High

25.2.6. Missing SmaRTClock Detector

The missing SmaRTClock detector is a one-shot circuit enabled by setting MCLKINT (RTC0CN.6) to 1. When the SmaRTClock Missing Clock Detector is enabled, OSCFAIL (RTC0CN.5) is set by hardware if SmaRTClock oscillator remains high or low for more than 100 µs.

A SmaRTClock Missing Clock detector timeout can trigger an interrupt, wake the device from a low power mode, or reset the device. See Section “13. Interrupts” on page 79, Section “16. Power Management” on page 94, and Section “27. Reset Sources and Supply Monitor” on page 322 for more information.

Note: The SmaRTClock Missing Clock Detector should be disabled when making changes to the oscillator settings in RTC0XCN.

25.2.7. SmaRTClock Oscillator Crystal Valid Detector

The SmaRTClock oscillator crystal valid detector is an oscillation amplitude detector circuit used during crystal startup to determine when oscillation has started and is nearly stable. The output of this detector can be read from the CLKVLD bit (RTX0XCN.4).

Notes:

1. The CLKVLD bit has a blanking interval of 2 ms. During the first 2 ms after turning on the crystal oscillator, the output of CLKVLD is not valid.
2. This SmaRTClock crystal valid detector (CLKVLD) is not intended for detecting an oscillator failure. The missing SmaRTClock detector (CLKFAIL) should be used for this purpose.

25.3. SmaRTClock Timer and Alarm Function

The SmaRTClock timer is a 32-bit counter that, when running (RTC0TR = 1), is incremented every SmaRTClock oscillator cycle. The timer has an alarm function that can be set to generate an interrupt, wake the device from a low power mode, or reset the device at a specific time. See Section “13. Interrupts” on page 79, Section “16. Power Management” on page 94, and Section “27. Reset Sources and Supply Monitor” on page 322 for more information.

The SmaRTClock timer includes an Auto Reset feature, which automatically resets the timer to zero one SmaRTClock cycle after an alarm occurs. When using Auto Reset, the Alarm match value should always be set to 1 count less than the desired match value. Auto Reset can be enabled by writing a 1 to ALRM (RTC0CN.2).

25.3.1. Setting and Reading the SmaRTClock Timer Value

The 32-bit SmaRTClock timer can be set or read using the six CAPTUREn internal registers. Note that the timer does not need to be stopped before reading or setting its value. The following steps can be used to set the timer value:

1. Write the desired 32-bit set value to the CAPTUREn registers.
2. Write 1 to RTC0SET. This will transfer the contents of the CAPTUREn registers to the SmaRTClock timer.
3. Operation is complete when RTC0SET is cleared to 0 by hardware.

The following steps can be used to read the current timer value:

1. Write 1 to RTC0CAP. This will transfer the contents of the timer to the CAPTUREn registers.
2. Poll RTC0CAP until it is cleared to 0 by hardware.
3. A snapshot of the timer value can be read from the CAPTUREn registers

25.3.2. Setting a SmaRTClock Alarm

The SmaRTClock alarm function compares the 32-bit value of SmaRTClock Timer to the value of the ALARMn registers. An alarm event is triggered if the SmaRTClock timer is **equal to** the ALARMn registers. If Auto Reset is enabled, the 32-bit timer will be cleared to zero one SmaRTClock cycle after the alarm event.

The SmaRTClock alarm event can be configured to reset the MCU, wake it up from a low power mode, or generate an interrupt. See Section “13. Interrupts” on page 79, Section “16. Power Management” on page 94, and Section “27. Reset Sources and Supply Monitor” on page 322 for more information.

The following steps can be used to set up a SmaRTClock Alarm:

1. Disable SmaRTClock Alarm Events (RTC0AINT = 0).
2. Set the ALARMn registers to the desired value.
3. Enable SmaRTClock Alarm Events (RTC0AINT = 1).

When using the SmaRTClock in Self-Oscillate or Crystal Modes, the alarm is triggered every N + 2 RTC cycles except for the first alarm, which triggers after N RTC cycles. N is the value written into the 32-bit ALARM register.

When using the SmaRTClock with the internal LFO, the alarm is triggered every N/2 + 2 RTC cycles except for the first alarm, which triggers after N/2 RTC cycles. N is the value written into the 32-bit ALARM register. If N is odd, then the hardware uses N/2 rounded down.

Notes:

1. The ALRM bit, which is used as the SmaRTClock Alarm Event flag, is cleared by disabling SmaRTClock Alarm Events (RTC0AINT = 0).
2. If AutoReset is disabled, disabling (RTC0AINT = 0) then Re-enabling Alarm Events (RTC0AINT = 1) after a SmaRTClock Alarm without modifying ALARMn registers will automatically schedule the next alarm after 2^32 SmaRTClock cycles (approximately 36 hours using a 32.768 kHz crystal).
3. The SmaRTClock Alarm Event flag will remain asserted for a maximum of one SmaRTClock cycle. See Section “16. Power Management” on page 94 for information on how to capture a SmaRTClock Alarm event using a flag which is not automatically cleared by hardware.
4. When an external clock is used as the SmaRTClock source, the system clock speed must be more than 9 times the SmaRTClock speed for the SmaRTClock alarm functionality to work correctly.
5. When an external clock source is used as the SmaRTClock source, the system clock speed must be more than 5 times the SmaRTClock speed for the SmaRTClock alarm functionality to work correctly.

25.3.3. Software Considerations for Using the SmaRTClock Timer and Alarm

The SmaRTClock timer and alarm have two operating modes to suit varying applications. The two modes are described below:

Mode 1:

The first mode uses the SmaRTClock timer as a perpetual timebase which is never reset to zero. Every 36 hours, the timer is allowed to overflow without being stopped or disrupted. The alarm interval is software managed and is added to the ALRMn registers by software after each alarm. This allows the alarm match value to always stay ahead of the timer by one software managed interval. If software uses 32-bit unsigned addition to increment the alarm match value, then it does not need to handle overflows since both the timer and the alarm match value will overflow in the same manner.

This mode is ideal for applications which have a long alarm interval (e.g., 24 or 36 hours) and/or have a need for a perpetual timebase. An example of an application that needs a perpetual timebase is one whose wake-up interval is constantly changing. For these applications, software can keep track of the number of timer overflows in a 16-bit variable, extending the 32-bit (36 hour) timer to a 48-bit (272 year) perpetual timebase.

Mode 2:

The second mode uses the SmaRTClock timer as a general purpose up counter which is auto reset to zero by hardware after each alarm. The alarm interval is managed by hardware and stored in the ALRMn registers. Software only needs to set the alarm interval once during device initialization. After each alarm, software should keep a count of the number of alarms that have occurred in order to keep track of time.

This mode is ideal for applications that require minimal software intervention and/or have a fixed alarm interval. This mode is the most power efficient since it requires less CPU time per alarm.

Important Note: The alarm interval will be 2 more SmaRTClock cycles than the expected value because 1 cycle holds the alarm signal high and another cycle is used to reset the alarm.

25.4. RTC0 Control Registers

Register 25.1. RTC0ADR: RTC Address

Bit	7	6	5	4	3	2	1	0
Name	BUSY	AUTORD	Reserved	SHORT	ADDR			
Type	RW	RW	R	RW	RW			
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xAC								

Bit	Name	Function
7	BUSY	**RTC Interface Busy Indicator.** This bit indicates the RTC interface status. Writing a 1 to this bit initiates an indirect read.
6	AUTORD	**RTC Interface Autoread Enable.** When autoread is enabled, firmware should set the BUSY bit once at the beginning of each series of consecutive reads. Firmware must check if the RTC Interface is busy prior to reading RTC0DAT. 0: Disable autoread. Firmware must write the BUSY bit for each RTC indirect read operation. 1: Enable autoread. The next RTC indirect read operation is initiated when firmware reads the RTC0DAT register.
5	Reserved	Must write reset value.
4	SHORT	**Short Strobe Enable.** Enables/disables the Short Strobe feature. 0: Disable short strobe. 1: Enable short strobe.
3:0	ADDR	**RTC Indirect Register Address.** Sets the currently-selected RTC internal register.
Note: The ADDR bits increment after each indirect read/write operation that targets a CAPTUREn or ALARMn internal RTC register.		

Register 25.2. RTC0DAT: RTC Data

Bit	7	6	5	4	3	2	1	0
Name	RTC0DAT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xAD								

Bit	Name	Function
7:0	RTC0DAT	**RTC Data.** Holds data transferred to/from the internal RTC register selected by RTC0ADR.
Note: Read-modify-write instructions (orl, anl, etc.) should not be used on this register.		

Register 25.3. RTC0CN: RTC Control

Bit	7	6	5	4	3	2	1	0
Name	RTC0EN	MCLKEN	OSCFAIL	RTC0TR	RTC0AEN	ALRM	RTC0SET	RTC0CAP
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	X	0	0	0	0	0
Indirect Address: 0x04								

Bit	Name	Function
7	RTC0EN	**RTC Enable.** Enables/disables the RTC oscillator and associated bias currents. 0: Disable RTC oscillator. 1: Enable RTC oscillator.
6	MCLKEN	**Missing RTC Detector Enable.** Enables/disables the missing RTC detector. 0: Disable missing RTC detector. 1: Enable missing RTC detector.
5	OSCFAIL	**RTC Oscillator Fail Event Flag.** Set by hardware when a missing RTC detector timeout occurs. Must be cleared by firmware. The value of this bit is not defined when the RTC oscillator is disabled.
4	RTC0TR	**RTC Timer Run Control.** Controls if the RTC timer is running or stopped (holds current value). 0: RTC timer is stopped. 1: RTC timer is running.
3	RTC0AEN	**RTC Alarm Enable.** Enables/disables the RTC alarm function. Also clears the ALRM flag. 0: Disable RTC alarm. 1: Enable RTC alarm.
2	ALRM	**RTC Alarm Event Flag and Auto Reset Enable.** Reads return the state of the alarm event flag. Writes enable/disable the Auto Reset function.
1	RTC0SET	**RTC Timer Set.** Writing 1 initiates a RTC timer set operation. This bit is cleared to 0 by hardware to indicate that the timer set operation is complete.
0	RTC0CAP	**RTC Timer Capture.** Writing 1 initiates a RTC timer capture operation. This bit is cleared to 0 by hardware to indicate that the timer capture operation is complete.
Note: The ALRM flag will remain asserted for a maximum of one RTC cycle.		

Register 25.4. RTC0XCN: RTC Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	AGCEN	XMODE	BIASX2	CLKVLD	LFOEN	Reserved		
Type	RW	RW	RW	R	R	R		
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x05								

Bit	Name	Function
7	AGCEN	**RTC Oscillator Automatic Gain Control (AGC) Enable.** 0: Disable AGC. 1: Enable AGC.
6	XMODE	**RTC Oscillator Mode.** Selects Crystal or Self Oscillate Mode. 0: Self-Oscillate Mode selected. 1: Crystal Mode selected.
5	BIASX2	**RTC Oscillator Bias Double Enable.** Enables/disables the Bias Double feature. 0: Disable the Bias Double feature. 1: Enable the Bias Double feature.
4	CLKVLD	**RTC Oscillator Crystal Valid Indicator.** Indicates if oscillation amplitude is sufficient for maintaining oscillation. 0: Oscillation has not started or oscillation amplitude is too low to maintain oscillation. 1: Sufficient oscillation amplitude detected.
3	LFOEN	**Low Frequency Oscillator Enable and Select.** Overrides XMODE and selects the internal low frequency oscillator (LFOSC) as the RTC oscillator source. 0: XMODE determines RTC oscillator source. 1: LFOSC enabled and selected as RTC oscillator source.
2:0	Reserved	Must write reset value.

Register 25.5. RTC0XCF: RTC Oscillator Configuration

Bit	7	6	5	4	3	2	1	0
Name	AUTOSTP	LOADRDY	Reserved		LOADCAP			
Type	RW	R	R		RW			
Reset	0	0	0	0	X	X	X	X
Indirect Address: 0x06								

Bit	Name	Function
7	AUTOSTP	**Automatic Load Capacitance Stepping Enable.** Enables/disables automatic load capacitance stepping. 0: Disable load capacitance stepping. 1: Enable load capacitance stepping.
6	LOADRDY	**Load Capacitance Ready Indicator.** Set by hardware when the load capacitance matches the programmed value. 0: Load capacitance is currently stepping. 1: Load capacitance has reached it programmed value.
5:4	Reserved	Must write reset value.
3:0	LOADCAP	**Load Capacitance Programmed Value.** Holds the desired load capacitance value.

Register 25.6. CAPTURE0: RTC Timer Capture 0

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x00								

Bit	Name	Function
7:0	CAPTURE0	**RTC Timer Capture 0.** The CAPTURE3-CAPTURE0 registers are used to read or set the 32-bit RTC timer. Data is transferred to or from the RTC timer when the RTC0SET or RTC0CAP bits are set.

Register 25.7. CAPTURE1: RTC Timer Capture 1

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x01								

Bit	Name	Function
7:0	CAPTURE1	**RTC Timer Capture 1.** The CAPTURE3-CAPTURE0 registers are used to read or set the 32-bit RTC timer. Data is transferred to or from the RTC timer when the RTC0SET or RTC0CAP bits are set.

Register 25.8. CAPTURE2: RTC Timer Capture 2

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE2							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x02								

Bit	Name	Function
7:0	CAPTURE2	**RTC Timer Capture 2.** The CAPTURE3-CAPTURE0 registers are used to read or set the 32-bit RTC timer. Data is transferred to or from the RTC timer when the RTC0SET or RTC0CAP bits are set.

Register 25.9. CAPTURE3: RTC Timer Capture 3

Bit	7	6	5	4	3	2	1	0
Name	CAPTURE3							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x03								

Bit	Name	Function
7:0	CAPTURE3	**RTC Timer Capture 3.** The CAPTURE3-CAPTURE0 registers are used to read or set the 32-bit RTC timer. Data is transferred to or from the RTC timer when the RTC0SET or RTC0CAP bits are set.

Register 25.10. ALARM0: RTC Alarm Programmed Value 0

Bit	7	6	5	4	3	2	1	0
Name	ALARM0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x08								

Bit	Name	Function
7:0	ALARM0	**RTC Alarm Programmed Value 0.** The ALARM3-ALARM0 registers are used to set an alarm event for the RTC timer. The RTC alarm should be disabled (RTC0AEN=0) when updating these registers.

Register 25.11. ALARM1: RTC Alarm Programmed Value 1

Bit	7	6	5	4	3	2	1	0
Name	ALARM1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x09								

Bit	Name	Function
7:0	ALARM1	**RTC Alarm Programmed Value 1.** The ALARM3-ALARM0 registers are used to set an alarm event for the RTC timer. The RTC alarm should be disabled (RTC0AEN=0) when updating these registers.

Register 25.12. ALARM2: RTC Alarm Programmed Value 2

Bit	7	6	5	4	3	2	1	0
Name	ALARM2							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x0A								

Bit	Name	Function
7:0	ALARM2	**RTC Alarm Programmed Value 2.** The ALARM3-ALARM0 registers are used to set an alarm event for the RTC timer. The RTC alarm should be disabled (RTC0AEN=0) when updating these registers.

Register 25.13. ALARM3: RTC Alarm Programmed Value 3

Bit	7	6	5	4	3	2	1	0
Name	ALARM3							
Type	RW							
Reset	0	0	0	0	0	0	0	0
Indirect Address: 0x0B								

Bit	Name	Function
7:0	ALARM3	**RTC Alarm Programmed Value 3.** The ALARM3-ALARM0 registers are used to set an alarm event for the RTC timer. The RTC alarm should be disabled (RTC0AEN=0) when updating these registers.

26. Port I/O (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Crossbar, and Port Match)

Digital and analog resources on the C8051F97x family are externally available on the device's multi-purpose I/O pins. Port pins P0.0-P2.7 can be defined as general-purpose I/O (GPIO), assigned to one of the internal digital resources through the crossbar, or assigned to an analog function. Port pins P3.0-P6.1 can be used as GPIO. Port pin P5.2 is shared with the C2 Interface Data signal (C2D). 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 26.2 and Figure 26.3). The registers XBR0 and XBR1 are used to select internal digital functions.

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). Additionally, each bank of port pins (P0, P1, P2, P3, P4, and P5) have two selectable drive strength settings. The P6 pins only support digital open-drain mode and cannot be configured as digital push-pull pins.

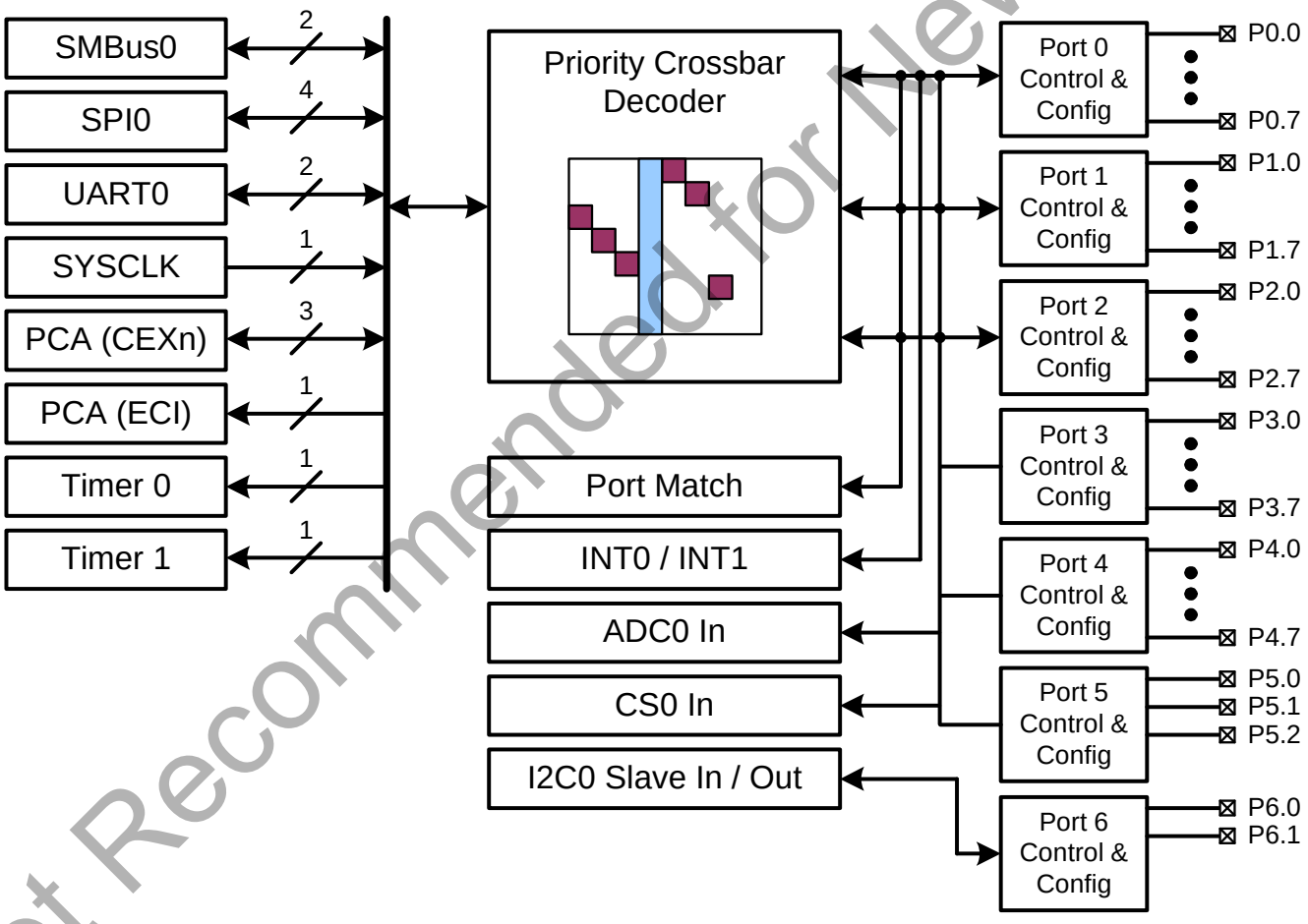


Figure 26.1. Port I/O Functional Block Diagram

26.1. General Port I/O Initialization

Port I/O initialization consists of the following steps:

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

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

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

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. When the WEAKPUD bit in XBR1 is ‘0’, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a ‘0’ 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 XBR2 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, Silicon Labs provides configuration utility software to determine the port I/O pin-assignments based on the crossbar register settings.

The crossbar must be enabled to use port pins as standard port I/O in output mode. Port output drivers of all crossbar pins are disabled whenever the crossbar is disabled.

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

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

26.2.1. Assigning Port I/O Pins to Analog Functions

Table 26.1 shows all available analog functions that require port I/O assignments. Table 26.1 shows the potential mapping of port I/O to each analog function.

Table 26.1. Port I/O Assignment for Analog Functions

Analog Function	Potentially Assignable Port Pins			SFR(s) used for Assignment
	QFN-48	QFN-32	QFN-24	
ADC Input	P0.0 – P5.2	P0.0 – P3.2, P5.2	P0.0 – P2.1, P5.2	PnMDIN, AMUX0Pn, Pn, ADC0 Registers
Capacitive Sense Input	P0.0 – P5.2	P0.0 – P3.2, P5.2	P0.0 – P2.1, P5.2	PnMDIN, AMUX0Pn, Pn, CS0 Registers
Voltage Reference (VREF)	P0.0	P0.0	P0.0	REF0CN, PnSKIP, P0.0 bit set, PnMDIN
External Oscillator Input (XTAL1)	P1.0	P0.5		OSCXCN, AMUX0Pn (cleared), Pn bit set, PnMDIN
External Oscillator Input (XTAL2)	P1.1	P0.6		OSCXCN, AMUX0Pn (cleared), P0.1 bit set, PnMDIN
SmaRTClock Oscillator Input (XTAL3)	P0.6	P0.3		RTC0CN, AMUX0Pn (cleared), P0.0 bit set, PnMDIN
SmaRTClock Oscillator Input (XTAL4)	P0.7	P0.4		RTC0CN, AMUX0Pn (cleared), P0.1 bit set, PnMDIN

26.2.2. Assigning Port I/O Pins to Digital Functions

Any port pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the crossbar for pin assignment; however, some digital functions bypass the crossbar in a manner similar to the analog functions listed above. Table 26.2 shows all digital functions available through the crossbar and the potential mapping of port I/O to each function.

Table 26.2. Port I/O Assignment for Digital Functions

Digital Function	Potentially Assignable Port Pins			SFR(s) Used for Assignment
	QFN-48	QFN-32	QFN-24	
SMBus0, UART0, SPI0, SYSCLK, PCA0 (CEX0-2 and ECI), T0, or T1.	Any port pin available for assignment by the crossbar. This includes P0.0 – P2.7 pins which have their PnSKIP bit set to '0'.			XBR0, XBR1, XBR2
Any pin used for GPIO	P0.0 – P5.2, P6.0 – P6.1	P0.0 – P3.2, P5.2, P6.0 – P6.1	P0.0 – P2.1, P5.2, P6.0 – P6.1	P0SKIP, P1SKIP, P2SKIP

26.2.3. Assigning Port I/O Pins to Fixed Digital Functions

Fixed digital functions include external clock input as well as external event trigger functions, which can be used to trigger events such as an ADC conversion, fire an interrupt or wake the device from idle mode when a transition occurs on a digital I/O pin. The fixed digital functions do not require dedicated pins and will function on both GPIO pins and pins in use by the crossbar. Fixed digital functions cannot be used on pins configured for analog I/O. Table 26.3 shows all available fixed digital functions and the potential mapping of port I/O to each function.

Table 26.3. Port I/O Assignment for Fixed Digital Functions

Function	Potentially Assignable Port Pins			SFR(s) used for Assignment
	QFN-48	QFN-32	QFN-24	
External Interrupt 0	P0.0 - P0.7			IT01CF
External Interrupt 1	P0.0 - P0.7			IT01CF
Conversion Start (CNVSTR)	P0.6			ADC0CN
Port Match	P0.0 - P2.7	P0.0 - P2.7	P0.0 - P2.1	P0MASK, P0MAT P1MASK, P1MAT P2MASK, P2MAT
I2C Slave 0	P6.0, P6.1			I2C0CN

26.3. Priority Crossbar Decoder

The priority crossbar decoder assigns a priority to each I/O function in order from top to bottom. When a digital resource is selected, the least-significant unassigned port pin is assigned to that resource. 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 P0.0 if VREF is used, XTAL1, XTAL2, XTAL3, or XTAL4 pins on a QFN-32 or QFN-48 package, P0.6 if the ADC is configured to use the external conversion start signal (CNVSTR), and any selected ADC or comparator inputs. The crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin.

Figure 26.2 shows all of the potential peripheral-to-pin assignments available to the crossbar. Note that this does not mean any peripheral can always be assigned to the highlighted pins. The actual pin assignments are determined by the priority of the enabled peripherals.

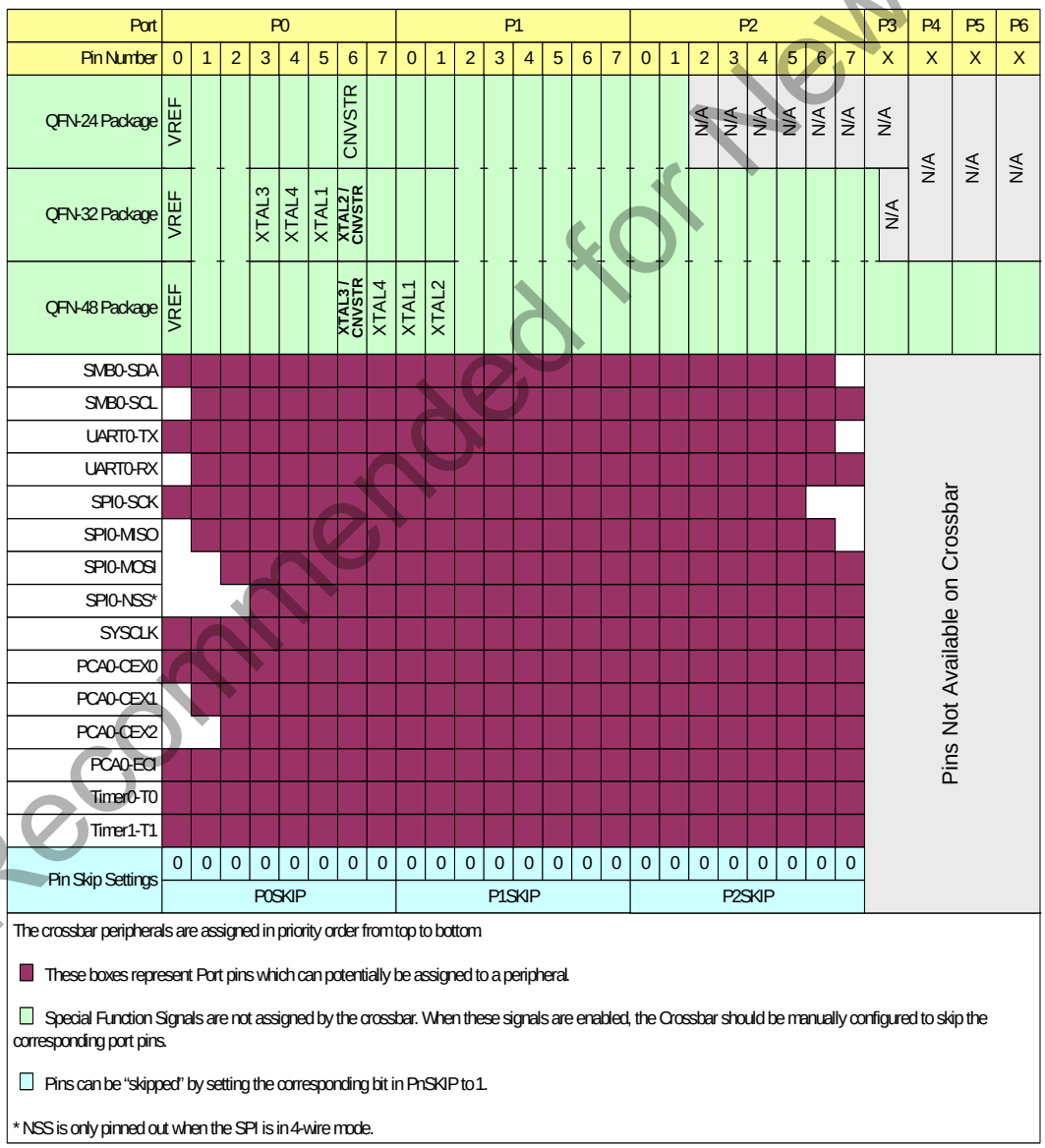


Figure 26.2. Crossbar Priority Decoder—Possible Pin Assignments

Registers XBR0 and XBR1are 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 UART0 is selected, the crossbar assigns both pins associated with UART0 (TX and RX). Standard port I/Os appear contiguously after the prioritized functions have been assigned.

Figure 26.3 shows an example of the resulting pin assignments of the device with SMBus0, SPI0, and two channels of PCA0 enabled and P0.3, P0.4, P1.0, and P1.1 skipped (P0SKIP = 0x18, P1SKIP = 0x03). SMBus0 is the highest priority and it will be assigned first. The next-highest enabled peripheral is SPI0. P0.2 is available, so SPI0 takes this pin. The next pins, MISO, MOSI, and NSS are routed to P0.5, P0.6, and P0.7, respectively, because P0.3 and P0.4 are skipped. The PCA0 CEX0 and CEX1are then routed to P1.2 and P1.3. The other pins on the device are available for use as general-purpose digital I/O or analog functions.

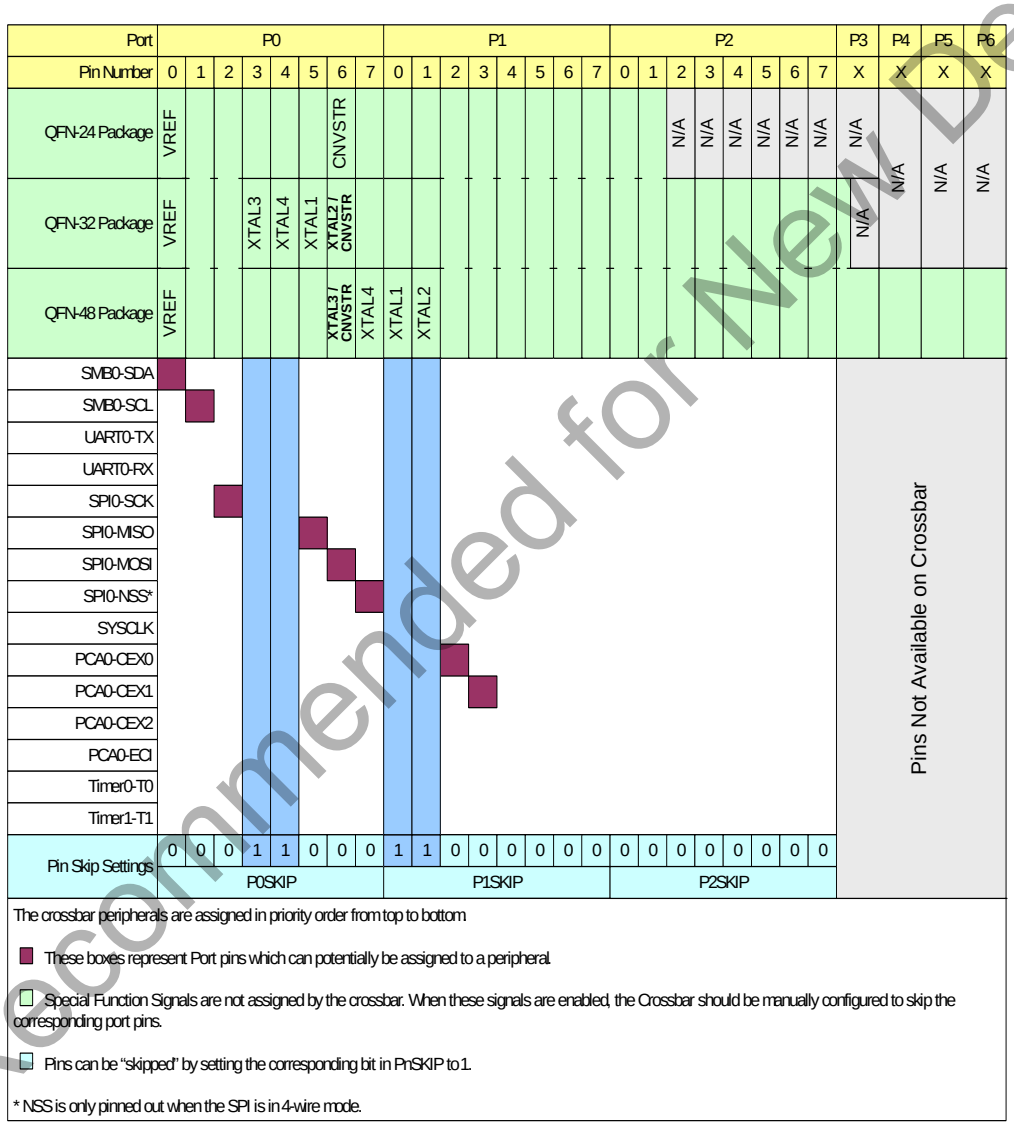


Figure 26.3. Crossbar Priority Decoder Example

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

26.4. Port I/O Modes of Operation

Port pins are configured by firmware as digital or analog I/O using the PnMDIN registers. On reset, all port I/O cells default to a high impedance state with weak pull-ups enabled. Until the crossbar is enabled, both the high and low port I/O drive circuits are explicitly disabled on all crossbar pins. Port pins configured as digital I/O may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

26.4.1. Configuring Port Pins For Analog Modes

Any pins to be used for analog functions should be configured for analog mode. When a pin is configured for analog I/O, its weak pullup, digital driver, and digital receiver are disabled. Port pins configured for analog functions will always read back a value of ‘0’ in the corresponding Pn Port Latch register. To configure a pin as analog, the following steps should be taken:

1. Clear the bit associated with the pin in the PnMDIN register to ‘0’. This selects analog mode for the pin.
2. Set the bit associated with the pin in the Pn register to ‘1’.
3. Skip the bit associated with the pin in the PnSKIP register to ensure the crossbar does not attempt to assign a function to the pin.

26.4.2. Configuring Port Pins For Digital Modes

Any pins to be used by digital peripherals or as GPIO should be configured as digital I/O (PnMDIN.n = ‘1’). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PnMDOUT registers.

Push-pull outputs (PnMDOUT.n = ‘1’) drive the port pad to the supply rails based on the output logic value of the port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the port pad to the low-side rail when the output logic value is ‘0’ and become high impedance inputs (both high low drivers turned off) when the output logic value is ‘1’.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the port pad to the high-side rail to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven low to minimize power consumption, and they may be globally disabled by setting WEAKPUD to ‘1’. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to minimize power consumption. Port pins configured for digital I/O always read back the logic state of the port pad, regardless of the output logic value of the port pin.

To configure a pin as digital input:

1. Set the bit associated with the pin in the PnMDIN register to ‘1’. This selects digital mode for the pin.
2. Clear the bit associated with the pin in the PnMDOUT register to ‘0’. This configures the pin as open-drain.
3. Set the bit associated with the pin in the Pn register to ‘1’. This tells the output driver to “drive” logic high. Because the pin is configured as open-drain, the high-side driver is not active, and the pin may be used as an input.

Open-drain outputs are configured exactly as digital inputs. However, the pin may be driven low by an assigned peripheral, or by writing ‘0’ to the associated bit in the Pn register if the signal is a GPIO.

To configure a pin as a digital, push-pull output:

1. Set the bit associated with the pin in the PnMDIN register to ‘1’. This selects digital mode for the pin.
2. Set the bit associated with the pin in the PnMDOUT register to ‘1’. This configures the pin as push-pull.

If a digital pin is to be used as a general-purpose I/O, or with a digital function that is not part of the crossbar, the bit associated with the pin in the PnSKIP register can be set to ‘1’ to ensure the crossbar does not attempt to assign a function to the pin.

26.4.3. Port Drive Strength

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

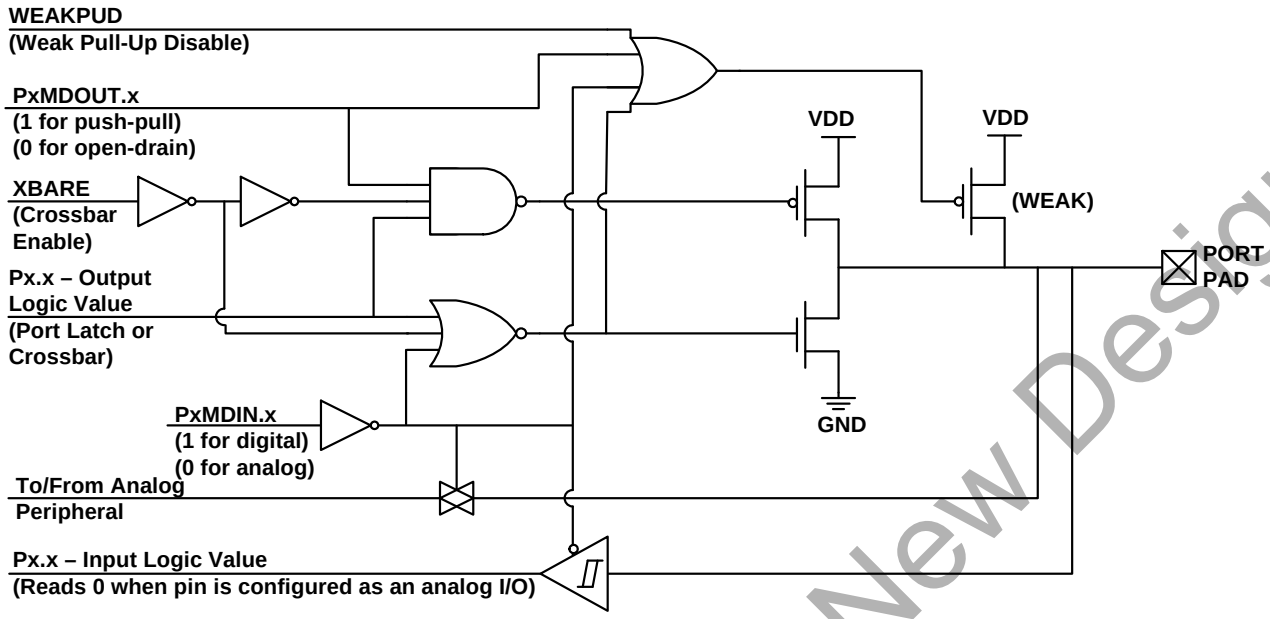


Figure 26.4. Port I/O Cell Block Diagram

26.5. Port Match

Port match functionality allows system events to be triggered by a logic value change on one or more port I/O pins. A software controlled value stored in the PnMATCH registers specifies the expected or normal logic values of the associated port pins (for example, P0MATCH.0 would correspond to P0.0). A port mismatch event occurs if the logic levels of the port's input pins no longer match the software controlled value. This allows software to be notified if a certain change or pattern occurs on the input pins regardless of the XBRn settings.

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

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

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

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

26.7. Port Configuration Registers

Register 26.1. XBR0: Port I/O Crossbar 0

Bit	7	6	5	4	3	2	1	0
Name	ECIE	PCA0ME			SYSCKE	SMB0E	SPI0E	URT0E
Type	RW	RW			RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x95								

Table 26.4. XBR0 Register Bit Descriptions

Bit	Name	Function
7	ECIE	**PCA0 External Counter Input Enable.** 0: ECI unavailable at Port pin. 1: ECI routed to Port pin.
6:4	PCA0ME	**PCA Module I/O Enable.** 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. 100-111: Reserved.
3	SYSCKE	**SYSCLK Output Enable.** 0: SYSCLK unavailable at Port pin. 1: SYSCLK output routed to Port pin.
2	SMB0E	**SMBus0 I/O Enable.** 0: SMBus0 I/O unavailable at Port pins. 1: SMBus0 I/O routed to Port pins.
1	SPI0E	**SPI I/O Enable.** 0: SPI I/O unavailable at Port pins. 1: SPI I/O routed to Port pins. The SPI can be assigned either 3 or 4 GPIO pins.
0	URT0E	**UART I/O Output Enable.** 0: UART I/O unavailable at Port pin. 1: UART TX, RX routed to Port pins P0.4 and P0.5.

Register 26.2. XBR1: Port I/O Crossbar 1

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	Reserved				T1E	T0E
Type	RW	RW	RW				RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x96								

Table 26.5. XBR1 Register Bit Descriptions

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5:2	Reserved	Must write reset value.
1	T1E	**T1 Enable.** 0: T1 unavailable at Port pin. 1: T1 routed to Port pin.
0	T0E	**T0 Enable.** 0: T0 unavailable at Port pin. 1: T0 routed to Port pin.

26.8. Port I/O Control Registers

Register 26.3. P0MASK: Port 0 Mask

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x8B								

Table 26.6. P0MASK Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Mask Value.** 0: P0.7 pin logic value is ignored and will not cause a port mismatch event. 1: P0.7 pin logic value is compared to P0MAT.7.
6	B6	**Port 0 Bit 6 Mask Value.** 0: P0.6 pin logic value is ignored and will not cause a port mismatch event. 1: P0.6 pin logic value is compared to P0MAT.6.
5	B5	**Port 0 Bit 5 Mask Value.** 0: P0.5 pin logic value is ignored and will not cause a port mismatch event. 1: P0.5 pin logic value is compared to P0MAT.5.
4	B4	**Port 0 Bit 4 Mask Value.** 0: P0.4 pin logic value is ignored and will not cause a port mismatch event. 1: P0.4 pin logic value is compared to P0MAT.4.
3	B3	**Port 0 Bit 3 Mask Value.** 0: P0.3 pin logic value is ignored and will not cause a port mismatch event. 1: P0.3 pin logic value is compared to P0MAT.3.
2	B2	**Port 0 Bit 2 Mask Value.** 0: P0.2 pin logic value is ignored and will not cause a port mismatch event. 1: P0.2 pin logic value is compared to P0MAT.2.
1	B1	**Port 0 Bit 1 Mask Value.** 0: P0.1 pin logic value is ignored and will not cause a port mismatch event. 1: P0.1 pin logic value is compared to P0MAT.1.
0	B0	**Port 0 Bit 0 Mask Value.** 0: P0.0 pin logic value is ignored and will not cause a port mismatch event. 1: P0.0 pin logic value is compared to P0MAT.0.

Register 26.4. P0MAT: Port 0 Match

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xF4								

Table 26.7. P0MAT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Match Value.** 0: P0.7 pin logic value is compared with logic LOW. 1: P0.7 pin logic value is compared with logic HIGH.
6	B6	**Port 0 Bit 6 Match Value.** 0: P0.6 pin logic value is compared with logic LOW. 1: P0.6 pin logic value is compared with logic HIGH.
5	B5	**Port 0 Bit 5 Match Value.** 0: P0.5 pin logic value is compared with logic LOW. 1: P0.5 pin logic value is compared with logic HIGH.
4	B4	**Port 0 Bit 4 Match Value.** 0: P0.4 pin logic value is compared with logic LOW. 1: P0.4 pin logic value is compared with logic HIGH.
3	B3	**Port 0 Bit 3 Match Value.** 0: P0.3 pin logic value is compared with logic LOW. 1: P0.3 pin logic value is compared with logic HIGH.
2	B2	**Port 0 Bit 2 Match Value.** 0: P0.2 pin logic value is compared with logic LOW. 1: P0.2 pin logic value is compared with logic HIGH.
1	B1	**Port 0 Bit 1 Match Value.** 0: P0.1 pin logic value is compared with logic LOW. 1: P0.1 pin logic value is compared with logic HIGH.
0	B0	**Port 0 Bit 0 Match Value.** 0: P0.0 pin logic value is compared with logic LOW. 1: P0.0 pin logic value is compared with logic HIGH.

Register 26.5. P0: Port 0 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = ALL; SFR Address: 0x80 (bit-addressable)								

Table 26.8. P0 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Latch.** 0: P0.7 is low. Set P0.7 to drive low. 1: P0.7 is high. Set P0.7 to drive or float high.
6	B6	**Port 0 Bit 6 Latch.** 0: P0.6 is low. Set P0.6 to drive low. 1: P0.6 is high. Set P0.6 to drive or float high.
5	B5	**Port 0 Bit 5 Latch.** 0: P0.5 is low. Set P0.5 to drive low. 1: P0.5 is high. Set P0.5 to drive or float high.
4	B4	**Port 0 Bit 4 Latch.** 0: P0.4 is low. Set P0.4 to drive low. 1: P0.4 is high. Set P0.4 to drive or float high.
3	B3	**Port 0 Bit 3 Latch.** 0: P0.3 is low. Set P0.3 to drive low. 1: P0.3 is high. Set P0.3 to drive or float high.
2	B2	**Port 0 Bit 2 Latch.** 0: P0.2 is low. Set P0.2 to drive low. 1: P0.2 is high. Set P0.2 to drive or float high.
1	B1	**Port 0 Bit 1 Latch.** 0: P0.1 is low. Set P0.1 to drive low. 1: P0.1 is high. Set P0.1 to drive or float high.
0	B0	**Port 0 Bit 0 Latch.** 0: P0.0 is low. Set P0.0 to drive low. 1: P0.0 is high. Set P0.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.6. P0MDIN: Port 0 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xEC								

Table 26.9. P0MDIN Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Input Mode.** 0: P0.7 pin is configured for analog mode. 1: P0.7 pin is configured for digital mode.
6	B6	**Port 0 Bit 6 Input Mode.** 0: P0.6 pin is configured for analog mode. 1: P0.6 pin is configured for digital mode.
5	B5	**Port 0 Bit 5 Input Mode.** 0: P0.5 pin is configured for analog mode. 1: P0.5 pin is configured for digital mode.
4	B4	**Port 0 Bit 4 Input Mode.** 0: P0.4 pin is configured for analog mode. 1: P0.4 pin is configured for digital mode.
3	B3	**Port 0 Bit 3 Input Mode.** 0: P0.3 pin is configured for analog mode. 1: P0.3 pin is configured for digital mode.
2	B2	**Port 0 Bit 2 Input Mode.** 0: P0.2 pin is configured for analog mode. 1: P0.2 pin is configured for digital mode.
1	B1	**Port 0 Bit 1 Input Mode.** 0: P0.1 pin is configured for analog mode. 1: P0.1 pin is configured for digital mode.
0	B0	**Port 0 Bit 0 Input Mode.** 0: P0.0 pin is configured for analog mode. 1: P0.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.7. P0MDOUT: Port 0 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xD9								

Table 26.10. P0MDOUT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Output Mode.** 0: P0.7 output is open-drain. 1: P0.7 output is push-pull.
6	B6	**Port 0 Bit 6 Output Mode.** 0: P0.6 output is open-drain. 1: P0.6 output is push-pull.
5	B5	**Port 0 Bit 5 Output Mode.** 0: P0.5 output is open-drain. 1: P0.5 output is push-pull.
4	B4	**Port 0 Bit 4 Output Mode.** 0: P0.4 output is open-drain. 1: P0.4 output is push-pull.
3	B3	**Port 0 Bit 3 Output Mode.** 0: P0.3 output is open-drain. 1: P0.3 output is push-pull.
2	B2	**Port 0 Bit 2 Output Mode.** 0: P0.2 output is open-drain. 1: P0.2 output is push-pull.
1	B1	**Port 0 Bit 1 Output Mode.** 0: P0.1 output is open-drain. 1: P0.1 output is push-pull.
0	B0	**Port 0 Bit 0 Output Mode.** 0: P0.0 output is open-drain. 1: P0.0 output is push-pull.

Register 26.8. P0SKIP: Port 0 Skip

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xB6								

Table 26.11. P0SKIP Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Skip.** 0: P0.7 pin is not skipped by the crossbar. 1: P0.7 pin is skipped by the crossbar.
6	B6	**Port 0 Bit 6 Skip.** 0: P0.6 pin is not skipped by the crossbar. 1: P0.6 pin is skipped by the crossbar.
5	B5	**Port 0 Bit 5 Skip.** 0: P0.5 pin is not skipped by the crossbar. 1: P0.5 pin is skipped by the crossbar.
4	B4	**Port 0 Bit 4 Skip.** 0: P0.4 pin is not skipped by the crossbar. 1: P0.4 pin is skipped by the crossbar.
3	B3	**Port 0 Bit 3 Skip.** 0: P0.3 pin is not skipped by the crossbar. 1: P0.3 pin is skipped by the crossbar.
2	B2	**Port 0 Bit 2 Skip.** 0: P0.2 pin is not skipped by the crossbar. 1: P0.2 pin is skipped by the crossbar.
1	B1	**Port 0 Bit 1 Skip.** 0: P0.1 pin is not skipped by the crossbar. 1: P0.1 pin is skipped by the crossbar.
0	B0	**Port 0 Bit 0 Skip.** 0: P0.0 pin is not skipped by the crossbar. 1: P0.0 pin is skipped by the crossbar.

Register 26.9. P0DRV: Port 0 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x99								

Table 26.12. P0DRV Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 0 Bit 7 Drive Strength.** 0: P0.7 output has low output drive strength. 1: P0.7 output has high output drive strength.
6	B6	**Port 0 Bit 6 Drive Strength.** 0: P0.6 output has low output drive strength. 1: P0.6 output has high output drive strength.
5	B5	**Port 0 Bit 5 Drive Strength.** 0: P0.5 output has low output drive strength. 1: P0.5 output has high output drive strength.
4	B4	**Port 0 Bit 4 Drive Strength.** 0: P0.4 output has low output drive strength. 1: P0.4 output has high output drive strength.
3	B3	**Port 0 Bit 3 Drive Strength.** 0: P0.3 output has low output drive strength. 1: P0.3 output has high output drive strength.
2	B2	**Port 0 Bit 2 Drive Strength.** 0: P0.2 output has low output drive strength. 1: P0.2 output has high output drive strength.
1	B1	**Port 0 Bit 1 Drive Strength.** 0: P0.1 output has low output drive strength. 1: P0.1 output has high output drive strength.
0	B0	**Port 0 Bit 0 Drive Strength.** 0: P0.0 output has low output drive strength. 1: P0.0 output has high output drive strength.

Register 26.10. P1MASK: Port 1 Mask

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x8C								

Table 26.13. P1MASK Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Mask Value.** 0: P1.7 pin logic value is ignored and will not cause a port mismatch event. 1: P1.7 pin logic value is compared to P1MAT.7.
6	B6	**Port 1 Bit 6 Mask Value.** 0: P1.6 pin logic value is ignored and will not cause a port mismatch event. 1: P1.6 pin logic value is compared to P1MAT.6.
5	B5	**Port 1 Bit 5 Mask Value.** 0: P1.5 pin logic value is ignored and will not cause a port mismatch event. 1: P1.5 pin logic value is compared to P1MAT.5.
4	B4	**Port 1 Bit 4 Mask Value.** 0: P1.4 pin logic value is ignored and will not cause a port mismatch event. 1: P1.4 pin logic value is compared to P1MAT.4.
3	B3	**Port 1 Bit 3 Mask Value.** 0: P1.3 pin logic value is ignored and will not cause a port mismatch event. 1: P1.3 pin logic value is compared to P1MAT.3.
2	B2	**Port 1 Bit 2 Mask Value.** 0: P1.2 pin logic value is ignored and will not cause a port mismatch event. 1: P1.2 pin logic value is compared to P1MAT.2.
1	B1	**Port 1 Bit 1 Mask Value.** 0: P1.1 pin logic value is ignored and will not cause a port mismatch event. 1: P1.1 pin logic value is compared to P1MAT.1.
0	B0	**Port 1 Bit 0 Mask Value.** 0: P1.0 pin logic value is ignored and will not cause a port mismatch event. 1: P1.0 pin logic value is compared to P1MAT.0.

Register 26.11. P1MAT: Port 1 Match

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xF5								

Table 26.14. P1MAT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Match Value.** 0: P1.7 pin logic value is compared with logic LOW. 1: P1.7 pin logic value is compared with logic HIGH.
6	B6	**Port 1 Bit 6 Match Value.** 0: P1.6 pin logic value is compared with logic LOW. 1: P1.6 pin logic value is compared with logic HIGH.
5	B5	**Port 1 Bit 5 Match Value.** 0: P1.5 pin logic value is compared with logic LOW. 1: P1.5 pin logic value is compared with logic HIGH.
4	B4	**Port 1 Bit 4 Match Value.** 0: P1.4 pin logic value is compared with logic LOW. 1: P1.4 pin logic value is compared with logic HIGH.
3	B3	**Port 1 Bit 3 Match Value.** 0: P1.3 pin logic value is compared with logic LOW. 1: P1.3 pin logic value is compared with logic HIGH.
2	B2	**Port 1 Bit 2 Match Value.** 0: P1.2 pin logic value is compared with logic LOW. 1: P1.2 pin logic value is compared with logic HIGH.
1	B1	**Port 1 Bit 1 Match Value.** 0: P1.1 pin logic value is compared with logic LOW. 1: P1.1 pin logic value is compared with logic HIGH.
0	B0	**Port 1 Bit 0 Match Value.** 0: P1.0 pin logic value is compared with logic LOW. 1: P1.0 pin logic value is compared with logic HIGH.

Register 26.12. P1: Port 1 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = ALL; SFR Address: 0x90 (bit-addressable)								

Table 26.15. P1 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Latch.** 0: P1.7 is low. Set P1.7 to drive low. 1: P1.7 is high. Set P1.7 to drive or float high.
6	B6	**Port 1 Bit 6 Latch.** 0: P1.6 is low. Set P1.6 to drive low. 1: P1.6 is high. Set P1.6 to drive or float high.
5	B5	**Port 1 Bit 5 Latch.** 0: P1.5 is low. Set P1.5 to drive low. 1: P1.5 is high. Set P1.5 to drive or float high.
4	B4	**Port 1 Bit 4 Latch.** 0: P1.4 is low. Set P1.4 to drive low. 1: P1.4 is high. Set P1.4 to drive or float high.
3	B3	**Port 1 Bit 3 Latch.** 0: P1.3 is low. Set P1.3 to drive low. 1: P1.3 is high. Set P1.3 to drive or float high.
2	B2	**Port 1 Bit 2 Latch.** 0: P1.2 is low. Set P1.2 to drive low. 1: P1.2 is high. Set P1.2 to drive or float high.
1	B1	**Port 1 Bit 1 Latch.** 0: P1.1 is low. Set P1.1 to drive low. 1: P1.1 is high. Set P1.1 to drive or float high.
0	B0	**Port 1 Bit 0 Latch.** 0: P1.0 is low. Set P1.0 to drive low. 1: P1.0 is high. Set P1.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.13. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xED								

Table 26.16. P1MDIN Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Input Mode.** 0: P1.7 pin is configured for analog mode. 1: P1.7 pin is configured for digital mode.
6	B6	**Port 1 Bit 6 Input Mode.** 0: P1.6 pin is configured for analog mode. 1: P1.6 pin is configured for digital mode.
5	B5	**Port 1 Bit 5 Input Mode.** 0: P1.5 pin is configured for analog mode. 1: P1.5 pin is configured for digital mode.
4	B4	**Port 1 Bit 4 Input Mode.** 0: P1.4 pin is configured for analog mode. 1: P1.4 pin is configured for digital mode.
3	B3	**Port 1 Bit 3 Input Mode.** 0: P1.3 pin is configured for analog mode. 1: P1.3 pin is configured for digital mode.
2	B2	**Port 1 Bit 2 Input Mode.** 0: P1.2 pin is configured for analog mode. 1: P1.2 pin is configured for digital mode.
1	B1	**Port 1 Bit 1 Input Mode.** 0: P1.1 pin is configured for analog mode. 1: P1.1 pin is configured for digital mode.
0	B0	**Port 1 Bit 0 Input Mode.** 0: P1.0 pin is configured for analog mode. 1: P1.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.14. P1MDOUT: Port 1 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xDC								

Table 26.17. P1MDOUT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Output Mode.** 0: P1.7 output is open-drain. 1: P1.7 output is push-pull.
6	B6	**Port 1 Bit 6 Output Mode.** 0: P1.6 output is open-drain. 1: P1.6 output is push-pull.
5	B5	**Port 1 Bit 5 Output Mode.** 0: P1.5 output is open-drain. 1: P1.5 output is push-pull.
4	B4	**Port 1 Bit 4 Output Mode.** 0: P1.4 output is open-drain. 1: P1.4 output is push-pull.
3	B3	**Port 1 Bit 3 Output Mode.** 0: P1.3 output is open-drain. 1: P1.3 output is push-pull.
2	B2	**Port 1 Bit 2 Output Mode.** 0: P1.2 output is open-drain. 1: P1.2 output is push-pull.
1	B1	**Port 1 Bit 1 Output Mode.** 0: P1.1 output is open-drain. 1: P1.1 output is push-pull.
0	B0	**Port 1 Bit 0 Output Mode.** 0: P1.0 output is open-drain. 1: P1.0 output is push-pull.

Register 26.15. P1SKIP: Port 1 Skip

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC6								

Table 26.18. P1SKIP Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Skip.** 0: P1.7 pin is not skipped by the crossbar. 1: P1.7 pin is skipped by the crossbar.
6	B6	**Port 1 Bit 6 Skip.** 0: P1.6 pin is not skipped by the crossbar. 1: P1.6 pin is skipped by the crossbar.
5	B5	**Port 1 Bit 5 Skip.** 0: P1.5 pin is not skipped by the crossbar. 1: P1.5 pin is skipped by the crossbar.
4	B4	**Port 1 Bit 4 Skip.** 0: P1.4 pin is not skipped by the crossbar. 1: P1.4 pin is skipped by the crossbar.
3	B3	**Port 1 Bit 3 Skip.** 0: P1.3 pin is not skipped by the crossbar. 1: P1.3 pin is skipped by the crossbar.
2	B2	**Port 1 Bit 2 Skip.** 0: P1.2 pin is not skipped by the crossbar. 1: P1.2 pin is skipped by the crossbar.
1	B1	**Port 1 Bit 1 Skip.** 0: P1.1 pin is not skipped by the crossbar. 1: P1.1 pin is skipped by the crossbar.
0	B0	**Port 1 Bit 0 Skip.** 0: P1.0 pin is not skipped by the crossbar. 1: P1.0 pin is skipped by the crossbar.

Register 26.16. P1DRV: Port 1 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x9A								

Table 26.19. P1DRV Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 1 Bit 7 Drive Strength.** 0: P1.7 output has low output drive strength. 1: P1.7 output has high output drive strength.
6	B6	**Port 1 Bit 6 Drive Strength.** 0: P1.6 output has low output drive strength. 1: P1.6 output has high output drive strength.
5	B5	**Port 1 Bit 5 Drive Strength.** 0: P1.5 output has low output drive strength. 1: P1.5 output has high output drive strength.
4	B4	**Port 1 Bit 4 Drive Strength.** 0: P1.4 output has low output drive strength. 1: P1.4 output has high output drive strength.
3	B3	**Port 1 Bit 3 Drive Strength.** 0: P1.3 output has low output drive strength. 1: P1.3 output has high output drive strength.
2	B2	**Port 1 Bit 2 Drive Strength.** 0: P1.2 output has low output drive strength. 1: P1.2 output has high output drive strength.
1	B1	**Port 1 Bit 1 Drive Strength.** 0: P1.1 output has low output drive strength. 1: P1.1 output has high output drive strength.
0	B0	**Port 1 Bit 0 Drive Strength.** 0: P1.0 output has low output drive strength. 1: P1.0 output has high output drive strength.

Register 26.17. P2MASK: Port 2 Mask

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x84								

Table 26.20. P2MASK Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Mask Value.** 0: P2.7 pin logic value is ignored and will not cause a port mismatch event. 1: P2.7 pin logic value is compared to P2MAT.7.
6	B6	**Port 2 Bit 6 Mask Value.** 0: P2.6 pin logic value is ignored and will not cause a port mismatch event. 1: P2.6 pin logic value is compared to P2MAT.6.
5	B5	**Port 2 Bit 5 Mask Value.** 0: P2.5 pin logic value is ignored and will not cause a port mismatch event. 1: P2.5 pin logic value is compared to P2MAT.5.
4	B4	**Port 2 Bit 4 Mask Value.** 0: P2.4 pin logic value is ignored and will not cause a port mismatch event. 1: P2.4 pin logic value is compared to P2MAT.4.
3	B3	**Port 2 Bit 3 Mask Value.** 0: P2.3 pin logic value is ignored and will not cause a port mismatch event. 1: P2.3 pin logic value is compared to P2MAT.3.
2	B2	**Port 2 Bit 2 Mask Value.** 0: P2.2 pin logic value is ignored and will not cause a port mismatch event. 1: P2.2 pin logic value is compared to P2MAT.2.
1	B1	**Port 2 Bit 1 Mask Value.** 0: P2.1 pin logic value is ignored and will not cause a port mismatch event. 1: P2.1 pin logic value is compared to P2MAT.1.
0	B0	**Port 2 Bit 0 Mask Value.** 0: P2.0 pin logic value is ignored and will not cause a port mismatch event. 1: P2.0 pin logic value is compared to P2MAT.0.

Register 26.18. P2MAT: Port 2 Match

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0x85								

Table 26.21. P2MAT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Match Value.** 0: P2.7 pin logic value is compared with logic LOW. 1: P2.7 pin logic value is compared with logic HIGH.
6	B6	**Port 2 Bit 6 Match Value.** 0: P2.6 pin logic value is compared with logic LOW. 1: P2.6 pin logic value is compared with logic HIGH.
5	B5	**Port 2 Bit 5 Match Value.** 0: P2.5 pin logic value is compared with logic LOW. 1: P2.5 pin logic value is compared with logic HIGH.
4	B4	**Port 2 Bit 4 Match Value.** 0: P2.4 pin logic value is compared with logic LOW. 1: P2.4 pin logic value is compared with logic HIGH.
3	B3	**Port 2 Bit 3 Match Value.** 0: P2.3 pin logic value is compared with logic LOW. 1: P2.3 pin logic value is compared with logic HIGH.
2	B2	**Port 2 Bit 2 Match Value.** 0: P2.2 pin logic value is compared with logic LOW. 1: P2.2 pin logic value is compared with logic HIGH.
1	B1	**Port 2 Bit 1 Match Value.** 0: P2.1 pin logic value is compared with logic LOW. 1: P2.1 pin logic value is compared with logic HIGH.
0	B0	**Port 2 Bit 0 Match Value.** 0: P2.0 pin logic value is compared with logic LOW. 1: P2.0 pin logic value is compared with logic HIGH.

Register 26.19. P2: Port 2 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = ALL; SFR Address: 0xA0 (bit-addressable)								

Table 26.22. P2 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Latch.** 0: P2.7 is low. Set P2.7 to drive low. 1: P2.7 is high. Set P2.7 to drive or float high.
6	B6	**Port 2 Bit 6 Latch.** 0: P2.6 is low. Set P2.6 to drive low. 1: P2.6 is high. Set P2.6 to drive or float high.
5	B5	**Port 2 Bit 5 Latch.** 0: P2.5 is low. Set P2.5 to drive low. 1: P2.5 is high. Set P2.5 to drive or float high.
4	B4	**Port 2 Bit 4 Latch.** 0: P2.4 is low. Set P2.4 to drive low. 1: P2.4 is high. Set P2.4 to drive or float high.
3	B3	**Port 2 Bit 3 Latch.** 0: P2.3 is low. Set P2.3 to drive low. 1: P2.3 is high. Set P2.3 to drive or float high.
2	B2	**Port 2 Bit 2 Latch.** 0: P2.2 is low. Set P2.2 to drive low. 1: P2.2 is high. Set P2.2 to drive or float high.
1	B1	**Port 2 Bit 1 Latch.** 0: P2.1 is low. Set P2.1 to drive low. 1: P2.1 is high. Set P2.1 to drive or float high.
0	B0	**Port 2 Bit 0 Latch.** 0: P2.0 is low. Set P2.0 to drive low. 1: P2.0 is high. Set P2.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.20. P2MDIN: Port 2 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xEE								

Table 26.23. P2MDIN Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Input Mode.** 0: P2.7 pin is configured for analog mode. 1: P2.7 pin is configured for digital mode.
6	B6	**Port 2 Bit 6 Input Mode.** 0: P2.6 pin is configured for analog mode. 1: P2.6 pin is configured for digital mode.
5	B5	**Port 2 Bit 5 Input Mode.** 0: P2.5 pin is configured for analog mode. 1: P2.5 pin is configured for digital mode.
4	B4	**Port 2 Bit 4 Input Mode.** 0: P2.4 pin is configured for analog mode. 1: P2.4 pin is configured for digital mode.
3	B3	**Port 2 Bit 3 Input Mode.** 0: P2.3 pin is configured for analog mode. 1: P2.3 pin is configured for digital mode.
2	B2	**Port 2 Bit 2 Input Mode.** 0: P2.2 pin is configured for analog mode. 1: P2.2 pin is configured for digital mode.
1	B1	**Port 2 Bit 1 Input Mode.** 0: P2.1 pin is configured for analog mode. 1: P2.1 pin is configured for digital mode.
0	B0	**Port 2 Bit 0 Input Mode.** 0: P2.0 pin is configured for analog mode. 1: P2.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.21. P2MDOUT: Port 2 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xDD								

Table 26.24. P2MDOUT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Output Mode.** 0: P2.7 output is open-drain. 1: P2.7 output is push-pull.
6	B6	**Port 2 Bit 6 Output Mode.** 0: P2.6 output is open-drain. 1: P2.6 output is push-pull.
5	B5	**Port 2 Bit 5 Output Mode.** 0: P2.5 output is open-drain. 1: P2.5 output is push-pull.
4	B4	**Port 2 Bit 4 Output Mode.** 0: P2.4 output is open-drain. 1: P2.4 output is push-pull.
3	B3	**Port 2 Bit 3 Output Mode.** 0: P2.3 output is open-drain. 1: P2.3 output is push-pull.
2	B2	**Port 2 Bit 2 Output Mode.** 0: P2.2 output is open-drain. 1: P2.2 output is push-pull.
1	B1	**Port 2 Bit 1 Output Mode.** 0: P2.1 output is open-drain. 1: P2.1 output is push-pull.
0	B0	**Port 2 Bit 0 Output Mode.** 0: P2.0 output is open-drain. 1: P2.0 output is push-pull.

Register 26.22. P2SKIP: Port 2 Skip

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC7								

Table 26.25. P2SKIP Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Skip.** 0: P2.7 pin is not skipped by the crossbar. 1: P2.7 pin is skipped by the crossbar.
6	B6	**Port 2 Bit 6 Skip.** 0: P2.6 pin is not skipped by the crossbar. 1: P2.6 pin is skipped by the crossbar.
5	B5	**Port 2 Bit 5 Skip.** 0: P2.5 pin is not skipped by the crossbar. 1: P2.5 pin is skipped by the crossbar.
4	B4	**Port 2 Bit 4 Skip.** 0: P2.4 pin is not skipped by the crossbar. 1: P2.4 pin is skipped by the crossbar.
3	B3	**Port 2 Bit 3 Skip.** 0: P2.3 pin is not skipped by the crossbar. 1: P2.3 pin is skipped by the crossbar.
2	B2	**Port 2 Bit 2 Skip.** 0: P2.2 pin is not skipped by the crossbar. 1: P2.2 pin is skipped by the crossbar.
1	B1	**Port 2 Bit 1 Skip.** 0: P2.1 pin is not skipped by the crossbar. 1: P2.1 pin is skipped by the crossbar.
0	B0	**Port 2 Bit 0 Skip.** 0: P2.0 pin is not skipped by the crossbar. 1: P2.0 pin is skipped by the crossbar.

Register 26.23. P2DRV: Port 2 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x9B								

Table 26.26. P2DRV Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 2 Bit 7 Drive Strength.** 0: P2.7 output has low output drive strength. 1: P2.7 output has high output drive strength.
6	B6	**Port 2 Bit 6 Drive Strength.** 0: P2.6 output has low output drive strength. 1: P2.6 output has high output drive strength.
5	B5	**Port 2 Bit 5 Drive Strength.** 0: P2.5 output has low output drive strength. 1: P2.5 output has high output drive strength.
4	B4	**Port 2 Bit 4 Drive Strength.** 0: P2.4 output has low output drive strength. 1: P2.4 output has high output drive strength.
3	B3	**Port 2 Bit 3 Drive Strength.** 0: P2.3 output has low output drive strength. 1: P2.3 output has high output drive strength.
2	B2	**Port 2 Bit 2 Drive Strength.** 0: P2.2 output has low output drive strength. 1: P2.2 output has high output drive strength.
1	B1	**Port 2 Bit 1 Drive Strength.** 0: P2.1 output has low output drive strength. 1: P2.1 output has high output drive strength.
0	B0	**Port 2 Bit 0 Drive Strength.** 0: P2.0 output has low output drive strength. 1: P2.0 output has high output drive strength.

Register 26.24. P3: Port 3 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0x0; SFR Address: 0xE1								

Table 26.27. P3 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 3 Bit 7 Latch.** 0: P3.7 is low. Set P3.7 to drive low. 1: P3.7 is high. Set P3.7 to drive or float high.
6	B6	**Port 3 Bit 6 Latch.** 0: P3.6 is low. Set P3.6 to drive low. 1: P3.6 is high. Set P3.6 to drive or float high.
5	B5	**Port 3 Bit 5 Latch.** 0: P3.5 is low. Set P3.5 to drive low. 1: P3.5 is high. Set P3.5 to drive or float high.
4	B4	**Port 3 Bit 4 Latch.** 0: P3.4 is low. Set P3.4 to drive low. 1: P3.4 is high. Set P3.4 to drive or float high.
3	B3	**Port 3 Bit 3 Latch.** 0: P3.3 is low. Set P3.3 to drive low. 1: P3.3 is high. Set P3.3 to drive or float high.
2	B2	**Port 3 Bit 2 Latch.** 0: P3.2 is low. Set P3.2 to drive low. 1: P3.2 is high. Set P3.2 to drive or float high.
1	B1	**Port 3 Bit 1 Latch.** 0: P3.1 is low. Set P3.1 to drive low. 1: P3.1 is high. Set P3.1 to drive or float high.
0	B0	**Port 3 Bit 0 Latch.** 0: P3.0 is low. Set P3.0 to drive low. 1: P3.0 is high. Set P3.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.25. P3MDIN: Port 3 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xEF								

Table 26.28. P3MDIN Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 3 Bit 7 Input Mode.** 0: P3.7 pin is configured for analog mode. 1: P3.7 pin is configured for digital mode.
6	B6	**Port 3 Bit 6 Input Mode.** 0: P3.6 pin is configured for analog mode. 1: P3.6 pin is configured for digital mode.
5	B5	**Port 3 Bit 5 Input Mode.** 0: P3.5 pin is configured for analog mode. 1: P3.5 pin is configured for digital mode.
4	B4	**Port 3 Bit 4 Input Mode.** 0: P3.4 pin is configured for analog mode. 1: P3.4 pin is configured for digital mode.
3	B3	**Port 3 Bit 3 Input Mode.** 0: P3.3 pin is configured for analog mode. 1: P3.3 pin is configured for digital mode.
2	B2	**Port 3 Bit 2 Input Mode.** 0: P3.2 pin is configured for analog mode. 1: P3.2 pin is configured for digital mode.
1	B1	**Port 3 Bit 1 Input Mode.** 0: P3.1 pin is configured for analog mode. 1: P3.1 pin is configured for digital mode.
0	B0	**Port 3 Bit 0 Input Mode.** 0: P3.0 pin is configured for analog mode. 1: P3.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.26. P3MDOUT: Port 3 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xDF								

Table 26.29. P3MDOUT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 3 Bit 7 Output Mode.** 0: P3.7 output is open-drain. 1: P3.7 output is push-pull.
6	B6	**Port 3 Bit 6 Output Mode.** 0: P3.6 output is open-drain. 1: P3.6 output is push-pull.
5	B5	**Port 3 Bit 5 Output Mode.** 0: P3.5 output is open-drain. 1: P3.5 output is push-pull.
4	B4	**Port 3 Bit 4 Output Mode.** 0: P3.4 output is open-drain. 1: P3.4 output is push-pull.
3	B3	**Port 3 Bit 3 Output Mode.** 0: P3.3 output is open-drain. 1: P3.3 output is push-pull.
2	B2	**Port 3 Bit 2 Output Mode.** 0: P3.2 output is open-drain. 1: P3.2 output is push-pull.
1	B1	**Port 3 Bit 1 Output Mode.** 0: P3.1 output is open-drain. 1: P3.1 output is push-pull.
0	B0	**Port 3 Bit 0 Output Mode.** 0: P3.0 output is open-drain. 1: P3.0 output is push-pull.

Register 26.27. P3DRV: Port 3 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x9C								

Table 26.30. P3DRV Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 3 Bit 7 Drive Strength.** 0: P3.7 output has low output drive strength. 1: P3.7 output has high output drive strength.
6	B6	**Port 3 Bit 6 Drive Strength.** 0: P3.6 output has low output drive strength. 1: P3.6 output has high output drive strength.
5	B5	**Port 3 Bit 5 Drive Strength.** 0: P3.5 output has low output drive strength. 1: P3.5 output has high output drive strength.
4	B4	**Port 3 Bit 4 Drive Strength.** 0: P3.4 output has low output drive strength. 1: P3.4 output has high output drive strength.
3	B3	**Port 3 Bit 3 Drive Strength.** 0: P3.3 output has low output drive strength. 1: P3.3 output has high output drive strength.
2	B2	**Port 3 Bit 2 Drive Strength.** 0: P3.2 output has low output drive strength. 1: P3.2 output has high output drive strength.
1	B1	**Port 3 Bit 1 Drive Strength.** 0: P3.1 output has low output drive strength. 1: P3.1 output has high output drive strength.
0	B0	**Port 3 Bit 0 Drive Strength.** 0: P3.0 output has low output drive strength. 1: P3.0 output has high output drive strength.

Register 26.28. P4: Port 4 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0x0; SFR Address: 0xE2								

Table 26.31. P4 Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Latch.** 0: P4.7 is low. Set P4.7 to drive low. 1: P4.7 is high. Set P4.7 to drive or float high.
6	B6	**Port 4 Bit 6 Latch.** 0: P4.6 is low. Set P4.6 to drive low. 1: P4.6 is high. Set P4.6 to drive or float high.
5	B5	**Port 4 Bit 5 Latch.** 0: P4.5 is low. Set P4.5 to drive low. 1: P4.5 is high. Set P4.5 to drive or float high.
4	B4	**Port 4 Bit 4 Latch.** 0: P4.4 is low. Set P4.4 to drive low. 1: P4.4 is high. Set P4.4 to drive or float high.
3	B3	**Port 4 Bit 3 Latch.** 0: P4.3 is low. Set P4.3 to drive low. 1: P4.3 is high. Set P4.3 to drive or float high.
2	B2	**Port 4 Bit 2 Latch.** 0: P4.2 is low. Set P4.2 to drive low. 1: P4.2 is high. Set P4.2 to drive or float high.
1	B1	**Port 4 Bit 1 Latch.** 0: P4.1 is low. Set P4.1 to drive low. 1: P4.1 is high. Set P4.1 to drive or float high.
0	B0	**Port 4 Bit 0 Latch.** 0: P4.0 is low. Set P4.0 to drive low. 1: P4.0 is high. Set P4.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.29. P4MDIN: Port 4 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
SFR Page = 0xF; SFR Address: 0xF1								

Table 26.32. P4MDIN Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Input Mode.** 0: P4.7 pin is configured for analog mode. 1: P4.7 pin is configured for digital mode.
6	B6	**Port 4 Bit 6 Input Mode.** 0: P4.6 pin is configured for analog mode. 1: P4.6 pin is configured for digital mode.
5	B5	**Port 4 Bit 5 Input Mode.** 0: P4.5 pin is configured for analog mode. 1: P4.5 pin is configured for digital mode.
4	B4	**Port 4 Bit 4 Input Mode.** 0: P4.4 pin is configured for analog mode. 1: P4.4 pin is configured for digital mode.
3	B3	**Port 4 Bit 3 Input Mode.** 0: P4.3 pin is configured for analog mode. 1: P4.3 pin is configured for digital mode.
2	B2	**Port 4 Bit 2 Input Mode.** 0: P4.2 pin is configured for analog mode. 1: P4.2 pin is configured for digital mode.
1	B1	**Port 4 Bit 1 Input Mode.** 0: P4.1 pin is configured for analog mode. 1: P4.1 pin is configured for digital mode.
0	B0	**Port 4 Bit 0 Input Mode.** 0: P4.0 pin is configured for analog mode. 1: P4.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.30. P4MDOUT: Port 4 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xC3								

Table 26.33. P4MDOUT Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Output Mode.** 0: P4.7 output is open-drain. 1: P4.7 output is push-pull.
6	B6	**Port 4 Bit 6 Output Mode.** 0: P4.6 output is open-drain. 1: P4.6 output is push-pull.
5	B5	**Port 4 Bit 5 Output Mode.** 0: P4.5 output is open-drain. 1: P4.5 output is push-pull.
4	B4	**Port 4 Bit 4 Output Mode.** 0: P4.4 output is open-drain. 1: P4.4 output is push-pull.
3	B3	**Port 4 Bit 3 Output Mode.** 0: P4.3 output is open-drain. 1: P4.3 output is push-pull.
2	B2	**Port 4 Bit 2 Output Mode.** 0: P4.2 output is open-drain. 1: P4.2 output is push-pull.
1	B1	**Port 4 Bit 1 Output Mode.** 0: P4.1 output is open-drain. 1: P4.1 output is push-pull.
0	B0	**Port 4 Bit 0 Output Mode.** 0: P4.0 output is open-drain. 1: P4.0 output is push-pull.

Register 26.31. P4DRV: Port 4 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xB9								

Table 26.34. P4DRV Register Bit Descriptions

Bit	Name	Function
7	B7	**Port 4 Bit 7 Drive Strength.** 0: P4.7 output has low output drive strength. 1: P4.7 output has high output drive strength.
6	B6	**Port 4 Bit 6 Drive Strength.** 0: P4.6 output has low output drive strength. 1: P4.6 output has high output drive strength.
5	B5	**Port 4 Bit 5 Drive Strength.** 0: P4.5 output has low output drive strength. 1: P4.5 output has high output drive strength.
4	B4	**Port 4 Bit 4 Drive Strength.** 0: P4.4 output has low output drive strength. 1: P4.4 output has high output drive strength.
3	B3	**Port 4 Bit 3 Drive Strength.** 0: P4.3 output has low output drive strength. 1: P4.3 output has high output drive strength.
2	B2	**Port 4 Bit 2 Drive Strength.** 0: P4.2 output has low output drive strength. 1: P4.2 output has high output drive strength.
1	B1	**Port 4 Bit 1 Drive Strength.** 0: P4.1 output has low output drive strength. 1: P4.1 output has high output drive strength.
0	B0	**Port 4 Bit 0 Drive Strength.** 0: P4.0 output has low output drive strength. 1: P4.0 output has high output drive strength.

Register 26.32. P5: Port 5 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	Reserved					B2	B1	B0
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	1	1	1
SFR Page = 0x0; SFR Address: 0xE3								

Table 26.35. P5 Register Bit Descriptions

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	B2	**Port 5 Bit 2 Latch.** 0: P5.2 is low. Set P5.2 to drive low. 1: P5.2 is high. Set P5.2 to drive or float high.
1	B1	**Port 5 Bit 1 Latch.** 0: P5.1 is low. Set P5.1 to drive low. 1: P5.1 is high. Set P5.1 to drive or float high.
0	B0	**Port 5 Bit 0 Latch.** 0: P5.0 is low. Set P5.0 to drive low. 1: P5.0 is high. Set P5.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.33. P5MDIN: Port 5 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved					B2	B1	B0
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	1	1	1
SFR Page = 0xF; SFR Address: 0xF2								

Table 26.36. P5MDIN Register Bit Descriptions

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	B2	**Port 5 Bit 2 Input Mode.** 0: P5.2 pin is configured for analog mode. 1: P5.2 pin is configured for digital mode.
1	B1	**Port 5 Bit 1 Input Mode.** 0: P5.1 pin is configured for analog mode. 1: P5.1 pin is configured for digital mode.
0	B0	**Port 5 Bit 0 Input Mode.** 0: P5.0 pin is configured for analog mode. 1: P5.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

Register 26.34. P5MDOUT: Port 5 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved					B2	B1	B0
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xFF								

Table 26.37. P5MDOUT Register Bit Descriptions

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	B2	**Port 5 Bit 2 Output Mode.** 0: P5.2 output is open-drain. 1: P5.2 output is push-pull.
1	B1	**Port 5 Bit 1 Output Mode.** 0: P5.1 output is open-drain. 1: P5.1 output is push-pull.
0	B0	**Port 5 Bit 0 Output Mode.** 0: P5.0 output is open-drain. 1: P5.0 output is push-pull.

Register 26.35. P5DRV: Port 5 Drive Strength

Bit	7	6	5	4	3	2	1	0
Name	Reserved					B2	B1	B0
Type	RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0x9D								

Table 26.38. P5DRV Register Bit Descriptions

Bit	Name	Function
7:3	Reserved	Must write reset value.
2	B2	**Port 5 Bit 2 Drive Strength.** 0: P5.2 output has low output drive strength. 1: P5.2 output has high output drive strength.
1	B1	**Port 5 Bit 1 Drive Strength.** 0: P5.1 output has low output drive strength. 1: P5.1 output has high output drive strength.
0	B0	**Port 5 Bit 0 Drive Strength.** 0: P5.0 output has low output drive strength. 1: P5.0 output has high output drive strength.

Register 26.36. P6: Port 6 Pin Latch

Bit	7	6	5	4	3	2	1	0
Name	Reserved						B1	B0
Type	RW						RW	RW
Reset	0	0	0	0	0	0	1	1
SFR Page = 0x0; SFR Address: 0xE4								

Table 26.39. P6 Register Bit Descriptions

Bit	Name	Function
7:2	Reserved	Must write reset value.
1	B1	**Port 6 Bit 1 Latch.** 0: P6.1 is low. Set P6.1 to drive low. 1: P6.1 is high. Set P6.1 to drive or float high.
0	B0	**Port 6 Bit 0 Latch.** 0: P6.0 is low. Set P6.0 to drive low. 1: P6.0 is high. Set P6.0 to drive or float high.

Notes:
1. Writing this register sets the port latch logic value for the associated I/O pins configured as digital I/O.
2. Reading this register returns the logic value at the pin, regardless if it is configured as output or input.

Register 26.37. P6MDIN: Port 6 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	Reserved						B1	B0
Type	RW						RW	RW
Reset	0	0	0	0	0	0	1	1
SFR Page = 0xF; SFR Address: 0x97								

Table 26.40. P6MDIN Register Bit Descriptions

Bit	Name	Function
7:2	Reserved	Must write reset value.
1	B1	**Port 6 Bit 1 Input Mode.** 0: P6.1 pin is configured for analog mode. 1: P6.1 pin is configured for digital mode.

Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.

Table 26.40. P6MDIN Register Bit Descriptions

Bit	Name	Function
0	B0	**Port 6 Bit 0 Input Mode.** 0: P6.0 pin is configured for analog mode. 1: P6.0 pin is configured for digital mode.
Note: Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled.		

27. Reset Sources and Supply Monitor

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

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

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

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

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

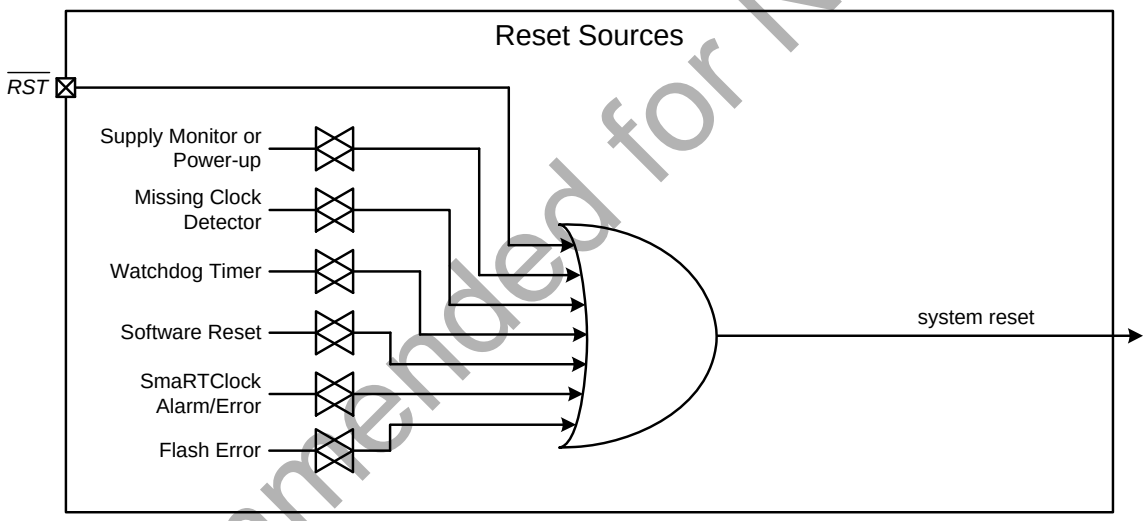


Figure 27.1. Reset Sources

27.1. Power-On Reset

During power-up, the POR circuit will fire. When POR fires, the device is held in a reset state and the $\overline{RST}$ pin is driven low until V_{DD} settles above V_{RST}. Two delays are present during the supply ramp time. First, a delay will occur before the POR circuitry fires and pulls the $\overline{RST}$ pin low. A second delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 27.2. plots the power-on reset timing. For ramp times less than 1 ms, the power-on reset time (T_{POR}) is typically less than 0.3 ms. Additionally, the power supply must reach V_{RST} before the POR circuit will release the device from reset.

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

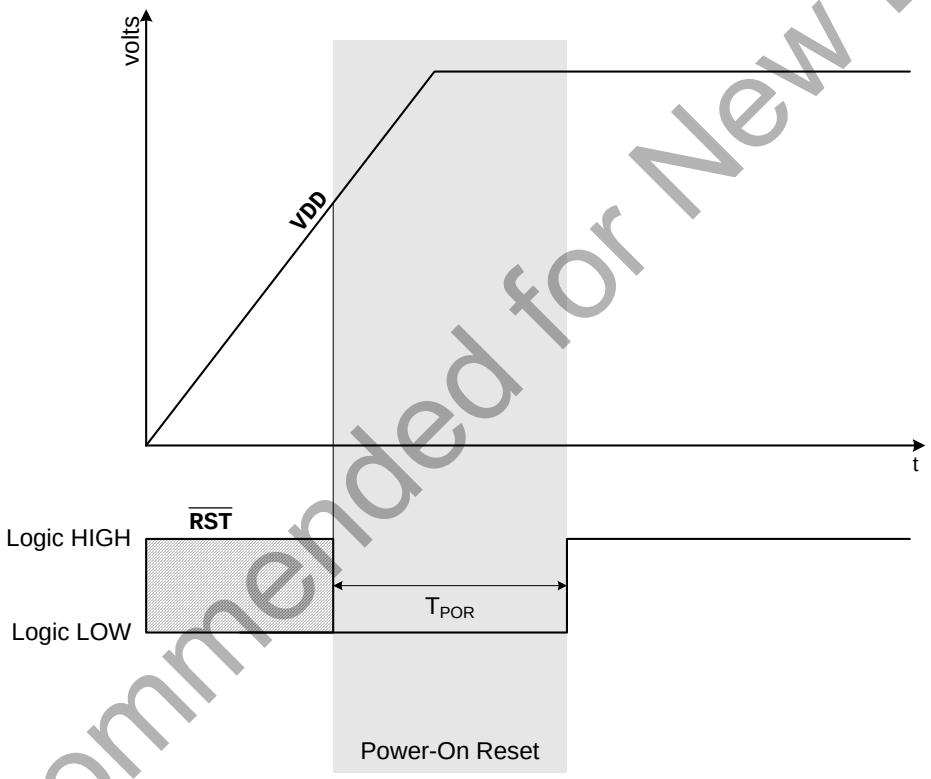


Figure 27.2. Power-on Reset Timing

27.2. Power-Fail Reset / Supply Monitor

C8051F97x devices have a supply monitor that is enabled and selected as a reset source after each power-on or power fail reset.

When enabled and selected as a reset source, any power down transition or power irregularity that causes V_{DD} to drop below V_{RST} will cause the $\overline{RST}$ pin to be driven low and the CIP-51 will be held in a reset state (see Figure 27.3). When V_{DD} returns to a level above V_{RST}, the CIP-51 will be released from the reset state.

After a power-fail reset, the PORSF flag reads 1, the contents of RAM invalid, and the V_{DD} supply monitor is enabled and selected as a reset source. The enable state of the V_{DD} supply monitor and its selection as a reset source is only altered by power-on and power-fail resets. For example, if the V_{DD} supply monitor is deselected as a reset source and disabled by software, then a software reset is performed, the V_{DD} supply monitor will remain disabled and deselected after the reset.

In battery-operated systems, the contents of RAM can be preserved near the end of the battery's usable life if the device is placed in Sleep Mode prior to a power-fail reset occurring. When the device is in Sleep Mode, the power-fail reset is automatically disabled and the contents of RAM are preserved as long as V_{DD} does not fall below V_{POR}. A large capacitor can be used to hold the power supply voltage above V_{POR} while the user is replacing the battery. Upon waking from Sleep mode, the enable and reset source select state of the V_{DD} supply monitor are restored to the value last set by the user.

To allow software early notification that a power failure is about to occur, the VDDOK bit is cleared when the V_{DD} supply falls below the V_{WARN} threshold. The VDDOK bit can be configured to generate an interrupt. See Section "13. Interrupts" on page 79 for more details.

Important Note: To protect the integrity of Flash contents, **the V_{DD} supply monitor must be enabled and selected as a reset source if software contains routines which erase or write Flash memory.** If the V_{DD} supply monitor is not enabled, any erase or write performed on Flash memory will cause a Flash Error device reset. memory. If the VDD supply monitor is not enabled, any erase or write performed on flash memory will be ignored.

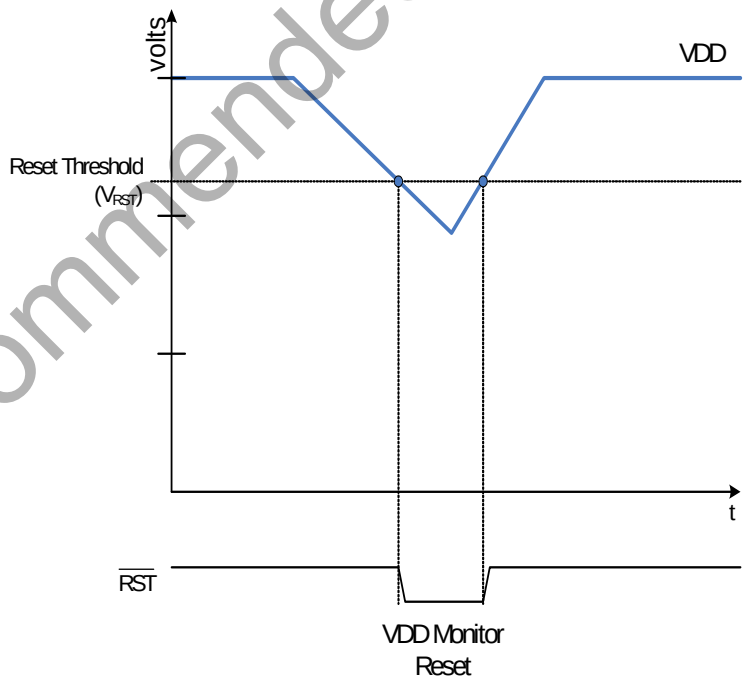


Figure 27.3. VDD Supply Monitor Threshold

27.3. Enabling the VDD Monitor

The VDD supply monitor is enabled by default. However, in systems which disable the supply monitor, it must be enabled before selecting it as a reset source. Selecting the VDD supply monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the VDD supply monitor and selecting it as a reset source. No delay should be introduced in systems where software contains routines that erase or write flash memory. The procedure for enabling the VDD supply monitor and selecting it as a reset source is:

1. Enable the VDD supply monitor (VMONEN = 1).
2. Wait for the VDD supply monitor to stabilize (optional).
3. Enable the VDD monitor as a reset source in the RSTSRC register.

27.4. External Reset

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

27.5. Missing Clock Detector Reset

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

27.6. 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 the PCA watchdog timer section. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit is set to '1'. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

27.7. 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.
- A flash read is attempted above user code space.
- A program read is attempted above user code space (i.e. a branch instruction to the reserved area).
- A flash read, write or erase attempt is restricted due to a flash security setting.

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

27.8. SmaRTClock Reset

The SmaRTClock can generate a system reset on two events: SmaRTClock Oscillator Fail or SmaRTClock Alarm. The SmaRTClock Oscillator Fail event occurs when the SmaRTClock Missing Clock Detector is enabled and the SmaRTClock clock is below approximately 20 kHz. A SmaRTClock alarm event occurs when the SmaRTClock Alarm is enabled and the SmaRTClock timer value matches the ALARMn registers. The SmaRTClock can be configured as a reset source by writing a 1 to the RTC0RE flag (RSTSRC.7). The SmaRTClock reset remains functional even when the device is in the low power Suspend or Sleep mode. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

27.9. Software Reset

Software may force a reset by writing a 1 to the SWRSF bit. The SWRSF bit will read 1 following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

27.10. Reset Sources Control Registers

Register 27.1. RSTSRC: Reset Source

Bit	7	6	5	4	3	2	1	0
Name	RTC0RE	FERROR	Reserved	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	X	X	X	X	X	X	X	X
SFR Page = 0x0; SFR Address: 0xEF								

Table 27.1. RSTSRC Register Bit Descriptions

Bit	Name	Function
7	RTC0RE	**RTC Reset Enable and Flag.** Read: This bit reads 1 if a RTC alarm or oscillator fail caused the last reset. Write: Writing a 1 to this bit enables the RTC as a reset source.
6	FERROR	**Flash Error Reset Flag.** This read-only bit is set to '1' if a flash read/write/erase error caused the last reset.
5	Reserved	Must write reset value.
4	SWRSF	**Software Reset Force and Flag.** Read: This bit reads 1 if last reset was caused by a write to SWRSF. Write: Writing a 1 to this bit forces a system reset.
3	WDTRSF	**Watchdog Timer Reset Flag.** This read-only bit is set to '1' if a watchdog timer overflow caused the last reset.
2	MCDRSF	**Missing Clock Detector Enable and Flag.** Read: This bit reads 1 if a missing clock detector timeout caused the last reset. Write: Writing a 1 to this bit enables the missing clock detector. The MCD triggers a reset if a missing clock condition is detected.
1	PORSF	**Power-On / Supply Monitor Reset Flag, and Supply Monitor Reset Enable.** Read: This bit reads 1 anytime a power-on or supply monitor reset has occurred. Write: Writing a 1 to this bit enables the supply monitor as a reset source.
0	PINRSF	**HW Pin Reset Flag.** This read-only bit is set to '1' if the RST pin caused the last reset.

Notes:
1. Reads and writes of the RSTSRC register access different logic in the device. Reading the register always returns status information to indicate the source of the most recent reset. Writing to the register activates certain options as reset sources. It is recommended to not use any kind of read-modify-write operation on this register.
2. When the PORSF bit reads back '1' all other RSTSRC flags are indeterminate.
3. Writing '1' to the PORSF bit when the supply monitor is not enabled and stabilized may cause a system reset.

27.11. Supply Monitor Control Registers

Register 27.2. VDM0CN: VDD Supply Monitor Control

Bit	7	6	5	4	3	2	1	0
Name	VDMEN	VDDSTAT	VDDOK	Reserved	VDDOKIE	Reserved		
Type	RW	R	R	R	RW	R		
Reset	1	X	X	0	1	0	0	0
SFR Page = 0x0; SFR Address: 0xFF								

Table 27.2. VDM0CN Register Bit Descriptions

Bit	Name	Function
7	VDMEN	**V_{DD} Supply Monitor Enable.** This bit turns the V_{DD} supply monitor circuit on/off. The V_{DD} Supply Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC. 0: Disable the V_{DD} supply monitor. 1: Enable the V_{DD} supply monitor.
6	VDDSTAT	**V_{DD} Supply Status.** This bit indicates the current power supply status. 0: V_{DD} is at or below the VRST threshold. 1: V_{DD} is above the VRST threshold.
5	VDDOK	**V_{DD} Supply Status (Early Warning).** This bit indicates the current VDD power supply status. 0: V_{DD} is at or below the VDDWARN threshold. 1: V_{DD} is above the VDDWARN threshold.
4	Reserved	Must write reset value.
3	VDDOKIE	**V_{DD} Early Warning Interrupt Enable.** Enables the V_{DD} Early Warning interrupt. 0: Disable the V_{DD} Early Warning interrupt. 1: Enable the V_{DD} Early Warning interrupt.
2:0	Reserved	Must write reset value.

28. Serial Peripheral Interface (SPI0)

The 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.

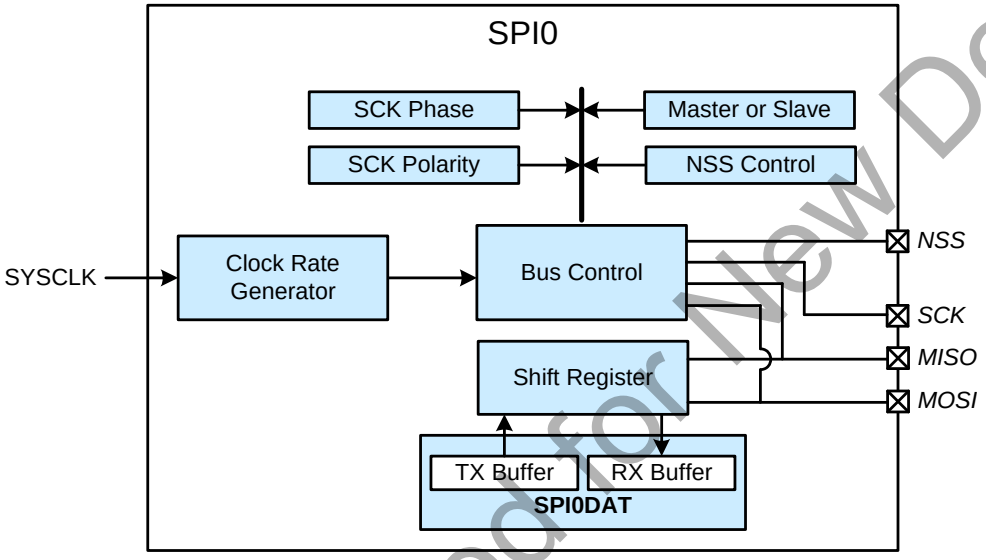


Figure 28.1. SPI0 Block Diagram

28.1. Signal Descriptions

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

28.1.1. 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.

28.1.2. 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.

28.1.3. 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.

28.1.4. 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 28.2, Figure 28.3, and Figure 28.4 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.

28.2. 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 28.2 shows a connection diagram between two master devices and a single slave 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 28.3 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 28.4 shows a connection diagram for a master device and a slave device in 4-wire mode.

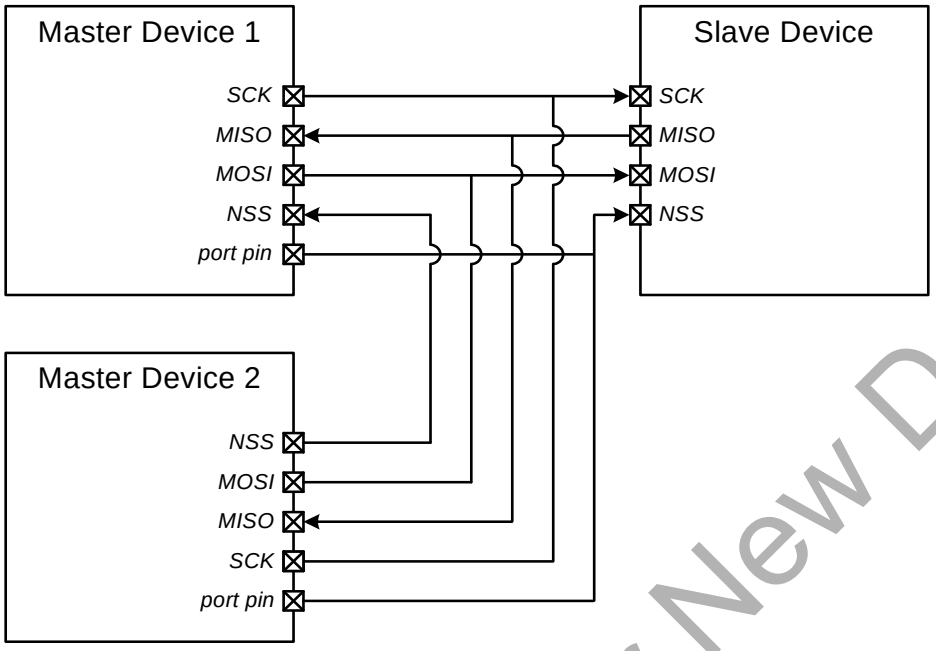


Figure 28.2. Multiple-Master Mode Connection Diagram

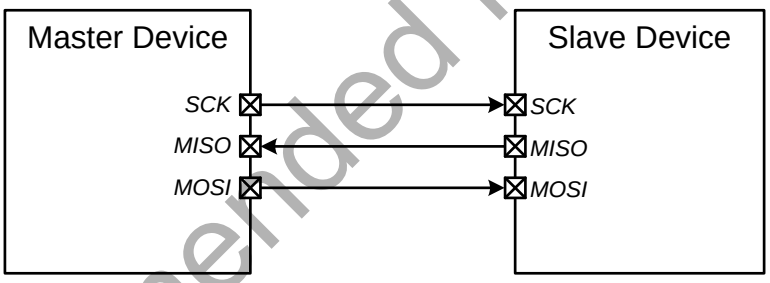


Figure 28.3. 3-Wire Single Master and 3-Wire Single Slave Mode Connection Diagram

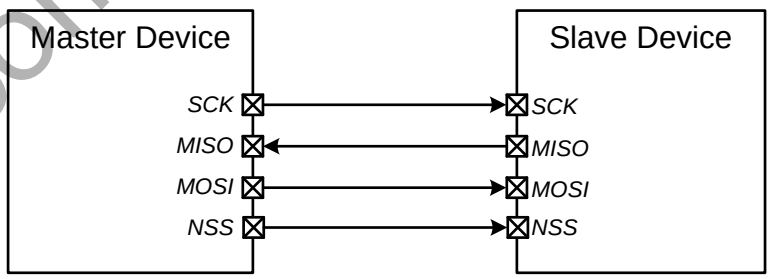


Figure 28.4. 4-Wire Single Master Mode and 4-Wire Slave Mode Connection Diagram

28.3. 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 28.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

The 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 28.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

28.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

All of the following bits must be cleared by software.

- 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.
- 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.
- 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 bit in SPI0CFG and SPIEN bit in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
- 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.

28.5. Serial Clock Phase and Polarity

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 28.5. For slave mode, the clock and data relationships are shown in Figure 28.6 and Figure 28.7. Note that CKPHA should be set to 0 on both the master and slave SPI when communicating between two Silicon Labs C8051 devices.

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

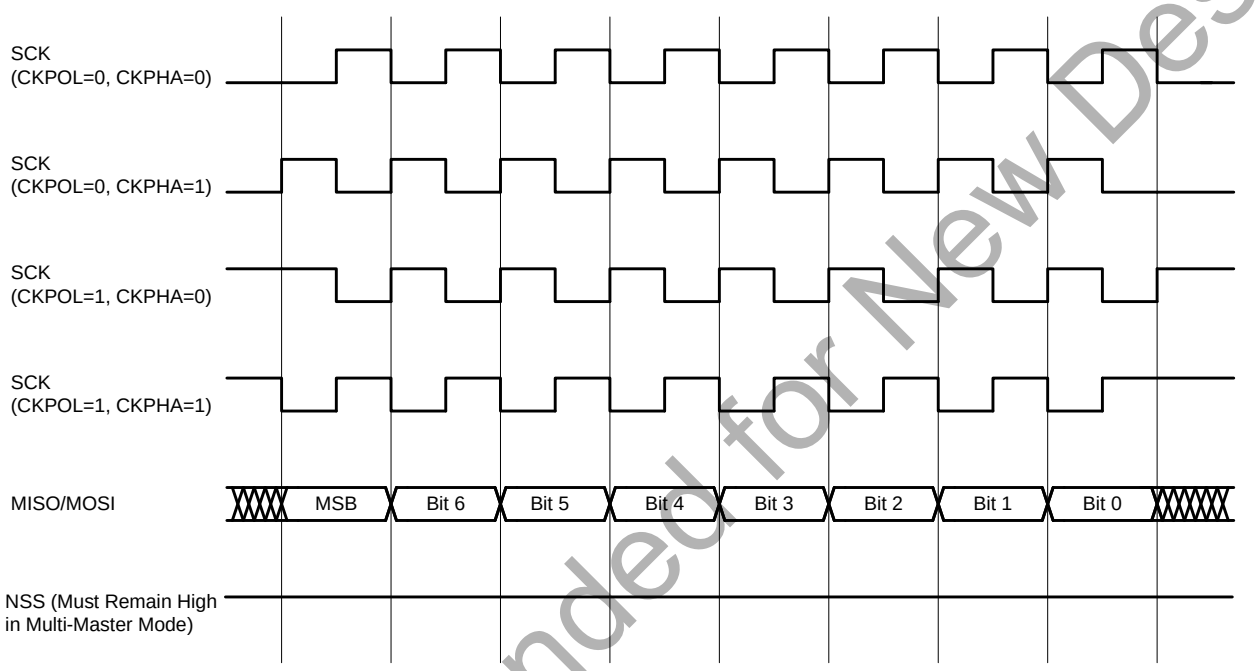


Figure 28.5. Master Mode Data/Clock Timing

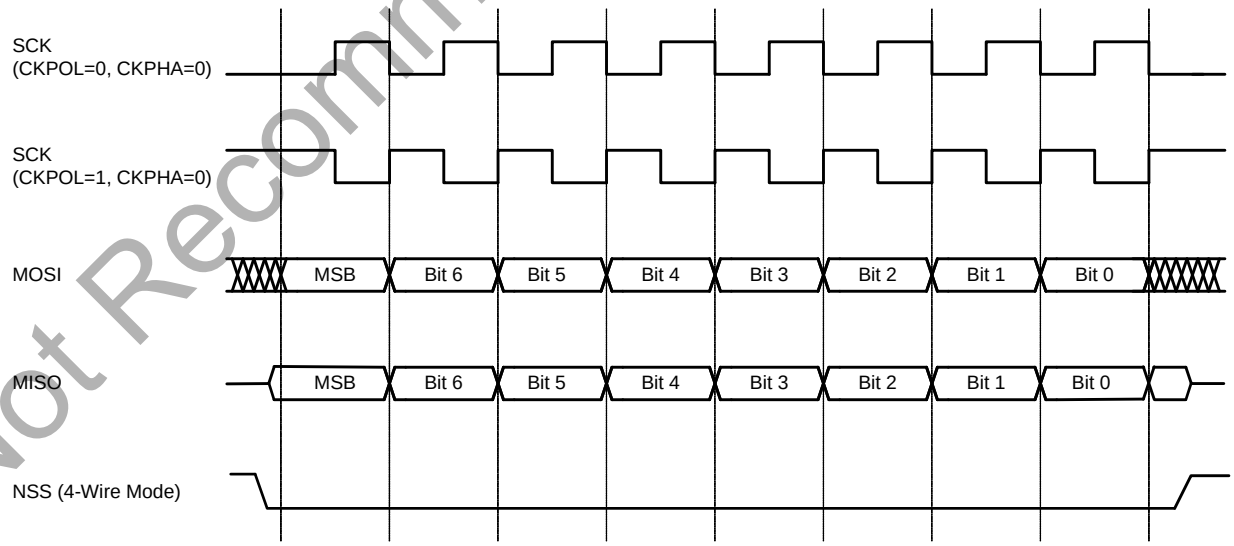


Figure 28.6. Slave Mode Data/Clock Timing (CKPHA = 0)

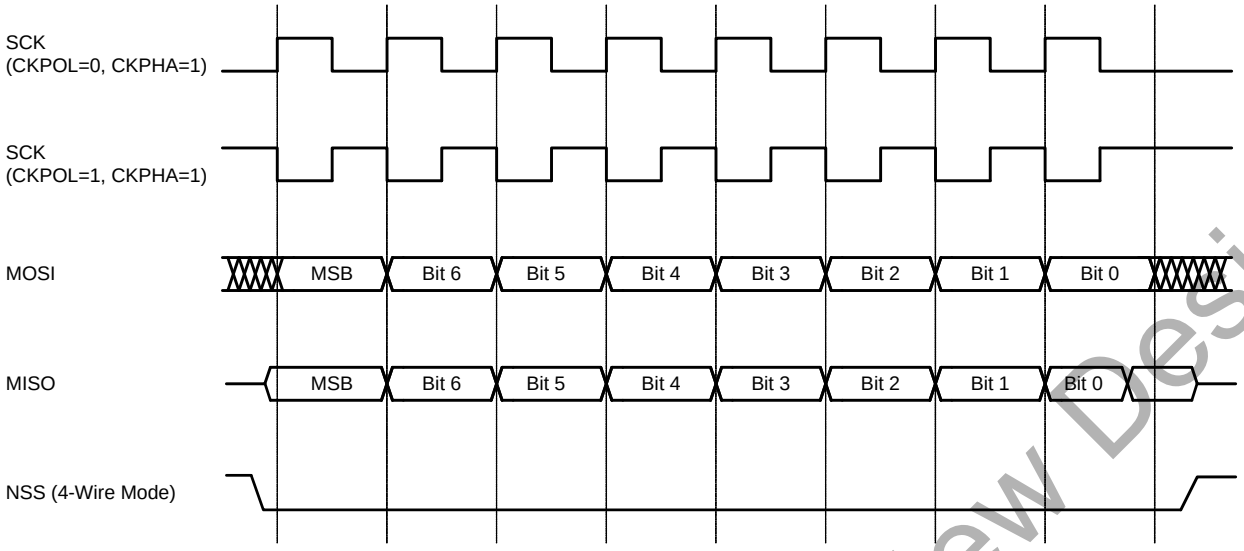
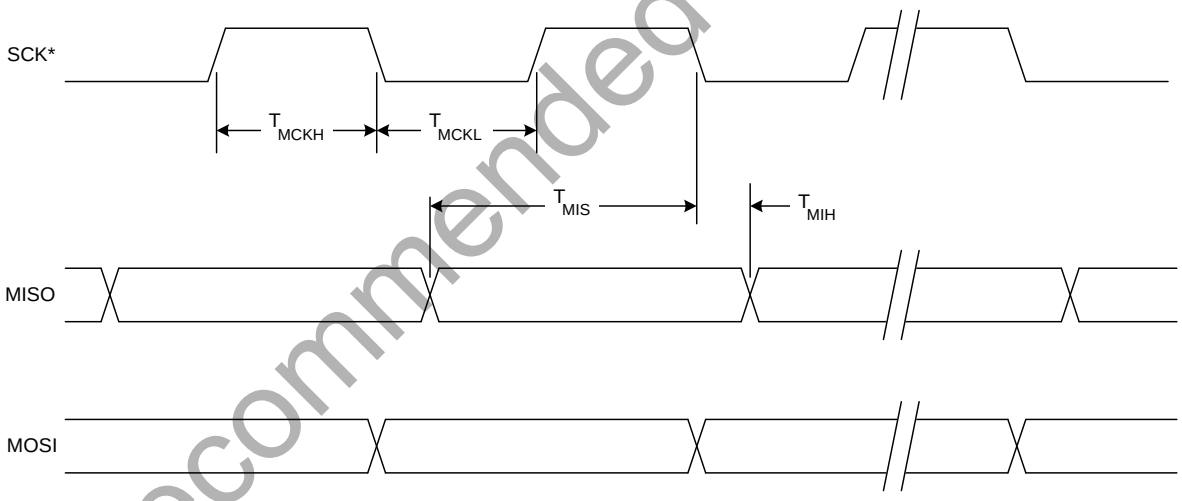


Figure 28.7. Slave Mode Data/Clock Timing (CKPHA = 1)

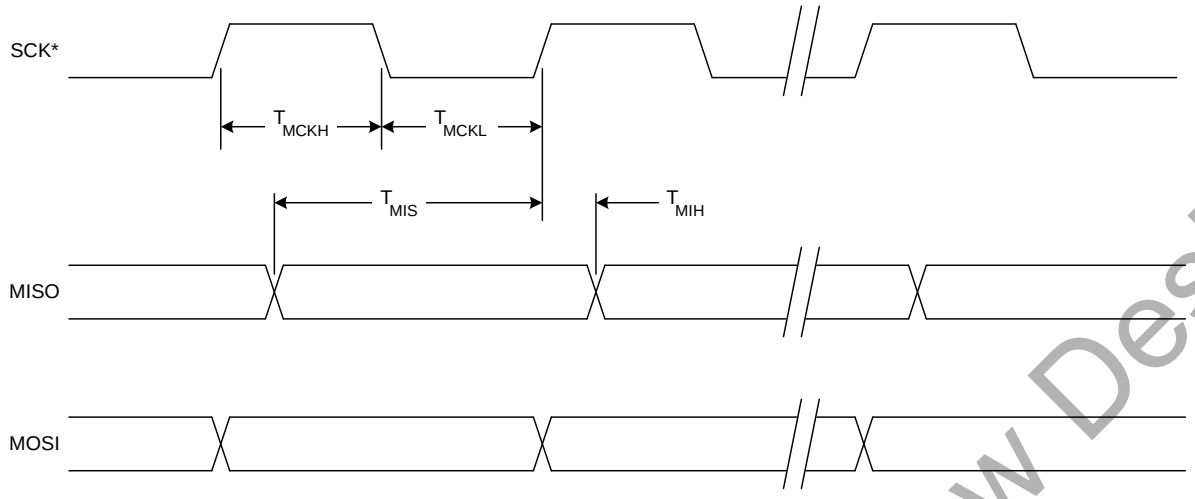
28.6. SPI Special Function Registers

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.



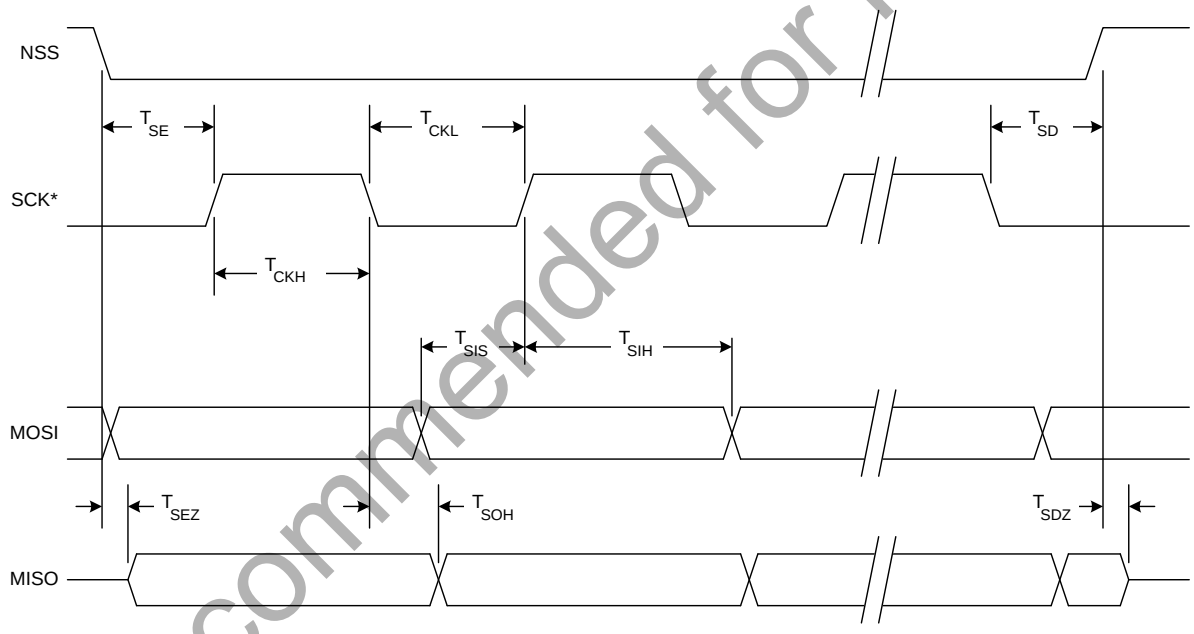
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 28.8. SPI Master Timing (CKPHA = 0)



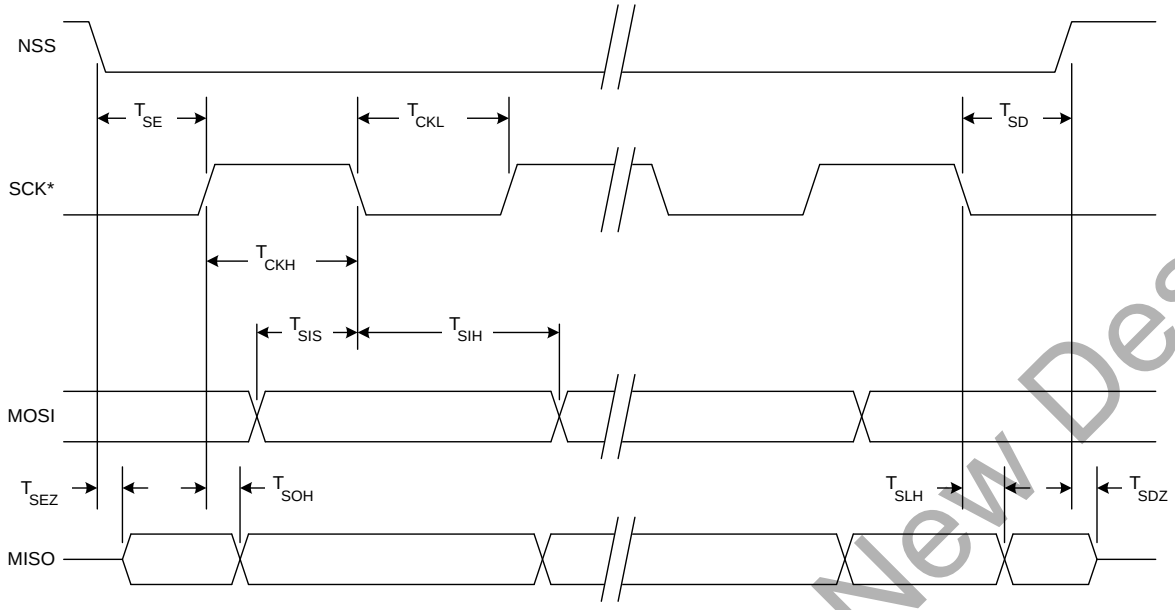
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 28.9. SPI Master Timing (CKPHA = 1)



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

Figure 28.10. SPI Slave Timing (CKPHA = 0)



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

Figure 28.11. SPI Slave Timing (CKPHA = 1)

Table 28.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing (See Figure 28.8 and Figure 28.9)				
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 28.10 and Figure 28.11)				
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).				

28.7. SPI Control Registers

Register 28.1. SPI0CFG: SPI0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT
Type	R	RW	RW	RW	R	R	R	R
Reset	0	0	0	0	0	1	1	1
SFR Page = 0x0; SFR Address: 0xA1								

Table 28.2. SPI0CFG Register Bit Descriptions

Bit	Name	Function
7	SPIBSY	**SPI Busy.** This bit is set to logic 1 when a SPI transfer is in progress (master or slave mode).
6	MSTEN	**Master Mode Enable.** 0: Disable master mode. Operate in slave mode. 1: Enable master mode. Operate as a master.
5	CKPHA	**SPI0 Clock Phase.** 0: Data centered on first edge of SCK period. 1: Data centered on second edge of SCK period.
4	CKPOL	**SPI0 Clock Polarity.** 0: SCK line low in idle state. 1: SCK line high in idle state.
3	SLVSEL	**Slave Selected Flag.** 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.
2	NSSIN	**NSS Instantaneous Pin Input.** 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.
1	SRMT	**Shift Register Empty.** This bit is valid in slave mode only and 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. SRMT = 1 when in Master Mode.
0	RXBMT	**Receive Buffer Empty.** This bit is valid in slave mode only and 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. 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.		

Register 28.2. SPI0CN: SPI0 Control

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	MODF	RXOVRN	NSSMD		TXBMT	SPIEN
Type	RW	RW	RW	RW	RW		R	RW
Reset	0	0	0	0	0	1	1	0
SFR Page = 0x0; SFR Address: 0xF8 (bit-addressable)								

Table 28.3. SPI0CN Register Bit Descriptions

Bit	Name	Function
7	SPIF	**SPI0 Interrupt Flag.** This bit is set to logic 1 by hardware at the end of a data transfer. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by firmware.
6	WCOL	**Write Collision Flag.** This bit is set to logic 1 if a write to SPI0DAT is attempted when TXBMT is 0. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by firmware.
5	MODF	**Mode Fault Flag.** This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD = 01). If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by firmware.
4	RXOVRN	**Receive Overrun Flag.** This bit is valid for slave mode only and is set to logic 1 by hardware 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. If SPI interrupts are enabled, an interrupt will be generated. This bit is not automatically cleared by hardware, and must be cleared by firmware.
3:2	NSSMD	**Slave Select Mode.** Selects between the following NSS operation modes: 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. NSS is an input to the device. 10: 4-Wire Single-Master Mode. NSS is an output and logic low. 11: 4-Wire Single-Master Mode. NSS is an output and logic high.
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.
0	SPIEN	**SPI0 Enable.** 0: Disable the SPI module. 1: Enable the SPI module.

Register 28.3. SPI0CKR: SPI0 Clock Rate

Bit	7	6	5	4	3	2	1	0
Name	SPI0CKR							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xA2								

Table 28.4. SPI0CKR Register Bit Descriptions

Bit	Name	Function
7:0	SPI0CKR	**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

Register 28.4. SPI0DAT: SPI0 Data

Bit	7	6	5	4	3	2	1	0
Name	SPI0DAT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xA3								

Table 28.5. SPI0DAT Register Bit Descriptions

Bit	Name	Function
7:0	SPI0DAT	**SPI0 Transmit and Receive Data.** The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0-DAT 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.

29. System Management Bus / I^{2}C (SMBus0)

The SMBus I/O interface is a two-wire, bidirectional serial bus. The SMBus is compliant with the System Management Bus Specification, version 1.1, and compatible with the I^{2}C serial bus.

Reads and writes to the SMBus 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 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. The SMBus peripherals can be fully driven by software (i.e., software accepts/rejects slave addresses, and generates ACKs), or hardware slave address recognition and automatic ACK generation can be enabled to minimize software overhead. A block diagram of the SMBus0 peripheral is shown in Figure 29.1.

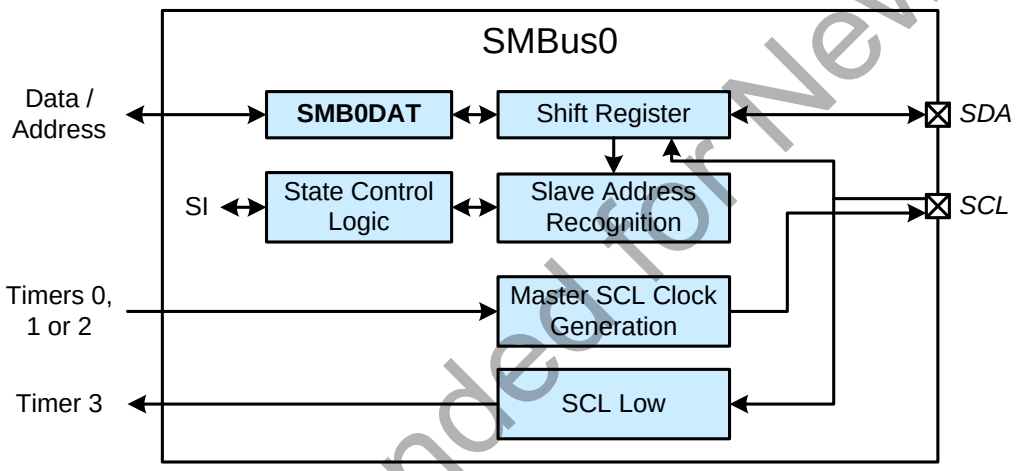


Figure 29.1. SMBus0 Block Diagram

29.1. Supporting Documents

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

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

29.2. SMBus Configuration

Figure 29.2 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. However, the maximum voltage on any port pin must conform to the electrical characteristics specifications. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pullup resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 300 ns and 1000 ns, respectively.

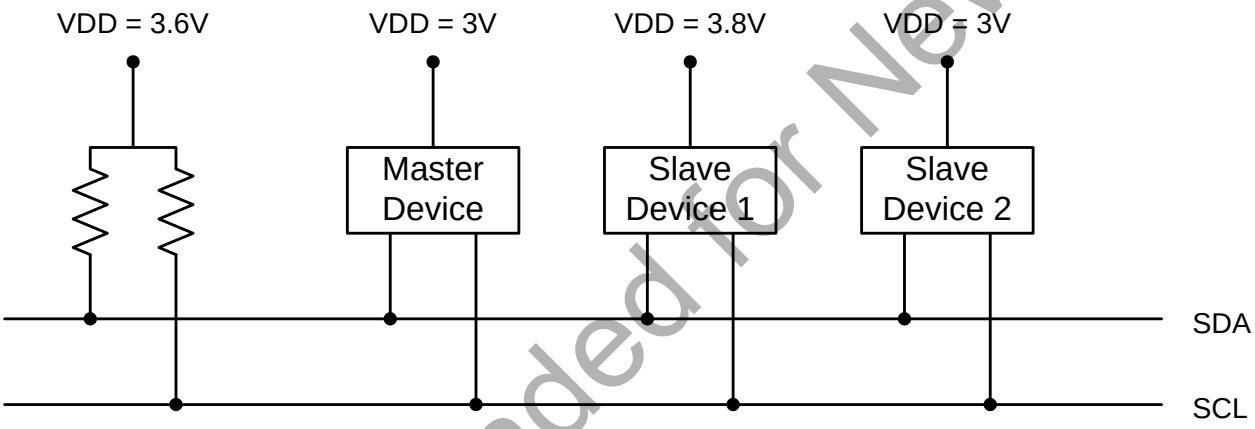


Figure 29.2. Typical SMBus Configuration

29.3. 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. 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. Bytes that are received (by a master or slave) are acknowledged (ACK) with a low SDA during a high SCL (see Figure 29.3). 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 29.3 illustrates a typical SMBus transaction.

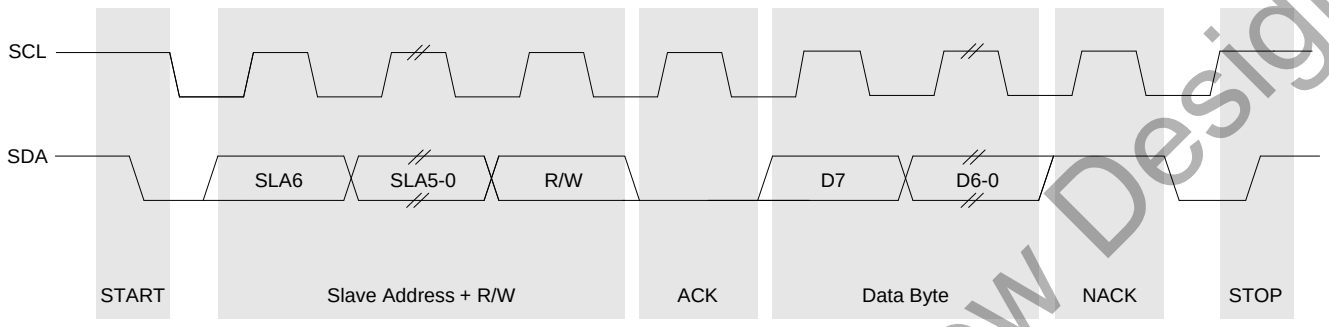


Figure 29.3. SMBus Transaction

29.3.1. Transmitter vs. Receiver

On the SMBus communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

29.3.2. 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 “29.3.5. SCL High (SMBus Free) Timeout” on page 345). 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.

29.3.3. 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.

29.3.4. 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.

For the SMBus0 interface, Timer 3 is used to implement SCL low timeouts. The SCL low timeout feature is enabled by setting the SMB0TOE bit in SMB0CF. The associated timer is forced to reload when SCL is high, and allowed to count when SCL is low. With the associated timer enabled and configured to overflow after 25 ms (and SMB0TOE set), the timer interrupt service routine can be used to reset (disable and re-enable) the SMBus in the event of an SCL low timeout.

29.3.5. 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 SMB0FTE 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 (as defined by the timer configured for the SMBus clock source). If the SMBus is waiting to generate a Master START, the START will be generated following this timeout. A clock source is required for free timeout detection, even in a slave-only implementation.

29.4. 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
- Optional hardware recognition of slave address and automatic acknowledgement of address/data

SMBus interrupts are generated for each data byte or slave address that is transferred. When hardware acknowledgement is disabled, the point at which the interrupt is generated depends on whether the hardware is acting as a data transmitter or receiver. When a transmitter (i.e., sending address/data, receiving an ACK), this interrupt is generated after the ACK cycle so that software may read the received ACK value; when receiving data (i.e., receiving address/data, sending an ACK), this interrupt is generated before the ACK cycle so that software may define the outgoing ACK value. If hardware acknowledgement is enabled, these interrupts are always generated after the ACK cycle. See Section 29.5 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. Table 29.5 provides a quick SMB0CN decoding reference.

29.4.1. 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 29.1. SMBus Clock Source Selection

SMBCS	SMBus0 Clock Source
00	Timer 0 Overflow
01	Timer 1 Overflow
10	Timer 2 High Byte Overflow
11	Timer 2 Low Byte Overflow

The SMBCS bit field selects 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 29.1.The selected clock source may be shared by other peripherals so long as the timer is left running at all times.

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

Equation 29.1. 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 29.1. 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 29.2.

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

Equation 29.2. Typical SMBus Bit Rate

Figure 29.4 shows the typical SCL generation described by Equation 29.2. 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 29.1.

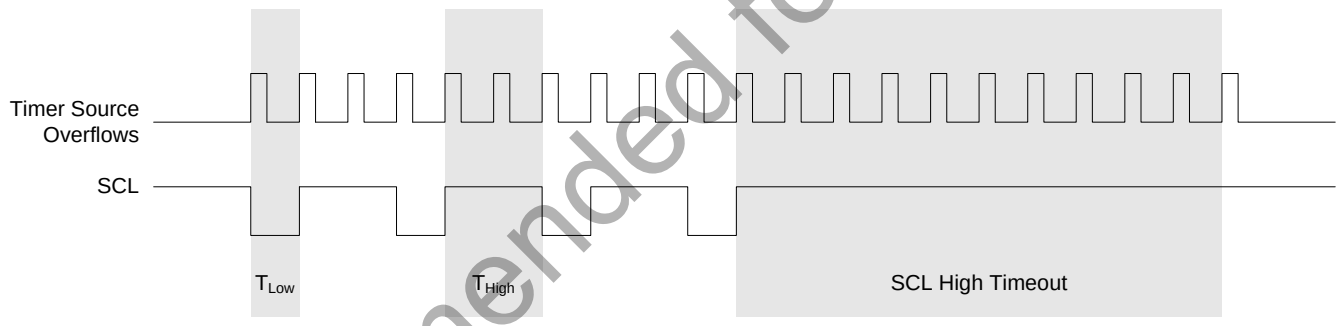


Figure 29.4. Typical SMBus SCL Generation

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 29.2 shows the minimum setup and hold times for the two EXTHOLD settings. Setup and hold time extensions are typically necessary for SMBus compliance when SYSCLK is above 10 MHz.

Table 29.2. 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

Table 29.2. Minimum SDA Setup and Hold Times

EXTHOLD	Minimum SDA Setup Time	Minimum SDA Hold Time
Note: Setup Time for ACK bit transmissions and the MSB of all data transfers. When using software acknowledgment, the s/w delay occurs between the time SMB0DAT or ACK is written and when SI0 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 “29.3.4. SCL Low Timeout” on page 344). The SMBus interface will force the associated timer to reload while SCL is high, and allow the timer to count when SCL is low. The timer 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 29.4).

29.4.2. SMB0CN Control Register

SMB0CN is used to control the interface and to provide status information. 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 indicates whether a device is the master or slave during the current transfer. TXMODE indicates whether the device is transmitting or receiving data for the current byte.

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.

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 29.3 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.

29.4.2.1. Software ACK Generation

When the EHACK bit in register SMB0ADM is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during 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.

29.4.2.2. Hardware ACK Generation

When the EHACK bit in register SMB0ADM is set to 1, automatic slave address recognition and ACK generation is enabled. More detail about automatic slave address recognition can be found in Section 29.4.3. As a receiver, the value currently specified by the ACK bit will be automatically sent on the bus during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQ bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, further slave events will be ignored until the next START is detected, and no interrupt will be generated.

Table 29.3 lists all sources for hardware changes to the SMB0CN bits. Refer to Table 29.5 for SMBus status decoding using the SMB0CN register.

Table 29.3. 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 (only when hardware ACK is not enabled).	■ 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 SIn 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.

29.4.3. Hardware Slave Address Recognition

The SMBus hardware has the capability to automatically recognize incoming slave addresses and send an ACK without software intervention. Automatic slave address recognition is enabled by setting the EHACK bit in register SMB0ADM to 1. This will enable both automatic slave address recognition and automatic hardware ACK generation for received bytes (as a master or slave). More detail on automatic hardware ACK generation can be found in Section 29.4.2.2.

The registers used to define which address(es) are recognized by the hardware are the SMBus Slave Address register and the SMBus Slave Address Mask register. A single address or range of addresses (including the General Call Address 0x00) can be specified using these two registers. The most-significant seven bits of the two registers are used to define which addresses will be ACKed. A 1 in a bit of the slave address mask SLVM enables a comparison between the received slave address and the hardware's slave address SLV for that bit. A 0 in a bit of

the slave address mask means that bit will be treated as a “don’t care” for comparison purposes. In this case, either a 1 or a 0 value are acceptable on the incoming slave address. Additionally, if the GC bit in register SMB0ADR is set to 1, hardware will recognize the General Call Address (0x00). Table 29.4 shows some example parameter settings and the slave addresses that will be recognized by hardware under those conditions.

Table 29.4. Hardware Address Recognition Examples (EHACK = 1)

Hardware Slave Address SLV	Slave Address Mask SLVM	GC bit	Slave Addresses Recognized by Hardware
0x34	0x7F	0	0x34
0x34	0x7F	1	0x34, 0x00 (General Call)
0x34	0x7E	0	0x34, 0x35
0x34	0x7E	1	0x34, 0x35, 0x00 (General Call)
0x70	0x73	0	0x70, 0x74, 0x78, 0x7C

29.4.4. 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.

29.5. 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. The position of the ACK interrupt when operating as a receiver depends on whether hardware ACK generation is enabled. As a receiver, the interrupt for an ACK occurs *before* the ACK with hardware ACK generation disabled, and *after* the ACK when hardware ACK generation is enabled. As a transmitter, interrupts occur *after* the ACK, regardless of whether hardware ACK generation is enabled or not.

29.5.1. Write Sequence (Master)

During a write sequence, an SMBus master writes data to a slave device. The master in this transfer will be a transmitter during the address byte, and a transmitter during all data bytes. 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. The interface will switch to Master Receiver Mode if SMB0DAT is not written following a Master Transmitter interrupt. Figure 29.5 shows a typical master write sequence. Two transmit data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur *after* the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

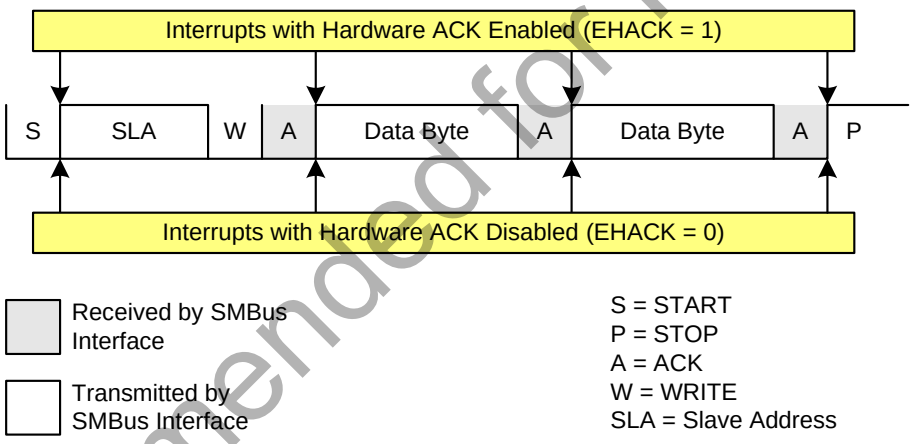


Figure 29.5. Typical Master Write Sequence

29.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. 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.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

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 for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 29.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the 'data byte transferred' interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs *before* the ACK with hardware ACK generation disabled, and *after* the ACK when hardware ACK generation is enabled.

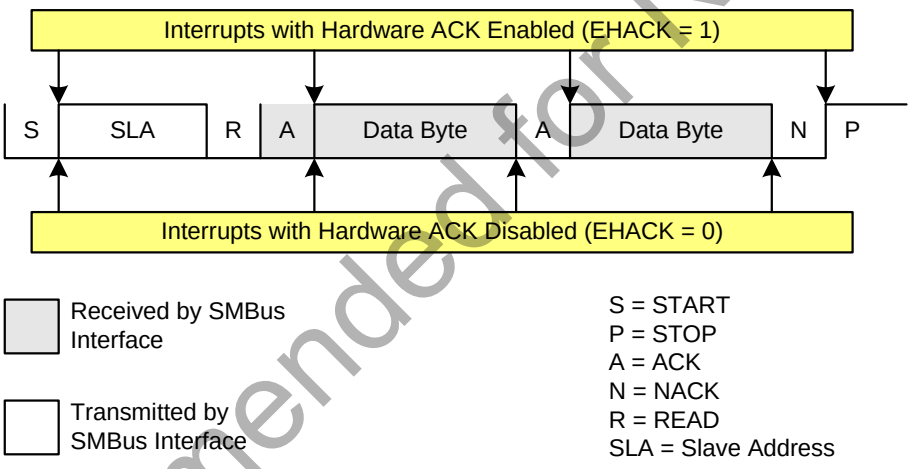


Figure 29.6. Typical Master Read Sequence

29.5.3. Write Sequence (Slave)

During a write sequence, an SMBus master writes data to a slave device. The slave in this transfer will be a receiver during the address byte, and a receiver during all data bytes. 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. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), 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.

If hardware ACK generation is disabled, the ACKRQ is set to 1 and an interrupt is generated after each received byte. Software must write the ACK bit at that time to ACK or NACK the received byte.

With hardware ACK generation enabled, the SMBus hardware will automatically generate the ACK/NACK, and then post the interrupt. **It is important to note that the appropriate ACK or NACK value should be set up by the software prior to receiving the byte when hardware ACK generation is enabled.**

The interface exits Slave Receiver Mode after receiving a STOP. The interface will switch to Slave Transmitter Mode if SMB0DAT is written while an active Slave Receiver. Figure 29.7 shows a typical slave write sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur at different places in the sequence, depending on whether hardware ACK generation is enabled. The interrupt occurs *before* the ACK with hardware ACK generation disabled, and *after* the ACK when hardware ACK generation is enabled.

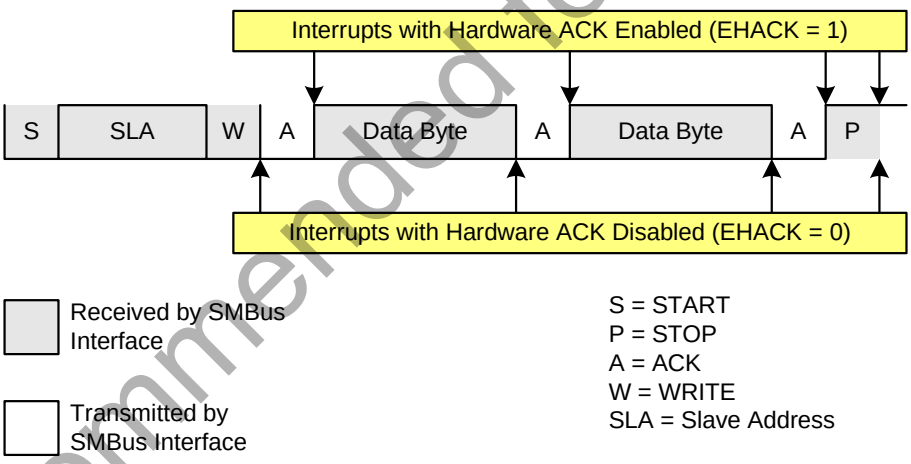


Figure 29.7. Typical Slave Write Sequence

29.5.4. Read Sequence (Slave)

During a read sequence, an SMBus master reads data from a slave device. The slave in this transfer will be a receiver during the address byte, and a transmitter during all data bytes. 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. If hardware ACK generation is disabled, upon entering Slave Receiver Mode, an interrupt is generated and the ACKRQ bit is set. The software must respond to the received slave address with an ACK, or ignore the received slave address with a NACK. If hardware ACK generation is enabled, the hardware will apply the ACK for a slave address which matches the criteria set up by SMB0ADR and SMB0ADM. The interrupt will occur after the ACK cycle.

If the received slave address is ignored (by software or hardware), slave interrupts will be inhibited until the next START is detected. If the received slave address is acknowledged, zero or more data bytes are transmitted. 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 (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. The interface will switch to slave receiver mode if SMB0DAT is not written following a Slave Transmitter interrupt. Figure 29.8 shows a typical slave read sequence. Two transmitted data bytes are shown, though any number of bytes may be transmitted. Notice that all of the “data byte transferred” interrupts occur *after* the ACK cycle in this mode, regardless of whether hardware ACK generation is enabled.

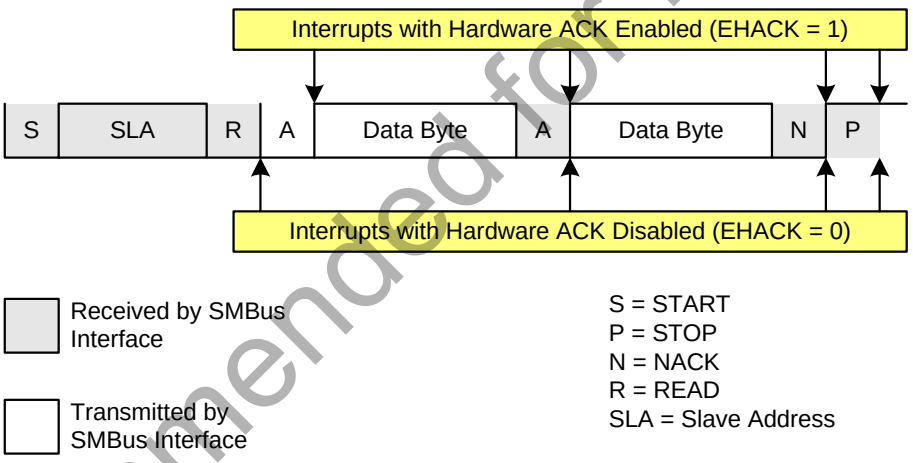


Figure 29.8. Typical Slave Read Sequence

29.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. The appropriate actions to take in response to an SMBus event depend on whether hardware slave address recognition and ACK generation is enabled or disabled. Table 29.5 describes the typical actions when hardware slave address recognition and ACK generation is disabled. Table 29.6 describes the typical actions when hardware slave address recognition and ACK generation is enabled. In the tables, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. The shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed by hardware but do not conform to the SMBus specification.

Table 29.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	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
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0-DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X	1000
Master Receiver	1000	1	0	X	A master data byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	1000
						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	1110
						Send ACK followed by repeated START.	1	0	1	1110
						Send NACK to indicate last byte, and send repeated START.	1	0	0	1110
						Send ACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	1	1100
						Send NACK and switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	0	1100

Table 29.5. SMBus Status Decoding: Hardware ACK Disabled (EHACK = 0) (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	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	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	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 + R/W was received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
		1	1	X	Lost arbitration as master; slave address + R/W received; ACK requested.	If Write, Acknowledge received address	0	0	1	0000
						If Read, Load SMB0DAT with data byte; ACK received address	0	0	1	0100
						NACK received address.	0	0	0	—
						Reschedule failed transfer; NACK received address.	1	0	0	1110
	0001	0	0	X	A STOP was detected while addressed as a Slave Transmitter or Slave Receiver.	Clear STO.	0	0	X	—
		1	1	X	Lost arbitration while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	1	0	X	A slave byte was received; ACK requested.	Acknowledge received byte; Read SMB0DAT.	0	0	1	0000
						NACK received byte.	0	0	0	—
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	1	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	0	—
						Reschedule failed transfer.	1	0	0	1110

Table 29.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	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
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X	1110
						Abort transfer.	0	1	X	—
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0-DAT.	0	0	X	1100
						End transfer with STOP.	0	1	X	—
						End transfer with STOP and start another transfer.	1	1	X	—
						Send repeated START.	1	0	X	1110
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT). Set ACK for initial data byte.	0	0	1	1000
Master Receiver	1000	0	0	1	A master data byte was received; ACK sent.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	1000
						Set NACK to indicate next data byte as the last data byte; Read SMB0DAT.	0	0	0	1000
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100
		0	0	0	A master data byte was received; NACK sent (last byte).	Read SMB0DAT; send STOP.	0	1	0	—
						Read SMB0DAT; Send STOP followed by START.	1	1	0	1110
						Initiate repeated START.	1	0	0	1110
						Switch to Master Transmitter Mode (write to SMB0DAT before clearing SI).	0	0	X	1100

Table 29.6. SMBus Status Decoding: Hardware ACK Enabled (EHACK = 1) (Continued)

Mode	Values Read				Current SMbus State	Typical Response Options	Values to Write			Next Status Vector Expected
	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	0001
		0	0	1	A slave byte was transmitted; ACK received.	Load SMB0DAT with next data byte to transmit.	0	0	X	0100
		0	1	X	A Slave byte was transmitted; error detected.	No action required (expecting Master to end transfer).	0	0	X	0001
	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	0	0	X	A slave address + R/W was received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
		0	1	X	Lost arbitration as master; slave address + R/W received; ACK sent.	If Write, Set ACK for first data byte.	0	0	1	0000
						If Read, Load SMB0DAT with data byte	0	0	X	0100
						Reschedule failed transfer	1	0	X	1110
	0001	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 while attempting a STOP.	No action required (transfer complete/aborted).	0	0	0	—
	0000	0	0	X	A slave byte was received.	Set ACK for next data byte; Read SMB0DAT.	0	0	1	0000
						Set NACK for next data byte; Read SMB0DAT.	0	0	0	0000
Bus Error Condition	0010	0	1	X	Lost arbitration while attempting a repeated START.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0001	0	1	X	Lost arbitration due to a detected STOP.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110
	0000	0	1	X	Lost arbitration while transmitting a data byte as master.	Abort failed transfer.	0	0	X	—
						Reschedule failed transfer.	1	0	X	1110

29.7. I2C / SMBus Control Registers

Register 29.1. SMB0CF: SMBus 0 Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS	
Type	RW	RW	R	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xC1								

Table 29.7. SMB0CF Register Bit Descriptions

Bit	Name	Function
7	ENSMB	**SMBus Enable.** This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	**SMBus Slave Inhibit.** When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	**SMBus Busy Indicator.** This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	**SMBus Setup and Hold Time Extension Enable.** This bit controls the SDA setup and hold times. 0: Disable SDA extended setup and hold times. 1: Enable SDA extended setup and hold times.
3	SMBTOE	**SMBus SCL Timeout Detection Enable.** This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	**SMBus Free Timeout Detection Enable.** When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS	**SMBus Clock Source Selection.** This field selects the SMBus clock source, which is used to generate the SMBus bit rate. See the SMBus clock timing section for additional details. 00: Timer 0 Overflow. 01: Timer 1 Overflow. 10: Timer 2 High Byte Overflow. 11: Timer 2 Low Byte Overflow.

Register 29.2. SMB0CN: SMBus 0 Control

Bit	7	6	5	4	3	2	1	0
Name	MASTER	TXMODE	STA	STO	ACKRQ	ARBLOST	ACK	SI
Type	R	R	RW	RW	R	R	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xC0 (bit-addressable)								

Table 29.8. SMB0CN Register Bit Descriptions

Bit	Name	Function
7	MASTER	**SMBus Master/Slave Indicator.** This read-only bit indicates when the SMBus is operating as a master. 0: SMBus operating in slave mode. 1: SMBus operating in master mode.
6	TXMODE	**SMBus Transmit Mode Indicator.** This read-only bit indicates when the SMBus is operating as a transmitter. 0: SMBus in Receiver Mode. 1: SMBus in Transmitter Mode.
5	STA	**SMBus Start Flag.** When reading STA, a '1' indicates that a start or repeated start condition was detected on the bus. Writing a '1' to the STA bit initiates a start or repeated start on the bus.
4	STO	**SMBus Stop Flag.** When reading STO, a '1' indicates that a stop condition was detected on the bus (in slave mode) or is pending (in master mode). When acting as a master, writing a '1' to the STO bit initiates a stop condition on the bus. This bit is cleared by hardware.
3	ACKRQ	**SMBus Acknowledge Request.** 0: No ACK requested. 1: ACK requested.
2	ARBLOST	**SMBus Arbitration Lost Indicator.** 0: No arbitration error. 1: Arbitration error occurred.
1	ACK	**SMBus Acknowledge.** When read as a master, the ACK bit indicates whether an ACK (1) or NACK (0) is received during the most recent byte transfer. As a slave, this bit should be written to send an ACK (1) or NACK (0) to a master request. Note that the logic level of the ACK bit on the SMBus interface is inverted from the logic of the register ACK bit.

Table 29.8. SMB0CN Register Bit Descriptions

Bit	Name	Function
0	SI	**SMBus Interrupt Flag.** This bit is set by hardware to indicate that the current SMBus state machine operation (such as writing a data or address byte) is complete. While SI is set, SCL is held low and SMBus is stalled. SI must be cleared by firmware. Clearing SI initiates the next SMBus state machine operation.

Register 29.3. SMB0ADR: SMBus 0 Slave Address

Bit	7	6	5	4	3	2	1	0
Name	SLV							GC
Type	RW							RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xF4								

Table 29.9. SMB0ADR Register Bit Descriptions

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

Register 29.4. SMB0ADM: SMBus 0 Slave Address Mask

Bit	7	6	5	4	3	2	1	0
Name	SLVM							EHACK
Type	RW							RW
Reset	1	1	1	1	1	1	1	0
SFR Page = 0x0; SFR Address: 0xF5								

Table 29.10. SMB0ADM Register Bit Descriptions

Bit	Name	Function
7:1	SLVM	**SMBus Slave Address Mask.** Defines which bits of register SMB0ADR are compared with an incoming address byte, and which bits are ignored. Any bit set to 1 in SLVM enables comparisons with the corresponding bit in SLV. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
0	EHACK	**Hardware Acknowledge Enable.** Enables hardware acknowledgement of slave address and received data bytes. 0: Firmware must manually acknowledge all incoming address and data bytes. 1: Automatic slave address recognition and hardware acknowledge is enabled.

Register 29.5. SMB0DAT: SMBus 0 Data

Bit	7	6	5	4	3	2	1	0
Name	SMB0DAT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xC2								

Table 29.11. SMB0DAT Register Bit Descriptions

Bit	Name	Function
7:0	SMB0DAT	**SMBus 0 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 safely read from or write to this register whenever the SI serial interrupt flag 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.

30. I²C Slave

The I2CSLAVE0 interface is a 2-wire, bidirectional serial bus that is compatible with the I²C Bus Specification 3.0. It is capable of transferring in high-speed mode (HS-mode) at speeds of up to 3.4 Mbps. Either the CPU or the DMA can write to the I²C interface, and the I²C interface can autonomously control the serial transfer of data. The interface also supports clock stretching for cases where the CPU may be temporarily prohibited from transmitting a byte or processing a received byte during an I²C transaction. It can also operate in sleep mode without an active system clock and wake the CPU when a matching slave address is received.

It operates only as an I²C slave device. The I2CSLAVE0 peripheral provides control of the SCL (serial clock) synchronization, SDA (serial data), SCL Clock stretching, I²C arbitration logic, and low power mode operation. The block diagram of the I2CSLAVE0 peripheral and the associated SFRs is shown in Figure 30.1.

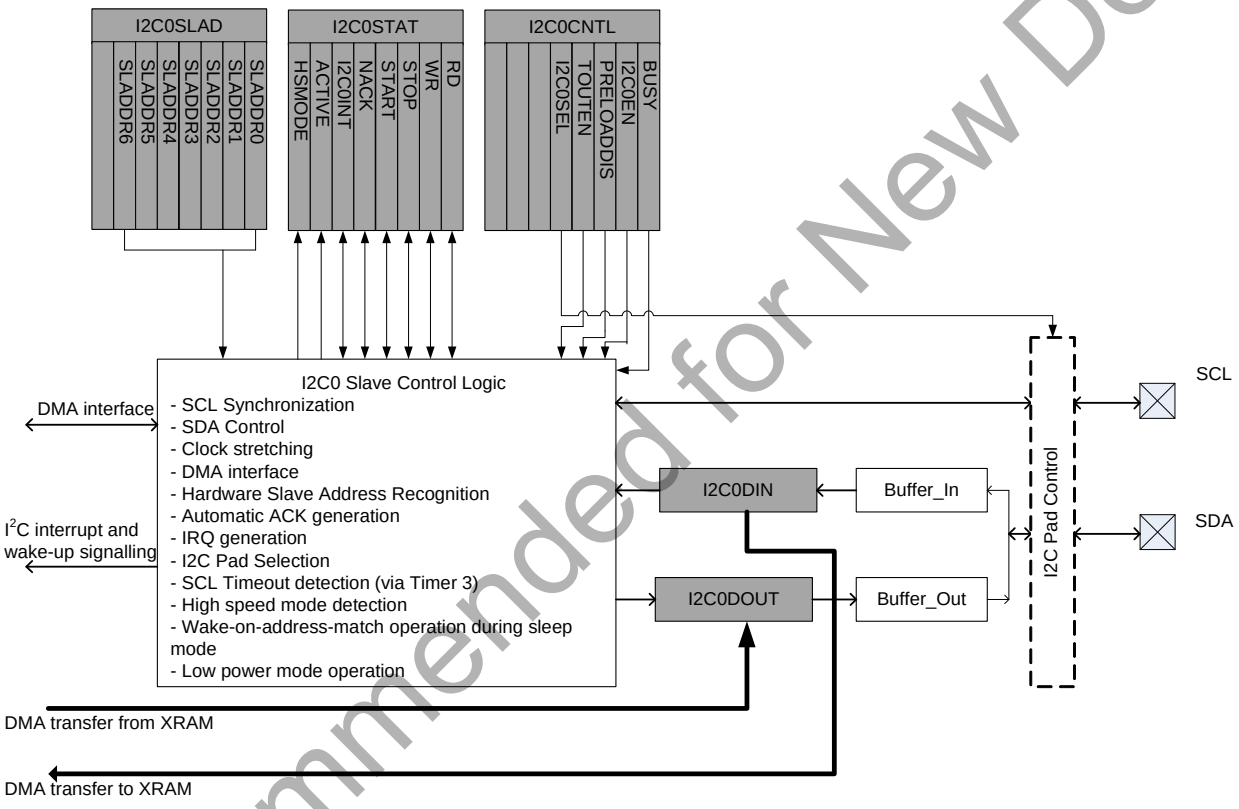


Figure 30.1. I2CSLAVE0 Block Diagram

30.1. Supporting Documents

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

- The I²C-bus specification and the user manual Rev. 0.3.

30.2. The I²C Configuration

Figure 30.2 shows a typical I²C configuration. The I²C specification allows any recessive voltage between 3.0 and 5.0 V; different devices on the bus may operate at different voltage levels.

Note: The port pins on the C8051F97x devices are not 5 V tolerant, therefore, the device may only be used in I²C networks where the supply voltage does not exceed V_{DD}.

The bidirectional SCL and SDA 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 the specifications defined in the I²C standard.

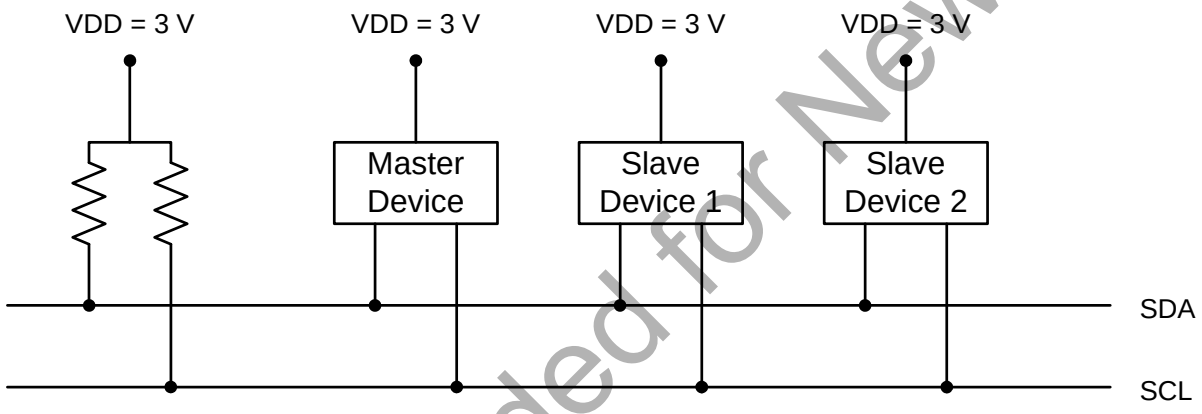


Figure 30.2. Typical I²C Configuration

30.3. I2CSLAVE0 Operation

The I2CSLAVE0 peripheral supports two types of data transfers: I²C Read data transfers where data is transferred from the C8051F97x's I²C slave peripheral to an I²C master, and I²C Write data transfers where data is transferred from an I²C master to the C8051F97x's I²C slave peripheral. The I²C master initiates both types of data transfers and provides the serial clock pulses that the I²C slave peripheral detects on the SCL pin.

A typical I²C 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. Bytes that are received are acknowledged (ACK) with a low SDA during a high SCL (refer to Figure 30.3).

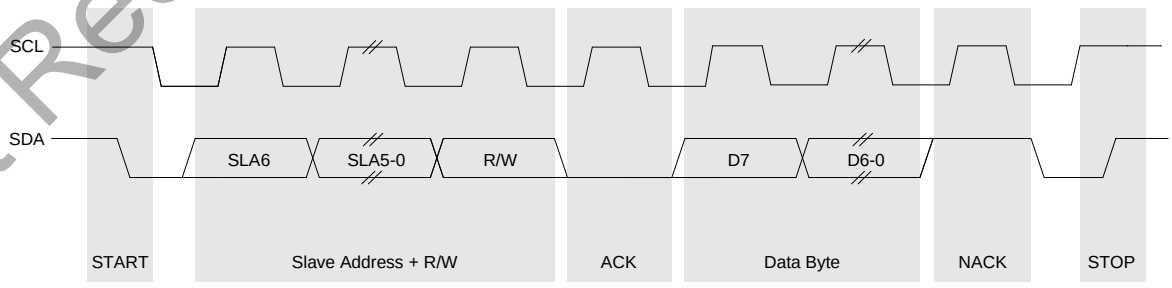


Figure 30.3. I²C Transaction

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 30.3 illustrates a typical I^2C transaction.

30.3.1. Transmitter vs. Receiver

On the I^2C communications interface, a device is the “transmitter” when it is sending an address or data byte to another device on the bus. A device is a “receiver” when an address or data byte is being sent to it from another device on the bus. The transmitter controls the SDA line during the address or data byte. After each byte of address or data information is sent by the transmitter, the receiver sends an ACK or NACK bit during the ACK phase of the transfer, during which time the receiver controls the SDA line.

30.3.2. Clock Stretching

The I^2C bus specification provides transaction pause mechanism, which allows the slave device to force the master into a wait state until the slave device is ready for the next byte transaction. This is performed by the I^2C slave holding the SCL line low. Hence, it is important that the master I^2C device must not drive the SCL line using a push-pull output.

In the C8051F97x I2CSLAVE0 peripheral, clock stretching is only performed on the SCL falling edge associated with the ACK or NACK bit. Clock stretching is always performed on every byte transaction that is addressed to the I2CSLAVE0 peripheral. Clock stretching is completed by the I2CSLAVE0 peripheral when it releases the SCL line from the low state. The I2CSLAVE0 peripheral releases the SCL line when any one of the following conditions are met:

- Software writes a 0 to the I2C0INT bit in I2C0STAT,
- DMA completes a data transfer to or from the I2CSLAVE0 peripheral in response to a DMA request from the I2CSLAVE0 peripheral.

30.3.3. 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 I2CSLAVE0 peripheral supports a timeout feature to allow the firmware to detect and handle this condition.

This feature is enabled when the TOUTEN bit in I2C0CNTL is set, and Timer 3 is configured to run in 16-bit auto-reload mode (T3SPLIT set to 0 in TMR3CN). When this feature is enabled, 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 a system-defined time (and TOUTEN bit set), the Timer 3 interrupt service routine can be used to detect and handle this error condition.

30.3.4. HS-mode

The I^2C Specification supports High-speed mode (HS-mode) transfer which allows devices to transfer data at rates of up to 3.4 Mbps and yet remain fully downward compatible with slower speed devices. This allows HS-mode devices to operate in a mixed-speed bus system. Refer to the I^2C Specification for details on the electrical and timing requirements for HS-mode operation. The I2CSLAVE0 peripheral is compatible with the I^2C HS-mode operation without any software intervention other than requiring that firmware enable the I2CSLAVE0 peripheral.

By default, the I^2C bus operates at speeds of up to Fast-mode (F/S mode) only, where the maximum transfer rate is 400 kbps. The I^2C bus switches to from F/S mode to HS-mode only after the following sequence of bits appear on the I^2C bus:

1. START bit (S)
2. 8-bit master code (0000 1XXX)
3. NACK bit (N)

The HS-mode master codes are reserved 8-bit codes which are not used for slave addressing or other purposes. An HS-mode compatible I^2C master device will switch the I^2C bus to HS-mode by transmitting the above sequence of bits on the I^2C bus at a transfer rate of not more than 400 kbps. After that, the master can switch to HS-mode to transfer data at a rate of up to 3.4 Mbps. The I^2C bus switches back to F/S mode when the I^2C master transmits a STOP bit. Figure 30.4 shows this in clearer detail.

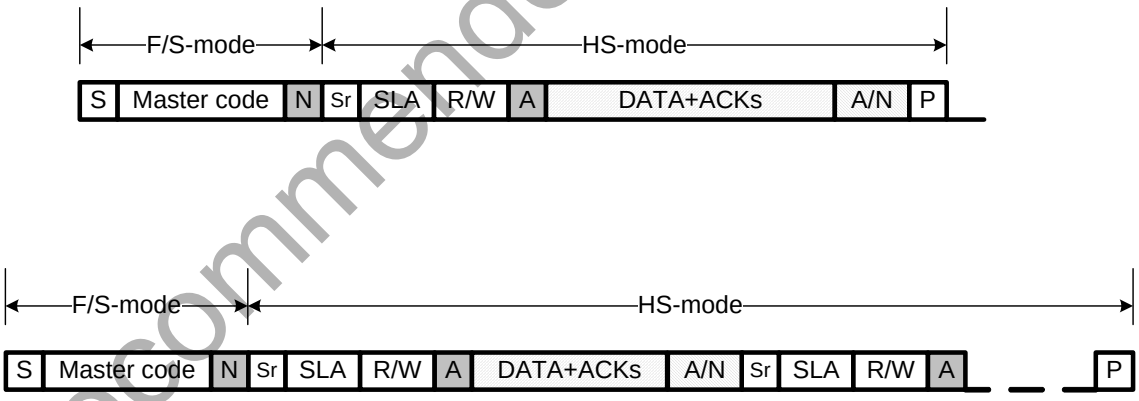


Figure 30.4. Data Transfer Switching between F/S Mode and HS-Mode

30.3.5. DMA and CPU Mode Operations

The I2CSLAVE0 peripheral can operate in either CPU or DMA mode. In CPU mode, all data transfers occur through software reading from the I2C0DIN register or writing to the I2C0DOUT register. By default, the I2CSLAVE0 peripheral operates in CPU mode for all I^2C Read and Write requests.

In DMA mode, all data transfers are executed by the DMA peripheral automatically without any CPU intervention. However, I2C0INT must still be cleared by firmware after a START+Slave address has been received. When a DMA channel has been selected and enabled for data transfer from I2C0DIN to XRAM, the I2CSLAVE0 peripheral operates in DMA mode for all I^2C Write requests. When a DMA channel has been selected and enabled for data transfer from XRAM to I2C0DOUT, the I2CSLAVE0 peripheral operates in DMA mode for all I^2C Read requests.

30.4. Using the I2CSLAVE0 Module

I2CSLAVE0 operates only in Slave mode. The interface provides timing and shifting control for serial transfers; higher level protocol is determined by user software. The I^2C interface provides the following application-independent features:

- Byte-wise serial data transfers
- SDA data synchronization
- Timeout recognition, as defined by the I2C0CNTL configuration register
- START/STOP detection
- Interrupt generation
- Status information
- Interfacing to the DMA to automate data transfers
- High-speed I^2C mode detection
- Automatic wakeup from Sleep mode when matching slave address is received
- Hardware recognition of slave address and automatic acknowledgment of address/data

An I2CSLAVE0 interrupt is generated when the RD, WR or STOP bit is set in the I2C0STAT SFR. It is also generated when the ACTIVE bit goes low to indicate the end of an I^2C bus transfer. Refer to the I2C0STAT SFR definition for complete details on the conditions for the setting and clearing of these bits.

30.4.1. I2C0CNTL Control Register

The I2C0CNTL register is used to control the I2CSLAVE0 interface (see SFR Definition 26.5). The two bits, I2C0SEL and I2C0EN, are used to enable and disable the I2CSLAVE0 interface and associated pins. The I2CSLAVE0 peripheral must be enabled in the following sequence:

1. Set I2C0SEL bit in the I2C0CNTL SFR.
2. Set I2C0EN bit in the I2C0CNTL SFR.

This correct sequence of enabling the I2CSLAVE0 ensures the peripheral processes the initial data transfers correctly.

TOUTEN is used to enable the SCL Low Timeout detection. Refer to section SCL Low Timeout for more details.

PRELOADDIS is used to control whether the data byte must be preloaded before the first SCL clock of an I^2C Read transaction. It should always be set to 1.

BUSY bit controls the automatic hardware acknowledgment of I^2C slave address match or I^2C Write transactions. When cleared to 0, automatic hardware acknowledgment is enabled.

30.4.2. I2C0STAT Status Register

The I2C0STAT register is used to provide status information (see SFR Definition 26.4). The HSMODE bit is used to indicate whether the I2C bus is operating in High-Speed mode as defined in the I2C Specification. This bit is set after the I2C bus is detected to be operating in HS-mode (refer to section HS-mode for details on how the I2C bus switch modes). When the I2C bus switches back to F/S-mode operation, the HSMODE bit is cleared.

The setting and clearing of the status bits are described in detail in section I2C Transfer Modes and the SFR Definition 26.4.

30.4.3. I2C0SLAD Slave Address Register

The I2CSLAVE0 peripheral can be configured to recognize a specific slave address and respond with an ACK without any software intervention. This feature is enabled by software in the following sequence writes to the I2CSLAVE0 registers;

1. Clearing BUSY bit in I2C0CNTL to enable automatic ACK response.
2. Writing the slave address to I2C0SLAD.
3. Setting I2C0SEL bit in I2C0CNTL to enable the SCL and SDA pads.
4. Setting I2C0EN bit in I2C0CNTL to enable the I2CSLAVE0 peripheral.

30.4.4. I2C0DIN Received Data Register

The I2C0DIN register holds the serial data that has just been received on the I^2C bus and addressed to the local device. When the I2CSLAVE0 is operating in CPU mode for I^2C Write operations, software may safely read from this register when the I2C0INT flag is set. It is not safe to read from this register in any one of the following conditions:

- The I2C0INT flag is cleared to logic 0.
- The I2CSLAVE0 is operating in DMA mode for I^2C Write operations.

When the I2CSLAVE0 is operating in DMA mode for I^2C Write operations, software should access the received data from the XRAM where the DMA has transferred the received data.

30.4.5. I2C0DOUT Transmit Data Register

The I2C0DOUT register holds the serial data that is to be transmitted on the I^2C bus. When the I2CSLAVE0 is operating in CPU mode for I^2C Read operations, software may safely write to this register when the I2C0INT flag is set. It is not safe to write to this register in any one of the following conditions:

- The I2C0INT flag is cleared to logic 0
- The I2CSLAVE0 is operating in DMA mode for I^2C Read operations

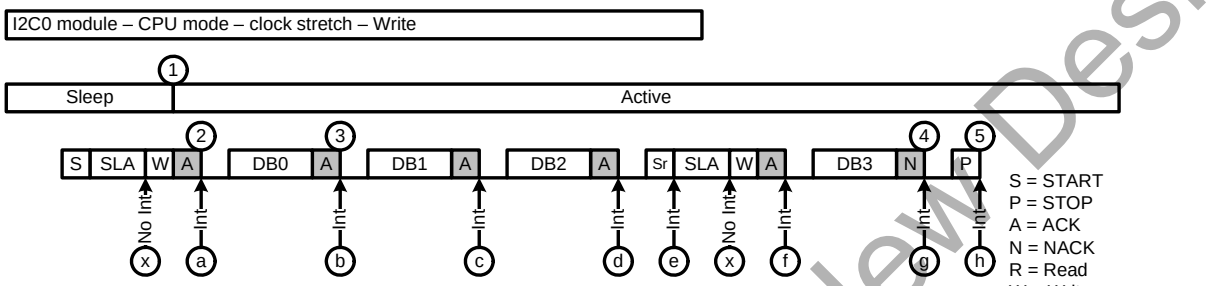
When the I2CSLAVE0 is operating in DMA mode for I^2C Read operations, software should setup the data to be transmitted in XRAM and configure the DMA to transfer the data from XRAM to the I2C0DOUT register.

30.5. I2C Transfer Modes

The I2CSLAVE0 interface may be operating in either I²C Write or I²C Read mode. Data transfers can also be controlled by DMA, depending on whether a DMA channel has selected I²C Read or Write as a data transfer function. The following sub-sections describe in detail the setting and clearing of various status bits in the I2C0STAT register during different modes of operations. In all modes, the I2CSLAVE0 peripheral performs clock stretching automatically on every SCL falling edge associated with the ACK or NACK bit.

30.5.1. I2C Write Sequence (CPU mode)

Figure 30.5 shows the details of how the I2C0STAT status bits change during an I²C Write data transfer.



	1	SLA+W wakes CPU from sleep mode. Automatic ACK if BUSY was 0.
	2	INT generated – starting clock stretch. CPU clears I2C0INT, which releases SCL
	3	CPU fetches DB0 from I2C0DIN; ACK bit cleared to '0'; CPU clears I2C0INT, which releases SCL
	4	Slave doesn't care whether it generates NACK or ACK. State machine response is the same for both.
	5	Stop bit (P) generates interrupt. No Clock Stretch. CPU clears I2C0INT
No int	x	I2C0STAT = x1001000 at 8th SCL rising edge
I2C0 int	a	I2C0STAT = x1101010; CPU clears START and I2C0INT
I2C0 int	b	I2C0STAT = x1100010; CPU fetches DB0 from I2C0DIN and clears I2C0INT
I2C0 int	c	I2C0STAT = x1100010; CPU fetches DB1 from I2C0DIN and clears I2C0INT
I2C0 int	d	I2C0STAT = x1100010; CPU fetches DB2 from I2C0DIN and clears I2C0INT
I2C0 int	e	I2C0STAT = x0100000; CPU services end of transaction and clears I2C0INT
I2C0 int	f	I2C0STAT = x1101010; CPU clears START and I2C0INT; CPU writes '1' to BUSY to NACK future data transfers
I2C0 int	g	I2C0STAT = x1110010; CPU fetches DB3 from I2C0DIN and clears I2C0INT, NACK; CPU writes '0' to BUSY to ACK future data transfers
I2C0 int	h	I2C0STAT = x0100100; CPU clears STOP and I2C0INT

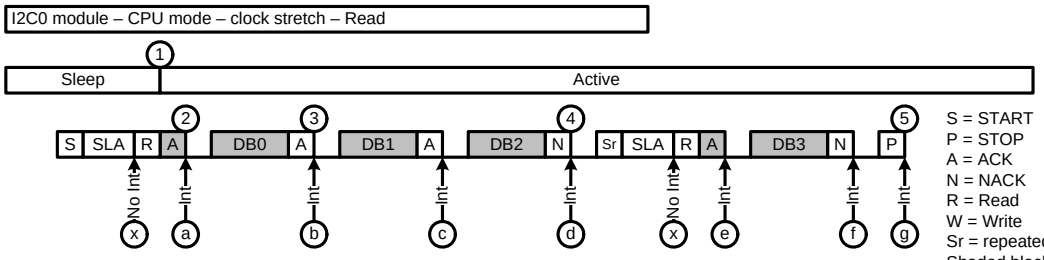
* At a, b, c, d, f, g : Bits are set/cleared at 9th SCL falling edge. CPU clears I2C0INT to release SCL

Figure 30.5. Typical I²C Write Sequence in CPU Mode

Note that at "f" in the above sequence, it is possible to leave the BUSY bit at 0. In this case, the master will receive an ACK instead at "g" and it would still be possible for the I²C master to generate a STOP bit immediately after the ACK.

30.5.2. I²C Read Sequence (CPU mode)

Figure 30.6 shows the details of how the I2C0STAT status bits change during an I²C Read data transfer.



1. SLA+R wakes CPU from sleep. SLA is ACK if BUSY was 0.
2. INT generated. CPU writes DB0 into I2C0DOUT. SCL is held until CPU clears I2C0INT. CPU clears I2C0INT, releasing SCL
3. SCL held low; NACK bit reads '0'; CPU writes DB1 into I2C0DOUT and clears I2C0INT
4. NACK switches I2C slave to IDLE state and generates INT. Any following DB are ignored. If RpStart follows after NACK, the START sticky bit will be set, but no interrupt.
5. STOP generates interrupt. No Clock Stretch. CPU clears I2C0INT

No int	x	I2C0STAT = x1001000 at 8th SCL rising edge
I2C0 int	a	I2C0STAT = x1101001; CPU clears START. CPU writes DB0 into I2C0DOUT and clears I2C0INT
I2C0 int	b	I2C0STAT = x1100001; CPU writes DB1 into I2C0DOUT and clears I2C0INT
I2C0 int	c	I2C0STAT = x1100001; CPU writes DB2 into I2C0DOUT and clears I2C0INT
I2C0 int	d	I2C0STAT = x0110001; CPU prepares for STOP or RpStart and clears I2C0INT and clears NACK
I2C0 int	e	I2C0STAT = x1101001; CPU clears START. CPU writes DB3 into I2C0DOUT and clears I2C0INT
I2C0 int	f	I2C0STAT = x0110001; CPU prepares for STOP or RpStart and clears I2C0INT and clears NACK
I2C0 int	g	I2C0STAT = x0100100; CPU clears STOP and clears I2C0INT

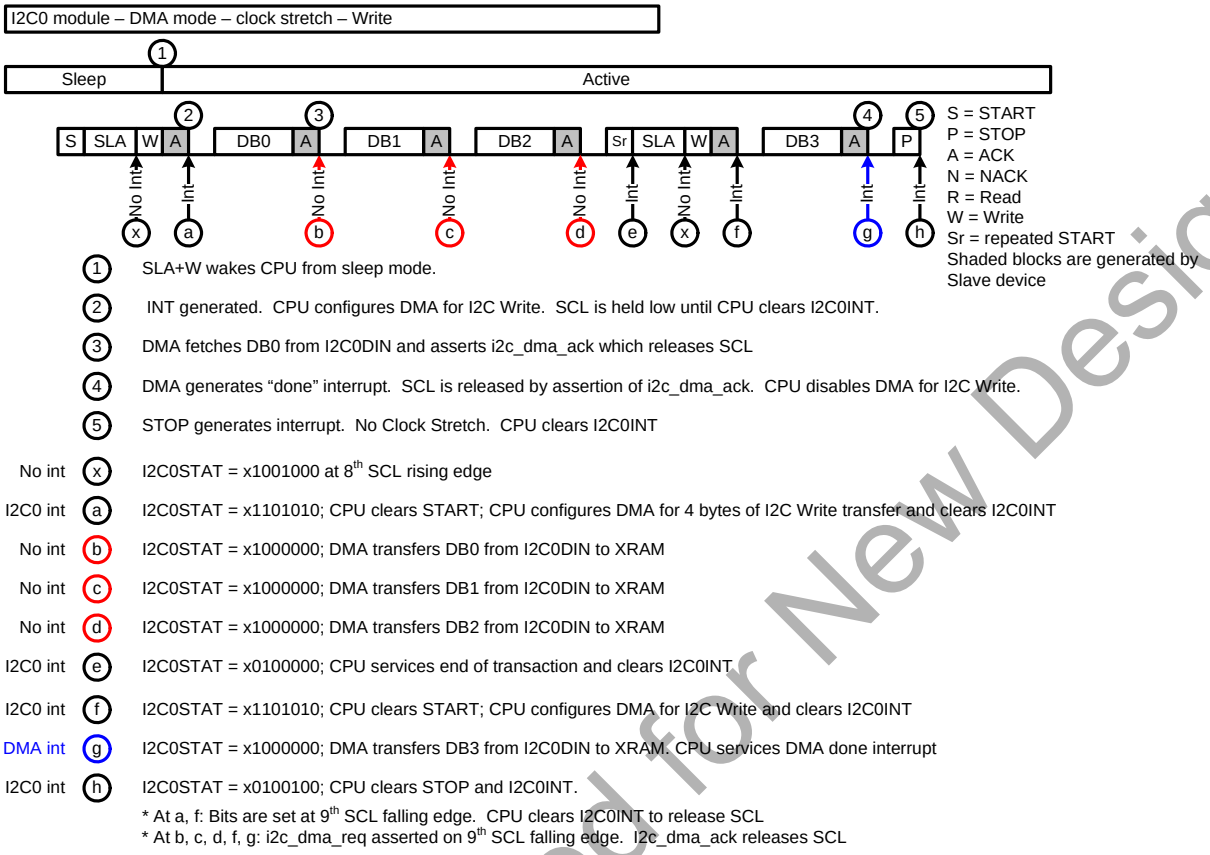
* At a, b, c, d, e, f : Bits are set/cleared at 9th SCL falling edge. CPU clears I2C0INT to release SCL

Figure 30.6. Typical I2C Read Sequence in CPU Mode

Note that the I²C Master MUST always generate a NACK before it can generate a repeated START bit or a STOP bit. This is because the NACK will cause the I²C Slave to release the SDA line for the I²C Master to generate the START or STOP bit.

30.5.3. I²C Write Sequence (DMA mode)

Figure 30.7 shows the details of how the I2C0STAT status bits change during an I²C Write data transfer.

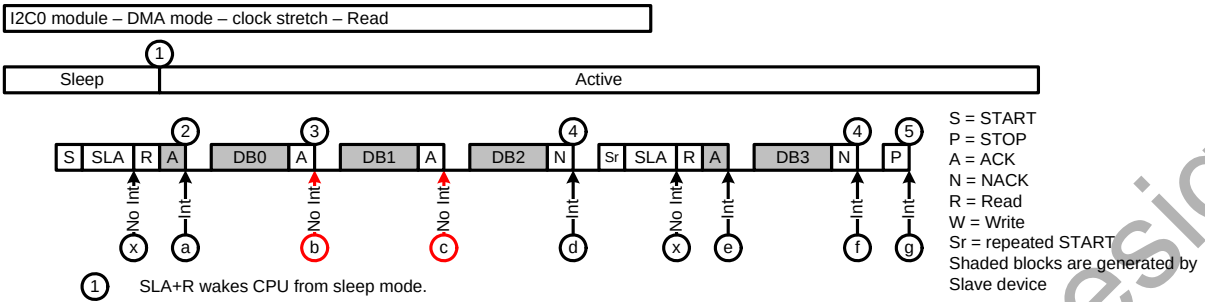


	1	SLA+W wakes CPU from sleep mode.
	2	INT generated. CPU configures DMA for I2C Write. SCL is held low until CPU clears I2C0INT.
	3	DMA fetches DB0 from I2C0DIN and asserts i2c_dma_ack which releases SCL
	4	DMA generates “done” interrupt. SCL is released by assertion of i2c_dma_ack. CPU disables DMA for I2C Write.
	5	STOP generates interrupt. No Clock Stretch. CPU clears I2C0INT
No int	x	I2C0STAT = x1001000 at 8th SCL rising edge
I2C0 int	a	I2C0STAT = x1101010; CPU clears START; CPU configures DMA for 4 bytes of I2C Write transfer and clears I2C0INT
No int	b	I2C0STAT = x1000000; DMA transfers DB0 from I2C0DIN to XRAM
No int	c	I2C0STAT = x1000000; DMA transfers DB1 from I2C0DIN to XRAM
No int	d	I2C0STAT = x1000000; DMA transfers DB2 from I2C0DIN to XRAM
I2C0 int	e	I2C0STAT = x0100000; CPU services end of transaction and clears I2C0INT
I2C0 int	f	I2C0STAT = x1101010; CPU clears START; CPU configures DMA for I2C Write and clears I2C0INT
DMA int	g	I2C0STAT = x1000000; DMA transfers DB3 from I2C0DIN to XRAM. CPU services DMA done interrupt
I2C0 int	h	I2C0STAT = x0100100; CPU clears STOP and I2C0INT.

* At a, f: Bits are set at 9th SCL falling edge. CPU clears I2C0INT to release SCL
* At b, c, d, f, g: i2c_dma_req asserted on 9th SCL falling edge. I2c_dma_ack releases SCL

Figure 30.7. Typical I²C Write Sequence in DMA Mode

30.5.4. I²C Read Sequence (DMA Mode)



	1	SLA+R wakes CPU from sleep mode.
	2	INT generated. CPU configures DMA for I2C Read. CPU performs SW DMA request to transfer DB0 from XRAM to I2C0DOUT. SCL is held until CPU clears I2C0INT.
	3	DMA transfers DB1 from XRAM to I2C0DOUT and asserts i2c_dma_ack which releases SCL
	4	NACK switches I2C slave into IDLE state. Any following DB are ignored. If RpStart follows NACK, the START sticky bit will be set, but no interrupt.
	5	STOP generates interrupt. No Clock Stretch. CPU clears I2C0INT
No int	x	I2C0INT = x1001000 at 8th SCL rising edge
I2C0 int	a	I2C0INT = x1101001; CPU clears START. CPU configures DMA for I2C Read; CPU starts SW DMA to transfer DB0 from XRAM to I2CDOUT. CPU clears I2C0INT
No int	b	I2C0STAT = x1000000; DMA transfers DB1 from XRAM to I2C0DOUT
No int	c	I2C0STAT = x1000000; DMA transfers DB2 from XRAM to I2C0DOUT
I2C0 int	d	I2C0STAT = x0110001; NACK causes I2C to not generate i2c_dma_req. CPU clears I2C0INT, NACK. CPU disables DMA.
I2C0 int	e	I2C0INT = x1101001; CPU clears START. CPU configures DMA for I2C Read; CPU starts SW DMA to transfer DB3 from XRAM to I2CDOUT. CPU clears I2C0INT
I2C0 int	f	I2C0STAT = x0110001; NACK causes I2C to not generate i2c_dma_req. CPU clears I2C0INT, NACK. CPU disables DMA.
I2C0 int	g	I2C0STAT = x0100100; CPU clears STOP and clears I2C0INT

* At a, d, e, f: Bits are set at 9th SCL falling edge. SCL released when CPU clear I2C0INT
* At b, c: i2c_dma_req asserted at 9th SCL falling edge. SCL released by i2c_dma_ack

Figure 30.8. Typical I²C Read Sequence in DMA Mode

30.6. I2CSLAVE0 Slave Registers

Register 30.1. I2C0DIN: I2C0 Received Data

Bit	7	6	5	4	3	2	1	0
Name	I2C0DIN							
Type	R							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xA5								

Bit	Name	Function
7:0	I2C0DIN	**I2C0 Received Data.** This field is the data byte received from the I2C bus during a write operation.

Register 30.2. I2C0DOUT: I2C0 Transmit Data

Bit	7	6	5	4	3	2	1	0
Name	I2C0DOUT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xA4								

Bit	Name	Function
7:0	I2C0DOUT	**I2C0 Transmit Data.** This field is the data byte to transmit to the I2C bus during a read operation.

Register 30.3. I2C0SLAD: I2C0 Slave Address

Bit	7	6	5	4	3	2	1	0
Name	Reserved	I2C0SLAD						
Type	RW	RW						
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xAD								

Bit	Name	Function
7	Reserved	Must write reset value.
6:0	I2C0SLAD	**I2C Hardware Slave Address.** This field defines the I2C0 Slave Address for automatic hardware acknowledgement. When the received I2C address matches this field, hardware sets the I2C0INT bit in the I2C0STAT register.

Register 30.4. I2C0STAT: I2C0 Status

Bit	7	6	5	4	3	2	1	0
Name	HSMODE	ACTIVE	I2C0INT	NACK	START	STOP	WR	RD
Type	R	R	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0xF; SFR Address: 0xF8 (bit-addressable)								

Bit	Name	Function
7	HSMODE	**High Speed Mode.** This bit is set to 1 by hardware when a High Speed master code is received received and automatically clears when a STOP event occurs.
6	ACTIVE	**Bus Active.** This bit is set to 1 by hardware when an incoming slave address matches and automatically clears when the transfer completes with either a STOP or a NACK event.
5	I2C0INT	**I2C Interrupt.** This bit is set when a read (RD), write (WR), or a STOP event (STOP) occurs. This bit will also set when the ACTIVE bit goes low to indicate the end of a transfer. This bit will generate an interrupt, and hardware will automatically clear this bit after the RD and WR bits clear.
4	NACK	**NACK.** This bit is set by hardware when one of the following conditions are met: - A NACK is transmitted by either a Master or a Slave when the ACTIVE bit is high. - An I2C slave transmits a NACK to a matching slave address. Hardware will automatically clear this bit.
3	START	**Start.** This bit is set by hardware when a START is received and a matching slave address is received. Hardware will automatically clear this bit.
2	STOP	**Stop.** This bit is set by hardware when a STOP is received and the last slave address received matches the value in the I2C0SLAD register. Hardware will automatically clear this bit.
1	WR	**I2C Write.** This bit is set by hardware on the 9th SCL falling edge when one of the following conditions are met: - The I2C0 Slave responds with an ACK, and the DMA has not enabled I2C Write as a data transfer function. - The I2C0 Slave responds with a NACK, and the DMA has not enabled I2C Write as a data transfer function. - The current byte transaction has a matching I2C0 Slave address and the 8th bit was a WRITE bit (0). This bit will set the I2C0INT bit and generate an interrupt, if enabled. Hardware will automatically clear this bit.

Bit	Name	Function
0	RD	**I2C Read.** This bit is set by hardware on the 9th SCL falling edge when one of the following conditions are met: - The I2C Master responds with an ACK, and the DMA has not enabled I2C0 Slave Read as a data transfer function. - I2C Master responds with a NACK. - The current byte transaction has a matching I2C slave address and the 8th bit was a READ bit (1). This bit will set the I2C0INT bit and generate an interrupt, if enabled. Hardware will automatically clear this bit.

Register 30.5. I2C0CN: I2C0 Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved		PUEN	PINMD	TIMEOUT	PRELOAD	I2C0EN	BUSY
Type	R		RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	1	0	1
SFR Page = 0xF; SFR Address: 0xAC								

Bit	Name	Function
7:6	Reserved	Must write reset value.
5	PUEN	**I2C Pull-Up Enable.** 0: Disable internal pull-up resistors for the I2C0 Slave SCL and SDA pins. 1: Enable internal pull-up resistors for the I2C0 Slave SCL and SDA pins.
4	PINMD	**Pin Mode Enable.** 0: Set the I2C0 Slave pins in GPIO mode. 1: Set the I2C0 Slave pins in I2C mode.
3	TIMEOUT	**Timeout Enable.** When this bit is set, Timer 3 will start counting only when SCL is low. When SCL is high, Timer 3 will auto-reload with the value from the reload registers. Timer 3 must be configured for 16-bit auto-reload mode. 0: Disable I2C SCL timeout detection using Timer 3. 1: Enable I2C SCL timeout detection using Timer 3.
2	PRELOAD	**Preload Disable.** 0: Data bytes must be written into the I2C0DOUT register before the 8th SCL clock of the matching slave address byte transfer arrives for an I2C read operation. 1: Data bytes need not be preloaded for I2C read operations. The data byte can be written to I2C0DOUT during interrupt servicing or by the DMA.
1	I2C0EN	**I2C Enable.** This bit enables the I2C0 Slave module. PINMD must be enabled first before this bit is enabled. 0: Disable the I2C0 Slave module. 1: Enable the I2C0 Slave module.
0	BUSY	**Busy.** 0: Device will acknowledge an I2C master. 1: Device will not respond to an I2C master. All I2C data sent to the device will be NACKed.

31. Universal Asynchronous Receiver/Transmitter (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 “31.1. Enhanced Baud Rate Generation” on page 380). 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 (TI is set in SCON0), or a data byte has been received (RI 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).

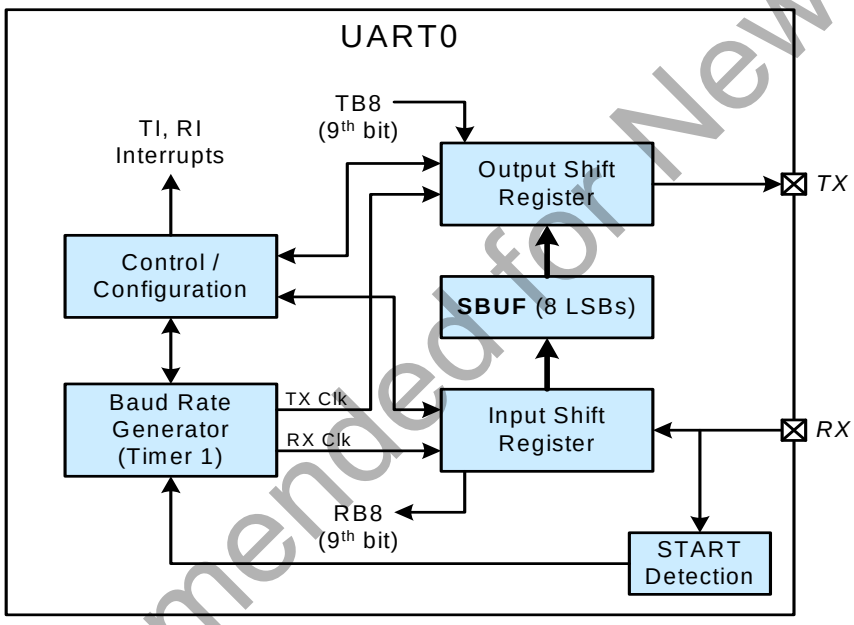


Figure 31.1. UART0 Block Diagram

31.1. 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 31.2), 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.

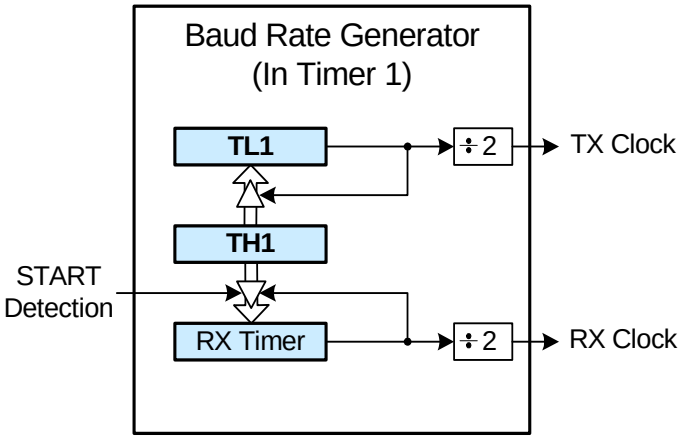


Figure 31.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload. 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 overflow rate, the UART0 baud rate is determined by Equation 31.1.

$$\text{UartBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

Equation 31.1. UART0 Baud Rate

Timer 1 overflow rate is selected as described in the Timer section. A quick reference for typical baud rates and system clock frequencies is given in Table 31.1.

31.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit in register SCON.

31.2.1. 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 TX pin and received at the RX pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB8 in the SCON register.

Data transmission begins when software writes a data byte to the SBUF0 register. The TI Transmit Interrupt Flag is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit 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: RI must be logic 0, and if MCE 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 RB8 and the RI flag is set. If these conditions are not met, SBUF0 and RB8 will not be loaded and the RI flag will not be set. An interrupt will occur if enabled when either TI or RI is set.

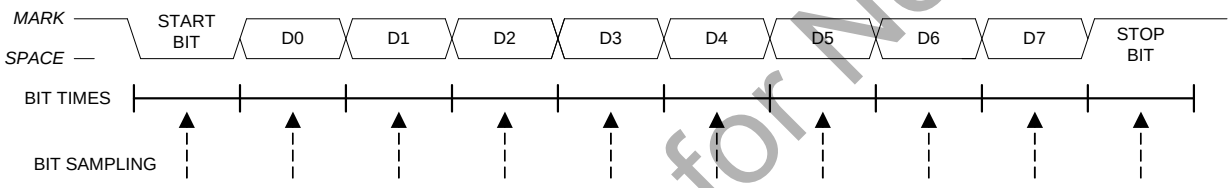


Figure 31.3. 8-Bit UART Timing Diagram

31.2.2. 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 TB8, 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 RB8 and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI Transmit Interrupt Flag is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN Receive Enable bit 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) RI must be logic 0, and (2) if MCE is logic 1, the 9th bit must be logic 1 (when MCE 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 RB8, and the RI flag is set to 1. If the above conditions are not met, SBUF0 and RB8 will not be loaded and the RI flag will not be set to 1. A UART0 interrupt will occur if enabled when either TI or RI is set to 1.

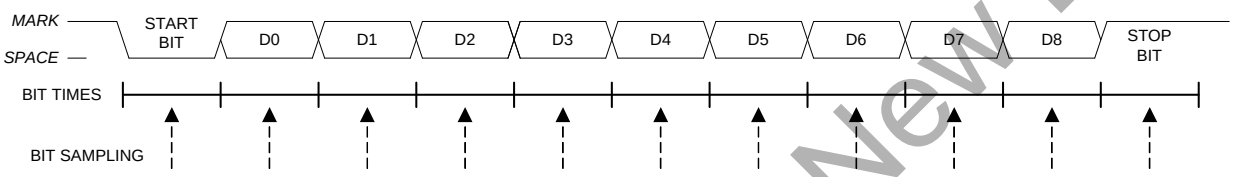


Figure 31.4. 9-Bit UART Timing Diagram

31.3. 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 MCE bit 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 (RB8 = 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 MCE bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE 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 MCE 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).

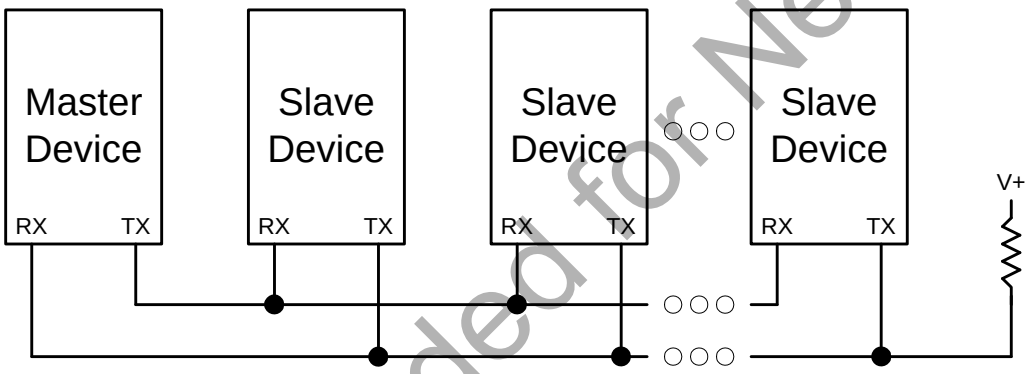


Figure 31.5. UART Multi-Processor Mode Interconnect Diagram

Table 31.1. Timer Settings for Standard Baud Rates Using The Internal 24.5 MHz Oscillator

	Frequency: 49 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscillator Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)[1]	T1M[1]	Timer 1 Reload Value (hex)
SYSCLK from Internal Osc.	230400	–0.32%	106	SYSCLK	XX[2]	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK/4	01	0	0x96
	14400	0.15%	1704	SYSCLK/12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK/12	00	0	0x96
	2400	–0.32%	10176	SYSCLK/48	10	0	0x96
	1200	0.15%	20448	SYSCLK/48	10	0	0x2B

Notes:
1. SCA1–SCA0 and T1M bit definitions can be found in Timer1 chapter.
2. X = Don't care.

31.4. UART Control Registers

Register 31.1. SCON0: UART0 Serial Port Control

Bit	7	6	5	4	3	2	1	0
Name	SMODE	Reserved	MCE	REN	TB8	RB8	TI	RI
Type	RW	R	RW	RW	RW	RW	RW	RW
Reset	0	1	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0x98 (bit-addressable)								

Table 31.2. SCON0 Register Bit Descriptions

Bit	Name	Function
7	SMODE	**Serial Port 0 Operation Mode.** Selects the UART0 Operation Mode. 0: 8-bit UART with Variable Baud Rate (Mode 0). 1: 9-bit UART with Variable Baud Rate (Mode 1).
6	Reserved	Must write reset value.
5	MCE	**Multiprocessor Communication Enable.** This bit enables checking of the stop bit or the 9th bit in multi-drop communication buses. The function of this bit is dependent on the UART0 operation mode selected by the SMODE bit. In Mode 0 (8-bits), the peripheral will check that the stop bit is logic 1. In Mode 1 (9-bits) the peripheral will check for a logic 1 on the 9th bit. 0: Ignore level of 9th bit / Stop bit. 1: RI is set and an interrupt is generated only when the stop bit is logic 1 (Mode 0) or when the 9th bit is logic 1 (Mode 1).
4	REN	**Receive Enable.** 0: UART0 reception disabled. 1: UART0 reception enabled.
3	TB8	**Ninth Transmission Bit.** The logic level of this bit will be sent as the ninth transmission bit in 9-bit UART Mode (Mode 1). Unused in 8-bit mode (Mode 0).
2	RB8	**Ninth Receive Bit.** RB8 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.
1	TI	**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 firmware.

Table 31.2. SCON0 Register Bit Descriptions

Bit	Name	Function
0	RI	**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 firmware.

Register 31.2. SBUF0: UART0 Serial Port Data Buffer

Bit	7	6	5	4	3	2	1	0
Name	SBUF0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x99								

Table 31.3. SBUF0 Register Bit Descriptions

Bit	Name	Function
7:0	SBUF0	**Serial Data Buffer.** 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.

32. Timers (Timer0, Timer1, Timer2, and Timer3)

Each MCU in the C8051F97x family 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 timers for timing peripherals 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 are also identical and offer both 16-bit and split 8-bit timer functionality with auto-reload capabilities. Timer 2 and Timer 3 both offer a capture function, but are different in their system-level connections. Timer 2 is capable of performing a capture function on an external signal input routed through the crossbar, while the Timer 3 capture is dedicated to the low-frequency oscillator output. Table 32.1 summarizes the modes available to each timer.

Table 32.1. Timer Modes

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	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
8-bit counter/timer 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 prescaled clock from which Timer 0 and/or Timer 1 may be clocked.

Timer 0/1 may then be configured to use this prescaled clock signal or the system clock. Timer 2 and Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock frequency can be counted. The input signal need not be periodic, but it must be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

All four timers are capable of clocking other peripherals and triggering events in the system. The individual peripherals select which timer to use for their respective functions. Table 32.2 summarizes the peripheral connections for each timer. Note that the Timer 2 and Timer 3 high overflows apply to the full timer when operating in 16-bit mode or the high-byte timer when operating in 8-bit split mode.

Table 32.2. Timer Peripheral Clocking / Event Triggering

Function	T0 Overflow	T1 Overflow	T2 High Overflow	T2 Low Overflow	T3 High Overflow	T3 Low Overflow
UART0 Baud Rate		X				
SMBus0 Clock Rate	X	X	X	X		
SMBus0 SCL Low Timeout					X	
PCA0 Clock	X					
ADC0 Conversion Start	X		X*	X*	X*	X*

***Note:** The high-side overflow is used when the timer is in16-bit mode. The low-side overflow is used in 8-bit mode.

32.1. Timer 0 and Timer 1

Timer 0 and Timer 1 are each implemented as a16-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. Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register. 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 for the operating modes described below.

32.1.1. 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 in TCON is set and an interrupt will occur if Timer 0 interrupts are enabled.

The CT0 bit in the TMOD register selects the counter/timer's clock source. When CT0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register. Clearing CT selects the clock defined by the T0M bit in register CKCON. 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.

Setting the TR0 bit enables the timer when either GATE0 in the TMOD register is logic 0 or the input signal $\overline{\text{INT0}}$ is active as defined by bit IN0PL in register IT01CF. Setting GATE0 to 1 allows the timer to be controlled by the external input signal INT0, facilitating pulse width measurements

TR0	GATE0	$\overline{\text{INT0}}$	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
Note: X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal INT1 is used with Timer 1; the /INT1 polarity is defined by bit IN1PL in register IT01CF.

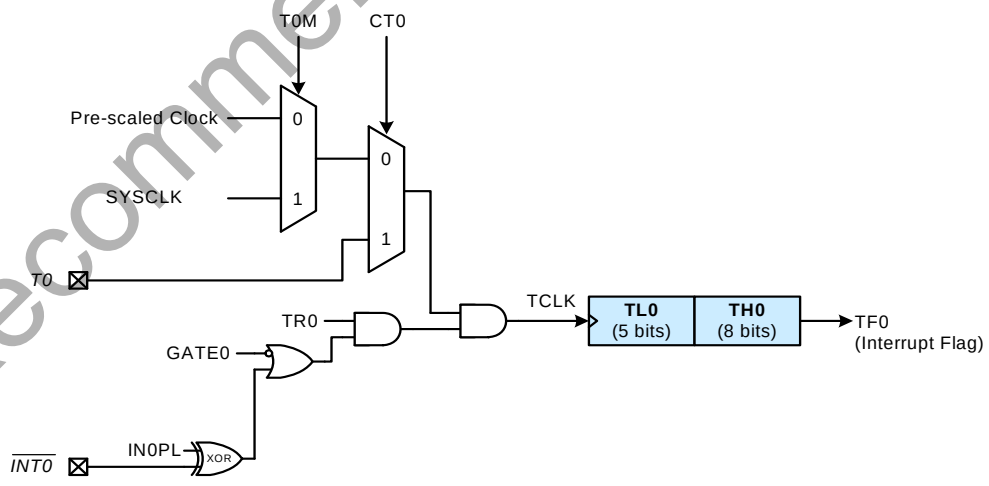


Figure 32.1. T0 Mode 0 Block Diagram

32.1.2. 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.

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

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

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

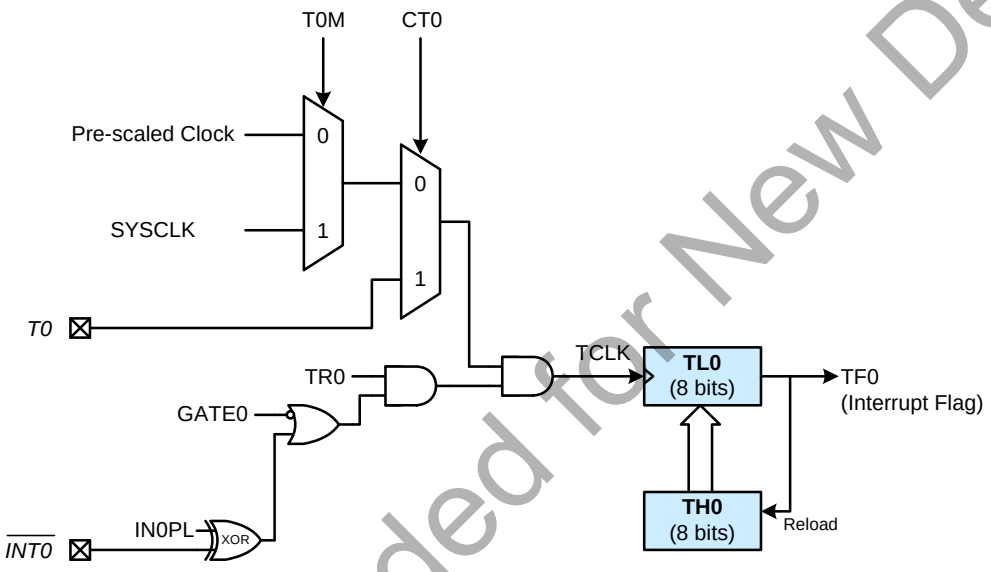


Figure 32.2. T0 Mode 2 Block Diagram

32.1.4. 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, CT0, 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.

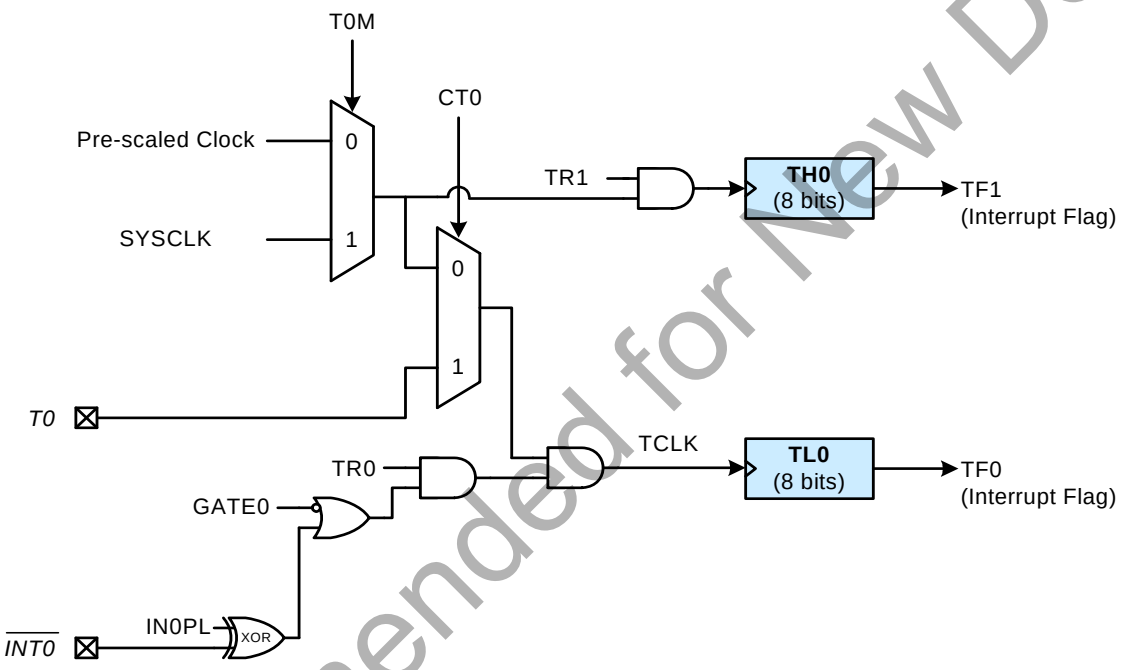


Figure 32.3. T0 Mode 3 Block Diagram

32.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit in TMR2CN defines the Timer 2 operation mode. Timer 2 can also be used in Capture Mode to measure the SmaRTClock or the Comparator 0 period with respect to another oscillator. The ability to measure the Comparator 0 period with respect to the system clock makes using Touch Sense switches very easy.

Timer 2 may be clocked by the system clock, the system clock divided by 12, SmaRTClock divided by 8, or Comparator 0 output. Note that the SmaRTClock divided by 8 and Comparator 0 output is synchronized with the system clock.

32.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT in the TMR2CN register is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, SmaRTClock divided by 8, or Comparator 0 output. 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 32.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LINT bit is set, an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

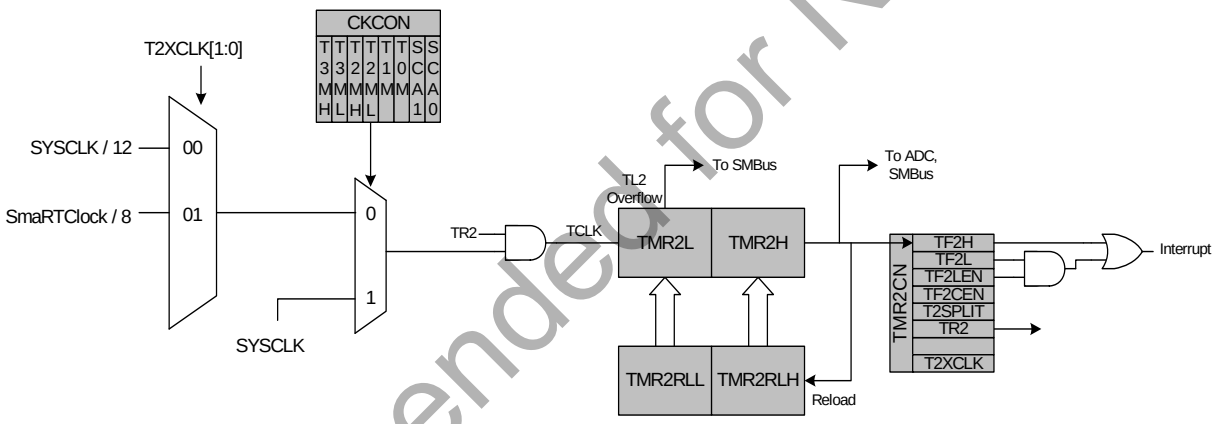


Figure 32.4. Timer 2 16-Bit Mode Block Diagram

32.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 32.5. 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, SmaRTClock divided by 8 or Comparator 0 output. 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 bits (T2XCLK[1:0] in TMR2CN), as follows:

T2MH	T2XCLK[1:0]	TMR2H Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock / 8
0	10	Reserved
0	11	Comparator 0
1	X	SYSCLK

T2ML	T2XCLK[1:0]	TMR2L Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock / 8
0	10	Reserved
0	11	Comparator 0
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LINT (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LINT 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.

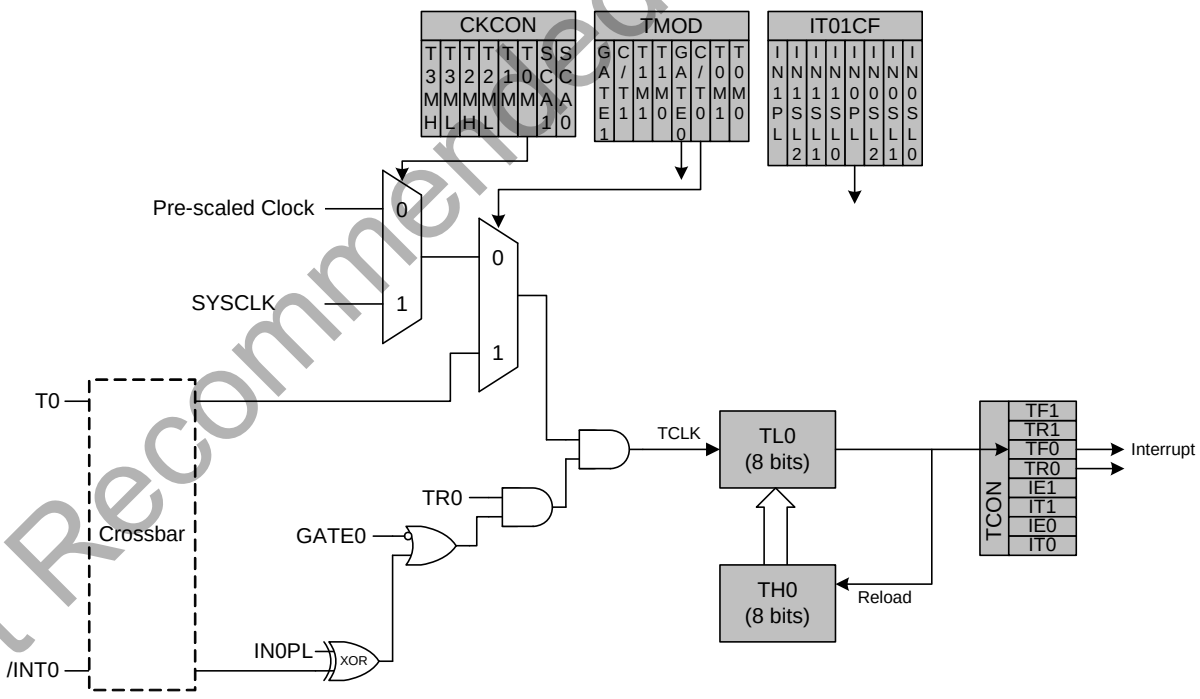


Figure 32.5. Timer 2 8-Bit Mode Block Diagram

32.2.3. Comparator 0/SmaRTClock Capture Mode

The Capture Mode in Timer 2 allows either Comparator 0 or the SmaRTClock period to be measured against the system clock or the system clock divided by 12. Comparator 0 and the SmaRTClock period can also be compared against each other. Timer 2 Capture Mode is enabled by setting TF2CINT to 1. Timer 2 should be in 16-bit auto-reload mode when using Capture Mode.

When Capture Mode is enabled, a capture event will be generated either every Comparator 0 rising edge or every 8 SmaRTClock clock cycles, depending on the T2XCLK1 setting. When the capture event occurs, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set (triggering an interrupt if Timer 2 interrupts are enabled). By recording the difference between two successive timer capture values, the Comparator 0 or SmaRTClock period can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading.

For example, if T2ML = 1b, T2XCLK1 = 0b, and TF2CINT = 1b, Timer 2 will clock every SYSCLK and capture every SmaRTClock clock divided by 8. If the SYSCLK is 24.5 MHz and the difference between two successive captures is 5984, then the SmaRTClock clock is as follows:

24.5 MHz/(5984/8) = 0.032754 MHz or 32.754 kHz.

This mode allows software to determine the exact SmaRTClock frequency in self-oscillate mode and the time between consecutive Comparator 0 rising edges, which is useful for detecting changes in the capacitance of a Touch Sense Switch.

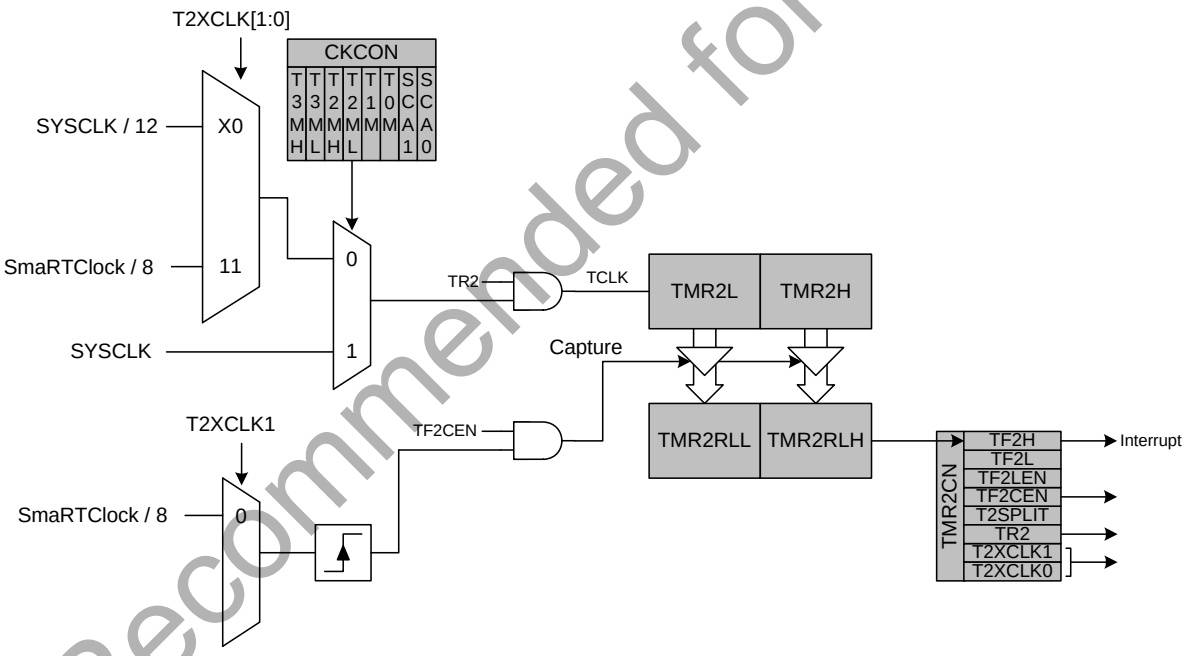


Figure 32.6. Timer 2 Capture Mode Block Diagram

32.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit in the TMR3CN register defines the Timer 3 operation mode. Timer 3 can also be used in Capture Mode to measure the external oscillator source or the SmaRTClock oscillator period with respect to another oscillator.

Timer 3 may be clocked by the system clock, the system clock divided by 12, external oscillator source divided by 8, or the SmaRTClock oscillator. The external oscillator source divided by 8 and SmaRTClock oscillator is synchronized with the system clock.

32.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT in the TMR3CN register is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, external oscillator clock source divided by 8, or SmaRTClock oscillator. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 32.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled (if EIE1.7 is set), an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LINT bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

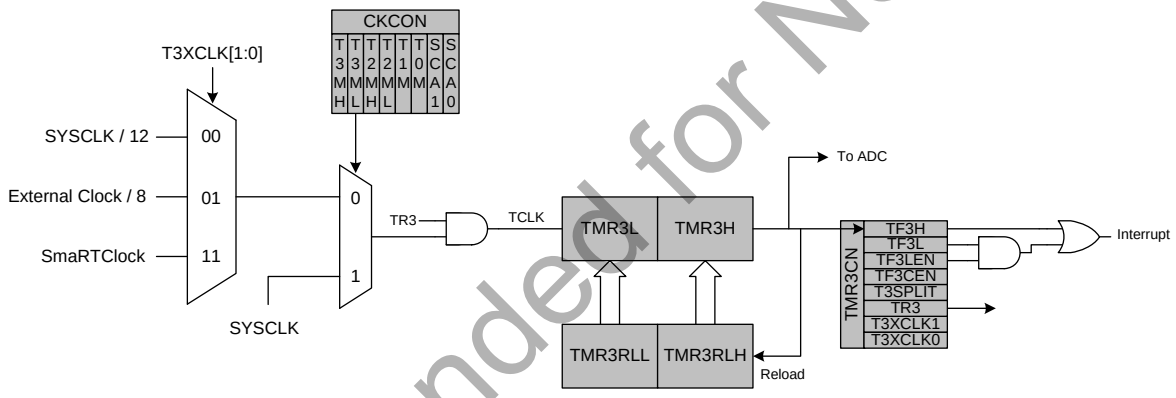


Figure 32.7. Timer 3 16-Bit Mode Block Diagram

32.3.2. 8-Bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 32.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, the external oscillator clock source divided by 8, or the SmaRTClock. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bits (T3XCLK[1:0] in TMR3CN), as follows:

T3MH	T3XCLK[1:0]	TMR3H Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock
0	10	Reserved
0	11	External Clock / 8
1	X	SYSCLK

T3ML	T3XCLK[1:0]	TMR3L Clock Source
0	00	SYSCLK / 12
0	01	SmaRTClock
0	10	Reserved
0	11	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 TF3LINT (TMR3CN.5) is set, an interrupt is generated each time either TMR3L or TMR3H overflows. When TF3LINT 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.

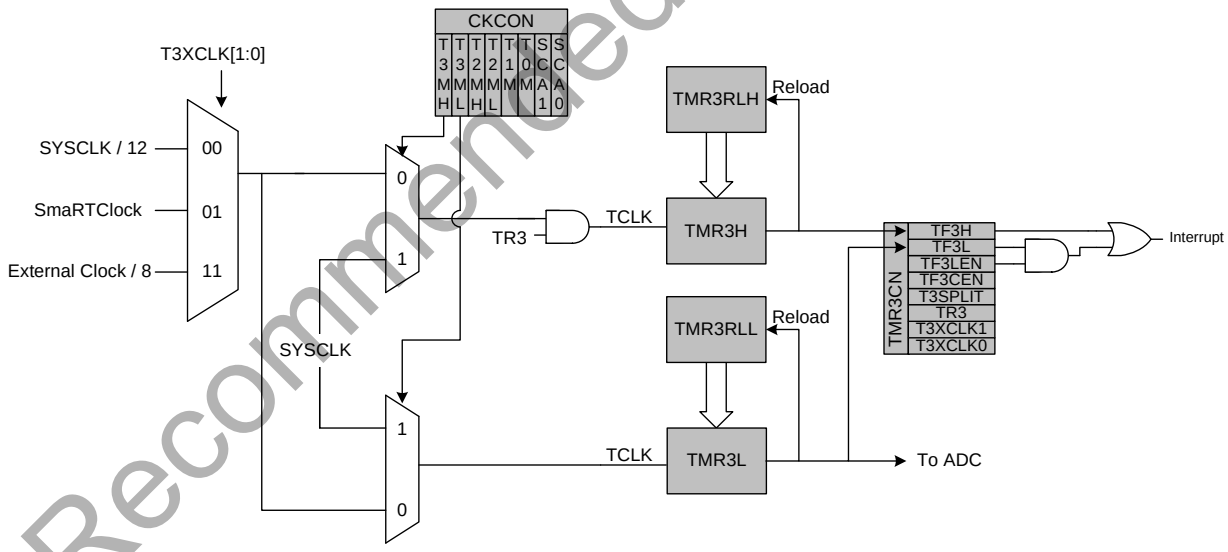


Figure 32.8. Timer 3 8-Bit Mode Block Diagram

32.3.3. SmaRTClock/External Oscillator Capture Mode

The Capture Mode in Timer 3 allows either SmaRTClock or the external oscillator period to be measured against the system clock or the system clock divided by 12. SmaRTClock and the external oscillator period can also be compared against each other.

Setting TF3CINT to 1 enables the SmaRTClock/External Oscillator Capture Mode for Timer 3. In this mode, T3SPLIT should be set to 0, as the full 16-bit timer is used.

When Capture Mode is enabled, a capture event will be generated either every SmaRTClock rising edge or every 8 external clock cycles, depending on the T3XCLK1 setting. When the capture event occurs, the contents of Timer 3 (TMR3H:TMR3L) are loaded into the Timer 3 reload registers (TMR3RLH:TMR3RLL) and the TF3H flag is set (triggering an interrupt if Timer 3 interrupts are enabled). By recording the difference between two successive timer capture values, the SmaRTClock or external clock period can be determined with respect to the Timer 3 clock. The Timer 3 clock should be much faster than the capture clock to achieve an accurate reading.

For example, if T3ML = 1b, T3XCLK1 = 0b, and TF3CINT = 1b, Timer 3 will clock every SYSCLK and capture every SmaRTClock rising edge. If SYSCLK is 24.5 MHz and the difference between two successive captures is 350 counts, then the SmaRTClock period is as follows:

350 x (1 / 24.5 MHz) = 14.2 µs.

This mode allows software to determine the exact frequency of the external oscillator in C and RC mode or the time between consecutive SmaRTClock rising edges, which is useful for determining the SmaRTClock frequency.

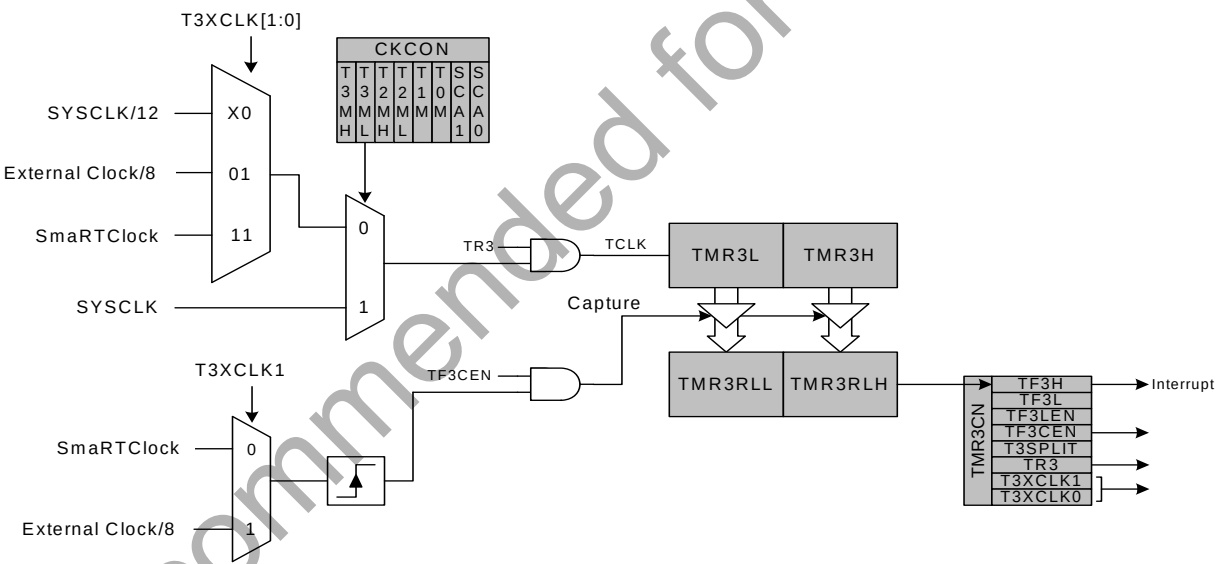


Figure 32.9. Timer 3 Capture Mode Block Diagram

32.4. Timer Control Registers

Register 32.1. CKCON: Clock Control

Bit	7	6	5	4	3	2	1	0
Name	T3MH	T3ML	T2MH	T2ML	T1M	T0M	SCA	
Type	RW	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x8E								

Table 32.3. CKCON Register Bit Descriptions

Bit	Name	Function
7	T3MH	**Timer 3 High Byte Clock Select.** Selects the clock supplied to the Timer 3 high byte (split 8-bit timer mode only). 0: Timer 3 high byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 high byte uses the system clock.
6	T3ML	**Timer 3 Low Byte Clock Select.** Selects the clock supplied to Timer 3. Selects the clock supplied to the lower 8-bit timer in split 8-bit timer mode. 0: Timer 3 low byte uses the clock defined by the T3XCLK bit in TMR3CN. 1: Timer 3 low byte uses the system clock.
5	T2MH	**Timer 2 High Byte Clock Select.** Selects the clock supplied to the Timer 2 high byte (split 8-bit timer mode only). 0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN. 1: Timer 2 high byte uses the system clock.
4	T2ML	**Timer 2 Low Byte Clock Select.** 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.
3	T1M	**Timer 1 Clock Select.** Selects the clock source supplied to Timer 1. Ignored when C/T1 is set to 1. 0: Timer 1 uses the clock defined by the prescale field, SCA. 1: Timer 1 uses the system clock.
2	T0M	**Timer 0 Clock Select.** Selects the clock source supplied to Timer 0. Ignored when C/T0 is set to 1. 0: Counter/Timer 0 uses the clock defined by the prescale field, SCA. 1: Counter/Timer 0 uses the system clock.

Table 32.3. CKCON Register Bit Descriptions

Bit	Name	Function
1:0	SCA	**Timer 0/1 Prescale.** These bits control the Timer 0/1 Clock Prescaler: 00: System clock divided by 12. 01: System clock divided by 4. 10: System clock divided by 48. 11: External oscillator divided by 8 (synchronized with the system clock).

Register 32.2. TCON: Timer 0/1 Control

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x88 (bit-addressable)								

Table 32.4. TCON Register Bit Descriptions

Bit	Name	Function
7	TF1	**Timer 1 Overflow Flag.** Set to 1 by hardware when Timer 1 overflows. This flag can be cleared by firmware but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
6	TR1	**Timer 1 Run Control.** Timer 1 is enabled by setting this bit to 1.
5	TF0	**Timer 0 Overflow Flag.** Set to 1 by hardware when Timer 0 overflows. This flag can be cleared by firmware but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
4	TR0	**Timer 0 Run Control.** Timer 0 is enabled by setting this bit to 1.
3	IE1	**External Interrupt 1.** This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by firmware but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine in edge-triggered mode.
2	IT1	**Interrupt 1 Type Select.** This bit selects whether the configured $\overline{\text{INT1}}$ interrupt will be edge or level sensitive. $\overline{\text{INT1}}$ is configured active low or high by the IN1PL bit in register IT01CF. 0: $\overline{\text{INT1}}$ is level triggered. 1: $\overline{\text{INT1}}$ is edge triggered.
1	IE0	**External Interrupt 0.** This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by firmware but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine in edge-triggered mode.
0	IT0	**Interrupt 0 Type Select.** This bit selects whether the configured $\overline{\text{INT0}}$ interrupt will be edge or level sensitive. $\overline{\text{INT0}}$ is configured active low or high by the IN0PL bit in register IT01CF. 0: $\overline{\text{INT0}}$ is level triggered. 1: $\overline{\text{INT0}}$ is edge triggered.

Register 32.3. TMOD: Timer 0/1 Mode

Bit	7	6	5	4	3	2	1	0
Name	GATE1	CT1	T1M		GATE0	CT0	T0M	
Type	RW	RW	RW		RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x89								

Table 32.5. TMOD Register Bit Descriptions

Bit	Name	Function
7	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 IT01CF.
6	CT1	**Counter/Timer 1 Select.** 0: Timer Mode. Timer 1 increments on the clock defined by T1M in the CKCON register. 1: Counter Mode. Timer 1 increments on high-to-low transitions of an external pin (T1).
5:4	T1M	**Timer 1 Mode Select.** These bits select the Timer 1 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Timer 1 Inactive
3	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 IT01CF.
2	CT0	**Counter/Timer 0 Select.** 0: Timer Mode. Timer 0 increments on the clock defined by T0M in the CKCON register. 1: Counter Mode. Timer 0 increments on high-to-low transitions of an external pin (T0).
1:0	T0M	**Timer 0 Mode Select.** These bits select the Timer 0 operation mode. 00: Mode 0, 13-bit Counter/Timer 01: Mode 1, 16-bit Counter/Timer 10: Mode 2, 8-bit Counter/Timer with Auto-Reload 11: Mode 3, Two 8-bit Counter/Timers

32.5. Timer 0/1 Registers

Register 32.4. TL0: Timer 0 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x8A								

Table 32.6. TL0 Register Bit Descriptions

Bit	Name	Function
7:0	TL0	**Timer 0 Low Byte.** The TL0 register is the low byte of the 16-bit Timer 0.

Register 32.5. TL1: Timer 1 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TL1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x8B								

Table 32.7. TL1 Register Bit Descriptions

Bit	Name	Function
7:0	TL1	**Timer 1 Low Byte.** The TL1 register is the low byte of the 16-bit Timer 1.

Register 32.6. TH0: Timer 0 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x8C								

Table 32.8. TH0 Register Bit Descriptions

Bit	Name	Function
7:0	TH0	**Timer 0 High Byte.** The TH0 register is the high byte of the 16-bit Timer 0.

Register 32.7. TH1: Timer 1 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TH1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x8D								

Table 32.9. TH1 Register Bit Descriptions

Bit	Name	Function
7:0	TH1	**Timer 1 High Byte.** The TH1 register is the high byte of the 16-bit Timer 1.

32.6. Timer 2 Registers

Register 32.8. TMR2CN: Timer 2 Control

Bit	7	6	5	4	3	2	1	0
Name	TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	T2XCLK	
Type	RW	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = ALL; SFR Address: 0xC8 (bit-addressable)								

Table 32.10. TMR2CN Register Bit Descriptions

Bit	Name	Function
7	TF2H	**Timer 2 High Byte Overflow Flag.** Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16-bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. This bit must be cleared by firmware.
6	TF2L	**Timer 2 Low Byte Overflow Flag.** Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. TF2L will be set when the low byte overflows regardless of the Timer 2 mode. This bit must be cleared by firmware.
5	TF2LEN	**Timer 2 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 2 Low Byte interrupts. If Timer 2 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 2 overflows.
4	TF2CEN	**Timer 2 Capture Enable.** When set to 1, this bit enables Timer 2 Capture Mode. If TF2CEN is set and Timer 2 interrupts are enabled, an interrupt will be generated based on the selected input capture source, and the current 16-bit timer value in TMR2H:TMR2L will be copied to TMR2RLH:TMR2RLL.
3	T2SPLIT	**Timer 2 Split Mode Enable.** When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload. 0: Timer 2 operates in 16-bit auto-reload mode. 1: Timer 2 operates as two 8-bit auto-reload timers.
2	TR2	**Timer 2 Run Control.** Timer 2 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in split mode.

Table 32.10. TMR2CN Register Bit Descriptions

Bit	Name	Function
1:0	T2XCLK	**Timer 2 External Clock Select.** This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. Note: External clock sources are synchronized with the system clock. 00: External Clock is SYSCLK/12. Capture trigger is RTC/8. 01: Capture trigger is RTC/8. 10: External Clock is SYSCLK/12. 11: External Clock is RTC/8.

Register 32.9. TMR2RLL: Timer 2 Reload Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xCA								

Table 32.11. TMR2RLL Register Bit Descriptions

Bit	Name	Function
7:0	TMR2RLL	**Timer 2 Reload Low Byte.** When operating in one of the auto-reload modes, TMR2RLL holds the reload value for the low byte of Timer 2 (TMR2L). When operating in capture mode, TMR2RLL is the captured value of TMR2L.

Register 32.10. TMR2RLH: Timer 2 Reload High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2RLH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xCB								

Table 32.12. TMR2RLH Register Bit Descriptions

Bit	Name	Function
7:0	TMR2RLH	**Timer 2 Reload High Byte.** When operating in one of the auto-reload modes, TMR2RLH holds the reload value for the high byte of Timer 2 (TMR2H). When operating in capture mode, TMR2RLH is the captured value of TMR2H.

Register 32.11. TMR2L: Timer 2 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2L							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xCC								

Table 32.13. TMR2L Register Bit Descriptions

Bit	Name	Function
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.

Register 32.12. TMR2H: Timer 2 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR2H							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xCD								

Table 32.14. TMR2H Register Bit Descriptions

Bit	Name	Function
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.

32.7. Timer 3 Registers

Register 32.13. TMR3CN: Timer 3 Control

Bit	7	6	5	4	3	2	1	0
Name	TF3H	TF3L	TF3LEN	TF3CEN	T3SPLIT	TR3	T3XCLK	
Type	RW	RW	RW	RW	RW	RW	R	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x91								

Table 32.15. TMR3CN Register Bit Descriptions

Bit	Name	Function
7	TF3H	**Timer 3 High Byte Overflow Flag.** Set by hardware when the Timer 3 high byte overflows from 0xFF to 0x00. In 16-bit mode, this will occur when Timer 3 overflows from 0xFFFF to 0x0000. When the Timer 3 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 3 interrupt service routine. This bit must be cleared by firmware.
6	TF3L	**Timer 3 Low Byte Overflow Flag.** Set by hardware when the Timer 3 low byte overflows from 0xFF to 0x00. TF3L will be set when the low byte overflows regardless of the Timer 3 mode. This bit must be cleared by firmware.
5	TF3LEN	**Timer 3 Low Byte Interrupt Enable.** When set to 1, this bit enables Timer 3 Low Byte interrupts. If Timer 3 interrupts are also enabled, an interrupt will be generated when the low byte of Timer 3 overflows.
4	TF3CEN	**Timer 3 Capture Enable.** When set to 1, this bit enables Timer 3 Capture Mode. If TF3CEN is set and Timer 3 interrupts are enabled, an interrupt will be generated based on the selected input capture source, and the current 16-bit timer value in TMR3H:TMR3L will be copied to TMR3RLH:TMR3RLL.
3	T3SPLIT	**Timer 3 Split Mode Enable.** When this bit is set, Timer 3 operates as two 8-bit timers with auto-reload. 0: Timer 3 operates in 16-bit auto-reload mode. 1: Timer 3 operates as two 8-bit auto-reload timers.
2	TR3	**Timer 3 Run Control.** Timer 3 is enabled by setting this bit to 1. In 8-bit mode, this bit enables/disables TMR3H only; TMR3L is always enabled in split mode.

Table 32.15. TMR3CN Register Bit Descriptions

Bit	Name	Function
1:0	T3XCLK	**Timer 3 External Clock Select.** This bit selects the external clock source for Timer 3. If Timer 3 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 3 Clock Select bits (T3MH and T3ML in register CKCON) may still be used to select between the external clock and the system clock for either timer. Note: External clock sources are synchronized with the system clock. 00: External Clock is SYSCLK/12. Capture trigger is RTC. 01: External Clock is External Oscillator/8. Capture trigger is RTC. 10: External Clock is SYSCLK/12. Capture trigger is External Oscillator/8. 11: External Clock is RTC. Capture trigger is External Oscillator/8.

Register 32.14. TMR3RLL: Timer 3 Reload Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x92								

Table 32.16. TMR3RLL Register Bit Descriptions

Bit	Name	Function
7:0	TMR3RLL	**Timer 3 Reload Low Byte.** When operating in one of the auto-reload modes, TMR3RLL holds the reload value for the low byte of Timer 3 (TMR3L). When operating in capture mode, TMR3RLL is the captured value of TMR3L.

Register 32.15. TMR3RLH: Timer 3 Reload High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3RLH							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x93								

Table 32.17. TMR3RLH Register Bit Descriptions

Bit	Name	Function
7:0	TMR3RLH	**Timer 3 Reload High Byte.** When operating in one of the auto-reload modes, TMR3RLH holds the reload value for the high byte of Timer 3 (TMR3H). When operating in capture mode, TMR3RLH is the captured value of TMR3H.

Register 32.16. TMR3L: Timer 3 Low Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3L							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x94								

Table 32.18. TMR3L Register Bit Descriptions

Bit	Name	Function
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.

Register 32.17. TMR3H: Timer 3 High Byte

Bit	7	6	5	4	3	2	1	0
Name	TMR3H							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0x95								

Table 32.19. TMR3H Register Bit Descriptions

Bit	Name	Function
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.

33. 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 three 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. 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, SmaRTClock divided by 8, Timer 0 overflows, 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 to 11-Bit PWM, or 16-Bit PWM (each mode is described in Section “33.3. Capture/Compare Modules” on page 423). 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 33.1

Important Note: The PCA Module 2 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 33.4 for details.

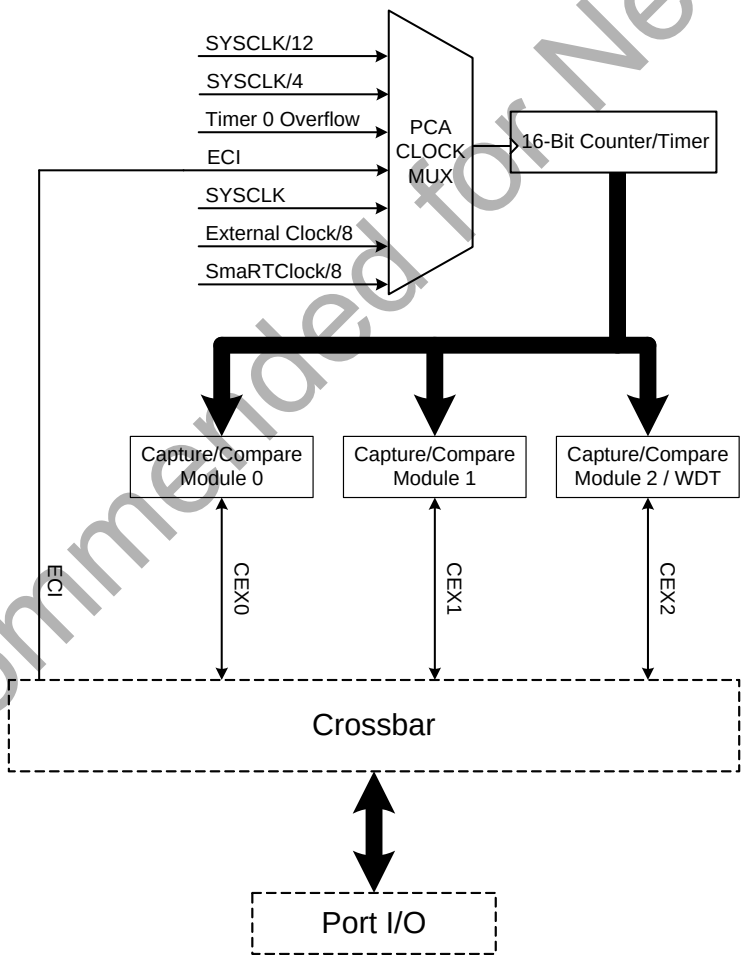


Figure 33.1. PCA Block Diagram

33.1. 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 33.1.

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. Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 33.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8[1]
1	1	0	SmaRTClock oscillator source divided by 8[2]
1	1	1	Reserved

Notes:
1. External oscillator source divided by 8 is synchronized with the system clock.
2. SmaRTClock oscillator source divided by 8 is synchronized with the system clock.

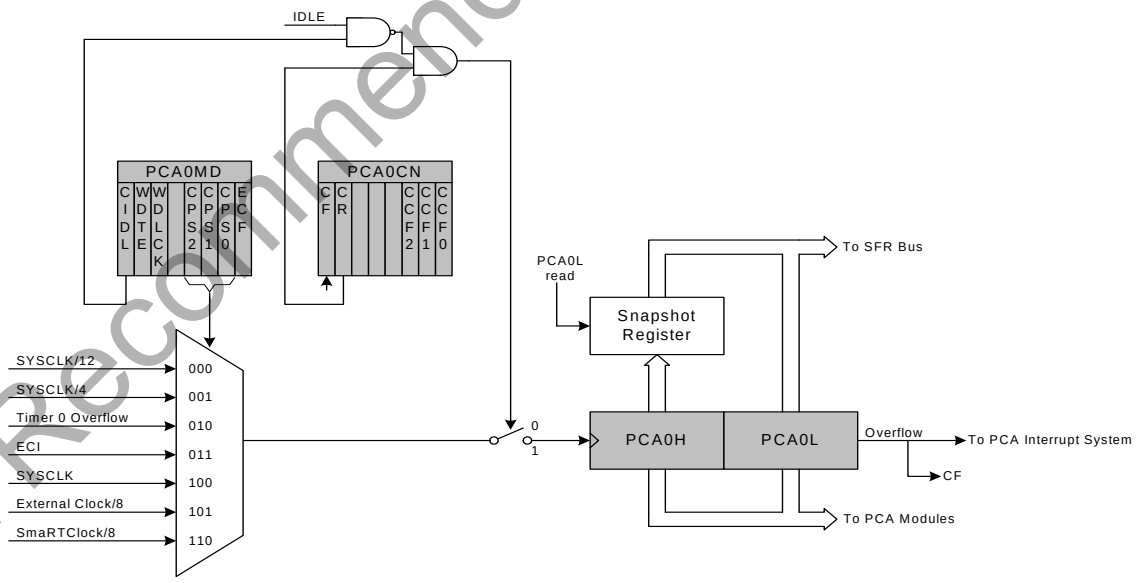


Figure 33.2. PCA Counter/Timer Block Diagram

33.2. PCA0 Interrupt Sources

Figure 33.3 shows a diagram of the PCA interrupt tree. There are five independent event flags that can be used to generate a PCA0 interrupt. They are: the main PCA counter overflow flag (CF), which is set upon a 16-bit overflow of the PCA0 counter, an intermediate overflow flag (COVF), which can be set on an overflow from the 8th, 9th, 10th, or 11th bit of the PCA0 counter, and the individual flags for each PCA channel (CCF0, CCF1, and CCF2), which are set according to the operation mode of that module. These event flags are always set when the trigger condition occurs. Each of these flags can be individually selected to generate a PCA0 interrupt, using the corresponding interrupt enable flag (ECF for CF, ECOV for COVF, and ECCFn for each CCFn). PCA0 interrupts must be globally enabled before any individual interrupt sources are recognized by the processor. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1.

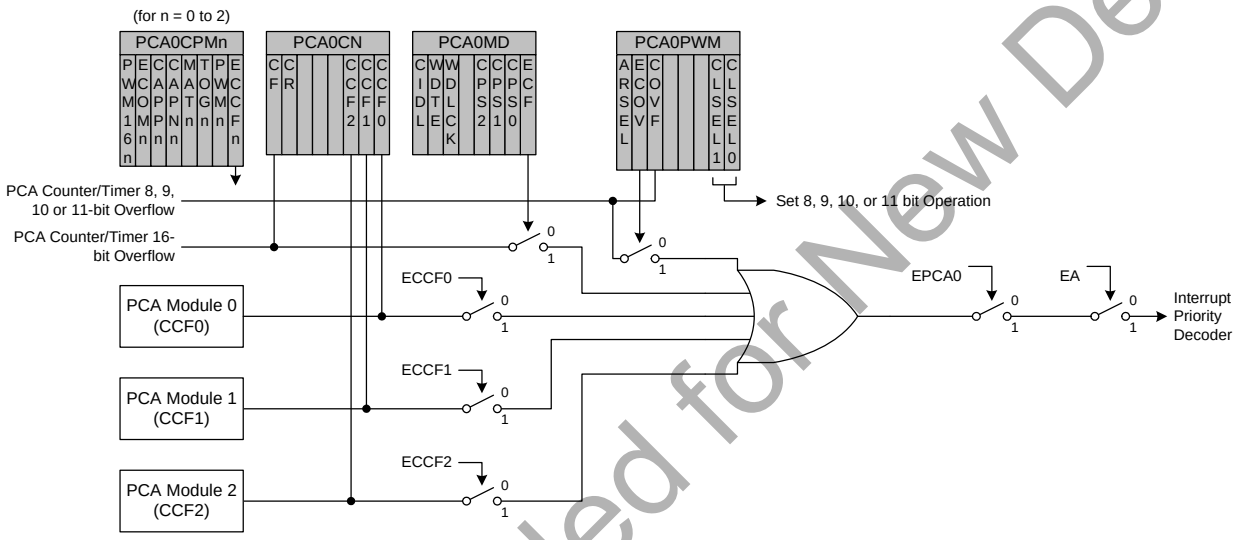


Figure 33.3. PCA Interrupt Block Diagram

33.3. 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 to 11-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 33.2 summarizes the bit settings in the PCA0CPMn and PCA0PWM registers used to select the PCA capture/compare module's operating mode. Note that all modules set to use 8, 9, 10, or 11-bit PWM mode must use the same cycle length (8–11 bits). Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt.

Table 33.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules

Operational Mode	PCA0CPMn								PCA0PWM				
Bit Number	7	6	5	4	3	2	1	0	7	6	5	4–2	1–0
Capture triggered by positive edge on CEXn	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
Capture triggered by negative edge on CEXn	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
Capture triggered by any transition on CEXn	X	X	1	1	0	0	0	A	0	X	B	XXX	XX
Software Timer	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
High Speed Output	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
Frequency Output	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11-Bit Pulse Width Modulator (Note 7)	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16-Bit Pulse Width Modulator	1	C	0	0	E	0	1	A	0	X	B	XXX	XX

Notes:

1. X = Don't Care (no functional difference for individual module if 1 or 0).
2. A = Enable interrupts for this module (PCA interrupt triggered on CCFn set to 1).
3. B = Enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL[1:0]).
4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0).
5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA0CPHn and PCA0CPLn.
6. E = When set, a match event will cause the CCFn flag for the associated channel to be set.
7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting.

33.3.1. 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. An interrupt request is generated if the CCFn interrupt for that module is 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.

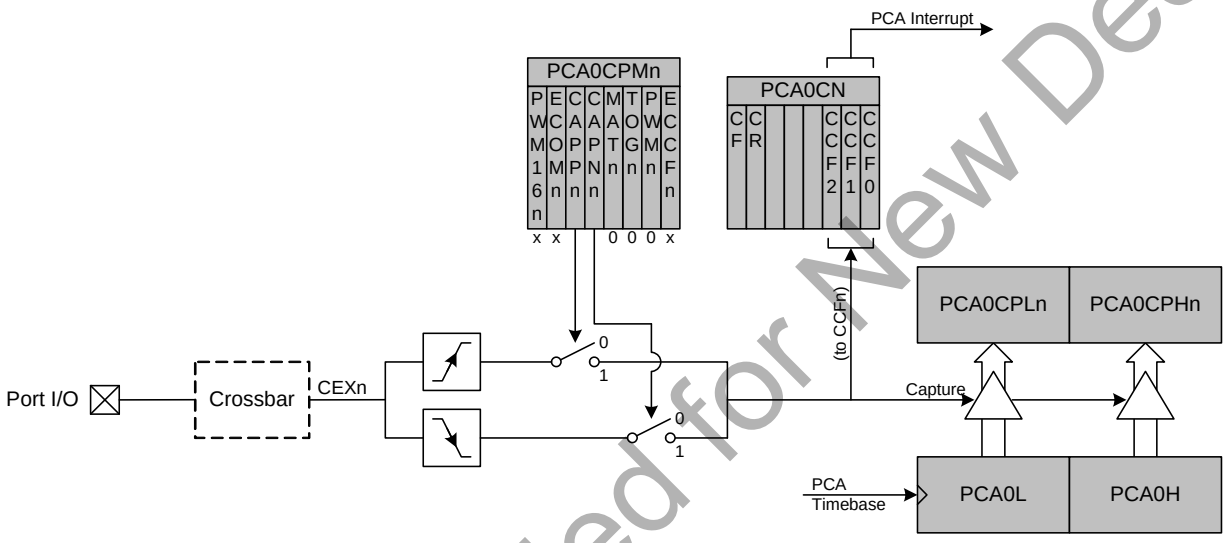


Figure 33.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

33.3.2. 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. An interrupt request is generated if the CCFn interrupt for that module is 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.

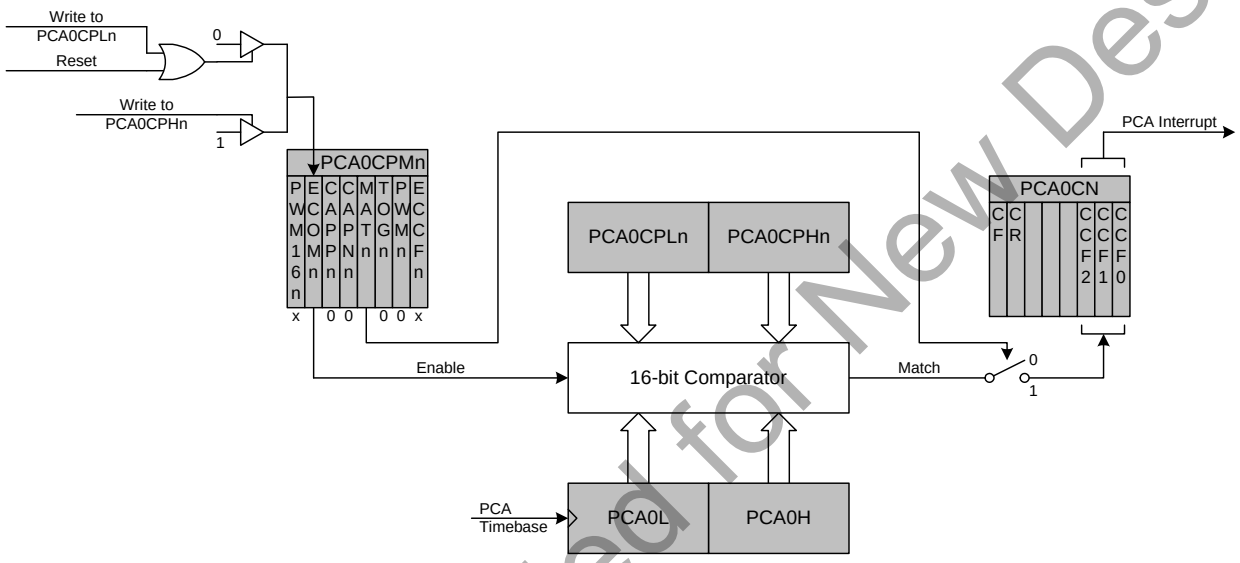


Figure 33.5. PCA Software Timer Mode Diagram

33.3.3. 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). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is 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 TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode. If ECOMn is cleared, the associated pin will retain its state, and not toggle on the next match event.

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.

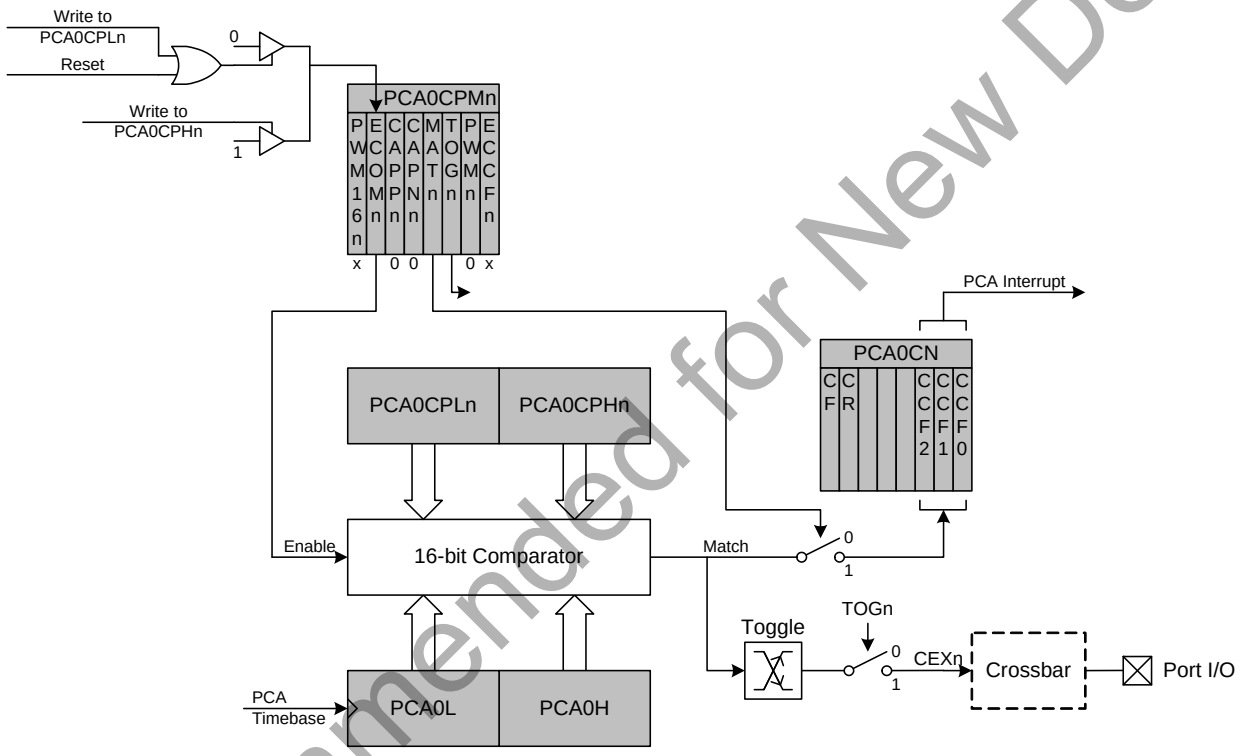


Figure 33.6. PCA High-Speed Output Mode Diagram

33.3.4. 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 33.1.

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

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

Equation 33.1. 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. Note that the MATn bit should normally be set to 0 in this mode. If the MATn bit is set to 1, the CCFn flag for the channel will be set when the 16-bit PCA0 counter and the 16-bit capture/compare register for the channel are equal.

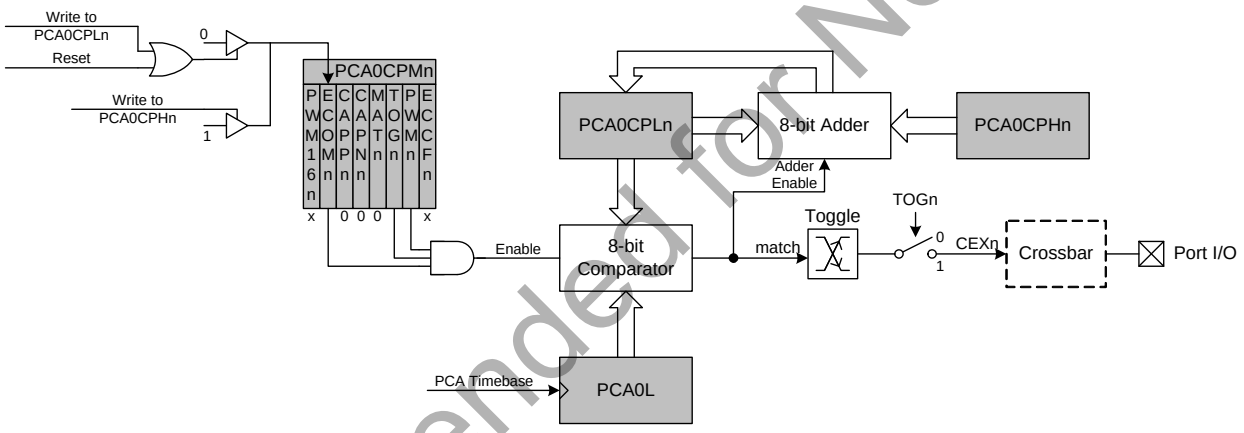


Figure 33.7. PCA Frequency Output Mode

33.3.5. 8-bit, 9-bit, 10-bit and 11-bit Pulse Width Modulator Modes

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, and the setting of the PWM cycle length (8, 9, 10 or 11-bits). For backwards-compatibility with the 8-bit PWM mode available on other devices, the 8-bit PWM mode operates slightly different than 9, 10 and 11-bit PWM modes. **It is important to note that all channels configured for 8/9/10/11-bit PWM mode will use the same cycle length.** It is not possible to configure one channel for 8-bit PWM mode and another for 11-bit mode (for example). However, other PCA channels can be configured to Pin Capture, High-Speed Output, Software Timer, Frequency Output, or 16-bit PWM mode independently.

33.3.6. 8-Bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 8-bit PWM mode 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 33.8). 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, and setting the CLSEL bits in register PCA0PWM to 00b enables 8-Bit Pulse Width Modulator mode. If the MATn bit is set to 1, the CCFn flag for the module will be set each time an 8-bit comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 256 PCA clock cycles. The duty cycle for 8-Bit PWM Mode is given in Equation 33.2.

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.

$$\text{Duty Cycle} = \frac{(256 - PCA0CPHn)}{256}$$

Equation 33.2. 8-Bit PWM Duty Cycle

Using Equation 33.2, 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.

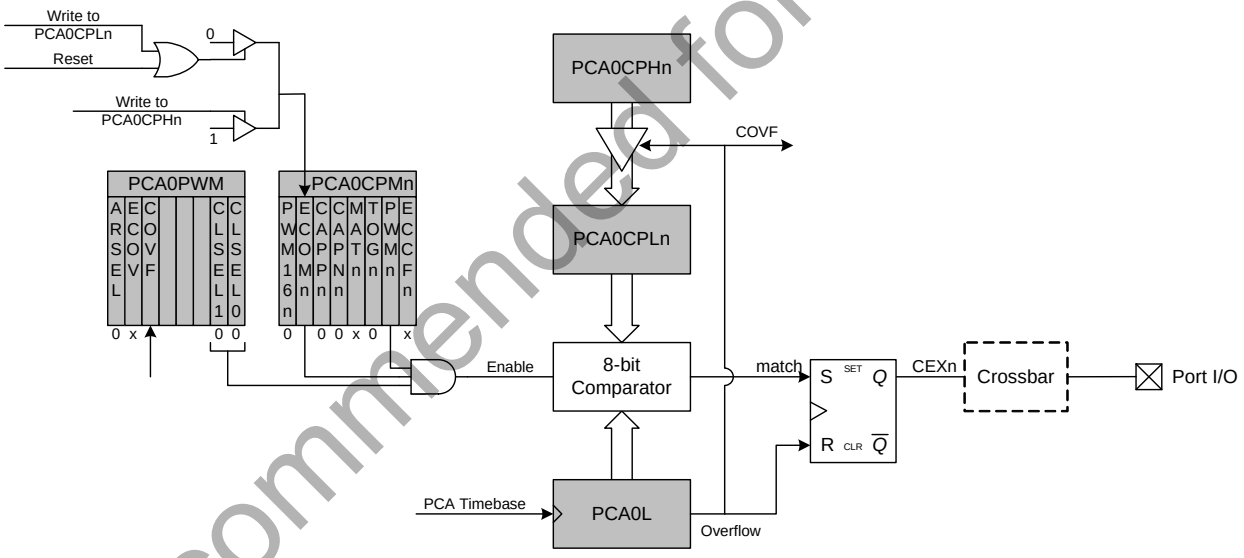


Figure 33.8. PCA 8-Bit PWM Mode Diagram

33.3.7. 9/10/11-bit Pulse Width Modulator Mode

The duty cycle of the PWM output signal in 9/10/11-bit PWM mode should be varied by writing to an “Auto-Reload” Register, which is dual-mapped into the PCA0CPHn and PCA0CPLn register locations. The data written to define the duty cycle should be right-justified in the registers. The auto-reload registers are accessed (read or written) when the bit ARSEL in PCA0PWM is set to 1. The capture/compare registers are accessed when ARSEL is set to 0.

When the least-significant N bits of the PCA0 counter match the value in the associated module’s capture/compare register (PCA0CPn), the output on CEXn is asserted high. When the counter overflows from the Nth bit, CEXn is asserted low (see Figure 33.9). Upon an overflow from the Nth bit, the COVF flag is set, and the value stored in the module’s auto-reload register is loaded into the capture/compare register. The value of N is determined by the CLSEL bits in register PCA0PWM.

The 9, 10 or 11-bit PWM mode is selected by setting the ECOMn and PWMn bits in the PCA0CPMn register, and setting the CLSEL bits in register PCA0PWM to the desired cycle length (other than 8-bits). If the MATn bit is set to 1, the CCFn flag for the module will be set each time a comparator match (rising edge) occurs. The COVF flag in PCA0PWM can be used to detect the overflow (falling edge), which will occur every 512 (9-bit), 1024 (10-bit) or 2048 (11-bit) PCA clock cycles. The duty cycle for 9/10/11-Bit PWM Mode is given in Equation 33.2, where N is the number of bits in the PWM cycle.

Important Note About PCA0CPHn and PCA0CPLn Registers: When writing a 16-bit value to the PCA0CPn 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.

$$\text{Duty Cycle} = \frac{(2^N - PCA0CPn)}{2^N}$$

Equation 33.3. 9, 10, and 11-Bit PWM Duty Cycle

A 0% duty cycle may be generated by clearing the ECOMn bit to 0.

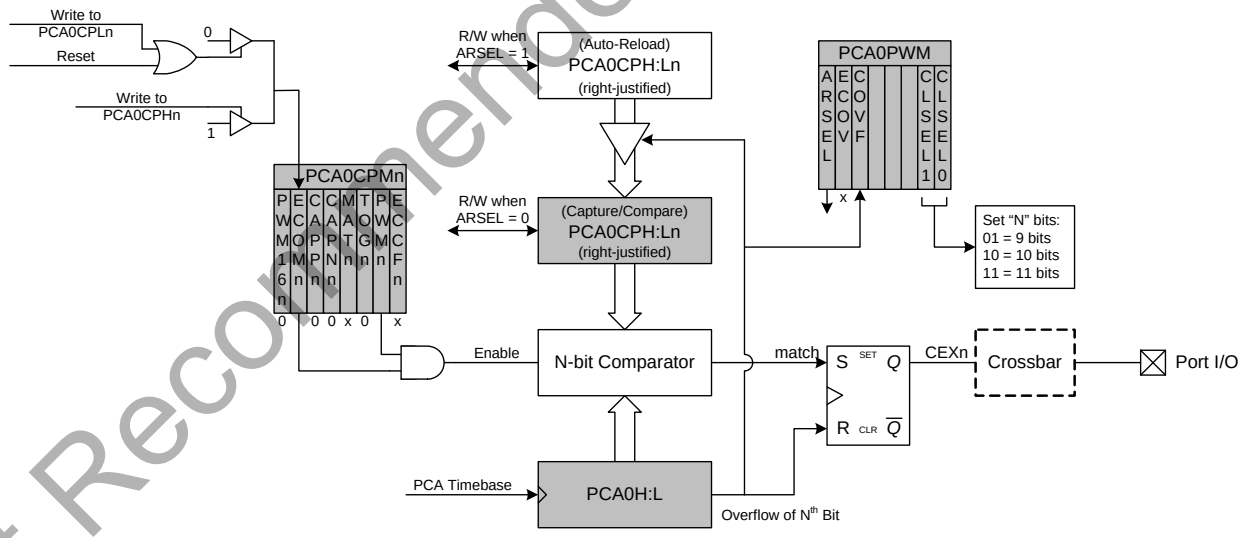


Figure 33.9. PCA 9, 10 and 11-Bit PWM Mode Diagram

33.3.8. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. 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 16-bit 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. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 33.4.

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.

$$\text{Duty Cycle} = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 33.4. 16-Bit PWM Duty Cycle

Using Equation 33.4, 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.

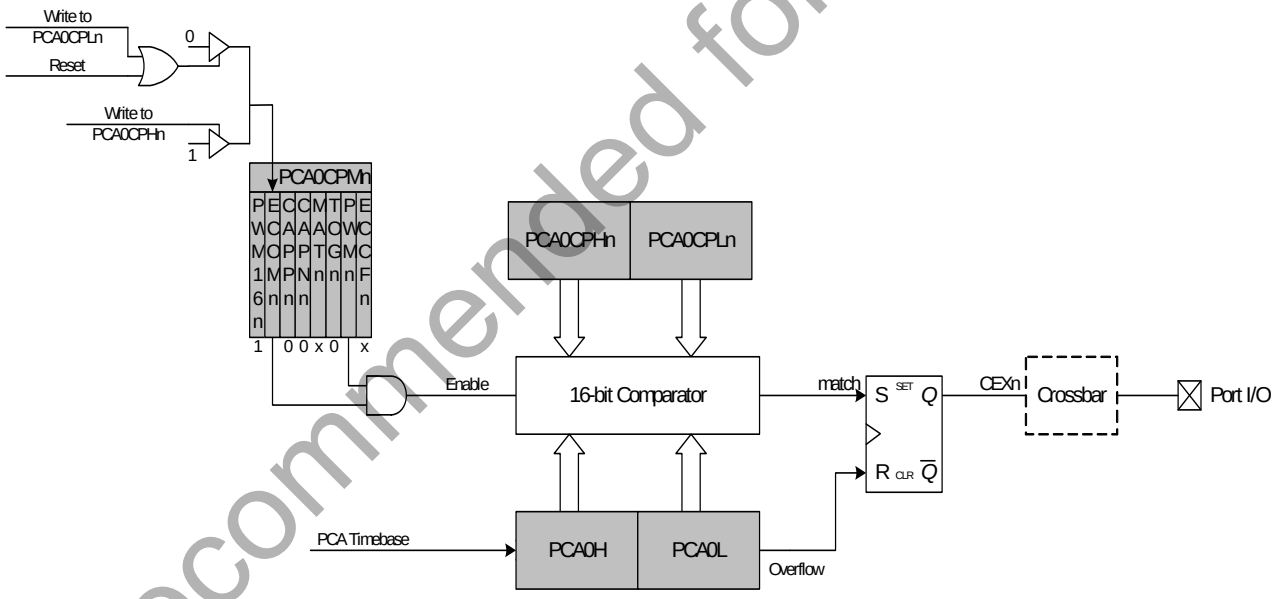


Figure 33.10. PCA 16-Bit PWM Mode

33.4. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 2. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 2 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA counter high byte; the Module 2 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.** The WDT will generate a reset shortly after code begins execution. To avoid this reset, the WDT should be explicitly disabled (and optionally re-configured and re-enabled if it is used in the system).

33.4.1. 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 2 is forced into software timer mode.
- Writes to the Module 2 mode register (PCA0CPM2) 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 bit (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH2 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 PCA0CPH2. Upon a PCA0CPH2 write, PCA0H plus the offset held in PCA0CPL2 is loaded into PCA0CPH2 (See Figure 33.11).

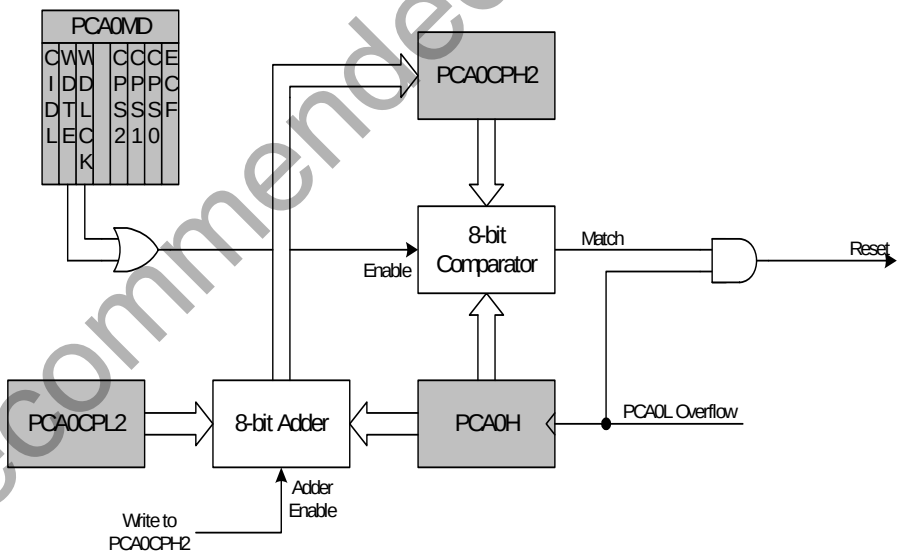


Figure 33.11. PCA Module 2 with Watchdog Timer Enabled

The 8-bit offset held in PCA0CPH2 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 33.5, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL2) + (256 - PCA0L)$$

Equation 33.5. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH2 and PCA0H. Software may force a WDT reset by writing a 1 to the CCF2 flag (PCA0CN.2) while the WDT is enabled.

33.4.2. 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 PCA0CPL2 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. Reset the WDT timer by writing to PCA0CPH2.

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 PCA0CPL2 defaults to 0x00. Using Equation 33.5, this results in a WDT timeout interval of 256 PCA clock cycles, or 3072 system clock cycles. Table 33.3 lists some example timeout intervals for typical system clocks.

Table 33.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL2	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
3,062,500[2]	255	257
3,062,500[2]	128	129.5
3,062,500[2]	32	33.1
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:

1. Assumes SYSCLK/12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal SYSCLK reset frequency = Internal Oscillator divided by 8.

33.5. PCA0 Control Registers

Register 33.1. PCA0CN: PCA Control

Bit	7	6	5	4	3	2	1	0
Name	CF	CR	Reserved			CCF2	CCF1	CCF0
Type	RW	RW	R			RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xD8 (bit-addressable)								

Table 33.4. PCA0CN Register Bit Descriptions

Bit	Name	Function
7	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 firmware.
6	CR	**PCA Counter/Timer Run Control.** This bit enables/disables the PCA Counter/Timer. 0: Stop the PCA Counter/Timer. 1: Start the PCA Counter/Timer running.
5:3	Reserved	Must write reset value.
2	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 firmware.
1	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 firmware.
0	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 firmware.

Register 33.2. PCA0MD: PCA Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL	WDTE	WDLCK	Reserved	CPS			ECF
Type	RW	RW	RW	R	RW			RW
Reset	0	1	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xD9								

Table 33.5. PCA0MD Register Bit Descriptions

Bit	Name	Function
7	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.
6	WDTE	**Watchdog Timer Enable.** If this bit is set, PCA Module 2 is used as the watchdog timer. 0: Disable Watchdog Timer. 1: Enable PCA Module 2 as the Watchdog Timer.
5	WDLCK	**Watchdog Timer Lock.** This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset. 0: Watchdog Timer Enable unlocked. 1: Watchdog Timer Enable locked.
4	Reserved	Must write reset value.
3:1	CPS	**PCA Counter/Timer Pulse Select.** These bits select the timebase source for the PCA counter. 000: System clock divided by 12. 001: System clock divided by 4. 010: Timer 0 overflow. 011: High-to-low transitions on ECI (max rate = system clock divided by 4). 100: System clock. 101: External clock divided by 8 (synchronized with the system clock). 110: RTC divided by 8. 111: Reserved.
0	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.

Register 33.3. PCA0PWM: PCA PWM Configuration

Bit	7	6	5	4	3	2	1	0
Name	ARSEL	ECOV	COVF	Reserved			CLSEL	
Type	RW	RW	RW	R			RW	
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDF								

Table 33.6. PCA0PWM Register Bit Descriptions

Bit	Name	Function
7	ARSEL	**Auto-Reload Register Select.** This bit selects whether to read and write the normal PCA capture/compare registers (PCA0CPn), or the Auto-Reload registers at the same SFR addresses. This function is used to define the reload value for 9 to 11-bit PWM modes. In all other modes, the Auto-Reload registers have no function. 0: Read/Write Capture/Compare Registers at PCA0CPHn and PCA0CPLn. 1: Read/Write Auto-Reload Registers at PCA0CPHn and PCA0CPLn.
6	ECOV	**Cycle Overflow Interrupt Enable.** This bit sets the masking of the Cycle Overflow Flag (COVF) interrupt. 0: COVF will not generate PCA interrupts. 1: A PCA interrupt will be generated when COVF is set.
5	COVF	**Cycle Overflow Flag.** This bit indicates an overflow of the 8th to 11th bit of the main PCA counter (PCA0). The specific bit used for this flag depends on the setting of the Cycle Length Select bits. The bit can be set by hardware or firmware, but must be cleared by firmware. 0: No overflow has occurred since the last time this bit was cleared. 1: An overflow has occurred since the last time this bit was cleared.
4:2	Reserved	Must write reset value.
1:0	CLSEL	**Cycle Length Select.** When 16-bit PWM mode is not selected, these bits select the length of the PWM cycle. This affects all channels configured for PWM which are not using 16-bit PWM mode. These bits are ignored for individual channels configured to 16-bit PWM mode. 00: 8 bits. 01: 9 bits. 10: 10 bits. 11: 11 bits.

Register 33.4. PCA0L: PCA Counter/Timer Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0L							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xF9								

Table 33.7. PCA0L Register Bit Descriptions

Bit	Name	Function
7:0	PCA0L	**PCA Counter/Timer Low Byte.** The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.
Note: When the WDTE bit is set to 1, the PCA0L register cannot be modified by firmware. To change the contents of the PCA0L register, the Watchdog Timer must first be disabled.		

Register 33.5. PCA0H: PCA Counter/Timer High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0H							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xFA								

Table 33.8. PCA0H Register Bit Descriptions

Bit	Name	Function
7:0	PCA0H	**PCA Counter/Timer High Byte.** The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer. Reads of this register will read the contents of a "snapshot" register, whose contents are updated only when the contents of PCA0L are read.
Note: When the WDTE bit is set to 1, the PCA0H register cannot be modified by firmware. To change the contents of the PCA0H register, the Watchdog Timer must first be disabled.		

Register 33.6. PCA0CPM0: PCA Channel 0 Capture/Compare Mode 0

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDA								

Table 33.9. PCA0CPM0 Register Bit Descriptions

Bit	Name	Function
7	PWM16	**Channel 0 16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	**Channel 0 Comparator Function Enable.** This bit enables the comparator function.
5	CAPP	**Channel 0 Capture Positive Function Enable.** This bit enables the positive edge capture capability.
4	CAPN	**Channel 0 Capture Negative Function Enable.** This bit enables the negative edge capture capability.
3	MAT	**Channel 0 Match Function Enable.** This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF0 bit in the PCA0MD register to be set to logic 1.
2	TOG	**Channel 0 Toggle Function Enable.** This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX0 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	**Channel 0 Pulse Width Modulation Mode Enable.** This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX0 pin. 8 to 11-bit PWM is used if PWM16 is cleared to 0; 16-bit mode is used if PWM16 is set to 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	**Channel 0 Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCF0) interrupt. 0: Disable CCF0 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF0 is set.

Register 33.7. PCA0CPL0: PCA Channel 0 Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPL0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xFB								

Table 33.10. PCA0CPL0 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPL0	**PCA Channel 0 Capture Module Low Byte.** The PCA0CPL0 register holds the low byte (LSB) of the 16-bit capture module. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOM bit to a 0.		

Register 33.8. PCA0CPH0: PCA Channel 0 Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPH0							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xFC								

Table 33.11. PCA0CPH0 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPH0	**PCA Channel 0 Capture Module High Byte.** The PCA0CPH0 register holds the high byte (MSB) of the 16-bit capture module. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOM bit to a 1.		

Register 33.9. PCA0CPM1: PCA Channel 1 Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDB								

Table 33.12. PCA0CPM1 Register Bit Descriptions

Bit	Name	Function
7	PWM16	**Channel 1 16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	**Channel 1 Comparator Function Enable.** This bit enables the comparator function.
5	CAPP	**Channel 1 Capture Positive Function Enable.** This bit enables the positive edge capture capability.
4	CAPN	**Channel 1 Capture Negative Function Enable.** This bit enables the negative edge capture capability.
3	MAT	**Channel 1 Match Function Enable.** This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF1 bit in the PCA0MD register to be set to logic 1.
2	TOG	**Channel 1 Toggle Function Enable.** This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX1 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	**Channel 1 Pulse Width Modulation Mode Enable.** This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX1 pin. 8 to 11-bit PWM is used if PWM16 is cleared to 0; 16-bit mode is used if PWM16 is set to 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	**Channel 1 Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCF1) interrupt. 0: Disable CCF1 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF1 is set.

Register 33.10. PCA0CPL1: PCA Channel 1 Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPL1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xE9								

Table 33.13. PCA0CPL1 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPL1	**PCA Channel 1 Capture Module Low Byte.** The PCA0CPL1 register holds the low byte (LSB) of the 16-bit capture module. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOM bit to a 0.		

Register 33.11. PCA0CPH1: PCA Channel 1 Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPH1							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xEA								

Table 33.14. PCA0CPH1 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPH1	**PCA Channel 1 Capture Module High Byte.** The PCA0CPH1 register holds the high byte (MSB) of the 16-bit capture module. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOM bit to a 1.		

Register 33.12. PCA0CPM2: PCA Channel 2 Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF
Type	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xDC								

Table 33.15. PCA0CPM2 Register Bit Descriptions

Bit	Name	Function
7	PWM16	**Channel 2 16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOM	**Channel 2 Comparator Function Enable.** This bit enables the comparator function.
5	CAPP	**Channel 2 Capture Positive Function Enable.** This bit enables the positive edge capture capability.
4	CAPN	**Channel 2 Capture Negative Function Enable.** This bit enables the negative edge capture capability.
3	MAT	**Channel 2 Match Function Enable.** This bit enables the match function. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCF2 bit in the PCA0MD register to be set to logic 1.
2	TOG	**Channel 2 Toggle Function Enable.** This bit enables the toggle function. When enabled, matches of the PCA counter with the capture/compare register cause the logic level on the CEX2 pin to toggle. If the PWM bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWM	**Channel 2 Pulse Width Modulation Mode Enable.** This bit enables the PWM function. When enabled, a pulse width modulated signal is output on the CEX2 pin. 8 to 11-bit PWM is used if PWM16 is cleared to 0; 16-bit mode is used if PWM16 is set to 1. If the TOG bit is also set, the module operates in Frequency Output Mode.
0	ECCF	**Channel 2 Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCF2) interrupt. 0: Disable CCF2 interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCF2 is set.

Register 33.13. PCA0CPL2: PCA Channel 2 Capture Module Low Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPL2							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xEB								

Table 33.16. PCA0CPL2 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPL2	**PCA Channel 2 Capture Module Low Byte.** The PCA0CPL2 register holds the low byte (LSB) of the 16-bit capture module. This register address also allows access to the low byte of the corresponding PCA channel's auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will clear the module's ECOM bit to a 0.		

Register 33.14. PCA0CPH2: PCA Channel 2 Capture Module High Byte

Bit	7	6	5	4	3	2	1	0
Name	PCA0CPH2							
Type	RW							
Reset	0	0	0	0	0	0	0	0
SFR Page = 0x0; SFR Address: 0xEC								

Table 33.17. PCA0CPH2 Register Bit Descriptions

Bit	Name	Function
7:0	PCA0CPH2	**PCA Channel 2 Capture Module High Byte.** The PCA0CPH2 register holds the high byte (MSB) of the 16-bit capture module. This register address also allows access to the high byte of the corresponding PCA channel's auto-reload value for 9 to 11-bit PWM mode. The ARSEL bit in register PCA0PWM controls which register is accessed.
Note: A write to this register will set the module's ECOM bit to a 1.		

34. C2 Interface

C8051F97x 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 bidirectional C2 data signal (C2D) to transfer information between the device and a host system. Details on the C2 protocol can be found in the C2 Interface Specification.

34.1. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and flash programming may be performed. C2CK is shared with the $\overline{\text{RST}}$ pin, while the C2D signal is shared with a port I/O pin. 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 and C2D pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 34.1.

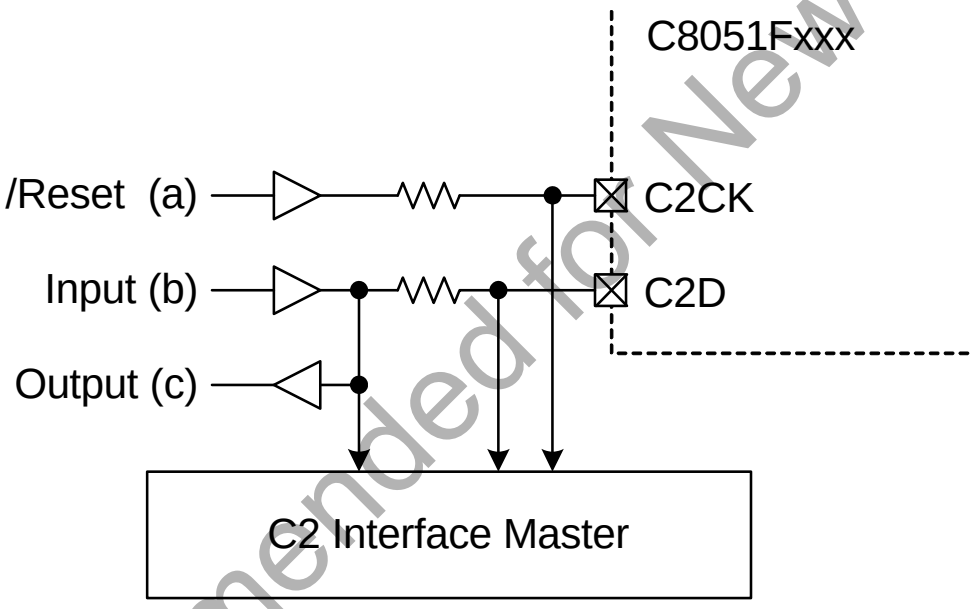


Figure 34.1. Typical C2 Pin Sharing

The configuration in Figure 34.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The $\overline{\text{RST}}$ pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

34.2. C2 Interface Registers

The following describes the C2 registers necessary to perform flash programming through the C2 interface. All C2 registers are accessed through the C2 interface, and are not available in the SFR map for firmware access.

Register 34.1. C2ADD: C2 Address

Bit	7	6	5	4	3	2	1	0
Name	C2ADD							
Type	RW							
Reset	0	0	0	0	0	0	0	0
This register is part of the C2 protocol.								

Table 34.1. C2ADD Register Bit Descriptions

Bit	Name	Function
7:0	C2ADD	**C2 Address.** The C2ADD register is accessed via the C2 interface. The value written to C2ADD selects the target data register for C2 Data Read and Data Write commands. 0x00: C2DEVID 0x01: C2REVID 0x02: C2FPCTL 0xB4: C2FPDAT

Register 34.2. C2DEVID: C2 Device ID

Bit	7	6	5	4	3	2	1	0
Name	C2DEVID							
Type	R							
Reset	0	0	1	0	1	0	0	1
C2 Address: 0x00								

Table 34.2. C2DEVID Register Bit Descriptions

Bit	Name	Function
7:0	C2DEVID	**Device ID.** This read-only register returns the 8-bit device ID: 0x29 (C8051F97x).

Register 34.3. C2REVID: C2 Revision ID

Bit	7	6	5	4	3	2	1	0
Name	C2REVID							
Type	R							
Reset	X	X	X	X	X	X	X	X
C2 Address: 0x01								

Table 34.3. C2REVID Register Bit Descriptions

Bit	Name	Function
7:0	C2REVID	**Revision ID.** This read-only register returns the 8-bit revision ID. For example: 0x01 = Revision A.

Register 34.4. C2FPCTL: C2 Flash Programming Control

Bit	7	6	5	4	3	2	1	0
Name	C2FPCTL							
Type	RW							
Reset	0	0	0	0	0	0	0	0
C2 Address: 0x02								

Table 34.4. C2FPCTL Register Bit Descriptions

Bit	Name	Function
7:0	C2FPCTL	**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.

Register 34.5. C2FPDAT: C2 Flash Programming Data

Bit	7	6	5	4	3	2	1	0
Name	C2FPDAT							
Type	RW							
Reset	0	0	0	0	0	0	0	0
C2 Address: 0xB4								

Table 34.5. C2FPDAT Register Bit Descriptions

Bit	Name	Function
7:0	C2FPDAT	**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. 0x03: Device Erase 0x06: Flash Block Read 0x07: Flash Block Write 0x08: Flash Page Erase

Document Change List

Revision 1.0 to Revision 1.1

- Updated packaging information for QFN48 and QFN32 packages in "5. QFN-48 Package Specifications" on page 43 and "6. QFN-32 Package Specifications" on page 46.
- Added a note about an additional divide-by-2 stage on RC and C oscillator modes in "24.3.2. External RC Mode" on page 243 and "24.3.3. External Capacitor Mode" on page 245.
- Fixed the first sentence in "26.3. Priority Crossbar Decoder" on page 281 that referred to UART0 as the top priority peripheral on the crossbar.
- Removed all ADC0MX channels other than ADC0.0 and marked them as Reserved, since pin selections are made using AMUX0.
- Updated Table 3.1, “Pin Definitions for C8051F970/3-A-GM (QFN-48),” on page 32, Table 3.2, “Pin Definitions for C8051F971/4-A-GM (QFN-32),” on page 36, and Table 3.3, “Pin Definitions for C8051F972/5-A-GM (QFN-24),” on page 39 to replace ADC0.n with AMUX0.n.
- Updated QFN-32 and QFN-24 pin definitions with correct pin numbering.
- Added wake-up request and RTC oscillator output to Table 3.1, “Pin Definitions for C8051F970/3-A-GM (QFN-48),” on page 32 and specified in Register 16.4, “PMU0MD: Power Management Unit Mode,” on page 103 that these outputs are not available on QFN-32 and QFN-24 packages.
- Removed a mention of UART0 routing to P0.4 and P0.5 in Register 26.1, “XBR0: Port I/O Crossbar 0,” on page 285.
- Updated Figure 22.4, “DMA Mode Operation Flow Chart,” on page 206 to remove clearing ACCMD to 0 and added a note in "22.6. DMA Mode Operation" on page 205 regarding generating the MAC output for two arrays.
- Updated all references of “QFN-28” to “QFN-24.”
- Added a note to "16.5. Sleep Mode" on page 97 that entering Sleep mode may cause a device to disconnect while debugging.
- Updated the PERIPH field in Register 21.6, “DMA0NCF: DMA0 Channel Configuration,” on page 195 to swap values 6 and 7.
- Updated references to MSTEN to refer to SPI0CFG instead of SPI0CN in "28. Serial Peripheral Interface (SPI0)" on page 328.
- Updated the example in "22.11.3. Initializing Memory Block Using DMA0 and MAC0" on page 212 to refer to the MAC0ITER register instead of MAC0ICT.
- Removed section 24.4.2 SMBus Pin Swap and 29.4.3 SMBus Timing Control because these features are not available on this device family.

Revision 0.1 to Revision 1.0

- Updated Capacitive Sense and ADC input channels listed on the front page.
- Removed mention of the -I temperature grade from Figure 4.1, “C8051F97x Part Numbering,” on page 41.
- Updated Digital Supply Current numbers in Table 1.2, “Global Electrical Characteristics,” on page 10 to reflect the latest data.
- Removed mention of 12-bit mode for ADC0.
- Added a note to "17.1. ADC0 Analog Multiplexer" on page 105, the ADC0MX register, and all AMUX0 registers regarding disconnecting the AMUX0 when measuring an internal signal with the ADC.
- Updated "24. Clocking Sources" on page 241 to mention that the external oscillator is not available on QFN-24 (C8051F972/5) packages.
- Updated "25. SmaRTClock (Real Time Clock, RTC0)" on page 253 references to RTC0CN at address 0x05 to correctly refer to RTC0XCN.
- Updated "25. SmaRTClock (Real Time Clock, RTC0)" on page 253 to remove mention of using an external CMOS clock with the SmaRTClock.
- Updated port pins associated with the crystal pins on each package in Table 26.1, “Port I/O Assignment for

Analog Functions,” on page 279.

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information

Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com