C8051F326/7

Full Speed USB, 16 kB Flash MCU Family

USB Function Controller
- USB specification 2.0 compliant
- Full speed (12 Mbps) or low speed (1.5 Mbps) operation
- Integrated clock recovery; no external crystal required for full speed or low speed
- Supports three fixed-function endpoints
- 256 Byte USB buffer memory
- Integrated transceiver; no external resistors required

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

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

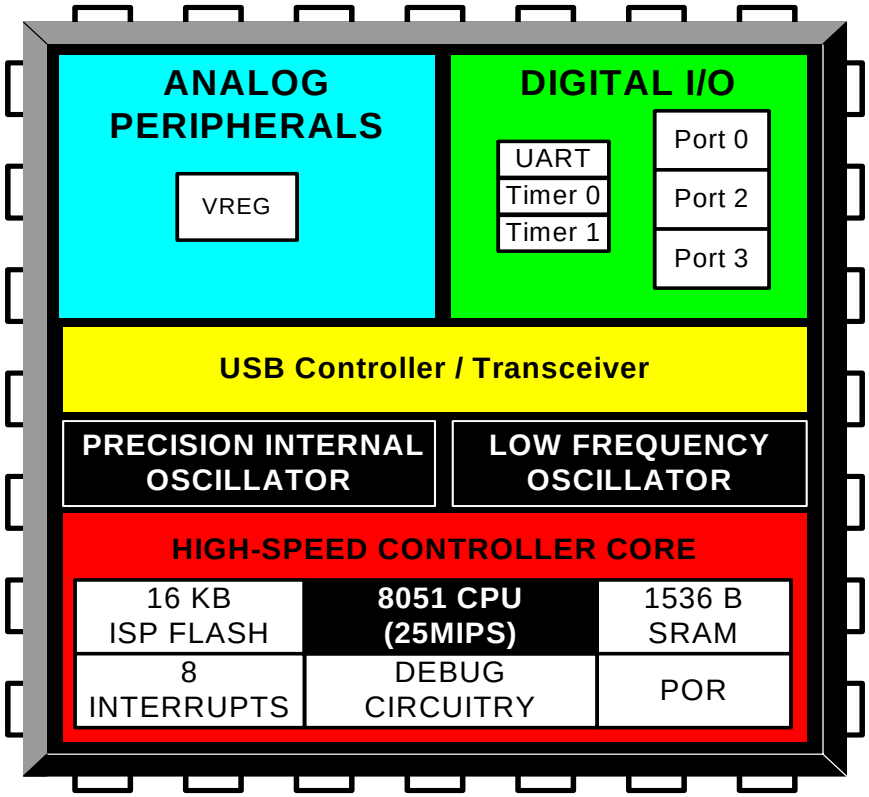
High-Speed 8051 µC Core
- Pipelined instruction architecture; executes 70% of instructions in 1 or 2 system clocks
- Up to 25 MIPS throughput with 25 MHz clock
- Expanded interrupt handler

Memory
- 1536 bytes internal RAM (1 k + 256 + 256 USB FIFO)
- 16k bytes Flash; In-system programmable in 512-byte sectors

Digital Peripherals
- 15 Port I/O; All 5 V tolerant with high sink current
- Enhanced UART
- Two general purpose 16-bit timers

Clock Sources
- Internal oscillator: 0.25% accuracy with clock recovery enabled. Supports all USB and UART modes
- External CMOS clock
- Can switch between clock sources on-the-fly; useful in power saving strategies

Packages
- 28-pin QFN
- Temperature Range: –40 to +85 °C



Copyright © 2008 by Silicon Laboratories

Table of Contents

1. System Overview 13
1.1. CIP-51™ Microcontroller Core 17
1.1.1. Fully 8051 Compatible 17
1.1.2. Improved Throughput 17
1.1.3. Additional Features 18
1.2. On-Chip Memory 19
1.3. Universal Serial Bus Controller 20
1.4. Voltage Regulator 20
1.5. On-Chip Debug Circuitry 21
1.6. Programmable Digital I/O 22
1.7. Serial Ports 22
2. Absolute Maximum Ratings 23
3. Global DC Electrical Characteristics 24
4. Pinout and Package Definitions 25
5. Voltage Regulator (REG0) 31
5.1. Regulator Mode Selection 31
5.2. VBUS Detection 31
6. CIP-51 Microcontroller 35
6.1. Instruction Set 36
6.1.1. Instruction and CPU Timing 36
6.1.2. MOVX Instruction and Program Memory 37
6.2. Memory Organization 41
6.2.1. Program Memory 41
6.2.2. Data Memory 42
6.2.3. General Purpose Registers 42
6.2.4. Bit Addressable Locations 42
6.2.5. Stack 42
6.2.6. Special Function Registers 43
6.2.7. Register Descriptions 45
6.3. Interrupt Handler 48
6.3.1. MCU Interrupt Sources and Vectors 48
6.3.2. External Interrupts 49
6.3.3. Interrupt Priorities 49
6.3.4. Interrupt Latency 49
6.3.5. Interrupt Register Descriptions 50
6.4. Power Management Modes 55
6.4.1. Idle Mode 55
6.4.2. Stop Mode 55
7. Reset Sources 57
7.1. Power-On Reset 58
7.2. Power-Fail Reset / VDD Monitor 59
7.3. External Reset 60
7.4. Missing Clock Detector Reset 60

7.5. Flash Error Reset 60
7.6. Software Reset 60
7.7. USB Reset 60
8. Flash Memory 63
8.1. Programming The Flash Memory 63
8.1.1. Flash Lock and Key Functions 63
8.1.2. Flash Erase Procedure 63
8.1.3. Flash Write Procedure 64
8.2. Non-volatile Data Storage 65
8.3. Security Options 65
9. External RAM 69
9.1. Accessing User XRAM 69
9.2. Accessing USB FIFO Space 70
10. Oscillators 71
10.1. Programmable Internal Oscillator 71
10.1.1. Adjusting the Internal Oscillator on C8051F326/7 Devices 72
10.1.2. Internal Oscillator Suspend Mode 72
10.2. Internal Low-Frequency (L-F) Oscillator 74
10.3. CMOS External Clock Input 74
10.4. 4x Clock Multiplier 75
10.5. System and USB Clock Selection 76
10.5.1. System Clock Selection 76
10.5.2. USB Clock Selection 76
11. Port Input/Output 79
11.1. Port I/O Initialization 81
11.2. General Purpose Port I/O 81
12. Universal Serial Bus Controller (USB0) 87
12.1. Endpoint Addressing 88
12.2. USB Transceiver 88
12.3. USB Register Access 90
12.4. USB Clock Configuration 94
12.5. FIFO Management 95
12.5.1. FIFO Split Mode 95
12.5.2. FIFO Double Buffering 95
12.5.3. FIFO Access 96
12.6. Function Addressing 97
12.7. Function Configuration and Control 98
12.8. Interrupts 101
12.9. The Serial Interface Engine 104
12.10. Endpoint0 104
12.10.1. Endpoint0 SETUP Transactions 104
12.10.2. Endpoint0 IN Transactions 105
12.10.3. Endpoint0 OUT Transactions 105
12.11. Configuring Endpoint1 108

12.12.Controlling Endpoint1 IN 108
12.12.1.Endpoint1 IN Interrupt or Bulk Mode 108
12.12.2.Endpoint1 IN Isochronous Mode 108
12.13.Controlling Endpoint1 OUT 112
12.13.1.Endpoint1 OUT Interrupt or Bulk Mode 112
12.13.2.Endpoint1 OUT Isochronous Mode 112
13. UART0 117
13.1.Baud Rate Generator 118
13.2.Data Format 120
13.3.Configuration and Operation 121
13.3.1.Data Transmission 121
13.3.2.Data Reception 121
13.3.3.Multiprocessor Communications 122
14. Timers 127
14.1.Timer 0 and Timer 1 Operating Modes 127
14.1.1.Mode 0: 13-bit Timer 128
14.1.2.Mode 1: 16-bit Timer 129
14.1.3.Mode 2: 8-bit Timer with Auto-Reload 129
14.1.4.Mode 3: Two 8-bit Timers (Timer 0 Only) 130
15. C2 Interface 135
15.1.C2 Interface Registers 135
15.2.C2 Pin Sharing 137
Document Change List 138
Contact Information 140

List of Figures

1. System Overview
Figure 1.1. C8051F326 Block Diagram 14
Figure 1.2. C8051F327 Block Diagram 15
Figure 1.3. Typical Connections for the C8051F326 16
Figure 1.4. Typical Connections for the C8051F327 16
Figure 1.5. Comparison of Peak MCU Execution Speeds 17
Figure 1.6. On-Chip Clock and Reset 18
Figure 1.7. On-Board Memory Map 19
Figure 1.8. USB Controller Block Diagram 20
Figure 1.9. Development/In-System Debug Diagram 21
2. Absolute Maximum Ratings
3. Global DC Electrical Characteristics
4. Pinout and Package Definitions
Figure 4.1. C8051F326 QFN-28 Pinout Diagram (Top View) 27
Figure 4.2. C8051F327 QFN-28 Pinout Diagram (Top View) 28
Figure 4.3. QFN-28 Package Drawing 29
Figure 4.4. QFN-28 Recommended PCB Land Pattern 30
5. Voltage Regulator (REG0)
Figure 5.1. REG0 Configuration: USB Bus-Powered 32
Figure 5.2. REG0 Configuration: USB Self-Powered 32
Figure 5.3. REG0 Configuration: USB Self-Powered, Regulator Disabled 33
Figure 5.4. REG0 Configuration: No USB Connection 33
6. CIP-51 Microcontroller
Figure 6.1. CIP-51 Block Diagram 35
Figure 6.2. Memory Map 41
7. Reset Sources
Figure 7.1. Reset Sources 57
Figure 7.2. Power-On and VDD Monitor Reset Timing 58
8. Flash Memory
Figure 8.1. Flash Program Memory Map and Security Byte 66
9. External RAM
Figure 9.1. External Ram Memory Map 69
Figure 9.2. XRAM Memory Map Expanded View 70
10. Oscillators
Figure 10.1. Oscillator Diagram 71
11. Port Input/Output
Figure 11.1. Port I/O Functional Block Diagram 79
Figure 11.2. Port I/O Cell Block Diagram 80
12. Universal Serial Bus Controller (USB0)
Figure 12.1. USB0 Block Diagram 87
Figure 12.2. USB0 Register Access Scheme 90
Figure 12.3. USB FIFO Allocation 95
13. UART0
Figure 13.1. UART0 Block Diagram 117

Figure 13.2. UART0 Timing Without Parity or Extra Bit 120
Figure 13.3. UART0 Timing With Parity 120
Figure 13.4. UART0 Timing With Extra Bit 120
Figure 13.5. Typical UART Interconnect Diagram 121
Figure 13.6. UART Multi-Processor Mode Interconnect Diagram 122
14. Timers
Figure 14.1. T0 Mode 0 Block Diagram 128
Figure 14.2. T0 Mode 2 Block Diagram 129
Figure 14.3. T0 Mode 3 Block Diagram 130
15. C2 Interface
Figure 15.1. Typical C2 Pin Sharing 137

List of Tables

1. System Overview
Table 1.1. Product Selection Guide 13
2. Absolute Maximum Ratings
Table 2.1. Absolute Maximum Ratings 23
3. Global DC Electrical Characteristics
Table 3.1. Global DC Electrical Characteristics 24
4. Pinout and Package Definitions
Table 4.1. Pin Definitions for the C8051F326/7 25
Table 4.2. QFN-28 Package Dimensions 29
Table 4.3. QFN-28 PCB Land Pattern Dimesions 30
5. Voltage Regulator (REG0)
Table 5.1. Voltage Regulator Electrical Specifications 31
6. CIP-51 Microcontroller
Table 6.1. CIP-51 Instruction Set Summary 37
Table 6.2. Special Function Register (SFR) Memory Map 43
Table 6.3. Special Function Registers 43
Table 6.4. TMOD.3 Control of /INT0 49
Table 6.5. Interrupt Summary 50
7. Reset Sources
Table 7.1. Reset Electrical Characteristics 62
8. Flash Memory
Table 8.1. Flash Electrical Characteristics 64
9. External RAM
10. Oscillators
Table 10.1. Typical USB Full Speed Clock Settings 76
Table 10.2. Typical USB Low Speed Clock Settings 76
Table 10.3. Internal Oscillator Electrical Characteristics 78
11. Port Input/Output
Table 11.1. Port I/O DC Electrical Characteristics (C8051F326) 85
Table 11.2. Port I/O DC Electrical Characteristics (C8051F327) 85
12. Universal Serial Bus Controller (USB0)
Table 12.1. Endpoint Addressing Scheme 88
Table 12.2. USB0 Controller Registers 93
Table 12.3. FIFO Configurations 95
Table 12.4. USB Transceiver Electrical Characteristics 115
13. UART0
Table 13.1. Baud Rate Generator Settings for Standard Baud Rates 119
14. Timers
Table 14.1. Timer Modes 127
Table 14.2. Timer 0 Operation 128
15. C2 Interface

List of Registers

SFR Definition 5.1. REG0CN: Voltage Regulator Control 34
SFR Definition 6.1. DPL: Data Pointer Low Byte 45
SFR Definition 6.2. DPH: Data Pointer High Byte 45
SFR Definition 6.3. SP: Stack Pointer 45
SFR Definition 6.4. PSW: Program Status Word 46
SFR Definition 6.5. ACC: Accumulator 46
SFR Definition 6.6. B: B Register 47
SFR Definition 6.7. IE: Interrupt Enable 51
SFR Definition 6.8. IP: Interrupt Priority 52
SFR Definition 6.9. EIE1: Extended Interrupt Enable 1 53
SFR Definition 6.10. EIP1: Extended Interrupt Priority 1 53
SFR Definition 6.11. EIE2: Extended Interrupt Enable 2 53
SFR Definition 6.12. EIP2: Extended Interrupt Priority 2 54
SFR Definition 6.13. PCON: Power Control 56
SFR Definition 7.1. VDM0CN: VDD Monitor Control 59
SFR Definition 7.2. RSTSRC: Reset Source 61
SFR Definition 8.1. PSCTL: Program Store R/W Control 66
SFR Definition 8.2. FLKEY: Flash Lock and Key 67
SFR Definition 8.3. FLSCL: Flash Scale 67
SFR Definition 9.1. EMI0CN: External Memory Interface Control 70
SFR Definition 10.1. OSCICN: Internal Oscillator Control 72
SFR Definition 10.2. OSCICL: Internal Oscillator Calibration 73
SFR Definition 10.3. OSCLCN: Internal L-F Oscillator Control 74
SFR Definition 10.4. CLKMUL: Clock Multiplier Control 75
SFR Definition 10.5. CLKSEL: Clock Select 77
SFR Definition 11.1. GPIOCN: Global Port I/O Control 82
SFR Definition 11.2. P0: Port0 82
SFR Definition 11.3. P0MDOUT: Port0 Output Mode 82
SFR Definition 11.4. P2: Port2 83
SFR Definition 11.5. P2MDOUT: Port2 Output Mode 83
SFR Definition 11.6. P3: Port3 83
SFR Definition 11.7. P3MDOUT: Port3 Output Mode 84
USB Register Definition 12.1. USB0XCN: USB0 Transceiver Control 89
USB Register Definition 12.2. USB0ADR: USB0 Indirect Address 91
USB Register Definition 12.3. USB0DAT: USB0 Data 92
USB Register Definition 12.4. INDEX: USB0 Endpoint Index 92
USB Register Definition 12.5. CLKREC: Clock Recovery Control 94
USB Register Definition 12.6. FIFOn: USB0 Endpoint FIFO Access 96
USB Register Definition 12.7. FADDR: USB0 Function Address 97
USB Register Definition 12.8. POWER: USB0 Power 99
USB Register Definition 12.9. FRAMEL: USB0 Frame Number Low 100
USB Register Definition 12.10. FRAMEH: USB0 Frame Number High 100
USB Register Definition 12.11. IN1INT: USB0 IN Endpoint Interrupt 101

USB Register Definition 12.12. OUT1INT: USB0 Out Endpoint Interrupt 101
USB Register Definition 12.13. CMINT: USB0 Common Interrupt 102
USB Register Definition 12.14. IN1IE: USB0 IN Endpoint Interrupt Enable 102
USB Register Definition 12.15. OUT1IE: USB0 Out Endpoint Interrupt Enable 103
USB Register Definition 12.16. CMIE: USB0 Common Interrupt Enable 103
USB Register Definition 12.17. E0CSR: USB0 Endpoint0 Control 106
USB Register Definition 12.18. E0CNT: USB0 Endpoint 0 Data Count 107
USB Register Definition 12.19. EINCSRL: USB0 IN Endpoint Control Low Byte 110
USB Register Definition 12.20. EINCSRH: USB0 IN Endpoint Control High Byte . . . 111
USB Register Definition 12.21. EOUTCSRL: USB0 OUT Endpoint Control Low Byte 113
USB Register Definition 12.22. EOUTCSRH: USB0 OUT
Endpoint Control High Byte 114
USB Register Definition 12.23. EOUTCNTL: USB0 OUT Endpoint Count Low 114
USB Register Definition 12.24. EOUTCNTH: USB0 OUT Endpoint Count High 114
SFR Definition 13.1. SCON0: UART0 Control 123
SFR Definition 13.2. SMOD0: UART0 Mode 124
SFR Definition 13.3. SBUF0: UART0 Data Buffer 125
SFR Definition 13.4. SBCON0: UART0 Baud Rate Generator Control 125
SFR Definition 13.5. SBRLH0: UART0 Baud Rate Generator High Byte 126
SFR Definition 13.6. SBRLL0: UART0 Baud Rate Generator Low Byte 126
SFR Definition 14.1. TCON: Timer Control 131
SFR Definition 14.2. TMOD: Timer Mode 132
SFR Definition 14.3. CKCON: Clock Control 133
SFR Definition 14.4. TL0: Timer 0 Low Byte 134
SFR Definition 14.5. TL1: Timer 1 Low Byte 134
SFR Definition 14.6. TH0: Timer 0 High Byte 134
SFR Definition 14.7. TH1: Timer 1 High Byte 134
C2 Register Definition 15.1. C2ADD: C2 Address 135
C2 Register Definition 15.2. DEVICEID: C2 Device ID 135
C2 Register Definition 15.3. REVID: C2 Revision ID 136
C2 Register Definition 15.4. FPCTL: C2 Flash Programming Control 136
C2 Register Definition 15.5. FPDAT: C2 Flash Programming Data 136

1. System Overview

C8051F326/7 devices are fully integrated mixed-signal system-on-a-chip MCUs. Highlighted features are listed below. Refer to Table 1.1 for specific product feature selection.

- High-speed pipelined 8051-compatible microcontroller core (up to 25 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- Universal serial bus (USB) function controller with three fixed-function endpoint pipes, integrated transceiver, and 256B FIFO RAM
- Supply voltage regulator
- Precision programmable 12 MHz internal oscillator and 4x clock multiplier
- 16k kB of on-chip Flash memory
- 1536 total bytes of on-chip RAM (256 + 1 k + 256 USB FIFO)
- Enhanced UART, serial interfaces implemented in hardware
- Two general-purpose 16-bit timers
- On-chip power-on reset, VDD monitor, and missing clock detector
- 15 Port I/O (5 V tolerant)

With on-chip power-on reset, VDD monitor, voltage regulator, and clock oscillator, C8051F326/7 devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

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

Each device is specified for 2.7–5.25 V operation over the industrial temperature range (–40 to +85 °C). For voltages above 3.6 V, the on-chip Voltage Regulator must be used. A minimum of 3.0 V is required for USB communication. The Port I/O and RST pins are tolerant of input signals up to 5 V. C8051F326/7 are available in two 28-pin QFN packages with different pinouts. The RoHS compliant devices are marked with a -GM suffix in the part number. The port I/O on C8051F326 devices is powered from a separate I/O supply allowing it to interface to low voltage logic.

Table 1.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	Flash Memory	RAM	Calibrated Internal Oscillator	USB	Supply Voltage Regulator	UART	Timers (16-bit)	Digital Port I/Os	Separate I/O Supply	Package
C8051F326-GM	25	16k	1536	✓	✓	✓	✓	2	15	✓	QFN-28
C8051F327-GM	25	16k	1536	✓	✓	✓	✓	2	15	—	QFN-28

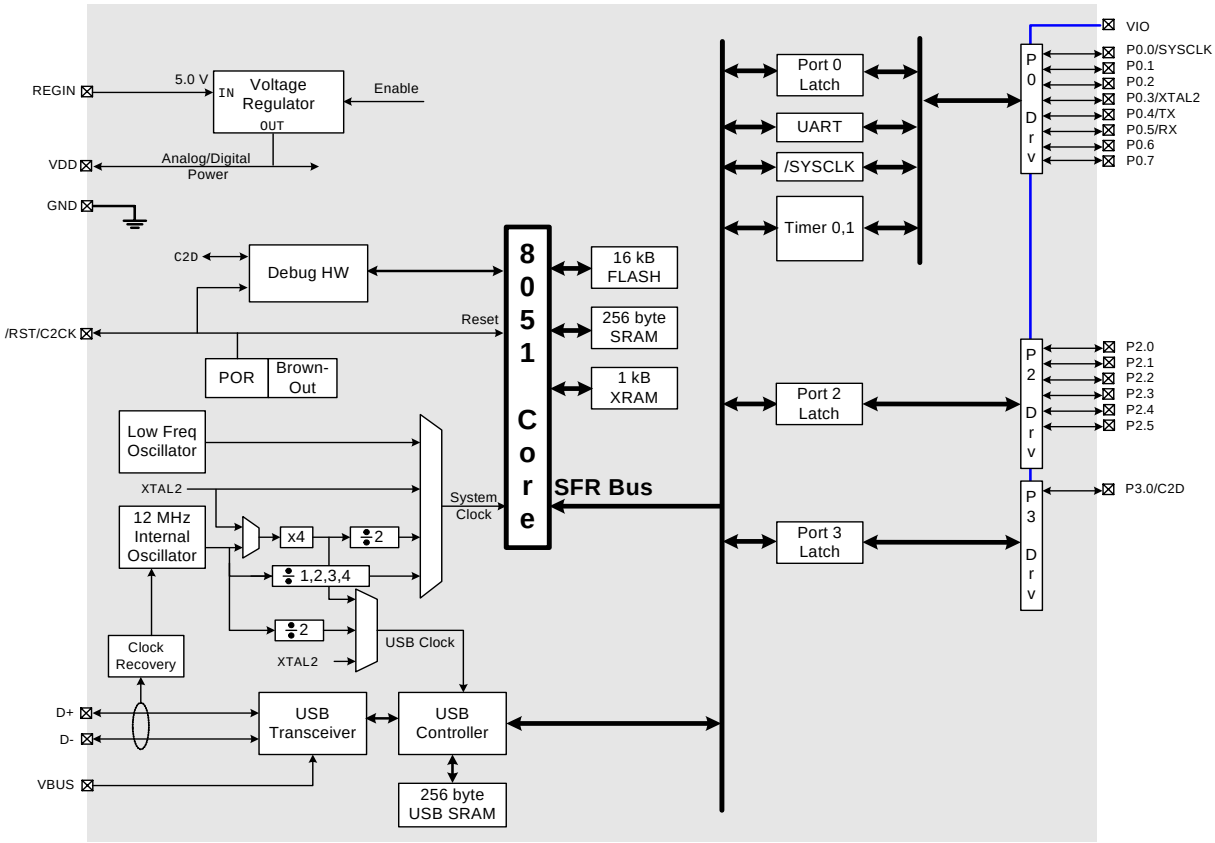


Figure 1.1. C8051F326 Block Diagram

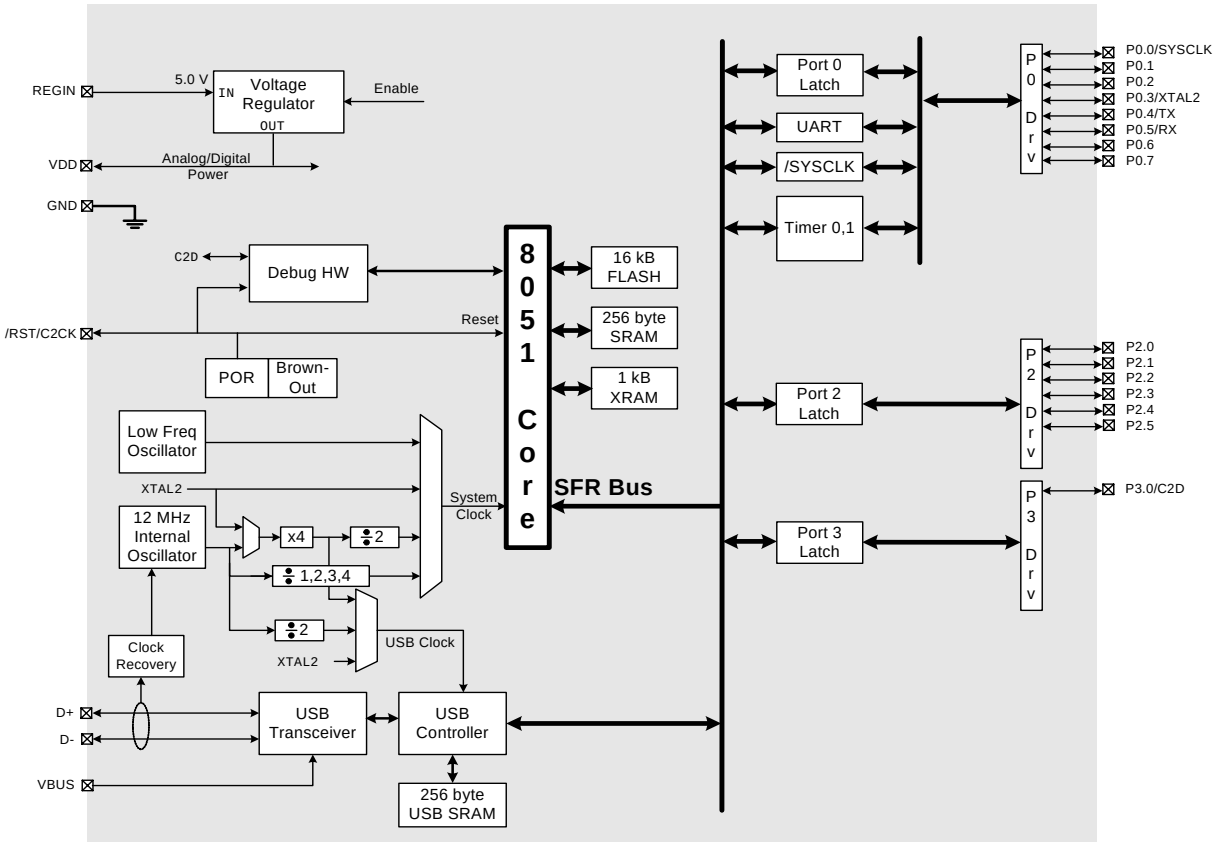


Figure 1.2. C8051F327 Block Diagram

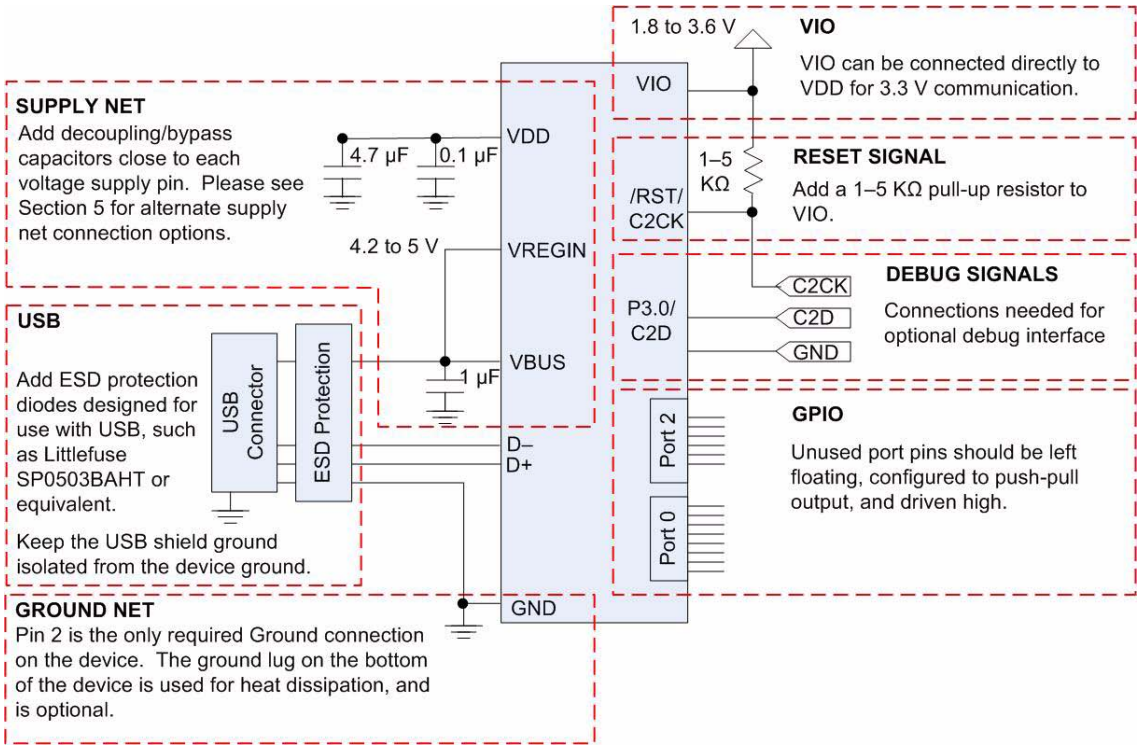


Figure 1.3. Typical Connections for the C8051F326

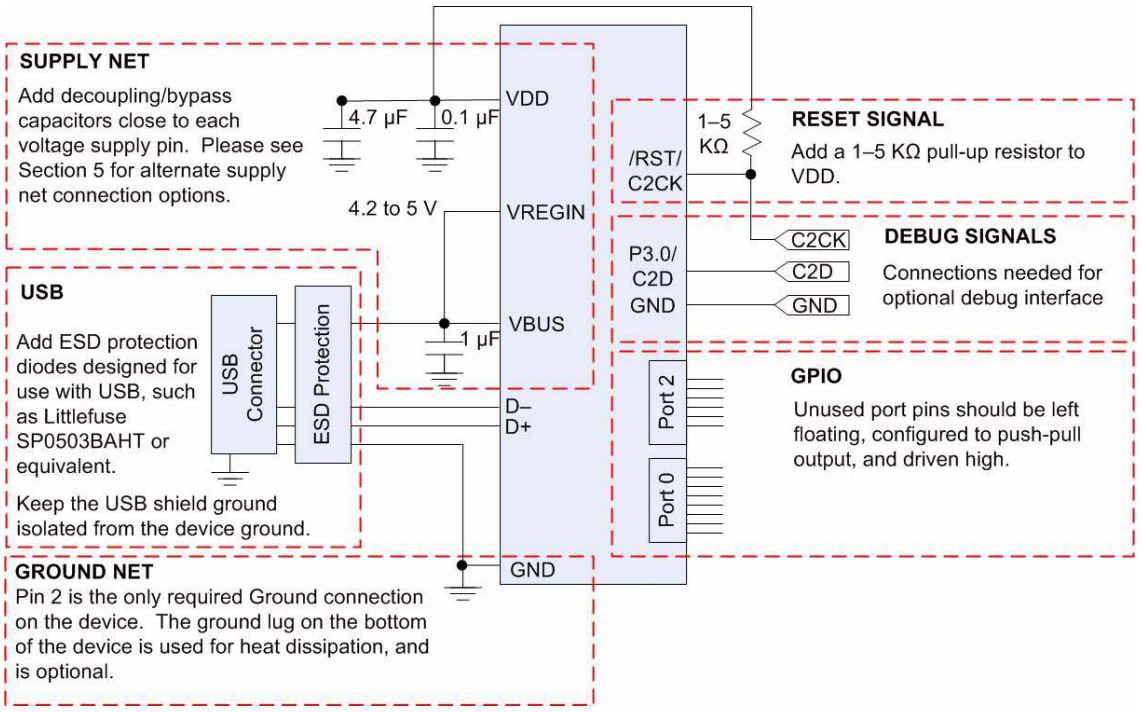


Figure 1.4. Typical Connections for the C8051F327

1.1. CIP-51™ Microcontroller Core

1.1.1. Fully 8051 Compatible

The C8051F326/7 family utilizes Silicon Laboratories' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The CIP-51 core offers all the peripherals included with a standard 8052, including two 16-bit counter/timers, a full-duplex UART with extended baud rate configuration, 1536 bytes of on-chip RAM, 128 byte Special Function Register (SFR) address space, and 15 I/O pins.

1.1.2. Improved Throughput

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

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

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

With the CIP-51's maximum system clock at 25 MHz, it has a peak throughput of 25 MIPS. Figure 1.5 shows a comparison of peak throughputs for various 8-bit microcontroller cores with their maximum system clocks.

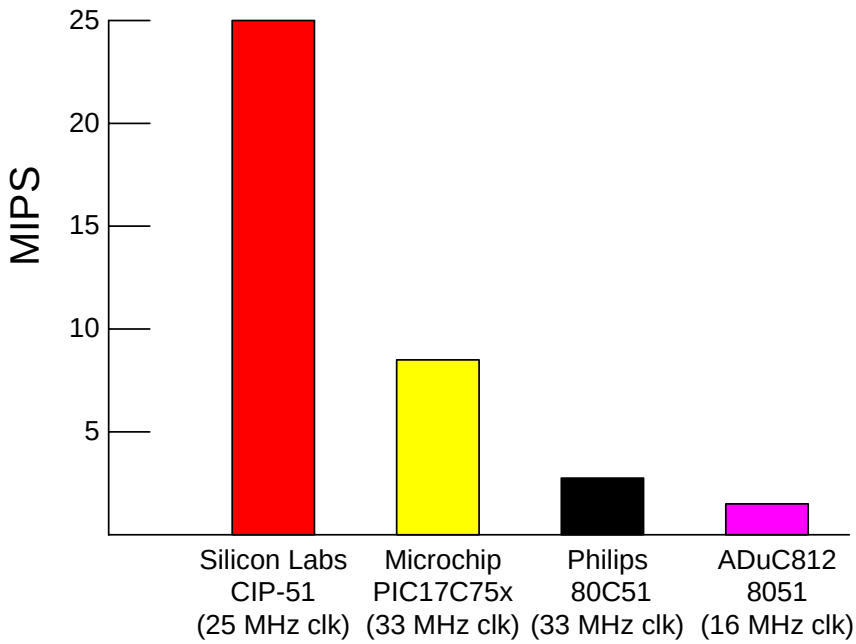


Figure 1.5. Comparison of Peak MCU Execution Speeds

1.1.3. Additional Features

The C8051F326/7 SoC family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

The extended interrupt handler provides 8 interrupt sources into the CIP-51. An interrupt driven system requires less intervention by the MCU, giving it more effective throughput. The interrupt sources are very useful when building multi-tasking, real-time systems.

Seven reset sources are available: power-on reset circuitry (POR), an on-chip VDD monitor (forces reset when power supply voltage drops below V_{RST} as given in Table 7.1 on page 62), the USB controller (USB bus reset or a VBUS transition), a Missing Clock Detector, a forced software reset, an external reset pin, and an errant Flash read/write protection circuit. Each reset source except for the POR, Reset Input Pin, or Flash error may be disabled by the user in software.

The internal oscillator is factory calibrated to 12 MHz ±1.5%, and the internal oscillator period may be user programmed in ~0.25% increments. An additional low-frequency oscillator is also available which facilitates low power operation. A clock recovery mechanism allows the internal oscillator to be used with the 4x Clock Multiplier as the USB clock source in Full Speed mode; the internal oscillator can also be used as the USB clock source in Low Speed mode. An external CMOS clock may also be used with the 4x Clock Multiplier. The system clock may be configured to use the internal oscillator, external clock, low-frequency oscillator, or the Clock Multiplier output divided by 2. If desired, the system clock source may be switched on-the-fly between oscillator sources. The external clock and internal low-frequency oscillator can be extremely useful in low power applications, allowing the MCU to run from a slow (power saving) clock source, while periodically switching to the high-frequency internal oscillator as needed.

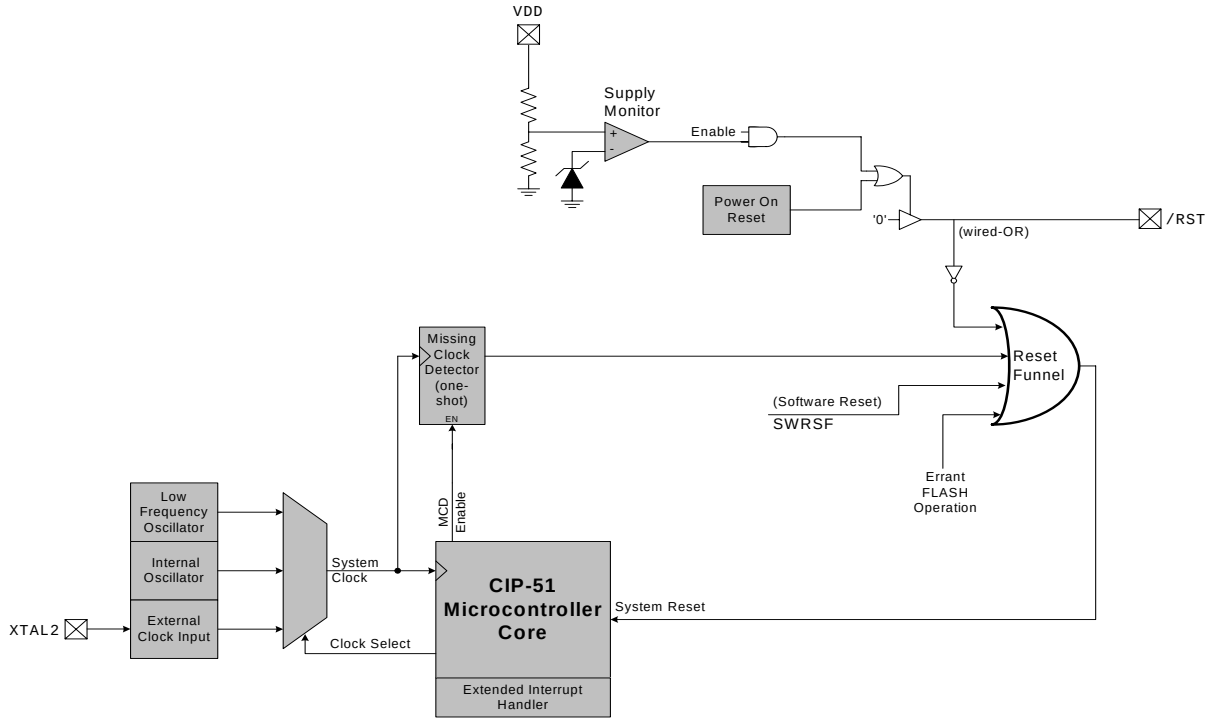


Figure 1.6. On-Chip Clock and Reset

1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

Program memory consists of 16k bytes of Flash. This memory may be reprogrammed in-system in 512 byte sectors, and requires no special off-chip programming voltage. See Figure 1.7 for the MCU system memory map.

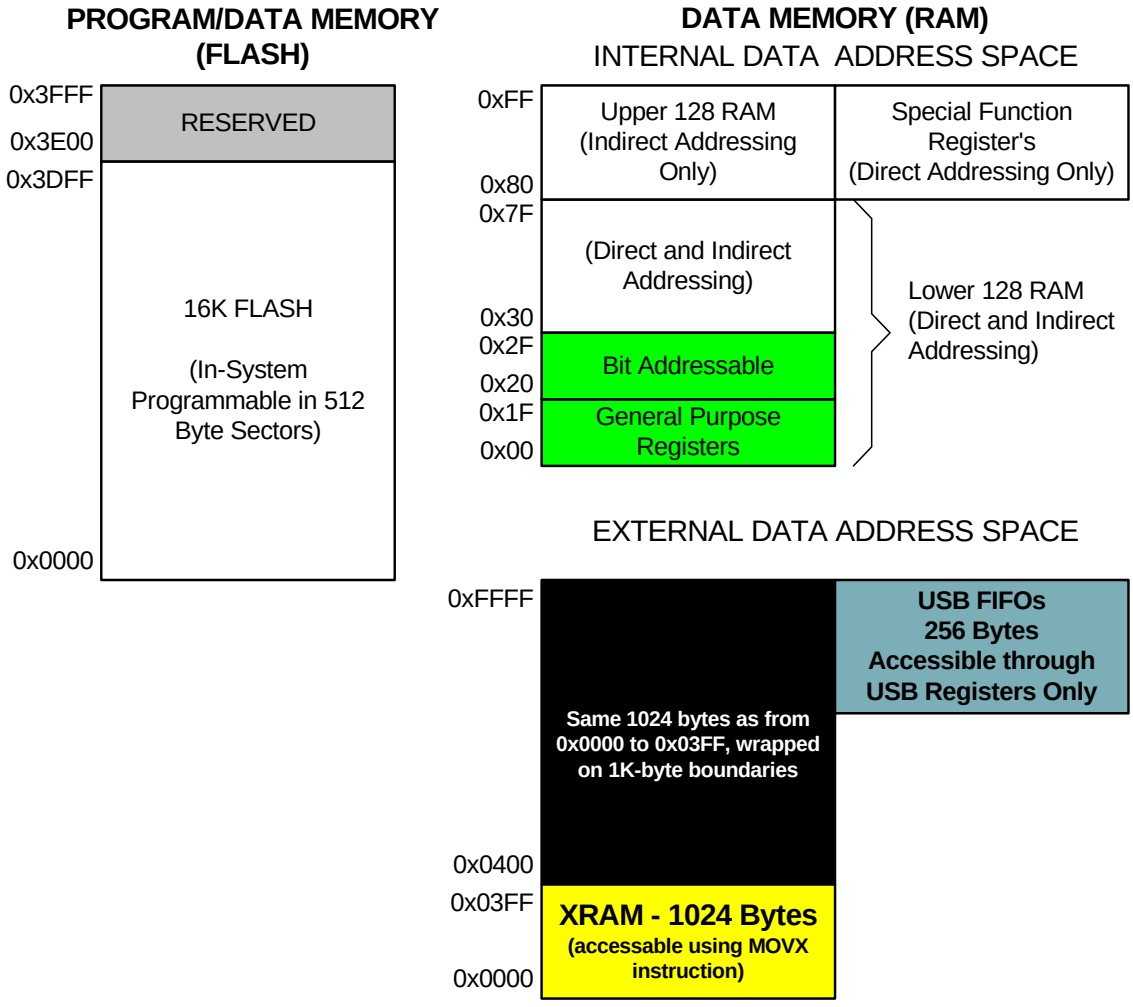


Figure 1.7. On-Board Memory Map

1.3. Universal Serial Bus Controller

The Universal Serial Bus Controller (USB0) is a USB 2.0 peripheral with integrated transceiver and endpoint FIFO RAM. The controller supports both full and low speed modes. A total of three endpoint pipes are available: a bi-directional control endpoint (Endpoint0) and a data endpoint (Endpoint1) with one IN pipe and one OUT pipe.

A 256 block of XRAM is used as dedicated USB FIFO space. This FIFO space is distributed between Endpoint0 and Endpoint1. Endpoint0 is 64 bytes, and Endpoint1 has a 64 byte IN pipe and a 128 byte OUT pipe.

USB0 can be operated as a Full or Low Speed function. The on-chip 4x Clock Multiplier and clock recovery circuitry allow both Full and Low Speed options to be implemented with the on-chip precision oscillator as the USB clock source. An external clock source can also be used with the 4x Clock Multiplier to generate the USB clock.

The USB Transceiver is USB 2.0 compliant, and includes on-chip matching and pullup resistors. The pullup resistors can be enabled/disabled in software, and will appear on the D+ or D– pin according to the software-selected speed setting (full or low speed).

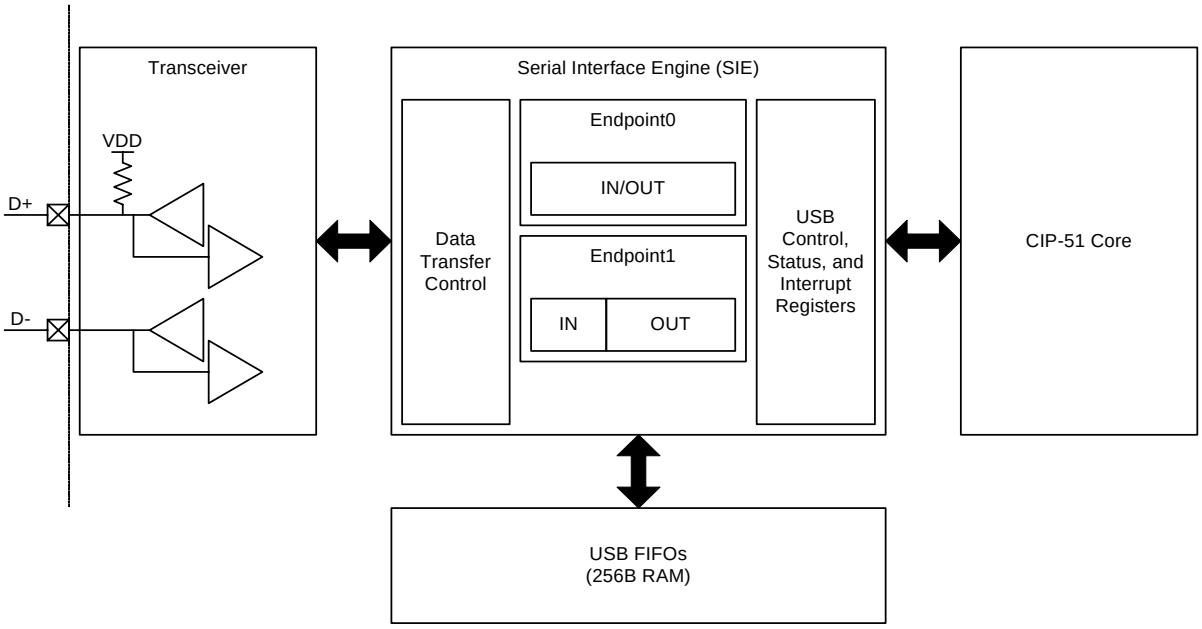


Figure 1.8. USB Controller Block Diagram

1.4. Voltage Regulator

C8051F326/7 devices include a voltage regulator (REG0). When enabled, the REG0 output appears on the VDD pin and can be used to power external devices. REG0 can be enabled/disabled by software.

1.5. On-Chip Debug Circuitry

C8051F326/7 devices include on-chip Silicon Laboratories 2-Wire (C2) debug circuitry that provides non-intrusive, full speed, in-circuit debugging of the production part *installed in the end application.*

The Silicon Laboratories' debugging system supports inspection and modification of memory and registers, breakpoints, and single stepping. No additional target RAM, program memory, timers, or communications channels are required. All the digital and analog peripherals are functional and work correctly while debugging. All the peripherals (except for the USB) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized.

The C8051F326DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F326/7 MCUs. The kit includes a Windows development environment, a serial adapter for connecting to the C2 port, and a target application board. All of the necessary communication cables and a wall-mount power supply are also supplied with the development kit. The Silicon Laboratories debug environment is a vastly superior configuration for developing and debugging embedded applications compared to standard MCU emulators, which use on-board "ICE Chips" and target cables and require the MCU in the application board to be socketed. The Silicon Laboratories debug environment enhances ease of use and preserves the performance of on-chip peripherals.

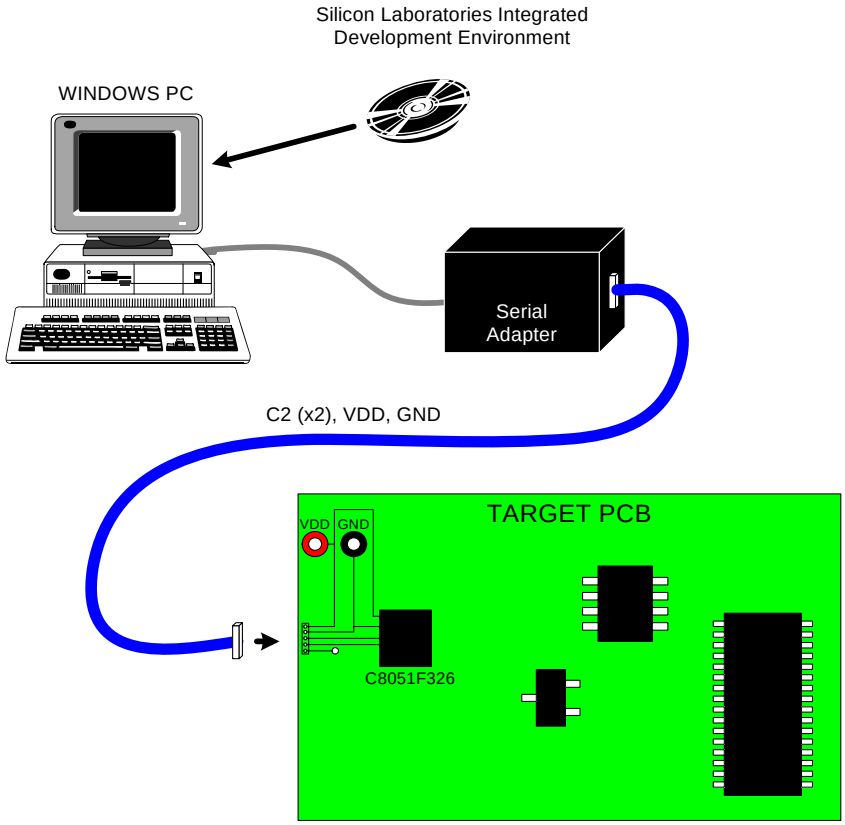


Figure 1.9. Development/In-System Debug Diagram

1.6. Programmable Digital I/O

C8051F326/7 devices include 15 I/O pins (one byte-wide Port, one 6-bit-wide and one 1-bit-wide Port). The C8051F326/7 Ports behave like typical 8051 Ports with a few enhancements. Each Port pin may be configured as a digital input or output pin. Pins selected as digital outputs may additionally be configured for push-pull or open-drain output. The “weak pullups” that are fixed on typical 8051 devices may be globally disabled, providing power savings capabilities.

1.7. Serial Ports

The C8051F326/7 Family includes a full-duplex UART with enhanced baud rate configuration. The serial interface is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little CPU intervention.

2. Absolute Maximum Ratings

Table 2.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient Temperature under Bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on any Port I/O Pin or $\overline{\text{RST}}$ with Respect to GND		–0.3	—	5.8	V
Voltage on VDD or VIO with Respect to GND		–0.3	—	4.2	V
Maximum Total Current through VDD, VIO, and GND		—	—	500	mA
Maximum Output Current Sunk by $\overline{\text{RST}}$ or any Port Pin		—	—	100	mA

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

3. Global DC Electrical Characteristics

Table 3.1. Global DC Electrical Characteristics

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

Parameter	Conditions	Min	Typ	Max	Units
I/O Supply Voltage (VIO)[1,2]		1.8	3.3	3.6	V
Core Supply Voltage (VDD)[3]		2.7	3.3	3.6	V
Core Supply Current with CPU Active	VDD = 3.3 V, Clock = 24 MHz VDD = 3.3 V, Clock = 3 MHz VDD = 3.3 V, Clock = 32 kHz	— — —	11 1.9 20	— — —	mA mA µA
Core Supply Current with CPU Inactive (not accessing Flash)	VDD = 3.3 V, Clock = 24 MHz VDD = 3.3 V, Clock = 3 MHz VDD = 3.3 V, Clock = 32 kHz	— — —	4.4 0.83 13	— — —	mA mA µA
Digital Supply Current (suspend mode or shutdown mode)	Oscillator not running	—	< 0.1	—	µA
Digital Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock)[4]		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. The I/O Supply Voltage (VIO) must be less than or equal to the Core Supply Voltage (VDD).
2. For C8051F327 devices, VIO is internally connected to VDD.
3. USB Requires 3.0 V Minimum Core Supply Voltage (VDD).
4. SYSCLK must be at least 32 kHz to enable debugging.

4. Pinout and Package Definitions

Table 4.1. Pin Definitions for the C8051F326/7

Name	Pin Numbers		Type	Description
	'F326	'F327		
VDD	6	6	Power In Power Out	2.7–3.6 V Core Supply Voltage Input. 3.3 V Voltage Regulator Output. See Section 5.
VIO	5	—	Power In	V I/O Supply Voltage Input. The voltage at this pin must be less than or equal to the Core Supply Voltage (V_{DD}) for the 'F326. On the 'F327, this pin is internally connected to V_{DD}.
GND	2	3		Ground.
$\overline{\text{RST}}$/ C2CK	9	9	D I/O D I/O	Device Reset. Open-drain output of internal POR or VDD monitor. An external source can initiate a system reset by driving this pin low for at least 15 µs. See Section 7. Clock signal for the C2 Debug Interface.
P3.0/ C2D	10	10	D I/O D I/O	Port 3.0. See Section 11 for a complete description. Bi-directional data signal for the C2 Debug Interface.
REGIN	7	7	Power In	5 V Regulator Input. This pin is the input to the on-chip voltage regulator.
VBUS	8	8	D In	VBUS Sense Input. This pin should be connected to the VBUS signal of a USB network. A 5 V signal on this pin indicates a USB network connection.
D+	3	4	D I/O	USB D+.
D–	4	5	D I/O	USB D–.
P0.0	1	2	D I/O	Port 0.0. See Section 11 for a complete description.
P0.1	28	1	D I/O	Port 0.1. See Section 11 for a complete description.
P0.2	27	28	D I/O	Port 0.2. See Section 11 for a complete description.
P0.3/ XTAL2	26	27	D I/O D In	Port 0.3. See Section 11 for a complete description. External Clock Input. See Section 10 for a complete description.
P0.4	25	26	D I/O	Port 0.4. See Section 11 for a complete description.
P0.5	24	25	D I/O	Port 0.5. See Section 11 for a complete description.

Table 4.1. Pin Definitions for the C8051F326/7 (Continued)

Name	Pin Numbers		Type	Description
	‘F326	‘F327		
P0.6	23	24	D I/O	Port 0.6. See Section 11 for a complete description.
P0.7	22	23	D I/O	Port 0.7. See Section 11 for a complete description.
P2.0	19	19	D I/O	Port 2.0. See Section 11 for a complete description.
P2.1	18	18	D I/O	Port 2.1. See Section 11 for a complete description.
P2.2	12	12	D I/O	Port 2.2. See Section 11 for a complete description.
P2.3	11	11	D I/O	Port 2.3. See Section 11 for a complete description.
P2.4	17	17	D I/O	Port 2.4. See Section 11 for a complete description.
P2.5	16	16	D I/O	Port 2.5. See Section 11 for a complete description.
N.C. pins for the ‘F326: 13, 14, 15, 20, and 21. N.C. pins for the ‘F327: 13, 14, 15, 20, 21, and 22.				

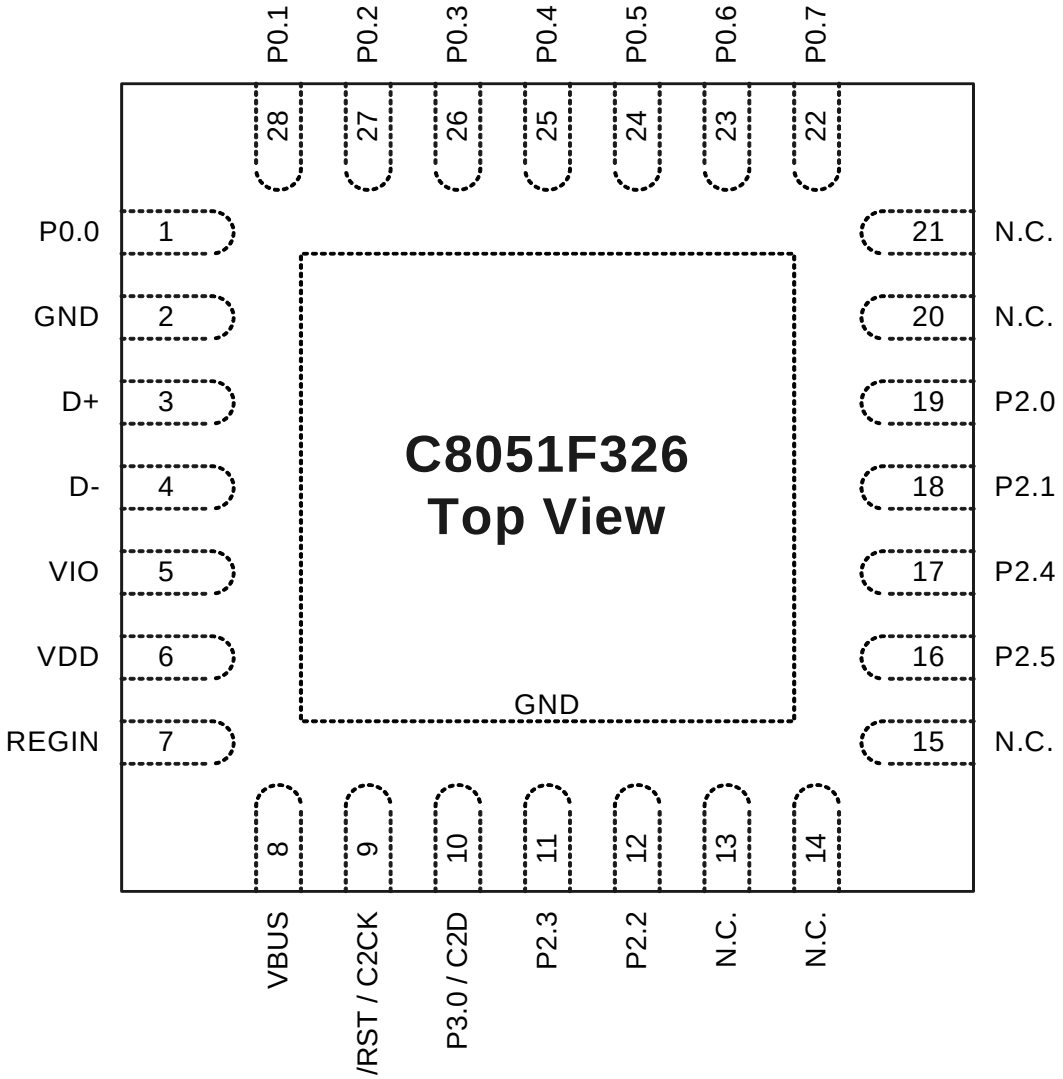


Figure 4.1. C8051F326 QFN-28 Pinout Diagram (Top View)

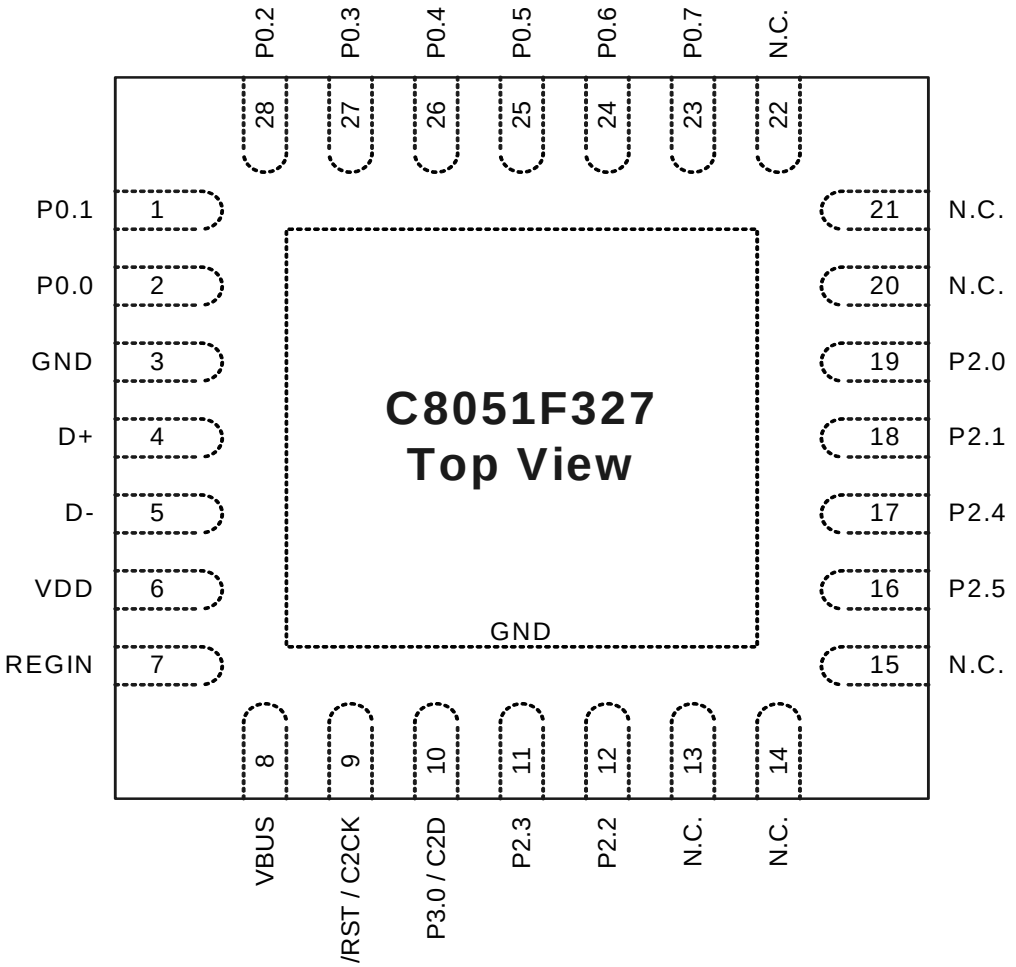


Figure 4.2. C8051F327 QFN-28 Pinout Diagram (Top View)

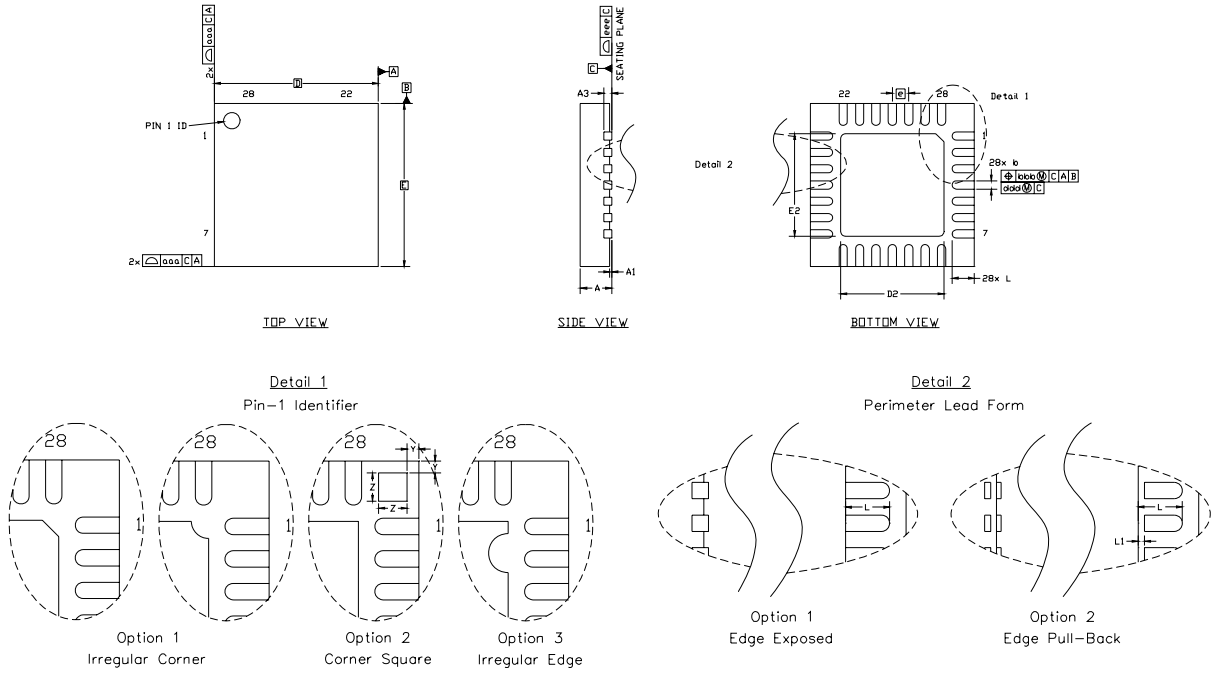


Figure 4.3. QFN-28 Package Drawing

Table 4.2. QFN-28 Package Dimensions

Dimension	Min	Typ	Max
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
A3		0.25 REF	
b	0.18	0.23	0.30
D		5.00 BSC.	
D2	2.90	3.15	3.35
e		0.50 BSC.	
E		5.00 BSC.	
E2	2.90	3.15	3.35

Dimension	Min	Typ	Max
L	0.35	0.55	0.65
L1	0.00	—	0.15
aaa		0.15	
bbb		0.10	
ddd		0.05	
eee		0.08	
Z		0.44	
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 VHHD 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-020C specification for Small Body Components.

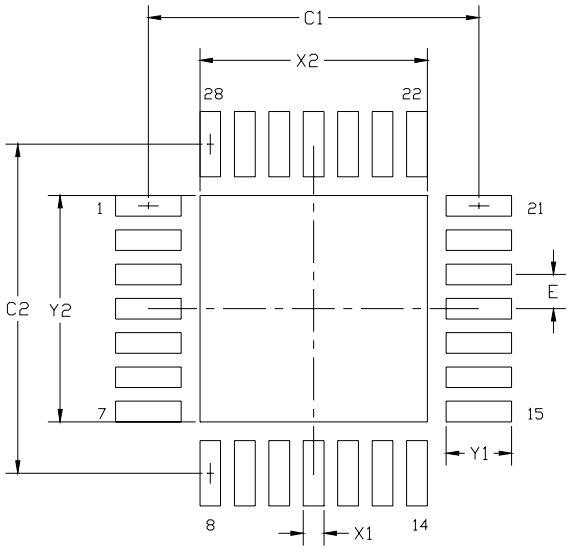


Figure 4.4. QFN-28 Recommended PCB Land Pattern

Table 4.3. QFN-28 PCB Land Pattern Dimensions

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

Dimension	Min	Max
X2	3.20	3.30
Y1	0.85	0.95
Y2	3.20	3.30

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

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

Stencil Design

5. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
6. The stencil thickness should be 0.125mm (5 mils).
7. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pins.
8. A 3x3 array of 0.90mm openings on a 1.1mm pitch should be used for the center pad to assure the proper paste volume (67% Paste Coverage).

Card Assembly

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

5. Voltage Regulator (REG0)

C8051F326/7 devices include a voltage regulator (REG0). When enabled, the REG0 output appears on the VDD pin and can be used to power external devices. REG0 can be enabled/disabled by software using bit REGEN in register REG0CN. See Table 5.1 for REG0 electrical characteristics.

The voltage regulator is enabled on reset. When the device is self-powered from a 3V supply net, the regulator may be disabled in order to save power. **Important Note: If the voltage at the regulator input (REGIN) is greater than the Core Supply Voltage (VDD), the voltage regulator should not be disabled. Otherwise, permanent damage to the device may occur.**

Note that the VBUS signal must be connected to the VBUS pin when using the device in a USB network. The VBUS signal should only be connected to the REGIN pin when operating the device as a bus-powered function. REG0 configuration options are shown in Figure 5.1 - Figure 5.4.

5.1. Regulator Mode Selection

REG0 offers a low power mode intended for use when the device is in suspend mode. In this low power mode, the REG0 output remains as specified; however the REG0 dynamic performance (response time) is degraded. See Table 5.1 for normal and low power mode supply current specifications. The REG0 mode selection is controlled via the REGMOD bit in register REG0CN.

5.2. VBUS Detection

When the USB Function Controller is used (see section Section “12. Universal Serial Bus Controller (USB0)” on page 87), the VBUS signal should be connected to the VBUS pin. The VBSTAT bit (register REG0CN) indicates the current logic level of the VBUS signal. If enabled, a VBUS interrupt will be generated when the VBUS signal matches the polarity selected by the VBPOL bit in register REG0CN. The VBUS interrupt is level-sensitive, and has no associated interrupt pending flag. The VBUS interrupt will be active as long as the VBUS signal matches the polarity selected by VBPOL. See Table 5.1 for VBUS input parameters.

Important Note: When USB is selected as a reset source, a system reset will be generated when the VBUS signal matches the polarity selected by the VBPOL bit. See Section “7. Reset Sources” on page 57 for details on selecting USB as a reset source.

Table 5.1. Voltage Regulator Electrical Specifications

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

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range		2.7	—	5.25	V
Output Voltage	Output Current = 1 to 100 mA	3.0	3.3	3.6	V
VBUS Detection Input Threshold		1.0	1.8	2.9	V
Bias Current	Normal Mode (REGMOD = ‘0’) Low Power Mode (REGMOD = ‘1’)	— —	75 41	111 61	µA
Dropout Voltage (V_{DO})*	IDD = 1 to 100 mA	—	1	—	mV/mA
***Note:** The minimum input voltage is 2.70 V or V_{DD} + V_{DO} (max load), whichever is greater.					

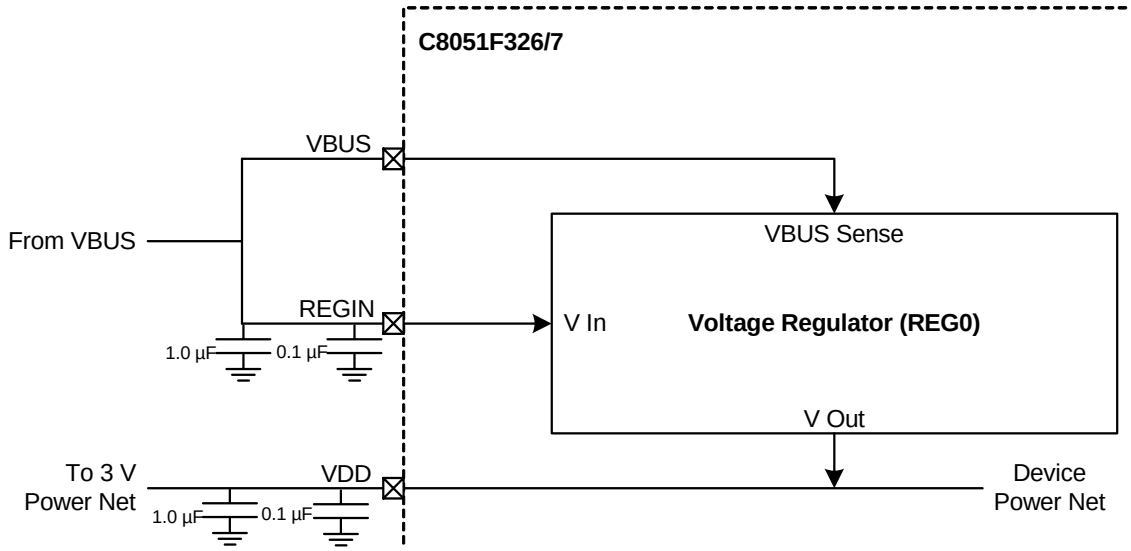


Figure 5.1. REG0 Configuration: USB Bus-Powered

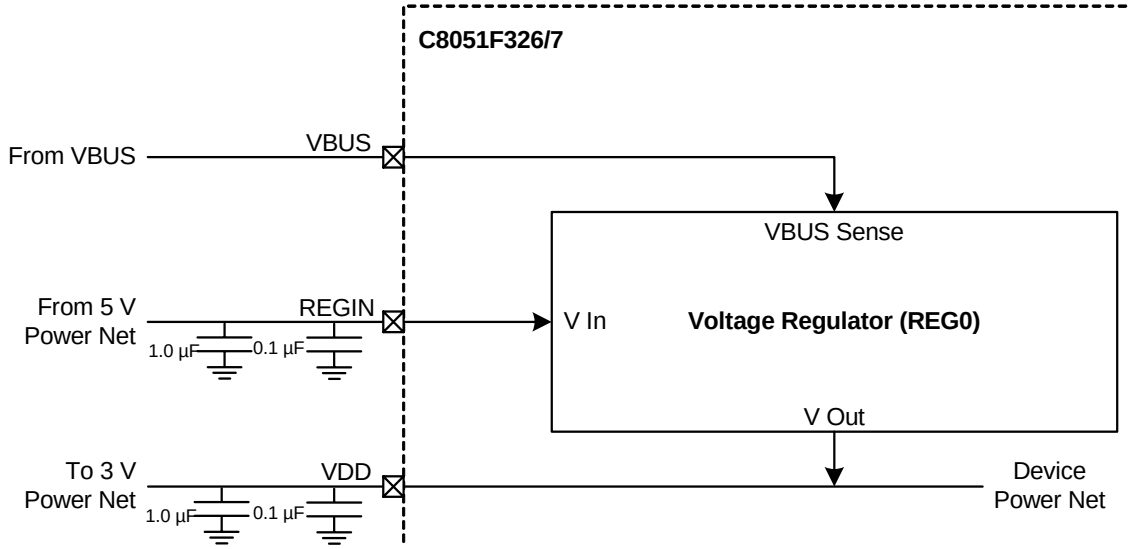


Figure 5.2. REG0 Configuration: USB Self-Powered

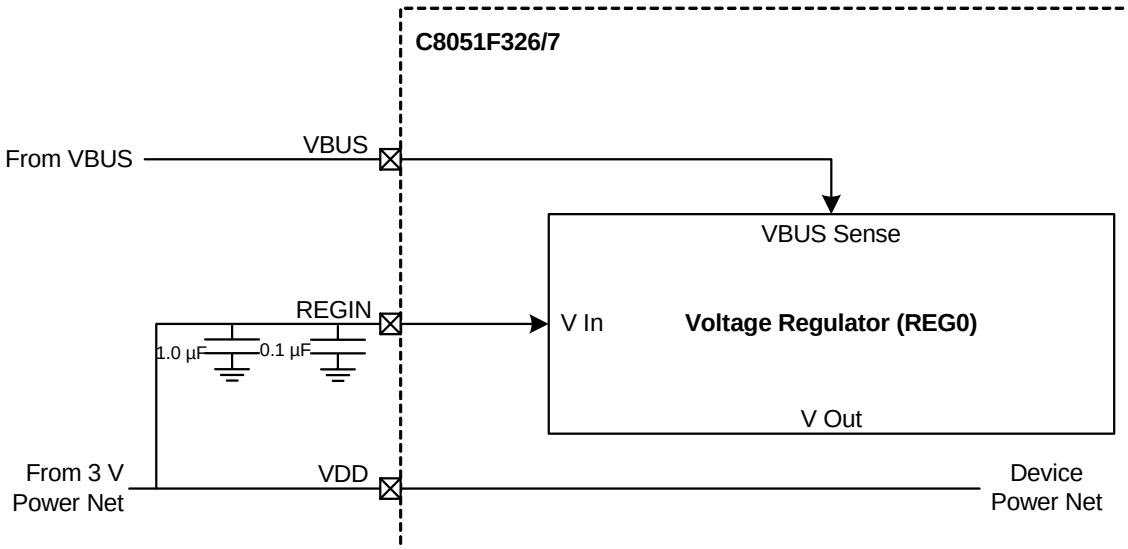


Figure 5.3. REG0 Configuration: USB Self-Powered, Regulator Disabled

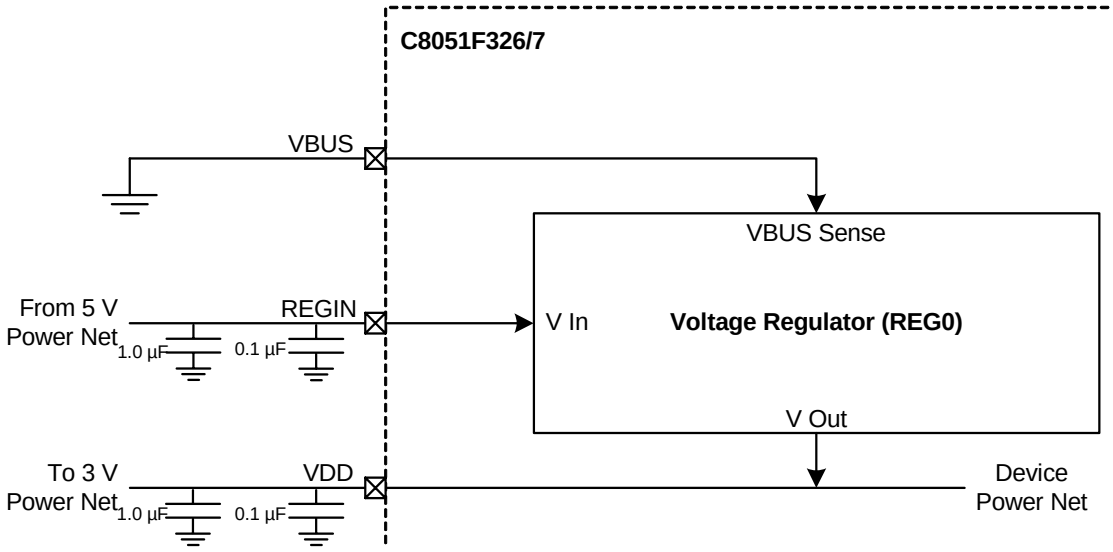


Figure 5.4. REG0 Configuration: No USB Connection

SFR Definition 5.1. REG0CN: Voltage Regulator Control

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

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

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

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

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

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

6. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. Included are two 16-bit counter/timers (see description in Section “14. Timers” on page 127), an enhanced full-duplex UART (see description in Section “13. UART0” on page 117), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (Section “6.2.6. Special Function Registers” on page 43), and 15 Port I/O (see description in Section “11. Port Input/Output” on page 79). The CIP-51 also includes on-chip debug hardware (see description in Section “15. C2 Interface” on page 135), and interfaces directly with the USB and other digital subsystems providing a complete 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 6.1 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
- 256 Bytes of Internal RAM
- 15 Port I/O
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

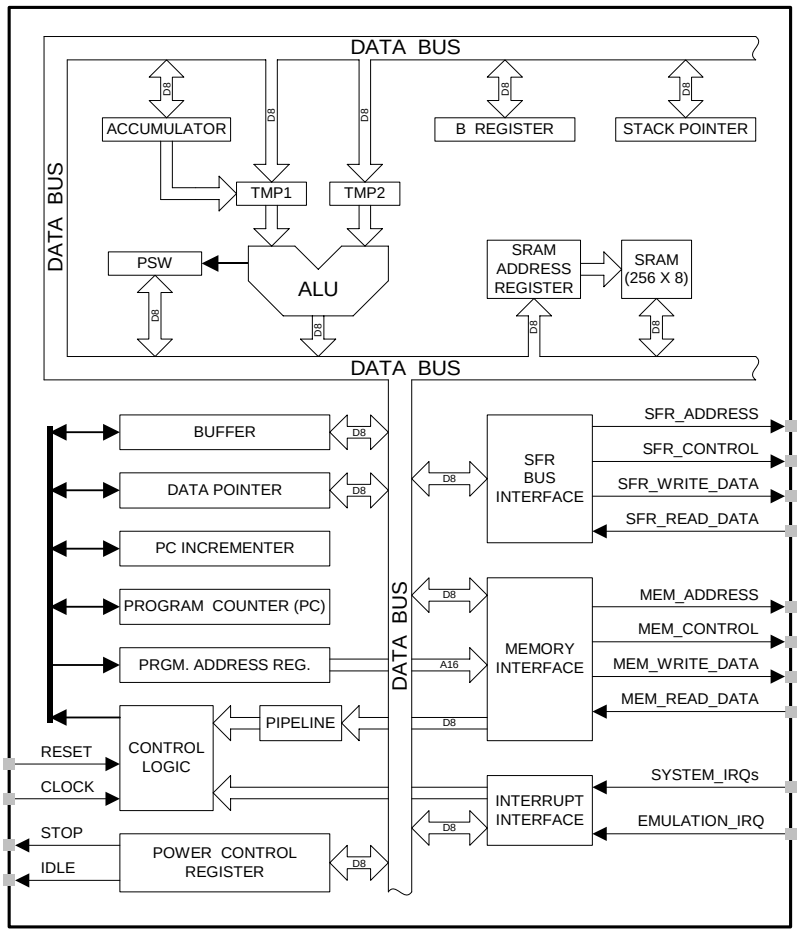


Figure 6.1. CIP-51 Block Diagram

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

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

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

Programming and Debugging Support
In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Laboratories 2-Wire Development Interface (C2). Note that the re-programmable Flash can also be read and changed a single byte at a time by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip 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. C2 details can be found in Section “15. C2 Interface” on page 135.

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

6.1. Instruction Set

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

6.1.1. Instruction and CPU Timing

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

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take one less clock cycle to complete when the branch is not taken as opposed to when the branch is taken. Table 6.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

6.1.2. MOVX Instruction and Program Memory

The MOVX instruction is typically used to access external data memory (Note: the C8051F326/7 does not support off-chip data or program memory). In the CIP-51, the MOVX write instruction is used to accesses external RAM (XRAM) and the on-chip program memory space implemented as re-programmable Flash memory. The Flash access feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section “8. Flash Memory” on page 63 for further details.

Table 6.1. 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

Table 6.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2
Boolean Manipulation			
CLR C	Clear Carry	1	1

Table 6.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
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 (2s complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps. **direct** - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF). **#data** - 8-bit constant **#data16** - 16-bit constant **bit** - Direct-accessed bit in Data RAM or SFR **addr11** - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 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. All mnemonics copyrighted © Intel Corporation 1980.

6.2. Memory Organization

The memory organization of the CIP-51 System Controller is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The CIP-51 memory organization is shown in Figure 6.2.

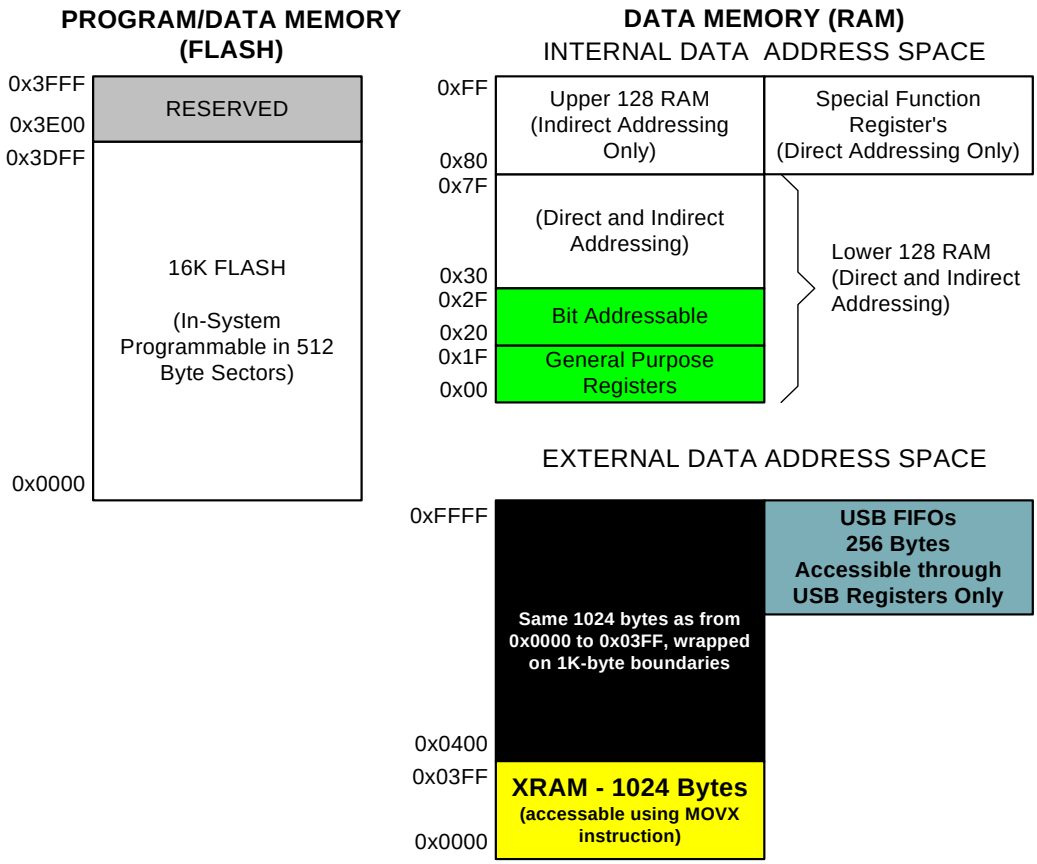


Figure 6.2. Memory Map

6.2.1. Program Memory

The CIP-51 core has a 64 kB program memory space. The C8051F326/7 implements 16k kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x3FFF. Addresses above 0x3DFF are reserved.

Program memory is normally assumed to be read-only. However, the CIP-51 can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX instruction. This feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to Section “8. Flash Memory” on page 63 for further details.

6.2.2. Data Memory

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

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 6.2 illustrates the data memory organization of the CIP-51.

6.2.3. General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in Figure 6.4). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

6.2.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV     C, 22h.3
```

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

6.2.5. Stack

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

6.2.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 6.2 lists the SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g., P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 6.3, for a detailed description of each register.

Table 6.2. Special Function Register (SFR) Memory Map

F8								VDM0CN
F0	B						EIP1	EIP2
E8								RSTSRC
E0	ACC		GPIOCN	OSCLCN			EIE1	EIE2
D8								
D0	PSW							USB0XCN
C8		REG0CN						
C0								
B8	IP	CLKMUL						
B0	P3		OSCICN	OSCICL			FLSCL	FLKEY
A8	IE	CLKSEL	EMI0CN					
A0	P2				P0MDOUT		P2MDOUT	P3MDOUT
98	SCON0	SBUF0	SMOD0					
90		SBCON0		SBRLL0	SBRLH0		USB0ADR	USB0DAT
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH				PCON
	0(8) (bit addressable)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Table 6.3. Special Function Registers

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

Register	Address	Description	Page
ACC	0xE0	Accumulator	46
B	0xF0	B Register	47
CKCON	0x8E	Clock Control	133
CLKMUL	0x91	Clock Multiplier	75
CLKSEL	0xA9	Clock Select	77
DPH	0x83	Data Pointer High	45
DPL	0x82	Data Pointer Low	45
EIE1	0xE6	Extended Interrupt Enable 1	53

Table 6.3. Special Function Registers (Continued)

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

Register	Address	Description	Page
EIE2	0xE7	Extended Interrupt Enable 2	53
EIP1	0xF6	Extended Interrupt Priority 1	53
EIP2	0xF7	Extended Interrupt Priority 2	54
EMI0CN	0xAA	External Memory Interface Control	70
FLKEY	0xB7	Flash Lock and Key	67
FLSCL	0xB6	Flash Scale	67
GPIOCN	0xE2	Global Port I/O Control	82
IE	0xA8	Interrupt Enable	51
IP	0xB8	Interrupt Priority	52
OSCICL	0xB3	Internal Oscillator Calibration	73
OSCICN	0xB2	Internal Oscillator Control	72
OSCLCN	0xE3	Low Frequency Internal Oscillator Control	74
P0	0x80	Port 0 Latch	82
P0MDOUT	0xA4	Port 0 Output Mode Configuration	82
P2	0xA0	Port 2 Latch	83
P2MDOUT	0xA6	Port 2 Output Mode Configuration	83
P3	0xB0	Port 3 Latch	83
P3MDOUT	0xA7	Port 3 Output Mode Configuration	84
PCON	0x87	Power Control	56
PSCTL	0x8F	Program Store R/W Control	66
PSW	0xD0	Program Status Word	46
RSTSRC	0xEF	Reset Source Configuration/Status	61
SBUF0	0x99	UART0 Data Buffer	125
SBCON0	0x91	Baudrate Generator 0 Control	125
SBRLH0	0x94	Baudrate Generator 0 Reload Value High Byte	126
SBRLL0	0x93	Baudrate Generator 0 Reload Value Low Byte	126
SCON0	0x98	UART0 Control	123
SMOD0	0x9A	UART0 Mode	124
SP	0x81	Stack Pointer	45
TCON	0x88	Timer/Counter Control	131
TH0	0x8C	Timer/Counter 0 High	134
TH1	0x8D	Timer/Counter 1 High	134
TL0	0x8A	Timer/Counter 0 Low	134
TL1	0x8B	Timer/Counter 1 Low	134
TMOD	0x89	Timer/Counter Mode	132
USB0ADR	0x96	Indirect Address Register	91
USB0DAT	0x97	Data Register	92
USB0XCN	0xD7	Transceiver Control	89
VDM0CN	0xFF	VDD Monitor Control	59

6.2.7. Register Descriptions

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

SFR Definition 6.1. DPL: Data Pointer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x82

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

SFR Definition 6.2. DPH: Data Pointer High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x83

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

SFR Definition 6.3. SP: Stack Pointer

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x81

Bits7–0: SP: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 6.4. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xD0

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

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

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

Bits4–3: RS1-RS0: Register Bank Select.
These bits select which register bank is used during register accesses.

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

Bit2: OV: Overflow Flag.
This bit is set to 1 under the following circumstances:
- An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
- A MUL instruction results in an overflow (result is greater than 255).
- A DIV instruction causes a divide-by-zero condition.

The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

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

Bit0: PARITY: Parity Flag.

SFR Definition 6.5. ACC: Accumulator

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE0

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

SFR Definition 6.6. B: B Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xF0

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

6.3. Interrupt Handler

The CIP-51 includes an extended interrupt system supporting a total of 8 interrupt sources with two priority levels. The allocation of interrupt sources between on-chip peripherals and external inputs pins varies according to the specific version of the device. Each interrupt source, with the exception of USB0, has one or more associated interrupt-pending flag(s) located in an SFR. USB0 interrupt sources are located in the USB registers. See Section “12.8. Interrupts” on page 101 for more details about the USB interrupt. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

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

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (IE-EIE2). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

Note: Any instruction which clears the EA bit should be immediately followed by an instruction which has two or more opcode bytes. For example:

```
// in 'C':

EA = 0;    // clear EA bit

EA = 0;    // ... followed by another 2-byte opcode


; in assembly:

CLR  EA    ; clear EA bit

CLR  EA    ; ... followed by another 2-byte opcode
```

If an interrupt is posted during the execution phase of a "CLR EA" opcode (or any instruction that clears the EA bit), and the instruction is followed by a single-cycle instruction, the interrupt may be taken. If the EA bit is read inside the interrupt service routine, it will return a '0'. When the "CLR EA" opcode is followed by a multi-cycle instruction, the interrupt will not be taken.

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

6.3.1. MCU Interrupt Sources and Vectors

The MCU supports 8 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order and control bits are summarized in Table 6.5 on page 50. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

6.3.2. External Interrupts

The /INT0 external interrupt source can be configured as edge or level sensitive. The IT0 bit (TCON.0, see Figure 14.1 on Page 128) selects level or edge sensitivity. When global port I/O inputs are enabled, /INT0 will monitor the voltage at the input pin. The CPU will vector to the /INT0 interrupt service routine whenever the pin detects the condition the external interrupt has been configured to monitor. TMOD.3 (GATE0) controls the functionality of /INT0 as is shown in Table 6.4.

Table 6.4. TMOD.3 Control of /INT0

	TMOD.3 = 0	**TMOD.3 = 1**
/INT0 Pinout	P0.0	P0.2
Edge Sensitivity	Rising Edge	Falling Edge
Level Sensitivity	Active High	Active Low

The /INT1 interrupt source provides an interrupt on two events, based on the logic level of GATE1 (TMOD.7). If GATE1 is set to logic 1, an interrupt is generated every two Low Frequency Internal Oscillator clock cycles. This allows the CPU to vector to the /INT1 interrupt service routine at a rate of 40 kHz. If GATE1 is set to logic 0, an interrupt is generated when the internal oscillator resumes from a suspended state.

The pending flags for the /INT0 and /INT1 interrupts are set upon reset. If the /INT0 or /INT1 interrupt is used, the respective flag should be cleared before enabling the interrupts to prevent an accidental interrupt. The pending flags are for the /INT0 and /INT1 interrupt are in the TCON register.

6.3.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP2) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 6.5.

6.3.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 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.

The CPU is stalled during Flash write/erase operations. Interrupt service latency will be increased for interrupts occurring while the CPU is stalled. The latency for these situations will be determined by the standard interrupt service procedure (as described above) and the amount of time the CPU is stalled.

Table 6.5. 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)
USB0	0x0043	8	Special*	N	N	EUSB0 (EIE1.1)	PUSB0 (EIP1.1)
VBUS Level	0x007B	15	N/A	N/A	N/A	EVBUS (EIE2.0)	PVBUS (EIP2.0)

***Note:** See Section “12.8. Interrupts” on page 101 for more details about the USB interrupt.

6.3.5. Interrupt Register Descriptions

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

SFR Definition 6.7. IE: Interrupt Enable

R/W	R	R	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	—	—	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xA8

Bit7: EA: Enable All Interrupts.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit6–5: Unused. Read = 00b. Write = don't care.

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

Bit3: ET1: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit2: EX1: Enable External Interrupt 1.
This bit sets the masking of External Interrupt 1.
0: Disable external interrupt 1.
1: Enable interrupt requests generated by the /INT1 input.

Bit1: ET0: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit0: EX0: Enable External Interrupt 0.
This bit sets the masking of External Interrupt 0.
0: Disable external interrupt 0.
1: Enable interrupt requests generated by the /INT0 input.

SFR Definition 6.8. IP: Interrupt Priority

R	R	R	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	PS0	PT1	PX1	PT0	PX0	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB8

Bit7–5: Unused. Read = 100b. Write = don't care.
Bit4: PS0: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupts set to high priority level.
Bit3: PT1: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority level.
1: Timer 1 interrupts set to high priority level.
Bit2: PX1: External Interrupt 1 Priority Control.
This bit sets the priority of the External Interrupt 1 interrupt.
0: External Interrupt 1 set to low priority level.
1: External Interrupt 1 set to high priority level.
Bit1: PT0: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.
Bit0: PX0: External Interrupt 0 Priority Control.
This bit sets the priority of the External Interrupt 0 interrupt.
0: External Interrupt 0 set to low priority level.
1: External Interrupt 0 set to high priority level.

SFR Definition 6.9. EIE1: Extended Interrupt Enable 1

R	R	R	R	R	R	R/W	R	Reset Value
—	—	—	—	—	—	EUSB0	—	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xE6

Bit7–2: Unused. Read = 000000b. Write = don't care.
Bit1: EUSB0: Enable USB0 Interrupt.
This bit sets the masking of the USB0 interrupt.
0: Disable all USB0 interrupts.
1: Enable interrupt requests generated by USB0.
Bit0: Unused. Read = 0. Write = don't care.

SFR Definition 6.10. EIP1: Extended Interrupt Priority 1

R	R	R	R	R	R	R/W	R	Reset Value
—	—	—	—	—	—	PUSB0	—	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xF6

Bit7–2: Unused. Read = 000000b. Write = don't care.
Bit1: PUSB0: USB0 Interrupt Priority Control.
This bit sets the priority of the USB0 interrupt.
0: USB0 interrupt set to low priority level.
1: USB0 interrupt set to high priority level.
Bit0: Unused. Read = 0. Write = don't care.

SFR Definition 6.11. EIE2: Extended Interrupt Enable 2

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

Bits7–1: Unused. Read = 0000000b. Write = don't care.
Bit0: EVBUS: Enable VBUS Level Interrupt.
This bit sets the masking of the VBUS interrupt.
0: Disable all VBUS interrupts.
1: Enable interrupt requests generated by VBUS level sense.

SFR Definition 6.12. EIP2: Extended Interrupt Priority 2

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

Bits7–1: Unused. Read = 0000000b. Write = don't care.
Bit0: PVBUS: VBUS Level Interrupt Priority Control.
This bit sets the priority of the VBUS interrupt.
0: VBUS interrupt set to low priority level.
1: VBUS interrupt set to high priority level.

6.4. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the peripherals and clocks active. In Stop mode, the CPU is halted, all interrupts, are inactive, and the internal oscillator is stopped (the voltage regulator, low frequency oscillator, and external clock remain in their selected state). Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. Figure 6.13 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

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

The internal oscillator can be placed in Suspend mode (see Section “10. Oscillators” on page 71). In Suspend mode, the internal oscillator is stopped until a non-idle USB event is detected, or the VBUS input signal matches the polarity selected by the VBPOL bit in register REG0CN (Figure 5.1 on Page 34).

6.4.1. Idle Mode

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

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

6.4.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the low frequency oscillator is not affected. Each analog peripheral (including the low frequency oscillator) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

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

SFR Definition 6.13. PCON: Power Control

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

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

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

Bit0: IDLE: Idle Mode Select.
Setting this bit will place the CIP-51 in Idle mode. This bit will always be read as 0.
1: CPU goes into Idle mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, Serial Ports, and USB0 are still active.)

7. Reset Sources

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

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

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

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For VDD Monitor and Power-On Resets, the RST pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. Refer to Section “10. Oscillators” on page 71 for information on selecting and configuring the system clock source. Program execution begins at location 0x0000.

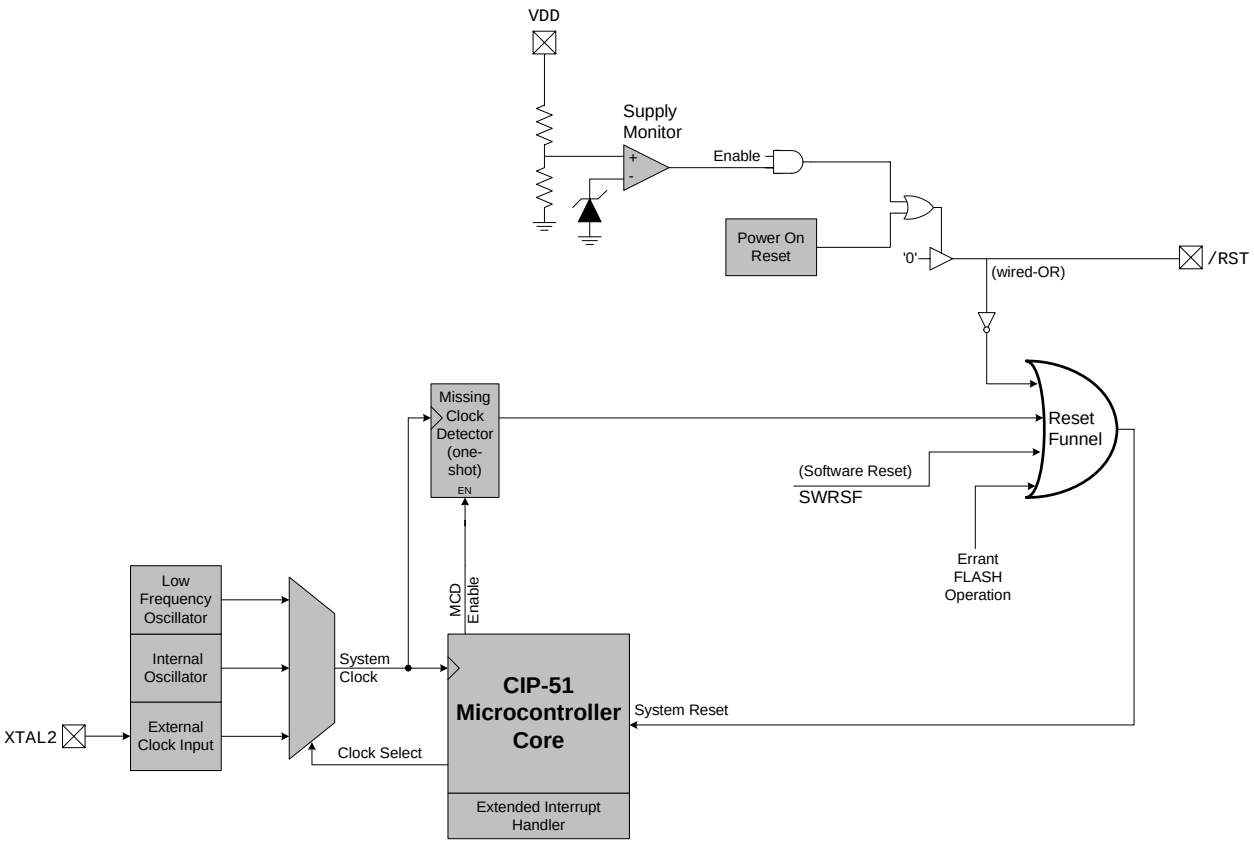


Figure 7.1. Reset Sources

7.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin is driven low until VDD settles above V_{RST}. A Power-On Reset delay ($T_{PORDelay}$) occurs before the device is released from reset; this delay is typically less than 0.3 ms. Figure 7.2. plots the power-on and VDD monitor reset timing.

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

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

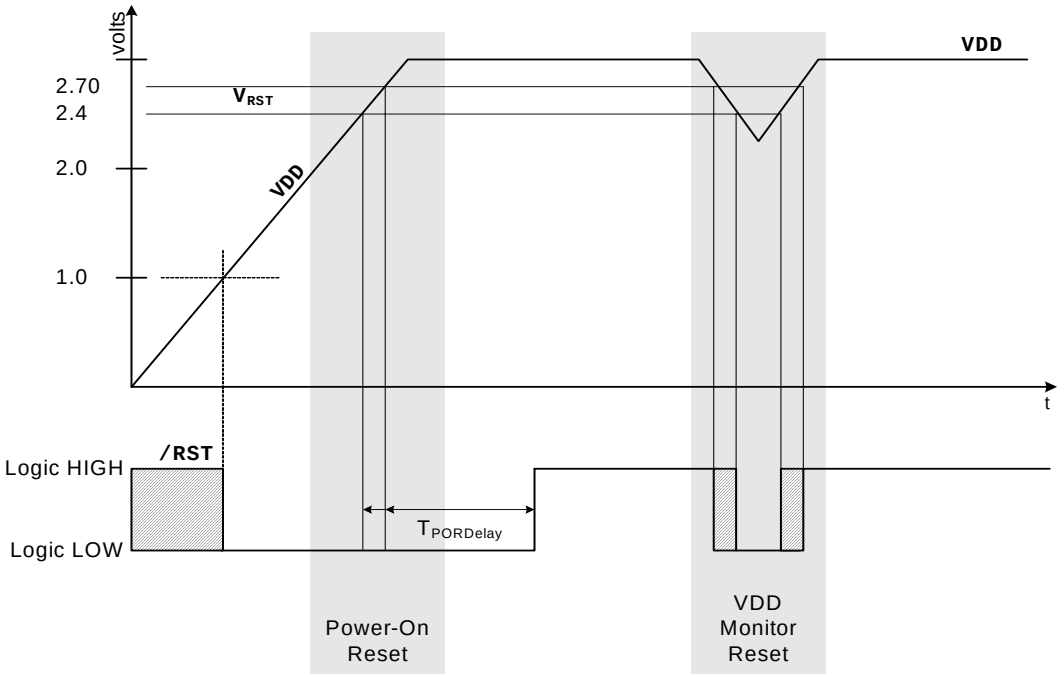


Figure 7.2. Power-On and VDD Monitor Reset Timing

7.2. Power-Fail Reset / VDD Monitor

When a power-down transition or power irregularity causes VDD to drop below V_{RST}, the power supply monitor will drive the $\overline{RST}$ pin low and hold the CIP-51 in a reset state (see Figure 7.2). When VDD returns to a level above V_{RST}, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if VDD dropped below the level required for data retention. If the PORSF flag reads '1', the data may no longer be valid. The VDD monitor is enabled after power-on resets; however its defined state (enabled/disabled) is not altered by any other reset source. For example, if the VDD monitor is enabled and a software reset is performed, the VDD monitor will still be enabled after the reset.

Important Note: The VDD monitor must be enabled before it is selected as a reset source. Selecting the VDD monitor as a reset source before it is enabled and stabilized will cause a system reset. The procedure for configuring the VDD monitor as a reset source is shown below:

Step 1. Enable the VDD monitor (VDM0CN.7 = '1').
Step 2. Wait for the VDD monitor to stabilize (see Table 7.1 for the VDD Monitor turn-on time).
Step 3. Select the VDD monitor as a reset source (RSTSRC.1 = '1').

See Figure 7.2 for VDD monitor timing. See Table 7.1 for complete electrical characteristics of the VDD monitor.

SFR Definition 7.1. VDM0CN: VDD Monitor Control

R/W	R	R	R	R	R	R	R	Reset Value
VDMEN	VDDSTAT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xFF

Bit7: VDMEN: VDD Monitor Enable.
This bit turns the VDD monitor circuit on/off. The VDD Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (Figure 7.2). The VDD Monitor must be allowed to stabilize before it is selected as a reset source. **Selecting the VDD monitor as a reset source before it has stabilized may generate a system reset.** See Table 7.1 for the minimum VDD Monitor turn-on time. The VDD Monitor is enabled following all POR resets.
0: VDD Monitor Disabled.
1: VDD Monitor Enabled.

Bit6: VDDSTAT: VDD Status.
This bit indicates the current power supply status (VDD Monitor output).
0: VDD is at or below the VDD monitor threshold.
1: VDD is above the VDD monitor threshold.

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

7.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 7.1 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

7.4. Missing Clock Detector Reset

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

7.5. Flash Error Reset

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

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to ‘1’ and a MOVX write operation is attempted above address 0x3DFF.
- A Flash read is attempted above user code space. This occurs when a MOVC operation is attempted above address 0x3DFF.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above 0x3DFF.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see Section “8.3. Security Options” on page 65).

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

7.6. Software Reset

Software may force a reset by writing a ‘1’ to the SWRSF bit (RSTSRC.4). The SWRSF bit will read ‘1’ following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

7.7. USB Reset

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

1. RESET signaling is detected on the USB network. The USB Function Controller (USB0) must be enabled for RESET signaling to be detected. See Section “12. Universal Serial Bus Controller (USB0)” on page 87 for information on the USB Function Controller.
2. The voltage on the VBUS pin matches the polarity selected by the VBPOL bit in register REG0CN. See Section “5. Voltage Regulator (REG0)” on page 31 for details on the VBUS detection circuit.

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

SFR Definition 7.2. RSTSRC: Reset Source

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

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

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

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

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

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

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

Bit1: PORSF: Power-On / VDD Monitor Reset Flag.
This bit is set anytime a power-on reset occurs. Writing this bit selects/deselects the VDD monitor as a reset source. **Note: writing '1' to this bit before the VDD monitor is enabled and stabilized can cause a system reset.** See register VDM0CN (Figure 7.1).
0: **Read:** Last reset was not a power-on or VDD monitor reset; **Write:** VDD monitor is not a reset source.
1: **Read:** Last reset was a power-on or VDD monitor reset; all other reset flags indeterminate; **Write:** VDD monitor is a reset source.

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

Note: Do not use read-modify-write instructions on this register.

Table 7.1. Reset Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
‘F326 $\overline{\text{RST}}$ Output Voltage	I_{OL} = –8.5 mA; VIO = 2.7 to 3.6 V I_{OL} = –8.5 mA; VIO = 2.0 V;	—	—	0.6 0.6	V
‘F327 $\overline{\text{RST}}$ Output Voltage	I_{OL} = –8.5 mA; VIO = 2.7 to 3.6 V	—	—	0.6	V
$\overline{\text{RST}}$ Input High Voltage*		0.7 x VIO	—	—	V
$\overline{\text{RST}}$ Input Low Voltage*		—	—	0.3 x VIO	V
‘F326 $\overline{\text{RST}}$ Pullup Current		10	26	40	µA
‘F327 $\overline{\text{RST}}$ Pullup Current		—	26	40	µA
VDD Monitor Threshold (V_{RST})		2.40	2.55	2.70	V
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	100	240	500	µs
Reset Time Delay	Delay between the release of any reset source and code execution at location 0x0000	5.0	—	—	µs
Minimum $\overline{\text{RST}}$ Low Time to Generate a System Reset		15	—	—	µs
VDD Monitor Turn-on Time		100	—	—	µs
VDD Monitor Supply Current		—	20	50	µA
***Note:** On 'F327 devices, VIO = VDD.					

8. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system, a single byte at a time, through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation. Refer to Table 8.1 for complete Flash memory electrical characteristics.

8.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “15. C2 Interface” on page 135.

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

8.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in Figure 8.2.

8.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY); and (2) Setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed must be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE bit (register PSCTL).
Step 8. Clear the PSEE bit (register PSCTI).

8.1.3. Flash Write Procedure

Flash bytes are programmed by software with the following sequence:

Step 1. Disable interrupts (recommended).
Step 2. Erase the 512-byte Flash page containing the target location, as described in Section “8.1.2. Flash Erase Procedure” on page 63.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
Step 8. Clear the PSWE bit (register PSCTL).

Steps 3-8 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.

Table 8.1. Flash Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F326/7	16384*	—	—	bytes
Endurance		20k	100k	—	Erase/Write
Erase Cycle Time	25 MHz System Clock	10	15	20	ms
Write Cycle Time	25 MHz System Clock	40	55	70	µs
***Note:** 512 bytes at location 0x3E00 to 0x3FFF are reserved.					

8.2. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction.

Note: MOVX read instructions always target XRAM.

8.3. Security Options

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

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock *n* 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where *n* is the 1's compliment number represented by the Security Lock Byte. See example below.

Security Lock Byte:	11111101b
1s Compliment:	00000010b
Flash pages locked:	2
Addresses locked:	0x0000 to 0x03FF

Important Notes About the Flash Security:

1. Clearing any bit of the Lock Byte to '0' will lock the Flash page containing the Lock Byte (in addition to the selected pages).
2. Locked pages cannot be read, written, or erased via the C2 interface.
3. Locked pages cannot be read, written, or erased by user firmware executing from unlocked memory space.
4. User firmware executing in a locked page may read and write Flash memory in any locked or unlocked page excluding the reserved area.
5. User firmware executing in a locked page may erase Flash memory in any locked or unlocked page excluding the reserved area and the page containing the Lock Byte.
6. Locked pages can only be unlocked through the C2 interface with a C2 Device Erase command.
7. If a user firmware Flash access attempt is denied (per restrictions #3, #4, and #5 above), a Flash Error system reset will be generated.

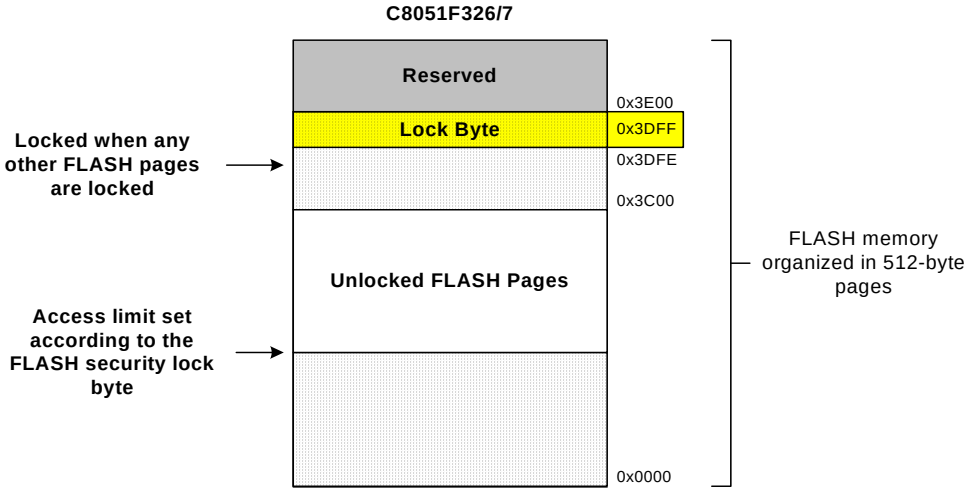


Figure 8.1. Flash Program Memory Map and Security Byte

SFR Definition 8.1. PSCTL: Program Store R/W Control

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

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

SFR Definition 8.2. FLKEY: Flash Lock and Key

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xB7

Bits7–0: FLKEY: Flash Lock and Key Register
Write:
This register must be written to before Flash writes or erases can be performed. Flash remains locked until this register is written to with the following key codes: 0xA5, 0xF1. The timing of the writes does not matter, as long as the codes are written in order. The key codes must be written for each Flash write or erase operation. Flash will be locked until the next system reset if the wrong codes are written or if a Flash operation is attempted before the codes have been written correctly.
Read:
When read, bits 1-0 indicate the current Flash lock state.
00: Flash is write/erase locked.
01: The first key code has been written (0xA5).
10: Flash is unlocked (writes/erases allowed).
11: Flash writes/erases disabled until the next reset.

SFR Definition 8.3. FLSCL: Flash Scale

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

Bits7: FOSE: Flash One-shot Enable
This bit enables the Flash read one-shot. When the Flash one-shot disabled, the Flash sense amps are enabled for a full clock cycle during Flash reads. At system clock frequencies below 10 MHz, disabling the Flash one-shot will increase system power consumption.
0: Flash one-shot disabled.
1: Flash one-shot enabled.
Bits6–0: Reserved. Read = 0. Must Write 0.

9. External RAM

The C8051F326/7 devices include 1280 bytes of on-chip XRAM. This XRAM space is split into user RAM (addresses 0x0000–0x03FF) and USB0 FIFO space. The USB0 FIFO space is only accessible through the USB FIFO registers.

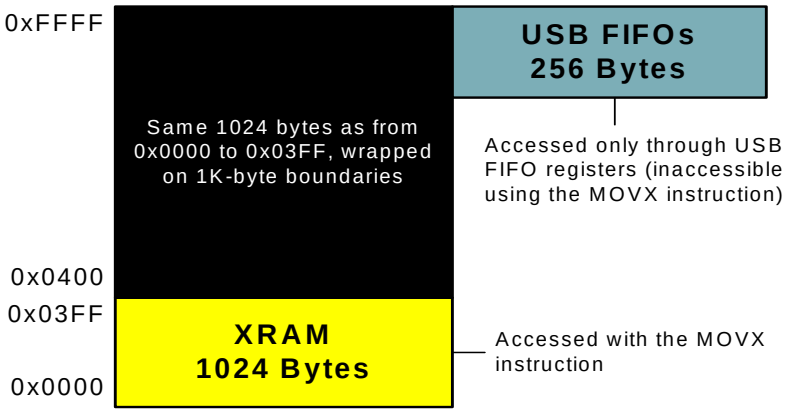


Figure 9.1. External Ram Memory Map

9.1. Accessing User XRAM

User XRAM can be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using MOVX indirect addressing mode. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN as shown in Figure 9.1). Note: the MOVX instruction is also used for writes to the Flash memory. See Section “8. Flash Memory” on page 63 for details. The MOVX instruction accesses XRAM by default.

For any of the addressing modes, the upper 6 bits of the 16-bit external data memory address word are "don't cares". As a result, the 1024-byte RAM is mapped modulo style over the entire 64k external data memory address range. For example, the XRAM byte at address 0x0000 is also at address 0x0400, 0x0800, 0x0C00, 0x1000, etc.

9.2. Accessing USB FIFO Space

The upper 256 bytes of XRAM functions as USB FIFO space. Figure 9.2 shows an expanded view of the FIFO space and user XRAM. FIFO space is accessed via USB FIFO registers; see Section “12.5. FIFO Management” on page 95 for more information on accessing these FIFOs. The FIFO block operates on the USB clock domain; thus the USB clock must be active when accessing FIFO space.

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

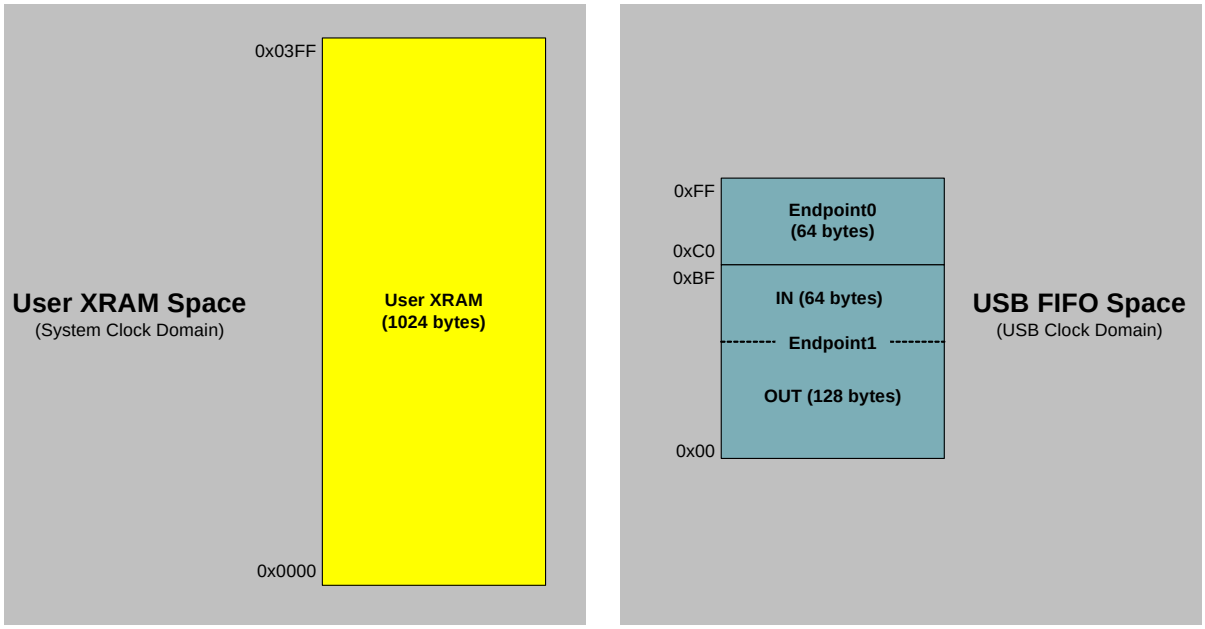


Figure 9.2. XRAM Memory Map Expanded View

SFR Definition 9.1. EMI0CN: External Memory Interface Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	PGSEL1	PGSEL0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xAA

Bits7–3: Unused. Read = 000000b. Write = don’t care.
Bits2–0: PGSEL[1:0]: XRAM Page Select Bits.
The XRAM Page Select Bits provide the high byte of the 16-bit external data memory address when using an 8-bit MOVX command, effectively selecting a 256-byte page of RAM. The upper 6-bits are "don't cares", so the 1k address block is repeated modulo over the entire 64k external data memory address space.

10. Oscillators

C8051F326/7 devices include a programmable internal oscillator, an external clock input circuit, a low frequency internal oscillator, and a 4x Clock Multiplier. The internal oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 10.1. The Low Frequency oscillator can be enabled/disabled and calibrated using the OSCLCN register, as shown in Figure 10.3. The system clock (SYSCLK) can be derived from the internal oscillator, external clock, low frequency oscillator, or the 4x Clock Multiplier divided by 2. The USB clock (USBCLK) can be derived from the internal oscillator divided by 2, external clock, or 4x Clock Multiplier. Oscillator electrical specifications are given in Table 10.3 on page 78.

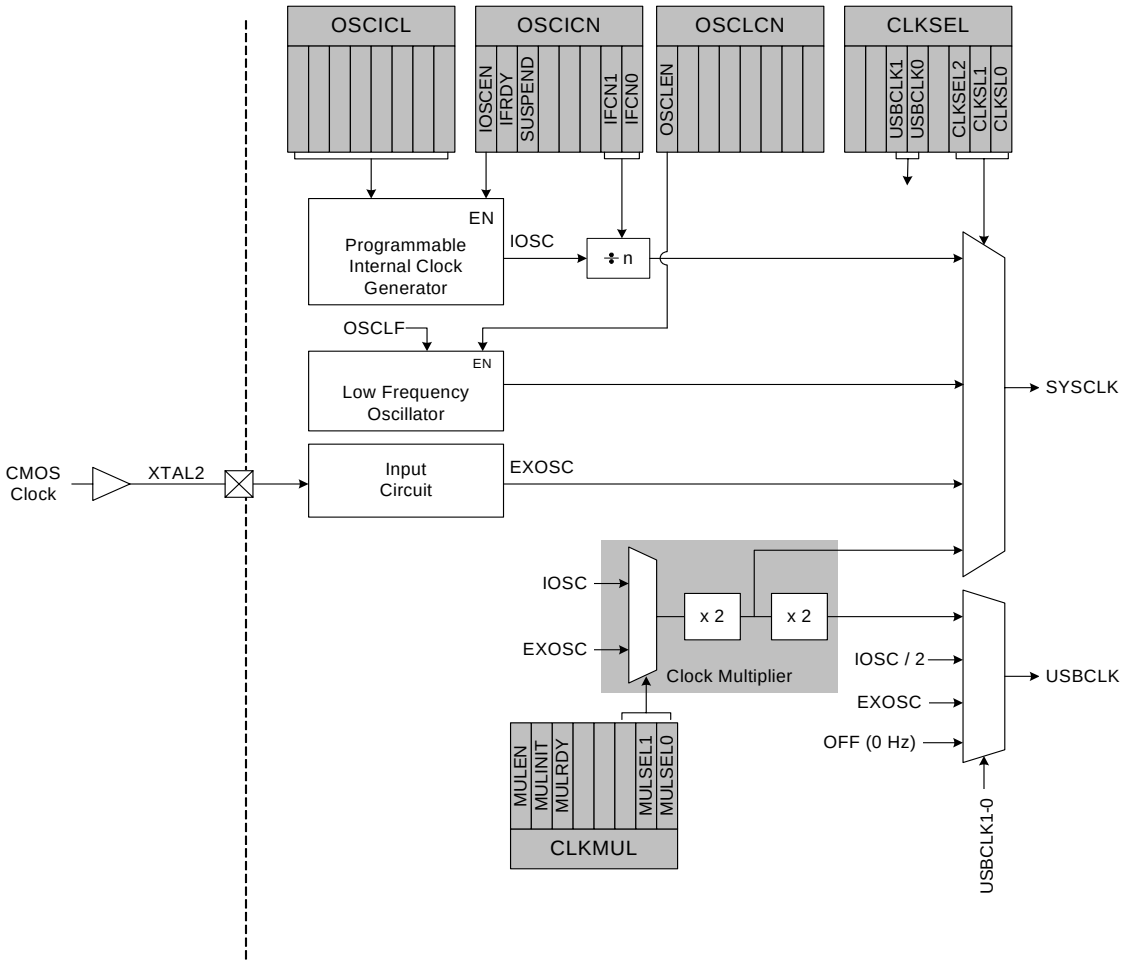


Figure 10.1. Oscillator Diagram

10.1. Programmable Internal Oscillator

All C8051F326/7 devices include a programmable internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be adjusted via the OSCICL register. On C8051F326/7 devices, OSCICL is factory calibrated to obtain a 12 MHz frequency. Electrical specifications for the precision internal oscillator are given in Table 10.3 on page 78. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, or 8, as defined by the IFCN bits in register OSCICN. The divide value defaults to 8 following a reset.

10.1.1. Adjusting the Internal Oscillator on C8051F326/7 Devices

The OSCICL reset value is factory calibrated to result in a 12 MHz internal oscillator with a ±1.5% accuracy; this frequency is suitable for use as the USB clock (see Section “10.5. System and USB Clock Selection” on page 76). Software may adjust the frequency of the internal oscillator using the OSCICL register.

Important Note: Once the internal oscillator frequency has been modified, the internal oscillator may not be used as the USB clock as described in Section “10.5. System and USB Clock Selection” on page 76. The internal oscillator frequency will reset to its original factory-calibrated frequency following any device reset, at which point the oscillator is suitable for use as the USB clock.

10.1.2. Internal Oscillator Suspend Mode

The internal oscillator may be placed in Suspend mode by writing ‘1’ to the SUSPEND bit in register OSCICN. In Suspend mode, the internal oscillator is stopped until a non-idle USB event is detected (Section “12. Universal Serial Bus Controller (USB0)” on page 87) or VBUS matches the polarity selected by the VBPOL bit in register REG0CN (Section “5.2. VBUS Detection” on page 31). Note that the USB transceiver must be enabled or in Suspend mode for a USB event to be detected.

SFR Definition 10.1. OSCICN: Internal Oscillator Control

R/W	R	R/W	R	R/W	R/W	R/W	R/W	Reset Value
IOSCEN	IFRDY	SUSPEND	—	—	—	IFCN1	IFCN0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2

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

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

Bit5: SUSPEND: Force Suspend
Writing a ‘1’ to this bit will force the internal oscillator to be stopped. The oscillator will be restarted on the next non-idle USB event (i.e., RESUME signaling) or VBUS interrupt event (see SFR Definition 5.1).

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

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

SFR Definition 10.2. OSCICL: Internal Oscillator Calibration

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

Bits4–0: OSCCAL: Oscillator Calibration Value
These bits determine the internal oscillator period.

Note: If the sum of the reset value of OSCCAL and ΔOSCCAL is greater than 31 or less than 0, then the device will not be capable of producing the desired frequency.

Note: The contents of this register are undefined when Clock Recovery is enabled. See Section "12.4. USB Clock Configuration" on page 94 for details on Clock Recovery.

10.2. Internal Low-Frequency (L-F) Oscillator

C8051F326/7 devices include a low-frequency oscillator. The OSCLCN register (see SFR Definition 10.3) is used to enabled the oscillator.

SFR Definition 10.3. OSCLCN: Internal L-F Oscillator Control

R/W	R	R	R	R	R	R	R	Reset Value
OSCLEN	—	—	—	—	—	—	—	0xxxxxxx
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE3

Bit7: OSCLEN: Internal L-F Oscillator Enable.
0: Internal L-F Oscillator Disabled.
1: Internal L-F Oscillator Enabled.
Bit6–0: Unused. Read = 0000000b. Write = don't care.

10.3. CMOS External Clock Input

A CMOS clock can be used as an external clock input. The CMOS clock should be wired to the XTAL2 pin (P0.3) as shown in Figure 10.1 on Page 71. Port pins must be configured when using the external oscillator circuit. The Port I/O Crossbar should be configured to allow digital inputs be setting INPUTEN (GPIOCN.6). Also, P0.3 should be configured to open drain mode. See Section "11. Port Input/Output" on page 79 for more information.

10.4. 4x Clock Multiplier

The 4x Clock Multiplier allows a 12 MHz oscillator to generate the 48 MHz clock required for Full Speed USB communication (see Section “12.4. USB Clock Configuration” on page 94). A divided version of the Multiplier output can also be used as the system clock. See Section “10.5. System and USB Clock Selection” on page 76 for details on system clock and USB clock source selection.

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

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

Important Note: When using an external clock as the input to the 4x Clock Multiplier, the external source must be stable before the Multiplier is initialized. See Section “10.5. System and USB Clock Selection” on page 76 for details on clock selection.

SFR Definition 10.4. CLKMUL: Clock Multiplier Control

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

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

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

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

Bits4–1: Unused. Read = 0000b. Write = don’t care.

Bit0: MULSEL: Clock Multiplier Input Select
This bit selects the clock supplied to the Clock Multiplier.

MULSEL	Selected Clock
0	Internal Oscillator
1	External Clock

10.5. System and USB Clock Selection

The internal oscillator requires little start-up time and may be selected as the system or USB clock immediately following the OSCICN write that enables the internal oscillator. If the external clock is selected as the system or USB clock, then startup times may vary based on the specifications of the external clock.

10.5.1. System Clock Selection

The CLKSL[2:0] bits in register CLKSEL select which oscillator source is used as the system clock. CLKSL[2:0] must be set to 001b for the system clock to run from the external clock; however the external clock may still clock certain peripherals (timers, UART, USB) when the internal oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal oscillator, external clock, low frequency oscillator, and 4x Clock Multiplier so long as the selected oscillator is enabled and can provide a stable clock.

10.5.2. USB Clock Selection

The USBCLK[1:0] bits in register CLKSEL select which oscillator source is used as the USB clock. The USB clock may be derived from the 4x Clock Multiplier output, internal oscillator divided by 2, or an external clock. The USB clock source may also be turned off. The USB clock must be 48 MHz when operating USB0 as a Full Speed Function; the USB clock must be 6 MHz when operating USB0 as a Low Speed Function. See Figure 10.5 for USB clock selection options.

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

Table 10.1. Typical USB Full Speed Clock Settings

Internal Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	Clock Multiplier	USBCLK = 00b
Clock Multiplier Input	Internal Oscillator*	MULSEL = 0b
Internal Oscillator	Divide by 1	IFCN = 11b
External Clock		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	Clock Multiplier	USBCLK = 10b
Clock Multiplier Input	External Clock	MULSEL = 1b
Port I/O	12 MHz CMOS Clock	INPUTEN = 1b (GPIOCN.6)
***Note:** Clock Recovery must be enabled for this configuration.		

Table 10.2. Typical USB Low Speed Clock Settings

Internal Oscillator		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	Internal Oscillator / 2	USBCLK = 01b
Internal Oscillator	Divide by 1	IFCN = 11b
External Clock		
Clock Signal	**Input Source Selection**	**Register Bit Settings**
USB Clock	External Clock	USBCLK = 10b
Port I/O	6 MHz CMOS Clock	INPUTEN = 1b (GPIOCN.6)

SFR Definition 10.5. CLKSEL: Clock Select

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

Bits7–6: Unused. Read = 0b. Write = don't care.

Bits5–4: USBCLK1–0: USB Clock Select

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

USBCLK	Selected Clock
00	4x Clock Multiplier
01	Internal Oscillator / 2
10	External Oscillator
11	Clock Off (0 Hz)

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

Bits2–0: CLKSL1–0: System Clock Select

These bits select the system clock source.

CLKSL	Selected Clock
000	Internal Oscillator (as determined by the IFCN bits in register OSCICN)
001	External Clock
010	4x Clock Multiplier / 2
011	Low Frequency Oscillator
1xx	RESERVED

Table 10.3. Internal Oscillator Electrical Characteristics

–40 to +85 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal High-Frequency Oscillator					
Internal Oscillator Frequency	Reset Frequency	11.82	12	12.18	MHz
Internal Oscillator Supply Current (from VDD)	OSCICN.7 = 1	—	574	—	µA
USB Clock Frequency[1]	Full Speed Mode	47.88	48	48.12	MHz
	Low Speed Mode	5.91	6	6.09	
Internal Low-Frequency Oscillator (Using Factory-Calibrated Settings)					
Internal Oscillator Frequency		—	88	—	KHz
Internal Oscillator Supply Current (from VDD)	25 ºC, VDD = 3.0 V, OSCLCN.7 = 1	—	17	—	µA
Power Supply Sensitivity	Constant Temperature	—	–3 ± 0.1[2]	—	%/V
Temperature Sensitivity	Constant Supply	—	20 ± 8	—	ppm/ºC

Notes:
1. Applies only to external oscillator sources.
2. Represents Mean ± 1 Standard Deviation.

11. Port Input/Output

On-Chip digital resources are available through 15 I/O pins. Port pins are organized as shown in Figure 11.1. Each of the Port pins can be used as general-purpose I/O (GPIO). Some port pins can be dedicated to special signals such as /SYSCLK, UART TX and RX, and XTAL2 external clock input.

All Port I/Os are 5 V tolerant (refer to Figure 11.2 for the Port cell circuit). The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,2,3). Complete Electrical Specifications for Port I/O are given in Table 11.1 on page 85.

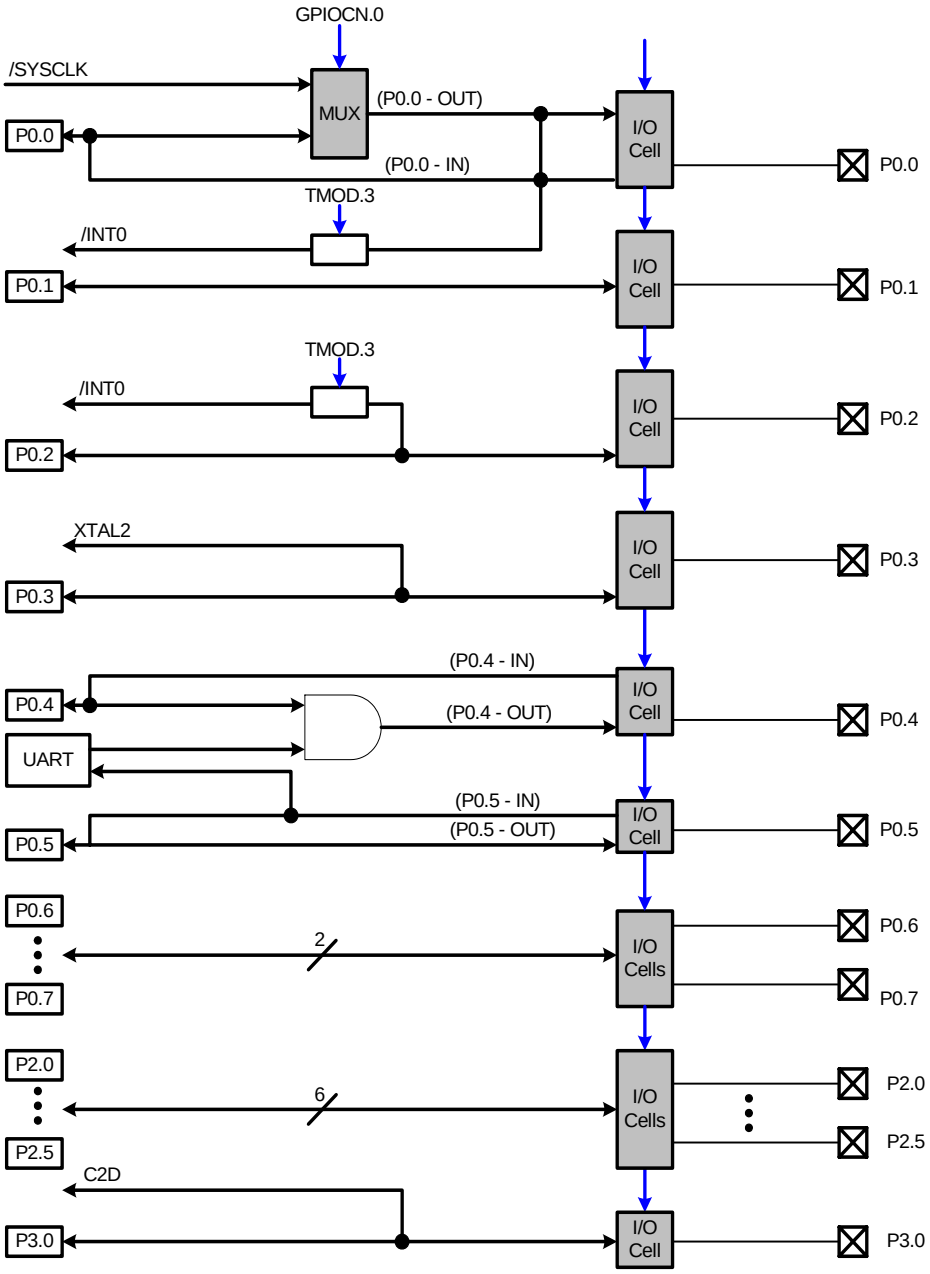


Figure 11.1. Port I/O Functional Block Diagram

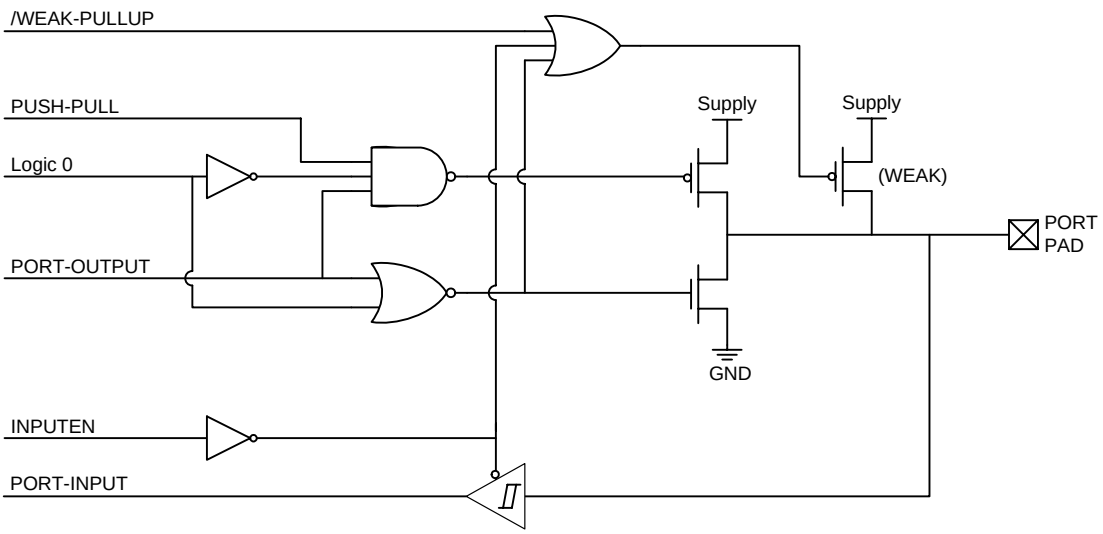


Figure 11.2. Port I/O Cell Block Diagram

11.1. Port I/O Initialization

Port I/O initialization consists of the following steps:

Step 1. Select if the port pin will be used as an output or input.
Step 2. If output, select the output mode: open-drain or push-pull.
Step 3. Configure the PnMDOUT and Pn latches according to the desired input or output configuration.
Step 4. Select if /SYSCLK will appear on the P0.0 output and configure GPIOCN.0.
Step 5. Enable Global Inputs (INPUTEN = ‘1).

Port pins can be used as digital inputs or outputs. To configure a Port pin as a digital input, write ‘0’ to the corresponding bit in register PnMDOUT, and write ‘1’ to the corresponding Port latch (register Pn). When a Port pin is read, the actual voltage at the pin is used to determine a logic 0 or logic 1 value; the Port latch is write-only.

Digital output pins can be configured to open-drain or push-pull. In open drain mode (corresponding bit in PnMDOUT is set to ‘0’), the low output driver is turned on when the Port latch is a logic 0 and turned off when the Port latch is a logic 1. The high output driver is always off, regardless of the Port latch setting. In open drain mode, an output port pin becomes a high impedance input when the Port latch is a logic 1. An external pullup resistor is recommended if the pin is intended for use as an output. This mode is useful when interfacing to 5V logic.

Each port pin has an internal weak pullup that is enabled when the WEAKPUD bit ‘0’, the port output mode is configured as open-drain, and the port latch is a logic 1 (pin is a high impedance input). The weak pullup is disabled if the pin is configured to push-pull mode or the Port latch is a logic 0 to avoid unnecessary power dissipation.

In push-pull mode (corresponding bit in PnMDOUT is set to ‘1’), one of the output drivers will always remain on. When the Port latch is a logic 0, the low output driver is turned on and the high output driver is off. When the Port latch is a logic 1, the low output driver is turned off and the high output driver is turned on. Note that in push-pull mode, the voltage at the port pin will reflect the logic level of the output Port latch. This mode cannot be used to drive logic levels higher than VIO or VDD.

After each port pin is properly configured as an input or output, special signals can be routed to select port pins. Special signals include /SYSCLK on P0.0, XTAL2 clock input on P0.3, UART TX on P0.4, and UART RX on P0.5. The /SYSCLK signal can be routed to P0.0 by setting GPIOCN.0 to ‘1’. The XTAL2 clock input is always routed to P0.3. The UART TX signal is always enabled, and ANDed with the P0.4 latch. When using the UART, the P0.4 Port latches should be logic ‘1’ to allow the UART to control the TX pin. If the Port latch is written ‘0’ at any time, the TX signal will be forced to a logic 0. When the UART is not used, the value of the TX signal is parked at logic 1 and P0.4 can be used as GPIO.

Important Note: Setting the INPUTEN bit in GPIOCN to ‘1’ globally enables digital inputs. Until global inputs are enabled, all port pins on the device remain as output only and cannot be used to sense the logic level on the port pin. INPUTEN must be set to ‘1’ in order to use UART RX, XTAL2, or the /INT0 input.

11.2. General Purpose Port I/O

Port0, Port2, and Port3 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 if INPUTEN is set to ‘1’. The exception to this is the execution of the read-modify-write instructions. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, and DJNZ. The MOV, CLR and SETB instructions are also read-modify-write when the destination is an individual bit in a Port SFR. For these instructions, the value of the register (not the pin) is read, modified, and written back to the SFR.

SFR Definition 11.1. GPIOCN: Global Port I/O Control

R/W	R/W	R	R	R	R	R	R/W	Reset Value
WEAKPUD	INPUTEN	—	—	—	—	—	SYSCLK	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE2

Bit7: WEAKPUD: Port I/O Weak Pullup Disable.
0: Weak Pullups enabled (except for I/O pins with Port latches set to logic 0 or are configured to push-pull mode).
1: Weak Pullups disabled.
Bit6: INPUTEN: Global Digital Input Enable.
0: Port I/O input path disabled; Port pins can be used as outputs only.
1: Port I/O input path enabled.
Bits5–1: Unused. Read = 00000b. Write = don't care.
Bit0: SYSCLK: /SYSCLK Enable
0: /SYSCLK unavailable at P0.0 pin. P0.0 Latch routed to P0.0 pin.
1: /SYSCLK routed to P0.0. P0.0 Latch unavailable at P0.0 pin.

SFR Definition 11.2. P0: Port0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0x80

Bits7–0: P0.[7:0]
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P0MDOUT.n bit = 0).
Read - Always reads '0' if INPUTEN = '0'. Otherwise, directly reads Port pin.
0: P0.n pin is logic low.
1: P0.n pin is logic high.

SFR Definition 11.3. P0MDOUT: Port0 Output Mode

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

Bits7–0: Output Configuration Bits for P0.7-P0.0 (respectively):
0: Corresponding P0.n Output is open-drain.
1: Corresponding P0.n Output is push-pull.

SFR Definition 11.4. P2: Port2

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xA0

Bits7–6: Unused. Read = 00b. Write = don't care.
Bits5–0: P2.[5:0]
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P2MDOUT.n bit = 0).
Read - Always reads '0' if INPUTEN = '0'. Otherwise, directly reads Port pin.
0: P2.n pin is logic low.
1: P2.n pin is logic high.

SFR Definition 11.5. P2MDOUT: Port2 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—							00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xA6

Bits7–6: Unused. Read = 00b. Write = don't care.
Bits5–0: Output Configuration Bit for P2.5–2.0:
0: P2.0 Output is open-drain.
1: P2.0 Output is push-pull.

SFR Definition 11.6. P3: Port3

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	—	P3.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xB0

Bits7–1: Unused. Read = 0000000b. Write = don't care.
Bit0: P3.0
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P3MDOUT.n bit = 0).
Read - Always reads '0' if INPUTEN = '0'. Otherwise, directly reads Port pin.
0: P3.n pin is logic low.
1: P3.n pin is logic high.

SFR Definition 11.7. P3MDOUT: Port3 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	—		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xA7

Bits7–1: Unused. Read = 0000000b. Write = don't care.
Bit0: Output Configuration Bit for P3.0:
0: P3.0 Output is open-drain.
1: P3.0 Output is push-pull.

Table 11.1. Port I/O DC Electrical Characteristics (C8051F326)

VDD = 2.7 to 3.6 V, –40 to +85 °C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
VIO = 2.7 to 3.6 V					
Output High Voltage	IOH = –10 µA; Port I/O push-pull IOH = –3 mA; Port I/O push-pull IOH = –10 mA; Port I/O push-pull	VIO – 0.1 VIO – 0.7 —	— — VIO – 0.8	— — —	V
Output Low Voltage	I_{OL} = 10 µA IOL = 8.5 mA IOL = 25 mA	— — —	— — 1.0	0.1 0.6 —	V
Input High Voltage		2.0	—	—	V
Input Low Voltage		—	—	0.8	V
Input Leakage Current	Weak Pullup Off Weak Pullup On, V_{IN} = 0 V	— —	— 25	± 1 50	µA
VIO = 1.8 V					
Output High Voltage	IOH = –10 µA; Port I/O push-pull IOH = –1 mA; Port I/O push-pull	VIO – 0.1 VIO – 0.4	— —	— —	V
Output Low Voltage	I_{OL} = 10 µA IOL = 3 mA	— —	— —	0.1 0.4	V
Input High Voltage		VDD x 0.7	—	—	V
Input Low Voltage		—	—	VDD x 0.3	V
Input Leakage Current	Weak Pullup Off Weak Pullup On, V_{IN} = 0 V	— —	— 6	±1 15	µA

Table 11.2. Port I/O DC Electrical Characteristics (C8051F327)

VDD = 2.7 to 3.6 V, –40 to +85 °C unless otherwise specified

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

NOTES:

12. Universal Serial Bus Controller (USB0)

C8051F326/7 devices include a complete Full/Low Speed USB function for USB peripheral implementations*. The USB Function Controller (USB0) consists of a Serial Interface Engine (SIE), USB Transceiver (including matching resistors and configurable pullup resistors), 256 Byte FIFO block, and clock recovery mechanism for crystal-less operation. No external components are required. The USB Function Controller and Transceiver is Universal Serial Bus Specification 2.0 compliant.

***Note:** The C8051F326/7 cannot be used as a USB Host device.

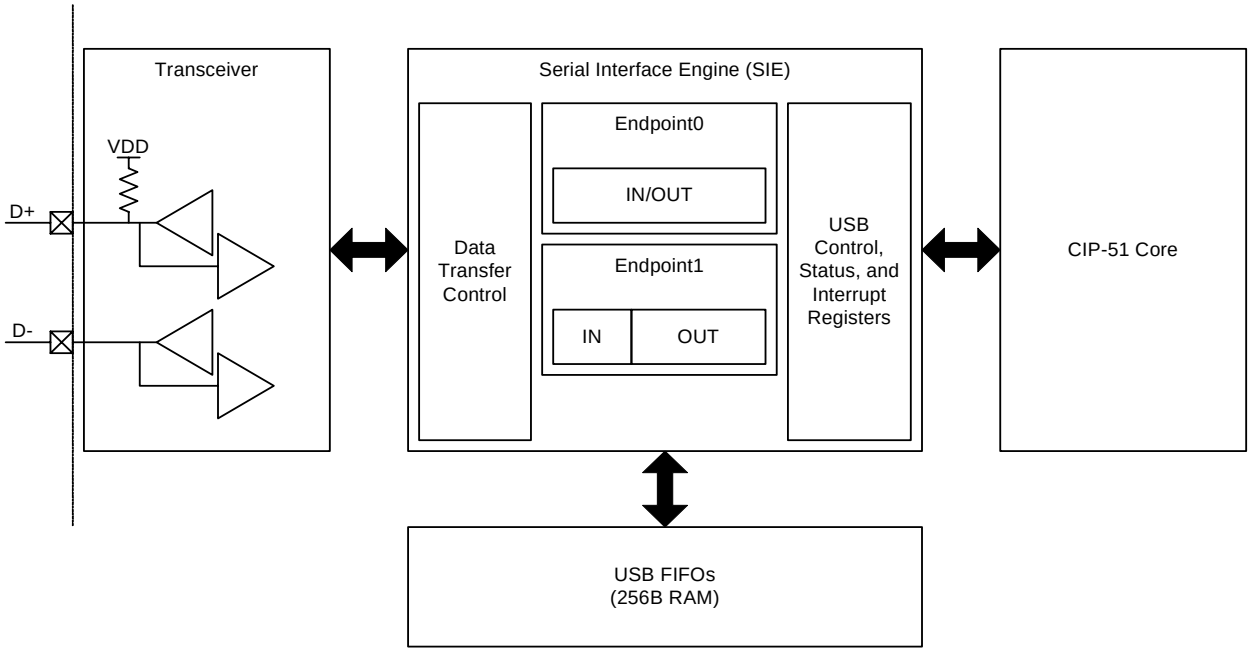


Figure 12.1. USB0 Block Diagram

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

12.1. Endpoint Addressing

A total of three endpoint pipes are available. The control endpoint (Endpoint0) always functions as a bi-directional IN/OUT endpoint. Endpoint 1 is implemented as a 64 byte IN pipe and a 128 byte OUT pipe:

Table 12.1. Endpoint Addressing Scheme

Endpoint	Associated Pipes	USB Protocol Address
Endpoint0	Endpoint0 IN	0x00
	Endpoint0 OUT	0x00
Endpoint1	Endpoint1 IN	0x81
	Endpoint1 OUT	0x01

12.2. USB Transceiver

The USB Transceiver is configured via the USB0XCN register shown in Figure 12.1. This configuration includes Transceiver enable/disable, pullup resistor enable/disable, and device speed selection (Full or Low Speed). When bit SPEED = '1', USB0 operates as a Full Speed USB function, and the on-chip pullup resistor (if enabled) appears on the D+ pin. When bit SPEED = '0', USB0 operates as a Low Speed USB function, and the on-chip pullup resistor (if enabled) appears on the D- pin. Bits4-0 of register USB0XCN can be used for Transceiver testing as described in Figure 12.1. The pullup resistor is enabled only when VBUS is present (see Section "5.2. VBUS Detection" on page 31 for details on VBUS detection).

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

USB Register Definition 12.1. USB0XCN: USB0 Transceiver Control

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

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

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

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

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

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

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

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

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

12.3. USB Register Access

The USB0 controller registers listed in Table 12.2 are accessed through two SFRs: USB0 Address (USB0ADR) and USB0 Data (USB0DAT). The USB0ADR register selects which USB register is targeted by reads/writes of the USB0DAT register. See Figure 12.2.

Endpoint control/status registers are accessed by first writing the USB register INDEX with the target endpoint number. Once the target endpoint number is written to the INDEX register, the control/status registers associated with the target endpoint may be accessed. See the “Indexed Registers” section of Table 12.2 for a list of endpoint control/status registers.

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

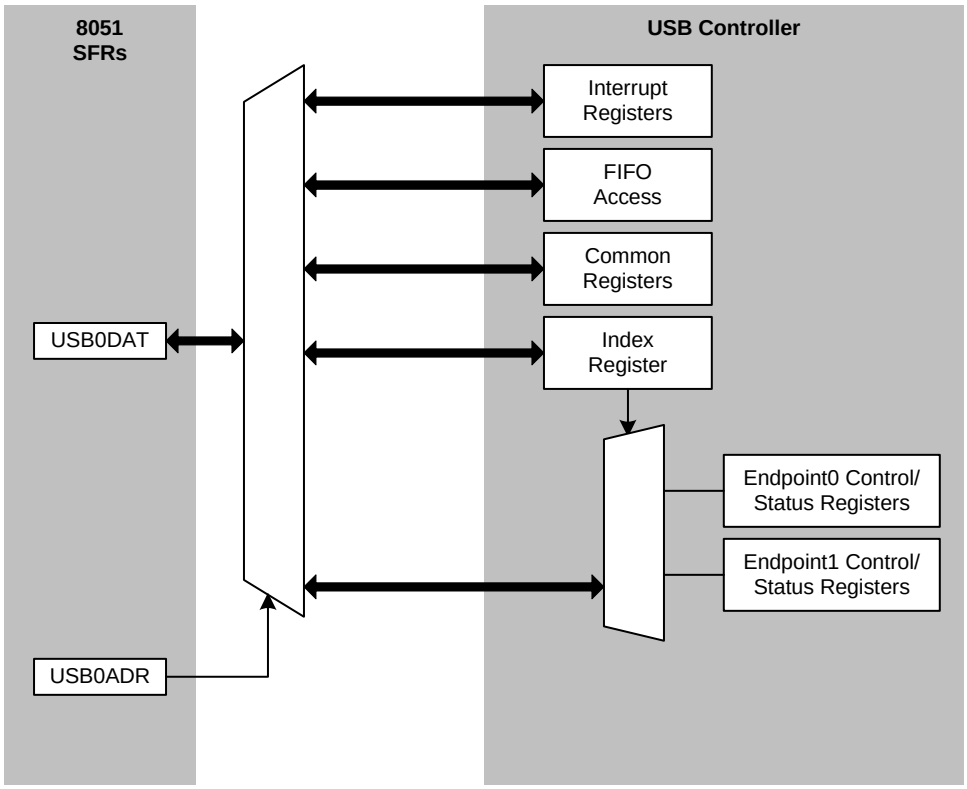


Figure 12.2. USB0 Register Access Scheme

USB Register Definition 12.2. USB0ADR: USB0 Indirect Address

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

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

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

Bits5–0: USBADDR: USB0 Indirect Register Address
These bits hold a 6-bit address used to indirectly access the USB0 core registers. Table 12.2 lists the USB0 core registers and their indirect addresses. Reads and writes to USB0DAT will target the register indicated by the USBADDR bits.

USB Register Definition 12.3. USB0DAT: USB0 Data

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

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

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

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

USB Register Definition 12.4. INDEX: USB0 Endpoint Index

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

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

INDEX	Target Endpoint
0x0	0
0x1	1
0x2–0xF	RESERVED

Table 12.2. USB0 Controller Registers

USB Register Name	USB Register Address	Description	Page Number
Interrupt Registers			
IN1INT	0x02	Endpoint0 and Endpoint1 IN Interrupt Flags	101
OUT1INT	0x04	Endpoint1 OUT Interrupt Flag	101
CMINT	0x06	Common USB Interrupt Flags	102
IN1IE	0x07	Endpoint0 and Endpoint1 IN Interrupt Enables	102
OUT1IE	0x09	Endpoint1 OUT Interrupt Enable	103
CMIE	0x0B	Common USB Interrupt Enable	103
Common Registers			
FADDR	0x00	Function Address	97
POWER	0x01	Power Management	99
FRAMEL	0x0C	Frame Number Low Byte	100
FRAMEH	0x0D	Frame Number High Byte	100
INDEX	0x0E	Endpoint Index Selection	92
CLKREC	0x0F	Clock Recovery Control	94
FIFOn	0x20-0x21	Endpoints0-1 FIFOs	96
Indexed Registers			
E0CSR	0x11	Endpoint0 Control / Status	106
EINCSRL		Endpoint IN Control / Status Low Byte	110
EINCSRH	0x12	Endpoint IN Control / Status High Byte	111
EOUTCSRL	0x14	Endpoint OUT Control / Status Low Byte	113
EOUTCSRH	0x15	Endpoint OUT Control / Status High Byte	114
E0CNT	0x16	Number of Received Bytes in Endpoint0 FIFO	107
EOUTCNTL		Endpoint OUT Packet Count Low Byte	114
EOUTCNTH	0x17	Endpoint OUT Packet Count High Byte	114

12.4. USB Clock Configuration

USB0 is capable of communication as a Full or Low Speed USB function. Communication speed is selected via the SPEED bit in SFR USB0XCN. When operating as a Low Speed function, the USB0 clock must be 6 MHz. When operating as a Full Speed function, the USB0 clock must be 48 MHz. Clock options are described in Section “10. Oscillators” on page 71. The USB0 clock is selected via SFR CLKSEL (see Figure 10.5 on Page 77). The USB transceiver must be enabled before enabling Clock Recovery.

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

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

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

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

USB Register Definition 12.5. CLKREC: Clock Recovery Control

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

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

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

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

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

12.5. FIFO Management

256 bytes of on-chip XRAM are used as FIFO space for USB0. This FIFO space is split between Endpoint0 and Endpoint1 as shown in Figure 12.3. FIFO space allocated for Endpoint1 is split into an IN and an OUT endpoint.

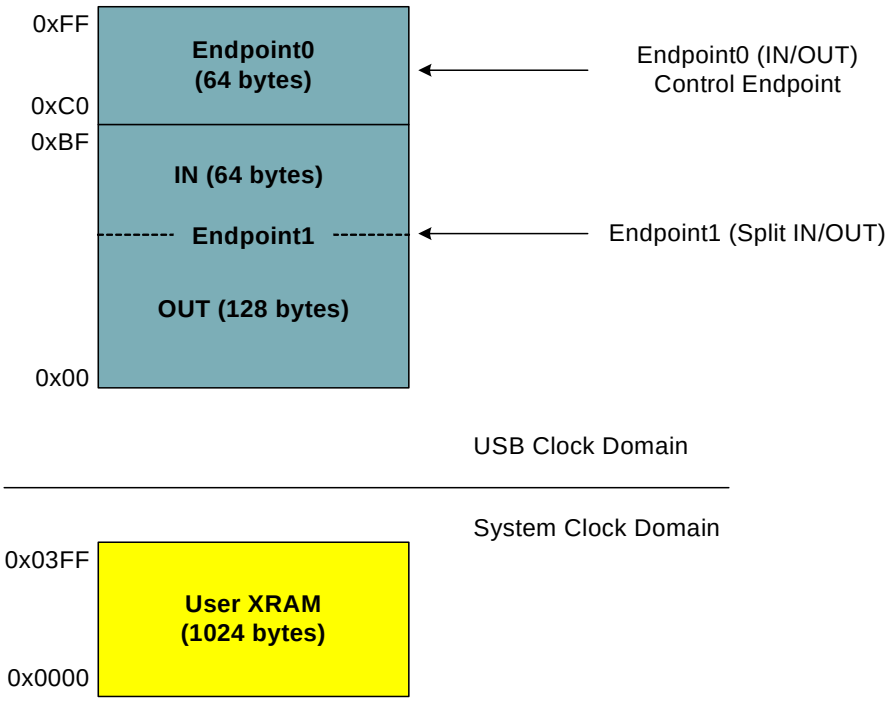


Figure 12.3. USB FIFO Allocation

12.5.1. FIFO Split Mode

The FIFO space for Endpoint1 is split such that the upper 64 bytes of the FIFO space is used by the IN endpoint, and the lower 128 bytes is used by the OUT endpoint.

The FIFO space for Endpoint0 is not split. The 64 byte FIFO space forms a single IN *or* OUT FIFO. Endpoint0 can transfer data in one direction at a time. The endpoint direction (IN/OUT) is determined by the DIRSEL bit in the corresponding endpoint's EINCSRH register (see Figure 12.20).

12.5.2. FIFO Double Buffering

The Endpoint1 FIFO can be configured for double-buffered mode. In this mode, the maximum packet size is halved and the FIFO may contain two packets at a time. This mode is only available for Endpoint1. Double buffering may be enabled for the IN Endpoint and/or the OUT endpoint. See Table 12.3 for a list of maximum packet sizes for each FIFO configuration.

Table 12.3. FIFO Configurations

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

12.5.1. FIFO Access

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

USB Register Definition 12.6. FIFOn: USB0 Endpoint FIFO Access

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

USB Addresses 0x20–0x21 provide access to the 2 pairs of endpoint FIFOs:

IN/OUT Endpoint FIFO	USB Address
0	0x20
1	0x21

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

12.6. Function Addressing

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

USB Register Definition 12.7. FADDR: USB0 Function Address

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

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

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

12.7. Function Configuration and Control

The USB register POWER (Figure 12.8) is used to configure and control USB0 at the device level (enable/disable, Reset/Suspend/Resume handling, etc.).

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

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

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

Suspend Mode: With Suspend Detection enabled (SUSEN = '1'), USB0 will enter Suspend Mode when Suspend signaling is detected on the bus. An interrupt will be generated if enabled (SUSINTE = '1'). The Suspend Interrupt Service Routine (ISR) should perform application-specific configuration tasks such as disabling appropriate peripherals and/or configuring clock sources for low power modes. See Section "10. Oscillators" on page 71 for more details on internal oscillator configuration, including the Suspend mode feature of the internal oscillator.

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

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

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

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

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

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

USB Register Definition 12.8. POWER: USB0 Power

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

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

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

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

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

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

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

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

USB Register Definition 12.9. FRAMEL: USB0 Frame Number Low

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

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

USB Register Definition 12.10. FRAMEH: USB0 Frame Number High

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

Bits7–3: Unused. Read = 00000b. Write = don’t care.
Bits2–0: Frame Number High Byte
This register contains bits10-8 of the last received frame number.

12.8. Interrupts

The read-only USB0 interrupt flags are located in the USB registers shown in Figure 12.11 through Figure 12.13. The associated interrupt enable bits are located in the USB registers shown in Figure 12.14 through Figure 12.16. A USB0 interrupt is generated when any of the USB interrupt flags is set to ‘1’. The USB0 interrupt is enabled via the EIE1 SFR (see Section “6.3. Interrupt Handler” on page 48).

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

USB Register Definition 12.11. IN1INT: USB0 IN Endpoint Interrupt

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

Bits7–2: Unused. Read = 000000b. Write = don’t care.
Bit1: IN1: IN Endpoint 1 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: IN Endpoint 1 interrupt inactive.
1: IN Endpoint 1 interrupt active.
Bit0: EP0: Endpoint 0 Interrupt-pending Flag
This bit is cleared when software reads the IN1INT register.
0: Endpoint 0 interrupt inactive.
1: Endpoint 0 interrupt active.

USB Register Definition 12.12. OUT1INT: USB0 Out Endpoint Interrupt

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

Bits7–2: Unused. Read = 000000b. Write = don’t care.
Bit1: OUT1: OUT Endpoint 1 Interrupt-pending Flag
This bit is cleared when software reads the OUT1INT register.
0: OUT Endpoint 1 interrupt inactive.
1: OUT Endpoint 1 interrupt active.
Bit0: Unused. Read = 0. Write = don’t care.

USB Register Definition 12.13. CMINT: USB0 Common Interrupt

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

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

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

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

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

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

USB Register Definition 12.14. IN1IE: USB0 IN Endpoint Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	IN1E	EP0E	00000011
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x07

Bits7–2: Unused. Read = 000000b. Write = don't care.

Bit1: IN1E: IN Endpoint 1 Interrupt Enable
0: IN Endpoint 1 interrupt disabled.
1: IN Endpoint 1 interrupt enabled.

Bit0: EP0E: Endpoint 0 Interrupt Enable
0: Endpoint 0 interrupt disabled.
1: Endpoint 0 interrupt enabled.

USB Register Definition 12.15. OUT1IE: USB0 Out Endpoint Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	—	—	OUT1E	—	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address:
								0x09

Bits7–2: Unused. Read = 000000b. Write = don't care.
Bit1: OUT1E: OUT Endpoint 1 Interrupt Enable
0: OUT Endpoint 1 interrupt disabled.
1: OUT Endpoint 1 interrupt enabled.
Bit0: Unused. Read = 0. Write = don't' care.

USB Register Definition 12.16. CMIE: USB0 Common Interrupt Enable

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

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

12.9. The Serial Interface Engine

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

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

12.10. Endpoint0

Endpoint0 is managed through the USB register E0CSR (Figure 12.17). The INDEX register must be loaded with 0x00 to access the E0CSR register.

An Endpoint0 interrupt is generated when:

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

The E0CNT register (Figure 12.18) holds the number of received data bytes in the Endpoint0 FIFO.

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

1. The host sends an OUT token during a OUT data phase after the DATAEND bit has been set to ‘1’.
2. The host sends an IN token during an IN data phase after the DATAEND bit has been set to ‘1’.
3. The host sends a packet that exceeds the maximum packet size for Endpoint0.
4. The host sends a non-zero length DATA1 packet during the status phase of an IN transaction.

Firmware sets the SDSTL bit (E0CSR.5) to ‘1’.

12.10.1. Endpoint0 SETUP Transactions

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

12.10.2.Endpoint0 IN Transactions

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

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

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

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

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

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

12.10.3.Endpoint0 OUT Transactions

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

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

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

Endpoint0 will remain in Receive mode until:

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

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

USB Register Definition 12.17. E0CSR: USB0 Endpoint0 Control

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

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

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

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

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

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

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

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

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

USB Register Definition 12.18. E0CNT: USB0 Endpoint 0 Data Count

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

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

Bits6–0: E0CNT: Endpoint 0 Data Count

This 7-bit number indicates the number of received data bytes in the Endpoint 0 FIFO. This number is only valid while bit OPRDY is a '1'.

12.11. Configuring Endpoint1

Endpoint1 is configured and controlled through a set of control/status registers: IN registers EINCSRL and EINCSRH, and OUT registers EOUTCSRL and EOUTCSRH. The endpoint control/status registers are mapped into the USB register address space based on the contents of the INDEX register (Figure 12.4).

12.12. Controlling Endpoint1 IN

Endpoint1 IN is managed via USB registers EINCSRL and EINCSRH. The IN endpoint can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing '1' to the ISO bit in register EINCSRH. Bulk and Interrupt transfers are handled identically by hardware.

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

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

12.12.1.Endpoint1 IN Interrupt or Bulk Mode

When the ISO bit (EINCSRH.6) is logic 0, Endpoint1 operates in Bulk or Interrupt Mode. Once it has been configured to operate in Bulk/Interrupt IN mode (typically following an Endpoint0 SET_INTERFACE command), firmware should load an IN packet into the endpoint IN FIFO and set the INPRDY bit (EINCSRL.0). Upon reception of an IN token, hardware will transmit the data, clear the INPRDY bit, and generate an interrupt.

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

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

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

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

12.12.2.Endpoint1 IN Isochronous Mode

When the ISO bit (EINCSRH.6) is set to '1', the target endpoint operates in Isochronous (ISO) mode. Once an endpoint has been configured for ISO IN mode, the host will send one IN token (data request) per frame; the location of data within each frame may vary. Therefore, it is recommended that double buffering be enabled when using Endpoint1 IN as an Isochronous endpoint.

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

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

The ISO Update feature (see Section “12.7. Function Configuration and Control” on page 98) can be useful in starting a double buffered ISO IN endpoint. If the host has already set up the ISO IN pipe (has begun transmitting IN tokens) when firmware writes the first data packet to the endpoint FIFO, the next IN token may arrive and the first data packet sent before firmware has written the second (double buffered) data packet to the FIFO. The ISO Update feature ensures that any data packet written to the endpoint FIFO will not be transmitted during the current frame; the packet will only be sent after a SOF signal has been received.

USB Register Definition 12.19. EINCSRL: USB0 IN Endpoint Control Low Byte

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

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

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

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

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

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

Bit2: UNDRUN: Data Underrun.
The function of this bit depends on the IN Endpoint mode:
ISO: Set when a zero-length packet is sent after an IN token is received while bit INPRDY = '0'.
Interrupt/Bulk: Set when a NAK is returned in response to an IN token.
This bit must be cleared by software.

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

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

USB Register Definition 12.20. EINCSRH: USB0 IN Endpoint Control High Byte

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

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

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

Bit5–4: Unused. Read = 00b. Write = don’t care.

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

Bits2-0: Unused. Read = 000b. Write = don’t care.

12.13. Controlling Endpoint1 OUT

Endpoint1 OUT is managed via USB registers EOUTCSRL and EOUTCSRH. It can be used for Interrupt, Bulk, or Isochronous transfers. Isochronous (ISO) mode is enabled by writing ‘1’ to the ISO bit in register EOUTCSRH. Bulk and Interrupt transfers are handled identically by hardware.

An Endpoint1 OUT interrupt may be generated by the following:

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

12.13.1.Endpoint1 OUT Interrupt or Bulk Mode

When the ISO bit (EOUTCSRH.6) is logic 0, Endpoint1 operates in Bulk or Interrupt mode. Once it has been configured to operate in Bulk/Interrupt OUT mode (typically following an Endpoint0 SET_INTERFACE command), hardware will set the OPRDY bit (EOUTCSRL.0) to ‘1’ and generate an interrupt upon reception of an OUT token and data packet. The number of bytes in the current OUT data packet (the packet ready to be unloaded from the FIFO) is given in the EOUTCNTH and EOUTCNTL registers. In response to this interrupt, firmware should unload the data packet from the OUT FIFO and reset the OPRDY bit to ‘0’.

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

Hardware will automatically set OPRDY when a packet is ready in the OUT FIFO. Note that if double buffering is enabled for Endpoint1, it is possible for two packets to be ready in the OUT FIFO at a time. In this case, hardware will set OPRDY to ‘1’ immediately after firmware unloads the first packet and resets OPRDY to ‘0’. A second interrupt will be generated in this case.

12.13.2.Endpoint1 OUT Isochronous Mode

When the ISO bit (EOUTCSRH.6) is set to ‘1’, Endpoint1 operates in Isochronous (ISO) mode. Once it has been configured for ISO OUT mode, the host will send exactly one data per USB frame; the location of the data packet within each frame may vary, however. Because of this, it is recommended that double buffering be enabled when Endpoint1 is used in Isochronous mode.

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

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

USB Register Definition 12.21. EOUTCSRL: USB0 OUT Endpoint Control Low Byte

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

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

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

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

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

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

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

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

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

USB Register Definition 12.22. EOUTCSRH: USB0 OUT Endpoint Control High Byte

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

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

USB Register Definition 12.23. EOUTCNTL: USB0 OUT Endpoint Count Low

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

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

USB Register Definition 12.24. EOUTCNTH: USB0 OUT Endpoint Count High

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

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

Table 12.4. USB Transceiver Electrical Characteristics

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

Parameters	Symbol	Conditions	Min	Typ	Max	Units
USB Operating Current		Full Speed Low Speed	— —	5.7 1.5	— —	mA
Transmitter						
Output High Voltage	V_{OH}		2.8	—		V
Output Low Voltage	V_{OL}		—	—	0.8	V
Output Crossover Point	V_{CRS}		1.3	—	2.0	V
Output Impedance	Z_{DRV}	Driving High Driving Low	— —	38 38	— —	W
Pullup Resistance	R_{PU}	Full Speed (D+ Pullup) Low Speed (D– Pullup)	1.425 —	1.5 —	1.575 —	kW
Output Rise Time	T_R	Low Speed Full Speed	75 4	— —	300 20	ns
Output Fall Time	T_F	Low Speed Full Speed	75 4	— —	300 20	ns
Receiver						
Differential Input Sensitivity	V_{DI}	| (D+) – (D–) |	0.2	—	—	V
Differential Input Common Mode Range	V_{CM}		0.8	—	2.5	V
Input Leakage Current	I_L	Pullups Disabled	—	<1.0	—	µA

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

13. UART0

UART0 is an asynchronous, full duplex serial port offering a variety of data formatting options. A dedicated baud rate generator with a 16-bit timer and selectable prescaler is included, which can generate a wide range of baud rates (details in Section “13.1. Baud Rate Generator” on page 118). A received data FIFO allows UART0 to receive up to three data bytes before data is lost and an overflow occurs.

UART0 has six associated SFRs. Three are used for the Baud Rate Generator (SBCON0, SBRLH0, and SBRLL0), two are used for data formatting, control, and status functions (SCON0, SMOD0), and one is used to send and receive data (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete). If additional bytes are available in the Receive FIFO, the RI0 bit cannot be cleared by software.

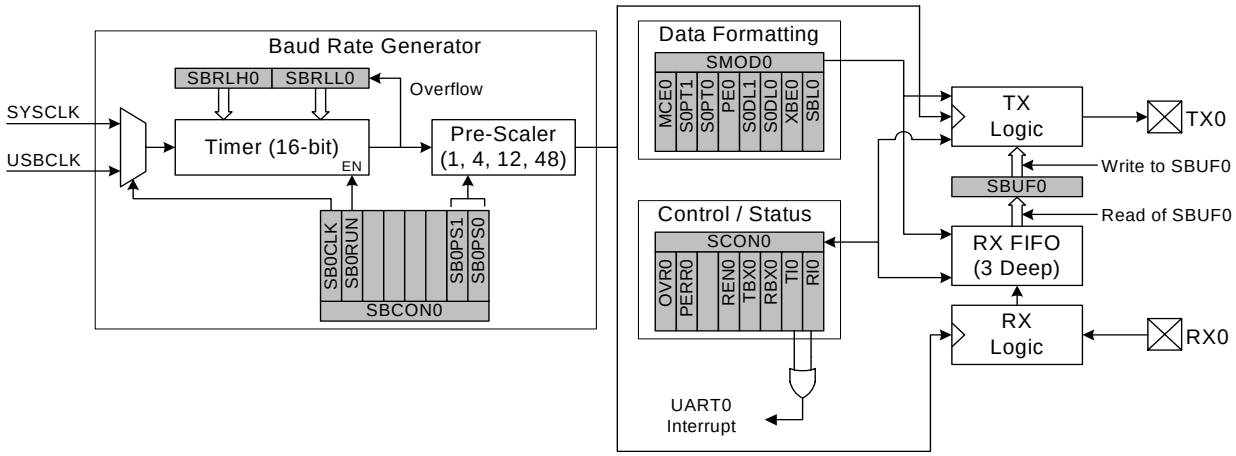


Figure 13.1. UART0 Block Diagram

13.1. Baud Rate Generator

The UART0 baud rate is generated by a dedicated 16-bit timer which runs from either the controller's core clock (SYSCLK) or the USB Clock (USBCLK), and has prescaler options of 1, 4, 12, or 48. The timer and prescaler options combined allow for a wide selection of baud rates over many clock frequencies.

The baud rate generator is configured using three registers: SBCON0, SBRLH0, and SBRLL0. The UART0 Baud Rate Generator Control Register (SBCON0, SFR Definition 13.4) enables or disables the baud rate generator, selects the clock source for the baud rate generator, and selects the prescaler value for the timer. The baud rate generator must be enabled for UART0 to function. Registers SBRLH0 and SBRLL0 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. The baud rate for UART0 is defined in Equation 13.1, where "BRG Clock" is the baud rate generator's selected clock source. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16.

$$\text{Baud Rate} = \frac{\text{BRG Clock}}{(65536 - (\text{SBRLH0:SBRLL0}))} \times \frac{1}{2} \times \frac{1}{\text{Prescaler}}$$

Equation 13.1. UART0 Baud Rate

A quick reference for typical baud rates and clock frequencies is given in Table 13.1.

Table 13.1. Baud Rate Generator Settings for Standard Baud Rates

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

13.2. Data Format

UART0 has a number of available options for data formatting. Data transfers begin with a start bit (logic low), followed by the data bits (sent LSB-first), a parity or extra bit (if selected), and end with one or two stop bits (logic high). The data length is variable between 5 and 8 bits. A parity bit can be appended to the data, and automatically generated and detected by hardware for even, odd, mark, or space parity. The stop bit length is selectable between 1 and 2 bit times, and a multi-processor communication mode is available for implementing networked UART buses. All of the data formatting options can be configured using the SMOD0 register, shown in SFR Definition 13.2. Figure 13.2 shows the timing for a UART0 transaction without parity or an extra bit enabled. Figure 13.3 shows the timing for a UART0 transaction with parity enabled (PE0 = 1). Figure 13.4 is an example of a UART0 transaction when the extra bit is enabled (XBE0 = 1). Note that the extra bit feature is not available when parity is enabled, and the second stop bit is only an option for data lengths of 6, 7, or 8 bits.

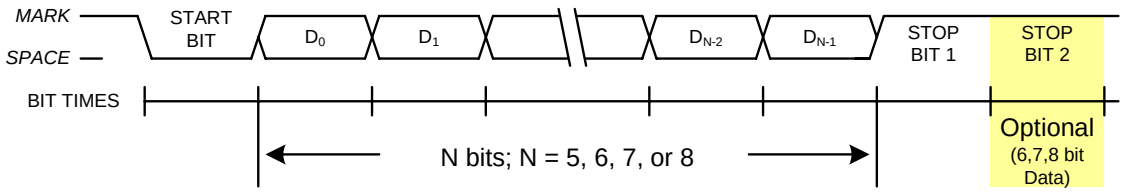


Figure 13.2. UART0 Timing Without Parity or Extra Bit

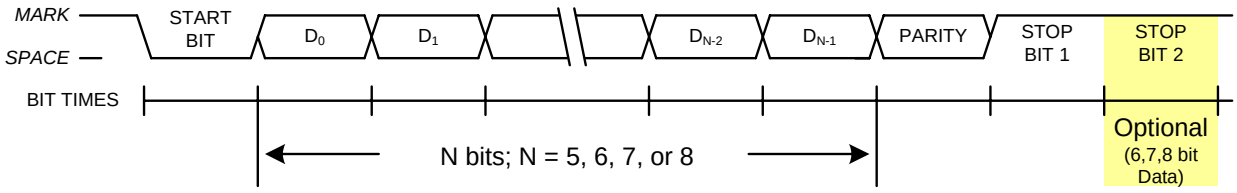


Figure 13.3. UART0 Timing With Parity

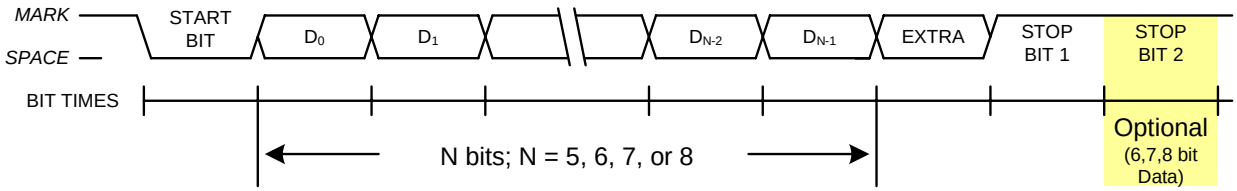


Figure 13.4. UART0 Timing With Extra Bit

13.3. Configuration and Operation

UART0 provides standard asynchronous, full duplex communication. It can operate in a point-to-point serial communications application, or as a node on a multi-processor serial interface. To operate in a point-to-point application, where there are only two devices on the serial bus, the MCE0 bit in SMOD0 should be cleared to '0'. For operation as part of a multi-processor communications bus, the MCE0 and XBE0 bits should both be set to '1'. In both types of applications, data is transmitted from the microcontroller on the TX0 pin, and received on the RX0 pin. The TX0 and RX0 pins are configured using the crossbar and the Port I/O registers, as detailed in Section "11. Port Input/Output" on page 79.

In typical UART communications, The transmit (TX) output of one device is connected to the receive (RX) input of the other device, either directly or through a bus transceiver, as shown in Figure 13.5.

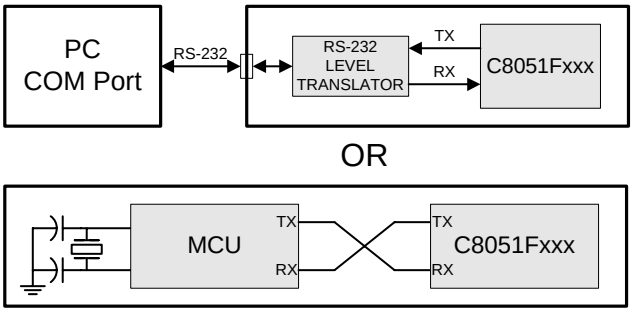


Figure 13.5. Typical UART Interconnect Diagram

13.3.1. Data Transmission

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) will be set at the end of any transmission (the beginning of the stop-bit time). If enabled, an interrupt will occur when TI0 is set.

If the extra bit function is enabled (XBE0 = '1') and the parity function is disabled (PE0 = '0'), the value of the TBX0 (SCON0.3) bit will be sent in the extra bit position. When the parity function is enabled (PE0 = '1'), hardware will generate the parity bit according to the selected parity type (selected with S0PT[1:0]), and append it to the data field. Note: when parity is enabled, the extra bit function is not available.

13.3.2. Data Reception

Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be stored in the receive FIFO if the following conditions are met: the receive FIFO (3 bytes deep) must not be full, and the stop bit(s) must be logic 1. In the event that the receive FIFO is full, the incoming byte will be lost, and a Receive FIFO Overrun Error will be generated (OVR0 in register SCON0 will be set to logic 1). If the stop bit(s) were logic 0, the incoming data will not be stored in the receive FIFO. If the reception conditions are met, the data is stored in the receive FIFO, and the RI0 flag will be set. Note: when MCE0 = '1', RI0 will only be set if the extra bit was equal to '1'. Data can be read from the receive FIFO by reading the SBUF0 register. The SBUF0 register represents the oldest byte in the FIFO. After SBUF0 is read, the next byte in the FIFO is loaded into SBUF0, and space is made available in the FIFO for another incoming byte. If enabled, an interrupt will occur when RI0 is set.

If the extra bit function is enabled (XBE0 = '1') and the parity function is disabled (PE0 = '0'), the extra bit for the oldest byte in the FIFO can be read from the RBX0 bit (SCON0.2). If the extra bit function is not

enabled, the value of the stop bit for the oldest FIFO byte will be presented in RBX0. When the parity function is enabled (PE0 = ‘1’), hardware will check the received parity bit against the selected parity type (selected with S0PT[1:0]) when receiving data. If a byte with parity error is received, the PERR0 flag will be set to ‘1’. This flag must be cleared by software. Note: when parity is enabled, the extra bit function is not available.

13.3.3. Multiprocessor Communications

UART0 supports multiprocessor communication between a master processor and one or more slave processors by special use of the extra data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its extra bit is logic 1; in a data byte, the extra bit is always set to logic 0.

Setting the MCE0 bit (SMOD0.7) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the extra bit is logic 1 (RBX0 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

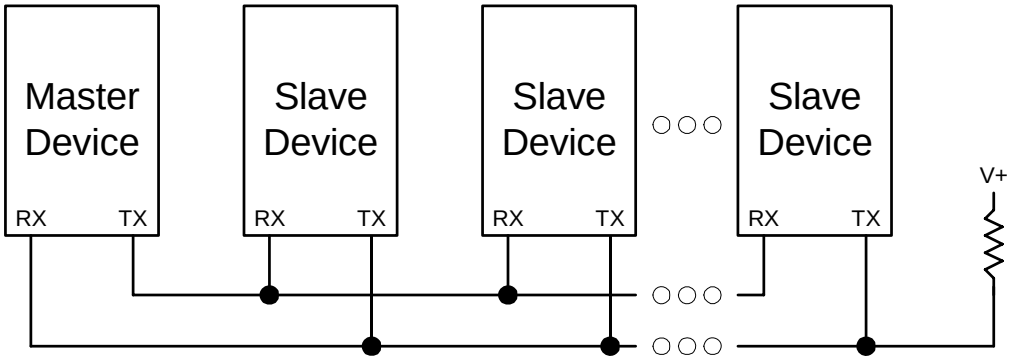


Figure 13.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 13.1. SCON0: UART0 Control

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
OVR0	PERR0	—	REN0	TBX0	RBX0	TI0	RI0	00100000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x98

Bit7: OVR0: Receive FIFO Overrun Flag.
This bit is used to indicate a receive FIFO overrun condition.
0: Receive FIFO Overrun has not occurred.
1: Receive FIFO Overrun has occurred (an incoming character was discarded due to a full FIFO).
This bit must be cleared to ‘0’ by software.

Bit6: PERR0: Parity Error Flag.
When parity is enabled, this bit is used to indicate that a parity error has occurred. It is set to ‘1’ when the parity of the oldest byte in the FIFO does not match the selected Parity Type.
0: Parity Error has not occurred.
1: Parity Error has occurred.
This bit must be cleared to ‘0’ by software.

Bit5: Unused. Read = 1b. Write = don’t care.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART receiver. When disabled, bytes can still be read from the receive FIFO.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: TBX0: Extra Transmission Bit.
The logic level of this bit will be assigned to the extra transmission bit when XBE0 is set to ‘1’. This bit is not used when Parity is enabled.

Bit2: RBX0: Extra Receive Bit.
RBX0 is assigned the value of the extra bit when XBE0 is set to ‘1’. If XBE0 is cleared to ‘0’, RBX0 will be assigned the logic level of the first stop bit. This bit is not valid when Parity is enabled.

Bit1: TI0: Transmit Interrupt Flag.
Set to a ‘1’ by hardware after data has been transmitted, at the beginning of the STOP bit. When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI0: Receive Interrupt Flag.
Set to ‘1’ by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to ‘1’ causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

SFR Definition 13.2. SMOD0: UART0 Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
MCE0	S0PT1	S0PT0	PE0	S0DL1	S0DL0	XBE0	SBL0	00001100
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x9A

Bit7: MCE0: Multiprocessor Communication Enable.
0: RI will be activated if stop bit(s) are ‘1’.
1: RI will be activated if stop bit(s) and extra bit are ‘1’ (extra bit must be enabled using XBE0).
Note: This function is not available when hardware parity is enabled.

Bits6–5: S0PT[1:0]: Parity Type.
00: Odd
01: Even
10: Mark
11: Space

Bit4: PE0: Parity Enable.
This bit activates hardware parity generation and checking. The parity type is selected by bits S0PT1-0 when parity is enabled.
0: Hardware parity is disabled.
1: Hardware parity is enabled.

Bits3–2: S0DL[1:0]: Data Length.
00: 5-bit data
01: 6-bit data
10: 7-bit data
11: 8-bit data

Bit1: XBE0: Extra Bit Enable
When enabled, the value of TBX0 will be appended to the data field.
0: Extra Bit Disabled.
1: Extra Bit Enabled.

Bit0: SBL0: Stop Bit Length
0: Short - Stop bit is active for one bit time (all data field lengths).
1: Long - Stop bit is active for two bit times (data length = 6, 7, or 8 bits).

SFR Definition 13.3. SBUF0: UART0 Data Buffer

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

SFR Address: 0x99

Bits7–0: SBUF0[7:0]: Serial Data Buffer Bits 7–0 (MSB–LSB)
This SFR is used to both send data from the UART and to read received data from the UART0 receive FIFO.
Write: 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.
Read: Reading SBUF0 retrieves data from the receive FIFO. When read, the oldest byte in the receive FIFO is returned, and removed from the FIFO. Up to three bytes may be held in the FIFO. If there are additional bytes available in the FIFO, the RI0 bit will remain at logic '1', even after being cleared by software.

SFR Definition 13.4. SBCON0: UART0 Baud Rate Generator Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SB0CLK	SB0RUN	Reserved	Reserved	Reserved	Reserved	SB0PS1	SB0PS0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x91

Bit7: SB0CLK: Baud Rate Generator Clock Source.
0: SYSCLK is used as Baud Rate Generator Clock Source.
1: USBCLK is used as Baud Rate Generator Clock Source.
Bit6: SB0RUN: Baud Rate Generator Enable.
0: Baud Rate Generator is disabled. UART0 will not function.
1: Baud Rate Generator is enabled.
Bits5–2: Reserved: Read = 0000b. Must write 0000b.
Bits1–0: SB0PS[1:0]: Baud Rate Prescaler Select.
00: Prescaler = 12
01: Prescaler = 4
10: Prescaler = 48
11: Prescaler = 1

SFR Definition 13.5. SBRLH0: UART0 Baud Rate Generator High Byte

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

SFR Address: 0x94

Bits7-0: SBRLH0[7:0]: High Byte of reload value for UART0 Baud Rate Generator.

SFR Definition 13.6. SBRLL0: UART0 Baud Rate Generator Low Byte

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

SFR Address: 0x93

Bits7-0: SBRLL0[7:0]: Low Byte of reload value for UART0 Baud Rate Generator.

14. Timers

Each MCU includes two 16-bit timers compatible with those found in the standard 8051. These timers can be used to measure time intervals and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation.

Table 14.1. Timer Modes

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

Timers 0 and 1 may be clocked by one of four sources, determined by the Timer Mode Select bits (T1M-T0M) and the Clock Scale bits (SCA1-SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See Figure 14.3 for pre-scaled clock selection). Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock.

Timers 0 and 1 have a gate mode which allows the timer to run only when an external interrupt is active (/INT0 for Timer 0 and /INT1 for Timer 1. This mode facilitates pulse width measurements on input on P0.2 (Timer 0) and Low Frequency oscillator calibration when used with Timer 1.

14.1. Timer 0 and Timer 1 Operating Modes

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (Section “6.3.5. Interrupt Register Descriptions” on page 50); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (SFR Definition 6.7). Both timers operate in one of four primary modes selected by setting the Mode Select bits T1M1-T0M0 in the Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

14.1.1. Mode 0: 13-bit Timer

Timer 0 and Timer 1 operate as 13-bit 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 timer. TL0 holds the five LSBs in bit positions TL0.4-TL0.0. The three upper bits of TL0 (TL0.7-TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or GATE0 is logic 1 and the input signal /INT0 is active. Setting GATE0 to logic 1 allows the timer to be controlled by the external input signal /INT0, facilitating pulse width measurements. When GATE0 is set to logic 1, the /INT0 input pin is P0.2.

Table 14.2. Timer 0 Operation

TR0	GATE0	/INT0	Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0 (P0.2 High)	Disabled
1	1	1 (P0.2 Low)	Enabled

X = Don't Care

See Table 6.4 on page 49 for detailed information on how GATE0 affects /INT0 functionality.

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. See Section “6.3.2. External Interrupts” on page 49 for a complete description of /INT0 and /INT1.

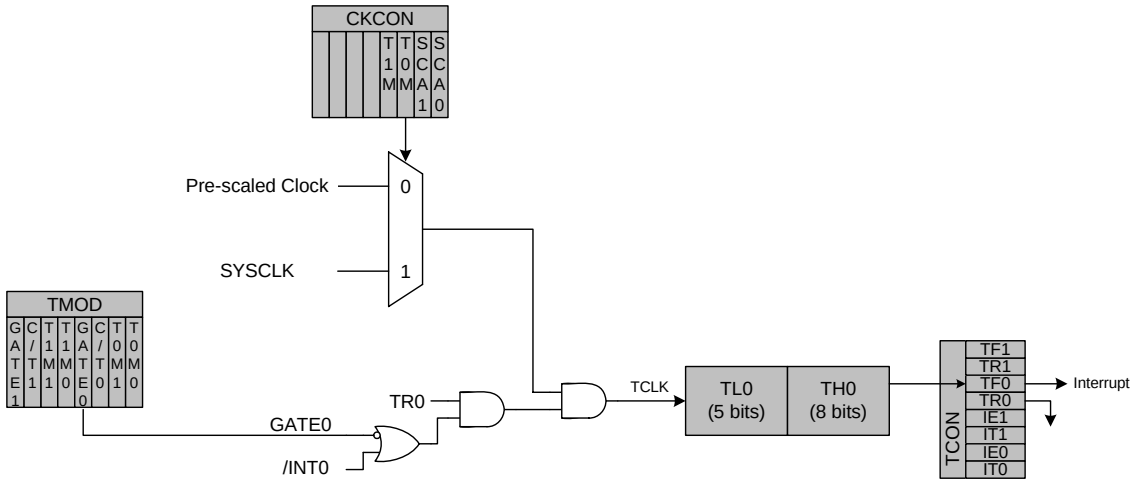


Figure 14.1. T0 Mode 0 Block Diagram

14.1.2. Mode 1: 16-bit Timer

Mode 1 operation is the same as Mode 0, except that the timer registers use all 16 bits. The timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

14.1.3. Mode 2: 8-bit Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when GATE0 (TMOD.3) is logic 0 or when GATE0 is logic 1 and the input signal /INT0 is active (see Section “6.3.2. External Interrupts” on page 49 for details on the external input signals /INT0 and /INT1).

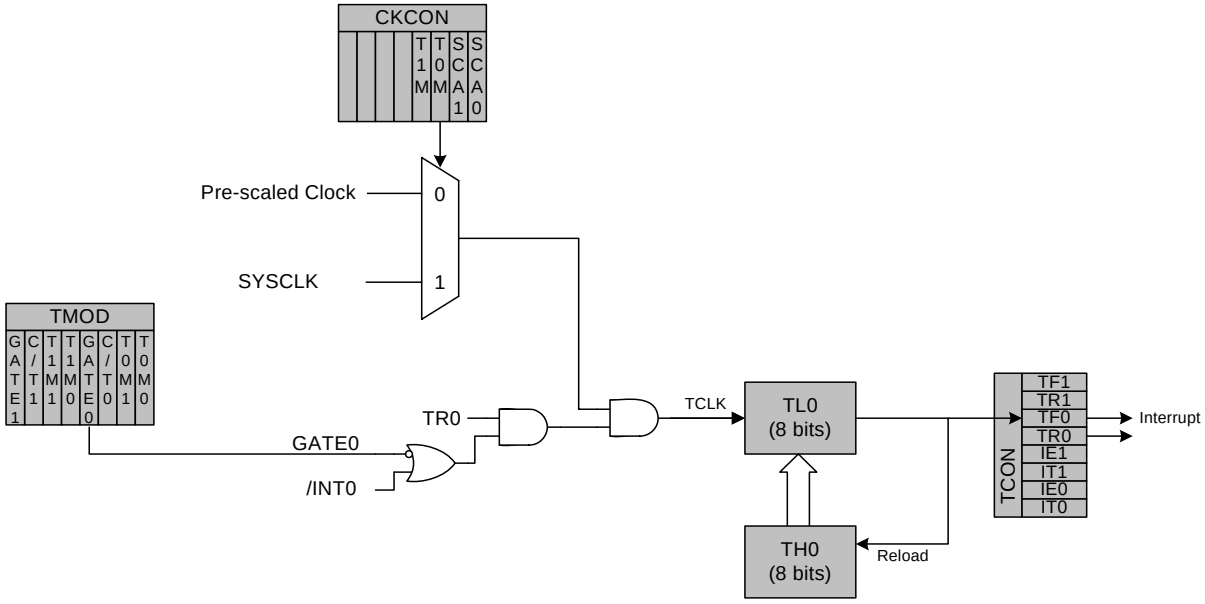


Figure 14.2. T0 Mode 2 Block Diagram

14.1.4. Mode 3: Two 8-bit Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit timers held in TL0 and TH0. The counter in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the UART. 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.

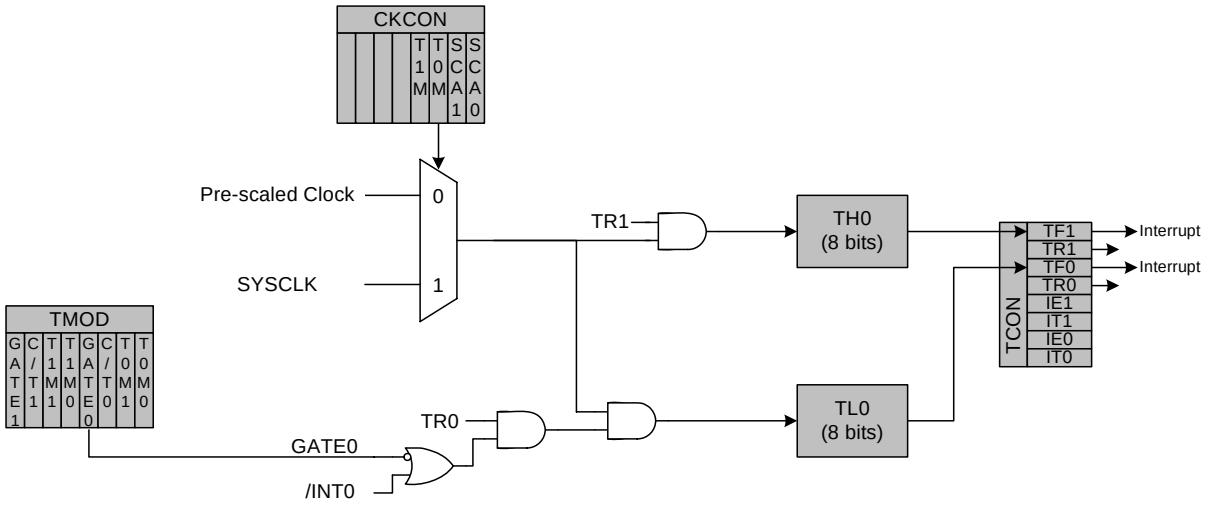


Figure 14.3. T0 Mode 3 Block Diagram

SFR Definition 14.1. TCON: Timer Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00001010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0x88

Bit7: TF1: Timer 1 Overflow Flag.
Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

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

Bit5: TF0: Timer 0 Overflow Flag.
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

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

Bit3: IE1: External Interrupt 1.
This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. When IT1 = 0, this flag is set to ‘1’ when /INT1 is active.

Bit2: IT1: Interrupt 1 Type Select.
This bit selects whether the configured /INT1 interrupt will be edge or level sensitive.
0: /INT1 is level triggered.
1: /INT1 is edge triggered.

Bit1: IE0: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. When IT0 = 0, this flag is set to ‘1’ when /INT0 is active.

Bit0: IT0: Interrupt 0 Type Select.
This bit selects whether the configured /INT0 interrupt will be edge or level sensitive.
0: /INT0 is level triggered.
1: /INT0 is edge triggered.

SFR Definition 14.2. TMOD: Timer Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	Reserved	T1M1	T1M0	GATE0	Reserved	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x89

Bit7: GATE1: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level. /INT1 is activated when the internal oscillator resumes from a suspended state.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 is active. /INT1 is activated every 2 low frequency oscillator clock cycles. This is a rate of 40kHz.

Bit6: Reserved. Read = 0b. Must write 0b.

Bits5–4: T1M1-T1M0: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit timer
0	1	Mode 1: 16-bit timer
1	0	Mode 2: 8-bit timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit3: GATE0: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level. /INT0 input pin is P0.0.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 is active. /INT0 input pin is P0.2.

Bit2: Reserved. Read = 0b. Must write 0b.

Bits1–0: T0M1-T0M0: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit timer
0	1	Mode 1: 16-bit timer
1	0	Mode 2: 8-bit timer with auto-reload
1	1	Mode 3: Two 8-bit timers

SFR Definition 14.3. CKCON: Clock Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	—	—	T1M	T0M	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8E

Bit7–4: Unused. Read = 0b. Write = don't care.

Bit3: T1M: Timer 1 Clock Select.
This select the clock source supplied to Timer 1.
0: Timer 1 uses the clock defined by the prescale bits, SCA1-SCA0.
1: Timer 1 uses the system clock.

Bit2: T0M: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0.
0: Timer 0 uses the clock defined by the prescale bits, SCA1-SCA0.
1: Timer 0 uses the system clock.

Bits1–0: SCA1-SCA0: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and/or Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48

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

SFR Definition 14.4. TL0: Timer 0 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8A

Bits 7–0: TL0: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 14.5. TL1: Timer 1 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8B

Bits 7–0: TL1: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

SFR Definition 14.6. TH0: Timer 0 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8C

Bits 7–0: TH0: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 14.7. TH1: Timer 1 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x8D

Bits 7–0: TH1: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

15. C2 Interface

C8051F326/7 devices include an on-chip Silicon Laboratories 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

15.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 15.1. C2ADD: C2 Address

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

Address	Description
0x00	Selects the Device ID register for Data Read instructions
0x01	Selects the Revision ID register for Data Read instructions
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions
0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions

C2 Register Definition 15.2. DEVICEID: C2 Device ID

Reset Value: 00001001

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit device ID: 0x0D (C8051F326/7).

C2 Register Definition 15.3. REVID: C2 Revision ID

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

This read-only register returns the 8-bit revision ID: 0x01 (Revision B).

C2 Register Definition 15.4. FPCTL: C2 Flash Programming Control

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

Bits7–0 FPCTL: Flash Programming Control Register.
This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 15.5. FPDAT: C2 Flash Programming Data

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

Bits7–0: FPDAT: C2 Flash Programming Data Register.
This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.

Code	Command
0x06	Flash Block Read
0x07	Flash Block Write
0x08	Flash Page Erase
0x03	Device Erase

15.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (/RST) and C2D (P3.0) pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 15.1.

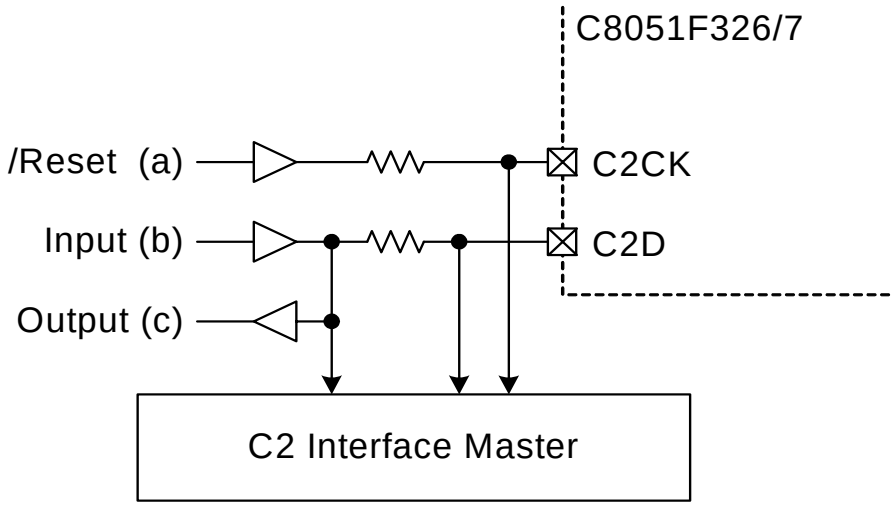


Figure 15.1. Typical C2 Pin Sharing

The configuration in Figure 15.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The /RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

DOCUMENT CHANGE LIST

Revision 0.5 to Revision 1.0

- Updated Section “1. System Overview” on page 13 and Table 1.1, “Product Selection Guide,” on page 13.
  - Changed “-GQ” references to “-GM”
- Added Figure 1.3. "Typical Connections for the C8051F326" on page 16 and Figure 1.4. "Typical Connections for the C8051F327" on page 16.
- Changed Figure 4.5. "Typical C8051F327 QFN-28 Landing Diagram" on page 31 to show ground connection on Pin 3.
- Replaced TBDs with values in Table 5.1, “Voltage Regulator Electrical Specifications,” on page 31.
- Replaced TBDs with values in Table 7.1, “Reset Electrical Characteristics,” on page 62.
- Moved USB Active characteristics from Table 3.1, “Global DC Electrical Characteristics,” on page 24 to Table 12.4, “USB Transceiver Electrical Characteristics,” on page 115.
- Added port information to Figure 11.1. "Port I/O Functional Block Diagram" on page 79.
- Added read/write state description to bits 7–6 in SFR Definition 11.4. “P2: Port2” on page 83.
- Clarified description of read state for bits 7–3 in USB Register Definition 12.10. “FRAMEH: USB0 Frame Number High” on page 100.
- Clarified description of read state for bits 7–2 in USB Register Definition 12.24. “EOUTCNTH: USB0 OUT Endpoint Count High” on page 114.
- Standardized descriptions for “unused” and “reserved” bits in SFR Definitions throughout document.

Revision 1.0 to Revision 1.1

- Updated package and land pattern drawings.

NOTES:

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: MCUinfo@silabs.com
Internet: www.silabs.com

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders