

Welcome to E-XFL.COM

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Not For New Designs
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 17x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f320-gq

20.3.2.Watchdog Timer Usage	238
20.4.Register Descriptions for PCA.....	239
21.C2 Interface	245
21.1.C2 Interface Registers.....	245
21.2.C2 Pin Sharing	247

1. System Overview

C8051F320/1 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 eight flexible endpoint pipes, integrated transceiver, and 1k FIFO RAM
- Supply Voltage Regulator (5-to-3 V)
- True 10-bit 200 ksps 17-channel single-ended/differential ADC with analog multiplexer
- On-chip Voltage Reference and Temperature Sensor
- On-chip Voltage Comparators (2)
- Precision programmable 12 MHz internal oscillator and 4x clock multiplier
- 16 kB of on-chip Flash memory
- 2304 total bytes of on-chip RAM (256 + 1k + 1k USB FIFO)
- SMBus/I²C, Enhanced UART, and Enhanced SPI serial interfaces implemented in hardware
- Four general-purpose 16-bit timers
- Programmable Counter/Timer Array (PCA) with five capture/compare modules and Watchdog Timer function
- On-chip Power-On Reset, VDD Monitor, and Missing Clock Detector
- 25/21 Port I/O (5 V tolerant)

With on-chip Power-On Reset, VDD monitor, Voltage Regulator, Watchdog Timer, and clock oscillator, C8051F320/1 devices are truly stand-alone System-on-a-Chip solutions. The Flash memory can be reprogrammed in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals and may individually shut down any or all peripherals for power savings.

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

Each device is specified for 2.7-to-3.6 V operation over the industrial temperature range (–40 to +85 °C). (Note that 3.0-to-3.6 V is required for USB communication.) The Port I/O and /RST pins are tolerant of input signals up to 5 V. C8051F320/1 are available in a 32-pin LQFP or a 28-pin QFN package.

C8051F320/1

Table 1.1. Product Selection Guide

								</							

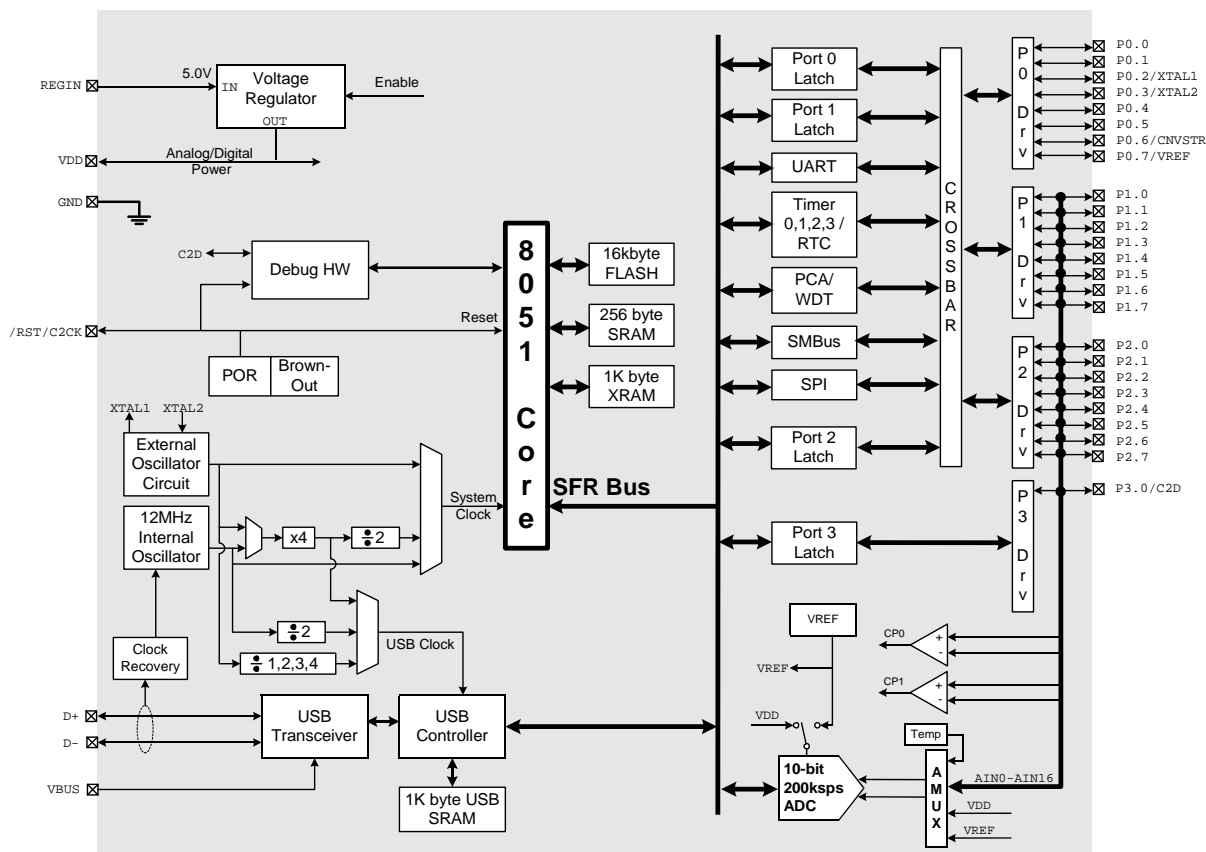


Figure 1.1. C8051F320 Block Diagram

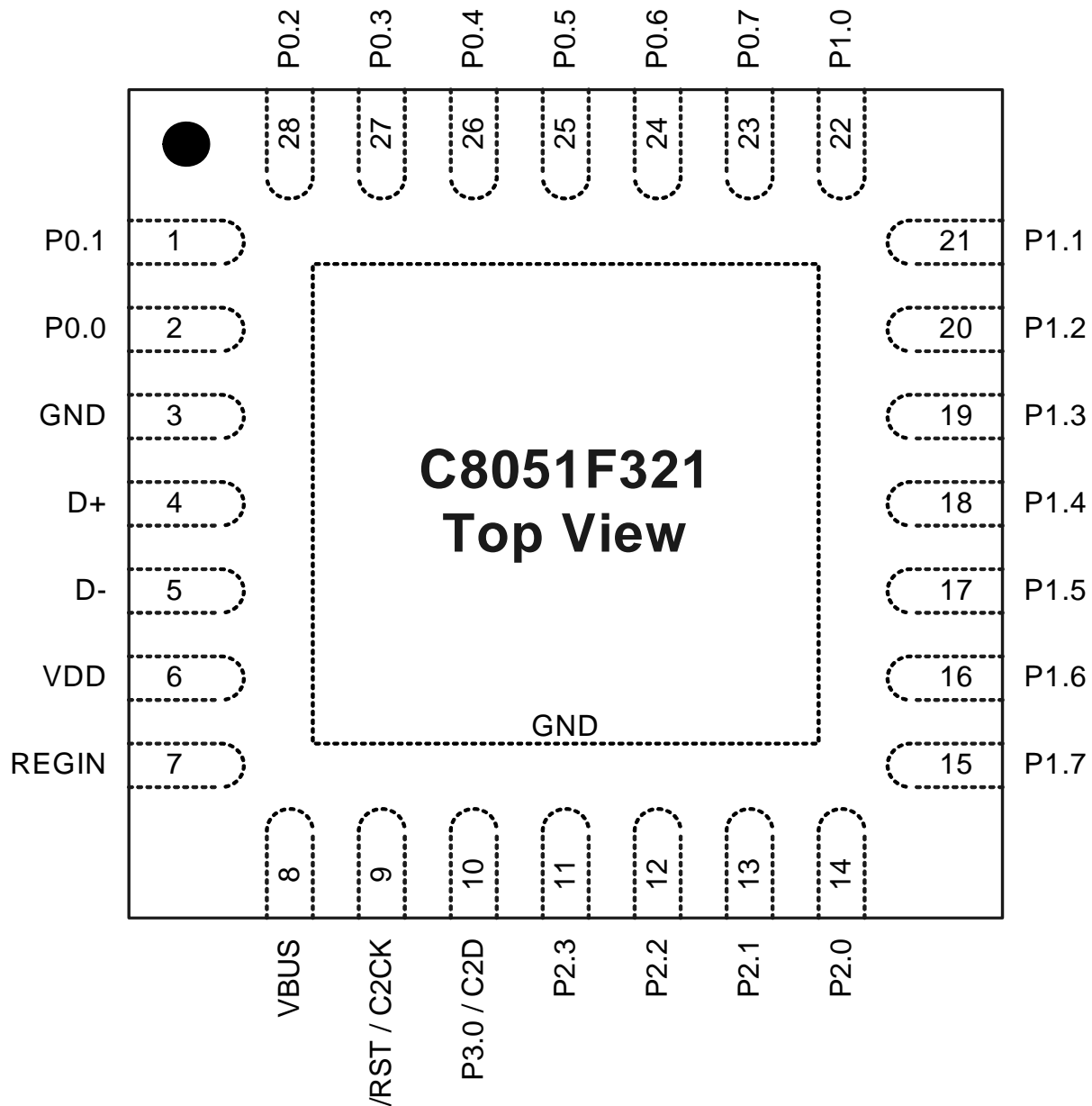


Figure 4.4. QFN-28 Pinout Diagram (Top View)

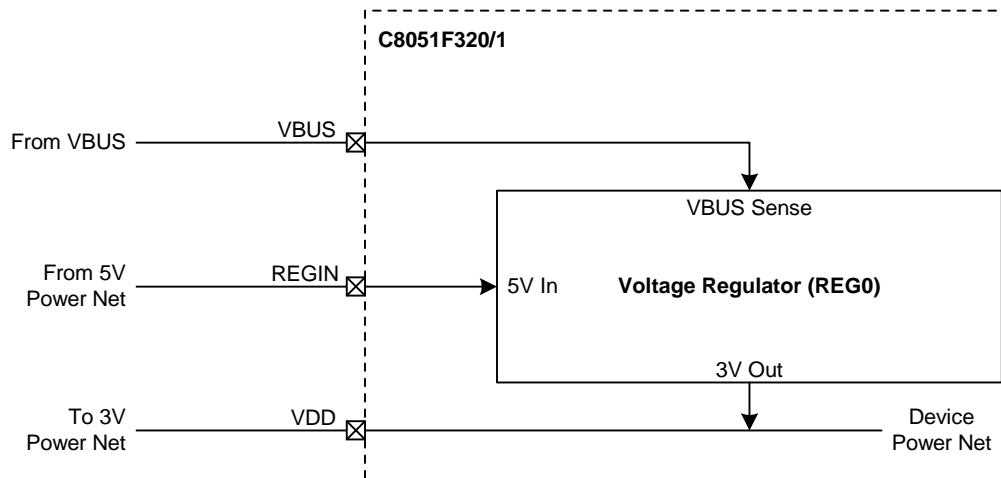


Figure 8.3. REG0 Configuration: USB Self-Powered

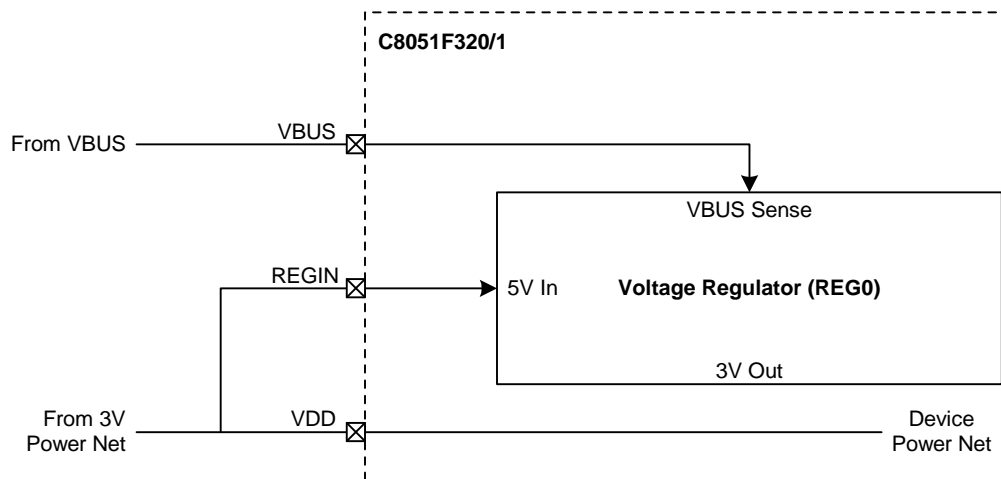


Figure 8.4. REG0 Configuration: USB Self-Powered, Regulator Disabled

SFR Definition 9.13. IT01CF: INT0/INT1 Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xE4

Note: Refer to Figure 19.1 for INT0/1 edge- or level-sensitive interrupt selection.

Bit7: IN1PL: /INT1 Polarity
 0: /INT1 input is active low.
 1: /INT1 input is active high.

Bits6–4: IN1SL2–0: /INT1 Port Pin Selection Bits

These bits select which Port pin is assigned to /INT1. Note that this pin assignment is independent of the Crossbar; /INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

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

Bit3: IN0PL: /INT0 Polarity
 0: /INT0 interrupt is active low.
 1: /INT0 interrupt is active high.

Bits2–0: IN0SL2–0: /INT0 Port Pin Selection Bits

These bits select which Port pin is assigned to /INT0. Note that this pin assignment is independent of the Crossbar; /INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to '1' the corresponding bit in register P0SKIP).

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

11. 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 11.1 for complete Flash memory electrical characteristics.

11.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “21. C2 Interface” on page 245.

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.

11.1.1. 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 11.2.

11.1.2. 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 PSCTL).

11.4. Flash Write and Erase Guidelines

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

To help prevent the accidental modification of Flash by firmware, the VDD Monitor must be enabled and enabled as a reset source on C8051F32x devices for the Flash to be successfully modified. **If either the VDD Monitor or the VDD Monitor reset source is not enabled, a Flash Error Device Reset will be generated when the firmware attempts to modify the Flash.**

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

11.4.1. VDD Maintenance and the VDD Monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum VDD rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external VDD brownout circuit to the /RST pin of the device that holds the device in reset until VDD reaches 2.7 V and re-asserts /RST if VDD drops below 2.7 V.
3. Keep the on-chip VDD Monitor enabled and enable the VDD Monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For 'C'-based systems, this will involve modifying the startup code added by the 'C' compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the VDD Monitor and enabling the VDD Monitor as a reset source. Code examples showing this can be found in AN201, "Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the VDD Monitor and enable the VDD Monitor as a reset source inside the functions that write and erase Flash memory. The VDD Monitor enable instructions should be placed just after the instruction to set PSWE to a '1', but before the Flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct, but "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a '1'. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

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

13.1.2. 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 15) or VBUS matches the polarity selected by the VBPOL bit in register REG0CN (Section 8.2). The transceiver is able to detect non-idle USB events even when it is placed in Suspend mode. On a non-idle USB event, a Resume interrupt is generated, on receipt of which the PHYEN bit should be set to '1' to re-enable the transceiver.

SFR Definition 13.1. 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	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xB2
<p>Bit7: IOSCEN: Internal Oscillator Enable Bit. 0: Internal Oscillator Disabled. 1: Internal Oscillator Enabled.</p> <p>Bit6: IFRDY: Internal Oscillator Frequency Ready Flag. 0: Internal Oscillator is not running at programmed frequency. 1: Internal Oscillator is running at programmed frequency.</p> <p>Bit5: SUSPEND: Force Suspend Writing a '1' to this bit will force the internal oscillator to be stopped. The oscillator will be re-started on the next non-idle USB event (i.e., RESUME signaling) or VBUS interrupt event (see Figure 8.1).</p> <p>Bits4–2: UNUSED. Read = 000b, Write = don't care.</p> <p>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.</p>								

SFR Definition 13.2. 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
<p>Bits7–5: Unused: Read = varies. Write = don't care.</p> <p>Bits4–0: OSCCAL: Oscillator Calibration Value These bits determine the internal oscillator period as per Equation 13.1.</p> <p>Note: The contents of this register are undefined when Clock Recovery is enabled. See Section “15.4. USB Clock Configuration” on page 146 for details on Clock Recovery.</p>								

13.2. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 13.1. A 10 M Ω resistor also must be wired across the XTAL1 and XTAL2 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 13.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see Figure 13.3)

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.2 and P0.3 are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P0.3 is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see Section “14.1. Priority Crossbar Decoder” on page 128 for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See Section “14.2. Port I/O Initialization” on page 130 for details on Port input mode selection.

13.2.1. Clocking Timers Directly Through the External Oscillator

The external oscillator source divided by eight is a clock option for the timers (Section “19. Timers” on page 209) and the Programmable Counter Array (PCA) (Section “20. Programmable Counter Array (PCA0)” on page 227). When the external oscillator is used to clock these peripherals, but is not used as the system clock, the external oscillator frequency must be less than or equal to the system clock frequency. In this configuration, the clock supplied to the peripheral (external oscillator / 8) is synchronized with the system clock; the jitter associated with this synchronization is limited to ± 0.5 system clock cycles.

13.2.2. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 13.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in Figure 13.3 (OSCXCN register). For example, a 12 MHz crystal requires an XFCN setting of 111b.

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

- Step 1. Enable the external oscillator.
- Step 2. Wait at least 1 ms.
- Step 3. Poll for XTLVLD => ‘1’.
- Step 4. Switch the system clock to the external oscillator.

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

SFR Definition 14.9. P1MDOUT: Port1 Output Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xA5

Bits7–0: Output Configuration Bits for P1.7-P1.0 (respectively): ignored if corresponding bit in register P1MDIN is logic 0.
 0: Corresponding P1.n Output is open-drain.
 1: Corresponding P1.n Output is push-pull.

SFR Definition 14.10. P1SKIP: Port1 Skip Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD5

Bits7–0: P1SKIP[7:0]: Port1 Crossbar Skip Enable Bits.
 These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (VREF input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
 0: Corresponding P1.n pin is not skipped by the Crossbar.
 1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 14.11. P2: Port2 Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: (bit addressable) 0xA0

Bits7–0: P2.[7:0]
 Write - Output appears on I/O pins per Crossbar Registers (when XBARE = '1').
 0: Logic Low Output.
 1: Logic High Output (high impedance if corresponding P2MDOUT.n bit = 0).
 Read - Always reads '0' if selected as analog input in register P2MDIN. Directly reads Port pin when configured as digital input.
 0: P2.n pin is logic low.
 1: P2.n pin is logic high.

Note: P2.7–P2.4 only available on C8051F320 devices. Writes to these Ports do not require XBARE = '1'.

15.1. Endpoint Addressing

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

Table 15.1. Endpoint Addressing Scheme

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

15.2. USB Transceiver

The USB Transceiver is configured via the USB0XCN register shown in Figure 15.1. This configuration includes Transceiver enable/disable, pull-up resistor enable/disable, and device speed selection (Full or Low Speed). When bit SPEED = '1', USB0 operates as a Full Speed USB function, and the on-chip pull-up resistor (if enabled) appears on the D+ pin. When bit SPEED = '0', USB0 operates as a Low Speed USB function, and the on-chip pull-up resistor (if enabled) appears on the D- pin. Bits4-0 of register USB0XCN can be used for Transceiver testing as described in Figure 15.1. The pull-up resistor is enabled only when VBUS is present (see Section "8.2. VBUS Detection" on page 67 for details on VBUS detection).

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

USB Register Definition 15.9. 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 15.10. FRAMEH: USB0 Frame Number High

R	R	R	R	R	R	R	R	Reset Value
-	-	-	-	-	Frame Number High			00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	USB Address: 0x0D
Bits7-3: Unused. Read = 0. Write = don't care. Bits2-0: Frame Number High Byte This register contains bits10-8 of the last received frame number.								

15.8. Interrupts

The read-only USB0 interrupt flags are located in the USB registers shown in Figure 15.11 through Figure 15.13. The associated interrupt enable bits are located in the USB registers shown in Figure 15.14 through Figure 15.16. A USB0 interrupt is generated when any of the USB interrupt flags is set to '1'. The USB0 interrupt is enabled via the EIE1 SFR (see Section "9.3. Interrupt Handler" on page 87).

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

USB Register Definition 15.12. OUT1INT: USB0 Out Endpoint Interrupt

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

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

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

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

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

15.12.2.Endpoints1-3 IN Isochronous Mode

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

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

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

The ISO Update feature (see Section 15.7) can be useful in starting a double buffered ISO IN endpoint. If the host has already set up the ISO IN pipe (has begun transmitting IN tokens) when firmware writes the first data packet to the endpoint FIFO, the next IN token may arrive and the first data packet sent before firmware has written the second (double buffered) data packet to the FIFO. The ISO Update feature ensures that any data packet written to the endpoint FIFO will not be transmitted during the current frame; the packet will only be sent after a SOF signal has been received.

USB Register Definition 15.22. EOUTCSRH: USB0 OUT Endpoint Control Low 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 15.23. 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 15.24. EOUTCNTH: USB0 OUT Endpoint Count High

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

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

Bits1–0: EPOCH: OUT Endpoint Count High Byte
 EPOCH 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'.

16.6. SMBus Status Decoding

The current SMBus status can be easily decoded using the SMB0CN register. In the table below, STATUS VECTOR refers to the four upper bits of SMB0CN: MASTER, TXMODE, STA, and STO. Note that the shown response options are only the typical responses; application-specific procedures are allowed as long as they conform to the SMBus specification. Highlighted responses are allowed but do not conform to the SMBus specification.

Table 16.4. SMBus Status Decoding

Mode	Values Read				Current SMBus State	Typical Response Options	Values Written		
	Status Vector	ACKRQ	ARBLOST	ACK			STA	STO	ACK
Master Transmitter	1110	0	0	X	A master START was generated.	Load slave address + R/W into SMB0DAT.	0	0	X
	1100	0	0	0	A master data or address byte was transmitted; NACK received.	Set STA to restart transfer.	1	0	X
						Abort transfer.	0	1	X
		0	0	1	A master data or address byte was transmitted; ACK received.	Load next data byte into SMB0DAT.	0	0	X
						End transfer with STOP.	0	1	X
						End transfer with STOP and start another transfer.	1	1	X
						Send repeated START.	1	0	X
						Switch to Master Receiver Mode (clear SI without writing new data to SMB0DAT).	0	0	X

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 18.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is no way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 18.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

18.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

Note that all of the following bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written. This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master, and for multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed and the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

18.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between an active-high or active-low clock. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships for master mode are shown in Figure 18.5. For slave mode, the clock and data relationships are shown in Figure 18.6 and Figure 18.7. Note that CKPHA must be set to '0' on both the master and slave SPI when communicating between two of the following devices: C8051F04x, C8051F06x, C8051F12x, C8051F31x, C8051F32x, and C8051F33x

The SPI0 Clock Rate Register (SPI0CKR) as shown in Figure 18.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz, whichever is

SFR Definition 19.1. TCON: Timer Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
						(bit addressable)		0x88
Bit7:	<p>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.</p>							
Bit6:	<p>TR1: Timer 1 Run Control. 0: Timer 1 disabled. 1: Timer 1 enabled.</p>							
Bit5:	<p>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.</p>							
Bit4:	<p>TR0: Timer 0 Run Control. 0: Timer 0 disabled. 1: Timer 0 enabled.</p>							
Bit3:	<p>IE1: External Interrupt 1. This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. When IT1 = 0, this flag is set to '1' when /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 9.13).</p>							
Bit2:	<p>IT1: Interrupt 1 Type Select. This bit selects whether the configured /INT1 interrupt will be edge or level sensitive. /INT1 is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 9.13). 0: /INT1 is level triggered. 1: /INT1 is edge triggered.</p>							
Bit1:	<p>IE0: External Interrupt 0. This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. When IT0 = 0, this flag is set to '1' when /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 9.13).</p>							
Bit0:	<p>IT0: Interrupt 0 Type Select. This bit selects whether the configured /INT0 interrupt will be edge or level sensitive. /INT0 is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 9.13). 0: /INT0 is level triggered. 1: /INT0 is edge triggered.</p>							