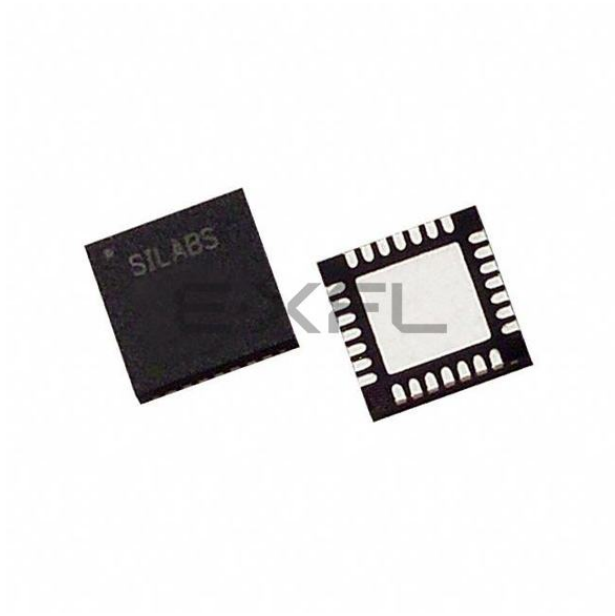

Welcome to E-XFL.COM

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	UART/USART, USB
Peripherals	POR
Number of I/O	15
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f327-gmr

List of Tables

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.

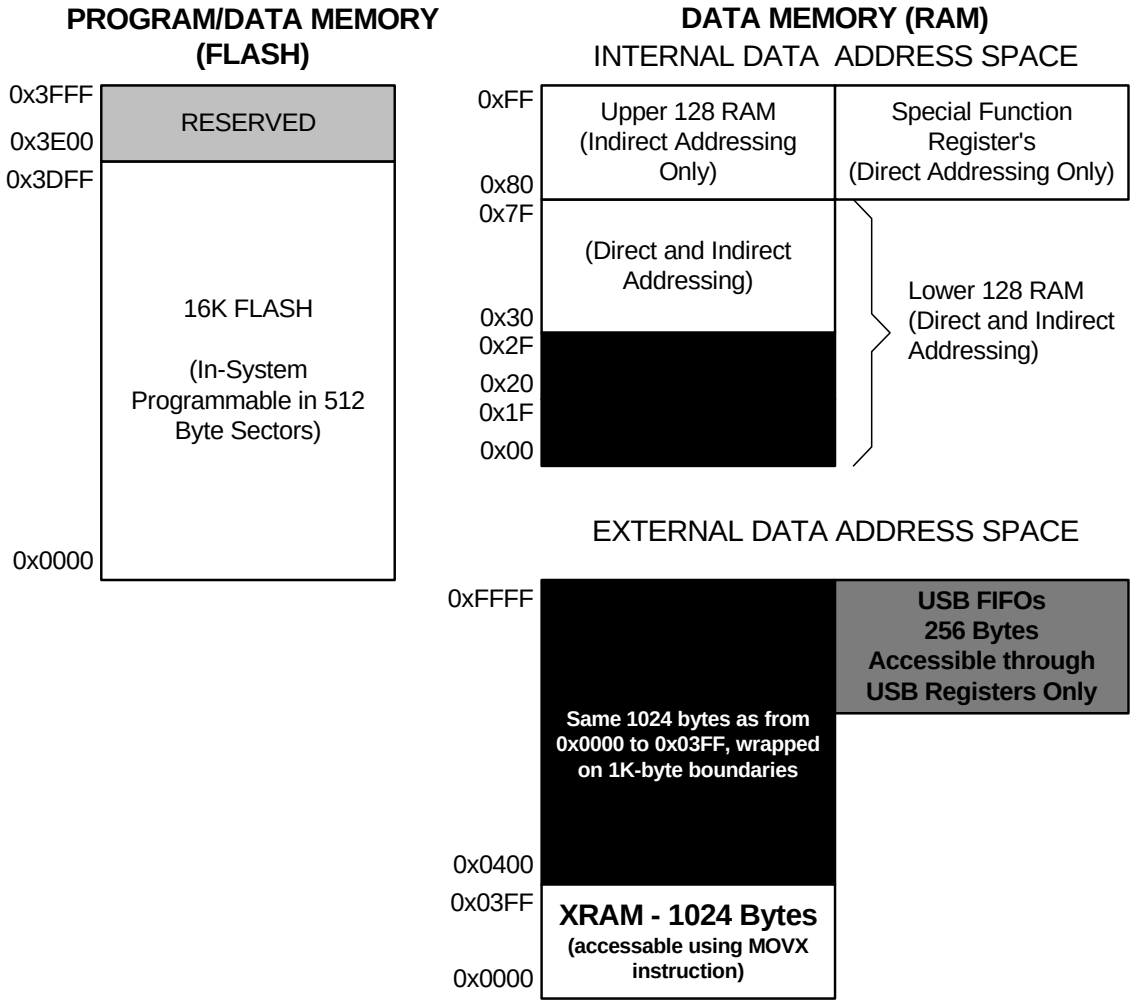

Figure 1.7. On-Board Memory Map

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.

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.					

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).

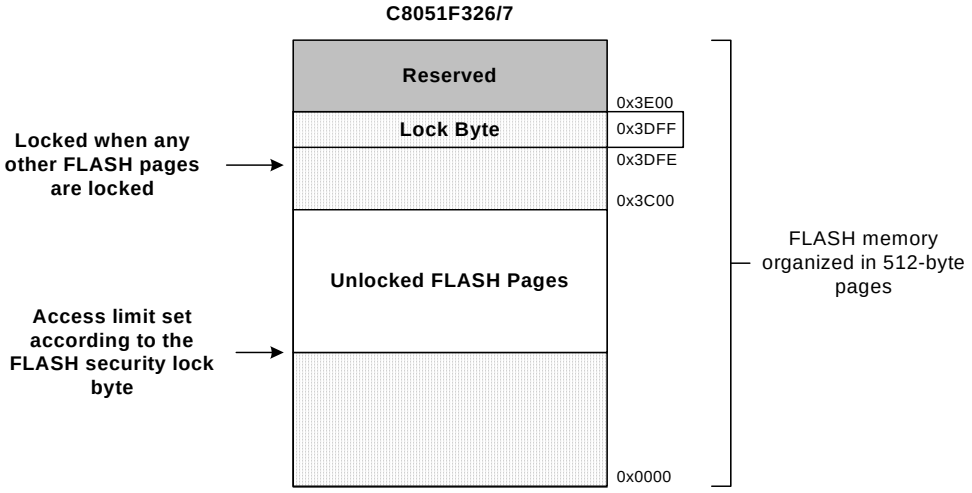

Figure 8.1. Flash Program Memory Map and Security Byte

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.

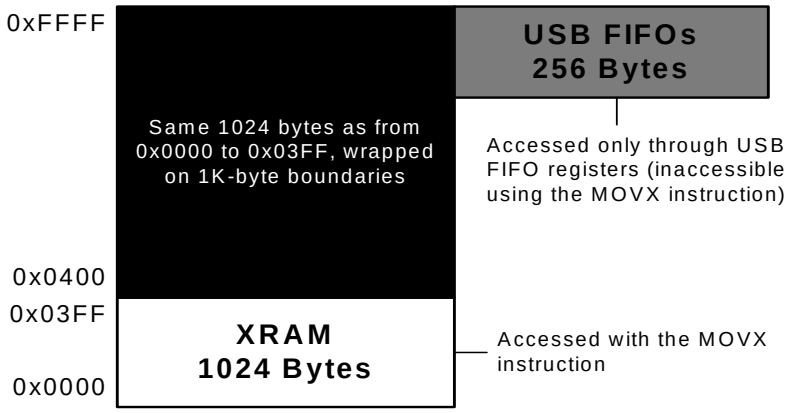

Figure 9.1. External Ram Memory Map

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.

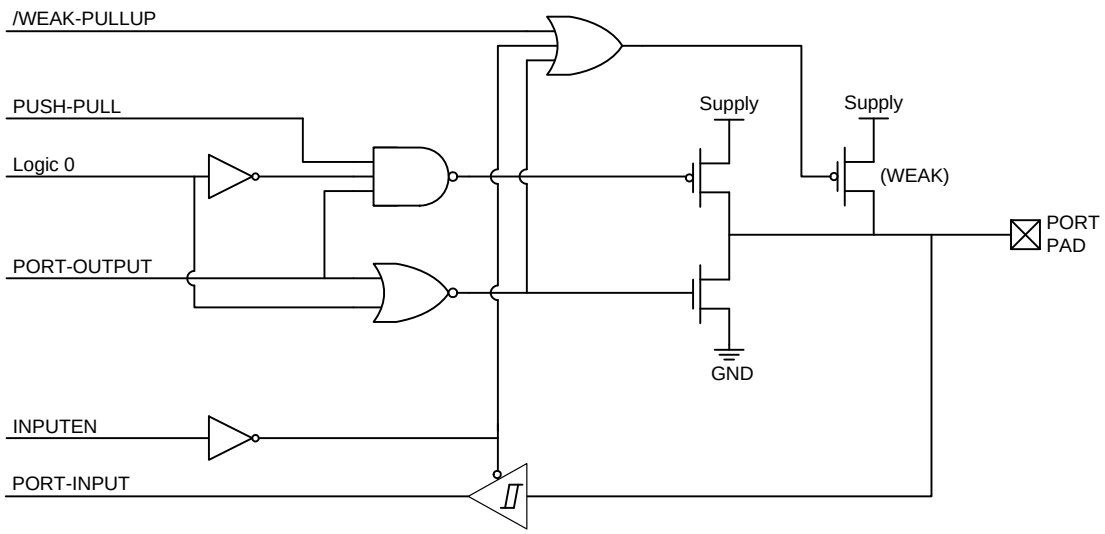

Figure 11.2. Port I/O Cell Block Diagram

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.

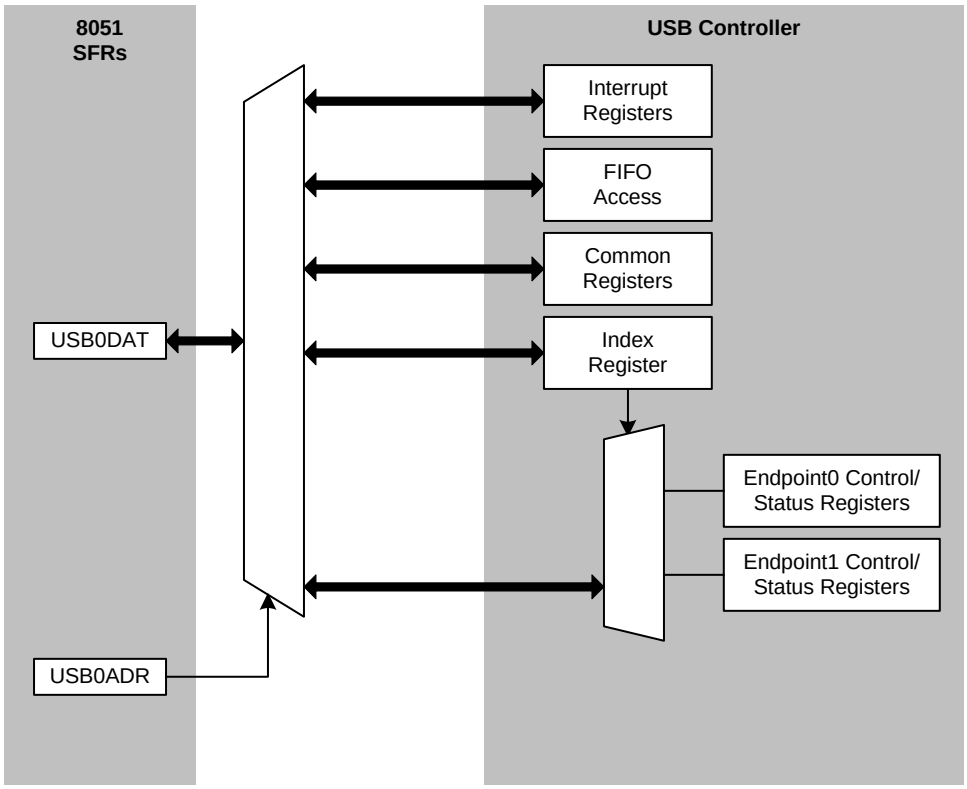

Figure 12.2. USB0 Register Access Scheme

USB Register Definition 12.8. 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.

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 12.19. 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.

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.						

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

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).

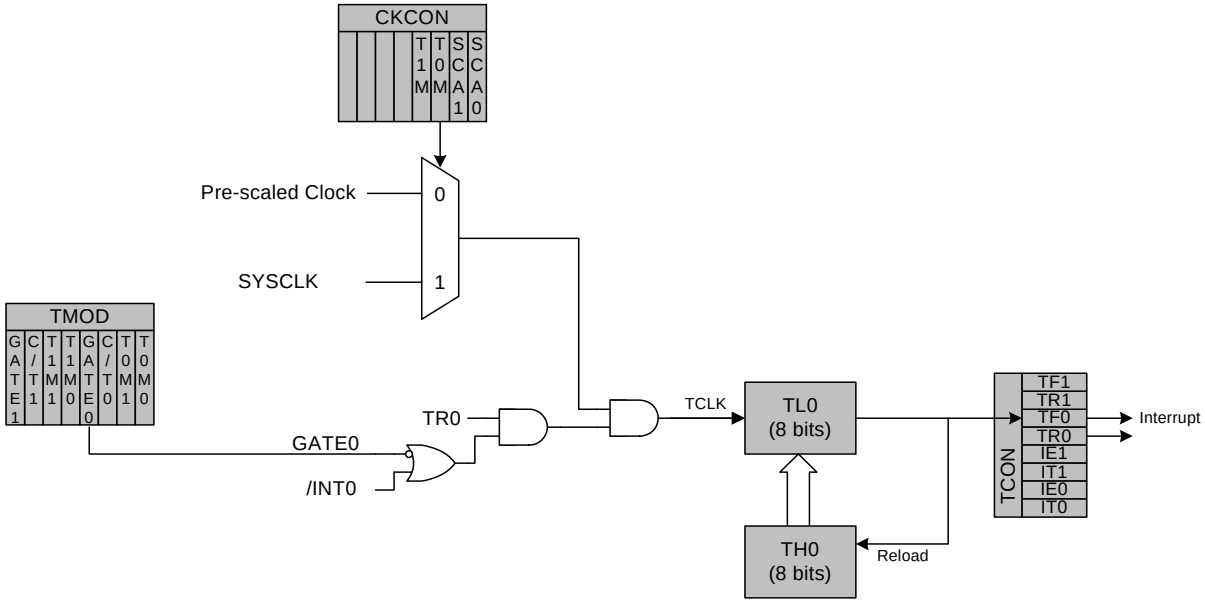

Figure 14.2. T0 Mode 2 Block Diagram