TEXAS INSTRUMENTS

TUSB3210

Universal Serial Bus General-Purpose Device Controller

Data Manual

April 2003

MSDS Bus Solutions

SLLS466B

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Contents

Section *Title* *Page*

List of Illustrations

List of Tables

1 Introduction

The TUSB3210 is a USB-based controller targeted as a general-purpose MCU with GPIO.The TUSB3210 has 8K × 8 RAM space for application development. A ROM based version of the TUSB3210 has 8K × 8 ROM space for predeveloped customer specific production applications. In addition, the programmability of the TUSB3210 makes it flexible enough to use for various other general USB I/O applications. Unique vendor identification and product identification (VID/PID) may be selected without the use of an external EEPROM. Using a 12-MHz crystal, the onboard oscillator generates the internal system clocks. The device can be programmed via an inter-IC (I^2C) serial interface at power on from an EEPROM, or optionally, the application firmware can be downloaded from a host PC via USB. The popular 8052-based microprocessor allows several third-party standard tools to be used for application development. In addition, the vast amounts of application code available in the general market can also be utilized (this may or may not require some code modification due to hardware variations).

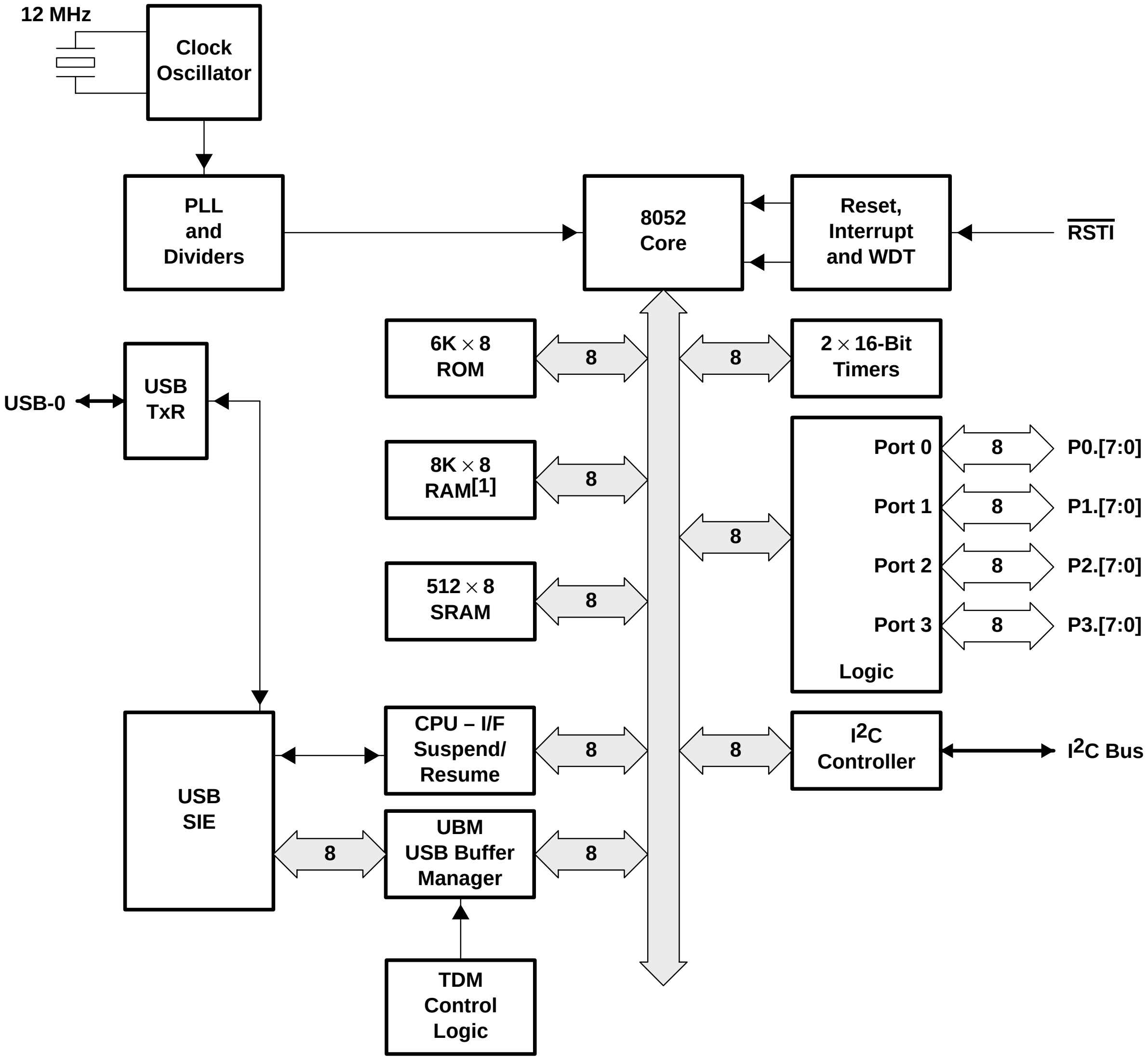
1.1 Features

- Multiproduct support with one code and one chip (up to 16 products with one chip)
- Fully compliant with the USB release 2.0 full-speed specification
- Supports 12 Mbits/s USB data rate (full speed)
- Supports USB suspend/resume and remote wake-up operation
- Integrated 8052 microcontroller with:
  - 256 × 8 RAM for internal data
  - 8K × 8 RAM code space available for downloadable firmware from host or I^2C port. [1]
  - 512 × 8 shared RAM used for data buffers and endpoint descriptor blocks (EDB) [2]
  - Four 8052 GPIO ports, ports 0,1, 2, and 3
  - Master I^2C controller for external slave device access
  - Watchdog timer
- Operates from a 12-MHz crystal
- On-chip PLL generates 48 MHz
- Supports a total of 3 input and 3 output (interrupt, bulk) endpoints
- Power-down mode
- 64-pin TQFP package
- Applications include keyboard, bar code reader, flash memory reader, general-purpose controller

[1] The TUSB3210 has 8K × 8 RAM for development.

[2] This is the buffer space for USB packet transactions.

1.2 Functional Block Diagram

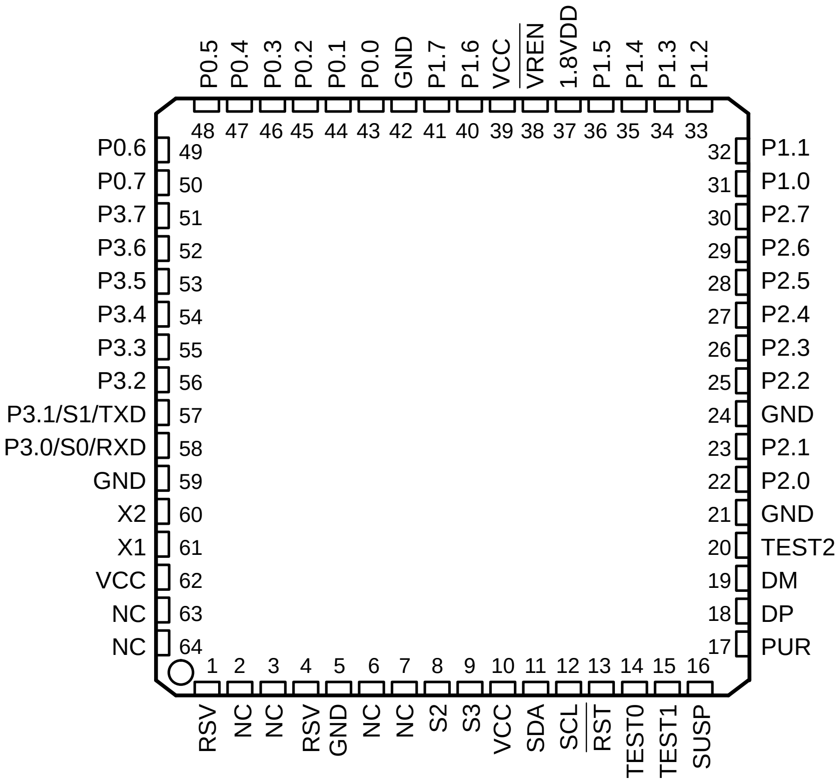


[1] 8K × 8 ROM version available. Contact TI Marketing.

Figure 1–1. TUSB3210 Block Diagram

1.3 Terminal Assignments

PM PACKAGE
(TOP VIEW)



1.4 Ordering Information

PRODUCT	PACKAGE	PACKAGE CODE	OPERATING TEMPERATURE RANGE	PACKAGE MARKING	ORDERING NUMBER	TRANSPORT MEDIA
TUSB3210PM	Plastic quad flatpack 64	PM	0°C to 70°C	TUSB3210PM	TUSB3210PM	160-piece tray

1.5 Terminal Functions

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
1.8VDD†	37	I/O	1.8 V. When $\overline{\text{VREN}}$ is low, 1.8 V must be applied to provide current for the core during suspend.
DM	19	I/O	Differential data-minus USB
DP	18	I/O	Differential data-plus USB
GND	5, 21 24, 42, 59	—	Power supply ground
NC	2, 3, 6, 7, 63, 64		No connection
P0.[0:7]	43, 44, 45,, 46, 47, 48, 49, 50	I/O	General-purpose I/O port 0 bits 0–7, Schmitt-trigger input, 100 µA active pullup, open-drain output‡
P1.[0:7]	31, 32, 33, 34, 35, 36, 40, 41	I/O	General-purpose I/O port 1 bits 0–7, Schmitt-trigger input, 100 µA active pullup, open-drain output‡
P2.[0:7]	22, 23, 25, 26, 27, 28, 29, 30	I/O	General-purpose I/O port 2 bits 0–7, Schmitt-trigger input, 100 µA active pullup, open-drain output‡
P3.0/S0/RXD	58	I/O	P3.0: General-purpose I/O port 3 bit 0, Schmitt-trigger input, 100 µA active pullup, open-drain output‡
P3.0/S0/RXD	58	I/O	S0: See Section 2.6.5.
P3.0/S0/RXD	58	I/O	RXD: Can be used as a UART interface
P3.1/S1/TXD	57	I/O	P3.1: General-purpose I/O port 3 bit 1, Schmitt-trigger input, 100 µA active pullup, open-drain output‡
P3.1/S1/TXD	57	I/O	S1: See Section 2.6.5.
P3.1/S1/TXD	57	I/O	TXD: Can be used as a UART interface
P3.2	56	I/O	General-purpose I/O port 3 bit 2, Schmitt-trigger input, 100 µA active pullup, open-drain output‡; $\overline{\text{INT0}}$ only used internally (see Section 2.9.4)
P3.3	55	I/O	General-purpose I/O port 3 bit 3, Schmitt-trigger input, 100 µA active pullup, open-drain output‡; may support $\overline{\text{INT1}}$ input, depending on configuration (see Figure 2–5)
P3.[4:7]	54, 53, 52, 51	I/O	General-purpose I/O port 3 bits 4–7, Schmitt-trigger input, 100 µA active pullup, open-drain output‡
PUR	17	O	Pullup resistor connection pin (3-state) push-pull CMOS output (±4 mA)
$\overline{\text{RST}}$	13	I	Controller master reset signal, Schmitt-trigger input, 100 µA active pullup
RSV	1, 4		Reserved (Do not connect these pins)
S2	8	I	General-purpose input, may be used for VID/PID selection under firmware control. This input has no internal pullup, so it must be driven/pulled either low or high and cannot be left unconnected.
S3	9	I	General-purpose input. This input has no internal pullup, so it must be driven/pulled either low or high and cannot be left unconnected.
SCL	12	O	Serial clock I^{2}C; push-pull output
SDA	11	I/O	Serial data I^{2}C; open-drain output‡
SUSP	16	O	Suspend status signal: suspended (HIGH); unsuspended (LOW)

† During normal operation, the internal 3.3- to 1.8-V voltage regulator of the TUSB3210 is enabled and provides power to the core. To save power during the suspend mode, the internal regulator is disabled. In this case, the pin becomes an input, and a simple external power source is required to provide power to the core. This source needs to supply a very limited amount of power (10 µA maximum) within the voltage range of 1 to 1.95 V.

‡ All open-drain output pins can sink up to 8 mA.

1.5 Terminal Functions (Continued)

TERMINAL		I/O	DESCRIPTION
NAME	NO.		
TEST0[†]	14	I	Test input0, Schmitt-trigger input, 100 μA active pullup
TEST1[†]	15	I	Test input1, Schmitt-trigger input, 100 μA active pullup
TEST2	20	I	Test input2, Schmitt-trigger input, 100 μA active pullup. This pin is reserved for testing purposes and should be left unconnected.
VCC	10, 39, 62	—	Power supply input, 3.3 V typical
$\overline{\text{VREN}}$	38	I	Voltage regulator enable: enable active LOW; disable active HIGH
X1	61	I	12-MHz crystal input
X2	60	O	12-MHz crystal output

[†] The functions controlled by TEST0 and TEST1 are shown in the following table. Because these two pins have internal pullups, they can be left unconnected for the default mode.

TEST0	TEST1	FUNCTION
0	0	Selects 48-MHz clock input (from an oscillator or other onboard clock source)
0	1	Reserved for testing purposes
1	0	Reserved for testing purposes
1	1	Selects 12-MHz crystal as clock source (default)

2 Functional Description

2.1 MCU Memory Map

Figure 2–1 illustrates the MCU memory map under boot and normal operation. It must be noted that the internal 256 bytes of IDATA are not shown since it is assumed to be in the standard 8052 location (0000 to 00FF). The shaded areas represent the internal ROM/RAM.

When the SDW bit = 0 (boot mode): The 6K-ROM is mapped to address (0000–17FF) and is duplicated in location (8000–97FF) in code space. The internal 8K-RAM is mapped to address range (0000–1FFF) in data space. Buffers, MMR and I/O are mapped to address range (FD80–FFFF) in data space.

When the SDW bit = 1 (Normal mode): The 6K-ROM is mapped to (8000–97FF) in code space. The internal 8K-RAM is mapped to address range (0000–1FFF) in code space. Buffers, MMR, and I/O are mapped to address range (FD80–FFFF) in data space.

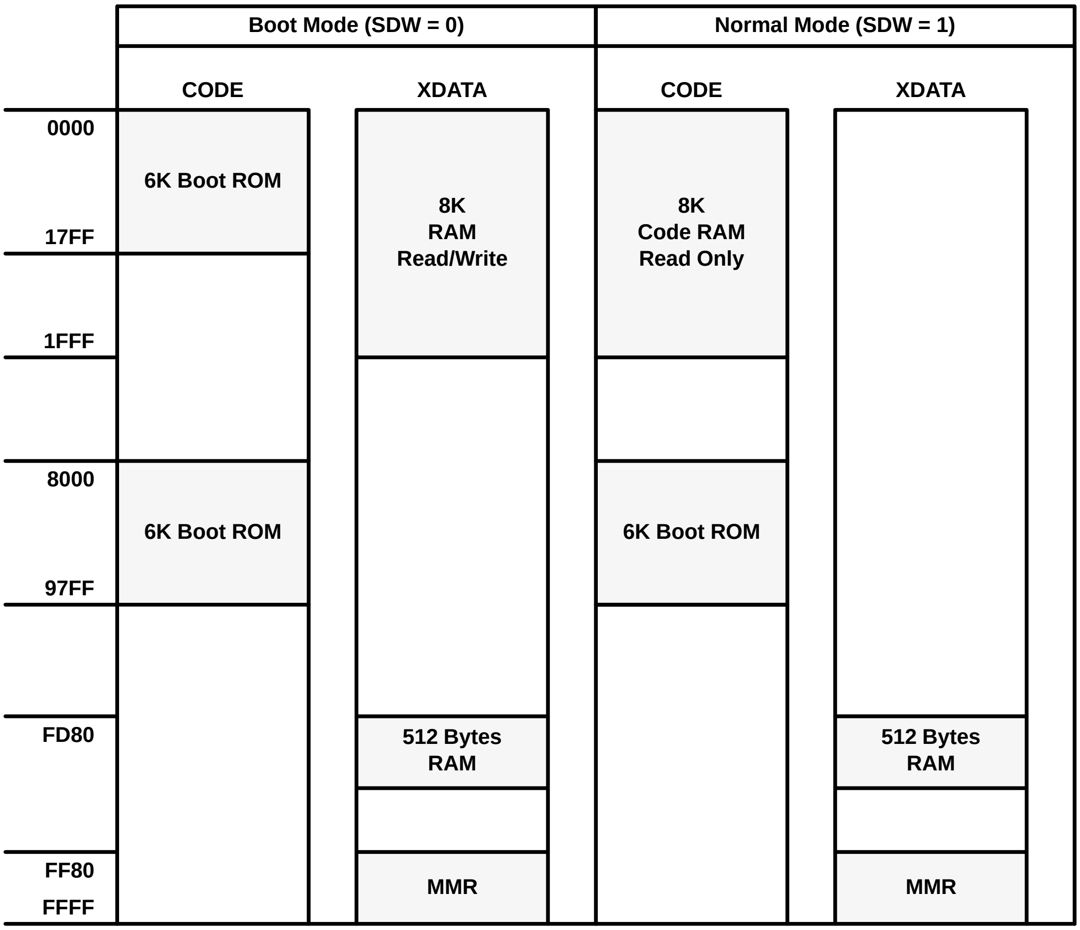


Figure 2–1. MCU Memory Map (TUSB3210)

2.2 Miscellaneous Registers

2.2.1 TUSB3210 Boot Operation

Because the code space is in RAM (with the exception of the boot ROM), the TUSB3210 firmware must be loaded from an external source. Two options for booting are available: an external serial EEPROM source can be connected to the I^{2}C bus, or the host can be used via the USB. On device reset, the SDW bit (in the ROM register) and the CONT bit in the USB control register (USBCTL) are cleared. This configures the memory space to boot mode (see memory map, Table 2–2) and keeps the device *disconnected* from the host.

The first instruction is fetched from location 0000 (which is in the 6K ROM). The 8K-RAM is mapped to XDATA space (location 0000h). The MCU executes a read from an external EEPROM and tests to determine if it contains the code (test for boot signature). If it contains the code, MCU reads from EEPROM and writes to the 8K-RAM in XDATA space. If not, the MCU proceeds to boot from the USB.

Once the code is loaded, the MCU sets SDW to 1. This switches the memory map to normal mode; i.e., the 8K-RAM is mapped to code space, and the MCU starts executing from location 0000h. Once the switch is done, the MCU sets CONT to 1 (in USBCTL register) This *connects* the device to the USB bus, resulting in the normal USB device enumeration.

2.2.2 MCNFG: MCU Configuration Register

This register is used to control the MCU clock rate.

7	6	5	4	3	2	1	0
48	XINT	RSV	R3	R2	R1	R0	SDW
R/W	R/W	R/O	R/O	R/O	R/O	R/O	R/W

BIT	NAME	RESET	FUNCTION	
0	SDW	0	This bit enables/disables boot ROM. In the ROM version of the controller, this bit has no effect.	
			SDW = 0	When clear, the MCU executes from the 6K boot ROM space. The boot ROM appears in two locations: 0000 and 8000h. The 8K RAM is mapped to XDATA space; therefore, read/write operation is possible. This bit is set by the MCU after the RAM load is completed. The MCU cannot clear this bit. It is cleared on power-up reset or function reset.
			SDW = 1	When set by the MCU, the 6K boot ROM maps to location 8000h, and the 8K RAM is mapped to code space, starting at location 0000h. At this point, the MCU executes from RAM, and write operation is disabled (no write operation is possible in code space).
4–1	R[3:0]	No effect	These bits reflect the device revision number.	
5	RSV	0	Reserved	
6	XINT	0	INT1 source control bit XINT = 0 INT1 is connected to P3.3-pin and operates as a standard INT1 interrupt. XINT = 1 INT1 is connected to the OR of port 2 inputs.	
7	RSV	0	Reserved	

2.2.3 PUR_n: GPIO Pullup Register for Port n (n = 0 to 3)

PUR_0: GPIO pullup register for port 0
PUR_1: GPIO pullup register for port 1
PUR_2: GPIO pullup register for port 2
PUR_3: GPIO pullup register for port 3

7	6	5	4	3	2	1	0
PORT_n.7	PORT_n.6	PORT_n.5	PORT_n.4	PORT_n.3	PORT_n.2	PORT_n.1	PORT_n.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
0–7	PORT_n.N (N = 0 to 7)	0	The MCU can write to this register. If the MCU sets this bit to 1, the internal pullup resistor is disconnected from the pin. If the MCU clears this bit to 0, the pullup resistor is connected to the pin. The pullup resistor is connected to the V_{CC} power supply.

2.2.4 INTCFG: Interrupt Configuration

7	6	5	4	3	2	1	0
RSV	RSV	RSV	RSV	I3	I2	I1	I0
R/O	R/O	R/O	R/O	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
0–3	I[3:0]	0010	The MCU can write to this register to set the interrupt delay time for port 2 on the MCU. The value of the lower nibble represents the delay in ms. Default after reset is 2 ms.
4–7	RSV	0	Reserved

2.2.5 WDCSR: Watchdog Timer, Control, and Status Register

A watchdog timer (WDT) with 1-ms clock is provided. The watchdog timer only works when a USB start-of-frame has been detected by the TUSB3210. If this register is not accessed for a period of 32 ms, the WDT counter resets the MCU (see Figure 2–2, Reset Diagram). When the IDL bit in PCON is set, the WDT is suspended until an interrupt is detected. At this point, the IDL bit is cleared and the WDT resumes operation. The WDE bit of this register is cleared only on power-up or USB-reset (if enabled). When the MCU writes a 1 to the WDE bit of this register the WDT starts running.

7	6	5	4	3	2	1	0
WDE	WDR	RSV	RSV	RSV	RSV	RSV	WDT
R/W	R/W	R/O	R/O	R/O	R/O	R/O	W/O

BIT	NAME	RESET	FUNCTION	
0	WDT	0	The MCU must write a 1 to this bit to prevent the WDT from resetting the MCU. If the MCU does not write a 1 in a period of 31 ms, the WDT resets the device. Writing a 0 has no effect on the WDT. (WDT is a 5-bit counter using a 1-ms CLK). This bit is read as 0.	
5–1	RSV	0	Reserved = 0	
6	WDR	0	Watchdog reset indication bit. This bit indicates if the reset occurred due to power-on reset or watchdog timer reset.	
			WDR = 0	A power-up or USB reset occurred.
			WDR = 1	A watchdog time-out reset occurred. To clear this bit, the MCU must write a 1. Writing a 0 has no effect.
7	WDE	0	Watchdog timer enable.	
			WDE = 0	Disabled
			WDE = 1	Enabled

2.2.6 PCON: Power Control Register (at SFR 87h)

7	6	5	4	3	2	1	0
SMOD	RSV	RSV	RSV	GF1	GF0	RSV	IDL
R/W	R/O	R/O	R/O	R/W	R/W	R/O	R/W

BIT	NAME	RESET	FUNCTION	
0	IDL	0	MCU idle mode bit. This bit can be set by the MCU and is cleared only by the $\overline{\text{INT1}}$ interrupt.	
			IDL = 0	The MCU is not in idle mode. This bit is cleared by the $\overline{\text{INT1}}$ interrupt logic when $\overline{\text{INT1}}$ is asserted for at least 400 µs.
			IDL = 1	The MCU is in idle mode and RAM is in low-power mode. The oscillator/APLL is off and the WDT is suspended. When in suspend mode, only $\overline{\text{INT1}}$ can be used to exit from idle state and generate an interrupt. $\overline{\text{INT1}}$ must be asserted for at least 400 µs for the interrupt to be recognized.
1	RSV	0	Reserved	
3–2	GF[1:0]	00	General-purpose bits. The MCU can write and read them.	
6–4	RSV	0	Reserved	
7	SMOD	0	Double baud-rate control bit. For more information see the UART serial interface in the M8052 core specification.	

2.3 Buffers + I/O RAM Map

The address range from FD80 to FFFF is reserved for data buffers, setup packet, endpoint descriptor blocks (EDB), and all I/O. RAM space of 512 bytes [FD80–FF7F] is used for EDB and buffers. The FF80–FFFF range is used for memory mapped registers (MMR). Table 2–1 represents the internal XDATA space allocation.

Table 2–1. XDATA Space

DESCRIPTION	ADDRESS RANGE	
Internal memory-mapped registers (MMR)	FFFF ↑ FF80	
Endpoint descriptor blocks (EDB)	FF7F ↑ FF08	**512-Byte RAM**
Setup packet buffer	FF07 ↑ FF00	
Input endpoint-0 buffer	FEFF ↑ FEF8	
Output endpoint-0 buffer	FEF7 ↑ FEF0	
Data buffers (368 bytes)	FEEF ↑ FD80	

Table 2–2. Memory-Mapped Register Summary (XDATA Range = FF80 → FFFF)

ADDRESS	REGISTER	DESCRIPTION
FFFF	FUNADR	FUNADR: Function address register
FFFE	USBSTA	USBSTA: USB status register
FFFD	USBMSK	USBMSK: USB interrupt mask register
FFFC	USBCTL	USBCTL: USB control register
↑	RESERVED	
FFF6	VIDSTA	VIDSTA: VID/PID status register
↑	RESERVED	
FFF3	I2CADR	I2CADR: I^{2}C address register
FFF2	I2CDAI	I2CDAI: I^{2}C data-input register
FFF1	I2CDAO	I2CDAO: I^{2}C data-output register
FFF0	I2CSTA	I2CSTA: I^{2}C status and control register
↑	RESERVED	
FF97	PUR3	Port 3 pullup resistor register
FF96	PUR2	Port 2 pullup resistor register
FF95	PUR1	Port 1 pullup resistor register
FF94	PUR0	Port 0 pullup resistor register
FF93	WDCSR	WDCSR: Watchdog timer, control and status register
FF92	VECINT	VECINT: Vector interrupt register
FF91	RESERVED	
FF90	MCNFG	MCNFG: MCU configuration register
↑	RESERVED	
FF84	INTCFG	INTCFG: Interrupt delay configuration register
FF83	OEPBCNT_0	OEPBCNT_0: Output endpoint-0 byte count register
FF82	OEPCNFG_0	OEPCNFG_0: Output endpoint-0 configuration register
FF81	IEPBCNT_0	IEPBCNT_0: Input endpoint-0 byte count register
FF80	IEPCNFG_0	IEPCNFG_0: Input endpoint-0 configuration register

2.4 Endpoint Descriptor Block (EDB-1 to EDB-3)

Data transfers between USB, MCU and external devices are defined by an endpoint descriptor block (EDB). Four input and four output EDBs are provided. With the exception of EDB-0 (I/O endpoint 0), all EDBs are located in SRAM as shown in Table 2–3. Each EDB contains information describing the X and Y buffers. In addition, it provides general status information.

Table 2–3. EDB and Buffer Allocations in XDATA

ADDRESS	SIZE	DESCRIPTION
FF7F ↑ FF60	32 bytes	RESERVED
FF5F ↑ FF58	8 bytes	Input endpoint 3: configuration
FF57 ↑ FF50	8 bytes	Input endpoint 2: configuration
FF4F ↑ FF48	8 bytes	Input endpoint 1: configuration
FF47 ↑ FF20	40 bytes	RESERVED
FF1F ↑ FF18	8 bytes	Output endpoint 3: configuration
FF17 ↑ FF10	8 bytes	Output endpoint 2: configuration
FF0F ↑ FF08	8 bytes	Output endpoint 1: configuration
FF07 ↑ FF00	8 bytes	Setup packet block
FEFF ↑ FEF8	8 bytes	Input endpoint 0: buffer
FEF7 ↑ FEF0	8 bytes	Output endpoint 0: buffer
FEEF ↑ FD80	368 bytes	Top of buffer space Buffer space Start of buffer space

Table 2–4 lists the EDB entries for EDB-1 to EDB-3. EDB-0 registers are described separately.

Table 2–4. EDB Entries in RAM (n = 1 to 3)

Offset	ENTRY NAME	DESCRIPTION
07	EPSIZXY_n	I/O endpoint_n: X/Y buffer size
06	EPBCTY_n	I/O endpoint_n: Y byte count
05	EPBBAY_n	I/O endpoint_n: Y buffer base address
04	SPARE	Not used
03	SPARE	Not used
02	EPBCTX_n	I/O endpoint_n: X byte count
01	EPBBAX_n	I/O endpoint_n: X buffer base address
00	EPCNF_n	I/O endpoint_n: configuration

2.4.1 OEPCNF_n: Output Endpoint Configuration (n = 1 to 3)

7	6	5	4	3	2	1	0
UBME	ISO	TOGLE	DBUF	STALL	USBIE	RSV	RSV
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/O

BIT	NAME	RESET	FUNCTION	
1–0	RSV	0	Reserved	
2	USBIE	x	USB interrupt enable on transaction completion. Set/clear by MCU.	
			USBIE = 0	No interrupt
			USBIE = 1	Interrupt on transaction completion
3	STALL	0	USB stall condition indication. Set/clear by MCU.	
			STALL = 0	No stall
			STALL = 1	USB stall condition. If set by MCU, a STALL handshake is initiated and the bit is cleared by MCU.
4	DBUF	x	Double buffer enable. Set/clear by MCU.	
			DBUF = 0	Primary buffer only (X–buffer only)
			DBUF = 1	Toggle bit selects buffer
5	TOGLE	x	USB toggle bit. This bit reflects the toggle sequence bit of DATA0, DATA1.	
6	ISO	x	ISO = 0	Non-isochronous transfer. This bit must be cleared by MCU since only Non-isochronous transfer is supported.
7	UBME	x	UBM enable/disable bit. Set/clear by MCU.	
			UBME = 0	UBM cannot use this endpoint.
			UBME = 1	UBM can use this endpoint.

2.4.2 OEPBBAX_n: Output Endpoint X-Buffer Base-Address (n = 1 to 3)

7	6	5	4	3	2	1	0
A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
7–0	A[10:3]	x	A[10:3] of X-buffer base address (padded with 3 LSB of zeros for a total of 11 bits). This value is set by the MCU. UBM or DMA uses this value as the start address of a given transaction. Furthermore, UBM or DMA does not change this value at the end of a transaction.

2.4.3 OEPBCTX_n: Output Endpoint X-Byte Count (n = 1 to 3)

7	6	5	4	3	2	1	0
NAK	C_6	C_5	C_4	C_3	C_2	C_1	C_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	C[6:0]	x	**X-Buffer Byte count:** 000 0000b → Count = 0 000 0001b → Count = 1 byte ⋮ 011 1111b → Count = 63 bytes 100 0000b → Count = 64 bytes Any value ≥ 100 0001b produces unpredictable results.
7	NAK	x	NAK = 0 No valid data in buffer. Ready for host out NAK = 1 Buffer contains a valid packet from host (host-out request is NAK)

2.4.4 OEPBBAY_n: Output Endpoint Y-Buffer Base-Address (n = 1 to 3)

7	6	5	4	3	2	1	0
A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
7–0	A[10:3]	x	A[10:3] of Y-buffer base address (padded with 3 LSB of zeros for a total of 11 bits). This value is set by the MCU. UBM or DMA uses this value as the start address of a given transaction. Furthermore, UBM or DMA does not change this value at the end of a transaction.

2.4.5 OEPBCTY_n: Output Endpoint Y-Byte Count (n = 1 to 3)

7	6	5	4	3	2	1	0
NAK	C_6	C_5	C_4	C_3	C_2	C_1	C_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	C[6:0]	x	**Y-Buffer Byte count:** 000 0000b → Count = 0 000 0001b → Count = 1 byte ⋮ 011 1111b → Count = 63 bytes 100 0000b → Count = 64 bytes Any value ≥ 100 0001b produces unpredictable results.
7	NAK	x	NAK = 0 No valid data in buffer. Ready for host out NAK = 1 Buffer contains a valid packet from host (host-out request is NAK)

2.4.6 OEPSIZXY_n: Output Endpoint X-/Y-Byte Count (n = 1 to 3)

7	6	5	4	3	2	1	0
RSV	S_6	S_5	S_4	S_3	S_2	S_1	S_0
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	S[6:0]	x	**X- and Y-Buffer size:** 000 0000b → Count = 0 000 0001b → Count = 1 byte ⋮ 011 1111b → Count = 63 bytes 100 0000b → Count = 64 bytes Any value ≥ 100 0001b produces unpredictable results.
7	RSV	0	Reserved

2.4.7 IEPCNF_n: Input Endpoint Configuration (n = 1 to 3)

7	6	5	4	3	2	1	0
UBME	ISO	TOGLE	DBUF	STALL	USBIE	RSV	RSV
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/O

BIT	NAME	RESET	FUNCTION	
1–0	RSV	x	Reserved = 0	
2	USBIE	x	USB interrupt enable on transaction completion.	
			USBIE = 0	No interrupt
			USBIE = 1	Interrupt on transaction completion
3	STALL	0	USB stall condition indication. Set by UBM, but can be set/cleared by MCU.	
			STALL = 0	No stall
			STALL = 1	USB stall condition. If set by MCU, a STALL handshake is initiated and the bit is cleared automaticallly.
4	DBUF	x	Double buffer enable	
			DBUF = 0	Primary buffer only (X-buffer only)
			DBUF = 1	Toggle bit selects buffer
5	TOGLE	x	USB Toggle bit. This bit reflects the toggle sequence bit of DATA0, DATA1	
6	ISO	x	ISO = 0	Non-isochronous transfer. This bit must be cleared by MCU because only non-isochronous transfer is supported.
7	UBME	x	UBM enable/disable bit. Set/clear by MCU.	
			UBME = 0	UBM cannot use this endpoint.
			UBME = 1	UBM can use this endpoint.

2.4.8 IEPBBAX_n: Input Endpoint X-Buffer Base-Address (n = 1 to 3)

7	6	5	4	3	2	1	0
A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
7–0	A[10:3]	x	A[10:3] of X-buffer base address (padded with 3 LSB of zeros for a total of 11 bits). This value is set by the MCU. UBM or DMA uses this value as the start address of a given transaction. Furthermore, UBM or DMA does not change this value at the end of a transaction.

2.4.9 IEPBCTX_n: Input Endpoint X-Byte Base-Address (n = 1 to 3)

7	6	5	4	3	2	1	0
NAK	C_6	C_5	C_4	C_3	C_2	C_1	C_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	C[6:0]	x	**X-Buffer Byte count:** 000 0000b → Count = 0 000 0001b → Count = 1 byte ⋮ 011 1111b → Count = 63 bytes 100 0000b → Count = 64 bytes Any value ≥ 100 0001b produces unpredictable results.
7	NAK	x	NAK = 0 Buffer contains a valid packet for host-in transaction NAK = 1 Buffer is empty (host-in request is NAK)

2.4.10 IEPBBAY_n: Input Endpoint Y-Buffer Base-Address (n = 1 to 3)

7	6	5	4	3	2	1	0
A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
7–0	A[10:3]	x	A[10:3] of Y-buffer base address (padded with 3 LSB of zeros for a total of 11 bits). This value is set by the MCU. UBM or DMA uses this value as the start address of a given transaction. Furthermore, UBM or DMA does not change this value at the end of a transaction.

2.4.11 IEPBCTY_n: Input Endpoint Y-Byte Count (n = 1 to 3)

7	6	5	4	3	2	1	0
NAK	C_6	C_5	C_4	C_3	C_2	C_1	C_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	C[6:0]	x	**X-BufferByte count:** 000 0000b → Count = 0 000 0001b → Count = 1 byte ⋮ 011 1111b → Count = 63 bytes 100 0000b → Count = 64 bytes Any value ≥ 100 0001b produces unpredictable results.
7	NAK	x	NAK = 0 Buffer contains a valid packet for host-in transaction NAK = 1 Buffer is empty (host-in request is NAK)

2.4.12 IEPSIZXY_n: Input Endpoint X-/Y-Buffer Size (n = 1 to 3)

7	6	5	4	3	2	1	0
RSV	S_6	S_5	S_4	S_3	S_2	S_1	S_0
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	S[6:0]	x	**X- and Y-Buffer size:** 000 0000b → Count = 0 000 0001b → Count = 1 byte ⋮ 011 1111b → Count = 63 bytes 100 0000b → Count = 64 bytes Any value ≥ 100 0001b produces unpredictable results.
7	RSV	x	Reserved

2.5 Endpoint-0 Descriptor Registers

Unlike EDB-1 to EDB-3, which are defined as memory entries in SRAM, endpoint-0 is described by a set of 4 registers (two for output and two for input). Table 2–5 defines the registers and their respective addresses used for EDB-0 description. EDB-0 has no *Base-Address Register*, because these addresses are hardwired to FEF8 and FEF0. Note that the bit positions have been preserved to provide consistency with EDB-n (n = 1 to 3).

Table 2–5. Input/Output EDB-0 Registers

ADDRESS	REGISTER NAME	DESCRIPTION	BASE ADDRESS
FF83	OEPBCNT_0	Output endpoint_0: byte-count register	
FF82	OEPCNFG_0	Output endpoint_0: configuration register	FEF0
FF81	IEPBCNT_0	Input endpoint_0: byte-count register	
FF80	IEPCNFG_0	Input endpoint_0: configuration register	FEF8

2.5.1 IEPCNFG_0: Input Endpoint-0 Configuration Register

7	6	5	4	3	2	1	0
UBME	RSV	TOGLE	RSV	STALL	USBIE	RSV	RSV
R/W	R/O	R/O	R/O	R/W	R/W	R/O	R/O

BIT	NAME	RESET	FUNCTION
1–0	RSV	0	Reserved
2	USBIE	0	USB interrupt enable on transaction completion. Set/clear by MCU USBIE = 0 No interrupt USBIE = 1 Interrupt on transaction completion
3	STALL	0	USB stall condition indication. Set/clear by MCU. STALL = 0 No stall STALL = 1 USB stall condition. If set by MCU, a STALL handshake is initiated and the bit is cleared automaticallly by next setup transaction.
4	RSV	0	Reserved
5	TOGLE	0	USB toggle bit. This bit reflects the toggle sequence bit of DATA0, DATA1.
6	RSV	0	Reserved
7	UBME	0	UBM enable/disable bit. Set/clear by MCU. UBME = 0 UBM cannot use this endpoint. UBME = 1 UBME = 1 UBM can use this endpoint.

2.5.2 IEPBCNT_0: Input Endpoint-0 Byte-Count Register

7	6	5	4	3	2	1	0
NAK	RSV	RSV	RSV	C_3	C_2	C_1	C_0
R/W	R/O	R/O	R/O	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
3–0	C[3:0]	0000	**Byte count:** 0000b → Count = 0 ⋮ 0111b → Count = 7 1000b → Count = 8 1001b to 1111b are reserved. (If used, defaults to 8)
6–4	RSV	0	Reserved
7	NAK	1	NAK = 0 Buffer contains a valid packet for host-in transaction. NAK = 1 Buffer is empty (host-in request is NAK)

2.5.3 OEPCNFG_0: Output Endpoint-0 Configuration Register

7	6	5	4	3	2	1	0
UBME	RSV	TOGLE	RSV	STALL	USBIE	RSV	RSV
R/W	R/O	R/O	R/O	R/W	R/W	R/O	R/O

BIT	NAME	RESET	FUNCTION	
1–0	RSV	0	Reserved	
2	USBIE	0	USB interrupt enable on transaction completion. Set/clear by MCU	
			USBIE = 0	No interrupt
			USBIE = 1	Interrupt on transaction completion
3	STALL	0	USB stall condition indication. Set/clear by MCU.	
			STALL = 0	No stall
			STALL = 1	USB stall condition. If set by MCU, a STALL handshake is initiated and the bit is cleared automaticallly.
4	RSV	0	Reserved	
5	TOGLE	0	USB toggle bit. This bit reflects the toggle sequence bit of DATA0, DATA1.	
6	RSV	0	Reserved	
7	UBME	0	UBM enable/disable bit. Set/clear by MCU.	
			UBME = 0	UBM cannot use this endpoint.
			UBME = 1	UBM can use this endpoint.

2.5.4 OEPBCNT_0: Output Endpoint-0 Byte-Count Register

7	6	5	4	3	2	1	0
NAK	RSV	RSV	RSV	C_3	C_2	C_1	C_0
R/W	R/O	R/O	R/O	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
3–0	C[3:0]	0000	**Byte count:** 0000b → Count = 0 ⋮ 0111b → Count = 7 1000b → Count = 8 1001b to 1111b are reserved. (If used, defaults to 8)
6–4	RSV	0	Reserved = 0
7	NAK	1	NAK = 0 No valid data in buffer. Ready for host out NAK = 1 Buffer contains a valid packet from host. (NAK the host)

2.6 USB Registers

2.6.1 FUNADR: Function Address Register

This register contains the device function address.

7	6	5	4	3	2	1	0
RSV	FA6	FA5	FA4	FA3	FA2	FA1	FA0
R/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
6–0	FA[6:0]	0000000	These bits define the current device address assigned to the function. MCU writes a value to this register as a result of *SET-ADDRESS* host command.
7	RSV	0	Reserved

2.6.2 USBSTA: USB Status Register

All bits in this register are set by the hardware and are cleared by MCU when writing a 1 to the proper bit location (writing a 0 has no effect). In addition, each bit can generate an interrupt if its corresponding mask bit is set (R/C notation indicates read and clear only by MCU).

7	6	5	4	3	2	1	0
RSTR	SUSR	RESR	PWOFF	PWON	SETUP	RSV	STPOW
R/C	R/C	R/C	R/C	R/C	R/C	R/O	R/C

BIT	NAME	RESET	FUNCTION	
0	STPOW	0	SETUP overwrite bit. Set by hardware when setup packet is received while there is already a packet in the setup buffer.	
			STPOW = 0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			STPOW = 1	SETUP overwrite
1	RSV	0	Reserved	
2	SETUP	0	SETUP transaction received bit. As long as SETUP is 1, IN and OUT on endpoint-0 is NAK regardless of the value of their real NAK bits.	
			SETUP = 0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			SETUP = 1	SETUP transaction received.
3	PWON	0	Power on request for port 3. This bit indicates if power-on to port 3 has been received. This bit generates a PWON interrupt (if enabled).	
			PWON = 0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			PWON = 1	Power on to port 3 has been received.
4	PWOFF	0	Power off request for port 3. This bit indicates whether power-off to port 3 has been received. This bit generates a PWOFF interrupt (if enabled).	
			PWOFF = 0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			PWOFF = 1	Power off to port 3 has been received.
5	RESR	0	Function resume request bit	
			RESR = 0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			RESR = 1	Function resume is detected
6	SUSR	0	Function suspended request bit. This bit is set in response to a global or selective suspend condition.	
			SUSR =0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			SUSR =1	Function suspend is detected.
7	RSTR	0	Function reset request bit. This bit is set in response to host initiating a port reset. This bit is not affected by USB function reset.	
			RSTR = 0	MCU can clear this bit by writing a 1. (Writing 0 has no effect.)
			RSTR = 1	Function reset is detected.

2.6.3 USBMSK: USB Interrupt Mask Register

7	6	5	4	3	2	1	0
RSTR	SUSR	RESR	PWOFF	PWON	SETUP	RSV	STPOW
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/W

BIT	NAME	RESET	FUNCTION	
0	STPOW	0	SETUP overwrite interrupt enable bit	
			STPOW = 0	STPOW interrupt disabled
			STPOW = 1	STPOW interrupt enabled
1	RSV	0	Reserved = 0	
2	SETUP	0	SETUP interrupt enable bit	
			SETUP = 0	SETUP interrupt disabled
			SETUP = 1	SETUP interrupt enabled
3	PWON	0	Power-on interrupt enable bit	
			PWON = 0	PWON interrupt disabled
			PWON = 1	PWON interrupt enabled
4	PWOFF	0	Power-off interrupt enable bit	
			PWOFF = 0	PWOFF interrupt disabled
			PWON = 1	PWOFF interrupt enabled
5	RESR	0	Function resume interrupt enable	
			RESR = 0	Function resume interrupt disabled
			RESR = 1	Function resume interrupt enabled
6	SUSR	0	Function suspend interrupt enable	
			SUSR = 0	Function suspend interrupt disabled
			SUSR = 1	Function suspend interrupt enabled
7	RSTR	0	Function reset interrupt enable	
			RSTR = 0	Function reset interrupt disabled
			RSTR = 1	Function reset interrupt enabled

2.6.4 USBCTL: USB Control Register

Unlike the other registers, this register is cleared by the power-up-reset signal only. The USB-reset cannot reset this register (see the reset diagram in Figure 2–2).

7	6	5	4	3	2	1	0
CONT	RSV	RWUP	FRSTE	RWE	B/S	SIR	DIR
R/W	R/O	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION	
0	DIR	0	As a response to a setup packet, the MCU decodes the request and sets or clears this bit to reflect the data transfer direction.	
			DIR = 0	USB data OUT transaction (from host to TUSB3210)
			DIR = 1	USB data IN transaction (from TUSB3210 to host)
1	SIR	0	SETUP interrupt status bit. This bit is controlled by the MCU to indicate to the hardware when SETUP interrupt is being served.	
			SIR = 0	SETUP interrupt is not served. MCU clears this bit before exiting the SETUP interrupt routine.
			SIR = 1	SETUP interrupt is in progress. MCU sets this bit when servicing the SETUP interrupt.
2	B/S	0	Bus/self power-control bit	
			B/S = 0	The device is bus-powered.
			B/S = 1	The device is self-powered.
3	RWE	0	Remote wake-up enable bit	
			RWE = 0	MCU clears this bit when host sends command to clear the feature.
			RWE = 1	MCU writes 1 to this bit when host sends *set device feature command* to enable the remote wake-up feature
4	FRSTE	1	Function reset connection bit. This bit connects/disconnects the USB function reset from the MCU reset.	
4	FRSTE	1	FRSTE = 0	Function reset is not connected to MCU reset.
4	FRSTE	1	FRSTE = 1	Function reset is connected to MCU reset.
5	RWUP	0	Device remote wake-up request. This bit is set by MCU and is cleared automatically.	
			RWUP = 0	Writing a 0 to this bit has no effect.
			RWUP = 1	When MCU writes a 1, a remote wake-up pulse is generated.
6	RSV	0	Reserved	
7	CONT	0	Connect/disconnect bit	
			CONT = 0	Upstream port is disconnected. Pullup disabled
			CONT = 1	Upstream port is connected. Pullup enabled

2.6.5 VIDSTA: VID/PID Status Register

This register is used to read the value on four external pins. The firmware can use this value to select one of the vendor identification/product identifications (VID/PID) stored in memory. The TUSB3210/D supports up to 16 unique VID/PIDs with application code to support different products. This provides a unique opportunity for original equipment manufacturers (OEM) to have one device ROM programmed to support up to 16 different product lines by using S0–S3 to select VID/PID and behavioral application code for the selected product.

7	6	5	4	3	2	1	0
RSV	RSV	RSV	RSV	S3	S2	S1	S0
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

BIT	NAME	RESET	FUNCTION
3–0	S[3:0]	x	VID/PID selection bits. These bits reflect the status of the external pins as defined by Table 2–6. Note that a pin tied low is reflected as 0 and a pin tied high is reflected as a 1.
7–4	RSV	0	Reserved = 0

Table 2–6. External Pin Mapping to S[3:0] in VIDSTA Register

VIDSTA REGISTER	PIN		COMMENTS
S[3:0]	NO.	NAME	
S0	58	P3.0	Dual function P3.0 I/O or S0 input
S1	57	P3.1	Dual function P3.1 I/O or S1 input
S2	8	S2	S2-pin is input
S3	9	S3	S3-pin is input

2.7 Function Reset and Power-Up Reset Interconnect

Figure 2–2 represents the logical connection of the USB-function-reset ($\overline{\text{USBR}}$) and power-up-reset ($\overline{\text{RST}}$)-pins. The internal $\overline{\text{RESET}}$ signal is generated from the $\overline{\text{RST}}$ pin ($\overline{\text{PURS}}$ signal) or from the USB-reset ($\overline{\text{USBR}}$ signal). The $\overline{\text{USBR}}$ can be enabled or disabled by the FRSTE bit in the USBCTL register (on power up FRSTE = 0). The internal $\overline{\text{RESET}}$ is used to reset all registers and logic, with the exception of the USBCTL and MISCTL registers. The USBCTL and MCU configuration registers (MCNFG) are cleared by $\overline{\text{PURS}}$ signal only.

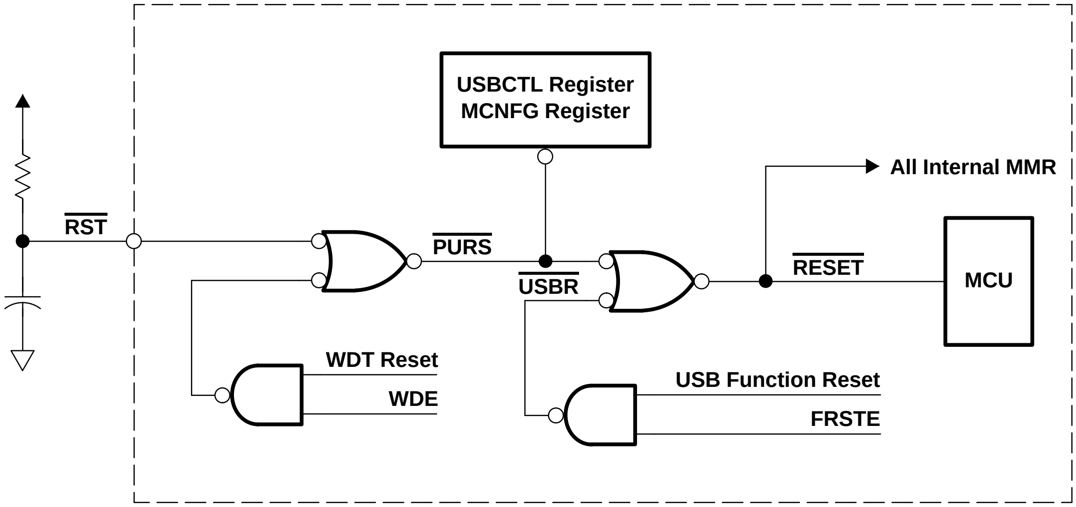


Figure 2–2. Reset Diagram

2.8 Pullup Resistor Connect/Disconnect

After reading firmware into RAM the TUSB3210 can re-enumerate using the new firmware (no need to physically disconnect and re-connect the cable). Figure 2–3 shows an equivalent circuit implementation for *Connect* and *Disconnect* from a USB upstream port (also see Figure 4–3b). When the CONT bit in the USBCTL register is 1, the CMOS driver sources V_{DD} to the pullup resistor (PUR pin) presenting a normal connect condition to the USB hub (high speed). When the CONT bit is 0, the PUR pin is driven low. In this state, the 1.5-kΩ resistor is connected to GND, resulting in device *disconnection* state. The PUR driver is a CMOS driver that can provide V_{DD}–0.1 V minimum at 8 mA source current.

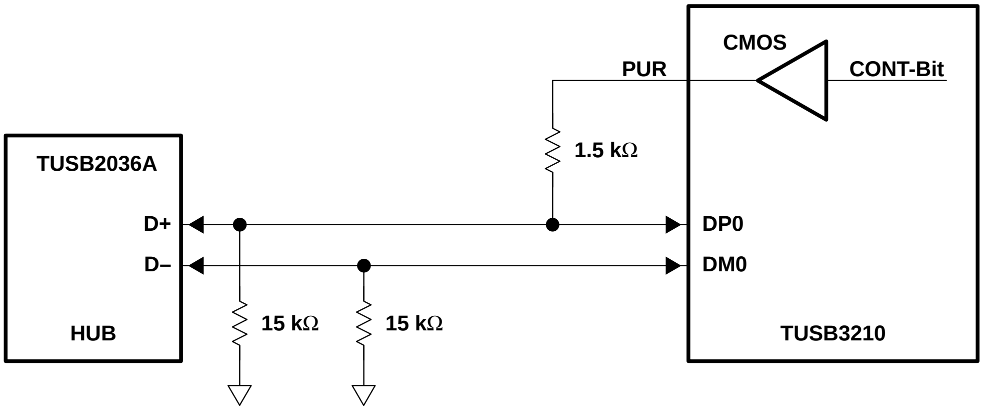


Figure 2–3. Pullup Resistor Connect/Disconnect Circuit

2.9 8052 Interrupt and Status Registers

All seven 8052-standard interrupt sources are preserved. SIE is the standard interrupt enable register, which controls the seven interrupt sources. All the additional interrupt sources are connected together as an OR to generate $\overline{\text{INT0}}$. $\overline{\text{INT0}}$ signal is provided to interrupt the MCU (see interrupt connection diagram, Figure 2–5).

Table 2–7. 8052 Interrupt Location Map

INTERRUPT SOURCE	DESCRIPTION	START ADDRESS	COMMENTS
ET2	Timer-2 interrupt	002Bh	
ES	UART interrupt	0023h	
ET1	Timer-1 interrupt	001Bh	
EX1	Internal $\overline{\text{INT1}}$ or INT1	0013h	Used for P2[7:0] interrupt
ET0	Timer-0 interrupt	000Bh	
$\overline{\text{INT0}}$	Internal $\overline{\text{INT0}}$	0003h	Used for all internal peripherals
Reset		0000h	

2.9.1 8052 Standard Interrupt Enable Register

7	6	5	4	3	2	1	0
EA	RSV	ET2	ES	ET1	EX1	ET0	$\overline{\text{INT0}}$
R/W	R/O	R/O	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION	
0	$\overline{\text{INT0}}$	0	Enable or disable interrupt-0	
			$\overline{\text{INT0}}$ = 0	Interrupt-0 is disabled.
			$\overline{\text{INT0}}$ = 1	Interrupt-0 is enabled.
1	ET0	0	Enable or disable timer-0 interrupt	
			ET0 = 0	Timer-0 interrupt is disabled.
			ET0 = 1	Timer-0 interrupt is enabled.
2	EX1	0	Enable or disable interrupt-1	
			EX1 = 0	Interrupt-1 is disabled.
			EX1 = 1	Interrupt-1 is enabled.
3	ET1	0	Enable or disable timer-1 interrupt	
			ET1 = 0	Timer-1 interrupt is disabled.
			ET1 = 1	Timer-1 interrupt is enabled.
4	ES	0	Enable or disable serial port interrupts	
			ES = 0	Serial port interrupt is disabled.
			ES = 1	Serial port interrupt is enabled.
5	ET2	0	Enable or disable timer-2 interrupt	
			ET1 = 0	Timer-2 interrupt is disabled.
			ET1 = 1	Timer-2 interrupt is enabled.
6	RSV	0		Reserved
7	EA	0	Enable or disable all interrupts (global disable)	
			EA = 0	Disable all interrupts.
			EA = 1	Each interrupt source is individually controlled.

2.9.2 Additional Interrupt Sources

All nonstandard 8052 interrupts (USB, I^{2}C, etc.) are connected as an OR to generate an internal $\overline{\text{INT0}}$. It must be noted that the external $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ are not used. Furthermore, $\overline{\text{INT0}}$ must be programmed as an active-low level interrupt (not edge triggered). A vector interrupt register is provided to identify all interrupt sources (see vector interrupt register definition, Section 2.9.3). Up to 64 interrupt vectors are provided. It is the responsibility of the MCU to read the vector and dispatch the proper interrupt routine.

2.9.3 VECINT: Vector Interrupt Register

This register contains a vector value identifying the internal interrupt source that trapped to location 0003h. Writing any value to this register removes the vector and updates the next vector value (if another interrupt is pending). Note that the vector value is offset. Therefore, its value is in increments of two (bit 0 is set to 0). When no interrupt is pending, the vector is set to 00h. Table 2–8 is a table of the vector interrupt values. As shown, the interrupt vector is divided into two fields; I[2:0] and G[3:0]. The I-field defines the interrupt source within a group (on a first-come, first-served basis) and the G-field, which defines the group number. Group G0 is the lowest and G15 is the highest priority.

7	6	5	4	3	2	1	0
G3	G2	G1	G0	I2	I1	I0	RSV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/O

BIT	NAME	RESET	FUNCTION
3–1	I[2:0]	000	This field defines the interrupt source in a given group. See Table 2–8: Vector Interrupt Values. Bit-0 is always = 0, therefore, vector values are offset by two.
7–4	G[3:0]	0000	This field defines the interrupt group. I[2:0] and G[3:0] combine to produce the actual interrupt vector.

Table 2–8. Vector Interrupt Values

G[3:0] (Hex)	I[2:0] (Hex)	VECTOR (Hex)	INTERRUPT SOURCE
0	0	00	No interrupt
1	0	10	RESERVED
1	1	12	Output endpoint 1
1	2	14	Output endpoint 2
1	3	16	Output endpoint 3
1	4–7	18–1E	RESERVED
2	0	20	RESERVED
2	1	22	Input endpoint 1
2	2	24	Input endpoint 2
2	3	26	Input endpoint 3
2	4–7	28–2E	RESERVED
3	0	30	STPOW packet received
3	1	32	SETUP packet received
3	2	34	PWON interrupt
3	3	36	PWOFF interrupt
3	4	38	RESR interrupt
3	5	3A	SUSR interrupt
3	6	3C	RSTR interrupt
3	7	3E	RESERVED
4	0	40	I^{2}C TXE interrupt
4	1	42	I^{2}C RXF interrupt
4	2	44	Input endpoint 0
4	3	46	Output endpoint 0
4	4–7	48 → 4E	RESERVED
5–F	X	90 → FE	RESERVED

2.9.4 Logical Interrupt Connection Diagram ($\overline{\text{INT0}}$)

Figure 2–5 represents the logical connection of the interrupt sources and the relation of the logical connection with $\overline{\text{INT0}}$. The priority encoder generates an 8-bit vector, corresponding to 64 interrupt sources (not all are used). The interrupt priorities are hard wired. Vector 46h is the highest and 12h is the lowest. Table 2–8 lists the interrupt source for each valid interrupt vector.

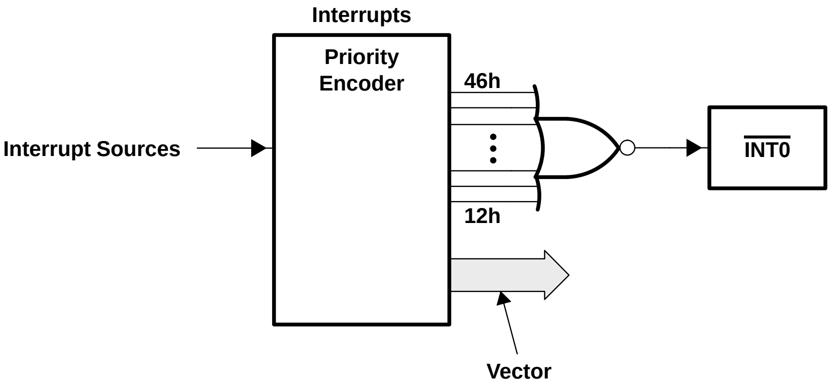


Figure 2–4. Internal Vector Interrupt ($\overline{\text{INT0}}$)

2.9.5 P2[7:0] Interrupt ($\overline{\text{INT1}}$)

Figure 2–6 illustrates the conceptual port-2 interrupt. All port-2 input signals are connected in a logical OR to generate the $\overline{\text{INT1}}$ interrupt. Note that the inputs are active low and $\overline{\text{INT1}}$ is programmed as an level-triggered interrupt. In addition, $\overline{\text{INT1}}$ is connected to the suspend/resume logic for remote wake-up support. As illustrated, the XINT bit in the MCU configuration register (MCNFG) is used to select the EX1 interrupt source. When XINT = 0, P3.3 is the source, and when XINT = 1, P2[7:0] is the source.

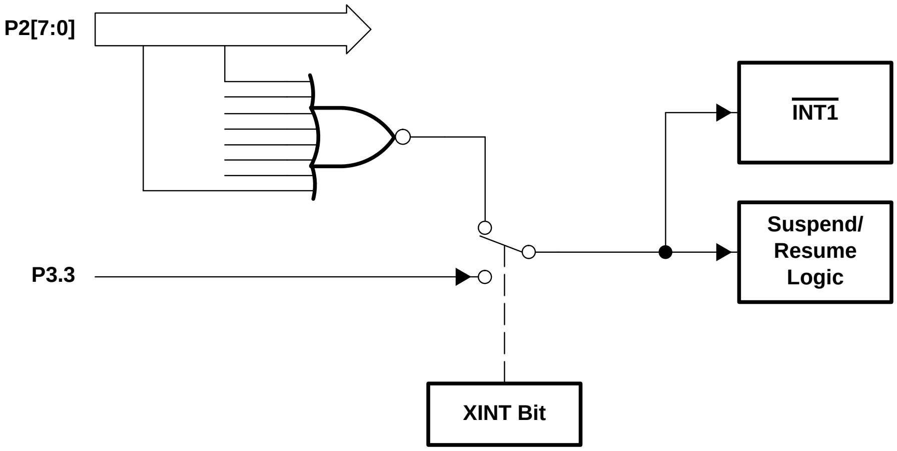


Figure 2–5. P2[7:0] Input Port Interrupt Generation

2.10 I^{2}C Registers

The TUSB3210 only supports a master-slave relationship; therefore, it does not support bus arbitration.

2.10.1 I2CSTA: I^{2}C Status and Control Register

This register is used to control the stop condition for read and write operations. In addition, it provides transmitter and receiver handshake signals with their respective interrupt enable bits.

7	6	5	4	3	2	1	0
RXF	RIE	ERR	1/4	TXE	TIE	SRD	SWR
R/C	R/W	R/C	R/W	R/C	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION	
0	SWR	0	Stop write condition. This bit defines whether the I^{2}C controller generates a stop condition when data from the I2CDAO register is transmitted to an external device.	
			SWR = 0	Stop condition is not generated when data from I2CDAO register is shifted out to an external device.
			SWR = 1	Stop condition is generated when data from I2CDAO register is shifted out to an external device.
1	SRD	0	Stop read condition. This bit defines whether the I^{2}C controller generates a stop condition when data is received and loaded into I2CDAI register.	
			SRD = 0	Stop condition is not generated when data from SDA line is shifted into the I2CDAI register.
			SRD = 1	Stop condition is generated when data from SDA line is shifted into the I2CDAI register.
2	TIE	0	I^{2}C transmitter empty interrupt enable.	
			TIE = 0	Interrupt disabled
			TIE = 1	Interrupt enabled
3	TXE	1	I^{2}C transmitter empty. This bit indicates that data can be written to the transmitter. It can be used for polling or it can generate an interrupt.	
			TXE = 0	Transmitter is full. This bit is cleared when the MCU writes a byte to the I2CDAO register.
			TXE = 1	Transmitter is empty. The I^{2}C controller sets this bit when the content of the I2CDAO register is copied to the SDA shift register.
4	1/4	0	Bus speed selection	
			¼ = 0	100 kHz bus speed
			¼ = 1	400 kHz bus speed
5	ERR	0	Bus error condition. This bit is set by the hardware when the device does not respond. It is cleared by the MCU.	
			ERR = 0	No bus error
			ERR = 1	Bus error condition has been detected. Clears when the MCU writes a 1. Writing a 0 has no effect.
6	RIE	0	I^{2}C receiver ready interrupt enable.	
6	RIE	0	RIE = 0	Interrupt disable
6	RIE	0	RIE = 1	Interrupt enable
7	RXF	0	I^{2}C receiver full. This bit indicates that the receiver contains new data. It can be used for polling or it can generate an interrupt.	
			RXF = 0	Receiver is empty. This bit is cleared when MCU reads the I2CDAI register.
			RXF = 1	Receiver contains new data. This bit is set by the I^{2}C controller when the received serial data has been loaded into the I2CDAI register.

2.10.2 I2CADR: I^{2}C Address Register

This register holds the device address and the read/write command bit.

7	6	5	4	3	2	1	0
A_6	A_5	A_4	A_3	A_2	A_1	A_0	R/W
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION	
0	R/W	0	Read/write command bit.	
			R/W = 0	Write operation
			R/W = 1	Read operation
7–1	A[6:0]	0000000	Seven address bits for device addressing	

2.10.3 I2CDAI: I^{2}C Data-Input Register

This register holds the received data from an external device.

7	6	5	4	3	2	1	0
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

BIT	NAME	RESET	FUNCTION
7–0	D[7:0]	0	8-bit input data from an I^{2}C device

2.10.4 I2CDAO: I^{2}C Data-Output Register

This register holds the data to be transmitted to an external device. Writing to this register starts the transfer on the SDA line.

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BIT	NAME	RESET	FUNCTION
7–0	D[7:0]	0	8-bit output data to an I^{2}C device

2.11 Read/Write Operations

2.11.1 Read Operation (Serial EEPROM)

A serial read requires a *dummy* byte write sequence to load in the 16-bit data word address. Once the device address word and data address word are clocked out and acknowledged by the device, the MCU starts a current address sequence. The following describes the sequence of events to accomplish this transaction:

Device Address + EEPROM [High Byte]

- The MCU sets I2CSTA[SRD] = 0. This prevents the I^{2}C controller from generating a stop condition after the content of the I2CDAI register is received.
- The MCU sets I2CSTA[SWR] = 0. This prevents the I^{2}C controller from generating a stop condition after the content of the I2CDAO register is transmitted.
- The MCU writes the device address (R/W bit = 0) to the I2CADR register (write operation).
- The MCU writes the high-byte of the EEPROM address into the I2CDAO register, starting the transfer on the SDA line.

- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CADR register is transmitted to the EEPROM (preceded by start condition on SDA).
- The content of the I2CDAO register is transmitted to the EEPROM (EEPROM address).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been transmitted.
- No stop condition is generated.

EEPROM [Low Byte]

- The MCU writes the low-byte of the EEPROM address into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CDAO register is transmitted to the device (EEPROM address).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been transmitted.
- This completes the *dummy* write operation. At this point, the EEPROM address is set and the MCU can do a single or a sequential read operation.

2.11.2 Current Address Read Operation

Once the EEPROM address is set, the MCU can read a single byte by executing the following steps:

1. The MCU sets I2CSTA[SRD] = 1, forcing the I^{2}C controller to generate a stop condition after the I2CDAI register is received.
2. The MCU writes the device address (R/W bit = 1) to the I2CADR register (read operation).
3. The MCU writes a dummy byte to the I2CDAO register, starting the transfer on the SDA line.
4. The RXF bit in I2CSTA is cleared.
5. The content of the I2CADR register is transmitted to the device, preceded by a start condition on SDA.
6. Data from the EEPROM is latched into the I2CDAI register (stop condition is transmitted).
7. The RXF bit in I2CSTA is set, and interrupts the MCU, indicating that the data is available.
8. The MCU reads the I2CDAI register. This clears the RXF bit (I2CSTA[RXF] = 0).

2.11.3 Sequential Read Operation

Once the EEPROM address is set, the MCU can execute a sequential read operation by executing the following steps (Note: this example illustrates a 32-byte sequential read):

1. Device Address
    - The MCU sets I2CSTA[SRD] = 0. This prevents the I^{2}C controller from generating a stop condition after the I2CDAI register is received.
    - The MCU writes the device address (R/W bit = 1) to the I2CADR register (read operation).
    - The MCU writes a dummy byte to the I2CDAO register, starting the transfer on the SDA line.
    - The RXF bit in I2CSTA is cleared.
    - The content of the I2CADR register is transmitted to the device (preceded by a start condition on SDA).

2. N-Byte Read (31 bytes)
   - Data from the device is latched into the I2CDAI register (stop condition is not transmitted).
   - The RXF bit in I2CSTA is set, and interrupts the MCU, indicating that data is available.
   - The MCU reads the I2CDAI register, clearing the RXF bit (I2CSTA[RXF] = 0).
   - This operation repeats 31 times.
3. Last-Byte Read (byte no. 32)
   - The MCU sets I2CSTA[SRD] = 1. This forces the I^2C controller to generate a stop condition after the I2CDAI register is received.
   - Data from the device is latched into the I2CDAI register (stop condition is transmitted).
   - The RXF bit in I2CSTA is set, and interrupts the MCU, indicating that data is available.
   - The MCU reads the I2CDAI register, clearing the RXF bit (I2CSTA[RXF] = 0).

2.11.4 Write Operation (Serial EEPROM)

The byte write operation involves three phases: 1) device address + EEPROM [high byte] phase, 2) EEPROM [low byte] phase, and 3) EEPROM [DATA]. The following describes the sequence of events to accomplish the byte write transaction:

Device Address + EEPROM [High byte]

- The MCU sets I2CSTA[SWR] = 0. This prevents the I^2C controller from generating a stop condition after the content of the I2CDAO register is transmitted.
- The MCU writes the device address (R/W bit = 0) to the I2CADR register (write operation).
- The MCU writes the high-byte of the EEPROM address into the I2CDAO register, starting the transfer on the SDA line.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CADR register is transmitted to the device (preceded by a start condition on SDA).
- The content of the I2CDAO register is transmitted to the device (EEPROM high-address).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been transmitted.

EEPROM [Low byte]

- The MCU writes the low byte of the EEPROM address into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CDAO register is transmitted to the device (EEPROM address).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been transmitted.

EEPROM [DATA]

- The MCU sets I2CSTA[SWR] = 1. This forces the I^{2}C controller to generate a stop condition after the content of the I2CDAO register is transmitted.
- The MCU writes the DATA to be written to the EEPROM into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CDAO register is transmitted to the device (EEPROM data).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been transmitted.
- The I^{2}C controller generates a stop condition after the content of the I2CDAO register is transmitted.

2.11.5 Page Write Operation

The page write operation is initiated the same way as byte write, with the exception that stop a condition is not generated after the first EEPROM [DATA] is transmitted. The following describes the sequence of writing 32 bytes in page mode:

Device Address + EEPROM [High byte]

- The MCU sets I2CSTA[SWR] = 0. This prevents the I^{2}C controller from generating a stop condition after the content of the I2CDAO register is transmitted.
- The MCU writes the device address (R/W bit = 0) to the I2CADR register (write operation).
- The MCU writes the high byte of the EEPROM address into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CADR register is transmitted to the device (preceded by a start condition on SDA).
- The content of the I2CDAO register is transmitted to the device (EEPROM address).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been sent.

EEPROM [Low byte]

- The MCU writes the low byte of the EEPROM address into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CDAO register is transmitted to the device (EEPROM address).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been sent.

31 Bytes EEPROM [DATA]

- The MCU writes the DATA to be written to the EEPROM into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CDAO register is transmitted to the device (EEPROM data).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been sent.
- This operation repeats 31 times.

Last Byte EEPROM [DATA]

- The MCU sets I2CSTA[SWR] = 1. This forces the I^{2}C controller to generate a stop condition after the content of the I2CDAO register is transmitted.
- The MCU writes the last DATA byte to be written to the EEPROM into the I2CDAO register.
- The TXE bit in I2CSTA is cleared, indicating busy.
- The content of the I2CDAO register is transmitted to the EEPROM (EEPROM data).
- The TXE bit in I2CSTA is set, and interrupts the MCU, indicating that the I2CDAO register has been sent.
- The I^{2}C controller generates a stop condition after the content of the I2CDAO register is transmitted, terminating the 32-byte page write operation.

3 Electrical Specifications

3.1 Absolute Maximum Ratings Over Operating Free-Air Temperature (unless otherwise noted)[†]

Supply voltage, V_{CC}	–0.5 V to 4 V
Input voltage, V_I	–0.5 V to V_{CC} + 0.5 V
Output voltage, V_O	–0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK}	±20 mA
Output clamp current, I_{OK}	±20 mA
Storage temperature	–65°C to 150°C

[†] Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

3.2 Commercial Operating Condition

PARAMETER		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	3	3.3	3.6	V
V_I	Input voltage	0		V_{CC}	V
V_{IH}	High level input voltage	2		V_{CC}	V
V_{IL}	Low level input voltage	0		0.8	V
T_A	Operating temperature	0		70	°C

3.3 Electrical Characteristics, T_A = 25°C, V_{CC} = 3.3 V ± 0.3V, GND = 0 V

PARAMETER		TEST CONDITIONS	MIN	NOM	MAX	UNIT
V_{OH}	High-level output voltage	I_{OH} = –4 mA	V_{CC}–0.5			V
V_{OL}	Low-level output voltage	I_{OL} = 4 mA			0.5	V
V_{IT+}	Positive input threshold voltage	$V_I = V_{IH}$			2	V
$V_{IT–}$	Negative input threshold voltage	$V_I = V_{IL}$	0.8			V
V_{hys}	Hysteresis ($V_{IT+} – V_{IT–}$)	$V_I = V_{IH}$		1		V
I_{IH}	High-level input current	$V_I = V_{IH}$			±1	µA
I_{IL}	Low-level input current	$V_I = V_{IL}$			±1	µA
I_{OZ}	Output leakage current (Hi-Z)	$V_I = V_{CC}$ or V_{SS}			10	µA
C_I	Input capacitance			5		pF
C_O	Output capacitance			7		pF
I_{CC}	Quiescent			25	45	mA
I_{CCx}	Suspend				45	µA
$I_{CCx1.8}$	Suspend 1.8 VDD				1	µA

4 Application

4.1 Examples

Figure 4–1 illustrates the port-3 pins that are assigned to drive the four example LEDs. For the connection example shown, P3[5:2] can sink up to 8 mA (open-drain output). Figure 4–2 illustrates the partial connection bus power mode. Figure 4–3 shows the USB upstream connection, and Figure 4–4 illustrates the downstream connection (only one port shown).

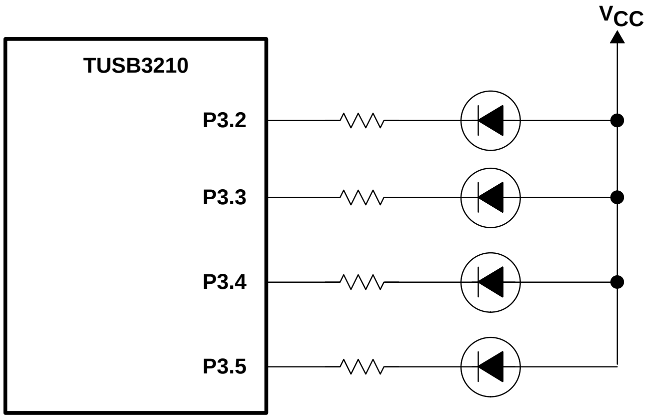


Figure 4–1. Example LED Connection

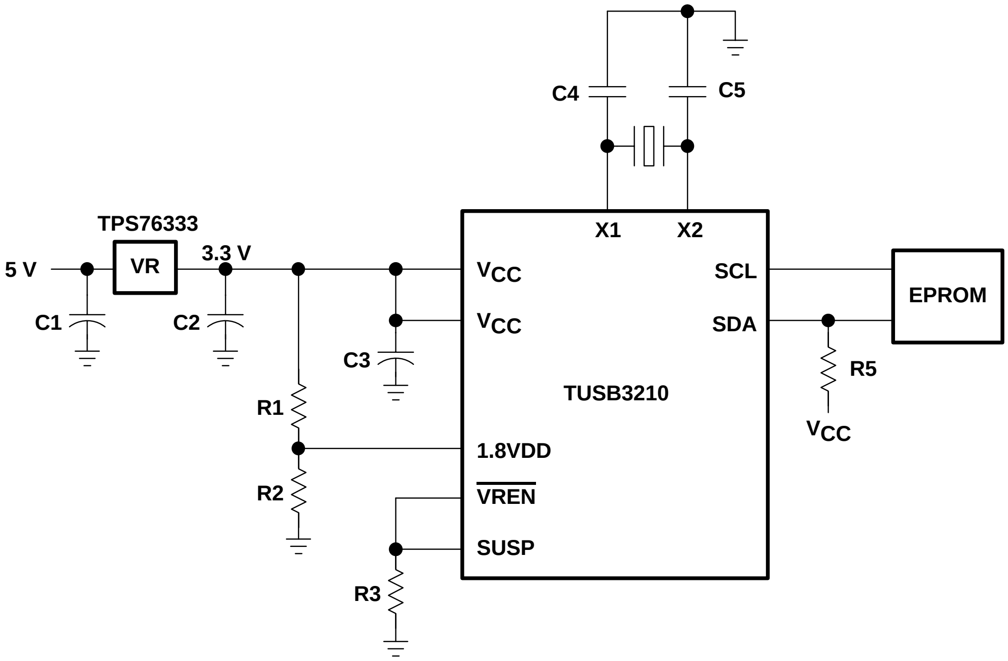


Figure 4–2. Partial Connection Bus Power Mode

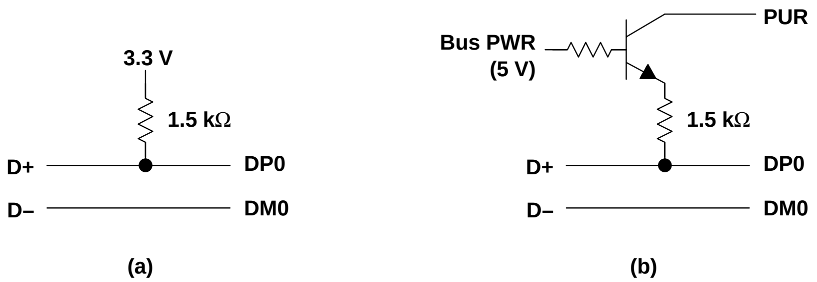
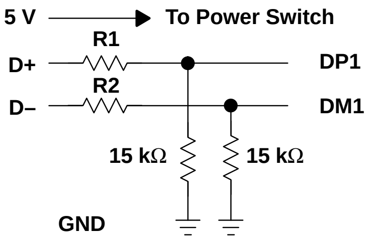


Figure 4–3. Upstream Connection (a) Non-Switching Power Mode (b) Switching Power Mode



NOTE: Ferrite beads can be used on power lines to help ESD.

Figure 4–4. Downstream Connection (Only One Port Shown)

4.2 Reset Timing

There are two requirements for the reset signal timing. First, the reset window should be between 100 µs and 10 ms. At power up, this time is measured from the time the power ramps up to 90% of the nominal V_{CC} until the reset signal goes high (above 1.2 V). The second requirement is that the clock has to be valid during the last 60 µsec of the reset window. These two requirements are depicted in Figure 4–5. Notice that when using a 12-MHz crystal or the 48-MHz oscillator, the clock signal may take several milliseconds to ramp up and become valid after power up. Therefore, the reset window may need to be elongated up to 10 ms to ensure that there is a 60-µs overlap with a valid clock.

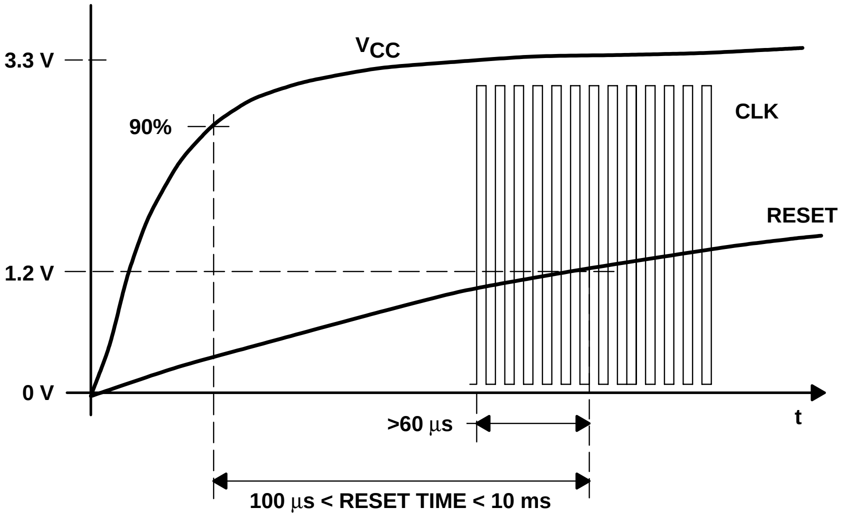
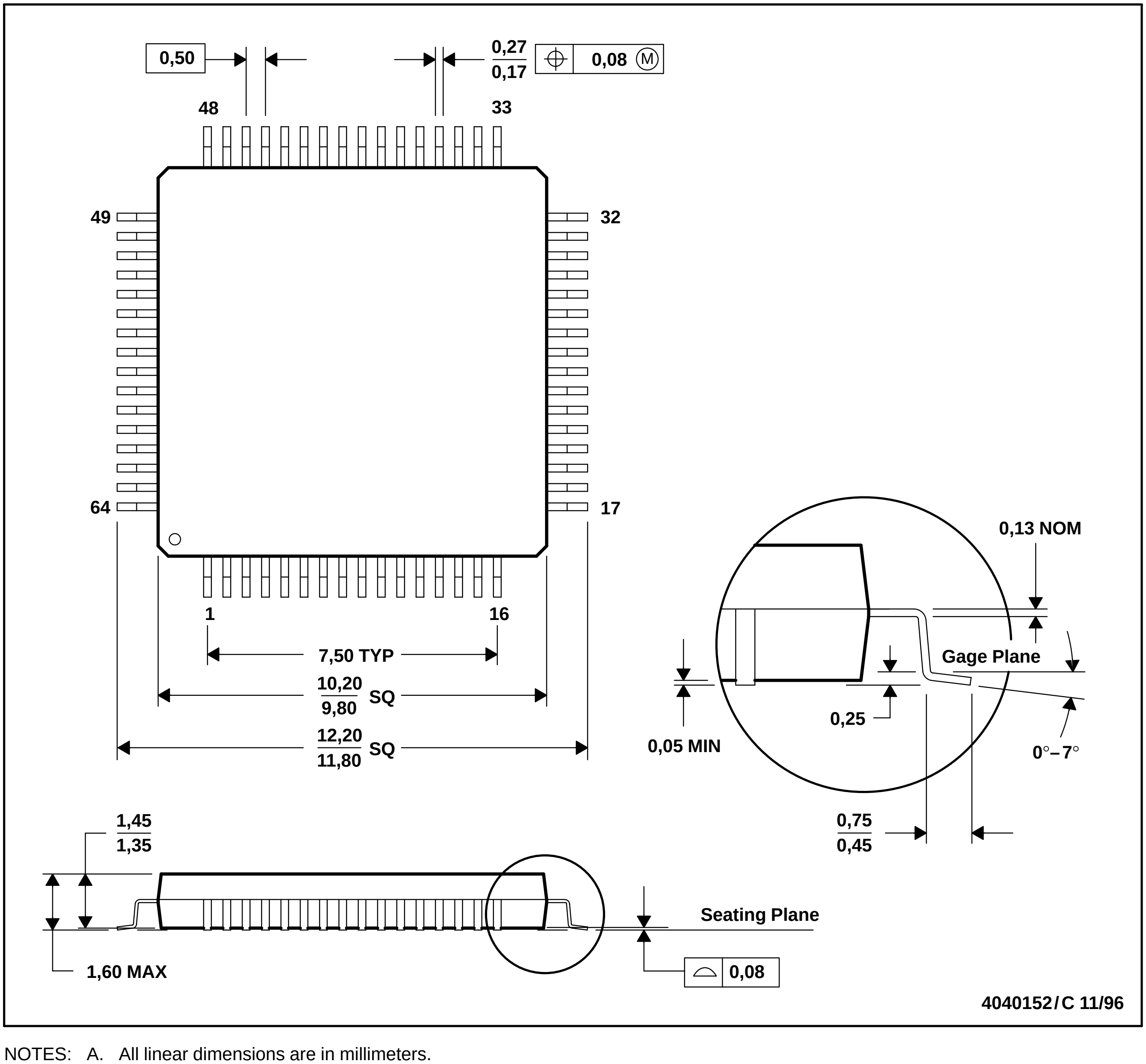


Figure 4–5. Reset Timing

5 Mechanical Data

PM (S-PQFP-G64) PLASTIC QUAD FLATPACK



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026
D. May also be thermally enhanced plastic with leads connected to the die pads.