microelectronics group

Lucent Technologies
Bell Labs Innovations

UNIVERSAL SERIAL BUS

USS-820B/USS-825B USB Device Controller

This advisory contains clarifications/changes to the Lucent Technologies Microelectronics Group *USS-820/ USS-825 USB Device Controller* Data Sheet (DS99-069CMPR) and applies only to Lucent's 44-pin MQFP USS-820B device and Lucent's 48-pin TQFP USS-825B device. All these exceptions will be corrected in the next version.

Specification Clarifications/Changes

1. Transceiver crossover voltage too low.

Correct Behavior: A USB host/hub/device is required to have a minimum value of 1.3 V as the crossover voltage of its DPLS and DMNS outputs.

Current Behavior: Under certain conditions of voltage and temperature, the USS-820B/USS-825B may exhibit a crossover voltage of slightly less than 1.3 V when transmitting. Although this has not caused a problem in most tested systems, certain hosts/ hubs have been shown to be more sensitive to this condition.

Indication: The host/hub to which the USS-820B/USS-825B is connected detects a false SE0 when the USS-820B/USS-825B is transmitting and discards the USS-820B/USS-825B 's packet(s).

Temporary Application Fix: Guarantee a minimum device voltage of 3.275 V.

2. Data misinterpreted as Token packet.

Correct Behavior: A device should only react to packets that are addressed to it.

Current Behavior: When other devices are present on the bus, the USS-820B/USS-825B can misinterpret data sent by the host to the other device(s) as a token packet addressed to itself. The failure occurs when the USS-820B/USS-825B misinterprets the 16-bit CRC of a data packet sent to another device as a valid IN token packet (PID, address, endpoint, CRC5) addressed to itself. The result is that the USS-820B/USS-825B responds to the packet with either a NAK or with a data packet, depending on whether or not it had an available data set for that endpoint.

Indication: Depending on the timing of the events and the host/hubs involved, the result is either: 1) the transfer with the other device is corrupted or 2) the host/hub to which the USS-820B/USS-825B is connected determines that the USS-820B/ USS-825B is transmitting when it should not be and disables the port, requiring a user unplug/plug. The risk of this bug manifesting itself is proportional to the volume and variety of traffic to the other device(s) on the bus, and the number of transmit endpoints that are enabled on the USS-820B/USS-825B.

Temporary Application Fix: While operating the USB peripheral containing the USS-820B/USS-825B, users may have to temporarily detach other USB devices connected to the USB bus.

Specification Clarifications/Changes (continued)

3. Multihub skew of SE0.

Correct Behavior: The USB specification allows each hub to add up to 15 ns of skew to the end of a packet before the SE0 field begins. The specification allows up to five hubs to be connected between the host and a device. The specification also allows up to 20 ns of clock jitter for a full-speed device. The total of these factors can cause a device to detect an extra data bit at the end of a packet before the SE0. A device is required to tolerate this condition when interpreting a bit stream from its USB port.

Current Behavior: The USS-820B/USS-825B cannot tolerate this extra bit. When the total of these factors approaches 1/2 bit time (approximately 40 ns.), the device may detect an extra bit. When this occurs, the USS-820B/USS-825B will discard the packet.

Indication: The USS-820B/USS-825B does not respond to packets that meet the USB specification. Once this occurs three times, the host will stop communicating with the device.

Temporary Application Fix: 1) Use only hubs that introduce a minimal amount of skew (no more than 5 ns for each hub) or 2) do not include multiple hubs between the USS-820B/USS-825B and the host. This failure should never occur if the USS-820B/USS-825B is directly connected to the USB host or is connected through a single hub that meets the USB specification for SE0 skew.

4. Device does not resume after suspend.

Correct Behavior: The USB specification requires a device to wake up within 10 ms after the end of reset/resume signalling following a suspend.

Current Behavior: Some applications of the USS-820B/USS-825B that use an external crystal (as opposed to an external oscillator) will fail to restart internal clocks within the allotted time period after coming out of a suspend. As a result the device will fail to respond to the host, which will cause the host/hub to disable the port to which the device is connected.

Temporary Application Fix: A 100 kΩ resistor should be connected between the XTAL1 and XTAL2 pins of the USS-820B (pins 2 and 3 of the 44-pin package [USS820B] or pins 3 and 4 of the 48-pin package [USS825B]).

For additional information, contact your Microelectronics Group Account Manager or the following:
INTERNET: **http://www.lucent.com/micro**
E-MAIL: **docmaster@micro.lucent.com**
N. AMERICA: Microelectronics Group, Lucent Technologies Inc., 555 Union Boulevard, Room 30L-15P-BA, Allentown, PA 18103
1-800-372-2447, FAX 610-712-4106 (In CANADA: **1-800-553-2448**, FAX 610-712-4106)
ASIA PACIFIC: Microelectronics Group, Lucent Technologies Singapore Pte. Ltd., 77 Science Park Drive, #03-18 Cintech III, Singapore 118256
Tel. (65) 778 8833, FAX (65) 777 7495
CHINA: Microelectronics Group, Lucent Technologies (China) Co., Ltd., A-F2, 23/F, Zao Fong Universe Building, 1800 Zhong Shan Xi Road, Shanghai 200233 P. R. China **Tel. (86) 21 6440 0468**, **ext. 316**, FAX (86) 21 6440 0652
JAPAN: Microelectronics Group, Lucent Technologies Japan Ltd., 7-18, Higashi-Gotanda 2-chome, Shinagawa-ku, Tokyo 141, Japan
Tel. (81) 3 5421 1600, FAX (81) 3 5421 1700
EUROPE: Data Requests: MICROELECTRONICS GROUP DATALINE: **Tel. (44) 1189 324 299**, FAX (44) 1189 328 148
Technical Inquiries: GERMANY: **(49) 89 95086 0** (Munich), UNITED KINGDOM: **(44) 1344 865 900** (Ascot),
FRANCE: **(33) 1 40 83 68 00** (Paris), SWEDEN: **(46) 8 594 607 00** (Stockholm), FINLAND: **(358) 9 4354 2800** (Helsinki),
ITALY: **(39) 02 6608131** (Milan), SPAIN: **(34) 1 807 1441** (Madrid)

microelectronics group

Lucent Technologies
Bell Labs Innovations

Preliminary Data Sheet, Rev. 3
April 1999

microelectronics group

Lucent Technologies
Bell Labs Innovations

UNIVERSAL SERIAL BUS

USS-820/USS-825 USB Device Controller

Features

- Full compliance with the *Universal Serial Bus Specification Revision 1.1*.
- Self-powered or bus-powered USB device. Meets USB power specifications for bus-powered devices.
- Full-speed USB device (12 Mbits/s).
- Supports control, interrupt, bulk, and isochronous endpoints.
- USB device controller with protocol control and administration for up to 16 USB endpoints.
- Programmable endpoint types and FIFO sizes and internal 1120-byte logical (2240-byte physical for dual-packet mode) shared FIFO storage allow a wide variety of configurations.
- Dual-packet mode of FIFOs reduces latency.
- Supports USB remote wake-up feature.
- On-chip crystal oscillator allows external 12 MHz crystal or 3 V/5 V clock source.
- On-chip analog PLL creates 48 MHz clock from internal 12 MHz clock.
- Integrated USB transceivers.
- 5 V tolerant I/O buffers allow operation in 3 V or 5 V system environment.
- Implemented in Lucent Technologies Microelectronics Group's 0.35 µm, 3 V standard-cell library.
- 44-pin MQFP (USS-820).
- 48-pin TQFP (USS-825).
- Evaluation kit available.

Applications

- Suitable for peripherals with embedded microprocessors.
- Glueless interface to microprocessor buses.
- Support of multifunction USB implementations, such as printer/scanner and integrated multimedia applications.
- Suitable for a broad range of device class peripherals in the USB standard.

Description

USS-820/USS-825 is a USB device controller that provides a programmable bridge between the USB and a local microprocessor bus. The USS-820 and USS-825 are functionally equivalent. The only difference between the two devices is their package type. The USS-820 is packaged in a 44-pin MQFP, while the USS-825 is packaged in a 48-pin TQFP. The USS-820/USS-825 allows PC peripherals to upgrade to USB connectivity without major redesign effort. It is programmable through a simple read/write register interface that is compatible with industry-standard USB microcontrollers. USS-820/USS-825 is designed in 100% compliance with the USB industry standard, allowing device-side USB products to be reliably installed using low-cost, off-the-shelf cables and connectors.

The integrated USB transceiver supports 12 Mbits/s full-speed operation. FIFO options support all four transfer types: control, interrupt, bulk, and isochronous, as described in *Universal Serial Bus Specification Revision 1.1,* with a wide range of packet sizes. Its double sets of FIFO enable the dual-packet mode feature. The dual-packet mode feature reduces latency by allowing simultaneous transfers on the host and microprocessor sides of a given unidirectional endpoint.

Note: Advisories are issued as needed to update product information. When using this data sheet for design purposes, please contact your Lucent Technologies Microelectronics Group Account Manager to obtain the latest advisory on this product.

Table of Contents

Contents **Page**

Description (continued)

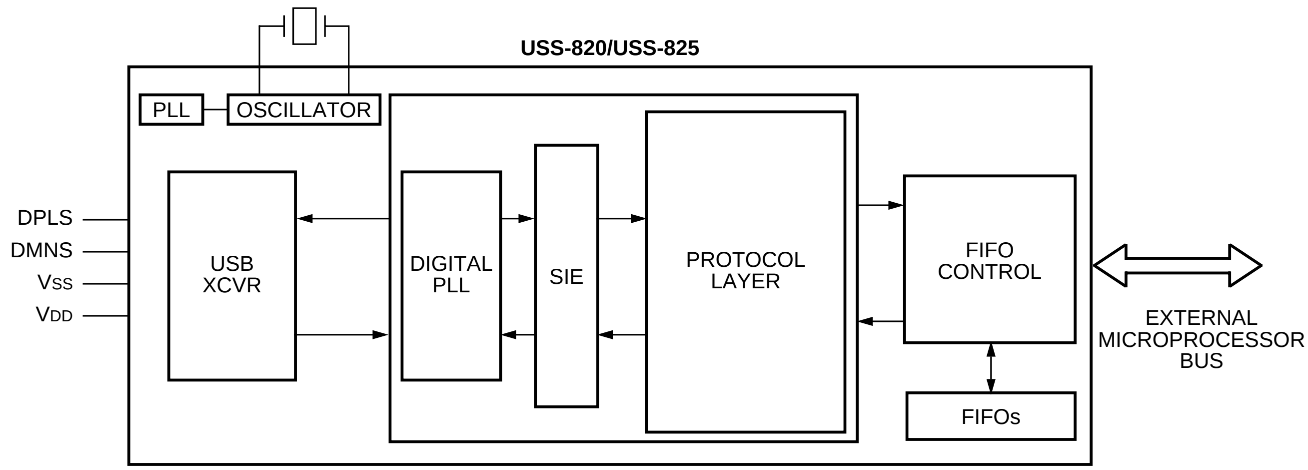


Figure 1. Block Diagram

The USS-820/USS-825 supports a maximum of eight bidirectional endpoints with 16 FIFOs (eight for transmit and eight for receive) associated with them. The FIFOs are on-chip, and sizes are programmable up to a total of 1120 logical bytes. When the dual-packet mode feature is enabled, the device uses a maximum of 2240 bytes of physical storage. This additional physical FIFO storage is managed by the device hardware and is transparent to the user.

The FIFO sizes supported are 16 bytes and 64 bytes for nonisochronous pipes and 64 bytes, 256 bytes, 512 bytes, and 1024 bytes for isochronous pipes. The FIFO size of a given endpoint defines the upper limit to maximum packet size that the hardware can support for that endpoint. This flexibility covers a wide range of data rates, data types, and combinations of applications.

The USS-820/USS-825 can be clocked either by connecting a 12 MHz crystal to the XTAL1 and XTAL2 pins, or by using a 12 MHz external oscillator. The internal 12 MHz clock period, which is a function of either of these clock sources, is referred to as the device clock period (t_{CLK}) throughout this data sheet.

Serial Interface Engine

The SIE is the USB protocol interpreter. It serves as a communicator between the USS-820/USS-825 and the host through the USB lines.

The SIE functions include:

- Package protocol sequencing
- SOP (start of packet), EOP (end of packet), RESUME, and RESET signal detection and generation
- NRZI data encoding/decoding and bit stuffing
- CRC generation and checking for token and data
- Serial-to-parallel and parallel-to-serial data conversion

Protocol Layer

The protocol layer manages the interface between the SIE and FIFO control blocks. It passes all USB OUT and SETUP packets through to the appropriate FIFO. It is the responsibility of firmware to correctly interpret and execute each USB SETUP command (as documented in the Firmware Responsibilities for USB SETUP Commands section) via the register interface. The protocol layer tracks the setup, data, and status stages of control transfers.

FIFO Control

USS-820/USS-825's FIFO control manager handles the data flow between the FIFOs and the device controller's protocol layer. It handles flow control and error handling/fault recovery to monitor transaction status and to relay control events via interrupt vectors.

Description (continued)

FIFO Programmability

Table 1 shows the programmable FIFO sizes. The size of the FIFO determines the maximum packet size that the hardware can support for a given endpoint. An endpoint is only allocated space in the shared FIFO storage if its RXEPEN/TXEPEN bit = 1. If the endpoint is disabled (RXEPEN/TXEPEN = 0), it is allocated 0 bytes. Register changes that affect the allocation of the shared FIFO storage among endpoints must not be made while there is valid data present in any of the enabled endpoints' FIFOs. Any such changes will render all FIFO contents undefined. Register bits that affect the FIFO allocation are the endpoint enable bits (the TXEPEN and RXEPEN bits of EPCON), the size bits of an enabled endpoint (FFSZ bits of TXCON and RXCON), and isochronous bits of an enabled endpoint (TXISO bit of TXCON and RXISO bit of RXCON).

Table 1. Programmable FIFO Sizes

FFSZ[1:0]	00	01	10	11
Nonisochronous	16 bytes	64 bytes	64 bytes	64 bytes
Isochronous	64 bytes	256 bytes	512 bytes	1024 bytes

Each FIFO can be programmed independently via the TXCON and RXCON registers, but the total logical size of the enabled endpoints (TX FIFOs + RX FIFOs) must not exceed 1120 bytes. The 1120-byte total allows a configuration with a full-sized, 1024-byte isochronous endpoint, a minimum-sized, 64-byte isochronous feedback endpoint, and the required, bidirectional, 16-byte control endpoint. When the dual-packet mode feature is enabled, the device uses a maximum of 2240 bytes of physical storage. This additional physical FIFO storage is managed by the device hardware and is transparent to the user.

FIFO Access

The transmit and receive FIFOs are accessed by the application through the register interface (see Tables 22—25 for transmit FIFO registers and Tables 26—29 for receive FIFO registers).

The transmit FIFO is written to via the TXDAT register, and the receive FIFO is read via the RXDAT register. The particular transmit/receive FIFO is specified by the EPINDEX register. Each FIFO is accessed serially, each RXDAT read increments the receive FIFO read pointer by 1, and each TXDAT write increments the transmit FIFO write pointer by 1.

Each FIFO consists of two data sets to provide the capability for simultaneous read/write access. Control of these pairs of data sets is managed by the hardware, invisible to the application, although the application must be aware of the implications. The receive FIFO read access is advanced to the next data set by setting the RXFFRC bit of RXCON. This bit clears itself after the advance is complete. The transmit FIFO write access is advanced to the next data set by writing the byte count to the TXCNTH/L registers.

The USB access to the receive and transmit FIFOs is managed by the hardware, although the control of the data sets can be overridden by the ARM and ATM bits of RXCON and TXCON, respectively. A successful USB transaction causes FIFO access to be advanced to the next data set. A failed USB transaction (e.g., for receive operations, FIFO overrun, data time-out, CRC error, bit stuff error; for transmit operations, FIFO underrun, no ACK from host) causes the FIFO read/write pointer to be reversed to the beginning of the data set to allow transmission retry for nonisochronous transfers.

Description (continued)

Transmit FIFO

The transmit FIFOs are circulating data buffers that have the following features:

- Support up to two separate data sets of variable sizes (dual-packet mode)
- Include byte counter register for storing the number of bytes in the data sets
- Protect against overwriting data in a full FIFO
- Can retransmit the current data set

All transmit FIFOs use the same architecture (see Figure 2). The transmit FIFO and its associated logic can manage up to two data sets, data set 0 (ds0) and data set 1 (ds1). Since two data sets can be used in the FIFO, back-to-back transmissions are supported. Dual-packet mode for transmit FIFOs is enabled by default. Single-packet mode can be enforced by firmware convention (see TXFIF register bits).

The CPU writes to the FIFO location that is specified by the write pointer. After a write, the write pointer automatically increments by 1. The read marker points to the first byte of data written to a data set, and the read pointer points to the next FIFO location to be read by the function interface. After a read, the read pointer automatically increments by 1.

When a good transmission is completed, the read marker can be advanced to the position of the read pointer to set up for reading the next data set. When a bad transmission is completed, the read pointer can be reversed to the position of the read marker to enable the function interface to reread the last data set for retransmission. The read marker advance and read pointer reversal can be achieved two ways: explicitly by firmware or automatically by hardware, as indicated by bits in the transmit FIFO control register (TXCON).

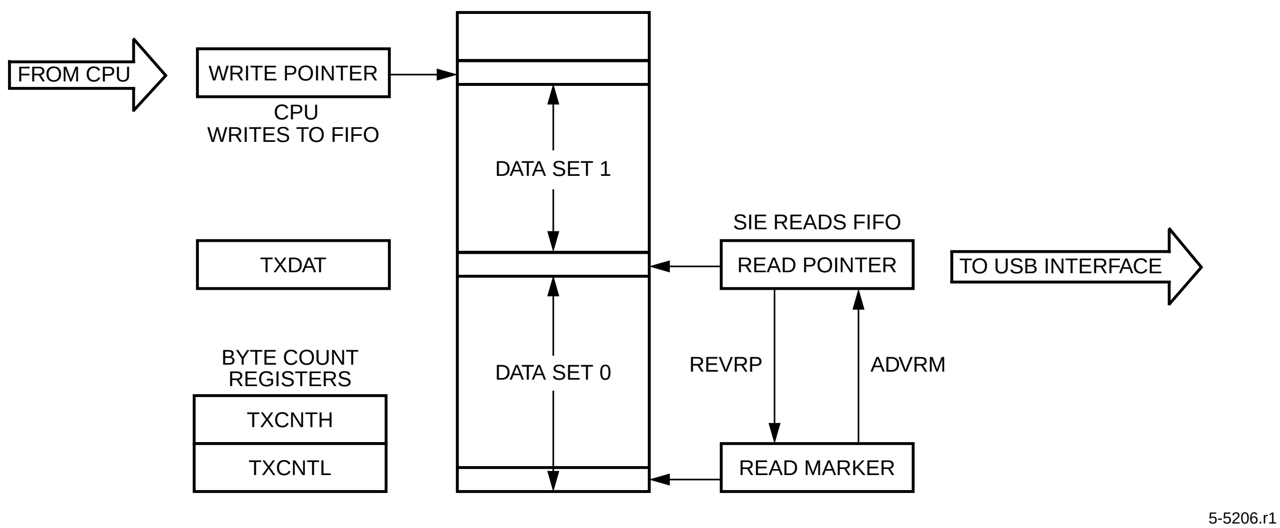


Figure 2. Transmit FIFO

Description (continued)

Receive FIFO

The receive FIFOs are circulating data buffers that have the following features:

- Support up to two separate data sets of variable sizes (dual-packet mode)
- Include byte count register that accesses the number of bytes in data sets
- Include flags to signal a full FIFO and an empty FIFO
- Can reread the last data set

Figure 3 shows a receive FIFO. A receive FIFO and its associated logic can manage up to two data sets, data set 0 (ds0) and data set 1 (ds1). Since two data sets can be used in the FIFO, back-to-back transmissions are supported. Single-packet mode is established by default after a USS-820/USS-825 device reset, which sets the RXSPM register bit. Firmware can enable dual-packet mode by clearing the RXSPM bit to 0.

The receive FIFO is symmetrical to the transmit FIFO in many ways. The SIE writes to the FIFO location specified by the write pointer. After a write, the write pointer automatically increments by 1. The write marker points to the first byte of data written to a data set, and the read pointer points to the next FIFO location to be read by the CPU. After a read, the read pointer automatically increments by 1.

When a good reception is completed, the write marker can be advanced to the position of the write pointer to set up for writing the next data set. When a bad transmission is completed, the write pointer can be reversed to the position of the write marker to enable the SIE to rewrite the last data set after receiving the data again. The write marker advance and write pointer reversal can be achieved two ways: explicitly by firmware or automatically by hardware, as specified by bits in the receive FIFO control register (RXCON).

The CPU should not read data from the receive FIFO before all bytes are received and successfully acknowledged because the reception may be bad.

To avoid overwriting data in the receive FIFO, the SIE can monitor the FIFO full flag (RXFULL bit in RXFLG). To avoid reading a byte when the FIFO is empty, the CPU can monitor the FIFO empty flag (RXEMP bit in RXFLG).

The CPU must not change the value of the EPINDEX register during the process of reading a data set from a particular receive FIFO. Once the CPU has read the first byte of a data set, the processor must ensure that the EPINDEX register setting remains unchanged until after the last byte is read from that data set. Registers other than EPINDEX may be read or written during this period, except for registers which affect the overall FIFO configuration, as described in the FIFO Programmability section. If EPINDEX is allowed to change during a data set read, incorrect data will be returned by the USS-820/USS-825 when subsequent bytes are read from the partially read data set.

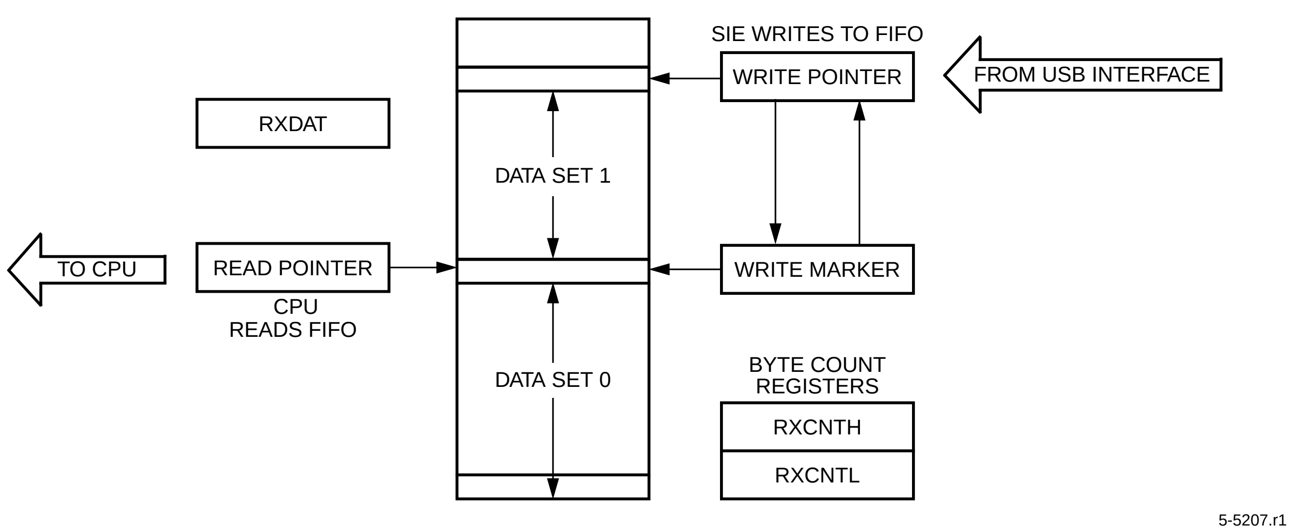
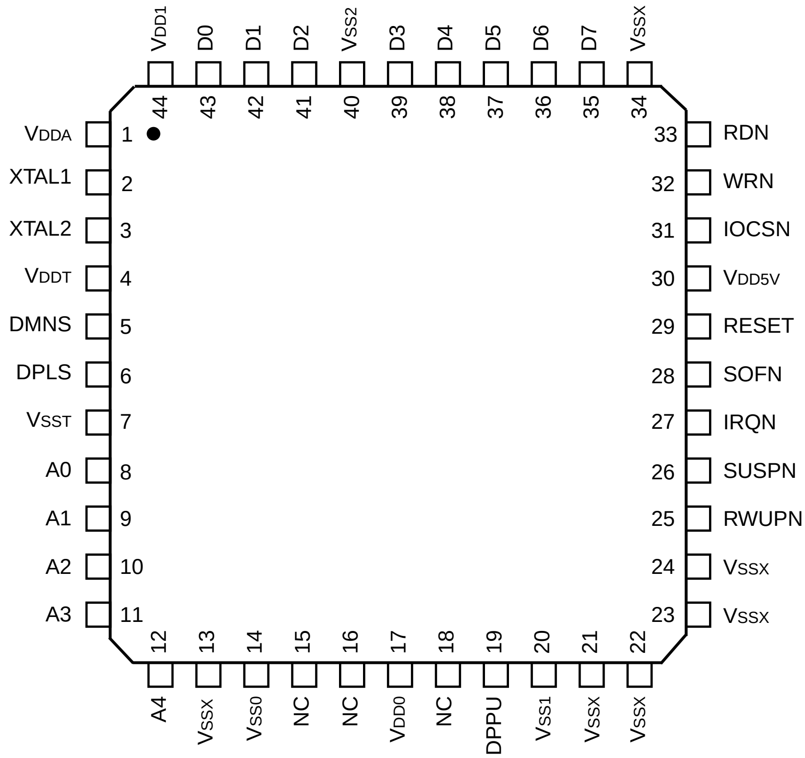


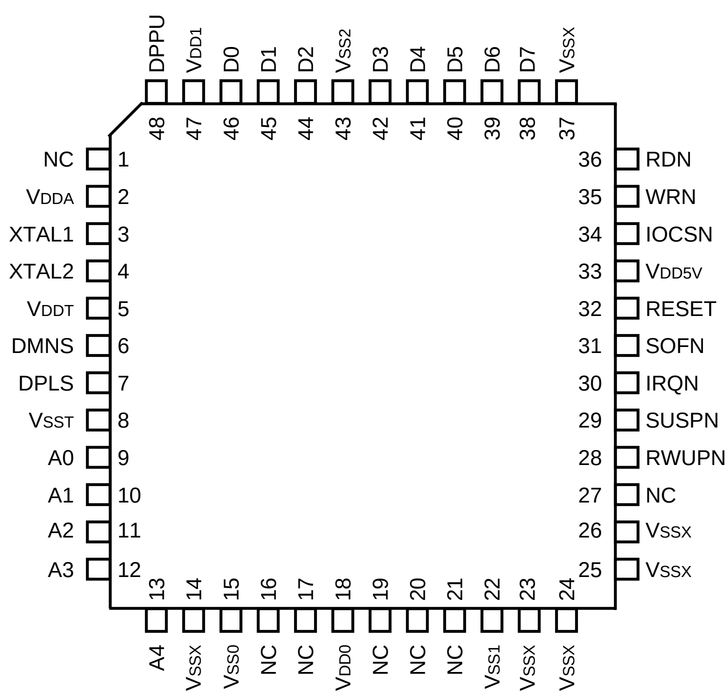
Figure 3. Receive FIFO

Pin Information



5-5428.r4

Figure 4. USS-820 Pin Diagram (44-Pin MQFP)



5-6404.r1

Figure 5. USS-825 Pin Diagram (48-Pin TQFP)

Pin Information (continued)

Table 2. Pin Descriptions

44-Pin MQFP (USS-820)	48-Pin TQFP (USS-825)	Symbol*	Type	Name/Description
1	2	VDDA	P	**3.3 V Power Supply for Analog PLL.**
2	3	XTAL1	I	**Crystal/Clock Input.** If the internal oscillator is used, this is the crystal input. If an external oscillator is used, this is the clock input.
3	4	XTAL2	O	**Crystal/Clock Output.** If the internal oscillator is used, this is the crystal output. If an external oscillator is used, this output should be left unconnected.
4	5	VDDT	P	**3.3 V Power Supply for USB Transceiver.**
5	6	DMNS	I/O	**USB Differential Data Bus Minus.**
6	7	DPLS	I/O	**USB Differential Data Bus Plus.**
7	8	VSST	P	**Device Ground for USB Transceiver.**
12, 11, 10, 9, 8	13, 12, 11, 10, 9	A[4:0]	I	**Address Bus.** This is the address bus for the controller to access the register set.
13, 14, 20, 21, 22, 23, 24, 34, 40	14, 15, 22, 23, 24, 25, 26, 37, 43	VSS0, VSS1, VSS2, VSSX	P	**Device Ground.**
15, 16, 18	1, 16, 17, 19, 20, 21, 27	NC	—	**No Connect.**
17, 44	18, 47	VDD0, VDD1	P	**3.3 V Power Supply.**
19	48	DPPU	O	**DPLS Pull-Up.** Can be used to supply power to the DPLS 1.5 kΩ pull-up resistor to allow firmware to simulate a device physical disconnect. This pin is directly controlled by the DPEN register bit.
25	28	RWUPN	I	**Remote Wake-Up (Active-Low).** Device is initiating a remote wake-up from a suspend condition. This input is ignored if SCR register bit RWUPE = 0.
26	29	SUSPN	O	**Suspend (Active-Low).** USB suspend has been detected; chip has entered suspend (low power) mode. This pin is deasserted when a wake-up event is detected.
27	30	IRQN	O	**Interrupt (Active-Low).** An interrupt signal is sent to the controller whenever an event such as TX/RX done, SUSPEND, RESET, or SOF occurs.
28	31	SOFN	O	**Start of Frame (Active-Low).** This signal is asserted low for eight tCLK periods when an SOF token is received.
29	32	RESET	I	**Reset.** When this signal is held high, all state machines and registers are set at the default state.
30	33	VDD5V	P	**5 V Power Supply for 5 V Tolerant Interface.** Connect to 3.3 V supply if a 3.3 V microprocessor interface is used.
31	34	IOCSN	I	**Chip Select (Active-Low).**
32	35	WRN	I	**Control Register Write (Active-Low).**
33	36	RDN	I	**Control Register Read (Active-Low).**
35, 36, 37, 38, 39, 41, 42, 43	38, 39, 40, 41, 42, 44, 45, 46	D[7:0]	I/O	**Data Bus.**

* Active-low signals within this document are indicated by an N following the symbol names.

Register Timing Characteristics

All register timing specifications assume a 100 pF load on the D[7:0] package pins and a 70 pF load on all other package pins.

Table 3. Timing Parameters

Symbol	Parameter	Min	Max	Unit
tCLK	Internal Clock Period	—	83.3	ns
tRST	RESET Assert Time	500	—	ns

Table 4. Register Access Timing—Special Function Register (SFR) Read

Symbol	Parameter	Min	Max	Unit
tRDASU	Read Address Setup Time (starts **before** the **trailing** edge of RDN or IOCSN, whichever is **first**)	60	—	ns
tRDAHD	Read Address Hold (starts **after** the **trailing** edge of RDN or IOCSN, whichever is **first**): Operational Suspended	 –10 3	 — —	 ns ns
tRDDV1, tRDDV2	Read Data Valid (from the **leading** edge of RDN or IOCSN or from address valid, whichever is **last**, to data valid): Operational Suspended	 — —	 74 33	 ns ns
tRDDZ	Read Data to Z State (starts **after** the **trailing** edge of RDN or IOCSN, whichever is **first**)	2	32	ns
tRDREC	Recovery Time Between Reads (from the **trailing** edge of RDN or IOCSN, whichever is **first**, to the next **leading** edge of RDN or IOCSN, whichever is **last**)	23	—	ns
tRDRECRXD	Recovery Time Between Consecutive RXDAT Reads (from the **trailing** edge of RDN or IOCSN, whichever is **first**, to the next **trailing** edge of RDN or IOCSN, whichever is **first**)	86	—	ns
tRDPW	Minimum Pulse Width (from the **leading** edge of RDN or IOCSN, whichever is **last**, to the **trailing** edge of RDN or IOCSN, which-ever is **first**)	23	—	ns

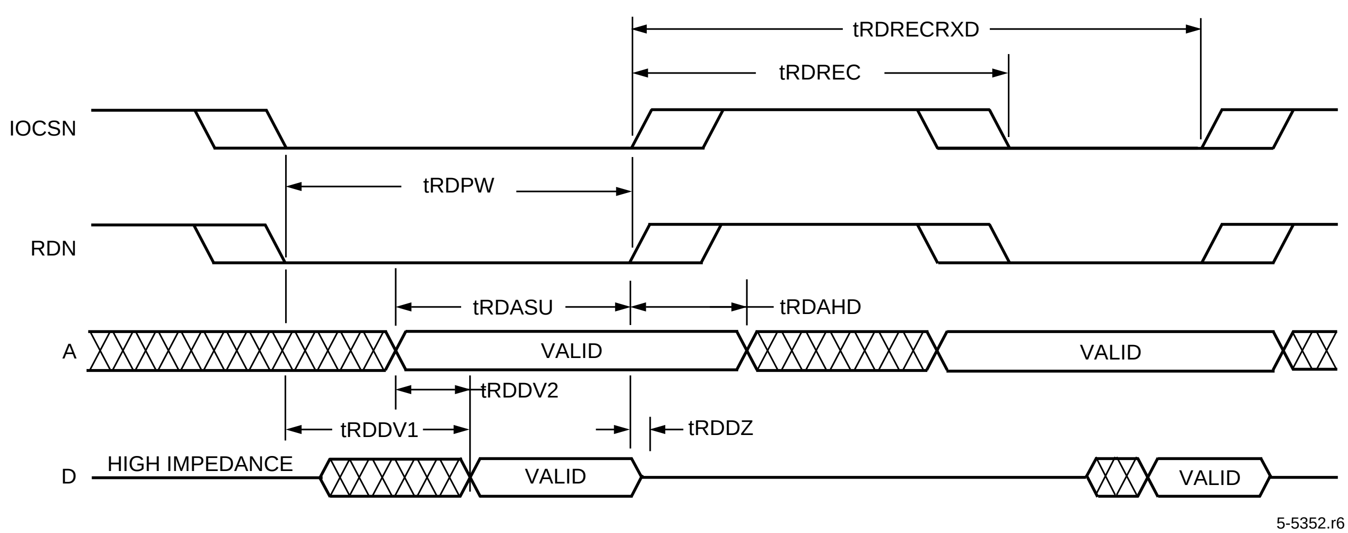


Figure 6. Register Access Timing—SFR Read

Register Timing Characteristics (continued)

Table 5. Register Access Timing—Special Function Register (SFR) Write

Symbol	Parameter	Min	Unit
tWRASU	Write Address Setup Time (starts **before** the **trailing** edge of WRN or IOCSN, whichever is **first**)	60	ns
tWRAHD	Write Address Hold (starts **after** the **trailing** edge of WRN or IOCSN, whichever is **first**)	–10	ns
tWRPW	Write Minimum Pulse Width (from the **leading** edge of WRN or IOCSN, whichever is **last**, to the **trailing** edge of WRN or IOCSN, whichever is **first**)	23	ns
tWRDSU	Write Data Setup (from data valid to the **trailing** edge of WRN or IOCSN, whichever is **first**)	60	ns
tWRDHD	Write Data Hold (from the **trailing** edge of WRN or IOCSN, whichever is **first**, to data not valid)	–10	ns
tWRREC	Recovery Time Between Write Attempts (from the **trailing** edge of WRN or IOCSN, whichever is **first**, to the next **leading** edge of WRN or IOCSN, whichever is **last**)	23	ns
tWRRECC	Recovery Time Between Write Completes (from the **trailing** edge of WRN or IOCSN, whichever is **first**, to the next **trailing** edge of WRN or IOCSN, whichever is **first**)	86	ns

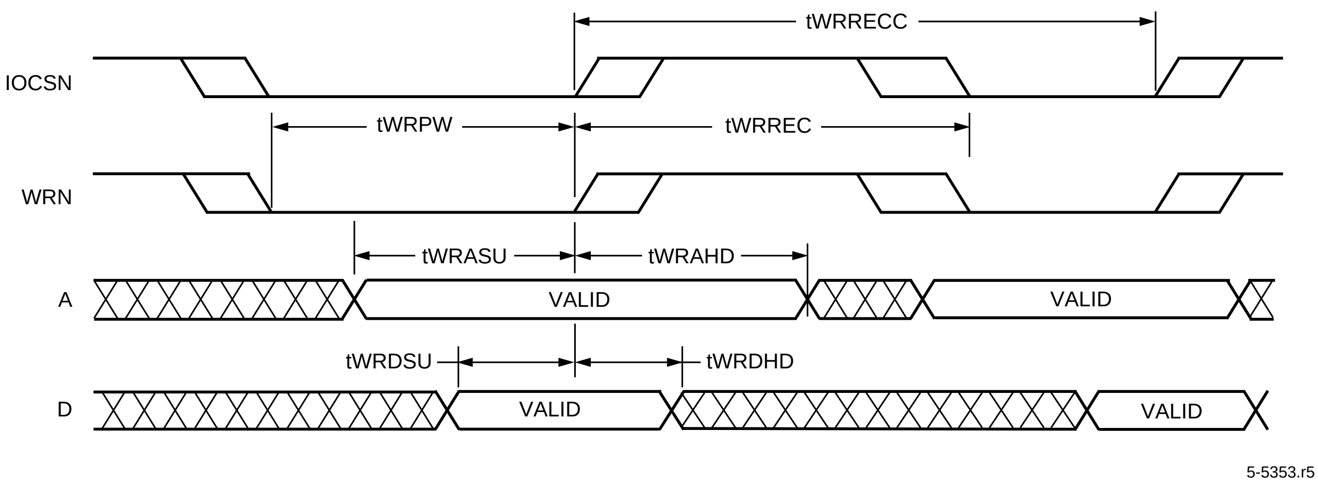


Figure 7. Register Access Timing—SFR Write

Register Interface

The USS-820/USS-825 is controlled through an asynchronous, read/write register interface. Registers are addressed via the A[4:0] pins, and control is provided through the RDN, WRN, and IOCSN pins. Reserved bits of registers should always be written with 0. Writing 1 to these bits may produce undefined results. These bits always return a 0 when read.

A register read is accomplished by placing the register address on the A bus and asserting the IOCSN and RDN pins. After read data valid (tRDDV), the register data will appear on the D bus. A register write is accomplished by placing the register address on the A bus and the data to be written on the D bus, and asserting the IOCSN and WRN pins.

Tables 6 and 7 show alphabetical and numerical listings of all the available special function registers (SFR) for the USS-820/USS-825. For reference purposes, an alphabetized list of SFR bit names is included in Appendix A. Tables 11—36 provide details for each of the registers. Some of these registers are replicated for each endpoint. The individual, endpoint-specific register is selected by the EPINDEX register.

Table 6. Special Function Registers (By Name)

Register	Description	Address	Table	Page
EPCON*	Endpoint Control Register	0BH†	18	20
EPINDEX	Endpoint Index Register	0AH	17	19
FADDR	Function Address Register	10H	21	24
LOCK	Suspend Power-Off Locking Register	19H	33	34
MCSR	Miscellaneous Control/Status Register	1CH	36	35
PEND	Pend Hardware Status Update Register	1AH	34	34
REV	Hardware Revision Register	18H	32	33
RXCNTH	Receive FIFO Byte-Count High Register	07H†	27	28
RXCNTL	Receive FIFO Byte-Count Low Register	06H†	27	28
RXCON	Receive FIFO Control Register	08H†	28	28
RXDAT	Receive FIFO Data Register	05H†	26	27
RXFLG	Receive FIFO Flag Register	09H†	29	30
RXSTAT*	Endpoint Receive Status Register	0DH†	20	22
SBI*	Serial Bus Interrupt Register	14H	13	16
SBI1*	Serial Bus Interrupt Register 1	15H	14	17
SBIE	Serial Bus Interrupt Enable Register	16H	11	15
SBIE1	Serial Bus Interrupt Enable Register 1	17H	12	15
SCR	System Control Register	11H	30	32
SCRATCH	Scratch Firmware Information Register	1BH	35	34
SOFH*	Start of Frame High Register	0FH	15	18
SOFL*	Start of Frame Low Register	0EH	16	19
SSR*	System Status Register	12H	31	33
TXCNTH	Transmit FIFO Byte-Count High Register	02H†	23	24
TXCNTL	Transmit FIFO Byte-Count Low Register	01H†	23	24
TXCON	USB Transmit FIFO Control Register	03H†	24	25
TXDAT	Transmit FIFO Data Register	00H†	22	24
TXFLG	Transmit FIFO Flag Register	04H†	25	26
TXSTAT	Endpoint Transmit Status Register	0CH†	19	21

* Contains shared bits. See Special Firmware Action for Shared Register Bits section.
† Indexed by EPINDEX.

Register Interface (continued)

Table 7. Special Function Registers (By Address)

Address	Register	Description	Table	Page
00H*	TXDAT	Transmit FIFO Data Register	22	24
01H*	TXCNTL	Transmit FIFO Byte-Count Low Register	23	24
02H*	TXCNTH	Transmit FIFO Byte-Count High Register	23	24
03H*	TXCON	USB Transmit FIFO Control Register	24	25
04H*	TXFLG	Transmit FIFO Flag Register	25	26
05H*	RXDAT	Receive FIFO Data Register	26	27
06H*	RXCNTL	Receive FIFO Byte-Count Low Register	27	28
07H*	RXCNTH	Receive FIFO Byte-Count High Register	27	28
08H*	RXCON	Receive FIFO Control Register	28	28
09H*	RXFLG	Receive FIFO Flag Register	29	30
0AH	EPINDEX	Endpoint Index Register	17	19
0BH*	EPCON†	Endpoint Control Register	18	20
0CH*	TXSTAT	Endpoint Transmit Status Register	19	21
0DH*	RXSTAT†	Endpoint Receive Status Register	20	22
0EH	SOFL†	Start of Frame Low Register	16	19
0FH	SOFH†	Start of Frame High Register	15	18
10H	FADDR	Function Address Register	21	24
11H	SCR	System Control Register	30	32
12H	SSR†	System Status Register	31	33
14H	SBI†	Serial Bus Interrupt Register	13	16
15H	SBI1†	Serial Bus Interrupt Register 1	14	17
16H	SBIE	Serial Bus Interrupt Enable Register	11	15
17H	SBIE1	Serial Bus Interrupt Enable Register 1	12	15
18H	REV	Hardware Revision Register	32	33
19H	LOCK	Suspend Power-Off Locking Register	33	34
1AH	PEND	Pend Hardware Status Update Register	34	34
1BH	SCRATCH	Scratch Firmware Information Register	35	34
1CH	MCSR	Miscellaneous Control/Status Register	36	35

* Indexed by EPINDEX.
† Contains shared bits. See Special Firmware Action for Shared Register Bits section.

Register Interface (continued)

Special Firmware Action for Shared Register Bits

Since the USS-820/USS-825 registers are not bit-addressable and contain several bits that may be written by either firmware or hardware (shared bits), special care must be taken to avoid incorrect behavior. In particular, firmware must be careful not to write a bit after hardware has updated the bit, but before firmware has recognized the hardware update of the bit.

There are two general cases where this may occur:

1. Direct collision—Firmware does a read-modify-write sequence to update a register bit, but between the firmware read and firmware write, hardware updates the bit. For example, in dual-packet mode, hardware could update an SBI bit while firmware is simultaneously resetting the same SBI bit. This would cause firmware to miss the fact that a new transfer has completed.
2. Indirect collision—Firmware does a read-modify-write sequence to update a register bit, but between the firmware read and firmware write, hardware updates a different bit in the same register. For example, firmware could do a read-modify-write to update the SOFODIS bit of the SOFH register, but at the same time, hardware could be updating the ASOF status bit. Firmware would inadvertently reset the ASOF bit without being aware of the hardware update.

These problems can be avoided through the use of the PEND register, which can only be written by firmware. Firmware must ensure that the PEND register bit is set before writing any registers that contain shared bits.

All shared register bits have two copies: a standard copy and a pended copy. The manner in which these register bits are updated varies depending on the value of the PEND register bit, as described in Table 8. The standard copy is the bit that is read and written during normal operation (PEND = 0). While PEND = 1, hardware updates only affect the pended copy, and firmware updates only affect the standard copy. When firmware resets the PEND bit, the pended copies of the shared bits are used to update the standard copies of the shared bits as described in Table 9. Through these means, hardware updates during a firmware read-modify-write sequence will not be missed.

Table 8. Shared Register Bit Update Behavior (ASOF Example)

Bit	Update Behavior While PEND = 0	Update Behavior While PEND = 1	Update Behavior When Firmware Resets PEND to 0
ASOF (standard copy)	Updated by hardware (firmware must not write this register)	Updated by firmware	Updated as documented in Table 9
ASOF (pended copy)	Not used	Updated by hardware	No longer used

Firmware must execute the following sequence when processing a shared bit (to avoid the direct collision case), or when writing a bit which resides in a register that contains shared bits (to avoid the indirect collision case):

- Set the PEND bit.
- Read the register with the shared bit [Read].
- If processing a shared bit, respond to the shared bit. For example, for an SBI bit, process any data sets present for that endpoint.
- Update the bit [Modify].
- Write the register with the shared bit with the modified data [Write].
- Reset the PEND bit.

When a data set is written to a receive FIFO, that FIFO's SBI(1) register bit will set. Firmware must process the indicated receive data set and, in doing so, manage that FIFO's SBI(1) bit according to the sequence described in this section. In dual-packet mode, it is possible that a second data set will be written to a receive FIFO before firmware has completed processing of the initial data set. This second data set could have been written either before or after firmware set the PEND bit to 1. Therefore, firmware cannot determine whether or not this second receive done indication was saved in the pended copy of the SBI(1) bit. Because of this uncertainty, firmware must process all receive data sets which are present in the indicated FIFO before resetting the PEND bit to 0. If the receive done indication of the second data set was in fact saved in the pended SBI(1) register, then the standard copy of the SBI(1) bit will be set when firmware resets the PEND bit to 0.

Register Interface (continued)

In this case, the SBI(1) bit will be set even though there is no corresponding data set present in the receive FIFO. Therefore, firmware must be prepared to service a receive done interrupt where no data sets are present in the indicated FIFO.

Table 9 shows the values loaded into each of the standard copies of the shared register bits when firmware resets the PEND register bit.

Table 9. Shared Register Update Values When Firmware Resets PEND

Register	Bit(s)	Update Value
SBI	All bits	Set to 1 if standard copy = 1 or pended copy = 1.
SBI1	All bits	Set to 1 if standard copy = 1 or pended copy = 1.
RXSTAT	RXSETUP	Loaded with pended copy if USB action updated RXSETUP while PEND was set.
RXSTAT	EDOVW	Set to 1 if standard copy = 1 or pended copy = 1.
EPCON	RXSTL	Set to 1 if standard copy = 1 or pended copy = 1.
SOFH	ASOF	Set to 1 if standard copy = 1 or pended copy = 1.
SOFH	TS	Loaded with pended copy if USB SOF was received while PEND was set.
SOFL	All bits	Loaded with pended copy if USB SOF was received while PEND was set.
SSR	RESET	Set to 1 if standard copy = 1 or pended copy = 1.

The register bits that are only updated by firmware, but reside in registers with shared bits and must therefore be updated only while PEND is set, are shown in Table 10.

Table 10. Register Bits Only Updated While PEND Is Set

Register	Bit(s)
RXSTAT	RXSEQ
EPCON	All bits except RXSTL
SOFH	SOFIE, SOFODIS
SSR	SUSPPO, SUSPDIS, RESUME, SUSPEND

Firmware should attempt to minimize the period during which PEND is set in order to minimize the distortion of the detection of hardware events.

Register Interface (continued)

Table 11. Serial Bus Interrupt Enable Register (SBIE)—Address: 16H; Default: 0000 0000B

This register enables and disables the receive and transmit done interrupts for function endpoints 0 through 3.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRXIE3	FTXIE3	FRXIE2	FTXIE2	FRXIE1	FTXIE1	FRXIE0	FTXIE0
R/W							

Bit	Symbol	Function/Description
7	FRXIE3	**Function Receive Interrupt Enable 3.** Enables receive done interrupt for endpoint 3 (FRXD3).
6	FTXIE3	**Function Transmit Interrupt Enable 3.** Enables transmit done interrupt for endpoint 3 (FTXD3).
5	FRXIE2	**Function Receive Interrupt Enable 2.** Enables receive done interrupt for endpoint 2 (FRXD2).
4	FTXIE2	**Function Transmit Interrupt Enable 2.** Enables transmit done interrupt for endpoint 2 (FTXD2).
3	FRXIE1	**Function Receive Interrupt Enable 1.** Enables receive done interrupt for endpoint 1 (FRXD1).
2	FTXIE1	**Function Transmit Interrupt Enable 1.** Enables transmit done interrupt for endpoint 1 (FTXD1).
1	FRXIE0	**Function Receive Interrupt Enable 0.** Enables receive done interrupt for endpoint 0 (FRXD0).
0	FTXIE0	**Function Transmit Interrupt Enable 0.** Enables transmit done interrupt for endpoint 0 (FTXD0).

For all bits, a 1 indicates the interrupt is enabled and causes an interrupt to be signaled to the microcontroller. A 0 indicates the associated interrupt source is disabled and cannot cause an interrupt. However, the interrupt bit's value is still reflected in the SBI register. All of these bits can be read/written by firmware.

Table 12. Serial Bus Interrupt Enable Register 1 (SBIE1)—Address: 17H; Default: 0000 0000B

It enables and disables the receive and transmit done interrupts for function endpoints 4 through 7.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRXIE7	FTXIE7	FRXIE6	FTXIE6	FRXIE5	FTXIE5	FRXIE4	FTXIE4
R/W							

Bit	Symbol	Function/Description
7	FRXIE7	**Function Receive Interrupt Enable 7.** Enables receive done interrupt for endpoint 7 (FRXD7).
6	FTXIE7	**Function Transmit Interrupt Enable 7.** Enables transmit done interrupt for endpoint 7 (FTXD7).
5	FRXIE6	**Function Receive Interrupt Enable 6.** Enables receive done interrupt for endpoint 6 (FRXD6).
4	FTXIE6	**Function Transmit Interrupt Enable 6.** Enables transmit done interrupt for endpoint 6 (FTXD6).
3	FRXIE5	**Function Receive Interrupt Enable 5.** Enables receive done interrupt for endpoint 5 (FRXD5).
2	FTXIE5	**Function Transmit Interrupt Enable 5.** Enables transmit done interrupt for endpoint 5 (FTXD5).
1	FRXIE4	**Function Receive Interrupt Enable 4.** Enables receive done interrupt for endpoint 4 (FRXD4).
0	FTXIE4	**Function Transmit Interrupt Enable 4.** Enables transmit done interrupt for endpoint 4 (FTXD4).

For all bits, a 1 indicates the interrupt is enabled and causes an interrupt to be signaled to the microcontroller. A 0 indicates the associated interrupt source is disabled and cannot cause an interrupt. However, the interrupt bit's value is still reflected in the SBI register. All of these bits can be read/written by firmware.

Register Interface (continued)

Table 13. Serial Bus Interrupt Register (SBI)—Address: 14H; Default: 0000 0000B

This register contains the USB function's transmit and receive done interrupt flags for nonisochronous endpoints. These bits are never set for isochronous endpoints.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRXD3	FTXD3	FRXD2	FTXD2	FRXD1	FTXD1	FRXD0	FTXD0
R/W (S*)							

Bit	Symbol	Function/Description
7	FRXD3	**Function Receive Done Flag, Endpoint 3.**
6	FTXD3	**Function Transmit Done Flag, Endpoint 3.**
5	FRXD2	**Function Receive Done Flag, Endpoint 2.**
4	FTXD2	**Function Transmit Done Flag, Endpoint 2.**
3	FRXD1	**Function Receive Done Flag, Endpoint 1.**
2	FTXD1	**Function Transmit Done Flag, Endpoint 1.**
1	FRXD0	**Function Receive Done Flag, Endpoint 0.**
0	FTXD0	**Function Transmit Done Flag, Endpoint 0.**

* S = shared bit. See Special Firmware Action for Shared Register Bits section.

For all bits in the interrupt flag register, a 1 indicates that an interrupt is actively pending; a 0 indicates that the interrupt is not active. The interrupt status is shown regardless of the state of the corresponding interrupt enable bit in the SBIE.

Hardware can only set bits to 1. In normal operation, firmware should only clear bits to 0. Firmware can also set the bits to 1 for test purposes. This allows the interrupt to be generated in firmware.

A set receive bit indicates either that valid data is waiting to be serviced in the RX FIFO for the indicated endpoint and that the data was received without error and has been acknowledged, or that data was received with a receive data error requiring firmware intervention to be cleared.

A set transmit bit indicates either that data has been transmitted from the TX FIFO for the indicated endpoint and has been acknowledged by the host, or that data was transmitted with an error requiring firmware intervention to be cleared.

Register Interface (continued)

Table 14. Serial Bus Interrupt 1 Register (SBI1)—Address: 15H; Default: 0000 0000B

This register contains the USB function's transmit and receive done interrupt flags for nonisochronous endpoints. These bits are never set for isochronous endpoints.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRXD7	FTXD7	FRXD6	FTXD6	FRXD5	FTXD5	FRXD4	FTXD4
R/W (S*)							

Bit	Symbol	Function/Description
7	FRXD7	**Function Receive Done Flag, Endpoint 7.**
6	FTXD7	**Function Transmit Done Flag, Endpoint 7.**
5	FRXD6	**Function Receive Done Flag, Endpoint 6.**
4	FTXD6	**Function Transmit Done Flag, Endpoint 6.**
3	FRXD5	**Function Receive Done Flag, Endpoint 5.**
2	FTXD5	**Function Transmit Done Flag, Endpoint 5.**
1	FRXD4	**Function Receive Done Flag, Endpoint 4.**
0	FTXD4	**Function Transmit Done Flag, Endpoint 4.**

* S = shared bit. See Special Firmware Action for Shared Register Bits section.

For all bits in the interrupt flag register, a 1 indicates that an interrupt is actively pending; a 0 indicates that the interrupt is not active. The interrupt status is shown regardless of the state of the corresponding interrupt enable bit in the SBIE.

Hardware can only set bits to 1. In normal operation, firmware should only clear bits to 0. Firmware can also set the bits to 1 for test purposes. This allows the interrupt to be generated in firmware.

A set receive bit indicates either that valid data is waiting to be serviced in the RX FIFO for the indicated endpoint and that the data was received without error and has been acknowledged, or that data was received with a receive data error requiring firmware intervention to be cleared.

A set transmit bit indicates either that data has been transmitted from the TX FIFO for the indicated endpoint and has been acknowledged by the host, or that data was transmitted with an error requiring firmware intervention to be cleared.

Register Interface (continued)

Table 15. Start of Frame High Register (SOFH)—Address: 0FH; Default: 0000 0000B

This register contains isochronous data transfer enable and interrupt bits and the upper 3 bits of the 11-bit time stamp received from the host.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SOFACK	ASOF	SOFIE	FTLOCK	SOFODIS	TS10	TS9	TS8
R	R/W (S*)	R/W (P*)	R	R/W (P*)	R/W (S*)		

Bit	Symbol	Function/Description
7	SOFACK	**SOF Token Received Without Error (Read Only)**. When set, this bit signifies that the 11-bit time stamp stored in SOFL and SOFH is valid. This bit is updated every time an SOF token is received from the USB bus, and it is cleared when an artificial SOF is generated by the frame timer. This bit is set and cleared by hardware.
6	ASOF	**Any Start of Frame.** This bit is set by hardware to signify that a new frame has begun. The interrupt can result either from the reception of an actual SOF packet or from an artificially generated SOF from the frame timer. This interrupt is asserted in hardware even if the frame timer is not locked to the USB bus frame timing. When set, this bit indicates that either the actual SOF packet was received or an artificial SOF was generated by the frame timer. Setting this bit to 1 by firmware has the same effect as when it is set by hardware. This bit must be cleared to 0 by firmware if SOFODIS = 1. If SOFODIS = 0, this bit clears itself after one t_{CLK}, and the SOFN device pin is driven low for eight t_{CLK} periods. In SOFODIS = 0 mode, because of the automatic resetting of ASOF, the system must detect start of frame via the SOFN device pin. This bit also serves as the SOF interrupt flag. This interrupt is only asserted in hardware if the SOF interrupt is enabled (SOFIE set) and the interrupt channel is enabled.
5	SOFIE	**SOF Interrupt Enable**. When set, setting the ASOF bit causes an interrupt request to be generated if the interrupt channel is enabled. Hardware reads this bit but does not write to it.
4	FTLOCK	**Frame Timer Lock (Read Only)**. When set, this bit signifies that the frame timer is presently locked to the USB bus frame time. When cleared, this bit indicates that the frame timer is attempting to synchronize the frame time.
3	SOFODIS	**SOF Pin Output Disable**. When set, no low pulse is driven to the SOF pin in response to setting the ASOF bit. The SOF pin is driven to 1 when SOFODIS is set. When this bit is clear, setting the ASOF bit causes the SOF pin to be toggled with a low pulse for eight t_{CLK} periods.
2:0	TS[10:8]	**Time Stamp Received from Host**. TS[10:8] are the upper 3 bits of the 11-bit frame number issued with an SOF token. This time stamp is valid only if the SOFACK bit is set.

* S = shared bit. P = PEND must be set when writing this bit. See Special Firmware Action for Shared Register Bits section.

Register Interface (continued)

Table 16. Start of Frame Low Register (SOFL)—Address: 0EH; Default: 0000 0000B

This register contains the lower 8 bits of the 11-bit time stamp received from the host.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
R/W (S*)							

Bit	Symbol	Function/Description
7:0	TS[7:0]	**Time Stamp Received from Host**. This time stamp is valid only if the SOFACK bit in the SOFH register is set. TS[7:0] are the lower 8 bits of the 11-bit frame number issued with an SOF token. The time stamp remains at its previous value if an artificial SOF is generated, and it is up to firmware to update it. These bits are set and cleared by hardware.

* S = shared bit. See Special Firmware Action for Shared Register Bits section.

Table 17. Endpoint Index Register (EPINDEX)—Address: 0AH; Default: 0000 0000B

This register identifies the endpoint pair. The register's contents select the transmit and receive FIFO pair and serve as an index to endpoint-specific special function registers (SFRs).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—					EPINX2	EPINX1	EPINX0
—					R/W		

Bit	Symbol	Function/Description
7:3	—	**Reserved.** Write 0s to these bits. Reads always return 0s.
2:0	EPINX[2:0]	**Endpoint Index.** **EPINDEX*** — **Function Endpoint** 0000 0000 — Function Endpoint 0 0000 0001 — Function Endpoint 1 0000 0010 — Function Endpoint 2 0000 0011 — Function Endpoint 3 0000 0100 — Function Endpoint 4 0000 0101 — Function Endpoint 5 0000 0110 — Function Endpoint 6 0000 0111 — Function Endpoint 7 The EPINDEX register must not be changed during a sequence of RXDAT reads of a particular data set. See the Receive FIFO section for more details.

* The EPINDEX register identifies the endpoint pair and selects the associated transmit and receive FIFO pair. The value in this register plus SFR addresses select the associated band of endpoint-indexed SFRs (TXDAT, TXCON, TXFLG, TXCNTH/L, RXDAT, RXCON, RXFLG, RXCNTH/L, EPCON, TXSTAT, and RXSTAT).

Register Interface (continued)

Table 18. Endpoint Control Register (EPCON)—Address: 0BH; Default: Endpoint 0 = 0011 0101B; Others = 0001 0000B

This SFR configures the operation of the endpoint specified by EPINDEX. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXSTL	TXSTL	CTLEP	RXSPM	RXIE	RXEPEN	TXOE	TXEPEN
R/W (S*)	R/W(P*)						

Bit	Symbol	Function/Description
7	RXSTL	**Stall Receive Endpoint**. When set, this bit stalls the receive endpoint. Firmware must clear this bit only after the host has intervened through commands sent down endpoint 0. When this bit is set and RXSETUP is clear, the receive endpoint responds with a STALL handshake to a valid OUT token. When this bit is set and RXSETUP is set, the receive endpoint will NACK. This bit does not affect the reception of SETUP tokens by a control endpoint. This bit is set by the hardware if the data phase of the status stage of a control transfer does not use the correct data PID (DATA1) or has more than 0 data bytes.
6	TXSTL	**Stall Transmit Endpoint**. When set, this bit stalls the transmit endpoint. Firmware must clear this bit only after the host has intervened through commands sent down endpoint 0. When this bit is set and RXSETUP is clear, the transmit endpoint responds with a STALL handshake to a valid IN token. When this bit is set and RXSETUP is set, the receive endpoint will NACK.
5	CTLEP	**Control Endpoint**. When set, this bit configures the endpoint as a control endpoint. Only control endpoints are capable of receiving SETUP tokens.
4	RXSPM	**Receive Single-Packet Mode**. When set, this bit configures the receive endpoint for single data packet operation. When enabled, only a single data packet is allowed to reside in the receive FIFO. **Note**: For control endpoints (CTLEP = 1), this bit should be set for single-packet mode operation as the recommended firmware model. But, it is possible to have a control endpoint configured in dual-packet mode as long as the firmware handles the endpoint correctly.
3	RXIE	**Receive Input Enable**. When set, this bit enables data from the USB to be written into the receive FIFO. If cleared, the endpoint responds to an OUT token by ignoring the data and returning a NACK handshake to the host (unless RXSTL is set, in which case a STALL is returned). This bit does not affect a valid SETUP token.
2	RXEPEN	**Receive Endpoint Enable**. When set, this bit enables the receive endpoint. When disabled, the endpoint does not respond to a valid OUT or SETUP token. This bit is hardware read only and has the highest priority among RXIE and RXSTL. **Note:** Endpoint 0 is enabled for reception upon reset.
1	TXOE	**Transmit Output Enable**. When set, this bit enables the data in TXDAT to be transmitted. If cleared, the endpoint returns a NACK handshake to a valid IN token if the TXSTL bit is not set.
0	TXEPEN	**Transmit Endpoint Enable**. When set, this bit enables the transmit endpoint. When disabled, the endpoint does not respond to a valid IN token. This bit is hardware read only. **Note:** Endpoint 0 is enabled for transmission upon reset.

* S = shared bit. P = PEND must be set when writing this bit. See Special Firmware Action for Shared Register Bits section.

Register Interface (continued)

Table 19. Endpoint Transmit Status Register (TXSTAT)—Address: 0CH; Default: 0000 0000B

This register contains the current endpoint status of the transmit FIFO specified by EPINDEX. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXSEQ	—		TXFLUSH	TXSOVW	TXVOID	TXERR	TXACK
R/W*	—		R	W	R		

Bit	Symbol	Function/Description
7	TXSEQ	**Transmitter Current Sequence Bit (Read, Conditional Write).*** This bit is transmitted in the next PID and toggled on a valid ACK handshake. This bit is toggled by hardware on a valid SETUP token. This bit can be written by firmware if the TXSOVW bit is set when written together with the next TXSEQ value.
6:5	—	**Reserved.** Write 0s to these bits. Reads always return 0s.
4	TXFLUSH	**Transmit FIFO Packet Flushed (Read Only).** When set, this bit indicates that hardware flushed a stale isochronous data packet from the transmit FIFO due to a TXFIF[1:0] = 11 condition at SOF. To guard against a missed IN token in isochronous mode, if, with TXFIF[1:0] = 11, no IN token is received for the current endpoint, hardware automatically flushes the oldest packet and decrements the TXFIF[1:0] value. This flush does not occur if there is only one data set present (TXFIF = 01/10).
3	TXSOVW	**Transmit Data Sequence Overwrite Bit.*** Writing a 1 to this bit allows the value of the TXSEQ bit to be overwritten. Writing a 0 to this bit has no effect on TXSEQ. This bit always returns 0 when read.
2	TXVOID	**Transmit Void (Read Only).** Indicates a void condition has occurred in response to a valid IN token. Transmit void is closely associated with the NACK/STALL handshake returned by the function after a valid IN token. This void condition occurs when the endpoint output is disabled (TXOE = 0) or stalled (TXSTL = 1), the corresponding receive FIFO contains a setup packet (RXSETUP = 1), the FIFO contains no valid data sets (TXFIF = 00), or there is an existing FIFO error (TXURF = 1 or TXOVF = 1). This bit is used to check any NACK/STALL handshake returned by the function. This bit does not affect the FTXDx, TXERR, or TXACK bits. This bit is updated by hardware at the end of a nonisochronous transaction in response to a valid IN token. For isochronous transactions, this bit is not updated until the next SOF.
1	TXERR	**Transmit Error (Read Only).** Indicates an error condition has occurred with the transmission. Complete or partial data has been transmitted. The error can be one of the following: 1. Data transmitted successfully but no handshake received. 2. Transmit FIFO goes into underrun condition while transmitting. The corresponding transmit done bit, FTXDx in SBI or SBI1, is set when active. For nonisochronous transactions, this bit is updated by hardware along with the TXACK bit at the end of data transmission (this bit is mutually exclusive with TXACK). For isochronous transactions, that bit is not updated until the next SOF.
0	TXACK	**Transmit Acknowledge (Read Only).** Indicates data transmission completed and acknowledged successfully. The corresponding transmit done bit, FTXDx in SBI or SBI1, is set when active. For nonisochronous transactions, this bit is updated by hardware along with the TXERR bit at the end of data transmission (this bit is mutually exclusive with TXERR). For isochronous transactions, that bit is not updated until the next SOF.

* For normal operation, this bit should not be modified by the user except as required by the implementation of USB standard commands, such as SET_CONFIGURATION, SET_INTERFACE, and CLEAR_FEATURE [stall]. The SIE handles all sequence bit tracking required by normal USB traffic, as documented in the USB specification, Section 8.6.

Register Interface (continued)

Table 20. Endpoint Receive Status Register (RXSTAT)—Address: 0DH; Default: 0000 0000B

This register contains the current endpoint status of the receive FIFO specified by EPINDEX. This register is an endpoint-indexed SFR.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	RXVOID	RXERR	RXACK
R/W* (P†)	R/W (S†)	R	R/W (S†)	W (P†)	R		

Bit	Symbol	Function/Description
7	RXSEQ	**Receiver Endpoint Sequence Bit (Read, Conditional Write).*** This bit is toggled on completion of an ACK handshake in response to an OUT token. This bit is set (or cleared) by hardware after reception of a SETUP token. If the RXSOVW bit is set, this bit can be written by firmware when written along with the new RXSEQ value. **Note:** Always verify this bit after writing to ensure that there is no conflict with hardware, which may occur if a new SETUP token is received.
6	RXSETUP	**Received SETUP Token**. This bit is set by hardware when a valid SETUP token has been received. When set, this bit causes received IN or OUT tokens to be NACKed until the bit is cleared to allow proper data management for the transmit and receive FIFOs from the previous transaction. IN or OUT tokens are NACKed even if the endpoint is stalled (RXSTL or TXSTL) to allow a control transaction to clear a stalled endpoint. Clear this bit upon detection of a SETUP token after the firmware is ready to complete the status stage of a control transaction. For a stalled control endpoint, this bit should not be cleared until the RXSTL/TXSTL bits have been cleared.
5	STOVW	**Start Overwrite Flag (Read Only).** This bit is set by hardware upon receipt of a SETUP token for any control endpoint to indicate that the receive FIFO is being overwritten with new SETUP data. When set, the FIFO state (RXFIF and read pointer) resets and is locked for this endpoint until EDOVW is set. This prevents a prior, ongoing firmware read from corrupting the read pointer as the receive FIFO is being cleared and new data is being written into it. This bit is cleared by hardware at the end of handshake phase transmission of the SETUP stage. This bit is used only for control endpoints.
4	EDOVW	**End Overwrite Flag.** This flag is set by hardware during the handshake phase of a SETUP stage. It is set after every SETUP packet is received and must be cleared prior to reading the contents of the FIFO. When set, the FIFO state (RXFIF and read pointer) remains locked for this endpoint until this bit is cleared. This prevents a prior, ongoing firmware read from corrupting the read pointer after the new data has been written into the receive FIFO. This bit is used only for control endpoints.
3	RXSOVW	**Receive Data Sequence Overwrite Bit.*** Writing a 1 to this bit allows the value of the RXSEQ bit to be overwritten. Writing a 0 to this bit has no effect on RXSEQ. This bit always returns 0 when read.

* For normal operation, this bit should not be modified by the user except as required by the implementation of USB standard commands, such as SET_CONFIGURATION, SET_INTERFACE, and CLEAR_FEATURE [stall]. The SIE handles all sequence bit tracking required by normal USB traffic, as documented in the USB specification, Section 8.6.
† S = shared bit. P = PEND must be set when writing this bit. See Special Firmware Action for Shared Register Bits section.

Register Interface (continued)

Table 20. Endpoint Receive Status Register (RXSTAT)—Address: 0DH; Default: 0000 0000B (continued)

Bit	Symbol	Function/Description
2	RXVOID	**Receive Void (Read Only).** Indicates a void condition has occurred in response to a valid OUT token. Receive void is closely associated with the NACK/STALL handshake returned by the function after a valid OUT token. This void condition occurs when the endpoint input is disabled (RXIE = 0) or stalled (RXSTL = 1), the FIFO contains a setup packet (RXSETUP = 1), the FIFO has no available data sets (RXFIF = 11, or RXFIF = 01/10 and RXSPM = 1), or there is an existing FIFO error (RXURF = 1 or RXOVF = 1). This bit is set and cleared by hardware. For nonisochronous transactions, this bit is updated by hardware at the end of the transaction in response to a valid OUT token. For isochronous transactions, it is not updated until the next SOF.
1	RXERR	**Receive Error (Read Only).** Set when an error condition has occurred with the reception of a SETUP or OUT transaction. Complete or partial data has been written into the receive FIFO. No handshake is returned. The error can be one of the following: 1. Data failed CRC check. 2. Bit stuffing error. 3. A receive FIFO goes into overrun or underrun condition while receiving. This bit is updated by hardware at the end of a valid SETUP or OUT token transaction (nonisochronous) or at the next SOF on each valid OUT token transaction (isochronous). The corresponding FRXDx bit of SBI or SBI1 is set when active. This bit is updated with the RXACK bit at the end of data reception and is mutually exclusive with RXACK.
0	RXACK	**Receive Acknowledge (Read Only).** This bit is set when data is received completely into a receive FIFO and an ACK handshake is sent. This read-only bit is updated by hardware at the end of a valid SETUP or OUT token transaction (nonisochronous) or at the next SOF on each valid OUT token transaction (isochronous). The corresponding FRXDx bit of SBI or SBI1 is set when active. This bit is updated with the RXERR bit at the end of data reception and is mutually exclusive with RXERR.

Register Interface (continued)

Table 21. Function Address Register (FADDR)—Address: 10H; Default: 0000 0000B

This SFR holds the address for the USB function. During bus enumeration, it is written with a unique value assigned by the host.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	A6	A5	A4	A3	A2	A1	A0
—	R/W						

Bit	Symbol	Function/Description
7	—	**Reserved.** Write 0 to this bit. Reads always return 0.
6:0	A[6:0]	**7-Bit Programmable Function Address.** This register is written by firmware as a result of commands received via endpoint 0. Hardware can only read this register.

Table 22. Transmit FIFO Data Register (TXDAT)—Address: 00H; Default: 0000 0000B

Data to be transmitted by the FIFO specified by EPINDEX is first written to the register. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXDAT7	TXDAT6	TXDAT5	TXDAT4	TXDAT3	TXDAT2	TXDAT1	TXDAT0
W							

Bit	Symbol	Function/Description
7:0	TXDAT[7:0]	**Transmit Data Byte (Write Only).** To write data to the transmit FIFO, write to this register. The write pointer is incremented automatically after a write.

Table 23. Transmit FIFO Byte-Count High and Low Registers (TXCNTH, TXCNTL)—Address: TXCNTH = 02H, TXCNTL = 01H; Default: TXCNTH = 0000 0000B; TXCNTL = 0000 0000B

Written by firmware to indicate the number of bytes written to the transmit FIFO specified by EPINDEX. This register is endpoint indexed. TXCNTL should be written after TXCNTH.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
—						BC9	BC8
—						R/W	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
R/W							

Bit	Symbol	Function/Description
15:10	—	**Reserved.** Write 0s to these bits. Reads always return 0s.
9:0	BC[9:0]	**Transmit Byte Count (Write, Conditional Read).*** 10-bit, ring buffer. These bits store transmit byte count (TXCNT).

* Read these bits only if TXFIF[1:0] = 0; otherwise, underrun errors may occur.

Note: To send a status stage after a control write, no data control command, or a null packet, write 0 to TXCNT.

Register Interface (continued)

Table 24. USB Transmit FIFO Control Register (TXCON)—Address: 03H; Default: 0000 0100B

This register controls the transmit FIFO specified by EPINDEX. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXCLR	FFSZ1	FFSZ0	—	TXISO	ATM	ADVRM	REVRP
R/W			—	R/W			

Bit	Symbol	Function/Description
7	TXCLR	**Transmit FIFO Clear.** Setting this bit flushes the transmit FIFO, resets all the read/write pointers and markers, resets the TXCNTH and TXCNTL registers, resets the TXFLUSH, TXVOID, TXERR, and TXACK bits of the TXSTAT register, sets the TXEMP bit in TXFLG, and clears all other bits in TXFLG. Hardware clears this bit after the flush. Setting this bit does not affect the TXSEQ bit in the TXSTAT register. This bit should only be set when the endpoint is known to be inactive or there is a FIFO error present.
6:5	FFSZ[1:0]	**FIFO Size.** These bits select the size of the transmit FIFO. **FFSZ[1:0] / Nonisochronous Size / Isochronous Size** 00 / 16 / 64 01 / 64 / 256 10 / 64 / 512 11 / 64 / 1024
4	—	**Reserved.** Write 0 to this bit. Reads always return 0.
3	TXISO	**Transmit Isochronous Data.** Firmware sets this bit to indicate that the transmit FIFO contains isochronous data. The SIE uses this bit to determine if a handshake is required at the end of a transmission.
2	ATM	**Automatic Transmit Management.*** Setting this bit (the default value) causes the read pointer and read marker to be adjusted automatically as indicated: **Status / Read Pointer / Read Marker** ACK / Unchanged / Advanced (1) NACK / Reversed (2) / Unchanged 1. To origin of next data set. 2. To origin of the data set last read. This bit should always be set, except for test purposes. Setting this bit disables ADVRM and REVRP. This bit can be set and cleared by firmware. Hardware neither clears nor sets this bit. This bit must always be set for isochronous endpoints (TXISO = 1).
1	ADVRM	**Advance Read Marker Control (Non-ATM Mode Only).*** Setting this bit prepares for the next packet transmission by advancing the read marker to the origin of the next data packet (the position of the read pointer). Hardware clears this bit after the read marker is advanced. This bit is effective only when the REVRP, ATM, and TXCLR bits are clear.
0	REVRP	**Reverse Read Pointer (Non-ATM Mode Only).*** In the case of a bad transmission, the same data stack may need to be available for retransmit. Setting this bit reverses the read pointer to point to the origin of the last data set (the position of the read marker) so that the SIE can reread the last set for retransmission. Hardware clears this bit after the read pointer is reversed. This bit is effective only when the ADVRM, ATM, and TXCLR bits are all clear.

* ATM mode is recommended. ADVRM and REVRP, which control the read marker and read pointer when ATM = 0, are used for test purposes.

Register Interface (continued)

Table 25. Transmit FIFO Flag Register (TXFLG)—Address: 04H; Default: 0000 1000B

These flags indicate the status of data packets in the transmit FIFO specified by EPINDEX. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXFIF1	TXFIF0	—		TXEMP	TXFULL	TXURF	TXOVF
R		—		R		R/W	

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function/Description</th></tr>
<tr><td>7:6</td><td>TXFIF[1:0]</td><td>Transmit FIFO Index Flags (Read Only). These flags indicate which data sets are present in the transmit FIFO (see below).

Data Sets Present
<table>
<tr><th>TXFIF[1:0]</th><th>ds1</th><th>ds0</th><th>Status</th></tr>
<tr><td>00</td><td>No</td><td>No</td><td>Empty</td></tr>
<tr><td>01</td><td>No</td><td>Yes</td><td>1 set</td></tr>
<tr><td>10</td><td>Yes</td><td>No</td><td>1 set</td></tr>
<tr><td>11</td><td>Yes</td><td>Yes</td><td>2 sets</td></tr>
</table>
The TXFIF bits are set in sequence after each write to TXCNT to reflect the addition of a data set. Likewise, the TXFIF1 and TFIF0 are cleared in sequence after each advance of the read marker to indicate that the set is effectively discarded. The bit is cleared whether the read marker is advanced by firmware (setting ADVRM) or automatically by hardware (ATM = 1). The next-state table for the TXFIF bits is shown below:
<table>
<tr><th>TXFIF[1:0]</th><th>Operation</th><th>Next TXFIF[1:0]</th></tr>
<tr><td>00</td><td>Write TXCNT</td><td>01</td></tr>
<tr><td>01</td><td>Write TXCNT</td><td>11</td></tr>
<tr><td>10</td><td>Write TXCNT</td><td>11</td></tr>
<tr><td>11</td><td>Write TXCNT</td><td>11 (TXOVF = 1)</td></tr>
<tr><td>00</td><td>Advance Read Marker</td><td>00</td></tr>
<tr><td>01</td><td>Advance Read Marker</td><td>00</td></tr>
<tr><td>11</td><td>Advance Read Marker</td><td>10/01</td></tr>
<tr><td>10</td><td>Advance Read Marker</td><td>00</td></tr>
<tr><td>XX</td><td>Reverse Read Pointer</td><td>Unchanged</td></tr>
</table>
In isochronous mode, TXOVF, TXURF, and TXFIF are handled using the following rule: firmware events cause status change immediately, while USB events cause status change only at SOF. TXFIF is incremented by firmware and decremented by the USB. Therefore, writes to TXCNT increment TXFIF immediately. However, a successful USB transaction any time within a frame decrements TXFIF only at SOF.

The TXFIF flags must be checked before and after writes to the transmit FIFO and TXCNT for traceability. See the TXFLUSH bit in TXSTAT.

Note: Firmware can enforce single-packet mode by only writing a new data set to the transmit FIFO if there are currently no data sets present in the FIFO (TXFIF = 00). To simplify firmware development, configure control endpoints in single-packet mode.</td></tr>
<tr><td>5:4</td><td>—</td><td>Reserved. Write 0s to these bits. Reads always return 0s.</td></tr>
<tr><td>3</td><td>TXEMP</td><td>Transmit FIFO Empty Flag (Read Only). Hardware sets this bit when firmware has not yet written any data bytes to the current FIFO data set being written. Hardware clears this bit when the empty condition no longer exists.

This bit always tracks the current transmit FIFO status regardless of isochronous or nonisochronous mode.</td></tr>
</table>

Register Interface (continued)

Table 25. Transmit FIFO Flag Register (TXFLG)—Address: 04H; Default: 0000 1000B (continued)

Bit	Symbol	Function/Description
2	TXFULL	**Transmit FIFO Full Flag (Read Only).** Hardware sets this bit when the number of bytes that firmware writes to the current transmit FIFO data set equals the FIFO size. Hardware clears this bit when the full condition no longer exists. This bit always tracks the current transmit FIFO status regardless of isochronous or nonisochronous mode. Check this bit to avoid causing a TXOVF condition.
1	TXURF	**Transmit FIFO Underrun Flag (Read, Clear Only).** Hardware sets this flag when a read is attempted from an empty transmit FIFO. (This is caused when the value written to TXCNT is greater than the number of bytes written to TXDAT.) This bit must be cleared by firmware through TXCLR. When this flag is set, the FIFO is in an unknown state; therefore, it is recommended that the FIFO is reset in the error management routine using the TXCLR bit in TXCON. When the transmit FIFO underruns, the read pointer does not advance; it remains locked in the empty position. When this bit is set, all transmissions are NACKed. In isochronous mode, TXOVF, TXURF, and TXFIF are handled using the following rule: firmware events cause status change immediately, while USB events cause status change only at SOF. Since underrun can only be caused by USB, TXURF is updated at the next SOF regardless of where the underrun occurs in the frame.
0	TXOVF	**Transmit FIFO Overrun Flag (Read, Clear Only).** This bit is set when an additional byte is written to a full FIFO, or TXCNT is written while TXFIF[1:0] = 11. This bit must be cleared by firmware through TXCLR. When this bit is set, the FIFO is in an unknown state; thus, it is recommended that the FIFO is reset in the error management routine using the TXCLR bit in TXCON. When the transmit FIFO overruns, the write pointer does not advance; it remains locked in the full position. Check this bit after loading the FIFO prior to writing the byte count register. When this bit is set, all transmissions are NACKed. In isochronous mode, TXOVF, TXURF, and TXFIF are handled using the following rule: firmware events cause status change immediately, while USB events cause status change only at SOF. Since overrun can only be caused by firmware, TXOVF is updated immediately. Check the TXOVF flag after writing to the transmit FIFO before writing to TXCNT.

Table 26. Receive FIFO Data Register (RXDAT)—Address: 05H; Default: 0000 0000B

Receive FIFO data specified by EPINDEX is stored and read from this register. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXDAT[7:0]							
R							

Bit	Symbol	Function/Description
7:0	RXDAT[7:0]	**Receive FIFO Data Register (Read Only).** To write to the receive FIFO, the SIE writes to this register. To read data from the receive FIFO, the CPU reads from this register. The write pointer and read pointer are incremented automatically after a write and read, respectively. The EPINDEX register must not be changed during a sequence of RXDAT reads of a particular data set. See the Receive FIFO section for more details.

Register Interface (continued)

Table 27. Receive FIFO Byte-Count High and Low Registers (RXCNTH, RXCNTL)—Address: RXCNTH = 07H, RXCNTL = 06H; Default: RXCNTH = 0000 0000B, RXCNTL = 0000 0000B

High and low registers are in a two-register ring buffer that is used to store the byte count for the data packets received in the receive FIFO specified by EPINDEX. These registers are endpoint indexed.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
—						BC9	BC8
—						R	

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
R							

Bit	Symbol	Function/Description
15:10	—	**Reserved.** Write 0s to these bits. Reads always return 0s.
9:0	BC[9:0]	**Receive Byte Count (Read Only).** 10-bit, ring buffer byte. Stores receive byte count (RXCNT).

Table 28. Receive FIFO Control Register (RXCON)—Address: 08H; Default: 0000 0100B

Controls the receive FIFO specified by EPINDEX. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXCLR	FFSZ1	FFSZ0	RXFFRC	RXISO	ARM	ADVWM	REVWP
R/W							

Bit	Symbol	Function/Description
7	RXCLR	**Receive FIFO Clear.** Setting this bit flushes the receive FIFO, resets all the read/write pointers and markers, resets the RXSETUP, STOVW, EDOVW, RXVOID, RXERR, and RXACK bits of the RXSTAT register, sets the RXEMP bit in RXFLG register, and clears all other bits in RXFLG register. Hardware clears this bit when the flush operation is completed. Setting this bit does not affect the RXSEQ bit of RXSTAT. This bit should only be set when the endpoint is disabled or there is a FIFO error present. Firmware should never set this bit to clear a SETUP packet. The next SETUP packet will automatically clear the receive FIFO.
6:5	FFSZ[1:0]	**FIFO Size.** These bits select the size of the receive FIFO. **FFSZ[1:0] — Nonisochronous Size — Isochronous Size** 00 — 16 — 64 01 — 64 — 256 10 — 64 — 512 11 — 64 — 1024
4	RXFFRC	**FIFO Read Complete.** When set, the receive FIFO is released when a data set read is complete. Setting this bit clears the RXFIF bit (in the RXFLG register), corresponding to the data set that was just read. Hardware clears this bit after the RXFIF bit is cleared. All data from this data set must have been read. **Note:** FIFO read complete only works if the STOVW and EDOVW bits are both cleared.

Register Interface (continued)

Table 28. Receive FIFO Control Register (RXCON)—Address: 08H; Default: 0000 0100B (continued)

Bit	Symbol	Function/Description
3	RXISO	**Receive Isochronous Data.** When set, this indicates that the receive FIFO is programmed to receive isochronous data and to set up the USB interface to handle an isochronous data transfer.
2	ARM	**Auto Receive Management.*** When set, the write pointer and write marker are adjusted automatically based on the following conditions: **RX Status** — **Write Pointer** — **Write Marker** ACK — Unchanged — Advanced NACK — Reversed — Unchanged This bit should always be set, except for test purposes. When this bit is set, setting REVWP or ADVWM has no effect. Hardware neither clears nor sets this bit. This bit can be set and cleared by firmware. This bit must always be set for isochronous endpoints (RXISO = 1).
1	ADVWM	**Advance Write Marker (Non-ARM Mode Only).*** When set, the write marker is advanced to the origin of the next data set. Advancing the write marker is used for back-to-back receptions. Hardware clears this bit after the write marker is advanced. Setting this bit is effective only when the REVWP, ARM, and RXCLR bits are clear.
0	REVWP	**Reverse Write Pointer (Non-ARM Mode Only).*** When set, the write pointer is returned to the origin of the last data set received, as identified by the write marker. The SIE can then reread the last data packet and write to the receive FIFO starting from the same origin when the host resends the same data packet. Hardware clears this bit after the write pointer is reversed. Setting this bit is effective only when the ADVWM, ARM, and RXCLR bits are clear. REVWP is used when a data packet is bad. When the function interface receives the data packet again, the write starts at the origin of the previous (bad) data set.

* ARM mode is recommended. ADVWM and REVWP, which control the write marker and write pointer when ARM = 0, are used for test purposes.

Register Interface (continued)

Table 29. Receive FIFO Flag Register (RXFLG)—Address: 09H; Default: 0000 1000B

These flags indicate the status of the data packets in the receive FIFO specified by EPINDEX. This register is endpoint indexed.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXFIF1	RXFIF0	—		RXEMP	RXFULL	RXURF	RXOVF
R		—		R		R/W	

Bit	Symbol	Function/Description
7:6	RXFIF[1:0]	**Receive FIFO Index Flags (Read Only).** These read-only flags indicate which data packets are present in the receive FIFO (see below). **Data Sets Present** **RXFIF[1:0] / ds1 / ds0 / Status** 00 / No / No / Empty 01 / No / Yes / 1 set 10 / Yes / No / 1 set 11 / Yes / Yes / 2 sets The RXFIF bits are updated after each write to RXCNT to reflect the addition of a data packet. Likewise, the RXFIF bits are cleared in sequence after each setting of the RXFFRC bit. The next-state table for RXFIF bits is shown below for operation in dual-packet mode. **RXFIF[1:0] / Operation / Next RXFIF[1:0]** 00 / Advance Write Marker / 01 01 / Advance Write Marker / 11 10 / Advance Write Marker / 11 11 / Advance Write Marker Not Possible—Device will NACK any OUT. / 11 00 / Set RXFFRC / 00 01 / Set RXFFRC / 00 11 / Set RXFFRC / 10/01 10 / Set RXFFRC / 00 00 / Reverse Write Pointer / Unchanged When the receive FIFO is programmed to operate in single-packet mode (RXSPM set in EPCON), valid RXFIF states are 00 and 01 only. In isochronous mode, RXOVF, RXURF, and RXFIF are handled using the following rule: firmware events cause status change immediately, while USB events cause status change only at SOF. RXFIF is incremented by the USB and decremented by firmware. Therefore, setting RXFFRC decrements RFIF immediately. However, a successful USB transaction within a frame increments RXFIF only at SOF. For traceability, the RXFIF flags must be checked before and after reads from the receive FIFO and the setting of RXFFRC in RXCON. **Note:** To simplify firmware development, it is recommended that control endpoints are used in single-packet mode only.
5:4	—	**Reserved.** Write 0s to these bits. Reads always return 0s.

Register Interface (continued)

Table 29. Receive FIFO Flag Register (RXFLG)—Address: 09H; Default: 0000 1000B (continued)

Bit	Symbol	Function/Description
3	RXEMP	**Receive FIFO Empty Flag (Read Only).** Hardware sets this flag when there are no data bytes present in the data set currently being read. Hardware clears the bit when the empty condition no longer exists. This bit always tracks the current status of the receive FIFO, regardless of isochronous or nonisochronous mode.
2	RXFULL	**Receive FIFO Full Flag (Read Only).** Hardware sets this flag when the data set currently being read contains the same number of data bytes as the size of the FIFO. Hardware clears the bit when the full condition no longer exists. This bit always tracks the current status of the receive FIFO regardless of isochronous or nonisochronous mode.
1	RXURF	**Receive FIFO Underrun Flag (Read, Clear Only).** Hardware sets this bit when an additional byte is read from an empty receive FIFO or when RXCNTH or RXCNTL is read while RXFIF[1:0] = 00. Hardware does not clear this bit, so it must be cleared by firmware through RXCLR. When the receive FIFO underruns, the read pointer does not advance. It remains locked in the empty position. When this bit is set, all transmissions are NACKed. In isochronous mode, RXOVF, RXURF, and RXFIF are handled using the following rule: firmware events cause status change immediately, while USB events cause status change only at SOF. Since underrun can only be caused by firmware, RXURF is updated immediately. The RXURF flag must be checked after reads from the receive FIFO before setting the RXFFRC bit in RXCON. **Note:** When this bit is set, the FIFO is in an unknown state. It is recommended that the FIFO is reset in the error management routine using the RXCLR bit in the RXCON register.
0	RXOVF	**Receive FIFO Overrun Flag (Read, Clear Only).** This bit is set when the SIE writes an additional byte to a full receive FIFO or writes a byte count to RXCNT with RXFIF[1:0] = 11. This bit must be cleared by firmware through RXCLR, although it can be cleared by hardware if a SETUP packet is received after an RXOVF error has already occurred. When this bit is set, all transmissions are NACKed. In isochronous mode, RXOVF, RXURF, and RXFIF are handled using the following rule: firmware events cause status change immediately, while USB events cause status change only at SOF. Since overrrun can only be caused by the USB, RXOVF is updated only at the next SOF regardless of where the overrun occurred during the current frame. **Note:** When this bit is set, the FIFO is in an unknown state. It is recommended that the FIFO is reset in the error management routine using the RXCLR bit in the RXCON register. When the receive FIFO overruns, the write pointer does not advance. It remains locked in the full position.

Register Interface (continued)

Table 30. System Control Register (SCR)—Address: 11H; Default: 0000 0000B

This register controls the FIFO mode, IRQ mask, and IRQ mode selection.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQPOL	RWUPE	IE_SUSP	IE_RESET	SRESET	IRQLVL	T_IRQ	—
R/W							—

Bit	Symbol	Function/Description
7	IRQPOL	**IRQ Polarity.** Determines the polarity of the IRQN output. When asserted, the IRQN output is active-high (default is active-low). Firmware must be careful to ensure that setting this bit does not cause a false interrupt to be detected and processed.
6	RWUPE	**Enable Remote Wake-Up Feature.** When set, remote wake-up is enabled.
5	IE_SUSP	**Enable Suspend Interrupt.** When set, the SUSPEND interrupt is enabled.
4	IE_RESET	**Enable Reset Interrupt.** When set, the RESET interrupt is enabled.
3	SRESET	**Software Reset.** Setting this bit to 1 in software places the USS-820/USS-825 in the RESET state. This is equivalent to asserting the hardware RESET pin, except that this feature is not available if the device is suspended. Setting this bit back to 0 leaves the USS-820/USS-825 in an unconfigured state that follows a hardware reset.
2	IRQLVL	**Interrupt Mode.** Level mode interrupt is selected when this bit is cleared. Pulse mode interrupt is selected when this bit is set. In pulse mode, IRQ signal is driven (high or low, depending on the IRQPOL setting) by USS-820/USS-825 for two t_{CLK} periods.
1	T_IRQ	**Global Interrupt Enable.** When this bit is set, it enables hardware interrupt to be generated on IRQ pin when any of TX/RX bits, ASOF bit, RESET bit, or SUSPEND bit is set.
0	—	**Reserved.** Write 0 to this bit. Reads always return 0.

Register Interface (continued)

Table 31. System Status Register (SSR)—Address: 12H; Default: 0000 0000B

This register allows control and monitoring of the USB suspend and reset events.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1		Bit 0
—			SUSPPO	SUSPDIS	RESUME	SUSPEND		RESET
—			R/W (P*)			R	W (P*)	R/W (S*)

Bit	Symbol	Function/Description
7:5	—	**Reserved.** Write 0s to these bits. Reads always return 0s.
4	SUSPPO	**Suspend Power Off.** This bit must be set by firmware if the external device will be powered off during a suspend. The correct value of this bit must be established before firmware suspends the device.
3	SUSPDIS	**Suspend Disable.** When asserted, this bit disables the detection of a USB suspend event. This bit is for test purposes and should not be set during normal system operation.
2	RESUME	**Resume Detected.** For a complete description of the use of this bit, see the Suspend and Resume Behavior section of this document. When set, the USS-820/USS-825 has detected and responded to a wake-up condition, either global or remote. A global resume is indicated when the host asserts a non-IDLE state on the USB bus. A remote wake-up is indicated when the device asserts the RWUPN input pin (if that feature is enabled by the RWUPE bit). This bit should be reset by firmware as soon as possible after resuming to allow the next suspend event to be detected.
1	SUSPEND	**Suspend Detected (Read Only)/Suspend Control (Write Only).** For a complete description of the use of this bit, see the Suspend and Resume Behavior section of this document. This bit serves as both a read-only status bit and a write-only control bit. For this reason, firmware cannot do a simple read/modify/write sequence to update this register. Firmware must always explicitly specify the correct value of this SUSPEND control bit when writing SSR. The read-only status bit is set by hardware when a SUSPEND condition is detected on the USB bus, and clears itself after the SUSPEND condition ceases and the device resumes. The bit will remain set during device wake-up. The value of this read-only bit is not affected by firmware writes. The write-only control bit is only updated by firmware, and is used to suspend the device by setting the bit to 1, and then setting the bit to 0. This write sequence will cause the device to suspend regardless of the initial value of the bit, which cannot be read.
0	RESET	**USB Reset Detected.** When set, a RESET condition is detected on the USB bus. If interrupt is enabled (T_IRQ and IE_RESET set), an interrupt is generated to the controller. Firmware clears this bit.

* S = shared bit. P = PEND must be set when writing this bit. See Special Firmware Action for Shared Register Bits section.

Table 32. Hardware Revision Register (REV)—Address: 18H; Default: 0001 0000B

This register contains the hardware revision number, which will be incremented for each version of the hardware. This will allow firmware to query the hardware status and determine which functions or features are supported.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Main Hardware Revision Number				Sub Hardware Revision Number			
R							

Bit	Symbol	Function/Description
7:4	—	**Main Hardware Revision Number.**
3:0	—	**Sub Hardware Revision Number.**

Register Interface (continued)

Table 33. Suspend Power-Off Locking Register (LOCK)—Address: 19H; Default: 0000 0001B

This register contains the control and status which enables the USS-820/USS-825 locking mechanism. This feature protects the internal register set from being corrupted during and immediately after a suspend where the external controller is powered off. The feature is enabled by the SUSPLOE bit, and its proper usage is documented in *Special Action Required by USS-820 After Suspend* (AP97-058CMPR).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—							UNLOCKED
—							R/W

Bit	Symbol	Function/Description
7:1	—	**Reserved.**
0	UNLOCKED	**Locking Control/Status.** Use of this bit is described in application note, *Special Action Required by USS-820 After Suspend* (AP97-058CMPR).

Table 34. Pend Hardware Status Update Register (PEND)—Address: 1AH; Default: 0000 0000B

This register contains the PEND bit.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—							PEND
—							R/W

Bit	Symbol	Function/Description
7:1	—	**Reserved.**
0	PEND	**Pend.** When set, this bit modifies the behavior of other shared register bits. See the Special Firmware Action for Shared Register Bits section of this document for a detailed explanation.

Table 35. Scratch Firmware Information Register (SCRATCH)—Address: 1BH; Default: 0000 0000B

This register contains a 7-bit scratch field that can be used by firmware to save and restore information. One possible use would be to save the device's USB state (e.g., DEFAULT, ADDRESSED) during suspend power off. The register also contains the resume interrupt enable bit.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE_RESUME	SCRATCH						
R/W	R/W						

Bit	Symbol	Function/Description
7	IE_RESUME	**Enable Resume Interrupt.** When set, the RESUME interrupt is enabled.
6:0	SCRATCH	**Scratch Information.**

Register Interface (continued)

**Table 36. Miscellaneous Control/Status Register (MCSR)—Address: 1CH;
Default: 0000 0000B (44-Pin MQFP—USS-820)
0001 0000B (48-Pin TQFP—USS-825)**

This register contains miscellaneous control and status bits.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	INIT	SUSPS	PKGID	—	—	SUSPLOE	DPEN
—	R	R	R	—	—	R/W	R/W

Bit	Symbol	Function/Description
7	—	**Reserved.**
6	INIT	**Device Initialized.** This bit will read 0 until internal clocks are turned on after a hardware reset. This bit is not affected by software reset. This bit can be used by firmware to determine when the device is operational after a hardware reset.
5	SUSPS	**Suspend Status.** Indicates the current suspended status of the device. This bit will be set when the device goes suspended and will remain set until internal clocks are turned back on at the end of a resume sequence.
4	PKGID	**Package Identification.** Indicates the package type. This bit will read 0 for the 44-pin MQFP package (USS-820) and 1 for the 48-pin TQFP package (USS-825). This value is established at the end of a hardware reset sequence.
3:2	—	**Reserved.**
1	SUSPLOE	**Suspend Lock Out Enable.** Enables the device locking mechanism, which will then engage on every device resume. The correct value of this bit must be established before firmware suspends the device.
0	DPEN	**DPLS Pull-Up Enable.** Controls the DDPU output pin, which may be used to power the external DPLS pull-up resistor. This can be used by firmware to make the device appear disconnected from the host without a physical disconnect. When DPEN = 1, the DPPU output pin is driven high. When DPEN = 0, the DPPU output pin is 3-stated.

Interrupts

Figure 8 describes the device interrupt logic. Each of the indicated USB events are logged in a status register bit. Each status bit has a corresponding enable bit that allows the event to cause an interrupt. Interrupts can be masked globally by the T_IRQ bit of the SCR register. The active level and signaling mode (level vs. pulse) of the IRQN output pin can be controlled by the IRQPOL and IRQLVL bits of the SCR register. All interrupts have equal priority—firmware establishes its own priority by the order in which it checks these status bits during interrupt processing.

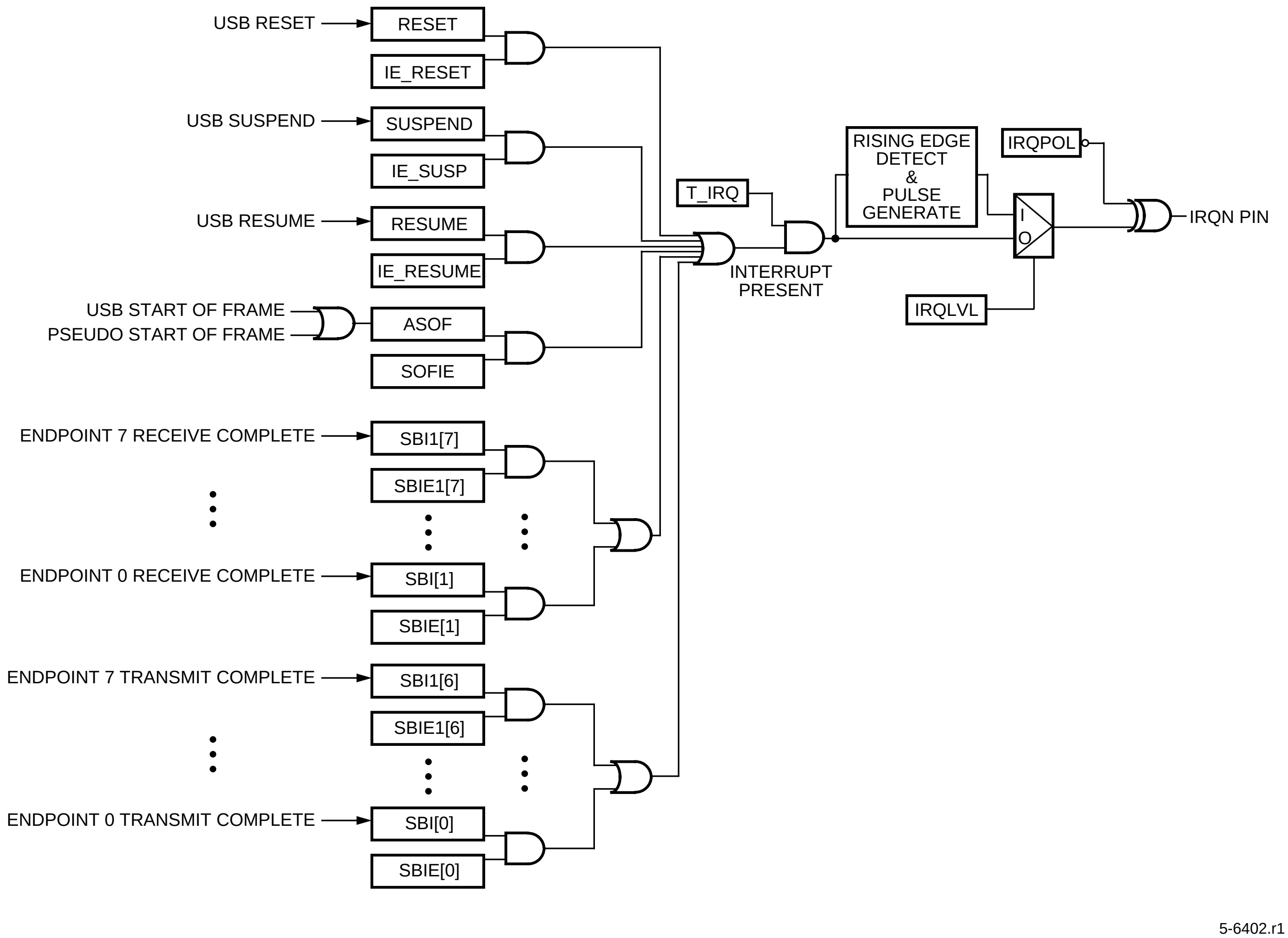


Figure 8. USS-820/USS-825 Interrupts

Firmware Responsibilities for USB SETUP Commands

All SETUP commands are passed through from the USB host to the corresponding receive FIFO (assuming no data transfer errors). Firmware must interpret and execute each command according to its USB definition.

Reception of a new SETUP command can be identified by the RXSETUP bit being set when a receive interrupt is generated. Any old data in the receive FIFO is overwritten by a new SETUP command. The STOVW register bit is set by hardware when a new SETUP packet is detected. When the complete SETUP packet has been written, hardware resets the STOVW bit and sets the EDOVW bit. If either the STOVW or EDOVW bit is set, the effect of any firmware actions on the FIFO pointers is blocked. This prevents the FIFO from underflowing as a result of firmware attempting to read the FIFO while hardware is writing a new setup packet. Firmware must reset the EDOVW bit, read the SETUP command from the FIFO, and then check the STOVW and EDOVW bits. If either is set, the SETUP that was just read out is old and should be discarded. Firmware must then proceed with reading the new SETUP command.

Firmware responsibilities for interpreting and executing USB standard commands are defined in Table 37.

Table 37. Firmware Responsibilities for USB SETUP Commands

USB Command	Firmware Responsibility
GET_STATUS	For device status, firmware should write two data bytes to transmit FIFO 0, where bit 0 of byte 0 indicates if the device is self-powered, and bit 1 indicates if the remote wake-up feature is supported (which should equal the value stored in the RWUPE register bit). For interface status, firmware should write two data bytes of zeros. For endpoint status, firmware should write two data bytes to transmit FIFO 0, where bit 0 of byte 0 is the RXSTL or TXSTL bit of the endpoint indicated by the SETUP command.
SET/CLEAR_FEATURE	For the DEVICE_REMOTE_WAKEUP feature, firmware should set/reset the RWUPE register bit. For the ENDPOINT_STALL feature, firmware should set/clear the RXSTL or TXSTL register bit indicated by the SETUP command. Firmware must also handle all side effects of these commands as documented in the USB specification, such as zeroing an endpoint's data toggle bit on CLEAR_FEATURE[stall].
SET_ADDRESS	Firmware should write the FADDR register with the device address indicated by the SETUP command. This write must not occur until after the status stage of the control transfer has completed successfully.
GET_CONFIGURATION, SET_CONFIGURATION, GET_INTERFACE, SET_INTERFACE	Firmware must maintain all information regarding which endpoints, interfaces, alternate settings, and configurations are supported and/or currently enabled. The enabled status of a particular endpoint direction, as specified by the current configuration, interface, and alternate setting, must be indicated in the corresponding RXEPEN or TXEPEN register bit. Firmware must also handle any side effects of these commands as documented in the USB specification, such as zeroing an endpoint's stall and data toggle bits on SET_INTERFACE or SET_CONFIGURATION.
GET_DESCRIPTOR, SET_DESCRIPTOR	Firmware must maintain all information regarding all types of descriptors and write the appropriate descriptor information to transmit FIFO 0 upon receiving GET_DESCRIPTOR, or read the appropriate descriptor information from receive FIFO 0 upon receiving SET_DESCRIPTOR.

Firmware Responsibilities for USB SETUP Commands (continued)

Firmware must keep track of the direction of data flow during a control transfer, and detect the start of the status stage by a change in that direction. For Control OUT transfers, the status stage will be an IN, and the firmware should write a zero-byte data packet to the transmit FIFO, assuming the command completed successfully. For control IN transfers, the status stage will be an OUT, and the firmware should read the data packet and set the RXFFRC register bit (like any other OUT transfer), again assuming the command completed successfully. This will cause an ACK to be sent to the host, indicating a successful completion.

Firmware should stall endpoint 0 if it receives a standard command that does not match any of the defined commands or a valid command that contains a parameter with a bad value (e.g., GET_STATUS[Endpoint “x”] when endpoint “x” is not enabled). Firmware should also stall if the data stage of a control transaction attempts to transfer more bytes than were indicated by the SETUP stage.

Firmware must interpret any vendor or class commands according to their definitions.

Other Firmware Responsibilities

Table 38. Other Firmware Responsibilities

USB Event	Firmware Responsibility
USB Reset	USB reset can be detected by reading a 1 from the RESET bit of the SSR register. If the USB interrupt is enabled (IE_RESET), this will be indicated by the IRQN output. At that time, firmware must reset any information it maintains regarding endpoints, interfaces, alternate settings, and configurations. All RXEPEN and TXEPEN endpoints should be set to 0, except for endpoint 0, which should be set to 1. The function address register FADDR should be set to 0. The data toggle bits for all endpoints should be set to 0 as well.
USB Suspend and Resume	Firmware must manage the SUSPEND and RESUME register bits, as documented in the following section, in order to meet the USB specifications for bus-powered devices.

Frame Timer Behavior

The USS-820/USS-825 contains an internal frame timer that allows the device to lock to the USB host frame timer, and to synthesize lost SOF packets, as required by the USB specification. The frame timer requires three valid SOF packets from the host in order to lock to the host frame timer. This locked status is indicated by the FTLOCK status bit in SOFH. In order to achieve this lock, the interval between each SOF must be within 45 clocks of the nominal 12,000 clocks, and the successive intervals must be within two clocks of each other. Both of these conditions will be true in a correctly functioning system with no bus errors. While the frame timer is locked, it will synthesize SOFs by setting ASOF, generating an SOF interrupt (if SOFIE = 1), and asserting the SOFN pin (if SOFODIS = 0) for up to three consecutive frames if SOF packets are no longer received from the host. The frame timer will become unlocked under any of the following conditions:

- Hard or soft reset.
- USB reset.
- The device goes suspended.
- No SOF packets are received from the host for three frames.
- An SOF is received that violates the USB specification for frame interval or previous frame length comparison.

Suspend and Resume Behavior

The USS-820/USS-825 detects a USB suspend condition if a J state is present on the bus for 3 ms, assuming the SUSPDIS register bit is 0. SUSPDIS should only be set for test purposes, never in a running system. When a suspend condition is detected, hardware sets the SUSPEND register status bit and, if it is enabled (IE_SUSP = 1), causes an interrupt.

When firmware detects that a suspend event has occurred, it must prepare itself, and any other system components for which it is responsible, for suspend mode. This will normally require turning off power to system components or placing them in low-power mode. When firmware is finished preparing for a device suspend, it should check the SUSPEND register status bit once more. If the status bit has reset, firmware should abort the suspend sequence, since the host has already awakened the device. If the status bit is still set, firmware should proceed with the suspend sequence. In this manner, the device will be guaranteed to see wake-up signaling of sufficient length from the host.

Suspend and Resume Behavior

(continued)

To suspend the device, firmware must set the SUSPEND register control bit to 1, and then reset the bit to 0. This action indicates to the USS-820/USS-825 that it can enter suspend mode. In order to guarantee correct behavior when resuming, the controller must not attempt any register reads until at least three tRDREC periods have elapsed since firmware resets the SUSPEND register control bit. Since firmware must have the PEND register bit set when modifying the SUSPEND bit, and since registers cannot be written while the device is suspended, firmware must remember to reset the PEND bit when the device resumes.

Since the SUSPEND register status bit will remain set while the device is suspended, a pending SUSPEND interrupt will remain until the device resumes. For this reason, firmware may wish to reset the IE_SUSP bit before suspending the device.

In order to meet the USB specification's power consumption limit for suspended, bus-powered devices, the USS-820/USS-825 must turn off its internal clocks. This occurs when the SUSPEND register control bit is reset by firmware as described above and is indicated by the hardware that asserts the SUSPN output pin. The system can be set up such that the SUSPN output controls power to other system components. While in suspend mode, the device's power consumption will be reduced to almost zero and will remain in this state until a wake-up is signaled. Wake-up can be signaled by either the host or the application. A host-signaled wake-up (global resume) is indicated when the host drives a K state on the USB bus. A remote wake-up is indicated by the application by asserting the RWUPN input pin. In either case, the device will initiate a wake-up sequence.

The USS-820/USS-825 starts a wake-up sequence by re-enabling its internal oscillator and PLL. Once the internally generated clocks are stable, it enables clocks to the entire chip, deasserts the SUSPN output, and sets the RESUME register bit, which causes an interrupt if IE_SUSP is set. The USS-820/USS-825 will require a maximum of 15 ms to resume functionality after a wake-up sequence is initiated. If the wake-up was a remote wake-up, the device drives a wake-up signal (K) on the USB bus for 12 ms. The deassertion of the SUSPN output indicates to the application that it should restore itself and other system components under its control to normal operating mode. When this action is complete, firmware must reset the RESUME register bit to allow detection of any subsequent suspend events.

In order to meet the power requirements while suspended, care must be exercised to guarantee that all board signals connected to the USS-820/USS-825 are at their proper state. Voltages on these pins must be guaranteed to be outside the switching threshold region (i.e., either a valid CMOS logic 1 or 0). The proper state will vary, depending on whether or not the application is powered off.

If the application is powered off, it is assumed that its pin buffers will provide the signals with a path to ground. The USS-820/USS-825 outputs will be 3-stated to avoid current flow in the application. USS-820/USS-825 input pins and bidirectionals (which will be 3-stated) will all be pulled to ground by the powered off application. Any external pull-ups that are connected to a powered off device must have their power removed during suspend, to avoid excessive power consumption. If an external oscillator is used, it will most likely need to be turned off by the SUSPN output in order to meet the suspend power requirement. When the oscillator is turned back on after a resume, while it is stabilizing, its output clock must not have a frequency greater than 12 MHz. Furthermore, during this stabilization period, the oscillator output must not provide more than 84,000 clocks.

Depending on the system design, the register interface signals (RDN, WRN, IOCSN) could have unknown values immediately after a suspend because of external components being powered off. In this case, firmware must configure the USS-820/USS-825 to enable the locking mechanism by setting the SUSPLOE bit. This mechanism protects the internal registers from being corrupted in this situation. Its behavior is documented in *Special Action Required by USS-820 After Suspend* Application Note (AP97-058CMPR).

While the USS-820/USS-825 is suspended, its internal registers may still be read. The interface timing for such reads is different from register reads during operational mode, and is specified in the Register Timing Characteristics section. Register writes must not be attempted while the device is suspended. Certain register reads during the nonsuspended state can cause other device register states to change. For example, a read of RXDAT can cause changes in the receive FIFO read pointer and related FIFO flags. These register reads must not be attempted while the device is suspended.

If the application's power is left on, then USS-820/USS-825 outputs and inputs continue to be driven by the USS-820/USS-825 and the application, respectively. In addition, the USS-820/USS-825 bidirectional pins are 3-stated in the USS-820/USS-825 and driven to 0 or 1 by the application.

Suspend and Resume Behavior (continued)

Table 39 summarizes the proper values on all USS-820/USS-825 pins during suspend. External logic refers to logic outside of the USS-820/USS-825, and most likely not direct outputs of the application. The DPLS and DMNS pins remain powered and logically connected to the USB host. If a crystal is connected to XTAL1 and XTAL2, power-down mode is managed by the internal logic. If an external oscillator is connected to XTAL1, the input must be a stable logic 1 during suspend mode. The “0/1 (1)” notation indicates that a logic 0 or 1 may be driven, but a 1 will occur naturally. In systems where devices external to the USS-820/USS-825 are powered off during suspend, any external pull-ups that are connected to those devices must also be powered off to avoid excessive current flow. Such systems must also supply a valid CMOS logic 0 or 1 to the USS-820/USS-825 inputs and 3-stated outputs during suspend. This may occur naturally by the powered off application providing a path to ground, which will appear as a logic 0 to the USS-820/USS-825.

Table 39. Proper Values During Suspend

USS-820/USS-825			
Pin	**Mode**	**If Application Powered On**	**If Application Powered Off**
A[4:0]	3-state	Application drives 0/1	Powered off—path to ground—0
D[7:0]	Any/3-state	Depends on state of RDN, IOCSN pins	Powered off—path to ground—0
RWUPN	In	External logic drives 1	External logic drives 1
SUSPN	Out	USS-820/USS-825 drives 0	USS-820/USS-825 drives 0
IRQN	Out/3-state	USS-820/USS-825 drives 0/1 (1)	Powered off—path to ground—0
SOFN	Out/3-state	USS-820/USS-825 drives 0/1 (1)	Powered off—path to ground—0
RESET	In	External logic drives 0	External logic drives 0
IOCSN	In	Application drives 0/1 (1)	Powered off—path to ground—0
RDN	In	Application drives 0/1	Powered off—path to ground—0
WRN	In	Application drives 1	Powered off—path to ground—0
XTAL1	In	External logic drives 1 (if oscillator)	External logic drives 1 (if oscillator)

Application Notes

1. The RESET input must remain asserted for a minimum period of time after power is stable. If internal oscillator clocking mode is used, this time is t_{OSC}, the amount of time required to allow the internal oscillator output to become stable. If external oscillator clocking mode is used, this time is t_{RST}. The USS-820/USS-825 WRN and RWUPN pins must not both be active (low) at the time that the RESET input is deasserted.
2. After changing the size (RXFFSZ/TXFFSZ), type (isochronous vs. nonisochronous), or enabled status (RXEPEN/TXEPEN) of a FIFO/endpoint, firmware must guarantee that at least 16 t_{CLK} periods have elapsed before attempting to access the FIFO data. This is required to allow the internal FIFO RAM to be reallocated.
3. Register writes are triggered by the rising edge of either WRN or IOCSN, whichever comes first, and are synchronized to the internal 12 MHz clock. Therefore, the actual write may not occur until as much as t_{CLK} ns after that first rising edge. This latency must be taken into account when performing subsequent register reads or writes.
4. If the SOFN pin is not used, it must be connected to an external pull-up or pull-down. The pin is actually bidirectional, where the input mode is only used in chip test modes. The pull-up or pull-down is required to avoid excessive power consumption by the input stage.
5. The IRQN and SOFN pins require external pull-ups or pull-downs if the external controller will be powered off during suspend. In that situation, those pins will be 3-stated until the USS-820/USS-825 has fully resumed. The pull-up or pull-down is needed to establish the desired level at the controller for the time interval from when the controller is powered on to the time when the USS-820/USS-825 has completed the resume.

Absolute Maximum Ratings

Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. These are absolute stress ratings only. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operations sections of this data sheet. Exposure to absolute maximum ratings for extended periods can adversely affect device reliability.

Table 40. Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
Ambient Operating Temperature Range	T_A	0	70	°C
Storage Temperature	T_{stg}	–40	125	°C
Voltage on Any Pin with Respect to Ground	—	V_{SS} – 0.3	V_{DD} + 0.3*	V
Power Supply Voltage with Respect to Ground	V_{DD}	—	4.6	V

* Except for 5 V tolerant buffers where V_{IN} max = V_{DD5} max + 0.5 V.

Electrical Characteristics

dc Characteristics

Table 41. dc Characteristics (T_A = 0 °C to 70 °C, V_{DD} = 3.3 V ± 0.3 V, V_{SS} = 0 V)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
USB Signals						
High-Z State Data Line Leakage	I_{LO}	0 V < V_{IN} < 3.3 V	–10	—	10	µA
Differential Receiver:						
Common-mode Range	CMR	—	0	—	V_{DD}	V
Sensitivity	V_{DI}	CMR = 0.8 V to 2.5 V	0.2	—	—	V
Single-ended Receiver:						
Low	V_{IL}	—	—	—	0.8	V
High	V_{IH}	—	2.0	—	—	V
Hysteresis	—	—	0.3	—	—	V
Output Voltage:						
Low	V_{OL}	R_L of 1.5 kΩ to 3.6 V	—	—	0.3	V
High	V_{OH}	R_L of 15 kΩ to GND	2.8	—	3.6	V
Other Signals						
Transceiver Capacitance	CIN	Pin to GND	—	—	20	pF
Input Voltage:						
Low	V_{IL}	—	—	—	0.8	V
High	V_{IH}	—	2.0	—	—	V
Output Voltage (SUSPN, IRQN, DPPU [48-pin]):						
Low	V_{OL}	I_{OL} = 6 mA	—	—	0.4	V
High	V_{OH}	I_{OL} = –6 mA	2.4	—	—	V
High	V_{OH}	I_{OL} = –1 mA	V_{DD} – 0.15	—	—	V
Output Voltage (D[7:0], SOFN, DPPU [44-pin]):						
Low	V_{OL}	I_{OL} = 10 mA	—	—	0.4	V
High	V_{OH}	I_{OL} = –10 mA	2.4	—	—	V
High	V_{OH}	I_{OL} = –1 mA	V_{DD} – 0.1	—	—	V

Electrical Characteristics (continued)

Table 41. dc Characteristics (T_A = 0 °C to 70 °C, V_{DD} = 3.3 V ± 0.3 V, V_{SS} = 0 V) (continued)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Device						
Power Dissipation:						
Configured	P_D	25 °C, V_{DD} = 3.6 V	—	—	260	mW
Preconfigured	P_{DP}	25 °C, V_{DD} = 3.6 V	—	—	180	mW
Suspended	P_{DS}	25 °C, V_{DD} = 3.6 V	—	—	TBD	µW
Power Supply Voltage	V_{DD}, V_{DDA}, V_{DDT}	—	3	3.3	3.6	V
	V_{DD5}	5 V environment	4.75	5	5.25	V
		3 V environment	3	3.3	3.6	V

Power Considerations

The USB specification places power consumption limits on bus-powered devices. The limit is tighter for a device that has not yet been configured. The tightest limit is for a suspended device.

The power consumption numbers listed in Table 41 for a preconfigured device make several assumptions, based on the fact that the device has not yet been configured. The calculation still assumes a fairly active application interface. The actual power consumption can be reduced by limiting the frequency of register reading.

The limit for suspended devices can only be met if careful measures are taken to control the interface to the USS-820/USS-825, as documented in the Suspend and Resume Behavior section.

One factor that significantly increases the power consumption is the capacitive load on the data pins. All specifications are based on a load of 100 pF on the data pins, unless noted otherwise. Minimizing this capacitance can cause a significant reduction in total power consumption. The capacitive load has no effect on the power consumption in suspend mode.

Another factor that adds to the overall power consumption is the type of load that the outputs drive. Connection to a non-CMOS load (TTL) will add a dc power consumption component to the overall chip power. The suspend mode power limit can only be met in this situation if the attached device is powered off.

All power dissipation numbers assume a maximum 50 pF load on the D pins, and CMOS loads on all outputs. For 100 pF loads on the D pins, add 62 mW to the configured and preconfigured power dissipation. For TTL loads, add 34 mW.

USB Transceiver Driver Characteristics

Table 42. USB Transceiver Driver Characteristics

Parameter	Symbol	Test Conditions	Min	Max	Unit
Rise and Fall Times:					
(10%—90%)	t_R	OEN = 0, C_L = 50 pF	4	20	ns
(90%—10%)	t_F	OEN = 0, C_L = 50 pF	4	20	ns
Rise/Fall Time Matching	t_{RFM}	OEN = 0, C_L = 50 pF	90	110	%
Crossover Point	V_{CRS}	OEN = 0, C_L = 50 pF	1.3	2.0	V
Output Impedance*	Z_{DRV}	OEN = 0	28	43	Ω

* At steady-state drive, when used with an external series resistor of 24 Ω.

Electrical Characteristics (continued)

Connection Requirements

USB Transceiver Connection

The physical connection of the USS-820/USS-825 to the USB bus requires only minimal components to provide proper USB electrical terminations.

Both DPLS and DMNS require 24 Ω ± 1% series resistors for USB impedance matching. Additionally, a 1.5 kΩ pull-up resistor is required on DPLS for full-speed/low-speed differentiation.

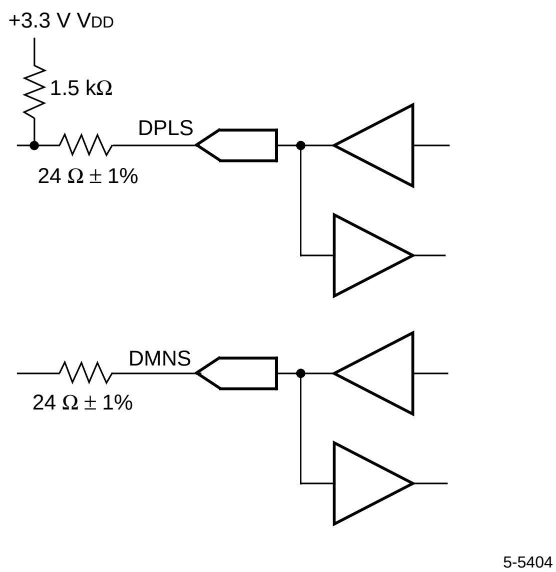


Figure 9. USB Transceiver Connection (Normal Mode)

For using the USS-820/USS-825 in a self-powered device, there are some additional considerations. The device must refrain from supplying power through the pull-up resistor if plugged into an unpowered bus. It must also ensure that the DPLS and DMNS lines are in an appropriate state when the device is powered but not plugged in. Figure 10 shows an example connection to meet these requirements.

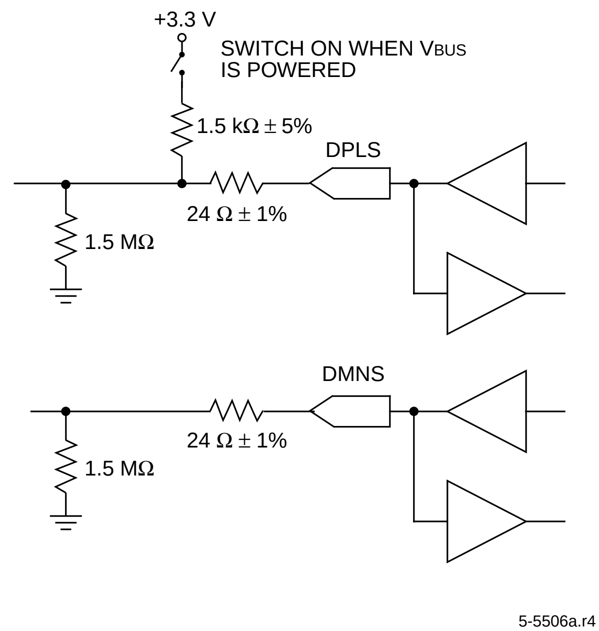


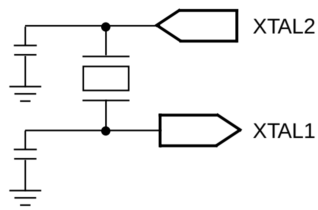
Figure 10. Self-Powered Device Example Connection

Electrical Characteristics (continued)

Oscillator Connection Requirements

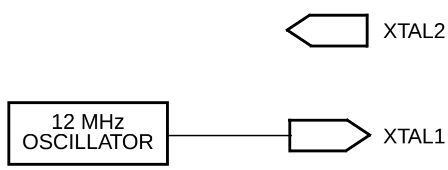
The USS-820/USS-825 requires an internal 48 MHz clock that it creates from an internal 12 MHz clock via a 4X PLL. Two methods of clock generation may be used to create this internal 12 MHz clock. Figure 11 shows the internal oscillator mode which requires only an external 12 MHz crystal and bias capacitors. The values of the capacitors should be chosen as indicated by the crystal manufacturer in order to cause the crystal to operate in a parallel resonant condition. A typical value is 15 pF, but the required value may differ, depending on the specific crystal and board characteristics of the application.

Alternatively, Figure 12 shows the configuration required to input a 12 MHz clock from an external oscillator. In either configuration, the external clock source must have the characteristics defined in Table 43.



5-5405.a.r1

Figure 11. Internal Oscillator Mode



5-5406.a.r1

Figure 12. External Oscillator Source

Table 43. Clock Characteristics

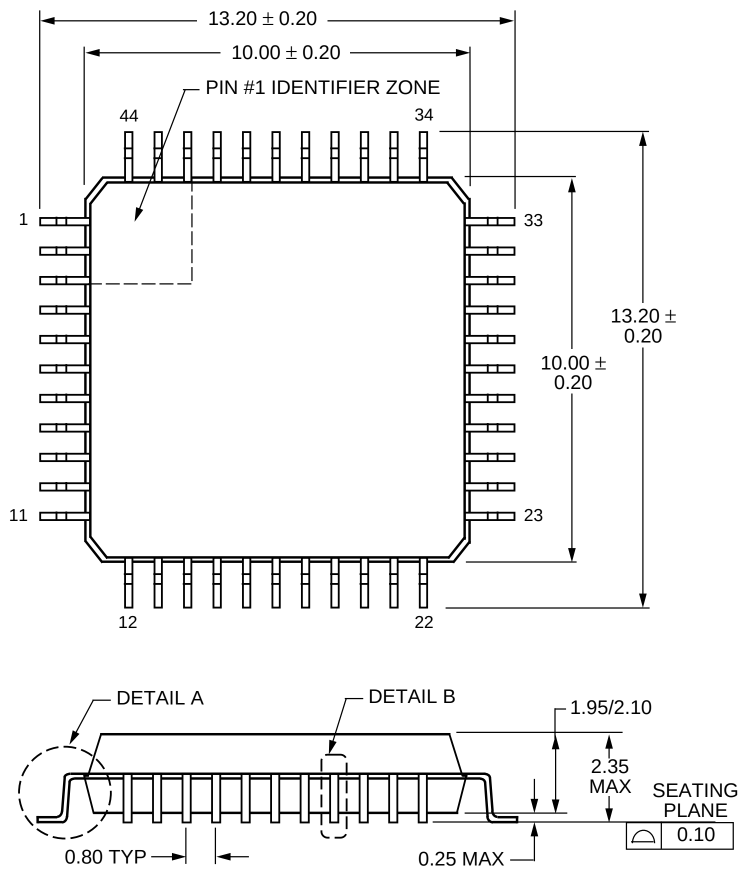
Parameter	Symbol	Min	Typ	Max	Unit
External Clock Source Frequency	f	11.976	12.000	12.024	MHz
Clock Period	t_{CYC}	83.1	83.3	83.5	ns
Clock Duty Cycle*	t_{CL}, t_{CH}	40	50	60	%
Oscillator Stable Time	t_{OSC}	—	—	7	ms

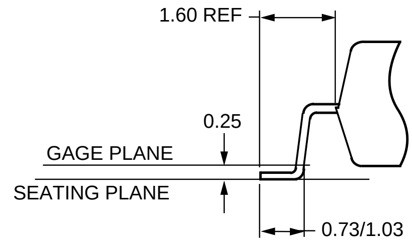
* Duty cycle applies to any frequency in an specified range.

Outline Diagrams

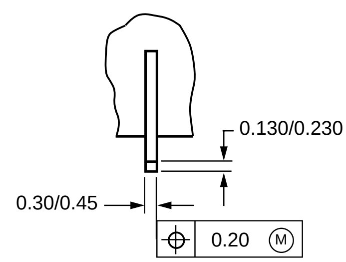
44-Pin MQFP (USS-820)

Dimensions are in millimeters.





DETAIL A



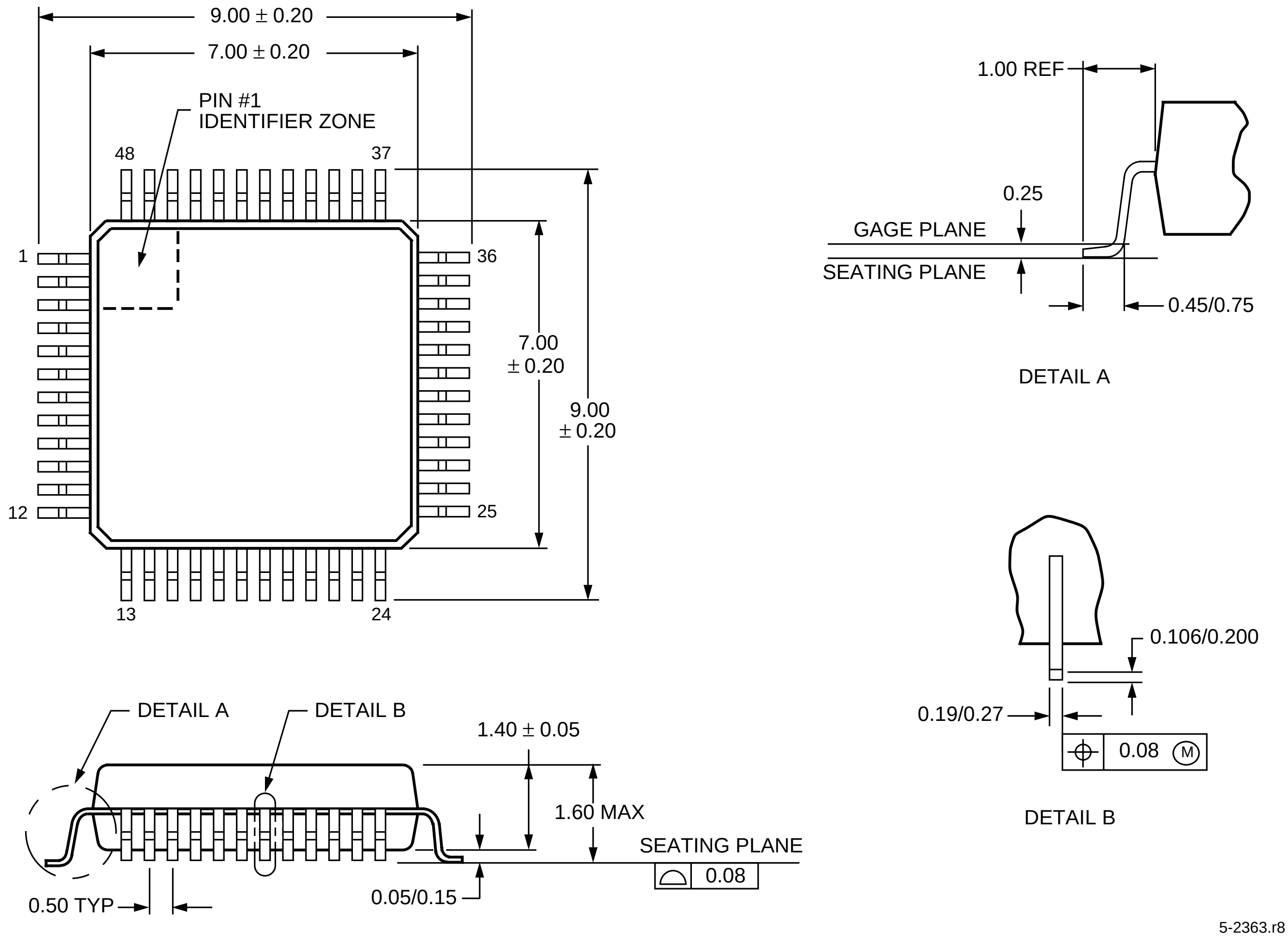
DETAIL B

5-2111.r11

Outline Diagrams (continued)

48-Pin TQFP (USS-825)

Dimensions are in millimeters.



Ordering Information

Device Code	Package	Comcode
USS820B-DB	44-Pin MQFP	108159716
USS825B-DB	48-Pin TQFP	108158593

Appendix A. Special Function Register Bit Names

Table 44. Alphabetical Listing of Special Function Register Bit Names

Bit Name	Register	Table	Page	Bit Name	Register	Table	Page
A[6:0]	FADDR	21	24	RXFIF[1:0]	RXFLG	29	30
ADVRM	TXCON	24	25	RXFULL	RXFLG	29	30
ADVWM	RXCON	28	29	RXIE	EPCON	18	20
ARM	RXCON	28	29	RXISO	RXCON	28	29
ASOF	SOFH	15	18	RXOVF	RXFLG	29	31
ATM	TXCON	24	25	RXSEQ	RXSTAT	20	22
BC[7:0]	RXCNTL	27	28	RXSETUP	RXSTAT	20	22
BC[7:0]	TXCNTL	23	24	RXSOVW	RXSTAT	20	22
BC[9:8]	RXCNTH	27	28	RXSPM	EPCON	18	20
BC[9:8]	TXCNTH	23	24	RXSTL	EPCON	18	20
CTLEP	EPCON	18	20	RXURF	RXFLG	29	31
DPEN	MCSR	36	35	RXVOID	RXSTAT	20	23
EDOVW	RXSTAT	20	22	SCRATCH	SCRATCH	35	34
EPINX[2:0]	EPINDEX	17	19	SOFACK	SOFH	15	18
FFSZ[1:0]	RXCON	28	28	SOFIE	SOFH	15	18
FFSZ[1:0]	TXCON	24	25	SOFODIS	SOFH	15	18
FRXD[3:0]	SBI	13	16	SRESET	SCR	30	32
FRXD[7:4]	SBI1	14	17	STOVW	RXSTAT	20	22
FRXIE[3:0]	SBIE	11	15	SUSPDIS	SSR	31	33
FRXIE[7:4]	SBIE1	12	15	SUSPEND	SSR	31	33
FTLOCK	SOFH	15	18	SUSPLOE	MCSR	36	35
FTXD[3:0]	SBI	13	16	SUSPPO	SSR	31	33
FTXD[7:4]	SBI1	14	17	SUSPS	MCSR	36	35
FTXIE[3:0]	SBIE	11	15	T_IRQ	SCR	30	32
FTXIE[7:4]	SBIE1	12	15	TS[10:8]	SOFH	15	18
IE_RESET	SCR	30	32	TS[7:0]	SOFL	16	19
IE_RESUME	SCRATCH	35	34	TXACK	TXSTAT	19	21
IE_SUSP	SCR	30	32	TXCLR	TXCON	24	25
INIT	MCSR	36	35	TXDAT[7:0]	TXDAT	22	24
IRQLVL	SCR	30	32	TXEMP	TXFLG	25	26
IRQPOL	SCR	30	32	TXEPEN	EPCON	18	20
PEND	PEND	34	34	TXERR	TXSTAT	19	21
PKGID	MCSR	36	35	TXFIF[1:0]	TXFLG	25	26
RESET	SSR	31	33	TXFLUSH	TXSTAT	19	21
RESUME	SSR	31	33	TXFULL	TXFLG	25	27
REVRP	TXCON	24	25	TXISO	TXCON	24	25
REVWP	RXCON	28	29	TXOE	EPCON	18	20
RWUPE	SCR	30	32	TXOVF	TXFLG	25	26
RXACK	RXSTAT	20	23	TXSEQ	TXSTAT	19	21
RXCLR	RXCON	28	28	TXSOVW	TXSTAT	19	21
RXDAT[7:0]	RXDAT	26	27	TXSTL	EPCON	18	20
RXEMP	RXFLG	29	31	TXURF	TXFLG	25	26
RXEPEN	EPCON	18	20	TXVOID	TXSTAT	19	21
RXERR	RXSTAT	20	23	UNLOCKED	LOCK	33	34
RXFFRC	RXCON	28	28				

Appendix B. USS-820 Register Map

Table 45. USS-820 Register Map

Register	USS-820 Register Map								Addr
TXDAT	TXDAT[7:0]								00*
TXCNTL	BC[7:0]								01*
TXCNTH	—						BC [9:8]		02*
TXCON	TXCLR	TXFFSZ[1:0]		—	TXISO	ATM	ADVRM	REVRP	03*
TXFLG	TXFIF[1:0]		—		TXEMP	TXFULL	TXURF	TXOVF	04*
RXDAT	RXDAT[7:0]								05*
RXCNTL	BC[7:0]								06*
RXCNTH	—						BC[9:8]		07*
RXCON	RXCLR	RXFFSZ [1:0]		RXFFRC	RXISO	ARM	ADVWM	REVWP	08*
RXFLG	RXFIF[1:0]		—		RXEMP	RXFULL	RXURF	RXOVF	09*
EPINDEX	—						EPINX[2:0]		0A
EPCON	RXSTL	TXSTL	CTLEP	RXSPM	RXIE	RXEPEN	TXOE	TXEPEN	0B*
TXSTAT	TXSEQ	—		TXFLUSH	TXSOVW	TXVOID	TXERR	TXACK	0C*
RXSTAT	RXSEQ	RXSETUP	STOVW	EDOVW	RXSOVW	RXVOID	RXERR	RXACK	0D*
SOFL	TS[7:0]								0E
SOFH	SOFACK	ASOF	SOFIE	FTLOCK	SOFODIS	TS[10:8]			0F
FADDR	—	A[6:0]							10
SCR	IRQPOL	RWUPE	IE_SUSP	IE_RESET	SRESET	IRQLVL	T_IRQ	—	11
SSR	—			SUSPPO	SUSPDIS	RESUME	SUSPEND	RESET	12
SBI	FRX03	FTX03	FRX02	FTX02	FRX01	FTX01	FRX00	FTX00	14
SBI1	FRX07	FTX07	FRX06	FTX06	FRX05	FTX05	FRX04	FTX04	15
SBIE	FRXIE3	FTXIE3	FRXIE2	FTXIE2	FRXIE1	FTXIE1	FRXIE0	FTXIE0	16
SBIE1	FRXIE7	FTXIE7	FRXIE6	FTXIE6	FRXIE5	FTXIE5	FRXIE4	FTXIE4	17
REV	REV[7:0]								18
LOCK	—							UNLOCKED	19
PEND	—							PEND	1A
SCRATCH	IE_RESUME	SCRATCH[6:0]							1B
MCSR	—	INIT	SUSPS	PKGID	—		SUSPLOE	DPEN	1C

* Indexed by EPINDEX.

For additional information, contact your Microelectronics Group Account Manager or the following:
INTERNET: **http://www.lucent.com/micro**
E-MAIL: **docmaster@micro.lucent.com**
N. AMERICA: Microelectronics Group, Lucent Technologies Inc., 555 Union Boulevard, Room 30L-15P-BA, Allentown, PA 18103
1-800-372-2447, FAX 610-712-4106 (In CANADA: **1-800-553-2448**, FAX 610-712-4106)
ASIA PACIFIC: Microelectronics Group, Lucent Technologies Singapore Pte. Ltd., 77 Science Park Drive, #03-18 Cintech III, Singapore 118256
Tel. (65) 778 8833, FAX (65) 777 7495
CHINA: Microelectronics Group, Lucent Technologies (China) Co., Ltd., A-F2, 23/F, Zao Fong Universe Building, 1800 Zhong Shan Xi Road, Shanghai 200233 P. R. China **Tel. (86) 21 6440 0468**, **ext. 316**, FAX (86) 21 6440 0652
JAPAN: Microelectronics Group, Lucent Technologies Japan Ltd., 7-18, Higashi-Gotanda 2-chome, Shinagawa-ku, Tokyo 141, Japan
Tel. (81) 3 5421 1600, FAX (81) 3 5421 1700
EUROPE: Data Requests: MICROELECTRONICS GROUP DATALINE: **Tel. (44) 1189 324 299**, FAX (44) 1189 328 148
Technical Inquiries: GERMANY: **(49) 89 95086 0** (Munich), UNITED KINGDOM: **(44) 1344 865 900** (Ascot),
FRANCE: **(33) 1 40 83 68 00** (Paris), SWEDEN: **(46) 8 594 607 00** (Stockholm), FINLAND: **(358) 9 4354 2800** (Helsinki),
ITALY: **(39) 02 6608131** (Milan), SPAIN: **(34) 1 807 1441** (Madrid)

microelectronics group

April 1999
DS99-069CMPR-03 (Replaces DS99-069CMPR-02)

Lucent Technologies
Bell Labs Innovations