Features

- **AVR® 8-bit RISC Microcontroller with 83 ns Instruction Cycle Time**
- **5-port USB Hub with Embedded Function Controller**
- **Up to 64K Bytes of External Program Memory**
- **512 Bytes of SRAM**
- **32 x 8 General-purpose Working Registers**
- **32 Programmable I/O Port Pins**
- **SPI Serial Interface**
- **Programmable Serial UART**
- **One 8-bit Timer/Counter with Separate Prescaler**
- **One 16-bit Timer/Counter with Separate Prescaler**
- **Input Capture and PWM**
- **External and Internal Interrupt Sources**
- **Programmable Watchdog Timer**
- **Integrated Hub and Function Controllers**
- **Programmable 6/12 MHz Oscillator with On-chip PLL**
- **5V Operation with On-chip 3.3V Power Supply**
- **100-lead LQFP Package**

USB Hub and Embedded Function Controller

AT43USB320A

Description

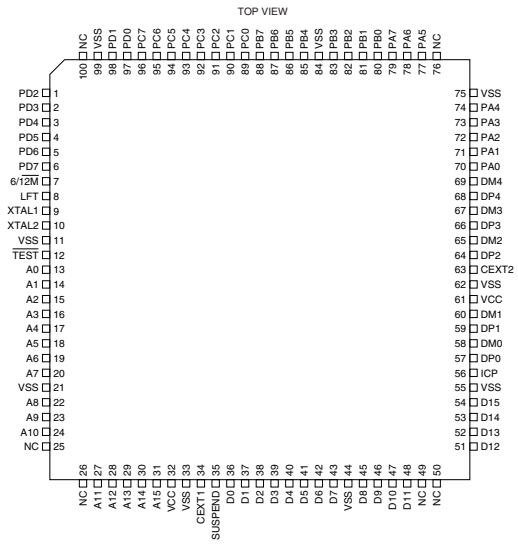
The Atmel AT43USB320A is a USB microcontroller based on the Atmel AVR. The USB module of the AT43USB320A is a compound device, consisting of a 5-port hub and an attached embedded function.

Internally the AT43USB320A consists of a USB hub and function interface, a hub repeater, and an AVR microcontroller with a dedicated external program memory bus. To the USB host, the embedded function appears as an attached port of the hub with its own device address and endpoints. The hub of the AT43USB320A can be disabled, in which case the chip becomes a USB function controller with four endpoints.

(continued)

Pin Configuration

100-lead LQFP



Rev. 1443B–01/01

The USB hardware block consists of a USB transceiver, SIE, hub repeater, endpoint controllers and an interface to the microcontroller. The processor of the microcontroller is the Atmel AVR Enhanced RISC core. The USB hardware interfaces to the USB host at the transaction layer while the microcontroller firmware handles the USB protocol layers above and also handles the peripheral control functions.

Application Example

Figure 1. AT43USB320A as a USB Hub and Embedded Function

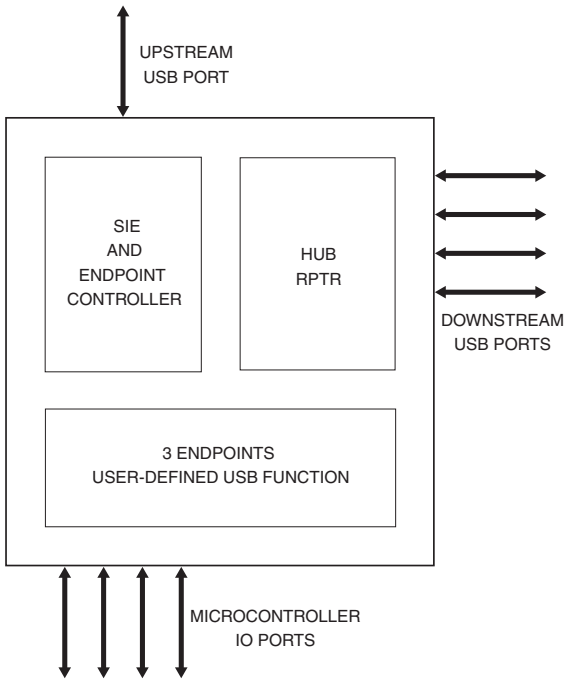


Figure 2. AT43USB320A as a USB Function

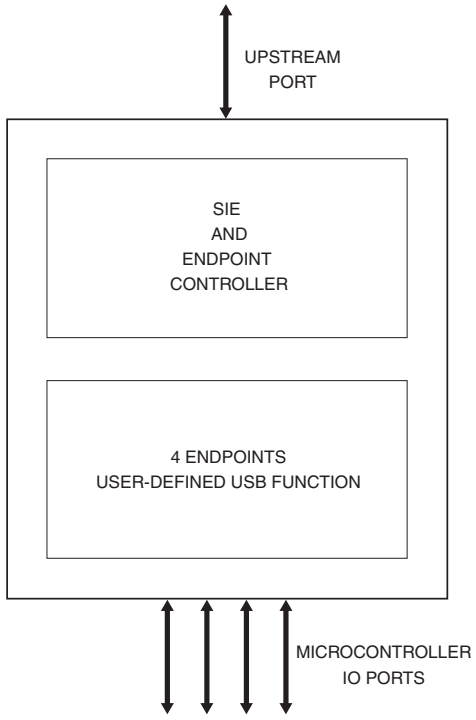


Figure 3. Internal Block Diagram (1)

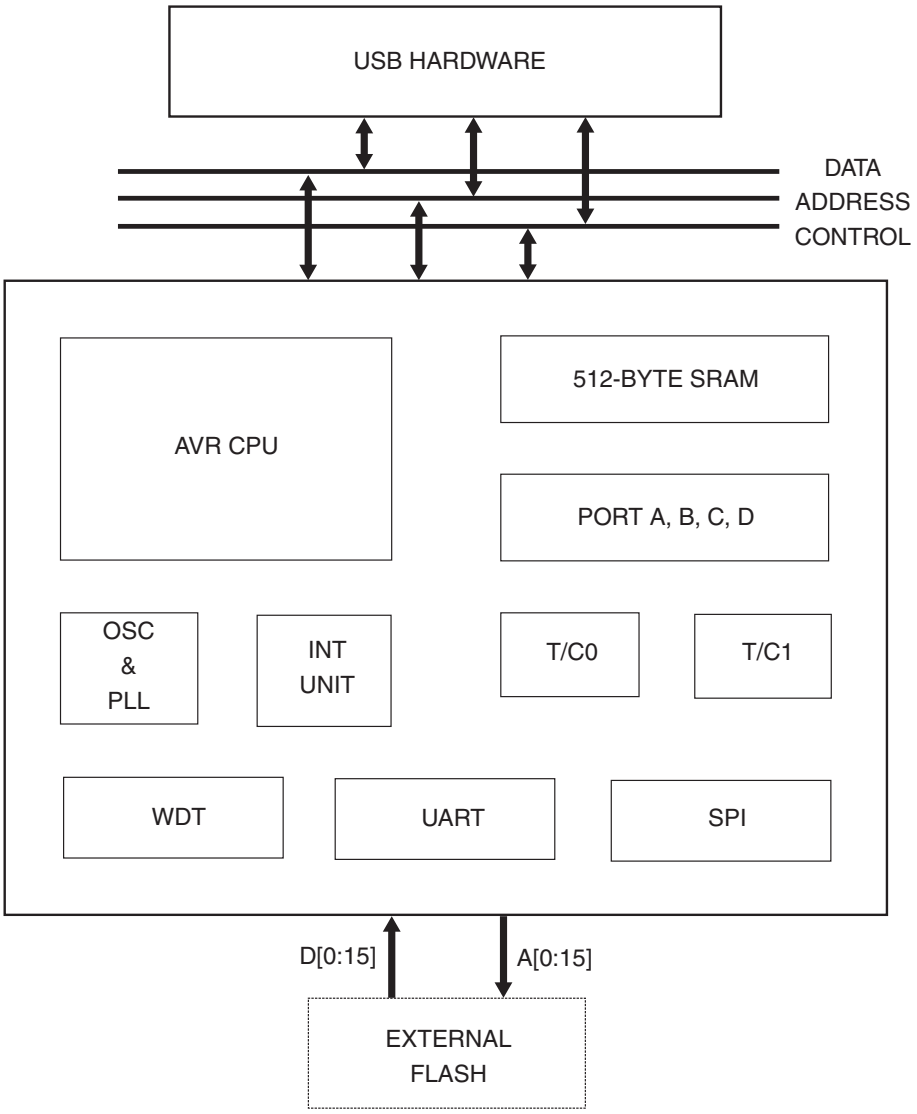
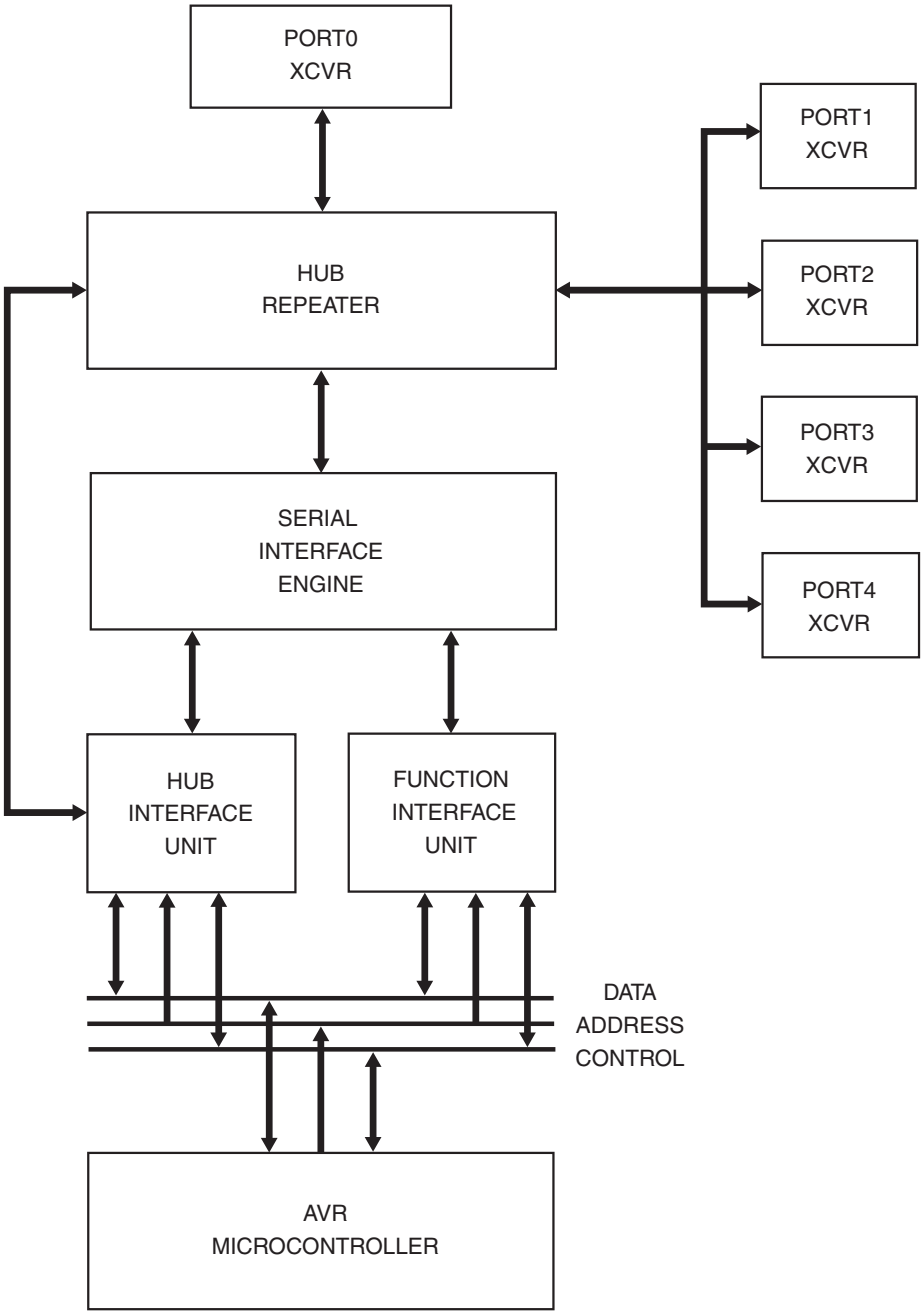


Figure 4. Internal Block Diagram (2)



Pin Assignment

Type:

I = Input
O = Output
OD = Output, open drain
B = Bi-directional
V = Power supply, ground

Table 1. Pin Assignment in Numerical Order

Pin #	Signal	Type
1	PD2	B
2	PD3	B
3	PD4	B
4	PD5	B
5	PD6	B
6	PD7	B
7	6/$\overline{12M}$	I
8	LFT	O
9	XTAL1	I
10	XTAL2	O
11	VSS	V
12	$\overline{TEST}$	I
13	A0	B
14	A1	B
15	A2	B
16	A3	B
17	A4	B
18	A5	B
19	A6	B
20	A7	B
21	VSS	V
22	A8	B
23	A9	B
24	A10	B
25	NC	
26	NC	
27	A11	B
28	A12	B
29	A13	B
30	A14	B
31	A15	B
32	VCC	V
33	VSS	V

Pin #	Signal	Type
34	CEXT1	O
35	$\overline{SUSPEND}$	O
36	D0	I
37	D1	I
38	D2	I
39	D3	I
40	D4	I
41	D5	I
42	D6	I
43	D7	I
44	VSS	V
45	D8	I
46	D9	I
47	D10	I
48	D11	I
49	NC	
50	NC	
51	D12	I
52	D13	I
53	D14	I
54	D15	I
55	VSS	V
56	ICP	V
57	DP0	B
58	DM0	B
59	DP1	B
60	DM1	B
61	VCC	V
62	VSS	V
63	CEXT2	O
64	DP2	B
65	DM2	B
66	DP3	B

Pin #	Signal	Type
67	DM3	B
68	DP4	B
69	DM4	B
70	PA0	B
71	PA1	B
72	PA2	B
73	PA3	B
74	PA4	B
75	VSS	V
76	NC	
77	PA5	B
78	PA6	B
79	PA7	B
80	PB0	B
81	PB1	B
82	PB2	B
83	PB3	B
84	VSS	V
85	PB4	B
86	PB5	B
87	PB6	B
88	PB7	B
89	PC0	B
90	PC1	B
91	PC2	B
92	PC3	B
93	PC4	B
94	PC5	B
95	PC6	B
96	PC7	B
97	PD0	B
98	PD1	B
99	VSS	V
100	NC	

Signal Description

Signal	Type	Name and Function
6/$\overline{12M}$	I	**Crystal Select** – Selects 6 MHz or 12 MHz crystal operation. If this pin is strapped high, the crystal is 6 MHz. If strapped low, 12 MHz. The internal clock for the microcontroller is always running at 12 MHz.
A[0:15]	O	**Address[0:15]** – These are the address pins for the external program memory Flash.
D[0:15]	I	**Data[0:15]** – These are the program memory data input pins.
DP0	B	**Upstream Plus USB I/O** – This pin should be connected to CEXT2 through an external 1.5 kΩ pull-up resistor. DP0 and DM0 form the differential signal pin pairs connected to the host controller or an upstream hub.
DM0	B	**Upstream Minus USB I/O**
DP[1:4]	B	**Port Plus USB I/O** – Each of these pins must be connected to VSS through an external 15 kΩ resistor. DP[1:4] and DM[1:4] are the differential signal pin pairs to connect downstream USB devices.
DM[1:4]	B	**Port Minus USB I/O** – Each of these pins should be connected to VSS through an external 15 kΩ resistor.
VSS	V	**Ground**
ICP	I	**Input Capture** – Input signal for the Timer/Counter1 input capture function
LFT	I	**PLL Filter** – For proper operation of the PLL, this pin should be connected through a 2.2 nF capacitor in parallel with a 100Ω resistor in series with a 10 nF capacitor to ground (VSS).
PA[0:7]	B	**Port A** – This is an 8-bit bi-directional IO port with 4 mA drive strength.
PB[0:7]	B	**Port B** – This is an 8-bit bi-directional IO port with 4 mA drive strength. PB[0,1,4:8] have dual functions as shown in the following table: **Port Pin** — **Alternate Function** PB0 — T0, timer/counter0 external input PB1 — T1, timer/counter1 external input PB4 — $\overline{SS}$, SPI slave port select or SCL, I^2C serial bus clock PB5 — MOSI, SPI slave port select input PB6 — MISO, SPI master data in, slave data out PB7 — SCK, SPI master clock out, slave clock in
PC[0:7]	B	**Port C** – This is an 8-bit bi-directional IO port with 4 mA drive strength.
PD[0:7]	B	**Port D** – This is an 8-bit bi-directional IO port with 4 mA drive strength. PD0 through PD3 also serve as the signal pins for the serial port and external interrupts as listed below: **Port Pin** — **Alternate Function** PD0 — RXD, serial input port PD1 — TXD, serial output port PD2 — INT0, external interrupt 0 PD3 — INT1, external interrupt 1
$\overline{TEST}$	I	**Test Input** – If $\overline{TEST}$ is low, the AT43USB320A enters the test mode. In normal operation, this pin should be strapped high.
VCC	V	**5V Power Supply**
CEXT1, 2	I	**External Capacitor** – For proper operation of the on-chip regulator, a 0.27 µF capacitor must be connected to these pins. The CEXT1 and CEXT2 pins cannot be connected to each other.
XTAL1	I	**Oscillator Input** – Input to the inverting oscillator amplifier
XTAL2	O	**Oscillator Output** – Output of the inverting oscillator amplifier
$\overline{SUSPEND}$	O	**Suspend** – This signal is asserted when the AT43USB320A enters the suspend mode and will be de-asserted upon waking up. This pin is used to put the external program memory in the low current stage if desired.

Functional Description

USB Hardware Block

USB Interface

The USB Interface consists of a Serial Interface Engine, a Serial Bus Controller and a System Interface. The SIE performs the clock/data separation, NRZI encoding and decoding, bit insertion and deletion, CRC generation and checking, and the serial-parallel data conversion. The Serial Bus Controller consists of a protocol engine and two USB devices with their own device addresses. One device is for the hub, with one control endpoint and one interrupt endpoint. The other for the peripheral function(s), with one control endpoint and two programmable endpoints. Each endpoint, except for the hub's interrupt endpoint, has its own 8-byte-deep FIFO. The hub's interrupt endpoint has a data register that contains the hub status change bits. The Serial Bus Controller manages the device addresses, monitors the status of the transactions, manages the FIFOs and communicates to the microcontroller through a set of status and control registers. The System Interface connects the USB Interface to the microcontroller.

USB Hub Repeater

The USB hub repeater contains the switching circuitry for the hub function. It is responsible for port connectivity setup and teardown, overcurrent detection and port power control. It also supports exception handling such as bus fault detection and recovery, and connect/disconnect detection.

USB Hub and Function Controller

The hub and function controller is implemented in the microcontroller's firmware.

The firmware available from Atmel for the hub controller provides complete support for USB hub functions for both full-speed and low-speed devices without requiring user intervention. It detects USB device attach and detach and automatically enables and disables the ports as required. It manages hub connectivity: power control, port connect/disconnect, suspend and fault recovery. The hub controller interprets and responds to all device as well as hub specific requests from the USB host and manages the device and hub descriptors. Power switch control and overcurrent protection signals are provided to control external power supply regulators. In addition, an option for an output pin indicating the connectivity status of the hub's downstream ports are also available.

The firmware for the Function Controller supports the enumeration for HID class devices. Other device classes are simple extensions.

AVR Microcontroller

The peripherals and features of the AT43USB320A microcontroller are similar to those of the AT90S8515, with the exception of the following modifications:

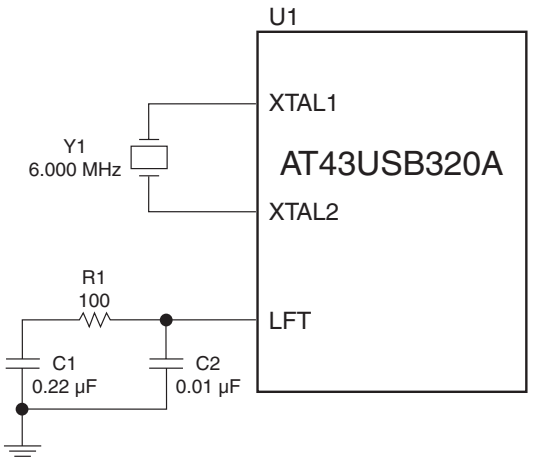
1. 64K bytes program memory addressing range
2. No on-chip program memory
3. No analog comparator
4. No external data memory accesses
5. No internal pull-ups in the GP IO port pins

The microcontroller shares all the same control and status registers of the AVR90S8515. The registers for managing the USB operations are mapped into its SRAM space.

Oscillator and Clock Generator

The external crystal for the on-chip oscillator is selectable as 6 MHz or 12 MHz. The PLL multiplies the oscillator's clock to 48 MHz for the USB block. The 6/$\overline{12M}$ input pin selects the multiplier for the PLL.

Figure 5. Oscillator Specifications



The oscillator of the AT43USB320A is of a special low-power design and needs no external components apart from the crystal. The crystal should be of the parallel resonance type, 6 MHz (or 12 MHz, if so selected) requiring 10 pF load capacitance. If a crystal with higher load capacitance is used, this can be added in the form of external capacitors between the crystal pins and ground. The crystal must have an accuracy and stability of 100 ppm or better. The USB specification demands that full-speed devices have a frequency accuracy of 500 ppm or better, which is why a good high-quality crystal is required. A ceramic resonator will work with the AT43USB320A, but would make the hub non-compliant and unreliable.

The microcontroller always runs from a 12 MHz clock that is generated by the USB hardware. While the nominal and average period of this clock is 83.3 ns, it may have single cycles that deviate by ± 20.8 ns during a phase adjustment by the SIE's clock/data separator.

On-chip Power Supply

The AT43USB320A contains an on-chip power supply that generates 3.3V with a capacity of 30 mA from the 5V power input. This on-chip power supply is intended to supply the AT43USB320A operation only and should not be used for other purposes.

One should be careful when the GPIO pins are required to supply high loads. If the application requires that the GPIO supply currents beyond the capability of the on-chip power supply, the AT43USB320A should be supplied by an external 3.3V power supply. In this case, the 5V VCC power supply pin should be left unconnected and the 3.3V power supplied to the chip through the CEXT pin.

Interrupt Handling

Please refer to the Atmel AVR manual regarding the Interrupt and Interrupt Vectors. In the AT43USB320A, the external interrupt INT0 is used for the USB Suspend/Resume Interrupt. This is achieved by internally OR'ing the external INT0 signal with the Suspend/Resume signals from the USB hardware. The vector for this interrupt is Vector 2 at program address $00.

The interrupt vector assigned to the Analog Comparator module of the AT90S8515, Vector 13 at program address $00C, is used by the USB hardware endpoint status and control functions.

All the interrupts are maskable through the interrupt enable register. The suspend and resume interrupts are cleared when the firmware writes a zero to the interrupt bit. All other interrupts are cleared when the microcontroller sets a bit in an interrupt acknowledge register. There is one bit for each interrupt source.

Endpoint Interrupts

The endpoint interrupts are triggered by setting or clearing one or more bits in the endpoint's Control and Status Register which are caused by events during packet transactions. These triggers are different for control and non-control endpoints as described below. Please refer to the Control and Status Register.

Interrupt for non-control endpoints:

1. RX OUT Packet 0 -> 1 (OUT endpoints)
2. TX Packet Ready 1 -> 0 (IN endpoints)
3. TX Complete 0 -> 1 (IN endpoints)

Interrupt for control endpoints:

1. RX OUT Packet 0 -> 1
2. RX SETUP 0 -> 1
3. TX Packet Ready 1 -> 0
4. TX Complete 0 -> 1

Function Interface Unit

The Function Interface Unit provides the interface between the microcontroller and the SIE for the embedded function. It manages transactions at the packet level with minimal intervention from the microcontroller. It contains three endpoints: a control endpoint and two programmable endpoints. Each endpoint has an 8-byte bi-directional FIFO to buffer the data to be transmitted or data received during the data phase of a transaction.

The FIU is designed to operate in the single-packet mode and to manage the USB packet protocol layer. To operate the FIU, the firmware must first enable the endpoints of the FIU. After being enabled, the endpoints are in the receive mode by default. The FIU notifies the microcontroller when a valid token has been received. The data contained in the data packet will be supplied in the FIFO. The microcontroller transfers the data to and from the host by interacting with each endpoint's FIFO and Control and Status Registers.

For example, when transmitting an IN packet, the microcontroller simply loads the data into the endpoint's FIFO and sets a bit in the control and acknowledge registers. The FIU will assemble the data in a USB packet and signals the microcontroller when the host has acknowledged receipt of the packet. The UFI performs automatic data retries and data toggles.

For SETUP tokens, the microcontroller must parse the Device Request and then responds appropriately. After a SETUP token, there may be zero or more DATA IN or DATA OUT packets for which the microcontroller must either supply or receive the data. The maximum packet size is fixed at 8 bytes.

Control Transfers at Function EP0

The next few sections describe how the AT43USB320A's USB hardware and firmware operate during a control transfer between the host and the hub's or function's control endpoint.

Note: DATA1/DATA0 = Data packet with DATA1 or DATA0 PID
DATA1(0) = Zero length DATA1 packet

Control Transfers at Control Endpoint EP0

The description given below is for the function control endpoint, but applies to the hub control endpoint as well if the proper registers are used.

Figure 6 describes the three possible types of control transfers: Control Write, Control Read and No-data Control.

Figure 6. Control Transfers

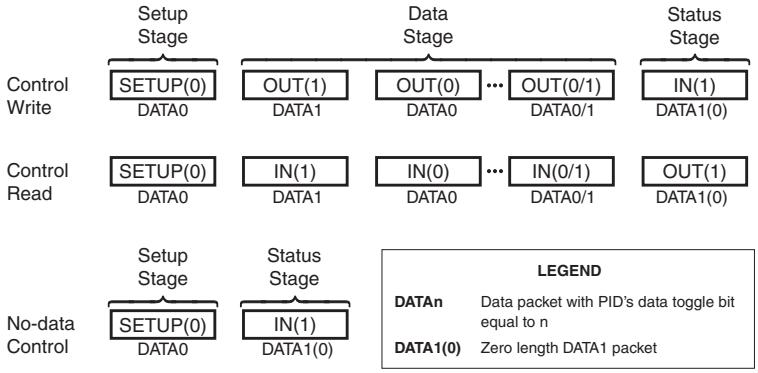


Figure 7 shows how the various state transitions are triggered. Additional decision-making may take place within the response states to determine the next expected state. Unmarked arcs represent transitions that trigger immediately following completion of the response state processing. Stable states, those requiring an interrupt to exit having no unmarked arcs as exit paths, are shown in bold.

Figure 7. State Diagram

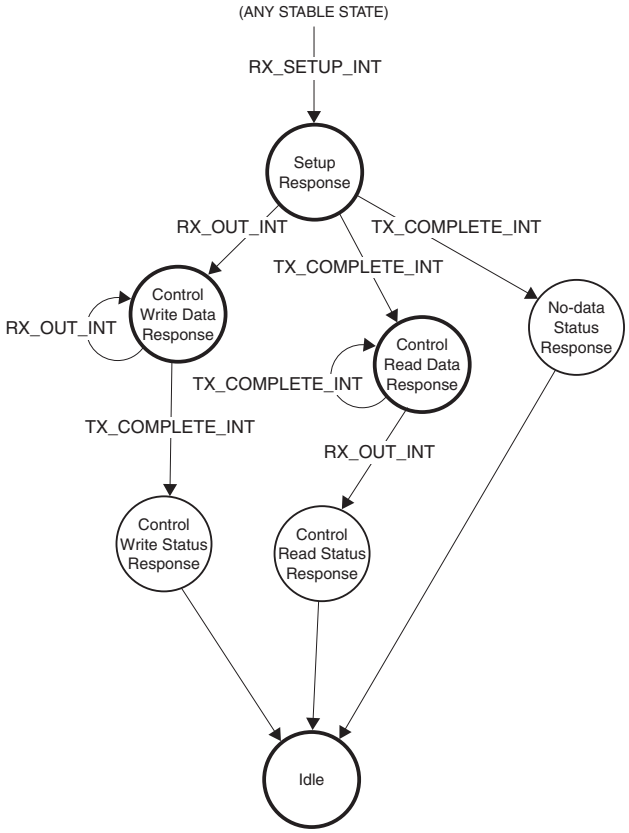


Table 2. SETUP Stage

Hardware	Firmware
1. SETUP token, Data from Host 2. ACK to Host 3. Store data in FIFO 4. Set RX SETUP => INT	 5. Read UISR 6. Read HCSR0/FCSR0 7. Read Byte Count 8. Read FIFO 9. Parse data 10. Write to H/FCAR0: a. If Control Read: set DIR, clear RX SETUP, fill FIFO, set TX Packet Ready b. If Control Write: clear DIR c. If no Data Stage: set Data End, clear DIR set Force STALL d. If unsupported command: set Force STALL 11. Set UIAR[HEP0 INTA]

Table 3. STATUS Stage, No DATA Stage

Hardware	Firmware
1. IN token from Host 2. Send DATA1(0) 3. ACK from Host 4. Set TX Complete => INT	 5. Read UISR 6. Read HCSR0/FCSR0 7. If SET ADDRESS, program address, Set GLB_STATE[HADD Enable] 8. Acknowledge TX Complete, clear Data End, set Force STALL in H/FCAR0 9. Set UIAR[HEP0 INTA]

Table 4. DATA Stage, Control READ

Hardware	Firmware
1. IN token from Host 2. a. If TX Packet Ready = 1, send Data0/DATA1 b. If TX Packet Ready = 0, send NAK 3. ACK from Host 4. Clear TX Packet Ready Set TX Complete => INT	5. Read UISR 6. Read HCSR0/FCSR0 7. Acknowledge TX Complete in H/FCAR0: a. If more data: fill FIFO, set TX Packet Ready, set DIR b. If no more data: set Force STALL, set DATA END 8. Set UIAR[HEP0 INTA]

Note: Repeat 1 through 8.

Table 5. STATUS/Early STATUS Stage with READ DATA Stage

Hardware	Firmware
1. OUT token from Host 2. DATA1(0) from Host 3. TX Complete = 0? a. If yes, ACK to Host Set RX OUT => INT b. If no, NAK to Host	4. Read UISR 5. Read HCSR0/FCSR0 6. Acknowledge RX OUT, set Data End, set Force Stall in H/FCAR0. **Comment:** A SETUP token will clear Data End. Not cleared by firmware in case Host retries 1 through 3. 7. Set UIAR[HEP0 INTA]

Table 6. DATA Stage, Control WRITE

Hardware	Firmware
1. OUT token from Host	
2. Put DATA0/DATA1 into FIFO	
3. ACK to Host	
4. Set RX OUT => INT	
	5. Read UISR
	6. Read HCSR0/FCSR0
	7. Read FIFO
	8. Acknowledge RX OUT interrupt in H/FCAR0. If last packet, set Data End, set Force STALL.
	9. Set UIAR[HEP0 INTA]

Note: Repeat 1 through 9 until DATA packet.

Table 7. STATUS Stage with WRITE DATA Stage

Hardware	Firmware
1. IN token from Host	
2. Send Data1(0)	
3. ACK from Host	
4. Set TX Complete => INT	
	5. Read UIR
	6. Read CSR
	7. Acknowledge TX Complete interrupt, clear Data End, set Force STALL in H/FCAR0
	8. Set UIAR[HEP0 INTA]

Interrupt/Bulk Transfers at Function EP1 and 2

1. Hardware automatically starts FEP1/2 in receive mode and NAKs all IN token as long as CAR[TX Packet Ready] is cleared.
2. microcontroller checks CAR[TX Packet Ready], if it is 0, writes the data into the FIFO, then sets CAR[TX Packet Ready].
3. At the next IN token, the hardware sends the packet out and waits for an ACK. Until an ACK is received, the hardware will re-transmit the data in the FIFO.
4. After receiving an ACK, hardware clears CAR[TX Packet Ready] signaling successful completion to the microcontroller.

Hub Interface Unit

The hub interface unit provides the interface between the microcontroller on one side and the SIE and repeater on the other side. Except for the functions associated with the hub repeater and the ports, it is similar to the Function Interface Unit. The HIU's interrupt endpoint has no FIFO. Just a single byte transmit buffer/register containing the hub and port status change bitmap.

Control transactions for the hub control endpoint proceed exactly the same way as those described for the embedded function. The firmware updates the Hub and Port Status Change Bitmap register whenever a hub or port status has changed. No firmware is required for the hub's Endpoint1 transactions. This endpoint will automatically send the Hub and Port Status Change Bitmap as a response to an IN token if a change occurred since the last IN token. If no change has occurred, the interrupt endpoint will respond with a NAK.

Hub and Port Power Control

For most flexibility, the USB hardware of the AT43USB320A is designed to accommodate hubs of various capabilities: bus- or self-powered, per port or global overcurrent sensing, switched or unswitched port power, individual or gang power switching. While the interface to the external power supply monitoring and switching is achieved through the microcontroller's I/O ports, the USB hardware in the repeater contains the circuitry to handle all the possible combinations of port power management tasks.

The AT43USB320A is specified as a hub with individual overcurrent monitoring and per port power switching. It is also assumed that external power switches like the Micrel MIC2026 are used.

Overcurrent Sensing

1. *Global Overcurrent Protection.* In this mode, the Port Overcurrent Indicator and Port Overcurrent Indicator Change should be set to 0s. For the AT43USB320A, a MIC2545 is required to switch power to all four USB ports. The FLG output of the MIC2545 should be connected to one input port of the microcontroller. When an overcurrent occurs, FLG is asserted and the firmware should set the Hub Overcurrent Indicator and Hub Overcurrent Indicator Change and switch off power to the hub.
2. *Individual Port Overcurrent Protection.* The Hub Overcurrent Indicator and Hub Overcurrent Indicator Change bits should be set to 0s. One MIC2026 is required for each USB port. Each $\overline{\text{FLG}}$ output of the MIC2026 should be connected to one input port of the microcontroller. An overcurrent is indicated by $\overline{\text{FLG}}$ being asserted. The firmware sets the corresponding port's Overcurrent Indicator and the Overcurrent Indicator Change bits and switches off power to the port. At the next IN token from the host, the AT43USB320A reports the status change.

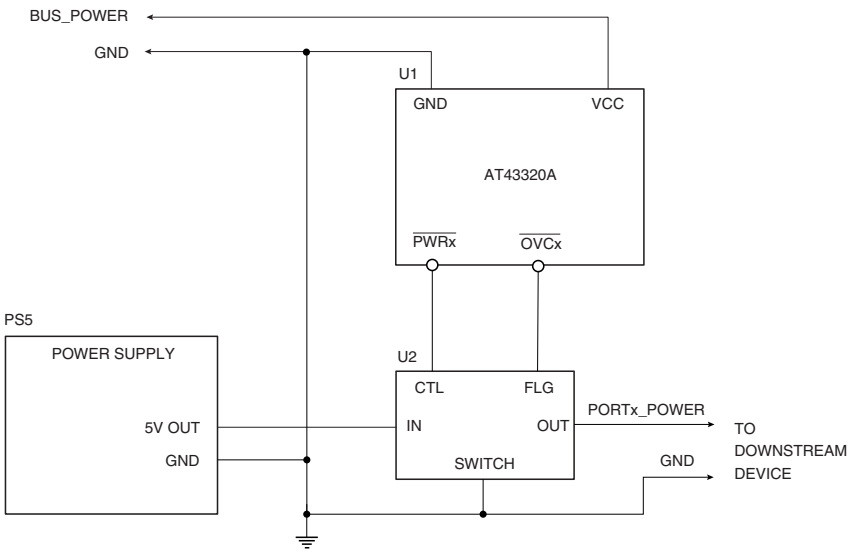
Port Power Switching

1. *Gang Power Switching.* One of the microcontroller I/O port pin must be programmed as an output to control the external switch, $\overline{PWR}$. Switch ON is requested by the USB host through the SetPortFeature (PORT_POWER) request. The firmware sets the corresponding port's Power Control Bit in the HPPWR. The firmware asserts $\overline{PWR}$ any time one of the port's Power Control Bits is set. Switch OFF is executed upon receipt of a ClearPortFeature (PORT_POWER) or upon detecting an overcurrent condition. The firmware clears the Power Control Bit. Only if all four of the Power Control Bits of Ports 1 - 4 are cleared should the firmware de-assert the $\overline{PWR}$ pin.
2. *Individual Power Switching.* One microcontroller I/O port pin must be assigned for each USB port to control the external switch, $\overline{PWRx}$, where x = 1 - 4. Each of the Power Control Bits control one $\overline{PWRx}$.

Power Management Circuit

The following diagram shows a simplified diagram of a typical power management circuit of an AT43USB320A-based hub design.

Figure 8. AT43USB320A-based Hub Design



Suspend and Resume

The AT43USB320A enters suspend only when requested by the USB host through bus inactivity for at least 3 ms. The USB hardware would detect this request, sets the GLB SUSP bit of SPRSR, Suspend/Resume Register, and interrupts the microcontroller if the interrupt is enabled. The microcontroller should shut down any peripheral activity and enter the power-down mode by setting the SE and SM bits of MCUCR and then execute the SLEEP instruction. The USB hardware shuts off the oscillator and PLL.

Global Resume

Global Resume is signaled by a "J" to "K" state change on Port 0. The USB hardware enables the oscillator/PLL, propagates the RESUME signaling, and sets the RSM bit of the SPRSR which generates an interrupt. The microcontroller starts executing where it left off and services the interrupt. As part of the ISR, the firmware clears the GLB SUSP bit.

Remote Wake-up

While the AT43USB320A is in global suspend, resume signaling is also possible through remote wake-up if the remote wake-up feature is enabled. Remote wake-up is defined as a port connect, port disconnect or resume signaling received at Ports 1 - 4 or, in case of the embedded function, through an external interrupt.

A remote wake-up initiated at Ports 1 - 4 is similar in many respects to a global wake-up. The USB hardware enables the oscillator/PLL, propagates the RESUME signaling, and sets the RSM bit of the SPRSR which generates an interrupt. The microcontroller starts executing where it left off and services the interrupt. As part of the ISR, the firmware clears the GLB SUSP bit.

A remote wake-up from the embedded function is initiated through the external interrupt, INT0, which enables the oscillator/PLL and the USB hardware. The USB hardware drives RESUME signaling and sets the FRMWUP and RSM bits of SPRSR which generates an interrupt to the microcontroller. The microcontroller starts executing where it left off and services the interrupt. As part of the ISR, the firmware clears the GLB SUSP bit.

At completion of RESUME signaling, the USB hardware sets the Port Suspend Status Change bits of the Hub Port Status Change Registers.

Selective Suspend and Resume

See section on Hub Port Control Register, HPCON.

The Suspend and Resume Process

Global Suspend

Hardware	Firmware
1. Host stops sending packets	
2. Global suspend signaling detected	
3. Stop downstream signaling	
4. Set GBL SUS bit -> interrupt	
	5. Shut down any peripheral activity
	6. Set SE and SM bits of MCUCR
	7. Execute SLEEP instruction
8. SLEEP bit detected	
9. Shut off oscillator	

Global Resume

Hardware	Firmware
1. Host resumes signaling	
2. Resume signaling detected	
3. Propagate signaling downstream	
4. Enable oscillator	
5. Set RSM bit -> interrupt	
	6. Reset RSM and GBL SUSP bits
	7. Enable peripheral activity

Remote Wake-up, Ports 1 - 4

Hardware	Firmware
1. Connect/disconnect/port resume detected	
2. Propagate resume signaling	
3. Enable oscillator	
4. Set RSM bit -> interrupt	
	5. Reset RSM and GBL SUSP bits
	6. Enable peripheral activity

Remote Wake-up, Embedded Function

Hardware	Firmware
1. External event activates INT	
2. Propagate resume signaling	
3. Enable oscillator	
4. Set RSM and FRMWUP bits -> interrupt	
	5. Clear GLB SUSP, RSM, FRMWUP bits
	6. Enable peripheral activity

Selective Suspend, Ports 1 - 4

Hardware	Firmware
	1. Set or Clear Port Feature PORT_SUSPEND decoded
	2. Write HPCON[2:0] and HPADD[2:0] bits
3. Suspend or resume port per command	

Selective Suspend, Embedded Function

Hardware	Firmware
	1. Set Port Feature PORT_SUSPEND decoded
	2. Disable Port 5's endpoints.

Selective Resume, Embedded Function

Hardware	Firmware
	1. Clear Port Feature PORT_SUSPEND decoded
	2. Clear Port 5 suspend status bit.
	3. Wait 23 ms, then set enable status bit and suspend change bit
	4. Enable Port 5 endpoints
5. Send updated port status at next IN to Endpoint1.	

USB Reset

If the USB hardware receives a reset from the host through Port 0, SEO for at least four low-speed USB clock cycles, it will generate a reset to the microcontroller. The USB hardware generates a reset pulse of 24 oscillator periods. This USB initiated reset signal is internally OR'ed with the external microcontroller reset signal, $\overline{RST}$.

Please refer to the AT90S8515 manual for details about the reset.

I/O Memory

I/O Address	SRAM Address	Name	Function
$3F	$5F	SREG	Status Register
$3E	$5E	SPH	Stack Pointer High
$3D	$5D	SPL	Stack Pointer Low
$3B	$5B	GIMSK	General Interrupt Mask Register
$3A	$5A	GIFR	General Interrupt Flag Register
$39	$59	TIMSK	Timer/Counter Interrupt Mask Register
$38	$58	TIFR	Timer/Counter Interrupt Mask Register
$35	$55	MCUCR	MCU General Control Register
$33	$53	TCCR0	Timer/Counter0 Control Register
$32	$52	TCNT0	Timer/Counter0 (8-bit)
$2F	$4F	TCCR1A	Timer/Counter1 Control Register A
$2E	$4E	TCCR1B	Timer/Counter1 Control Register B
$2D	$4D	TCNT1H	Timer/Counter1 High Byte
$2C	$4C	TCNT1L	Timer/Counter1 Low Byte
$2B	$4B	OCR1AH	Timer/Counter1 Compare Register A High Byte
$2A	$4A	OCR1AL	Timer/Counter1 Compare Register A Low Byte
$29	$49	OCR1BH	Timer/Counter1 Compare Register B High Byte
$28	$48	OCR1BL	Timer/Counter1 Compare Register B Low Byte
$25	$45	ICR1H	T/C1 Input Capture Register High
$24	$44	ICR1L	T/C1 Input Capture Register Low
$21	$41	WDTCR	Watchdog Timer Counter Register
$1B	$4B	PORTA	Data Register, Port A
$1A	$3A	DDRA	Data Direction Register, Port A
$19	$39	PINA	Input Pins, Port A
$18	$38	PORTB	Data Register, Port B
$17	$37	DDRB	Data Direction Register, Port B
$16	$36	PINB	Input Pins, Port B
$15	$35	PORTC	Data Register, Port C
$14	$34	DDRC	Data Direction Register, Port C
$13	$33	PINC	Input Pins, Port C
$12	$32	PORTD	Data Register, Port D
$11	$31	DDRD	Data Direction Register, Port D
$10	$30	PIND	Input Pins, Port D
$0F	$2F	SPDR	SPI I/O Data Register
$0E	$2E	SPSR	SPI Status Register
$0D	$2D	SPCR	SPI Control Register
$0C	$2C	UDR	UART I/O Data Register
$0B	$2B	USR	UART Status Register
$0A	$2A	UCR	UART Control Register
$09	$29	UBRR	UART Baud Rate Register

Interrupt Vectors

Vector #	Program Address	Source	Interrupt Definition
1	$000	RESET	Hardware Pin and Watchdog Reset
2	$001	SUSP/RESM	USB Suspend and Resume
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$006	TIMER1 OVF	Timer/Counter1 Overflow
8	$007	TIMER0 OVF	Timer/Counter0 Overflow
9	$008	SPI, STC	SPI Serial Transfer Complete
10	$009	UART RX	UART Rx Complete
11	$00A	UART UDRE	UART Data Register Empty
12	$00B	UART TX	UART Tx Complete
13	$00C	USB Hardware	USB Hardware Interrupt

USB Interrupt Sources

Interrupt	Description
SOF Received	Whenever USB hardware decodes a valid Start of Frame. The frame number is stored in the two Frame Number Registers.
EOF1	Activated whenever the hub's frame timer reaches its EOF2 time-point.
EOF2	Activated whenever the hub's frame timer reaches its EOF1 time-point.
Function EP0 Interrupt	Function Interface Unit Endpoint0 Interrupt
Function EP1 Interrupt	For an OUT endpoint it indicates that Function Endpoint1 has received a valid OUT packet and that the data is in the FIFO. For an IN endpoint it means that the endpoint has received an IN token, sent out the data in the FIFO and received an ACK from the Host. The FIFO is now ready to be written by new data from the microcontroller.
Function EP2 Interrupt	See Function EP1 Interrupt.
Hub EP0 Interrupt	Hub Interface Unit Endpoint0 Interrupt
Hub EP1 Interrupt	Indicates that Function Endpoint1 has received an IN token, sent out the data in the Hub Change Status Register and received an ACK from the Host.
FRWUP	USB hardware has received an embedded function remote wake-up request.
GLB SUSP	USB hardware has received global suspend signaling and is preparing to put the hub in the suspend mode. The microcontroller's firmware should place the embedded function in the suspend state.
RSM	USB hardware received resume signaling and is propagating the resume signaling. The microcontroller's firmware should take the embedded function out of the suspended state.

USB Register Set

The following sections describe each of the registers of the hub and function interfaces. The registers described are for a generic USB core hardware that can be used for USB devices other than the AT43USB320A. All the bits in these registers are readable and writable by the microcontroller.

Note: R = Read
W = Write

USB Hub and Function Register Summary

Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UISR	1FF7	SOF INT	EOF2 INT	EOF1 INT	HEP1 INT	HEP0 INT	FEP2 INT	FEP1 INT	FEP0 INT
UIER	1FF3	SOF IE	EOF2 IE	EOF1 IE	HEP1 IE	HEP0 IE	FEP2 IE	FEP1 IE	FEP0 IE
UIAR	1FF5	SOF INTA	EOF2 INTA	EOF1 INTA	HEP1 INTA	HEP0 INTA	FEP2 INTA	FEP1 INTA	FEP0 INTA
HADDR	1FEF	SAEN	HADD6	HADD5	HADD4	HADD3	HADD2	HADD1	HADD0
FADDR	1FEE	FEN	FADD6	FADD5	FADD4	FADD3	FADD2	FADD1	FADD0
FRM_ NUM_ L	1FFC	FCL7	FCL6	FCL5	FCL4	FCL3	FCL2	FCL1	FCL0
FRM_ NUM_ H	1FFD	SOF OK					FCH10	FCH9	FCH8
HENDP0_ CNTR	1FE7	EPEDS				DTGLE	EPDIR	EPTYPE1	EPTYPE0
HENDP1_ CNTR	1FE6	EPEDS				DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP0_ CNTR	1FE5	EPEDS				DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP1_ CNTR	1FE4	EPEDS				DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP2_ CNTR	1FE3	EPEDS				DTGLE	EPDIR	EPTYPE1	EPTYPE0
HDR0	1FD7	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR0	1FD5	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR1	1FD4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR2	1FD3	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
HBYTE_ CNT0	1FCF				BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_ CNT0	1FCD				BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_ CNT1	1FCC				BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_ CNT2	1FCB				BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
HCSR0	1FDF					STALL SNT	RX SETUP	RX OUT PACK	TX CMPLT
HCSR1	1FDE					STALL SNT	RX SETUP	RX OUT PACK	TX CMPLT
FCSR0	1FDD					STALL SNT	RX SETUP	RX OUT PACK	TX CMPLT
FCSR1	1FDC					STALL SNT	RX SETUP	RX OUTPACK	TX CMPLT
FCSR2	1FDB					STALL SNT	RX SETUP	RX OUT PACK	TX CMPLT
HCAR0	1FA7	CTL DIR	DATA END	FORCE STALL	TX PACK RDY	CSRACK3	CSRACK2	CSRACK1	CSRACK0
HCAR1	1FA6	CTL DIR	DATA END	FORCE STALL	TX PACK RDY	CSRACK3	CSRACK2	CSRACK1	CSRACK0

USB Hub and Function Register Summary (Continued)

Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FCAR0	1FA5	CTL DIR	DATA END	FORCE STALL	TX PACK RDY	CSRACK3	CSRACK2	CSRACK1	CSRACK0
FCAR1	1FA4	CTL DIR	DATA END	FORCE STALL	TX PACK RDY	CSRACK3	CSRACK2	CSRACK1	CSRACK0
FCAR2	1FA3	CTL DIR	DATA END	FORCE STALL	TX PACK RDY	CSRACK3	CSRACK2	CSRACK1	CSRACK0
GLC_ STATE	1FFB	OVC MODE	SW PWR	PWR MODE			RMWUPE	CONFG	HADD EN
HSTR	1FC7					OVLSC	LPSC	OVI	LPS
HPCON	1FC5		HPCON2	HPCON1	HPCON0		HPADD2	HPADD1	HPADD0
HPSTAT1	1FB8		LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT2	1FB9		LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT3	1FBA		LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT4	1FBB		LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT5	1FBC		LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
PSTATE1	1FA8							DPSTATE	DMSTATE
PSTATE2	1FA9							DPSTATE	DMSTATE
PSTATE3	1FAA							DPSTATE	DMSTATE
PSTATE4	1FAB							DPSTATE	DMSTATE
PSTATE5	1FAC							DPSTATE	DMSTATE
HPSTCH1	1FB0				RSTSC	POCIC	PSSC	PESC	PCSC
HPSTCH2	1FB1				RSTSC	POCIC	PSSC	PESC	PCSC
HPSTCH3	1FB2				RSTSC	POCIC	PSSC	PESC	PCSC
HPSTCH4	1FB3				RSTSC	POCIC	PSSC	PESC	PCSC
HPSTCH5	1FB4				RSTSC	POCIC	PSSC	PESC	PCSC
SPRSR	1FFA						FRWUP	RSM	GLB SUSP
SPRSIE	1FF9						FRWUP IE	RSM IE	GLB SUSP IE

UISR – USB Interrupt Status Registe

Register Address: 0x1FF7

Default State: 0x00r

UISR – USB Interrupt Status Registe

Bit	Field	USB Hardware	Description
7	SOF INT	W	Start of Frame Interrupt. Asserted after the receipt of a valid SOF packet
6	EOF2 INT	W	EOF2 Interrupt. Asserted 10 clocks before the expected start of a frame
5	EOF1 INT	W	EOF1 Interrupt. Asserted 32 clocks before the expected start of a frame
4	HEP1 INT	W	Hub Endpoint1 Interrupt
3	HEP0 INT	W	Hub Endpoint0 Interrupt
2	FEP2 INT	W	Function Endpoint2 Interrupt
1	FEP1 INT	W	Function Endpoint1 Interrupt
0	FEP0 INT	W	Function Endpoint0 Interrupt

The hub and function interrupt bits will be set by the hardware whenever the following bits in the corresponding endpoint's Control and Status Register are modified by the USB hardware:

1. RX OUT Packet is set (control and OUT endpoints)
2. TX Packet Ready is cleared (control and IN endpoints)
3. RX SETUP is set (control endpoints only)
4. TX Complete is set (control endpoints only)

UIER – USB Interrupt Enable Register

Register Address: 0x1FF3

Default State: 0x00

The bits in this register have the following meaning:

1 = enable interrupt

0 = disable interrupt

UIER – USB Interrupt Enable Register

Bit	Field	USB Hardware	Description
7	SOF IE	R	Enable SOF Interrupt
6	EOF2 IE	R	Enable EOF2 Interrupt
5	EOF1 IE	R	Enable EOF1 Interrupt
4	HEP1 IE	R	Enable Hub Endpoint1 Interrupt
3	HEP0 IE	R	Enable Hub Endpoint0 Interrupt
2	FEP2 IE	R	Enable Function Endpoint2 Interrupt
1	FEP1 IE	R	Enable Function Endpoint1 Interrupt
0	FEP0 IE	R	Enable Function Endpoint0 Interrupt

UIAR – USB Interrupt Acknowledge Register

The bits in this register are used to indirectly clear the bits of the UISR. A bit in the UISR is cleared by the microcontroller if a one is written in the corresponding bit of UIAR.

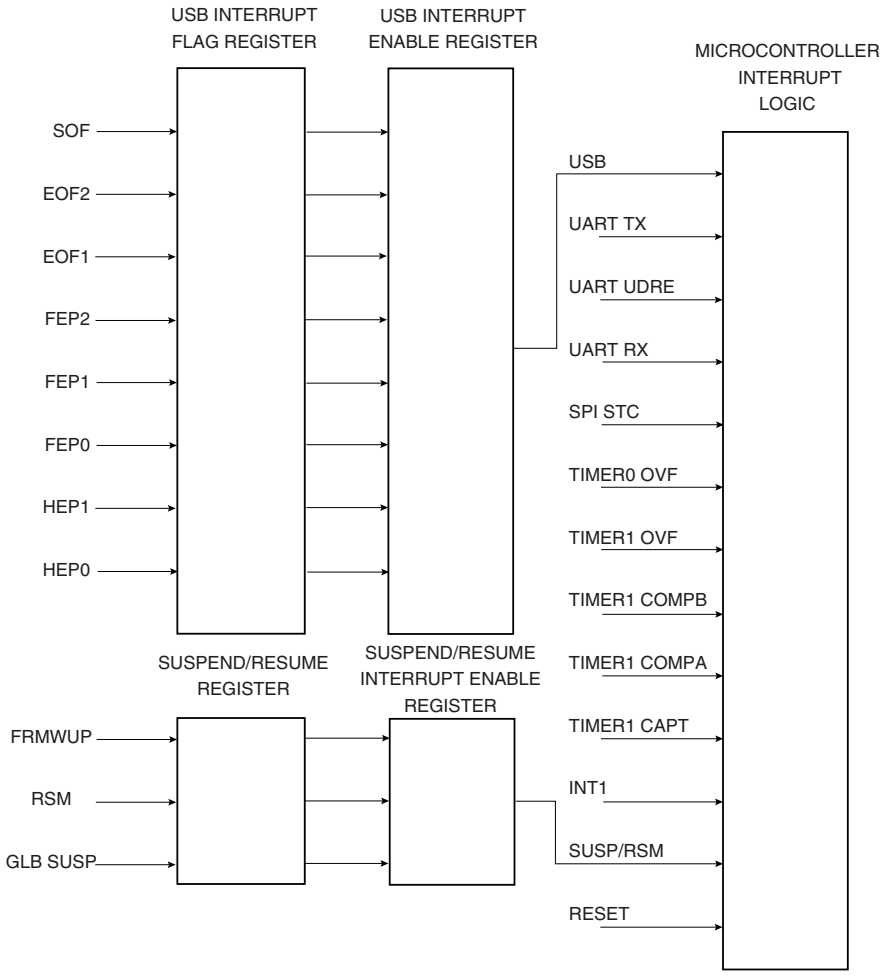
Register Address: 0x1FF5

Default State: 0x00

UIAR – USB Interrupt Acknowledge Register

Bit	Field	USB Hardware	Description
7	SOF INTA	W	Start of Frame Interrupt Acknowledge
6	EOF2 INTA	W	EOF2 Interrupt Acknowledge
5	EOF1 INTA	W	EOF1 Interrupt Acknowledge
4	HEP1 INTA	W	Hub Endpoint1 Interrupt Acknowledge
3	HEP0 INTA	W	Hub Endpoint0 Interrupt Acknowledge
2	FEP2 INTA	W	Function Endpoint2 Interrupt Acknowledge
1	FEP1 INTA	W	Function Endpoint1 Interrupt Acknowledge
0	FEP0 INTA	W	Function Endpoint0 Interrupt Acknowledge

Figure 9. AT43USB320A Interrupt Structure



HADDR: Hub Address Register

The hub interface unit contains an address register that contains the hub address assigned by the host. When a SET_ADDRESS command is received from the host, firmware must write the assigned address into this register. The USB hardware uses the new address only after the status phase of the transaction is completed when the microcontroller has enabled the new address by setting bit 0 of the Hub Global State Register. After power up or reset, this register will contain the value of 0x00.

The Single Address Enable bit allows the microcontroller to configure the AT43USB320A into a single address or a composite device. Once this capability is enabled, the Function Endpoint0 (FEP0) is converted from a control endpoint to a programmable endpoint. All the endpoints operate on the single address.

Register Address: 0x1FEF

Default State: 0x00

HADDR: Hub Address Register

Bit	Mnemonic	USB Hardware	Description
7	SAEN	R	Single Address Enable
6 - 0	HADD[6:0]	R	Hub Address

FADDR: Function Address Register

The Function Interface Unit contains an address register that contains the function address assigned by the host. When a SET_ADDRESS command is received from the host, firmware must write the assigned address into this register. The USB hardware uses the new address only after the status phase of the transaction is completed. After power-up or reset, this register will contain the value of 0x00.

The Function Enable bit (FEN) allows the firmware to enable or disable the function endpoints if the AT43USB320A is programmed as a hub/function compound device. The firmware should set this bit after receipt of a reset through the hub, SetPortFeature (PORT_RESET). Once this bit is set, the USB hardware passes packets between the host and the embedded function.

When the single address bit is set, the state of FEN is ignored.

Register Address: 0x1FEE

Default State: 0x00

FADDR: Function Address Register

Bit	Field	USB Hardware	Description
7	FEN	R	Function Enable
6 - 0	FADD[6:0]	R	Function Address

FRM_NUM_L: Frame Number Low Register

Register Address: 0x1FFC

Default State: 0x00

FRM_NUM_L: Frame Number Low Register

Bit	Field	USB Hardware	Description
7 - 0	FCL[7:0]	W	These are the lower 8 bits of the 11-bit frame number of the SOF packet.

FRM_NUM_H: Frame Number High Register

Register Address: 0x1FFD

Default State: 0x00

FRM_NUM_H: Frame Number High Register

Bit	Field	USB Hardware	Description
7	SOF OK	W	Valid Start of Frame Packet
6 - 3	Reserved		Reserved and set to 0
2 - 0	FCH[10:8]	W	These are the upper 3 bits of the 11-bit frame number of the SOF packet.

Endpoint Control Register

Register Address:

0x1FE7, HENDP0_CNTR Hub Endpoint0

0x1FE6, HENDP1_CNTR Hub Endpoint1

0x1FE5, FENDP0_CNTR Function Endpoint0

0x1FE4, FENDP1_CNTR Function Endpoint1

0x1FE3, FENDP2_CNTR Function Endpoint2

Default State: 0x000000b

Endpoint Control Register

Bit	Field	USB Hardware	Description
7	EPEDS	R	Endpoint Enable/Disable 0 = Disable endpoint 1 = Enable endpoint
6 - 4	Reserved		Reserved and set to 0
3	DTGLE	W	Data Toggle. Identifies DATA0 or DATA1 packets
2	EPDIR	R	Endpoint Direction. Only applicable for non-control endpoints 0 = Out 1 = In
1 - 0	EPTYPE	R	Endpoint Type. These bits program the type of endpoint. **Bit1 Bit0 Type** 0 0 Control 0 1 Isochronous 1 0 Bulk 1 1 Interrupt

FIFO Data Registers

FIFO data registers. These are dual function buffer registers. Received data are read by the microcontroller from the endpoint's FIFO through these data registers. In the transmit mode, the microcontroller writes to the FIFO through this register.

Register Address:

0x1FD7, HDR0 Hub Endpoint0

0x1FD5, FDR0 Function Endpoint0

0x1FD4, FDR1 Function Endpoint1

0x1FD3, FDR2 Function Endpoint2

Default State: 00000000b

FIFO Data Registers

Bit	Field	USB Hardware	Description
7 - 0	FIFO DATA [7:0]	R/W	Data to be written to FIFO or data to be read from the FIFO

Hub Endpoint1 has a single byte data register instead of a FIFO. This data register contains the Hub and Port Status Change Bitmap. This data register is automatically updated by the USB hardware and is not accessible by the firmware. The bits in this register when read by the host will be:

FIFO Data Registers

Bit	Mnemonic	USB Hardware	Description
7 - 6	0		Reserved
5	STCP5	R	Port 5 Status Change
4	STCP4	R	Port 4 Status Change
3	STCP3	R	Port 3 Status Change
2	STCP2	R	Port 2 Status Change
1	STCP1	R	Port 1 Status Change
0	STCHH	R	Hub Status Change

Byte Count Registers

Each endpoint, except Hub Endpoint1, has a register that stores the number of bytes to be sent or that was received by the USB hardware. The AT43USB320A hardware limits the maximum data length to 8 bytes only. Hub Endpoint1 has a single-byte data register, the Hub Status Change Bitmap, instead of a FIFO and has no byte count register.

Register Address:

0x1FCF, HBYTE_CNT0 Hub Endpoint0

0x1FCD, FBYTE_CNT0 Function Endpoint0

0x1FCC, FBYTE_CNT1 Function Endpoint1

0x1FCB, FBYTE_CNT2 Function Endpoint2

Default State: 00000000b

Byte Count Registers

Bit	Field	USB Hardware	Description
7 - 5	Reserved		
4 - 0	BYTCT[4:0]	R/W	Length of data packet in FIFO

Controller Service Routine Register

Register Address:

0x1FDF, HCSR0 Hub Endpoint0

0x1FDE, HCSR1 Hub Endpoint1

0x1FDD, FCSR0 Function Endpoint0

0x1FDC, FCSR1 Function Endpoint1

0x1FDB, FCSR2 Function Endpoint2

Default State: 00000000b

Controller Service Routine Register

Bit	Field	USB Hardware	Description
7 - 4	Reserved		Reserved and read as 0s
3	Stall Sent	W	The USB hardware sets this bit after a STALL is sent to the host.
2	RX SETUP	W	The USB hardware sets this bit when it receives a valid setup packet from the Host. This bit is used by control endpoints only.
1	RX OUT Packet	W	Indicates that the USB hardware has decoded an OUT token and that the data is in the FIFO.
0	TX Complete	W	The hardware sets this bit to indicate to a control endpoint that it has received an ACK handshake from the host.

The contents of this register is read by the firmware and is cleared indirectly by writing a one to the corresponding bit of the CAR, Control and Acknowledge Register.

- **STALL Sent**

The USB hardware sets this bit after a STALL has been sent. The firmware uses this bit when responding to a Get Status[Endpoint] request.

- **RX SETUP**

This bit is used by control endpoints only to signal to the microcontroller that the USB hardware has received a valid SETUP packet and that the data portion of the packet is stored in the FIFO. The hardware will clear all other bits in this register while setting RX SETUP.

If interrupt is enabled, the microcontroller will be interrupted when RX SETUP is set. After the completion of reading the data from the FIFO, firmware should clear this bit.

- **RX OUT Packet**

The USB hardware sets this bit after it has stored the data of an OUT transaction in the FIFO. While this bit is set, the hardware will NAK all OUT tokens. The USB hardware will not overwrite the data in the FIFO except for an early setup.

RX OUT Packet is used for the following operations:

1. Control write transactions by a control endpoint
2. OUT transaction with DATA1 PID to complete the status phase of a control endpoint
3. By a BULK OUT or ISO OUT or INT OUT endpoint

Setting this bit causes an interrupt to the microcontroller if the interrupt is enabled. Firmware clears this bit after the FIFO contents have been read.

- **TX Complete**

This bit is used by a control endpoint hardware to signal to the microcontroller that it has successfully completed certain transactions. TX Complete is set at the completion of a:

1. Control read data stage
2. Status stage without data stage
3. Status stage after a control write transaction

Control and Acknowledge Register

Register Address:

0x1FA7, HCAR0 Hub Endpoint0

0x1FA6, HCAR1 Hub Endpoint1

0x1FA5, FCAR0 Function Endpoint0

0x1FA4, FCAR1 Function Endpoint1

0x1FA3, FCAR2 Function Endpoint2

Default State: 00000000b

Control and Acknowledge Register

Bit	Field	USB Hardware	Description
7	DIR	R	Set by microcontroller to indicate to the USB hardware the direction of a control transfer. 0 = control write or no data stage 1 = control read This bit is used by control endpoints only.
6	Data End	R	Indicate that the microcontroller has placed the last data packet in FIFO, or that the microcontroller has processed the last data packet it expects from the Host.
5	Force Stall	R	Set by the microcontroller to indicate a stalled endpoint. The hardware will send a STALL handshake as a response to the next IN or OUT token, or whenever there is a control transfer without a Data Stage.
4	TX Packet Ready	R/C	Indicates that the microcontroller has loaded the FIFO with a packet of data. This bit is cleared by hardware after the USB Host acknowledges the packet. For ISO endpoints, this bit is cleared unconditionally after the data is sent.
3	CSRACK3	R	Acknowledge Stall Sent interrupt. Firmware sets this bit to clear CSR bit 3. The 1 written is not actually stored and thus does not have to be cleared.
2	CSRACK2	R	Acknowledge RX Setup interrupt. Firmware sets this bit to clear CSR bit 2. The 1 written is not actually stored and thus does not have to be cleared.
1	CSRACK1	R	Acknowledge RX OUT Packet interrupt. Firmware sets this bit to clear CSR bit 1. The 1 written is not actually stored and thus does not have to be cleared.
0	CSRACK0	R	Acknowledge TX Complete interrupt. Firmware sets this bit to clear CSR bit 0. The 1 written is not actually stored and thus does not have to be cleared.

- **Control Direction**

This bit is used by control endpoints only and is used by firmware to indicate the direction of a control transfer. It is written by the firmware after it receives a RX SETUP interrupt. The hardware uses this bit to determine the status phase of a control transfer.

- **Data End**

This bit is used by control endpoints only together with bit 1 (TX Packet Ready) to signal the USB hardware to go to the STATUS phase after the packet currently residing in the FIFO is transmitted. After the hardware completes the STATUS phase it will interrupt the microcontroller without clearing this bit.

- **Force STALL**

The microcontroller sets this bit if it wants to force a STALL. A STALL is sent if any of the following conditions is encountered:

1. An unsupported request is received
2. The host continues to ask for data after the data is exhausted
3. The control transfer has no data stage

- **TX Packet Ready**

This bit is used for the following operations:

1. Control read transactions by a control endpoint
2. IN transactions with DATA1 PID to complete the status phase for a control endpoint, when this bit is zero but Data End set high (bit 4)
3. By a BULK IN or ISO IN or INT IN endpoint

The microcontroller should write into the FIFO only if this bit is cleared. After it has completed writing the data, it should set this bit. This data can be of zero length.

Hardware clears this bit after it receives an ACK. If the interrupt is enabled and if the TX Complete bit is set, clearing the TX Packet Ready bit by the hardware causes an interrupt to the microcontroller.

Hub and Port Registers

GLB_STATE: Global State Register

Register Address: 0x1FFB

Default State: 00000000b

GLB_STATE: Global State Register

Bit	Field	USB Hardware	Description
7	OVC MODE	R	Overcurrent Sensing Mode. This bit informs the USB hardware whether the hub implements overcurrent sensing globally or individually. 0 = Global overcurrent sensing 1 = Individual overcurrent sensing
6	SW PWR	R	Switched Power. This bit informs the USB hardware whether the hub performs port power switching or whether port power follows hub power. 0 = No power switching 1 = Power switching
5	PWR MODE	R	Power Switching Mode. This bit informs the USB hardware whether power switching is ganged or per port. The bit is valid only if port power is switched, i.e. SW PWR = 1. 0 = Ganged power switching 1 = Per port power switching
3 - 4	Reserved		Reserved and set to 0
2	RMWUPE	R	Remote Wake-up Enable. This bit is set if the host enables the hub's remote wake-up feature.
1	CONFG	R	Configured. This bit is set by firmware after a valid SET_CONFIGURATION request is received. It is cleared by a reset or by a SET_CONFIGURATION with a value of 0.
0	HADD Enable	R	Hub Address Enable. This bit is set by firmware after the status phase of a SET_ADDRESS request transaction so the hub will use the new address starting at the next transaction.

HSTR: Hub Status Register

Register Address: 0x1FC7

Default State: 00000000b

HSTR: Hub Status Register

Bit	Mnemonic	USB Hardware	Description
7 - 4			Reserved
3	OVLSC	R	Overcurrent Status Change 0 = No change has occurred on Overcurrent Indicator 1 = Overcurrent Indicator has changed
2	LPSC	R	Hub Local Power Status Change 0 = No change has occurred on Local Power Status 1 = Local Power Status has changed
1	OVI	R	Overcurrent Indicator 0 = All power operations normal 1 = An overcurrent exist on a hub wide basis
0	LPS	R	Hub Local Power Status 0 = Local power supply is good 1 = Local power supply is lost (inactive)

The AT43USB320A supports all kinds of port power overcurrent protection and switching as well as bus-powered or self-powered hubs. The signals for sensing overcurrent and controlling the external power switches must be implemented with the AT43USB320A's GPIO pins.

1. Hub local power status, bits 0 and 2, are optional features and apply to hubs that report on a global basis. If this feature is not used, both of these bits should be programmed to zero. To use this feature, the firmware needs to know the status of the local power supply, which requires an input pin and extra internal or external circuitry.
2. Hub overcurrent status, bits 1 and 3, apply to self-powered hubs with bus-powered SIE only, or hubs that are programmable as self-/bus-powered. The AT43USB320A firmware should clear these two bits to zero.

The firmware uses bits 1 and 3 to generate bit 0 of the Hub and Port Status Change Bitmap, which is transmitted through the Hub Endpoint1 Data Register. Bit 0 of this register is a one whenever bit 1 or 3 of HSTATR is a one.

HPCON: Hub Port Control Register

Register Address: 0x1FC5

Default State: 00000000b

The microcontroller firmware uses the bits in this register to command the hub hardware to perform functions related to the port status. The lower three bits determine which port is being addressed.

HPCON: Hub Port Control Register

Bit	Mnemonic	USB Hardware	Description
7			Reserved. Set to 0s
6:4	HPCON 2:0	R	Encoded Hub Port Control Command. These bits are set and cleared by firmware upon receipt of a Host request. **Bit6 Bit5 Bit4 Action** 0 0 0 Disable port 0 0 1 Enable port 0 1 0 Reset and enable port 0 1 1 Suspend port 1 0 0 Resume port
3			Reserved. Set to 0s
2:0	HPADD 2:0	R	Hub Port Address. These bit define which port is being addressed for the command defined by bits [6:4]. **Bit2 Bit1 Bit0 Port Addressed** 1 0 1 Port 5 1 0 0 Port 4 0 1 1 Port 3 0 1 0 Port 2 0 0 1 Port 1

- **Disable Port = ClearPortFeature (PORT_ENABLE)**

Action: USB hardware places addressed port in disabled state. Port 5 is placed in disabled state by firmware.

- **Enable Port = SetPortFeature (PORT_ENABLE)**

Action: USB hardware places addressed port in enabled state. Firmware places Port 5 in enabled state.

- **Reset and Enable Port = SetPortFeature (PORT_RESET)**

Action: USB hardware drives reset signaling through addressed port. USB hardware and firmware resets their embedded function registers to the default state.

- **Suspend Port = SetPortFeature (PORT_SUSPEND)**

Action: USB hardware places port in idle state and stops propagating traffic through the addressed port.

USB hardware. Firmware places Port 5 in suspend state by disabling its endpoints and placing the peripheral function in its low power state.

- **Resume Port = ClearPortFeature (PORT_SUSPEND)**

Action: USB hardware sends resume signaling to addressed port and then enables port. Firmware takes the embedded function out of the suspend state and enables Port 5's endpoints.

- **Selective Suspend and Resume**

The host can selectively suspend and resume a port through the SetPortFeature (PORT_SUSPEND) and ClearPortFeature (PORT_SUSPEND).

A port enters the suspend state after the microcontroller interprets the suspend request and sets the appropriate bits of the Hub Port Control Register, HPCON. From this point on the hub repeater hardware is responsible for proper actions in placing Ports 1 - 4 in the suspend mode. For Port 5, the embedded function, the hardware will stop responding to any normal bus traffic, but the microcontroller firmware must place all external circuitry associated with the function in the low power state.

A port exits from the suspend state when the hub receives a ClearPortFeature (PORT_SUSPEND) or SetPortFeature (PORT_RESET). If the ClearPortFeature (PORT_SUSPEND) is directed towards Ports 1 - 4, the USB hardware drives a “K” downstream for at least 20 ms followed by a low-speed EOP. It then places the port in the enabled state. A ClearPortFeature (PORT_SUSPEND) to Port 5 (the embedded function) causes the firmware to wait 20 ms, take the embedded function out of the suspended state and then enable the port.

The ports can also exit from the suspended state through a remote wake-up if this feature is enabled. For Ports 1 - 4, this means detection of a connect/disconnect or an upstream directed “J” to “K” signaling. Remote wake-up for the embedded function is initiated through an external interrupt at INT0.

Hub Port Status Register

Register Address:

0x1FB8, HPSTAT1 Port1

0x1FB9, HPSTAT2 Port2

0x1FBA, HPSTAT3 Port3

0x1FBB, HPSTAT4 Port4

0x1FBC, HPSTAT5 Port5

Default State: 00000000b

The bits in this register are used by the microcontroller firmware when reporting a port's status through the Port Status Field, wPortStatus. Bits 3 (POCI) and 5 (PPSTAT) are used by the USB hardware and are the only two bits that the firmware should set or clear. All other bits should not be modified by firmware.

Hub Port Status Register

Bit	Mnemonic	USB Hardware	Description
7			Reserved
6	LSP	W	Low-speed Device Attached 0 = Full-speed device attached to this port 1 = Low-speed device attached to this port Set to 0 for Port 5 (full-speed only). Set and cleared by hardware upon detection of device at EOF2.
5	PPSTAT	R	Port Power Status 0 = Port is powered OFF 1 = Port is powered ON Set to 1 for Port 5. Set and cleared based on present status of port power.
4	PRSTAT	W	Port Reset Status 0 = Reset signaling not asserted 1 = Reset signaling asserted Set and cleared by hardware as a result of initiating a port reset by Port Control Register.
3	POCI	R	Port Overcurrent Indicator 0 = Power normal 1 = Overcurrent exist on port Set to 0 for Port 5. Set and cleared by firmware upon detection of an overcurrent or removal of an overcurrent.

Hub Port Status Register (Continued)

Bit	Mnemonic	USB Hardware	Description
2	PSSTAT	W	Port Suspend Status 0 = Port not suspended 1 = Port suspended Set and cleared by hardware as controlled through Port Control Register.
1	PESTAT	W	Port Enable Status 0 = Port is disabled 1 = Port is enabled Set and cleared by hardware as controlled through Port Control register.
0	PCSTAT	W	Port Connect Status 0 = No device on this port 1 = Device present on this port Set to 1 for Port 5. Set and cleared by hardware after sampling of connect status at EOF2.

Hub Port State Register

These registers contain the state of the ports' DP and DM pins, which will be sent to the host upon receipt of a Get_Bus_State request.

Register Address:

0x1FA8, PSTATE1 Port1

0x1FA9, PSTATE2 Port2

0x1FAA, PSTATE3 Port3

0x1FAB, PSTATE4 Port4

0x1FAC, PSTATE5 Port5

Default State: 00000000b

Hub Port State Register

Bit	Mnemonic	USB Hardware	Function
7 - 6			Reserved. Set to 0.
1	DPSTATE	W	DP State. Value of DP at last EOF Set and cleared by hardware at EOF2 Set to 1 for Port 5.
0	DMSTATE	W	DM State. Value of DM at last EOF Set and cleared by hardware at EOF2 Set to 1 for Port 5.

Hub Port Status Change Register

Register Address:

0x1FB0 HPSTCH1 Port1

0x1FB1 HPSTCH2 Port2

0x1FB2 HPSTCH3 Port3

0x1FB3 HPSTCH4 Port4

0x1FB4 HPSTCH5 Port5

Default State: 00000000b

The microcontroller firmware uses the bits in this register to monitor when a port status change has occurred, which then gets reported to the host through the Port Change Field, wPortChange. If any one of the bits in this register is set, the firmware must set the corresponding port's change bit in the Hub Data Register 1, which holds the hub and port status change bitmap.

Except for bit 5, the Port Overcurrent Indicator Change, the bits in this register are set by the USB hardware. Otherwise, the firmware should only clear these bits..

Hub Port Status Change Register

Bit	Mnemonic	USB Hardware	Function
7:5			Reserved. Set to 0s.
4	RSTSC	W	Port Reset Status Change 0 = No change 1 = Reset complete
3	POCIC	R	Port Overcurrent Indicator Change 0 = No change has occurred on Overcurrent Indicator 1 = Overcurrent Indicator has changed
2	PSSC	W	Port Suspend Status Change 0 = No change 1 = Resume completed Cleared by firmware via host request. Ports 1 - 4 set by hardware upon completion of firmware initiated resume process. Port 5 set by firmware 20ms after the next EOF2 after completion of resume process.
1	PESC	W	Port Enable/Disable Status Change 0 = No change has occurred on Port Enable/Disable Status 1 = Port Enable/Disable status has changed Set by hardware due to babble, physical disconnect or overcurrent. Cleared by host request except for Port 5, in which case it is set by hardware at EOF2 due to hardware events.
0	PCSC	W	Port Connect Status Change 0 = No change has occurred on Current Connect Status 1 = Current Connect Status has changed Cleared by firmware via Host request. This bit is set by hardware at EOF2 except for Port 5, where it is set at the next EOF2 after completion of a hub reset.

- **RSTSC, Reset Status Change Bit**

Set by hardware after it completes RESET signaling, which is initiated when the Reset and Enable Port command is detected at the Port Control Register, HPCON. Firmware sends this command when it decodes a SetPortFeature (PORT_RESET) request from the host.

At EOF2 after hardware completes the port reset, hardware sets the Port Enable Status bit and clears the Port Reset Status bit of the Hub Port Status Register, HPSTAT.

Cleared by firmware, ClearPortFeature (PORT_RESET).

- **Port Overcurrent Indicator Change Bit**

Applies to hubs with individual overcurrent reporting only firmware sets this bit as a result of detecting overcurrent at the ports $\overline{OVC}$ pin Firmware clears bit through ClearPortFeature (PORT_OVER_CURRENT).

For Port 5, this bit is always cleared.

- **Port Suspend Status Change Bit (= Resume Complete)**

Set by hardware after hardware completes RESUME signaling.

RESUME signaling is initiated through global resume, selective resume, remote wake-up.

Cleared by firmware through ClearPortFeature (PORT_SUSPEND).

- **Port Enable/Disable Status Change Bit**

Set by hardware at EOF2 due to babble, overcurrent or physical disconnect. For Port 5, set only for babble. Cleared by firmware through ClearPortFeature (PORT_ENABLE).

- **Port Connect Status Change Bit**

Set by hardware at EOF2 after it detects a connect or disconnect at a port, except at Port 5. Hardware sets this bit for Port 5 after a hub reset. Cleared by firmware through ClearPortFeature (PORT_CONNECTION).

SPRSR: Suspend/Resume Register

Register Address: 0x1FFA

Default State: xxxxx000b

SPRSR: Suspend/Resume Register

Bit	Field	USB Hardware	Description
7 - 3			Reserved
2	FRWUP	W	Function Remote Wake-up. The USB hardware sets this bit to signal that an external interrupt causes a remote wake-up. This remote wake-up should be assigned to a peripheral function.
1	RSM	W	Resume. The USB hardware sets this bit when a USB resume signaling is detected at any of its port except Port 5.
0	GLB SUSP	W	Global Suspend. The USB hardware sets this bit when a USB global suspend signaling is detected.

The GLB SUSP, RSM and FRWUP bits can be programmed to cause an interrupt to the microcontroller. This is achieved by setting the corresponding bits of the Suspend/Resume Interrupt Enable Register.

SPRSIE: Suspend/Resume Interrupt Enable Register

Register Address: 0x1FF9

Default State: xxxxx000b

SPRSIE: Suspend/Resume Interrupt Enable Register

Bit	Field	USB Hardware	Description
7 - 3			Reserved
2	FRWUP IE	R	Enable Function Remote Wake-up Interrupt 1 = enable 0 = disable
1	RSM IE	R	Enable Resume Signaling Interrupt 1 = enable 0 = disable
0	GLB SUSP IE	R	Enable Global Suspend Signaling Interrupt 1 = enable 0 = disable

Electrical Specification

Absolute Maximum Ratings*

Operating Temperature	-55°C to + 125°C
Storage Temperature	-65°C to + 150°C
Voltage on Any Pin with Respect to Ground	-1.0V to + 5.25V
Maximum Operating Voltage	6.6V
DC Output Current	16 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = 0°C to 85°C, V_{CC} = 4.4 to 5.25V, unless otherwise noted.

Power Supply

Symbol	Parameter	Min	Max	Unit
V_{CC}	5V Power Supply	4.4	5.25	V
I_{CC}	5V Supply Current		40.0	mA
I_{CCS}	Suspended Device Current		300.0	µA

USB Signals: DPx, DMx

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	Input Level High (Driven)		2.0		V
V_{IHZ}	Input Level High (Floating)		2.7		V
V_{IL}	Input Level Low			0.8	V
V_{DI}	Differential Input Sensitivity	DPx and DMx	0.2		V
V_{CM}	Differential Common Mode Range		0.8	2.5	V
V_{OL1}	Static Output Low	R_L of 1.5 kΩ to 3.6V		0.3	V
V_{OH1}	Static Output High	R_L of 15 kΩ to GND	2.8	3.6	V
V_{CRS}	Output Signal Crossover		1.3	2.0	V
C_{IN}	Input Capacitance			20.0	pF

Non-USB Signals

Symbol	Parameter	Condition	Min	Max	Unit
V_{OL2}	Output Low Level, PA, PB, PC, PD	I_{OL} = 4 mA		0.5	V
V_{OH2}	Output High Level, PA, PB, PC, PD	I_{OH} = 4 mA	V_{CEXT} - 0.5		V
V_{IL2}	Input Low Level, PA, PB, PC, PD			0.3 V_{CEXT}	V
V_{IH2}	Input High Level, PA, PB, PC, PD		0.7 V_{CEXT}		V
$C_{IN/OUT}$	Input/Output Capacitance	1 MHz		10.0	pF
V_{IL3}	Input Low Level, D[0:15]			0.3 V_{CEXT}	V
V_{IH3}	Input High Level, D[0:15]		0.7 V_{CEXT}		V
C_{IN}	Input Capacitance	1 MHz		10.0	pF
V_{OL4}	Output Low Level, A[0:15]	I_{OL} = 2 mA		0.5	V
V_{OH4}	Output High Level, A[0:15]	I_{OH} = 2 mA		V_{CEXT} - 0.5	V
C_{OUT}	Output Capacitance	1 MHz		10.0	pF

Note: V_{CEXT} is voltage at CEXT1, CEXT2 pins.

Oscillator Signals: XTAL1, XTAL2

Symbol	Parameter	Condition	Min	Max	Unit
V_{LH}	XTAL1 Switching Level		0.47	1.20	V
V_{HL}	XTAL2 Switching Level		0.67	1.44	V
C_{X1}	Input Capacitance, XTAL1			10.0	pF
C_{X2}	Output Capacitance, XTAL2			10.0	pF
C_{12}	Equivalent Load Capacitance			5.0	pF
t_{SU}	Start-up Time	6 MHz, Fundamental		2.0	ms
D_L	Drive Level	3.6V, 6 MHz crystal, 40Ω Equiv Series Resistor		50.0	µW

Note: XTAL2 must not be used to drive other circuitry.

AC Characteristics

DPx, DMx Driver Characteristics, Full-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
t_R	Rise Time	C_L = 50 pF	4	20	ns
t_F	Fall Time	C_L = 50 pF	4	20	ns
t_{RFM}	t_R/t_F Matching		90	110	%
z_{DRV}	Driver Output Resistance[1]	Steady State Drive	28	44	Ω

Note: 1. With external 22Ω series resistor.

DP0, DM0 Source Timings, Full-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
t_{DRATE}	Full-speed Data Rate[1]	Average Bit Rate	11.97	12.03	Mbs
t_{FRAME}	Frame Interval[1]		0.9995	1.0005	ms
t_{RFI}	Consecutive Frame Interval Jitter[1]	No Clock Adjustment		42	ns
t_{RFIADJ}	Consecutive Frame Interval Jitter[1]	With Clock Adjustment		126	ns
t_{DJ1} t_{DJ2}	Source Differential Driver Jitter to Next Transition for Paired Transitions		-2 -1	2 1	ns ns
t_{FDEOP}	Source Jitter for Differential Transition to SEO Transitions		-2	5	ns
t_{DEOP}	Differential to EOP Transition Skew		-2	5	ns
t_{JR1} t_{JR2}	Recvr Data Jitter Tolerance to Next Transition for Paired Transitions		-18.5 -9	18.5 9	ns ns
t_{FEOPt}	Source SEO Interval of EOP		160	175	ns
t_{EOPR}	Receiver SEO Interval of EOP		82		ns
t_{FST}	Width of SEO Interval during Differential Transition			14	ns

Note: 1. With 6.000 MHz, 100 ppm crystal.

DPx, DMx Driver Characteristics, Low-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
t_R	Rise Time	C_L = 200 - 600 pF	75	300	ns
t_F	Fall Time	C_L = 200 - 600 pF	75	300	ns
t_{RFM}	t_R/t_F Matching		80	125	%

DPx, DMx Hub Timings, High-speed Operation

Symbol	Parameter	Min	Max	Unit
t_{HDD2}	Hub Differential Data Delay without Cable		44	ns
t_{HDJ1} t_{HDJ2}	Hub Differential Driver Jitter to Next Transition for Paired Transitions	-3 -1	3 1	ns ns
t_{FSOP}	Data Bit Width Distortion after SOP	-5	5	ns
t_{FEOPD}	Hub EOP Delay Relative to t_{HDD}	0	15	ns
t_{FHESK}	Hub EOP Output Width Skew	-15	15	ns

DPx, DMx Hub Timings, Low-speed Operation

Symbol	Parameter	Min	Max	Unit
t_{LHDD}	Hub Differential Data Delay		300	ns
t_{LHDJ1} t_{LHDJ2} t_{LUHJ1} t_{LUHJ2}	Downstream Hub Differential Driver Jitter to Next Transition, Downstream for Paired Transitions, Downstream to Next Transition, Upstream for Paired Transitions, Upstream	-45 -15 -45 -45	45 15 45 45	ns ns ns ns
t_{SOP}	Data Bit Width Distortion after SOP	-60	60	ns
t_{LEOPD}	Hub EOP Delay Relative to t_{HDD}	0	200	ns
t_{LHESK}	Hub EOP Output Width Skew	-300	300	ns

Hub Event Timings

Symbol	Parameter	Condition	Min	Max	Unit
t_{DCNN}	Time to Detect a Downstream Port Connect Event Awake Hub Suspended Hub		2.5 2.5	2000 12000	µs µs
t_{DDIS}	Time to Detect a Disconnect Event on Downstream Port Awake Hub Suspended Hub		2.5 2.5	2000 10000	µs µs
t_{URSM}	Time from Detecting Downstream Resume to Rebroadcast			100	µs
t_{DRST}	Duration of Driving Reset to a Downstream Device	Only for a SetPortFeature (PORT_RESET) request	10	20	ms
t_{URLK}	Time to Detect a Long “K” from Upstream		2.5	5.5	µs
t_{URLSEO}	Time to Detect a Long SEO from Upstream		2.5	5.5	µs

Timing Waveforms

Figure 10. Data Signal Rise and Fall Time

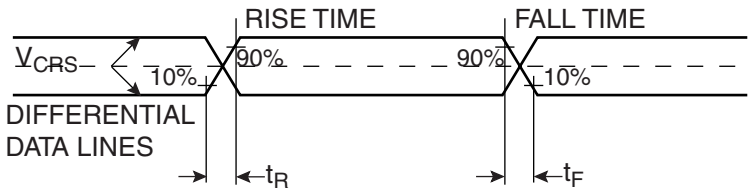


Figure 11. Full-speed Load

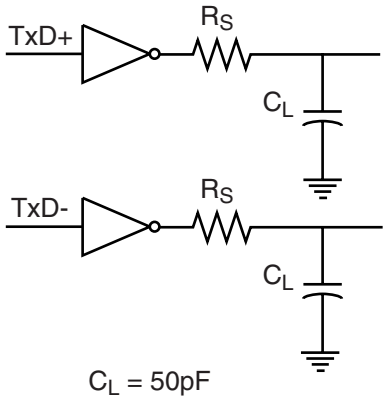


Figure 12. Low-speed Downstream Port Load

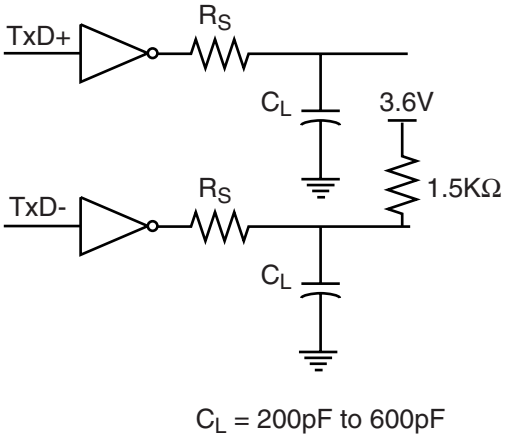


Figure 13. Differential Data Jitter

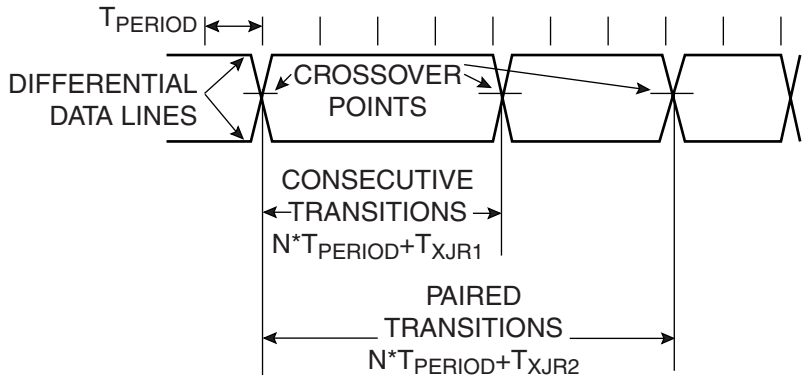


Figure 14. Differential-to-EOP Transition Skew and EOP Width

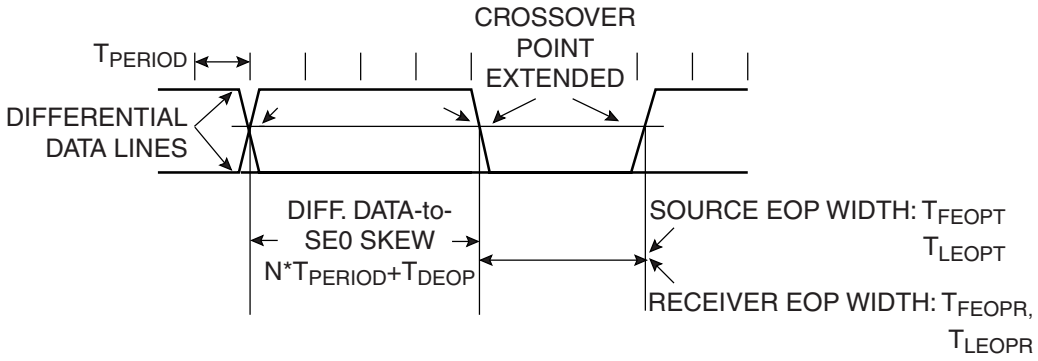


Figure 15. Receiver Jitter Tolerance

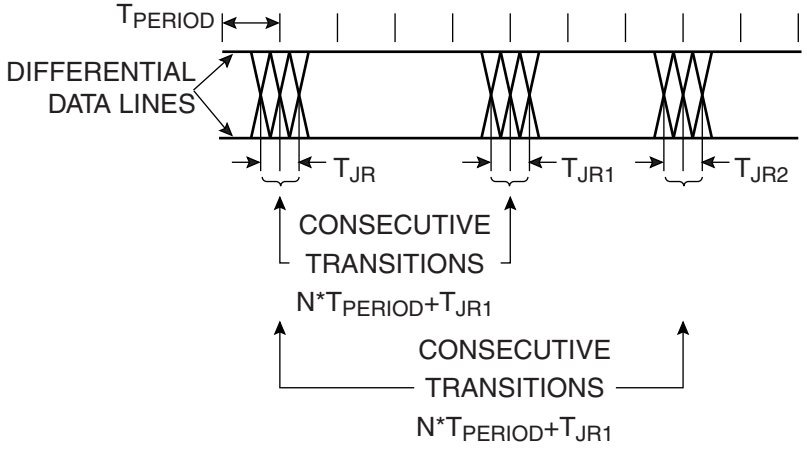
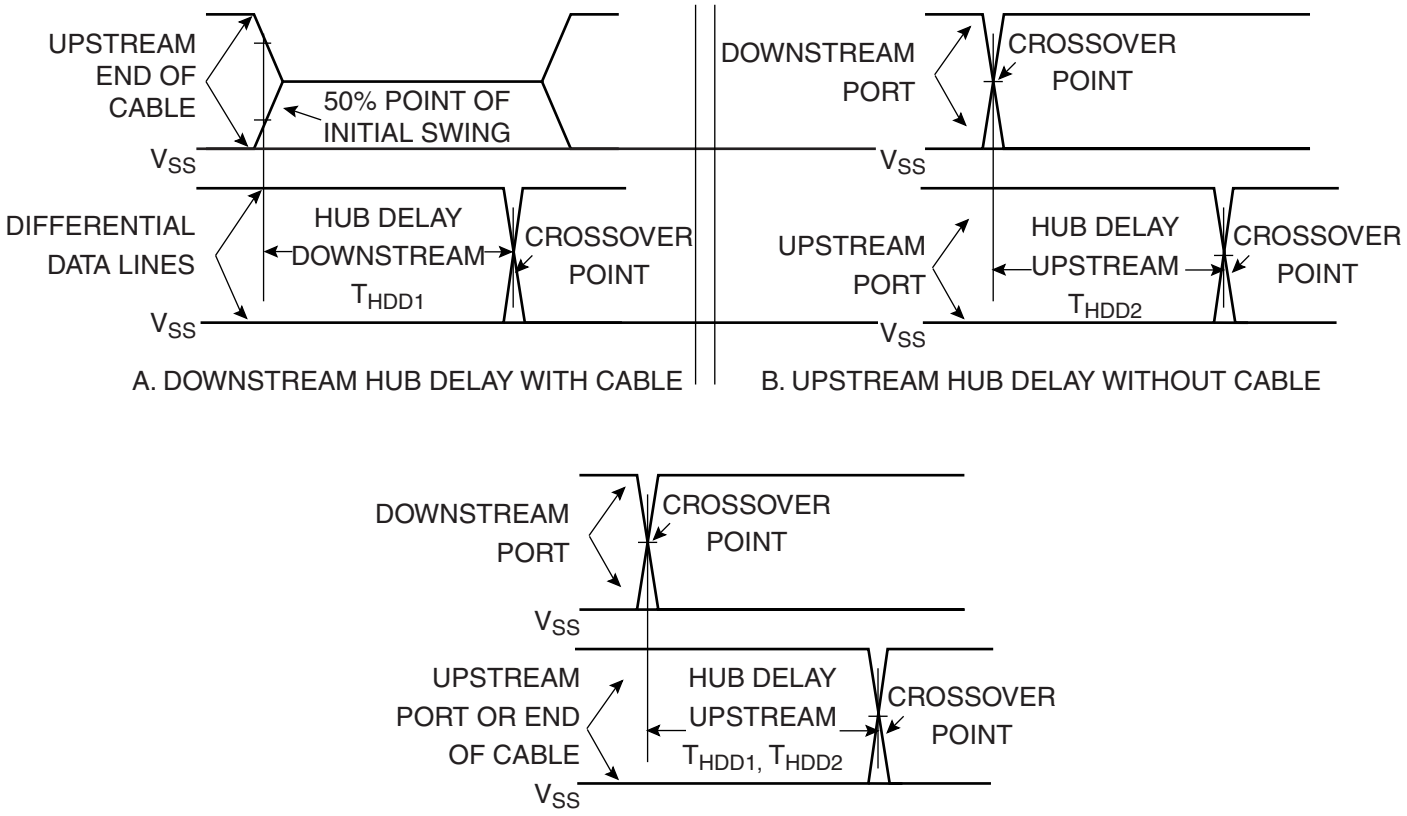


Figure 16. Hub Differential Delay, Differential Jitter and SOP Distortion



Hub Differential Jitter:

$T_{HDJ1} = T_{HDDX}(J) - T_{HDDX}(K)$ or $T_{HDDX}(K) - T_{HDDX}(J)$ Consecutive Transitions

$T_{HDJ2} = T_{HDDX}(J) - T_{HDDX}(J)$ or $T_{HDDX}(K) - T_{HDDX}(K)$ Paired Transitions

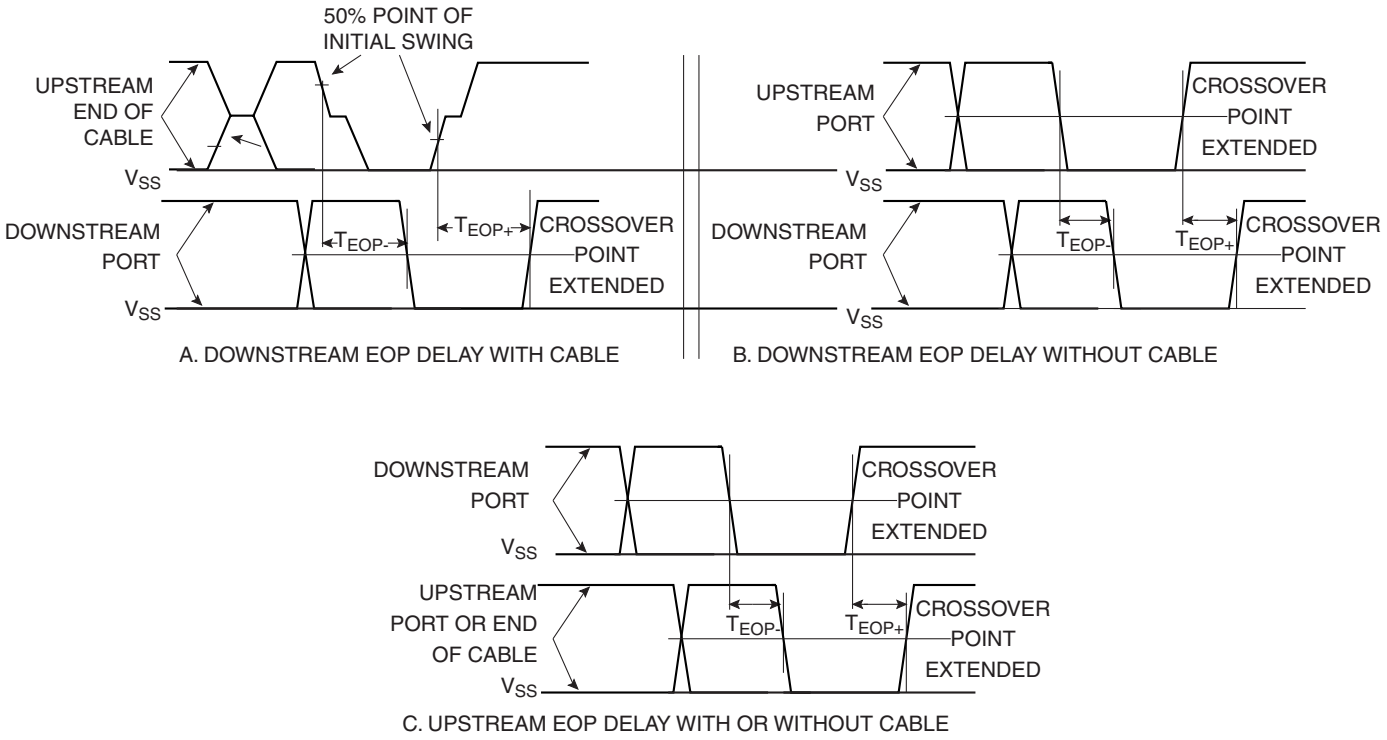
Bit after SOP Width Distortion (same as Data Jitter for SOP and Next J Transition):

$T_{SOP} = T_{HDDX}(NEXTJ) - T_{HDDX}(SOP)$

Low-speed timings are determined in the same way for:

T_{LHDD}, T_{LDHJ1}, T_{LDJH2}, T_{LUHJ1}, T_{LUJH2}, and T_{LSOP}

Figure 17. Hub EOP Delay and EOP Skew



EOP Delay:

$T_{EOPD} = T_{EOP} - T_{HDDX}$

EOP Skew:

$T_{HESK} = T_{EOP+} - T_{EOP-}$

Low-speed timings are determined in the same way for:

T_{LEOPD} and T_{LHESK}

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

1443B–01/01/xM