SAA1160A

Embedded USB host controller

Rev. 01 — 20 May 2010 **Product data sheet**

1. General description

The SAA1160A is an embedded Universal Serial Bus (USB) Host Controller (HC) that complies with *Universal Serial Bus Specification Rev. 2.0*, supporting data transfer at full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s). The SAA1160A provides two downstream ports. Each downstream port has an OverCurrent (OC) detection input pin and power supply switching control output pin. The downstream ports for the HC can be connected with any USB compliant USB devices and USB hubs that have USB upstream ports.

The SAA1160A is well suited for embedded systems and portable devices that require a USB host. The SAA1160A brings high flexibility to the systems that have it built in. For example, a system that has the SAA1160A built in allows it to be connected to a device that has a USB upstream port, such as a USB printer, USB camera, USB keyboard, USB mouse, among others.

2. Features and benefits

- Complies with *Universal Serial Bus Specification Rev. 2.0*
- Supports data transfer at full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s)
- Adapted from *Open Host Controller Interface Specification for USB Release 1.0a*
- Selectable one or two downstream ports for HC
- High-speed parallel interface to most of the generic microprocessors and Reduced Instruction Set Computer (RISC) processors such as:
  - Hitachi SuperH SH-3 and SH-4
  - MIPS-based RISC
  - ARM7, ARM9 and StrongARM
- Maximum 15 MB/s data transfer rate between the microprocessor and the HC
- Supports single-cycle and burst mode DMA operations
- Built-in FIFO buffer RAM for the HC (4 kB)
- Endpoints with double buffering to increase throughput and ease real-time data transfer for ISOchronous (ISO) transactions
- 6 MHz crystal oscillator with integrated PLL for low EMI
- Built-in software selectable internal 15 kΩ pull-down resistors for HC downstream ports
- Dedicated pins for suspend sensing output and wake-up control input for flexible applications
- Operation at either +5 V or +3.3 V power supply voltage
- Operating temperature range from –40 °C to +85 °C
- Available in an LQFP64 package (SOT314-2)

3. Applications

- Personal Digital Assistant (PDA)
- Digital camera
- Third-generation (3-G) phone
- Set-Top Box (STB)
- Information Appliance (IA)
- Photo printer
- MP3 jukebox
- Game console

4. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
SAA1160AHL	LQFP64	plastic low profile quad flat package; 64 leads; body 10 × 10 × 1.4 mm	SOT314-2

5. Block diagram

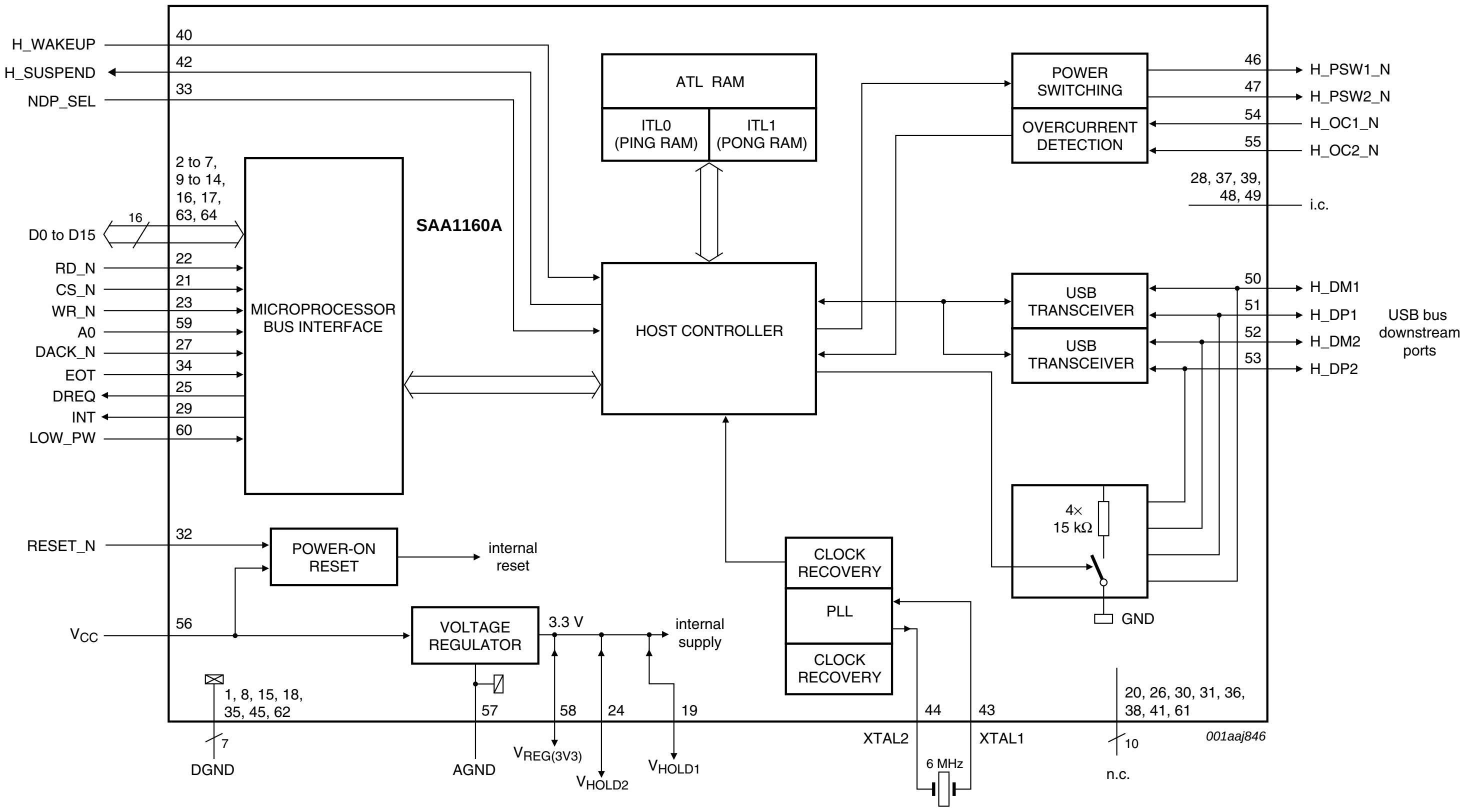


Fig 1. Block diagram

6. Pinning information

6.1 Pinning

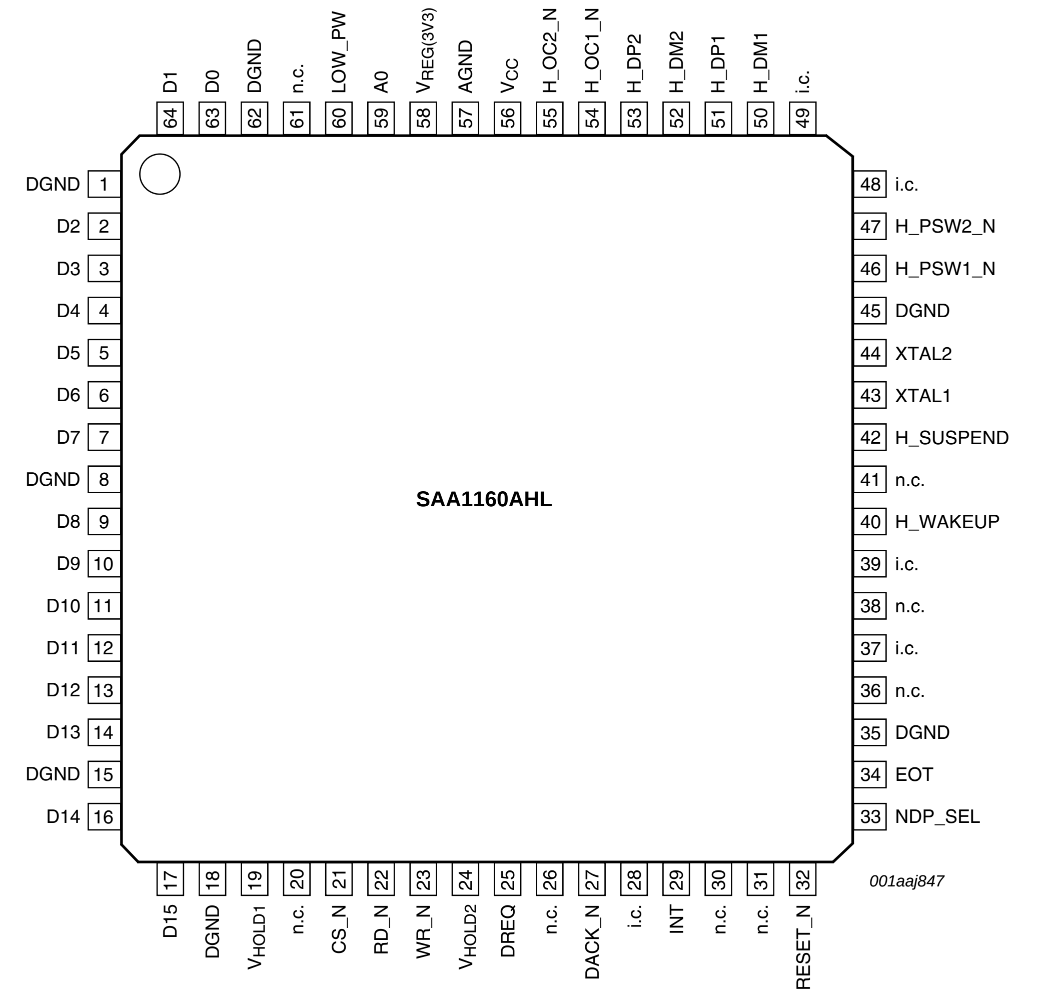


Fig 2. Pin configuration

6.2 Pin description

Table 2. Pin description

Symbol[1]	Pin	Type	Description
DGND	1	-	digital ground
D2	2	I/O	bit 2 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D3	3	I/O	bit 3 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D4	4	I/O	bit 4 of bidirectional data; slew-rate controlled; TTL input; 3-state output

Table 2. Pin description *...continued*

Symbol[1]	Pin	Type	Description
D5	5	I/O	bit 5 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D6	6	I/O	bit 6 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D7	7	I/O	bit 7 of bidirectional data; slew-rate controlled; TTL input; 3-state output
DGND	8	-	digital ground
D8	9	I/O	bit 8 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D9	10	I/O	bit 9 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D10	11	I/O	bit 10 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D11	12	I/O	bit 11 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D12	13	I/O	bit 12 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D13	14	I/O	bit 13 of bidirectional data; slew-rate controlled; TTL input; 3-state output
DGND	15	-	digital ground
D14	16	I/O	bit 14 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D15	17	I/O	bit 15 of bidirectional data; slew-rate controlled; TTL input; 3-state output
DGND	18	-	digital ground
V_{HOLD1}	19	I/O	voltage holding pin 1; internally connected to the $V_{REG(3V3)}$ and V_{HOLD2} pins. When V_{CC} is connected to 5 V, this pin will output 3.3 V, hence do not connect it to 5 V. When V_{CC} is connected to 3.3 V, this pin can either be connected to 3.3 V or left unconnected. In **all** cases, decouple this pin to DGND.
n.c.	20	-	not connected; leave this pin open
CS_N	21	I	chip select input
RD_N	22	I	read strobe input
WR_N	23	I	write strobe input
V_{HOLD2}	24	I/O	voltage holding pin 2; internally connected to the $V_{REG(3V3)}$ and V_{HOLD1} pins. When V_{CC} is connected to 5 V, this pin will output 3.3 V, hence do not connect it to 5 V. When V_{CC} is connected to 3.3 V, this pin can either be connected to 3.3 V or left unconnected. In **all** cases, decouple this pin to DGND.
DREQ	25	O	HC DMA request output (programmable polarity); signals to the DMA controller that the SAA1160A wants to start a DMA transfer; see Section 10.4.1
n.c.	26	-	not connected; leave this pin open
DACK_N	27	I	HC DMA acknowledge input; when not in use, this pin must be connected to V_{CC} via an external 10 kΩ resistor
i.c.	28	-	internally connected; connect to supply[2]

Table 2. Pin description *…continued*

Symbol[1]	Pin	Type	Description
INT	29	O	HC interrupt output; programmable level, edge triggered and polarity; see Section 10.4.1
n.c.	30	-	not connected; leave this pin open
n.c.	31	O	not connected; leave this pin open
RESET_N	32	I	reset input (Schmitt trigger); a LOW level produces an asynchronous reset (internal pull-up resistor)
NDP_SEL	33	I	indicates to the HC software the Number of Downstream Ports (NDP) present: **0 —** select 1 downstream port **1 —** select 2 downstream ports only changes the value of the NDP field in the HcRhDescriptorA register; both ports will always be enabled; see Section 10.3.1 (internal pull-up resistor)
EOT	34	I	DMA master device to inform the SAA1160A of end of DMA transfer; active level is programmable; when not in use, this pin must be connected to V_{CC} via an external 10 kΩ resistor; see Section 10.4.1
DGND	35	-	digital ground
n.c.	36	-	not connected; leave this pin open
i.c.	37	I	internally connected; connect to ground[3]
n.c.	38	-	not connected; leave this pin open
i.c.	39	-	internally connected; connect to ground[4]
H_WAKEUP	40	I	HC wake-up input; generates a remote wake-up from the suspend state (active HIGH); when not in use, this pin must be connected to DGND via an external 10 kΩ resistor (internal pull-down resistor)
n.c.	41	-	not connected; leave this pin open
H_SUSPEND	42	O	HC suspend state indicator output; active HIGH
XTAL1	43	I	crystal input; connected directly to a 6 MHz crystal; when this pin is connected to an external clock source, pin XTAL2 must be left open
XTAL2	44	O	crystal output; connected directly to a 6 MHz crystal; when pin XTAL1 is connected to an external clock source, this pin must be left open
DGND	45	-	digital ground
H_PSW1_N	46	O	power switching control output for downstream port 1; open-drain output
H_PSW2_N	47	O	power switching control output for downstream port 2; open-drain output
i.c.	48	-	internally connected; connect to ground[5]
i.c.	49	-	internally connected; connect to supply[6]
H_DM1	50	AI/O	USB D– data line for HC downstream port 1
H_DP1	51	AI/O	USB D+ data line for HC downstream port 1
H_DM2	52	AI/O	USB D– data line for HC downstream port 2; when not in use, this pin must be left open

Table 2. Pin description *…continued*

Symbol[1]	Pin	Type	Description
H_DP2	53	AI/O	USB D+ data line for HC downstream port 2; when not in use, this pin must be left open
H_OC1_N	54	I	overcurrent sensing input for HC downstream port 1
H_OC2_N	55	I	overcurrent sensing input for HC downstream port 2
V_{CC}	56	-	digital power supply input. This pin supplies the internal regulator output to pins $V_{REG(3V3)}$, V_{HOLD1} and V_{HOLD2}. When connected to $V_{REG(3V3)}$, it will bypass the internal regulator.
AGND	57	-	analog ground
$V_{REG(3V3)}$	58	I/O	internal 3.3 V regulator output; when pin V_{CC} is connected to 5 V, this pin outputs 3.3 V. When pin V_{CC} is connected to 3.3 V, connect this pin to 3.3 V.
A0	59	I	address input; selects command (A0 = 1) or data (A0 = 0)
LOW_PW	60	I	The LOW_PW pin is used to select auxiliary registers. If low-current consumption is not needed during suspend, this pin may be connected to DGND.
n.c.	61	-	not connected; leave this pin open
DGND	62	-	digital ground
D0	63	I/O	bit 0 of bidirectional data; slew-rate controlled; TTL input; 3-state output
D1	64	I/O	bit 1 of bidirectional data; slew-rate controlled; TTL input; 3-state output

[1] Symbol names ending with underscore N (for example, NAME_N) represent active LOW signals.

[2] This pin must be connected to V_{CC} via an external 10 kΩ resistor.

[3] This pin must be connected to DGND via an external 10 kΩ resistor.

[4] This pin must be connected to DGND via an external 1 MΩ resistor.

[5] This pin must be connected to AGND via an external 15 kΩ resistor. If low-current consumption is not needed during suspend, this pin may be left open.

[6] This pin must be connected to $V_{REG(3V3)}$ via an external 1.5 kΩ resistor. If low-current consumption is not needed during suspend, this pin may be left open.

7. Functional description

7.1 PLL clock multiplier

A 6 MHz to 48 MHz clock multiplier Phase-Locked Loop (PLL) is integrated on-chip. This allows for the use of a low-cost 6 MHz crystal, which also minimizes EMI. No external components are required for the operation of the PLL.

7.2 Bit clock recovery

The bit clock recovery circuit recovers the clock from the incoming USB data stream by using a 4 times oversampling principle. It is able to track jitter and frequency drift as specified in *Universal Serial Bus Specification Rev. 2.0.*

7.3 Analog Transceivers (ATX)

Two sets of transceivers are embedded in the chip for downstream ports with USB connector type A. The integrated transceivers are compliant with the *Universal Serial Bus Specification Rev. 2.0*. These transceivers interface directly with the USB connectors and cables through external termination resistors.

7.4 NXP Serial Interface Engine (SIE)

The NXP SIE implements the full USB protocol layer. It is completely hardwired for speed and needs no firmware intervention. The functions of this block include: synchronization pattern recognition, parallel to serial conversion, bit (de)stuffing, CRC checking and generation, Packet IDentifier (PID) verification and generation, address recognition, and handshake evaluation and generation.

8. Microprocessor bus interface

8.1 Programmed I/O (PIO) addressing mode

A generic PIO interface is defined for speed and ease-of-use. It also allows direct interfacing to most microcontrollers. To a microcontroller, the SAA1160A appears as a memory device with a 16-bit data bus and uses the A0 address line to access internal control registers and FIFO buffer RAM. Therefore, the SAA1160A occupies only two I/O ports or two memory locations of a microprocessor (see Table 3). External microprocessors can read from or write to the SAA1160As internal control registers and FIFO buffer RAM through the PIO operating mode. Figure 3 shows the PIO interface between a microprocessor and the SAA1160A.

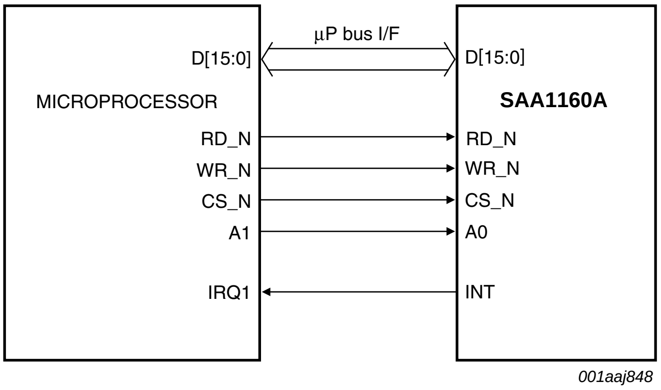


Fig 3. Programmed I/O interface between a microprocessor and the SAA1160A

8.2 DMA mode

The SAA1160A also provides the DMA mode for external microprocessors to access its internal FIFO buffer RAM. Data can be transferred by the DMA operation between a microprocessors system memory and the SAA1160As internal FIFO buffer RAM.

Remark: The DMA operation must be controlled by the external microprocessor systems DMA controller (Master).

Figure 4 shows the DMA interface between a microprocessor system and the SAA1160A. The SAA1160A provides a DMA channel controlled by DREQ for DACK_N signals for the DMA transfer between a microprocessors system memory and the SAA1160A HCs internal FIFO buffer RAM.

The EOT signal is an external end-of-transfer signal used to terminate the DMA transfer. Some microprocessors may not have this signal. In this case, the SAA1160A provides an internal EOT signal to terminate the DMA transfer as well. Setting the HcDMAConfiguration register (21H to read, A1H to write) enables the SAA1160As HC internal DMA counter for the DMA transfer. When the DMA counter reaches the value set in the HcTransferCounter register (22H to read, A2H to write), an internal EOT signal will be generated to terminate the DMA transfer.

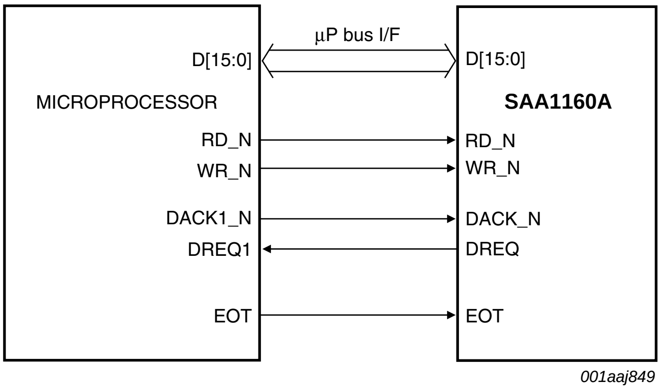


Fig 4. DMA interface between a microprocessor and the SAA1160A

8.3 Control registers access by PIO mode

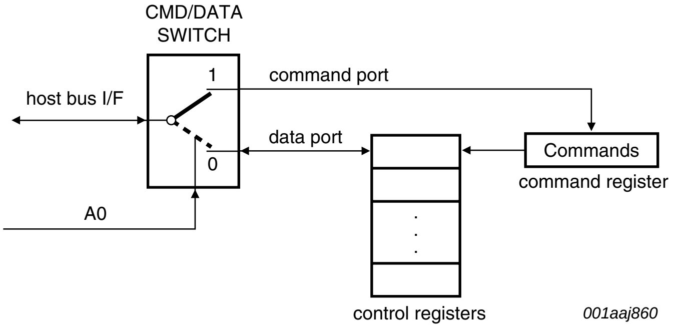
8.3.1 I/O port addressing

Table 3 shows the SAA1160As I/O port addressing. Complete decoding of the I/O port address should include the chip select signal CS_N and the address line A0. However, the direction of access of I/O ports is controlled by the RD_N and WR_N signals. When RD_N is LOW, the microprocessor reads data from the SAA1160As data port. When WR_N is LOW, the microprocessor writes a command to the command port, or writes data to the data port.

Table 3. I/O port addressing

Port	CS_N	LOW_PW	A0	Access	Data bus width (bits)	Description
0	0	0	0	R/W	16	HC data port
1	0	0	1	W	16	HC command port
2	0	1	0	R/W	16	auxiliary data port
3	0	1	1	W	16	auxiliary command port

Figure 5 illustrates how an external microprocessor accesses the SAA1160As internal control registers.



When A0 = 0, microprocessor accesses the data port.

When A0 = 1, microprocessor accesses the command port.

Fig 5. Access to internal control registers

8.3.2 Register access phases

The SAA1160As register structure is a command-data register pair structure. A complete register access cycle comprises a command phase followed by a data phase. The command (also known as the index of a register) points the SAA1160A to the next register to be accessed. A command is 8 bits long. On a microprocessors 16-bit data bus, a command occupies the lower byte, with the upper byte filled with zeros.

Figure 6 shows a complete 16-bit register access cycle for the SAA1160A. The microprocessor writes a command code to the command port, and then reads from or writes the data word to the data port. Take the example of a microprocessor attempting to read the SAA1160As ID, which is saved in the HCs HcChipID register (index 27H, read only). The 16-bit register access cycle is therefore:

1. The microprocessor writes the command code of 27H (0027H in 16-bit width) to the HC command port.
2. The microprocessor reads the data word of the chips ID from the HC data port.

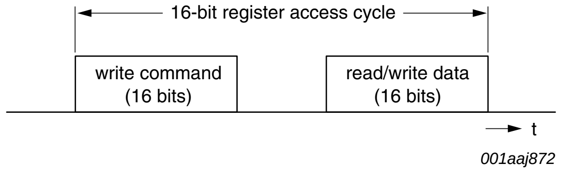


Fig 6. 16-bit register access cycle

Most of the SAA1160As internal control registers are 16-bit wide. Some of the internal control registers, however, are 32-bit wide. Figure 7 shows how the SAA1160As 32-bit internal control register is accessed. The complete cycle of accessing a 32-bit register consists of a command phase followed by two data phases. In the two data phases, the microprocessor first reads or writes the lower 16-bit data, followed by the upper 16-bit data.

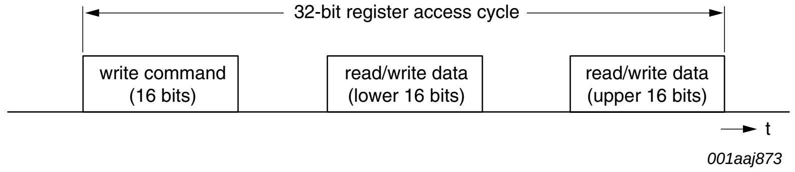
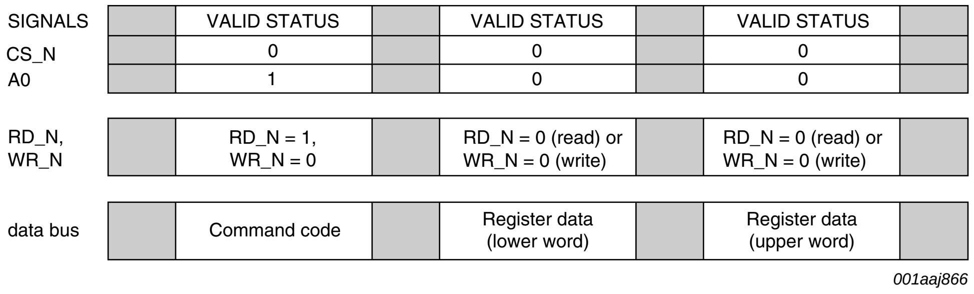


Fig 7. 32-bit register access cycle

To further describe the complete access cycles of the internal control registers, the status of some pins of the microprocessor bus interface are shown in Figure 8.



SIGNALS		VALID STATUS		VALID STATUS		VALID STATUS	
CS_N		0		0		0	
A0		1		0		0	
RD_N, WR_N		RD_N = 1, WR_N = 0		RD_N = 0 (read) or WR_N = 0 (write)		RD_N = 0 (read) or WR_N = 0 (write)	
data bus		Command code		Register data (lower word)		Register data (upper word)	

001aaj866

Fig 8. Accessing HC control registers

8.4 FIFO buffer RAM access by PIO mode

Since the SAA1160As internal memory is structured as a FIFO buffer RAM, the FIFO buffer RAM is mapped to dedicated register fields. Therefore, accessing the internal FIFO buffer RAM is similar to accessing the internal control registers in multiple data phases.

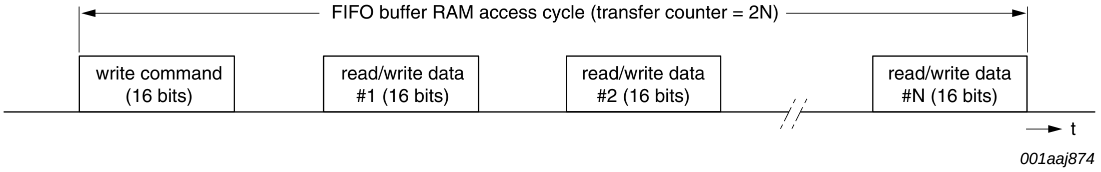

FIFO buffer RAM access cycle (transfer counter = 2N)
write command (16 bits)
read/write data #1 (16 bits)
read/write data #2 (16 bits)
read/write data #N (16 bits)
t
001aaj874

Fig 9. Internal FIFO buffer RAM access cycle

Figure 9 shows a complete access cycle of the HC internal FIFO buffer RAM. For a write cycle, the microprocessor first writes the FIFO buffer RAMs command code to the command port, and then writes the data words one by one to the data port until half of the transfers byte count is reached. The HcTransferCounter register (22H to read, A2H to write) is used to specify the byte count of a FIFO buffer RAMs read cycle or write cycle. Every access cycle must be in the same access direction. The read cycle procedure is similar to the write cycle.

8.5 FIFO buffer RAM access by DMA mode

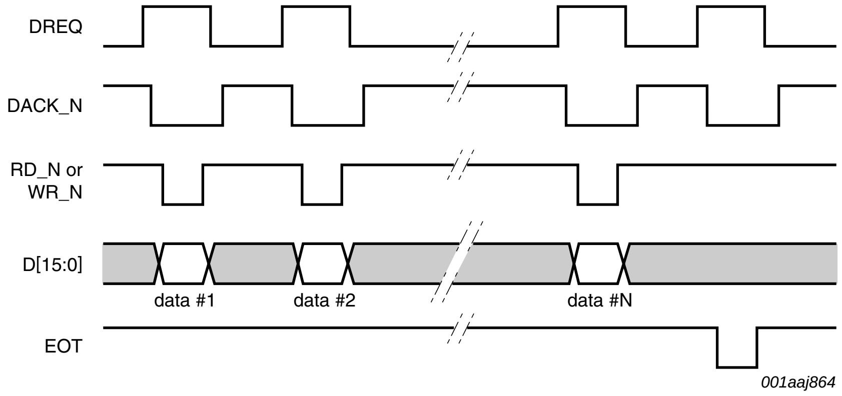
The DMA interface between a microprocessor and the SAA1160A is shown in Figure 4.

When doing a DMA transfer, at the beginning of every burst the SAA1160A outputs a DMA request to the microprocessor via pin DREQ. After receiving this signal, the microprocessor will reply with a DMA acknowledge to the SAA1160A via pin DACK_N, and at the same time, execute the DMA transfer through the data bus. In the DMA mode, the microprocessor must issue a read or write signal to the SAA1160As pins RD_N or WR_N. The SAA1160A will repeat the DMA cycles until it receives an EOT signal to terminate the DMA transfer.

The SAA1160A supports both external and internal EOT signals. The external EOT signal is received as input on pin EOT, and generally comes from the external microprocessor. The internal EOT signal is generated inside the SAA1160A.

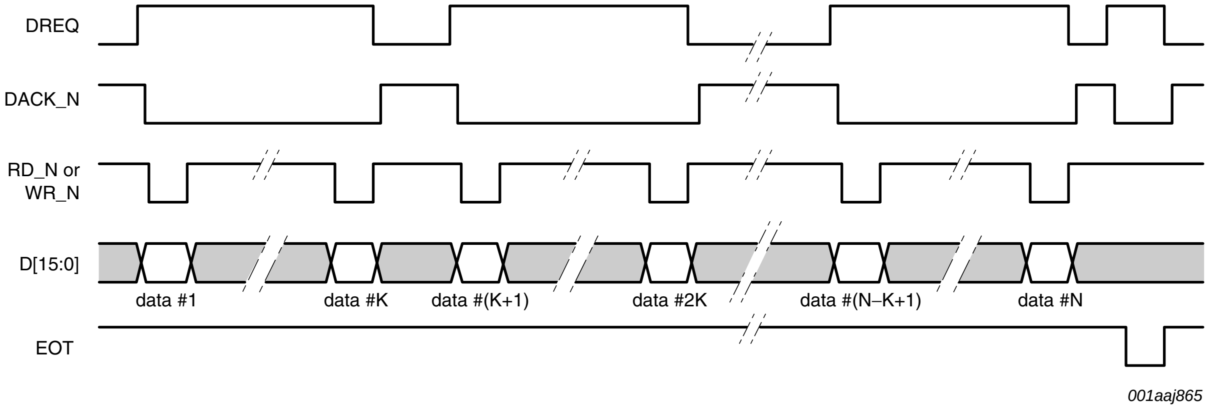
To select either EOT method, set the appropriate DMA configuration register (see Section 10.4.2). For example, setting DMACounterSelect (bit 2) of the HcDMAConfiguration register (21H to read, A1H to write) to logic 1 will enable the DMA counter for DMA transfer. When the DMA counter reaches the value of the HcTransferCounter register, the internal EOT signal will be generated to terminate the DMA transfer.

The SAA1160A supports either single-cycle DMA operation or burst mode DMA operation; see Figure 10 and Figure 11.



N = 1/2 byte count of transfer data.

Fig 10. DMA transfer in single-cycle mode



N = 1/2 byte count of transfer data, K = number of cycles/burst.

Fig 11. DMA transfer in burst mode

In Figure 10 and Figure 11, the DMA transfer is configured such that DREQ is active HIGH and DACK_N is active LOW.

8.6 Interrupts

The SAA1160A has an interrupt request pin INT.

8.6.1 Pin configuration

The interrupt output signals have four configuration modes:

Mode 0	Mode 0 level trigger, active LOW
Mode 1	Mode 1 level trigger, active HIGH
Mode 2	Mode 2 edge trigger, active LOW
Mode 3	Mode 3 edge trigger, active HIGH

Figure 12 shows these four interrupt configuration modes. They are programmable through the HcHardware Configuration register (see Section 10.4.1), which is also used to disable or enable the signals.

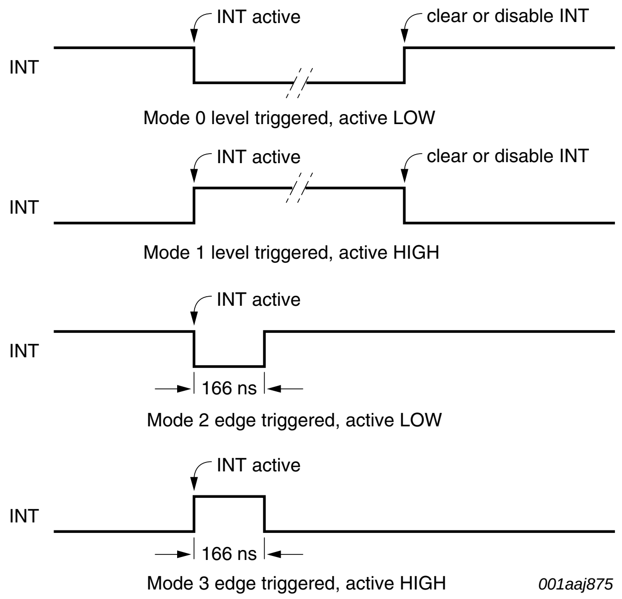

INT active
clear or disable INT
INT
Mode 0 level triggered, active LOW
INT active
clear or disable INT
INT
Mode 1 level triggered, active HIGH
INT active
INT
166 ns
Mode 2 edge triggered, active LOW
INT active
INT
166 ns
Mode 3 edge triggered, active HIGH
001aaj875

Fig 12. Interrupt pin operating modes

8.6.2 Interrupt output pin (INT)

To program the four configuration modes of the HCs interrupt output signal (INT), set InterruptPinTrigger and InterruptOutputPolarity (bits 1 and 2) of the HcHardwareConfiguration register (20H to read, A0H to write). InterruptPinEnable (bit 0) is used as the master enable setting for pin INT.

INT has many associated interrupt events as shown as in Figure 13.

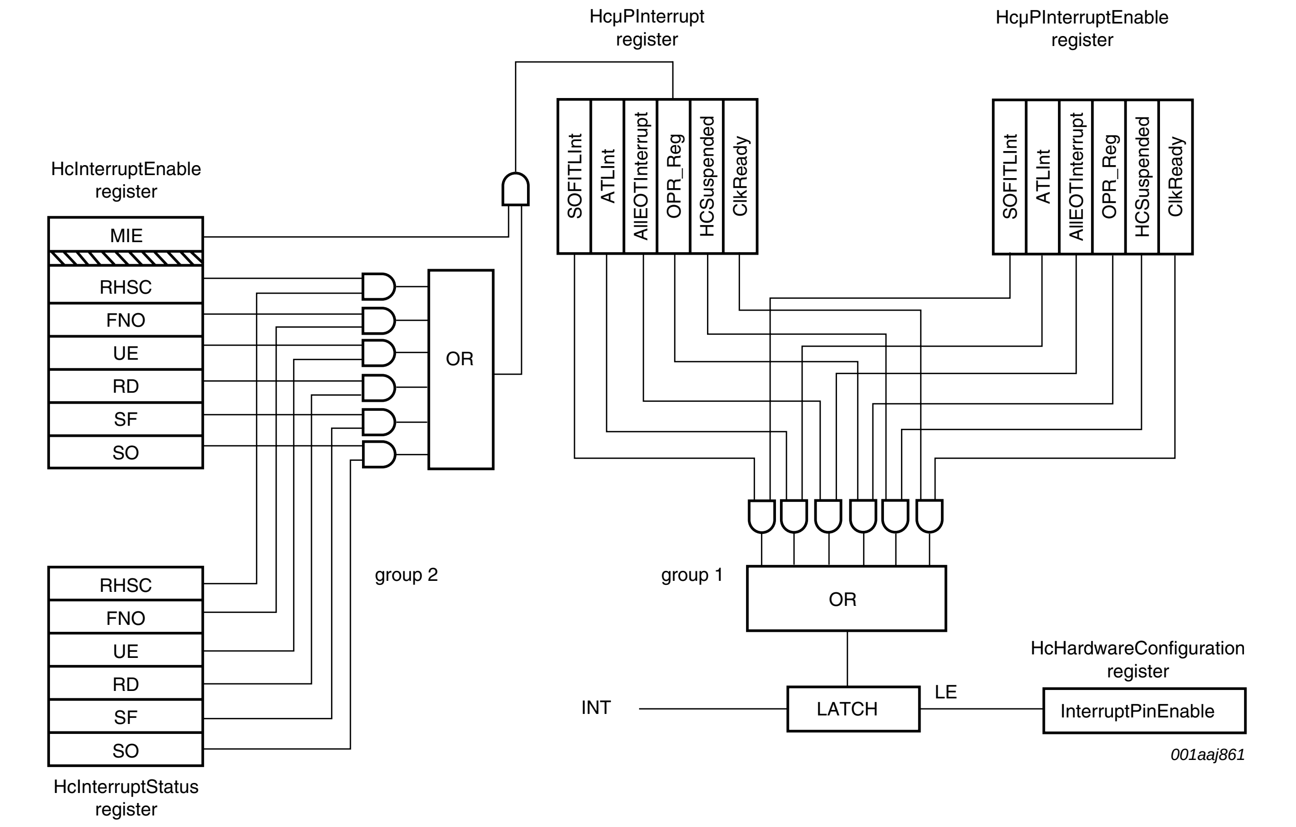


Fig 13. HC interrupt logic

There are two groups of interrupts represented by group 1 and group 2 in Figure 13. A pair of registers control each group.

Group 2 contains six possible interrupt events (recorded in the HcInterruptStatus register). On occurrence of any of these events, the corresponding bit would be set to logic 1; and if the corresponding bit in the HcInterruptEnable register is also logic 1, the 6-input OR gate would output a logic 1. This output is AND-ed with the value of MIE (bit 31 of HcInterruptEnable). Logic 1 at the AND gate will cause the OPR bit in the HcμPInterrupt register to be set to logic 1.

Group 1 contains six possible interrupt events, one of which is the output of group 2 interrupt sources. The HcμPInterrupt and HcμPInterruptEnable registers work in the same way as the HcInterruptStatus and HcInterruptEnable registers in the interrupt group 2. The output from the 6-input OR gate is connected to a latch, which is controlled by InterruptPinEnable (bit 0 of the HcHardwareConfiguration register).

In the event in which the software wishes to temporarily disable the interrupt output of the SAA1160A Host Controller, the following procedure should be followed:

1. Make sure that the InterruptPinEnable bit in the HcHardwareConfiguration register is set to logic 1.
2. Clear all bits in the HcμPInterrupt register.
3. Set the InterruptPinEnable bit to logic 0.

To re-enable the interrupt generation:

1. Set all bits in the HcμPInterrupt register.
2. Set the InterruptPinEnable bit to logic 1.

Remark: The InterruptPinEnable bit in the HcHardwareConfiguration register latches the interrupt output. When this bit is set to logic 0, the interrupt output will remain unchanged, regardless of any operations on the interrupt control registers.

If INT1 is asserted, and the Host Controller Driver (HCD) wishes to temporarily mask off the INT signal without clearing the HcμPInterrupt register, the following procedure should be followed:

1. Make sure that the InterruptPinEnable bit is set to logic 1.
2. Clear all bits in the HcμPInterruptEnable register.
3. Set the InterruptPinEnable bit to logic 0.

To re-enable the interrupt generation:

1. Set all bits in the HcμPInterruptEnable register according to the HCD requirements.
2. Set the InterruptPinEnable bit to logic 1.

9. Host Controller (HC)

9.1 HCs four USB states

The SAA1160As USB HC has four USB states—USBOperational, USBReset, USBSuspend and USBResume—that define the HCs USB signalling and bus states responsibilities. The signals are visible to the Host Controller Driver (HCD) via the SAA1160A USB HCs control registers.

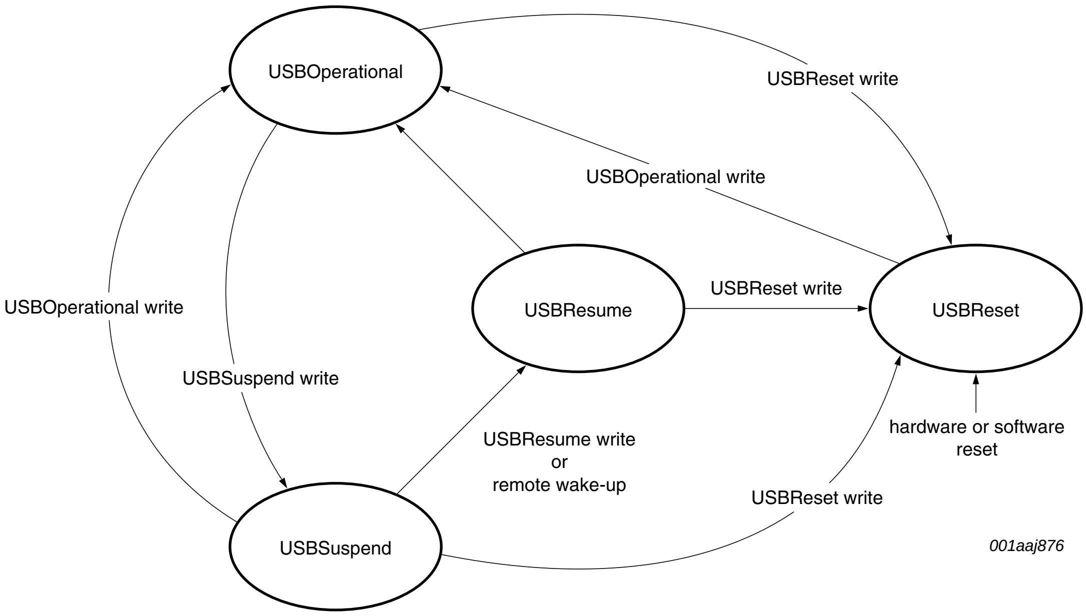


Fig 14. The SAA1160A HCs USB states

The USB states are reflected in the HostControllerFunctionalState field of the HcControl register (01H to read, 81H to write), which is located at bits 7 and 6 of the register.

The HCD can perform only the USB state transitions shown in Figure 14.

Remark: The software reset in Figure 14 is not caused by the HcSoftwareReset command. It is caused by the HostControllerReset field of the HcCommandStatus register (02H to read, 82H to write).

9.2 Generating USB traffic

USB traffic can be generated only when the SAA1160A USB HC is in the USBOperational state. Therefore, the HCD must set the HostControllerFunctionalState field of the HcControl register before generating USB traffic.

A simplistic flow diagram showing when and how to generate USB traffic is shown in Figure 15. For greater accuracy, refer to the *Universal Serial Bus Specification Rev. 2.0* for the USB protocol and the SAA1160A USB HCs register usage.

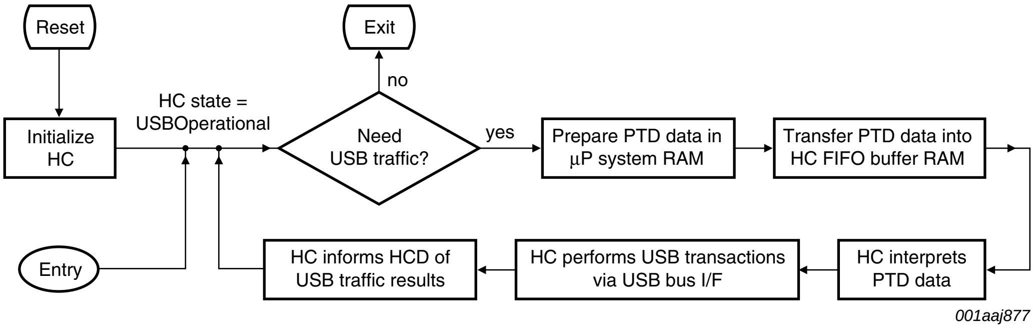


Fig 15. SAA1160A HC USB transaction loop

Description of Figure 15:

1. **Reset**

   This includes hardware reset by pin RESET_N and software reset by the HcSoftwareReset command (A9H). The reset function will clear all the HCs internal control registers to their reset status. After reset (see Section 13), the HCD must initialize the SAA1160A USB HC by setting some registers.

2. **Initialize HC**

   It includes:

   a. Write a value of AA37H to the auxiliary register B0H.

   b. Setting the physical size for the HCs internal FIFO buffer RAM by setting the HcITLBufferLength register (2AH to read, AAH to write) and the HcATLBufferLength register (2BH to read, ABH to write).

   c. Setting the HcHardwareConfiguration register according to requirements.

   d. Clearing interrupt events, if required.

   e. Enabling interrupt events, if required.

   f. Setting the HcFmInterval register (0DH to read, 8DH to write).

   g. Setting the HCs Root Hub registers.

   h. Setting the HcControl register to move the HC into the USBOperational state.

   i. Write a value of 2344H to the auxiliary register BAH.

   j. Write a value of A0H to the auxiliary register B8H.

   k. Write a value of 80H to the auxiliary register B8H.

   See also Section 9.5.

3. **Entry**

   The normal entry point. The microprocessor returns to this point when there are HC requests.

4. **Need USB traffic**

   USB devices need the HC to generate USB traffic when they have USB traffic requests such as:

   a. Connecting to or disconnecting from downstream ports.

   b. Issuing the Resume signal to the HC.

   To generate USB traffic, the HCD must enter the USB transaction loop.

5. **Prepare PTD data in μP system RAM**

   The communication between the HCD and the SAA1160A HC is in the form of Proprietary Transfer Descriptor (PTD) data. The PTD data provides USB traffic information about the commands, status and USB data packets.

   The physical storage media of PTD data for the HCD is the microprocessors system RAM. For the SAA1160As HC, the storage media is the internal FIFO buffer RAM.

   The HCD prepares PTD data in the microprocessors system RAM for transfer to the SAA1160As HC internal FIFO buffer RAM.

6. **Transfer PTD data into HCs FIFO buffer RAM**

   When PTD data is ready in the microprocessors system RAM, the HCD must transfer the PTD data from the microprocessors system RAM into the SAA1160As internal FIFO buffer RAM.

7. **HC interprets PTD data**

   The HC determines what USB transactions are required based on the PTD data that has been transferred into the internal FIFO buffer RAM.

8. **HC performs USB transactions via USB bus interface**

   The HC performs the USB transactions with the specified USB device endpoint through the USB bus interface.

9. **HC informs HCD of the USB traffic results**

   The USB transaction status and the feedback from the specified USB device endpoint will be put back into the SAA1160As HC internal FIFO buffer RAM in PTD data format. The HCD can read back the PTD data from the internal FIFO buffer RAM.

9.3 PTD data structure

The Proprietary Transfer Descriptor (PTD) data structure provides communication between the HCD and the SAA1160As USB HC. The PTD data contains information required by the USB traffic. PTD data consists of a PTD followed by its payload data, as shown in Figure 16.

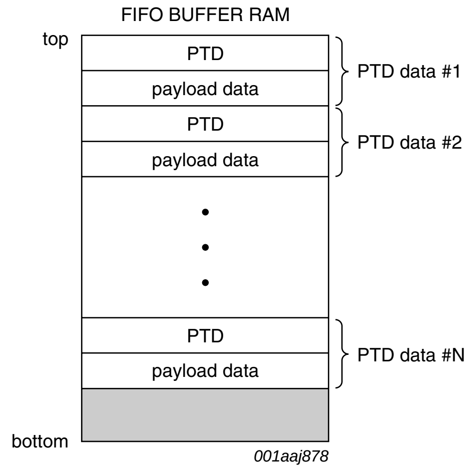


Fig 16. PTD data in FIFO buffer RAM

The PTD data structure is used by the HC to define a buffer of data that will be moved to or from an endpoint in the USB device. This data buffer is set up for the current frame (1 ms frame) by the HCD. The payload data for every transfer in the frame must have a PTD as the header to describe the characteristic of the transfer. The PTD data is double word (normally 4 bytes long) aligned.

9.3.1 PTD data header definition

The PTD forms the header of the PTD data. It tells the HC the transfer type, where the payload data should go, and the actual size of the payload data. A PTD is an 8-byte data structure that is very important for HCD programming.

Table 4. Proprietary Transfer Descriptor (PTD): bit allocation

Bit	7	6	5	4	3	2	1	0
Byte 0	ActualBytes[7:0]							
Byte 1	CompletionCode[3:0]				Active	Toggle	ActualBytes[9:8]	
Byte 2	MaxPacketSize[7:0]							
Byte 3	EndpointNumber[3:0]				Last	Speed	MaxPacketSize[9:8]	
Byte 4	TotalBytes[7:0]							
Byte 5	reserved[1]		B5_5	reserved[1]	DirectionPID[1:0]		TotalBytes[9:8]	
Byte 6	Format	FunctionAddress[6:0]						
Byte 7	reserved[1]							

[1] In write mode the default value is logic 0 and in read mode the default value is don’t care.

Table 5. Proprietary Transfer Descriptor (PTD): bit description

Symbol	Access	Description	
ActualBytes[9:0]	R/W	Contains the number of bytes that were transferred for this PTD.	
CompletionCode[3:0]	R/W	0000 NoError	General TD or isochronous data packet processing completed with no detected errors.
		0001 CRC	Last data packet from endpoint contained a CRC error.
		0010 BitStuffing	Last data packet from endpoint contained a bit stuffing violation.
		0011 DataToggleMismatch	Last packet from endpoint had data toggle PID that did not match the expected value.
		0100 Stall	TD was moved to the Done queue because the endpoint returned a STALL PID.
		0101 DeviceNotResponding	Device did not respond to token (IN) or did not provide a handshake (OUT).
		0110 PIDCheckFailure	Check bits on PID from endpoint failed on data PID (IN) or handshake (OUT).
		0111 UnexpectedPID	Received PID was not valid when encountered or PID value is not defined.
		1000 DataOverrun	The amount of data returned by the endpoint exceeded either the size of the maximum data packet allowed from the endpoint (found in the MaximumPacketSize field of endpoint descriptor) or the remaining buffer size.
		1001 DataUnderrun	The endpoint returned is less than MaximumPacketSize and that amount was not sufficient to fill the specified buffer.
		1010 reserved	-
		1011 reserved	-
		1100 BufferOverrun	During an IN, the HC received data from an endpoint faster than it could be written to system memory.
		1101 BufferUnderrun	During an OUT, the HC could not retrieve data from the system memory fast enough to keep up with the USB data rate.
Active	R/W	Set to logic 1 by firmware to enable the execution of transactions by the HC. When the transaction associated with this descriptor is completed, the HC sets this bit to logic 0, indicating that a transaction for this element will not be executed when it is next encountered in the schedule.	
Toggle	R/W	Used to generate or compare the data PID value (DATA0 or DATA1). It is updated after each successful transmission or reception of a data packet.	
MaxPacketSize[9:0]	R	The maximum number of bytes that can be sent to or received from the endpoint in a single data packet.	
EndpointNumber[3:0]	R	USB address of the endpoint within the function.	
Last	R	Last PTD of a list (ITL or ATL). Logic 1 indicates that the PTD is the last PTD.	
Speed	R	Speed of the endpoint: **0 —** full speed **1 —** low speed	
TotalBytes[9:0]	R	Specifies the total number of bytes to be transferred with this data structure. For Bulk and Control only, this can be greater than MaximumPacketSize.	

All information provided in this document is subject to legal disclaimers.

Table 5. Proprietary Transfer Descriptor (PTD): bit description *...continued*

Symbol	Access	Description	
DirectionPID[1:0]	R	00	SETUP
		01	OUT
		10	IN
		11	reserved
B5_5	R/W	Specifies number of PTD retries within 1 ms time frame. If B5_5 is set to logic 0, the HC will retry the token on the bus for the entire frame, even if the device not acknowledges at the first attempt. If B5_5 is set to logic 1, the PTD will be tried only once in a frame. For control or bulk PTDs, this bit may be set to logic 0, while it need to be set to logic 1 for interrupt or isochronous PTDs in order to utilize the USB bus properly.	
Format	R	The format of this data structure. If this is a Control, Bulk or Interrupt endpoint, then Format = 0. If this is an Isochronous endpoint, then Format = 1.	
FunctionAddress[6:0]	R	This is the USB address of the function containing the endpoint that this PTD refers to.	

9.4 HCs internal FIFO buffer RAM structure

9.4.1 Partitions

According to the *Universal Serial Bus Specification Rev. 2.0*, there are four types of USB data transfers: Control, Bulk, Interrupt and Isochronous.

The HCs internal FIFO buffer RAM has a physical size of 4 kB. This internal FIFO buffer RAM is used for transferring data between the microprocessor and USB peripheral devices. This on-chip buffer RAM can be partitioned into two areas: Acknowledged Transfer List (ATL) buffer and Isochronous Transfer List (ITL) buffer. The ITL buffer is a ping-pong structured FIFO buffer RAM that is used to keep the payload data and their PTD header for Isochronous transfers. The ATL buffer is a non ping-pong structured FIFO buffer RAM that is used for the other three types of transfers.

The ITL buffer can be further partitioned into ITL0 and ITL1 for the ping-pong structure. The ITL0 and ITL1 buffers always have the same size. The microprocessor can put ISO data into either the ITL0 buffer or the ITL1 buffer. When the microprocessor accesses an ITL buffer, the HC can take over the other ITL buffer at the same time. This architecture improves the ISO transfer performance.

The HCD can assign the logical size for the ATL buffer and ITL buffers at any time, but normally at initialization after power-on reset. This is done by setting the HcATLBufferLength register (2BH to read, ABH to write) and the HcITLBufferLength register (2AH to read, AAH to write), respectively. The total length (ATL buffer + ITL buffer) should not exceed the maximum RAM size of 4 kB. Figure 17 shows the partitions of the internal FIFO buffer RAM. When assigning buffer RAM sizes, follow this formula:

ATL buffer length + 2 × (ITL buffer size) ≤ 1000H (that is, 4 kB)

where: ITL buffer size = ITL0 buffer length = ITL1 buffer length

The following assignments are examples of legal uses of the internal FIFO buffer RAM:

- ATL buffer length = 800H, ITL buffer length = 400H.

  This is the maximum use of the internal FIFO buffer RAM.
- ATL buffer length = 400H, ITL buffer length = 200H.

  This is insufficient use of the internal FIFO buffer RAM.
- ATL buffer length = 1000H, ITL buffer length = 0H.

  This will use the internal FIFO buffer RAM for only ATL transfers.

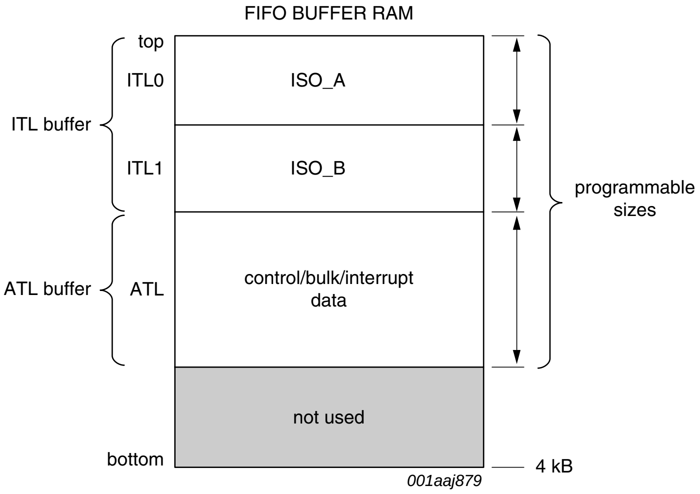


Fig 17. HC internal FIFO buffer RAM partitions

The actual requirement for the buffer RAM needs to reach not the maximum size. You can make your selection based on your application.

The following are some calculations of the ISO_A or ISO_B space for a frame of data:

- Maximum number of useful data sent during one USB frame is 1280 B (20 ISO packets of 64 B). The total RAM size needed is:

  20 × 8 + 1280 = 1440 B.
- Maximum number of packets for different endpoints sent during one USB frame is 150 (150 ISO packets of 1 B). The total RAM size needed is:

  150 × 8 + 150 × 1 = 1350 B.
- The Ping buffer RAM (ITL0) and the Pong buffer RAM (ITL1) have a maximum size of 2 kB each. All data needed for one frame can be stored in the Ping or the Pong buffer RAM.

When the embedded system wants to initiate a transfer to the USB bus, the data needed for one frame is transferred to the ATL buffer or the ITL buffer. The microprocessor detects the buffer status through interrupt routines. When the HcBufferStatus register (2CH to read only) indicates that the buffer is empty, then the microprocessor writes data into the buffer. When the HcBufferStatus register indicates that the buffer is full, the data is ready on the buffer, and the microprocessor needs to read data from the buffer.

For every 1 ms, there might be many events to generate interrupt requests to the microprocessor for data transfer or status retrieval. However, each of the interrupt types defined in this specification can be enabled or disabled by setting HcμPInterruptEnable register bits accordingly.

The data transfer can be done via the PIO mode or the DMA mode. The data transfer rate can go up to 15 MB/s. In the DMA operation, the single-cycle or multi-cycle burst modes are supported. Multi-cycle burst modes of 1, 4 or 8 cycles per burst are supported for the SAA1160A.

9.4.2 Data organization

PTD data is used for every data transfer between a microprocessor and the USB bus, and the PTD data resides in the buffer RAM. For an OUT or SETUP transfer, the payload data is placed just after the PTD, after which the next PTD is placed. For an IN transfer, RAM space is reserved for receiving a number of bytes that is equal to the total bytes of the transfer. After this, the next PTD and its payload data are placed (see Figure 18).

Remark: The PTD is defined for both the ATL and ITL type data transfer. For ITL, the PTD data is put into ITL buffer RAM, and the SAA1160A takes care of the ping-pong action for the ITL buffer RAM access.

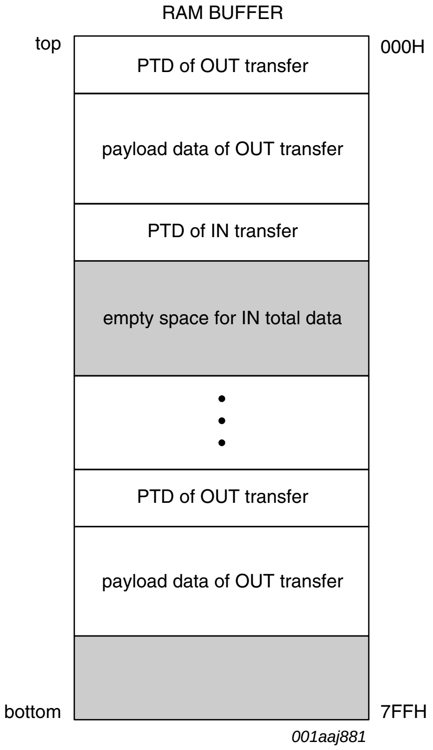


Fig 18. Buffer RAM data organization

The PTD data (PTD header and its payload data) is a structure of double word alignment. This means that the memory address is organized in blocks of 4 B. Therefore, the first byte of every PTD and the first byte of every payload data are located at an address that is a multiple of 4. Figure 19 illustrates an example in which the first payload data is 14 B long, meaning that the last byte of the payload data is at the location 15H. The next addresses (16H and 17H) are not multiples of 4. Therefore, the first byte of the next PTD will be located at the next multiple-of-four address (18H).

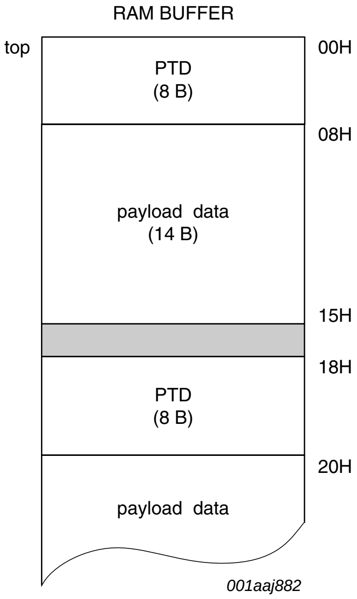


Fig 19. PTD data with double word alignment in buffer RAM

9.4.3 Operation and C program example

Figure 20 shows the block diagram for internal FIFO buffer RAM operations in the PIO mode. The SAA1160A provides one register as the access port for each buffer RAM. For the ITL buffer RAM, the access port is the ITLBufferPort register (40H to read, C0H to write). For the ATL buffer RAM, the access port is the ATLBufferPort register (41H to read, C1H to write). The buffer RAM is an array of bytes while the access port is a 16-bit register. Therefore, each read/write operation on the port accesses two consecutive memory locations, incrementing the pointer of the internal buffer RAM by two.

The lower byte of the access port register corresponds to the data byte at the even location of the buffer RAM, and the upper byte corresponds to the next data byte at the odd location of the buffer RAM. Regardless of the number of data bytes to be transferred, the command code must be issued merely once, and it will be followed by a number of accesses of the data port (see Section 8.4).

When the pointer of the buffer RAM reaches the value of the HcTransferCounter register, an internal EOT signal will be generated to set bit 2, AllEOTInterrupt, of the HcμPInterrupt register and update the HcBufferStatus register, to indicate that the whole data transfer has been completed.

For ITL buffer RAM, every Start Of Frame (SOF) signal (1 ms) will cause toggling between ITL0 and ITL1 but this depends on the buffer status. If both ITL0BufferFull and ITL1BufferFull of the HcBufferStatus register are already logic 1, meaning that both ITL0 and ITL1 buffer RAMs are full, the toggling will not happen. In this case, the microprocessor will always have access to ITL1.

When programming ATL transfers, a dummy PTD need to be appended at the end of all ATL PTDs and payload for each 1 ms. A dummy PTD is a PTD header with the active bit set to logic 0. Without dummy PTD the SAA1160A will not process the original PTDs and payload. Independent of the number of devices connected, only one dummy PTD is required for each 1 ms.

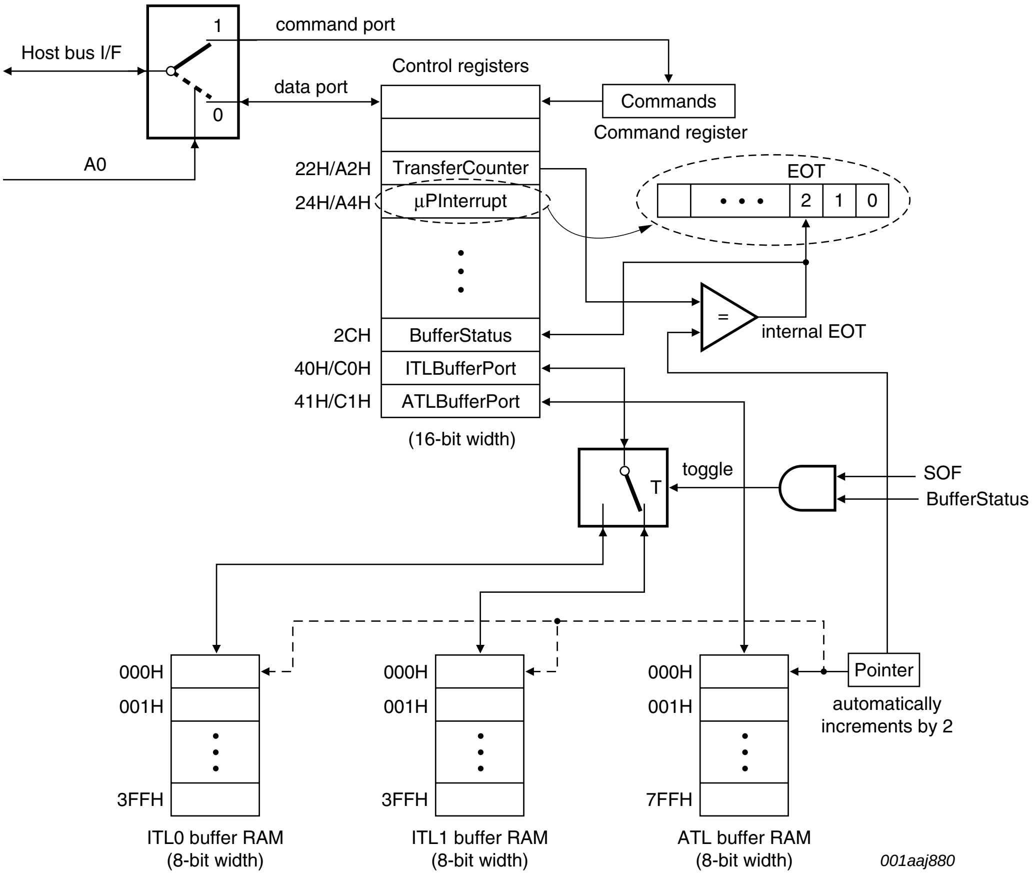


Fig 20. PIO access to internal FIFO buffer RAM

Following is an example of a C program that shows how to write data into the ATL buffer RAM. The total number of data bytes to be transferred is 88 (decimal) that will be set into the HcTransferCounter register as 58H. The data consists of five types of PTD data:

1. The first PTD header (IN) is 8 B, followed by 16 B of space reserved for its payload data.
2. The second PTD header (IN) is also 8 B, followed by 8 B of space reserved for its payload data.
3. The third PTD header (OUT) is 8 B, followed by 16 B of payload data with values beginning from 0H to FH incrementing by 1.
4. The fourth PTD header (OUT) is also 8 B, followed by 8 B of payload data with values beginning from 0H to EH incrementing by 2.
5. The fifth PTD header (OUT) is 8 B and serves as a dummy PTD.

In all PTDs, we have assigned device address as 5 and endpoint 1. ActualBytes is always zero (0). TotalBytes equals the number of payload data bytes transferred. However, note that for bulk and control transfers, TotalBytes can be greater than MaxPacketSize.

Table 6 shows the results after running this program.

However, if communication with a peripheral USB device is desired, the device should be connected to the downstream port and pass enumeration.

```
//The example program for writing ATL buffer RAM
#include <conio.h>
#include <stdio.h>
#include <dos.h>

//Define register commands
#define wHcTransferCounter 0x22
#define wHcuPInterrupt 0x24
#define wHcATLBufferLength 0x2b
#define wHcBufferStatus 0x2c

// Define I/O Port Address for HC
#define HcDataPort 0x290
#define HcCmdPort 0x292

//Declare external functions to be used
unsigned int HcRegRead(unsigned int wIndex);
void HcRegWrite(unsigned int wIndex,unsigned int wValue);

void main(void)
{
unsigned int i;
unsigned int wCount,wData;

// Prepare PTD data to be written into HC ATL buffer RAM:
unsigned int PTDData[0x2c]=
{

0x0800,0x1010,0x0810,0x0005, //PTD header for IN token #1

//Reserved space for payload data of IN token #1
0x0000,0x0000,0x0000,0x0000, 0x0000,0x0000,0x0000,0x0000,

0x0800,0x1008,0x0808,0x0005, //PTD header for IN token #2

//Reserved space for payload data of IN token #2
0x0000,0x0000,0x0000,0x0000,

0x0800,0x1010,0x0410,0x0005, //PTD header for OUT token #1

0x0100,0x0302,0x0504,0x0706, //Payload data for OUT token #1
0x0908,0x0b0a,0x0d0c,0x0f0e,
0x0800,0x1008,0x0408,0x0005, //PTD header for OUT token #2

0x0200,0x0604,0x0a08,0x0e0c, //Payload data for OUT token #2

0x0000,0x1800,0x0400,0x0005 //dummy PTD
};
HcRegWrite(wHcuPInterrupt,0x04); //Clear EOT interrupt bit
//HcRegWrite(wHcITLBufferLength,0x0);
HcRegWrite(wHcATLBufferLength,0x1000); //RAM full use for ATL
//Set the number of bytes to be transferred
HcRegWrite(wHcTransferCounter,0x58);

wCount = 0x2c; //Get word count
output(HcCmdPort,0x00c1); //Command for ATL buffer write

//write 88 (0x58) bytes of data into ATL buffer RAM
for (i=0;i<wCount;i++)
 {
 outport(HcDataPort,PTDData[i]);
 };

//Check EOT interrupt bit
 wData = HcRegRead(wHcuPInterrupt);
 printf("\n HC Interrupt Status = %xH.\n",wData);

//Check Buffer status register
 wData = HcRegRead(wHcBufferStatus);
 printf("\n HC Buffer Status = %xH.\n",wData);
 }

//
// Read HC 16-bit registers
//
unsigned int HcRegRead(unsigned int wIndex)
 {
 unsigned int wValue;

 outport(HcCmdPort,wIndex & 0x7f);
 wValue = inport(HcDataPort);

 return(wValue);

}
//
// Write HC 16-bit registers
//
void HcRegWrite(unsigned int wIndex,unsigned int wValue)
 {
 outport(HcCmdPort,wIndex | 0x80);
 outport(HcDataPort,wValue);
 }
```

Table 6. Run results of the C program example

Observed items	HC not initialized and not in USBOperational state	HC initialized and in USBOperational state	Comments
HcµPInterrupt register			
Bit 1 (ATLInt)	0	1	microprocessor must read ATL
Bit 2 (AllEOTInterrupt)	1	1	transfer completed
HcBufferStatus register			
Bit 2 (ATLBufferFull)	1	1	transfer completed
Bit 5 (ATLBufferDone)	0	1	PTD data processed by HC
USB traffic on USB bus	no	yes	OUT packets can be seen

9.5 HC operational model

Upon power up, the HCD sets up all operational registers (32-bit). The FSLargestDataPacket field (bits 30 to 16) of the HcFmInterval register (0DH to read, 8DH to write) and the HcLSThreshold register (11H to read, 91H to write) determine the end of the frame for full-speed and low-speed packets. By programming these fields, the effective USB bus usage can be changed. Furthermore, the size of the ITL buffers (HcITLBufferLength, 2AH to read, AAH to write) is programmed.

If a USB frame contains both isochronous and acknowledged packets, two interrupts will be generated per frame.

One interrupt is issued concurrently with the SOF. This interrupt (ITLInt is set in the HcµPInterrupt register) triggers reading and writing of the ITL buffer by the microprocessor, after which the interrupt is cleared by the microprocessor.

Next the programmable ATL Interrupt (bit ATLInt is set in the HcµPInterrupt register) is issued, which triggers reading and writing of the ATL buffer by the microprocessor, after which the interrupt is cleared by the microprocessor. If the microprocessor cannot handle the ISO interrupt before the next ISO interrupt, disrupted ISO traffic can result.

To be able to send more than one packet to the same Control or Bulk endpoint in the same frame, the Active bit and the TotalBytes field are introduced (see Table 5). Bit Active is cleared only if all data of the Proprietary Transfer Descriptor (PTD) have been transferred or if a transaction at that endpoint contained a fatal error. If all PTDs of the ATL are serviced once and the frame is not over yet, the HC starts looking for a PTD with bit Active still set. If such a PTD is found and there is still enough time in this frame, another transaction is started on the USB bus for this endpoint.

For ISO processing, the HCD also has to take care of the BufferStatus register (2CH, read only) for the ITL buffer RAM operations. After the HCD writes ISO data into ITL buffer RAM, the ITL0BufferFull or ITL1BufferFull bit (depending on whether it is ITL0 or ITL1) will be set to logic 1.

After the HC processes the ISO data in the ITL buffer RAM, the corresponding ITL0BufferDone or ITL1BufferDone bit will automatically be set to logic 1.

The HCD can clear the buffer status bits by a read of the ITL buffer RAM. This must be done within the 1 ms frame from which ITL0BufferDone or ITL1BufferDone was set. Failure to do so will cause the ISO processing to stop and a power-on reset or software reset will have to be applied to the HC, a USB reset to the USB bus must **not** be made.

For example, the HCD writes ISO_A data into the ITL0 buffer in the first frame. This will cause the HcBufferStatus register to show that the ITL0 buffer is full by setting bit ITL0BufferFull to logic 1. At this stage, the HCD cannot write ISO data into the ITL0 buffer RAM again.

In the second frame, the HC will process the ISO_A data in the ITL0 buffer. At the same time, the HCD can write ISO_B data into the ITL1 buffer. When the next SOF comes (the beginning of the third frame), both ITL1BufferFull and ITL0BufferDone are automatically set to logic 1.

In the third frame, the HCD has to read at least two bytes (one word) of the ITL0 buffer to clear **both** the ITL0BufferFull and ITL0BufferDone bits. If both are not cleared, when the next SOF comes (the beginning of the fourth frame) the ITL0BufferDone and ITL0BufferFull bits will be cleared automatically. This also applies to the ITL1 buffer because ITL0 and ITL1 are ping-pong structured buffers. To recover from this state, a power-on reset or software reset will have to be applied.

9.5.1 Time domain behavior

In example 1 (Figure 21), the CPU is fast enough to read back and download a scenario before the next interrupt. Note that on the ISO interrupt of frame N:

- The ISO packet for frame N + 1 will be written
- The AT packet for frame N + 1 will be written

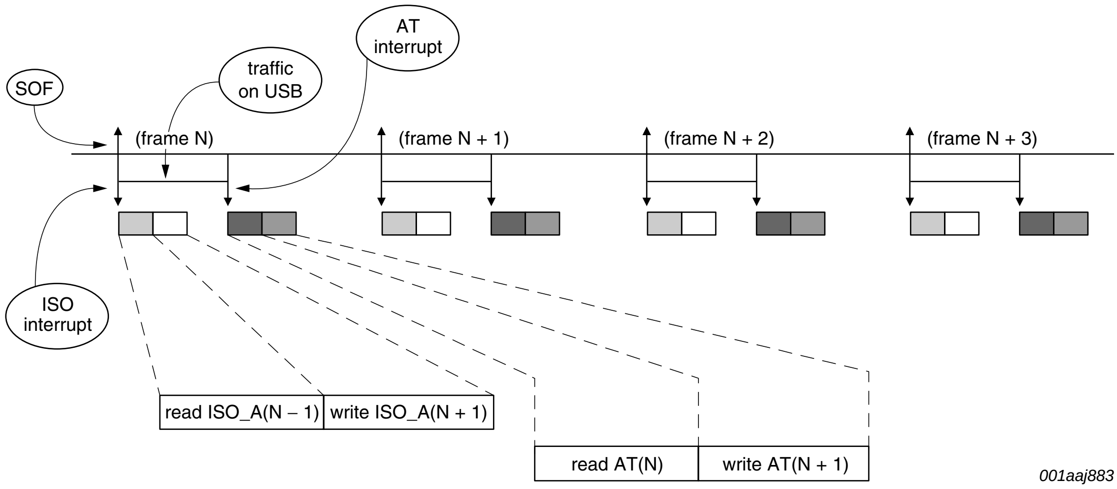


Fig 21. HC time domain behavior: example 1

In example 2 (Figure 22), the microprocessor is still busy transferring the AT data when the ISO interrupt of the next frame (N + 1) is raised. As a result, there will be no AT traffic in frame N + 1. The HC does not raise an AT interrupt in frame N + 1. The AT part is simply postponed until frame N + 2. On the AT N + 2 interrupt, the transfer mechanism is back to the normal operation. This simple mechanism ensures, among other things, that Control transfers are not dropped systematically from the USB in case of an overloaded microprocessor.

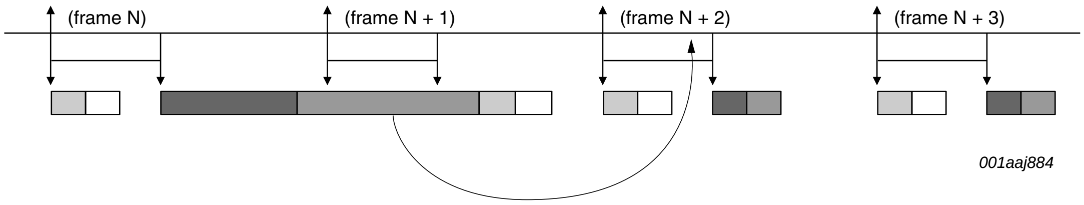


Fig 22. HC time domain behavior: example 2

In example 3 (Figure 23), the ISO part is still being written while the Start of Frame (SOF) of the next frame has occurred. This will result in undefined behavior for the ISO data on the USB bus in frame N + 1 (depending on whether the exact timing data is corrupted or not). The HC should not raise an AT interrupt in frame N + 1.

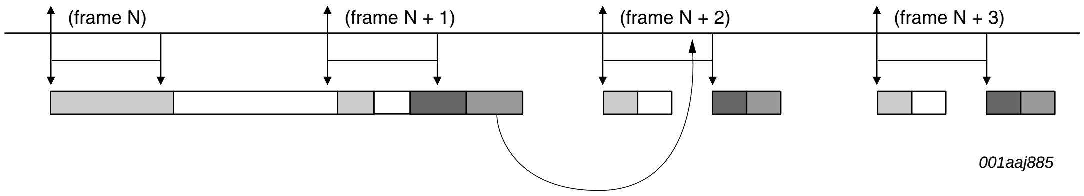


Fig 23. HC time domain behavior: example 3

9.5.2 Control transaction limitations

The different phases of a Control transfer (SETUP, Data and Status) should never be put in the same ATL.

9.6 Microprocessor loading

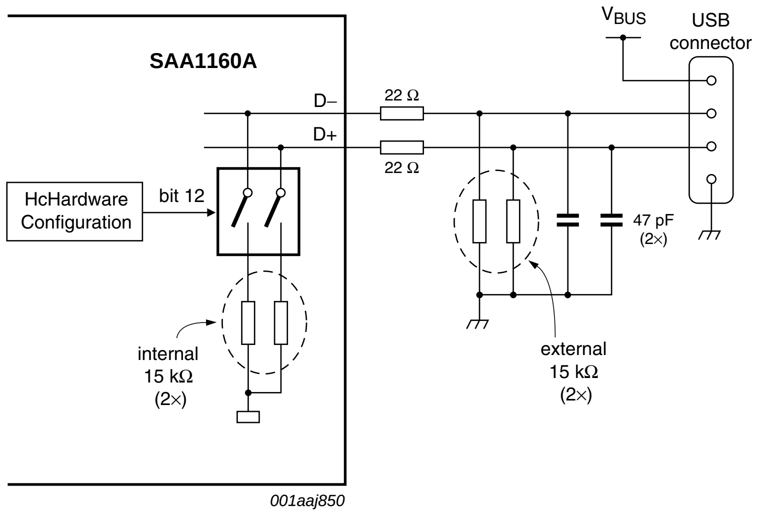
The maximum amount of data that can be transferred for an endpoint in one frame is 1023 B. The number of USB packets that are needed for this batch of data depends on the maximum packet size that is specified.

The HCD has to schedule the transactions in a frame. On the other hand, the microprocessor must have the ability to handle the interrupts coming from the HC every 1 ms. It must also be able to do the scheduling for the next frame, reading the frame information from and writing the next frame information to the buffer RAM in the time between the end of the current frame and the start of the next frame.

9.7 Internal pull-down resistors for downstream ports

There are four internal 15 kΩ pull-down resistors built in the SAA1160A for the two downstream ports: two resistors for each port. These resistors are software selectable by programming bit 12 (2_DownstreamPort15Kresistorsel) of the HcHardwareConfiguration register (20H to read, A0H to write). When bit 12 is logic 0, external 15 kΩ pull-down resistors are used. When bit 12 is logic 1, internal 15 kΩ pull-down resistors are used. See Figure 24.

This feature is a cost-saving option. However, the power-on reset default value of bit 12 is logic 0. If using the internal resistors, the HCD must check this bit status after every reset, because a reset action (hardware or software) will clear this bit.



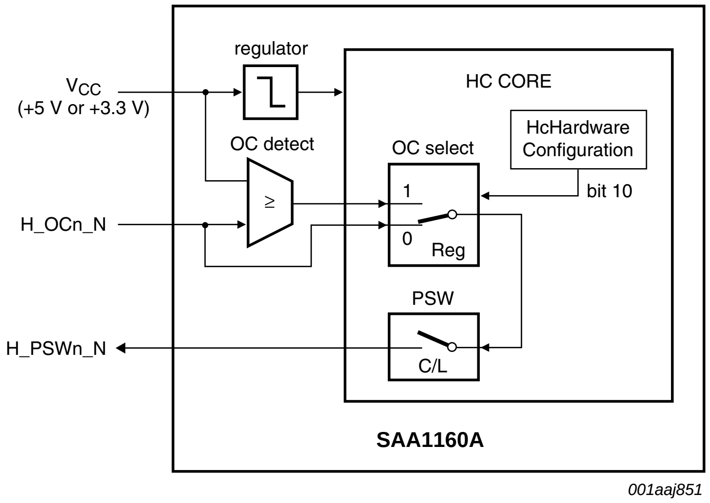
Using either internal or external 15 kΩ resistors.

Fig 24. Use of 15 kΩ pull-down resistors on downstream ports

9.8 Overcurrent detection and power switching control

A downstream port provides 5 V power supply to V_{BUS}. The SAA1160A has built-in hardware functions to monitor the downstream ports loading conditions and control their power switching. These hardware functions are implemented by the internal power switching control circuit and overcurrent detection circuit. H_PSW1_N and H_PSW2_N are power switching control output pins (active LOW, open-drain) for downstream ports 1 and 2, respectively. H_OC1_N and H_OC2_N are overcurrent detection input pins for downstream ports 1 and 2, respectively.

Figure 25 shows the SAA1160A downstream port power management scheme.



'n' represents the downstream port numbers (n = 1 or 2).

Fig 25. Downstream port power management scheme

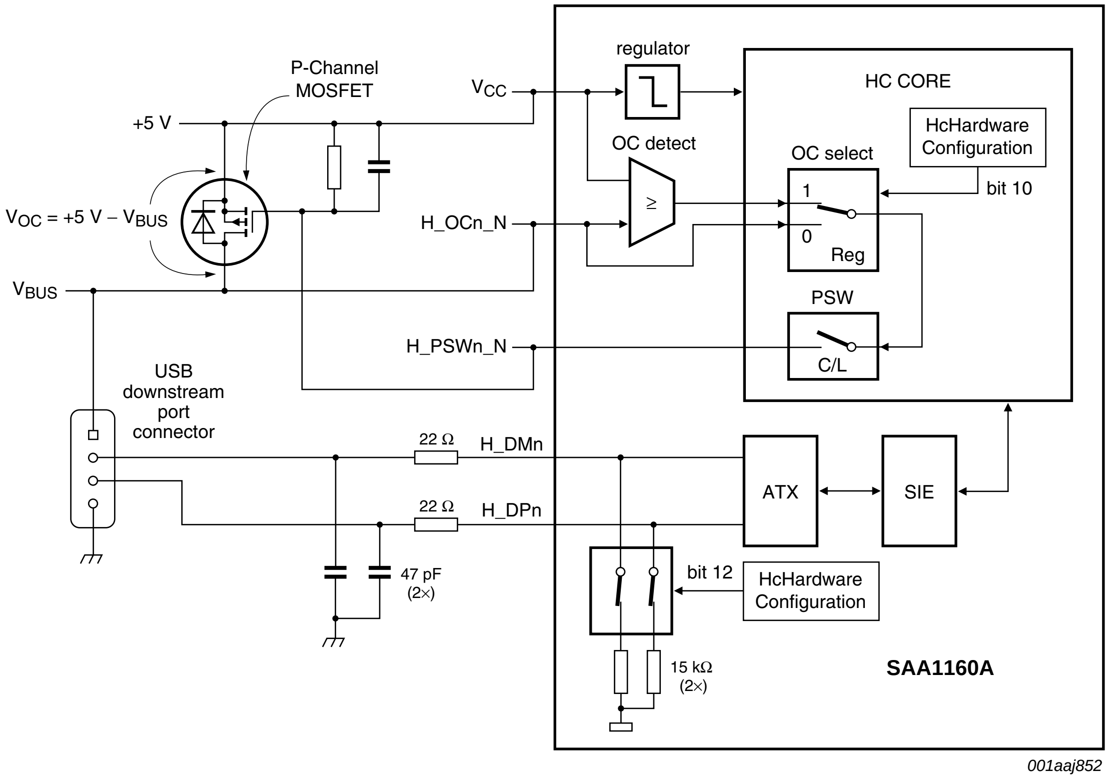
9.8.1 Using an internal OC detection circuit

The internal OC detection circuit can be used only when V_{CC} (pin 56) is connected to a 5 V power supply. The HCD must set AnalogOCEnable, bit 10 of the HcHardwareConfiguration register, to logic 1.

An application using the internal OC detection circuit and internal 15 kΩ pull-down resistors is shown in Figure 26. In this example, the HCD must set both AnalogOCEnable and DownstreamPort15Kresistorsel to logic 1. They are bit 10 and bit 12 of the HcHardwareConfiguration register, respectively.

When H_OCn_N detects an overcurrent status on a downstream port, H_PSWn_N will output HIGH, a logic 1 to turn off the 5 V power supply to the downstream port V_{BUS}. When there is no such detection, H_PSWn_N will output LOW, a logic 0 to turn on the 5 V power supply to the downstream port V_{BUS}.

In general applications, a P-channel MOSFET can be used as the power switch for V_{BUS}. Connect the 5 V power supply to the source of the P-channel MOSFET, V_{BUS} to the drain, and H_PSWn_N to the gate. Call the voltage drop across the drain and source, the overcurrent detection voltage (V_{OC}). For the internal overcurrent detection circuit, a voltage comparator has been designed-in, with a typical voltage threshold (ΔV_{trip}) of 225 mV. When V_{OC} exceeds V_{trip}, H_PSWn_N will output a HIGH level, logic 1 to turn off the P-channel MOSFET. The selection of a P-channel MOSFET with a different R_{DSon} will result in a different overcurrent threshold.



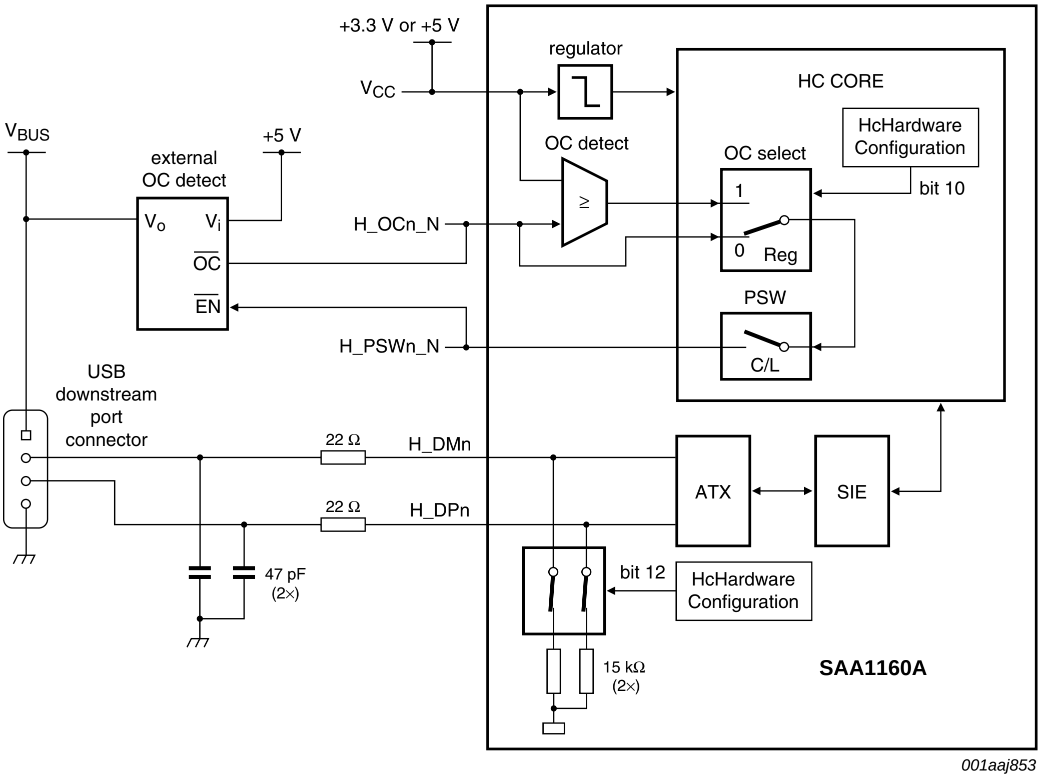
‘n’ represents the downstream port number (n = 1 or 2).

Fig 26. Using an internal OC detection circuit

9.8.2 Using an external OC detection circuit

When V_{CC} (pin 56) is connected to a 3.3 V instead of the 5 V power supply, the internal OC detection circuit cannot be used. An external OC detection circuit must be used instead. Regardless of the V_{CC} value, an external OC detection circuit can always be used. To use an external OC detection circuit, AnalogOCEnable, bit 10 of the HcHardwareConfiguration register, should be logic 0. By default after reset, this bit is already logic 0; therefore, the HCD does not need to clear this bit.

Figure 27 shows how to use an external OC detection circuit.



'n' represents the downstream port number (n = 1 or 2).

Fig 27. Using an external OC detection circuit

9.9 Suspend and wake-up

9.9.1 HC suspended state

The HC can be put into suspended state by setting the HcControl register (01H to read, 81H to write). See Figure 14 for the HCs flow of USB state changes.

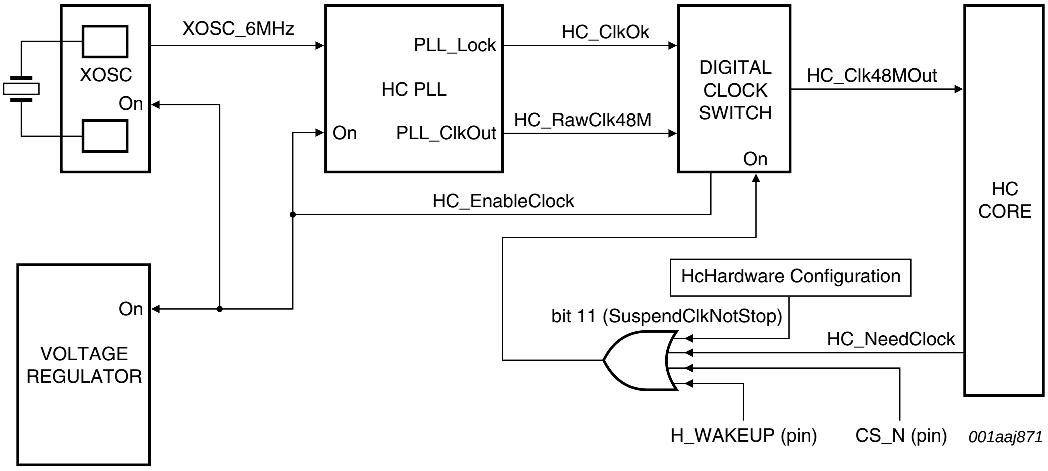


Fig 28. SAA1160A suspend and resume clock scheme

In the suspended state, the device will consume considerably less power by turning off the internal 48 MHz clock, PLL and crystal, and setting the internal regulator to Power-down mode. The SAA1160A suspend and resume clock scheme is shown in Figure 28.

Pin H_SUSPEND is the sensing output pin for the HCs suspended state. When the HC goes into the USBSuspend state, this pin will output a HIGH level (logic 1). This pin is cleared to LOW (logic 0) level only when the HC is put into a USBReset state or USBOperational state (refer to the HcControl register bits 7 to 6; 01H to read, 81H to write). Bit 11, SuspendClkNotStop, of the HcHardwareConfiguration register (20H to read, A0H to write), defines if the HC internal clock is stopped or kept running when the HC goes into the USBSuspend state. After the HC enters the USBSuspend state for 1.3 ms, the internal clock will be stopped if bit SuspendClkNotStop is logic 0.

The typical current consumption during the operational mode is approximately 50 mA. When the HC is put into suspend, however, the current consumption drops to approximately 22 mA. To achieve typical current consumption in the range of microampere, see Section 9.2.

9.9.2 HC wake-up from suspended state

There are three methods to wake up the HC from the USBSuspend state: hardware wake-up, software wake-up, and USB bus resume. They are described as follows.

9.9.2.1 Wake-up by pin H_WAKEUP

Pins H_SUSPEND and H_WAKEUP provide hardware wake-up, a way of remote wake-up control for the HC without the need to access the HC internal registers. H_WAKEUP is an external wake-up control input pin for the HC. After the HC goes into the USBSuspend state, it can be woken up by sending a HIGH-level pulse to pin H_WAKEUP. This will turn on the HCs internal clock, and set bit 6, ClkReady, of the HcμPInterrupt register (24H to read, A4H to write). Under the USBSuspend state, once pin H_WAKEUP goes HIGH, after 160 μs, the internal clock will be up. If pin H_WAKEUP continues to be HIGH, then the internal clock will be kept running, and the microprocessor can set the HC into USBOperational state during this time. If H_WAKEUP goes LOW for more than 1.14 ms, the internal clock stops and the HC goes back into the USBSuspend state.

9.9.2.2 Wake-up by pin CS_N (software wake-up)

During the USBSuspend state, an external microprocessor issues a chip select signal through pin CS_N to the SAA1160A. This method of access to the SAA1160A internal registers is a software wake-up.

9.9.2.3 Wake-up by USB devices

For the USB bus resume, a USB device attached to the root hub port issues a resume signal to the HC through the USB bus, switching the HC from the USBSuspend state to the USBResume state. This will also set bit ResumeDetected of the HcInterruptStatus register (03H to read, 83H to write).

No matter which method is used to wake up the HC from the USBSuspend state, the corresponding interrupt bits must be enabled before the HC goes into the USBSuspend state so that the microprocessor can receive the correct interrupt request to wake up the HC.

10. HC registers

The HC contains a set of on-chip control registers. These registers can be read or written by the Host Controller Driver (HCD). The Control and Status register sets, Frame Counter register sets, and Root Hub register sets are grouped under the category of HC Operational registers (32 bit). These operational registers are made compatible to OpenHCI (Host Controller Interface) operational registers. This allows the OpenHCI HCD to be easily ported to the SAA1160A.

Reserved bits may be defined in future releases of this specification. To ensure interoperability, the HCD must not assume that a reserved field contains logic 0. Furthermore, the HCD must always preserve the values of the reserved field. When a R/W register is modified, the HCD must first read the register, modify the bits desired, and then write the register with the reserved bits still containing the original value. Alternatively, the HCD can maintain an in-memory copy of previously written values that can be modified and then written to the HC register. When a ‘write to set’ or ‘clear the register’ is performed, bits written to reserved fields must be logic 0.

As shown in Table 7, the addresses (the commands for reading registers) of these 32-bit operational registers are similar to the offsets defined in the OHCI specification with the addresses being equal to offset divided by 4.

Table 7. HC registers summary

Address		Register	Width	Reference	Functionality
Read	Write				
00H	n.a.	HcRevision	32	Section 10.1.1 on page 39	HC control and status registers
01H	81H	HcControl	32	Section 10.1.2 on page 39	
02H	82H	HcCommandStatus	32	Section 10.1.3 on page 41	
03H	83H	HcInterruptStatus	32	Section 10.1.4 on page 42	
04H	84H	HcInterruptEnable	32	Section 10.1.5 on page 43	
05H	85H	HcInterruptDisable	32	Section 10.1.6 on page 44	
0DH	8DH	HcFmInterval	32	Section 10.2.1 on page 45	HC frame counter registers
0EH	n.a.	HcFmRemaining	32	Section 10.2.2 on page 46	
0FH	n.a.	HcFmNumber	32	Section 10.2.3 on page 47	
11H	91H	HcLSThreshold	32	Section 10.2.4 on page 48	
12H	92H	HcRhDescriptorA	32	Section 10.3.1 on page 49	HC Root Hub registers
13H	93H	HcRhDescriptorB	32	Section 10.3.2 on page 50	
14H	94H	HcRhStatus	32	Section 10.3.3 on page 52	
15H	95H	HcRhPortStatus[1]	32	Section 10.3.4 on page 53	
16H	96H	HcRhPortStatus[2]	32	Section 10.3.4 on page 53	
20H	A0H	HcHardwareConfiguration	16	Section 10.4.1 on page 56	HC DMA and interrupt control registers
21H	A1H	HcDMAConfiguration	16	Section 10.4.2 on page 58	
22H	A2H	HcTransferCounter	16	Section 10.4.3 on page 58	
24H	A4H	HcμPInterrupt	16	Section 10.4.4 on page 59	
25H	A5H	HcμPInterruptEnable	16	Section 10.4.5 on page 60	
27H	n.a.	HcChipID	16	Section 10.5.1 on page 61	HC miscellaneous registers
28H	A8H	HcScratch	16	Section 10.5.2 on page 62	
n.a.	A9H	HcSoftwareReset	16	Section 10.5.3 on page 62	
2AH	AAH	HcITLBufferLength	16	Section 10.6.1 on page 63	HC buffer RAM control registers
2BH	ABH	HcATLBufferLength	16	Section 10.6.2 on page 64	
2CH	n.a.	HcBufferStatus	16	Section 10.6.3 on page 64	
2DH	n.a.	HcReadBackITL0Length	16	Section 10.6.4 on page 65	
2EH	n.a.	HcReadBackITL1Length	16	Section 10.6.5 on page 65	
40H	C0H	HcITLBufferPort	16	Section 10.6.6 on page 66	
41H	C1H	HcATLBufferPort	16	Section 10.6.7 on page 67	

10.1 HC control and status registers

10.1.1 HcRevision register (R: 00H)

Code: 00H — read only

Table 8. HcRevision register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	REV[7:0]							
Reset	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 9. HcRevision register: bit description

Bit	Symbol	Description
31 to 8	-	reserved
7 to 0	REV[7:0]	**Revision:** This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. All HC implementations that are compliant with this specification will have a value of 10H.

10.1.2 HcControl register (R/W: 01H/81H)

The HcControl register defines the operating modes of the HC. RemoteWakeupEnable (RWE) is modified only by the HCD.

Code: 01H — read

Code: 81H — write

Table 10. HcControl register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved					RWE	RWC	reserved
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	HCFS[1:0]		reserved					
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11. HcControl register: bit description

Bit	Symbol	Description
31 to 11	-	reserved
10	RWE	**RemoteWakeupEnable:** This bit is used by the HCD to enable or disable the remote wake-up feature upon the detection of upstream resume signaling. When this bit is set and the ResumeDetected bit in HcInterruptStatus is set, a remote wake-up is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.
9	RWC	**RemoteWakeupConnected:** This bit indicates whether the HC supports remote wake-up signaling. If remote wake-up is supported and used by the system, it is the responsibility of system firmware to set this bit during POST. The HC clears the bit upon a hardware reset but does not alter it upon a software reset. Remote wake-up signaling of the host system is host-bus-specific and is not described in this specification.
8	-	reserved
7 to 6	HCFS[1:0]	**HostControllerFunctionalState** for USB: **00B —** USBReset **01B —** USBResume **10B —** USBOperational **11B —** USBSuspend A transition to USBOperational from another state causes Start Of Frame (SOF) generation to begin 1 ms later. The HCD may determine whether the HC has begun sending SOFs by reading the StartofFrame field of HcInterruptStatus. This field can be changed by the HC only when in the USBSuspend state. The HC can move from the USBSuspend state to the USBResume state after detecting the resume signaling from a downstream port. The HC enters USBReset after a software reset and a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signaling to downstream ports.
5 to 0	-	reserved

10.1.3 HcCommandStatus register (R/W: 02H/82H)

The HcCommandStatus register is used by the HC to receive commands issued by the HCD, and it also reflects the HCs current status. To the HCD, it appears to be a ‘write to set’ register. The HC must ensure that bits written as logic 1 become set in the register while bits written as logic 0 remain unchanged in the register. The HCD may issue multiple distinct commands to the HC without concern for corrupting previously issued commands. The HCD has normal read access to all bits.

The SchedulingOverrunCount field indicates the number of frames with which the HC has detected the scheduling overrun error. This occurs when the Periodic list does not complete before EOF. When a scheduling overrun error is detected, the HC increments the counter and sets the SchedulingOverrun field in the HcInterruptStatus register.

Code: 02H — read

Code: 82H — write

Table 12. HcCommandStatus register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved						SOC[1:0]	
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved							HCR
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13. HcCommandStatus register: bit description

Bit	Symbol	Description
31 to 18	-	reserved
17 to 16	SOC[1:0]	**SchedulingOverrunCount:** The field is incremented on each scheduling overrun error. It is initialized to 00B and wraps around at 11B. It will be incremented when a scheduling overrun is detected even if SchedulingOverrun in HcInterruptStatus has already been set. This is used by HCD to monitor any persistent scheduling problems.
15 to 1	-	reserved
0	HCR	**HostControllerReset:** This bit is set by the HCD to initiate a software reset of the HC. Regardless of the functional state of the HC, it moves to the USBSuspend state in which most of the operational registers are reset, except those stated otherwise, and no Host bus accesses are allowed. This bit is cleared by the HC upon the completion of the reset operation. The reset operation must be completed within 10 μs. This bit, when set, does not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports.

10.1.4 HcInterruptStatus register (R/W: 03H/83H)

This register provides the status of the events that cause hardware interrupts. When an event occurs, the HC sets the corresponding bit in this register. When a bit is set, a hardware interrupt is generated if the interrupt is enabled in the HcInterruptEnable register (see Section 10.1.5) and bit MasterInterruptEnable is set. The HCD can clear individual bits in this register by writing logic 1 to the bit positions to be cleared, but cannot set any of these bits. Conversely, the HC can set bits in this register, but cannot clear these bits.

Code: 03H — read

Code: 83H — write

Table 14. HcInteruptStatus register: bit allocation

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved	RHSC	FNO	UE	RD	SF	reserved	SO
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15. HcInterruptStatus register: bit description

Bit	**Symbol**	**Description**
31 to 7	-	reserved
6	RHSC	**RootHubStatusChange:** This bit is set when the content of HcRhStatus or the content of any of HcRhPortStatus[1:2] has changed.
5	FNO	**FrameNumberOverflow:** This bit is set when the MSB of HcFmNumber (bit 15) changes value.
4	UE	**UnrecoverableError:** This bit is set when the HC detects a system error not related to USB. The HC does not proceed with any processing nor signaling before the system error has been corrected. The HCD clears this bit after the HC has been reset. OHCI: Always set to logic 0.
3	RD	**ResumeDetected:** This bit is set when the HC detects that a device on the USB is asserting resume signaling from a state of no resume signaling. This bit is not set when HCD enters the USBResume state.
2	SF	**StartOfFrame:** At the start of each frame, this bit is set by the HC and an SOF generated.
1	-	reserved
0	SO	**SchedulingOverrun:** This bit is set when USB schedules for current frame overruns. A scheduling overrun will also cause the SchedulingOverrunCount of HcCommandStatus to be incremented.

10.1.5 HcInterruptEnable register (R/W: 04H/84H)

Each enable bit in the HcInterruptEnable register corresponds to an associated interrupt bit in the HcInterruptStatus register. The HcInterruptEnable register is used to control which events generate a hardware interrupt. A hardware interrupt is requested on the host bus when three conditions occur:

- A bit is set in the HcInterruptStatus register
- The corresponding bit in the HcInterruptEnable register is set
- Bit MasterInterruptEnable is set

Writing a logic 1 to a bit in this register sets the corresponding bit, whereas writing a logic 0 to a bit in this register leaves the corresponding bit unchanged. On a read, the current value of this register is returned.

Code: 04H — read

Code: 84H — write

Table 16. HcInterruptEnable register: bit allocation

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	MIE	reserved						
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved	RHSC	FNO	UE	RD	SF	reserved	SO
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17. HcInterruptEnable register: bit description

Bit	Symbol	Description
31	MIE	**MasterInterruptEnable** by the HCD: A logic 0 is ignored by the HC. A logic 1 enables interrupt generation by events specified in other bits of this register.
30 to 7	-	reserved
6	RHSC	**0 —** ignore **1 —** enable interrupt generation due to Root Hub Status Change
5	FNO	**0 —** ignore **1 —** enable interrupt generation due to frame Number Overflow
4	UE	**0 —** ignore **1 —** enable interrupt generation due to Unrecoverable Error

All information provided in this document is subject to legal disclaimers.

Table 17. HcInterruptEnable register: bit description *…continued*

Bit	Symbol	Description
3	RD	**0 —** ignore **1 —** enable interrupt generation due to Resume Detect
2	SF	**0 —** ignore **1 —** enable interrupt generation due to Start of Frame
1	-	reserved
0	SO	**0 —** ignore **1 —** enable interrupt generation due to Scheduling Overrun

10.1.6 HcInterruptDisable register (R/W: 05H/85H)

Each disable bit in the HcInterruptDisable register corresponds to an associated interrupt bit in the HcInterruptStatus register. The HcInterruptDisable register is coupled with the HcInterruptEnable register. Thus, writing a logic 1 to a bit in this register clears the corresponding bit in the HcInterruptEnable register, whereas writing a logic 0 to a bit in this register leaves the corresponding bit in the HcInterruptEnable register unchanged. On a read, the current value of the HcInterruptEnable register is returned.

Code: 05H — read

Code: 85H — write

Table 18. HcInterruptDisable register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	MIE	reserved						
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved	RHSC	FNO	UE	RD	SF	reserved	SO
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19. HcInterruptDisable register: bit description

Bit	Symbol	Description
31	MIE	A logic 0 is ignored by the HC. A logic 1 disables interrupt generation due to events specified in other bits of this register. This field is set after a hardware or software reset.
30 to 7	-	reserved
6	RHSC	**0 —** ignore **1 —** disable interrupt generation due to Root Hub Status Change

Table 19. HcInterruptDisable register: bit description *…continued*

Bit	Symbol	Description
5	FNO	**0 —** ignore **1 —** disable interrupt generation due to Frame Number Overflow
4	UE	**0 —** ignore **1 —** disable interrupt generation due to Unrecoverable Error
3	RD	**0 —** ignore **1 —** disable interrupt generation due to Resume Detect
2	SF	**0 —** ignore **1 —** disable interrupt generation due to Start of Frame
1	-	reserved
0	SO	**0 —** ignore **1 —** disable interrupt generation due to Scheduling Overrun

10.2 HC frame counter registers

10.2.1 HcFmInterval register (R/W: 0DH/8DH)

The HcFmInterval register contains a 14-bit value which indicates the bit time interval in a frame (that is, between two consecutive SOFs), and a 15-bit value indicating the full-speed maximum packet size that the HC may transmit or receive without causing a scheduling overrun. The HCD may carry out minor adjustments on the FrameInterval by writing a new value at each SOF. This allows the HC to synchronize with an external clock resource and to adjust any unknown clock offset.

Code: 0DH — read

Code: 8DH — write

Table 20. HcFmInterval register: bit allocation

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	FIT	FSMPS[14:8]						
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	FSMPS[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved		FI[13:8]					
Reset	0	0	1	0	1	1	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	FI[7:0]							
Reset	1	1	0	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21. HcFmInterval register: bit description

Bit	Symbol	Description
31	FIT	**FrameIntervalToggle:** The HCD toggles this bit whenever it loads a new value to FrameInterval.
30 to 16	FSMPS [14:0]	**FSLargestDataPacket:** Specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing a scheduling overrun. The field value is calculated by the HCD.
15 to 14	-	reserved
13 to 0	FI[13:0]	**FrameInterval:** Specifies the interval between two consecutive SOFs in bit times. The default value is 11999. The HCD must save the current value of this field before resetting the HC. Setting the HostControllerReset field of the HcCommandStatus register will cause the HC to reset this field to its default value. HCD may choose to restore the saved value upon completing the reset sequence.

10.2.2 HcFmRemaining register (R: 0EH)

The HcFmRemaining register is a 14-bit down counter showing the bit time remaining in the current frame.

Code: 0EH — read

Table 22. HcFmRemaining register: bit allocation

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	FRT	reserved						
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved		FR[13:8]					
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	FR[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 23. HcFmRemaining register: bit description

Bit	Symbol	Description
31	FRT	**FrameRemainingToggle:** This bit is loaded from the FrameIntervalToggle field of the HcFmInterval register whenever FrameRemaining reaches 0. This bit is used by the HCD for synchronization between FrameInterval and FrameRemaining.
30 to 14	-	reserved
13 to 0	FR[13:0]	**FrameRemaining:** This counter is decremented at each bit time. When it reaches zero, it is reset by loading the FrameInterval value specified in the HcFmInterval register at the next bit time boundary. When entering the USBOperational state, the HC reloads it with the content of the FrameInterval part of the HcFmInterval register and uses the updated value from the next SOF.

10.2.3 HcFmNumber register (R: 0FH)

The HcFmNumber register is a 16-bit counter. It provides a timing reference for events happening in the HC and the HCD. The HCD may use the 16-bit value specified in this register and generate a 32-bit frame number without requiring frequent access to the register.

Code: 0FH — read

Table 24. HcFmNumber register: bit allocation

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	FN[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	FN[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 25. HcFmNumber register: bit description

Bit	Symbol	Description
31 to 16	-	reserved
15 to 0	FN[15:0]	**FrameNumber:** This is incremented when HcFmRemaining is reloaded. It rolls over to 0000H after FFFFH. When the USBOperational state is entered, this will be incremented automatically. The HC will set bit StartofFrame in the HcInterruptStatus register.

10.2.4 HcLSThreshold register (R/W: 11H/91H)

The HcLSThreshold register contains an 11-bit value used by the HC to determine whether to commit to the transfer of a maximum of 8-byte LS packet before EOF. Neither the HC nor the HCD is allowed to change this value.

Code: 11H — read

Code: 91H — write

Table 26. HcLSThreshold register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved					LST[10:8]		
Reset	0	0	0	0	0	1	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	LST[7:0]							
Reset	0	0	1	0	1	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27. HcLSThreshold register: bit description

Bit	Symbol	Description
31 to 11	-	reserved
10 to 0	LST[10:0]	**LSThreshold:** Contains a value that is compared to the FrameRemaining field before a low-speed transaction is initiated. The transaction is started only if FrameRemaining ≥ this field. The value is calculated by the HCD, which considers transmission and set-up overhead.

10.3 HC Root Hub registers

All registers included in this partition are dedicated to the USB Root Hub, which is an integral part of the HC although it is functionally a separate entity. The Host Controller Driver (HCD) emulates USBD accesses to the Root Hub via a register interface. The HCD maintains many USB-defined hub features that are not required to be supported in hardware. For example, the Hubs Device, Configuration, Interface, and Endpoint Descriptors, as well as some static fields of the Class Descriptor, are maintained only in the HCD. The HCD also maintains and decodes the Root Hubs device address as well as other minor operations more suited for software than hardware.

The Root Hub registers were developed to match the bit organization and operation of typical hubs found in the system.

Four 32-bit registers have been defined:

- HcRhDescriptorA
- HcRhDescriptorB
- HcRhStatus
- HcRhPortStatus[1:NDP]

Each register is read and written as a double word. These registers are only written during initialization to correspond with the system implementation. The HcRhDescriptorA and HcRhDescriptorB registers are writable regardless of the HCs USB states. HcRhStatus and HcRhPortStatus are writable during the USBOperational state only.

10.3.1 HcRhDescriptorA register (R/W: 12H/92H)

The HcRhDescriptorA register is the first register of two describing the characteristics of the Root Hub. The descriptor length (11), descriptor type and hub controller current (0) fields of the hub Class Descriptor are emulated by the HCD. All other fields are located in registers HcRhDescriptorA and HcRhDescriptorB.

Code: 12H — read

Code: 92H — write

Table 28. HcRhDescriptorA register: bit description

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	POTPGT[7:0]							
Reset	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved			NOCP	OCPM	DT	NPS	PSM
Reset	0	0	0	0	1	0	0	1
Access	R	R	R	R/W	R/W	R	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved						NDP[1:0][1]	
Reset	0	0	0	0	0	0	1	0
Access	R	R	R	R	R	R	R	R

[1] NDP[1:0] = 10b, only if pin NDP_SEL is left floating or forced HIGH. If pin NDP_SEL is forced LOW, NDP[1:0] = 01b.

Table 29. HcRhDescriptorA register: bit description

Bit	Symbol	Description
31 to 24	POTPGT [7:0]	**PowerOnToPowerGoodTime:** This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. The unit of time is 2 ms. The duration is calculated as POTPGT × 2 ms.
23 to 13	-	reserved
12	NOCP	**NoOverCurrentProtection:** This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the OverCurrentProtectionMode field specifies global or per-port reporting. **0 —** overcurrent status is reported collectively for all downstream ports **1 —** no overcurrent reporting supported
11	OCPM	**OverCurrentProtectionMode:** This bit describes how the overcurrent status for the Root Hub ports is reported. At reset, this field reflects the same mode as PowerSwitchingMode. This field is valid only if the NoOverCurrentProtection field is cleared. **0 —** overcurrent status is reported collectively for all downstream ports **1 —** overcurrent status is reported on a per-port basis. On power-up, clear this bit and then set it to logic 1.
10	DT	**DeviceType:** This bit specifies that the Root Hub is not a compound device—it is not permitted. This field should always read/write 0.
9	NPS	**NoPowerSwitching:** This bit is used to specify whether power switching is supported or ports are always powered. When this bit is cleared, bit PowerSwitchingMode specifies global or per-port switching. **0 —** ports are power switched **1 —** ports are always powered on when the HC is powered on
8	PSM	**PowerSwitchingMode:** This bit is used to specify how the power switching of the Root Hub ports is controlled. This field is valid only if the NoPowerSwitching field is cleared. **0 —** all ports are powered at the same time **1 —** each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If bit PortPowerControlMask is set, the port responds to only port power commands (Set/ClearPortPower). If the port mask is cleared, then the port is controlled only by the global power switch (Set/ClearGlobalPower).
7 to 2	-	reserved
1 to 0	NDP[1:0]	**NumberDownstreamPorts:** These bits specify the number of downstream ports supported by the Root Hub. The maximum number of ports supported by the SAA1160A is 2.

10.3.2 HcRhDescriptorB register (R/W: 13H/93H)

The HcRhDescriptorB register is the second register of two describing the characteristics of the Root Hub. These fields are written during initialization to correspond with the system implementation.

Code: 13H — read

Code: 93H — write

Table 30. HcRhDescriptorB register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Access	R	R	R	R	R	R	R	R
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved					PPCM[2:0]		
Reset	N/A	N/A	N/A	N/A	N/A	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved							
Reset	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved					DR[2:0]		
Reset	N/A	N/A	N/A	N/A	N/A	0	0	0
Access	R	R	R	R	R	R/W	R/W	R/W

Table 31. HcRhDescriptorB register: bit description

Bit	Symbol	Description
31 to 19	-	reserved
18 to 16	PPCM[2:0]	**PortPowerControlMask:** Each bit indicates whether a port is affected by a global power control command when PowerSwitchingMode is set. When set, the ports power state is only affected by per-port power control (Set/ClearPortPower). When cleared, the port is controlled by the global power switch (Set/ClearGlobalPower). If the device is configured to global switching mode (PowerSwitchingMode = 0), this field is not valid. **Bit 0 —** reserved **Bit 1 —** Ganged-power mask on Port #1 **Bit 2 —** Ganged-power mask on Port #2
15 to 3	-	reserved
2 to 0	DR[2:0]	**DeviceRemovable:** Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable. **Bit 0 —** reserved **Bit 1 —** Device attached to Port #1 **Bit 2 —** Device attached to Port #2

All information provided in this document is subject to legal disclaimers.

10.3.3 HcRhStatus register (R/W: 14H/94H)

The HcRhStatus register is divided into two parts. The lower word of a double word represents the Hub Status field and the upper word represents the Hub Status Change field. Reserved bits should always be written as logic 0.

Code: 14H — read

Code: 94H — write

Table 32. HcRhStatus register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	CRWE	reserved						
Reset	0	0	0	0	0	0	0	0
Access	W	R	R	R	R	R	R	R
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved						OCIC	LPSC
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	DRWE	reserved						
Reset	0	0	0	0	0	0	0	0
Access	R/W	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved						OCI	LPS
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R/W

Table 33. HcRhStatus register: bit description

Bit	Symbol	Description
31	CRWE	On write—**ClearRemoteWakeupEnable:** Writing a logic 1 clears DeviceRemoveWakeupEnable. Writing a logic 0 has no effect.
30 to 18	-	reserved
17	OCIC	**OverCurrentIndicatorChange:** This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a logic 1. Writing a logic 0 has no effect.
16	LPSC	On read—**LocalPowerStatusChange:** The Root Hub does not support the local power status feature. Therefore, this bit is always read as logic 0. On write—**SetGlobalPower:** In global power mode (PowerSwitchingMode = 0), this bit is written to logic 1 to turn on power to all ports (clear PortPowerStatus). In per-port power mode, it sets PortPowerStatus only on ports whose bit PortPowerControlMask is not set. Writing a logic 0 has no effect.
15	DRWE	On read—**DeviceRemoteWakeupEnable:** This bit enables the bit ConnectStatusChange as a resume event, causing a state transition USBSuspend to USBResume and setting the ResumeDetected interrupt. **0 —** ConnectStatusChange is not a remote wake-up event **1 —** ConnectStatusChange is a remote wake-up event On write—**SetRemoteWakeupEnable:** Writing a logic 1 sets DeviceRemoveWakeupEnable. Writing a logic 0 has no effect.

Table 33. HcRhStatus register: bit description *…continued*

Bit	Symbol	Description
14 to 2	-	reserved
1	OCI	**OverCurrentIndicator:** This bit reports overcurrent conditions when global reporting is implemented. When set, an overcurrent condition exists. When clear, all power operations are normal. If per-port overcurrent protection is implemented this bit is always logic 0.
0	LPS	On read—**LocalPowerStatus:** The Root Hub does not support the local power status feature. Therefore, this bit is always read as logic 0. On write—**ClearGlobalPower:** In global power mode (PowerSwitchingMode = 0), this bit is written to logic 1 to turn off power to all ports (clear PortPowerStatus). In per-port power mode, it clears PortPowerStatus only on ports whose bit PortPowerControlMask is not set. Writing a logic 0 has no effect.

10.3.4 HcRhPortStatus[1:2] (R/W [1]: 15H/95H, [2]: 16H/96H)

The HcRhPortStatus[1:2] register is used to control and report port events on a per-port basis. NumberDownstreamPorts represents the number of HcRhPortStatus registers that are implemented in hardware. The lower word is used to reflect the port status, whereas the upper word reflects the status change bits. Some status bits are implemented with special write behavior. If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port status change must be postponed until the transaction completes. Reserved bits should always be written logic 0.

Code: [1] = 15H, [2] = 16H — read

Code: [1] = 95H, [2] = 96H — write

Table 34. HcRhPortStatus[1:2] register: bit allocation

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved			PRSC	OCIC	PSSC	PESC	CSC
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved						LSDA	PPS
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved			PRS	POCI	PSS	PES	CCS
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35. HcRshPortStatus[1:2] register: bit description

Bit	Symbol	Description
31 to 21	-	reserved
20	PRSC	**PortResetStatusChange:** This bit is set at the end of the 10 ms port reset signal. The HCD writes a logic 1 to clear this bit. Writing a logic 0 has no effect. **0 —** port reset is not complete **1 —** port reset is complete
19	OCIC	**PortOverCurrentIndicatorChange:** This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the PortOverCurrentIndicator bit. The HCD writes a logic 1 to clear this bit. Writing a logic 0 has no effect. **0 —** no change in PortOverCurrentIndicator **1 —** PortOverCurrentIndicator has changed
18	PSSC	**PortSuspendStatusChange:** This bit is set when the full resume sequence has been completed. This sequence includes the 20 s resume pulse, LS EOP, and 3 ms resynchronization delay. The HCD writes a logic 1 to clear this bit. Writing a logic 0 has no effect. This bit is also cleared when ResetStatusChange is set. **0 —** resume is not completed **1 —** resume is completed
17	PESC	**PortEnableStatusChange:** This bit is set when hardware events cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a logic 1 to clear this bit. Writing a logic 0 has no effect. **0 —** no change in PortEnableStatus **1 —** change in PortEnableStatus
16	CSC	**ConnectStatusChange:** This bit is set whenever a connect or disconnect event occurs. The HCD writes a logic 1 to clear this bit. Writing a logic 0 has no effect. If CurrentConnectStatus is cleared when a SetPortReset, SetPortEnable, or SetPortSuspend write occurs, this bit is set to force the driver to reevaluate the connection status since these writes should not occur if the port is disconnected. **0 —** no change in CurrentConnectStatus **1 —** change in CurrentConnectStatus **Remark:** If bit DeviceRemovable[NDP] is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.
15 to 10	-	reserved
9	LSDA	On read—**LowSpeedDeviceAttached:** This bit indicates the speed of the device connected to this port. When set, a low-speed device is connected to this port. When clear, a full-speed device is connected to this port. This field is valid only when the CurrentConnectStatus is set. **0 —** full-speed device attached **1 —** low-speed device attached On write—**ClearPortPower:** The HCD clears bit PortPowerStatus by writing a logic 1 to this bit. Writing a logic 0 has no effect.

Table 35. HcRshPortStatus[1:2] register: bit description *…continued*

Bit	Symbol	Description
8	PPS	On read—**PortPowerStatus:** This bit reflects the port power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. The HCD sets this bit by writing SetPortPower or SetGlobalPower. The HCD clears this bit by writing ClearPortPower or ClearGlobalPower. Which power control switches are enabled is determined by PowerSwitchingMode. In the global switching mode (PowerSwitchingMode = 0), only Set/ClearGlobalPower controls this bit. In per-port power switching (PowerSwitchingMode = 1), if bit PortPowerControlMask[NDP] for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only Set/ClearGlobalPower commands are enabled. When port power is disabled, CurrentConnectStatus, PortEnableStatus, PortSuspendStatus, and PortResetStatus should be reset. **0 —** port power is off **1 —** port power is on On write—**SetPortPower:** The HCD writes a logic 1 to set bit PortPowerStatus. Writing a logic 0 has no effect. **Remark:** This bit always reads logic 1 if power switching is not supported.
7 to 5	-	reserved
4	PRS	On read—**PortResetStatus:** When this bit is set by a write to SetPortReset, port reset signaling is asserted. When reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared. **0 —** port reset signal is not active **1 —** port reset signal is active On write—**SetPortReset:** The HCD sets the port reset signaling by writing a logic 1 to this bit. Writing a logic 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortResetStatus but instead sets ConnectStatusChange. This informs the driver that it attempted to reset a disconnected port.
3	POCI	On read—**PortOverCurrentIndicator:** This bit is valid only when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to logic 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal. **0 —** no overcurrent condition **1 —** overcurrent condition detected On write—**ClearSuspendStatus:** The HCD writes a logic 1 to initiate a resume. Writing a logic 0 has no effect. A resume is initiated only if PortSuspendStatus is set.

Table 35. HcRshPortStatus[1:2] register: bit description *…continued*

Bit	Symbol	Description
2	PSS	On read—**PortSuspendStatus:** This bit indicates whether the port is suspended or in the resume sequence. It is set by a SetSuspendState write and cleared when PortSuspendStatusChange is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared when PortResetStatusChange is set at the end of the port reset or when the HC is placed in the USBResume state. If an upstream resume is in progress, it should propagate to the HC. **0 —** port is not suspended **1 —** port is suspended On write—**SetPortSuspend:** The HCD sets bit PortSuspendStatus by writing a logic 1 to this bit. Writing a logic 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port.
1	PES	On read—**PortEnableStatus:** This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PortEnabledStatusChange to be set. The HCD sets this bit by writing SetPortEnable and clears it by writing ClearPortEnable. This bit cannot be set when CurrentConnectStatus is cleared. This bit is also set at the completion of a port reset when ResetStatusChange is set or port is suspended when SuspendStatusChange is set. **0 —** port is disabled **1 —** port is enabled On write—**SetPortEnable:** The HCD sets PortEnableStatus by writing a logic 1. Writing a logic 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected port.
0	CCS	On read—**CurrentConnectStatus:** This bit reflects the current state of the downstream port. **0 —** no device connected **1 —** device connected On write—**ClearPortEnable:** The HCD writes a logic 1 to this bit to clear bit PortEnableStatus. Writing a logic 0 has no effect. CurrentConnectStatus is not affected by any write. **Remark:** This bit always reads logic 1 when the attached device is nonremovable (DeviceRemoveable[NDP]).

10.4 HC DMA and interrupt control registers

10.4.1 HcHardwareConfiguration register (R/W: 20H/A0H)

Code: 20H — read

Code: A0H — write

Table 36. HcHardwareConfiguration register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved			2_Downstream Port15K resistorsel	Suspend ClkNotStop	AnalogOC Enable	reserved	DACKMode
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	EOTInput Polarity	DACKInput Polarity	DREQ Output Polarity	DataBusWidth[1:0]		Interrupt Output Polarity	Interrupt PinTrigger	InterruptPin Enable
Reset	0	0	1	0	1	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37. HcHardwareConfiguration register: bit description

Bit	Symbol	Description
15 to 13	-	reserved
12	2_DownstreamPort15K resistorsel	**0 —** use external 15 kΩ resistors for downstream ports **1 —** use built-in resistors for downstream ports
11	SuspendClkNotStop	**0 —** clock can be stopped **1 —** clock cannot be stopped
10	AnalogOCEnable	**0 —** use external OC detection; digital input **1 —** use on-chip OC detection; analog input
9	-	reserved
8	DACKMode	**0 —** normal operation; pin DACK_N is used with read and write signals **1 —** reserved
7	EOTInputPolarity	**0 —** active LOW **1 —** active HIGH
6	DACKInputPolarity	**0 —** active LOW **1 —** reserved
5	DREQOutputPolarity	**0 —** active LOW **1 —** active HIGH
4 to 3	DataBusWidth[1:0]	These bits are fixed at logic 0 and logic 1 for the SAA1160A. **01 —** 16 bits **Others —** reserved
2	InterruptOutputPolarity	**0 —** active LOW **1 —** active HIGH
1	InterruptPinTrigger	**0 —** interrupt is level-triggered **1 —** interrupt is edge-triggered
0	InterruptPinEnable	This bit is used as pin INTs master interrupt enable and should be used together with register HcµPInterruptEnable to enable pin INT. **0 —** pin INT is disabled **1 —** pin INT is enabled

10.4.2 HcDMAConfiguration register (R/W: 21H/A1H)

Code: 21H — read

Code: A1H — write

Table 38. HcDMAConfiguration register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved	BurstLen[1:0]		DMA Enable	reserved	DMA Counter Select	ITL_ATL_ DataSelect	DMARead WriteSelect
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39. HcDMAConfiguration register: bit description

Bit	Symbol	Description
15 to 7	-	reserved
6 to 5	BurstLen[1:0]	**00 —** single-cycle burst DMA **01 —** 4-cycle burst DMA **10 —** 8-cycle burst DMA **11 —** reserved
4	DMAEnable	**0 —** DMA is terminated **1 —** DMA is enabled This bit will be reset to logic 0 when DMA transfer is completed.
3	-	reserved
2	DMACounterSelect	**0 —** DMA counter not used. External EOT must be used **1 —** enables the DMA counter for DMA transfer. HcTransferCounter register must be filled with non-zero values for DREQ to be raised after bit DMA Enable is set.
1	ITL_ATL_DataSelect	**0 —** ITL buffer RAM selected for ITL data **1 —** ATL buffer RAM selected for ATL data
0	DMAReadWriteSelect	**0 —** read from the HC FIFO buffer RAM **1 —** write to the HC FIFO buffer RAM

10.4.3 HcTransferCounter register (R/W: 22H/A2H)

This register holds the number of bytes of a PIO or DMA transfer. For a PIO transfer, the number of bytes being read or written to the Isochronous Transfer List (ITL) or Acknowledged Transfer List (ATL) buffer RAM must be written into this register. For a DMA transfer, the number of bytes must be written into this register as well. However, for this counter to be read into the DMA counter, the HCD must set bit 2 (DMACounterSelect) of the HcDMAConfiguration register. The counter value for ATL must not be greater than 1000H, and for ITL it must not be greater than 800H. When the byte count of the data transfer reaches this value, the HC will generate an internal EOT signal to set bit 2 (AllEOTInterrupt) of the HcμPInterrupt register, and also update the HcBufferStatus register.

Code: 22H — read

Code: A2H — write

Table 40. HcTransferCounter register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	Counter value							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	Counter value							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41. HcTransferCounter register: bit description

Bit	Symbol	Description
15 to 0	Counter value	The number of data bytes to be read to or written from RAM.

10.4.4 HcμPInterrupt register (R/W: 24H/A4H)

All the bits in this register will be active on power-on reset. However, none of the active bits will cause an interrupt on the interrupt pin (INT) unless they are set by the respective bits in the HcμPInterruptEnable register, and together with bit 0 of the HcHardwareConfiguration register.

After this register (24H to read) is read, the bits that are active will not be reset, until logic 1 is written to the bits in this register (A4H to write) to clear it. To clear all the enabled bits in this register, the HCD must write FFH to this register.

Code: 24H — read

Code: A4H — write

Table 42. HcμPInterrupt register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved	ClkReady	HC Suspended	OPR_Reg	reserved	AllEOT Interrupt	ATLInt	SOFITLInt
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43. HcµPInterrupt register: bit description

Bit	Symbol	Description
15 to 7	-	reserved
6	ClkReady	**0 —** no event **1 —** clock is ready. After a wake-up is sent, there is a wait for clock ready. Maximum is 1 ms, and typical is 160 µs.
5	HC Suspended	**0 —** no event **1 —** the HC has been suspended. When the microprocessor wants to suspend the HC, the microprocessor must write to the HcControl register. And when all downstream devices are suspended, then the HC stops sending SOF; the HC is suspended by having the HcControl register written into.
4	OPR_Reg	**0 —** no event **1 —** there are interrupts from HC side. Need to read HcControl and HcInterrupt registers to detect type of interrupt on the HC (if the HC requires the operational register to be updated).
3	-	reserved
2	AllEOT Interrupt	**0 —** no event **1 —** implies that data transfer has been completed via PIO transfer or DMA transfer. Occurrence of internal or external EOT will set this bit.
1	ATLInt	**0 —** no event **1 —** implies that the microprocessor must read ATL data from the HC. This requires that the HcBufferStatus register must first be read.
0	SOFITLInt	**0 —** no event **1 —** implies that SOF indicates the 1 ms mark. The ITL buffer that the HC has handled must be read. To know the ITL buffer status, the HcBufferStatus register must first be read. This is for the microprocessor to get ISO data to or from the HC. For more information, see the sixth paragraph in Section 9.5.

10.4.5 HcµPInterruptEnable register (R/W: 25H/A5H)

The bits 6:0 in this register are the same as those in the HcµPInterrupt register. They are used together with bit 0 of the HcHardwareConfiguration register to enable or disable the bits in the HcµPInterrupt register.

On power-on, all bits in this register are masked with logic 0. This means no interrupt request output on the interrupt pin INT can be generated.

When the bit is set to logic 1, the interrupt for the bit is not masked but enabled.

Code: 25H — read

Code: A5H — write

Table 44. HcμPInterruptEnable register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved	ClkReady	HC Suspended Enable	OPR Interrupt Enable	reserved	EOT Interrupt Enable	ATL Interrupt Enable	SOF Interrupt Enable
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45. HcμPInterruptEnable register: bit description

Bit	Symbol	Description
15 to 7	-	reserved
6	ClkReady	**0 —** power-up value **1 —** enables ClkReady interrupt
5	HC Suspended Enable	**0 —** power-up value **1 —** enables HC suspended interrupt. When the microprocessor wants to suspend the HC, the microprocessor must write to the HcControl register. And when all downstream devices are suspended, then the HC stops sending SOF; the HC is suspended by having the HcControl register written into.
4	OPR Interrupt Enable	**0 —** power-up value **1 —** enables the 32-bit operational registers interrupt (if the HC requires the operational register to be updated)
3	-	reserved
2	EOT Interrupt Enable	**0 —** power-up value **1 —** enables the EOT interrupt which indicates an end of a read/write transfer
1	ATL Interrupt Enable	**0 —** power-up value **1 —** enables ATL interrupt.
0	SOF Interrupt Enable	**0 —** power-up value **1 —** enables the interrupt bit due to SOF (for the microprocessor DMA to get ISO data from the HC by first accessing the HcDMAConfiguration register)

10.5 HC miscellaneous registers

10.5.1 HcChipID register (R: 27H)

Read this register to get the ID of the SAA1160A silicon chip. The higher byte stands for the product name. The lower byte indicates the revision number of the product including engineering samples.

Code: 27H — read

Table 46. HcChipID register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	ChipID[15:8]							
Reset	0	1	1	0	0	0	0	1
Access	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Symbol	ChipID[7:0]							
Reset	0	0	1	0	0	0	1	1
Access	R	R	R	R	R	R	R	R

Table 47. HcChipID register: bit description

Bit	Symbol	Description
15 to 0	ChipID[15:0]	SAA1160As chip ID

10.5.2 HcScratch register (R/W: 28H/A8H)

This register is for the HCD to save and restore values when required.

Code: 28H — read

Code: A8H — write

Table 48. HcScratch register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	Scratch[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Symbol	Scratch[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 49. HcScratch register: bit description

Bit	Symbol	Description
15 to 0	Scratch[15:0]	Scratch register value

10.5.3 HcSoftwareReset register (W: A9H)

This register provides a means for software reset of the HC. To reset the HC, the HCD must write a reset value of F6H to this register. Upon receiving the reset value, the HC resets all the registers except its buffer memory.

Code: A9H — write

Table 50. HcSoftwareReset register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	Reset[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	Reset[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W

Table 51. HcSoftwareReset register: bit description

Bit	Symbol	Description
15 to 0	Reset[15:0]	Writing a reset value of F6H will cause the HC to reset all the registers except its buffer memory.

10.6 HC buffer RAM control registers

10.6.1 HcITLBufferLength register (R/W: 2AH/AAH)

Write to this register to assign the ITL buffer size in bytes: ITL0 and ITL1 are assigned the same value. For example, if HcITLBufferLength register is set to 2 kB, then ITL0 and ITL1 would be allocated 2 kB each.

Must follow the formula:

ATL buffer length + 2 × (ITL buffer size) $\leq$ 1000H (that is, 4 kB)

where: ITL buffer size = ITL0 buffer length = ITL1 buffer length.

Code: 2AH — read

Code: AAH — write

Table 52. HcITLBufferLength register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	ITLBufferLength[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	ITLBufferLength[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 53. HcITLBufferLength register: bit description

Bit	Symbol	Description
15 to 0	ITLBufferLength[15:0]	Assign ITL buffer length

10.6.2 HcATLBufferLength register (R/W: 2BH/ABH)

Write to this register to assign ATL buffer size.

Code: 2BH — read

Code: ABH — write

Remark: The maximum total RAM size is 1000H (4096 in decimal) bytes. That means ITL0 (length) + ITL1 (length) + ATL (length) ≤ 1000H bytes. For example, if ATL buffer length has been set to be 800H, then the maximum ITL buffer length can only be set as 400H.

Table 54. HcATLBufferLength register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	ATLBufferLength[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	ATLBufferLength[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 55. HcATLBufferLength register: bit description

Bit	Symbol	Description
15 to 0	ATLBufferLength[15:0]	Assign ATL buffer length

10.6.3 HcBufferStatus register (R: 2CH)

Code: 2CH — read

Table 56. HcBufferStatus register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	reserved		ATLBuffer Done	ITL1Buffer Done	ITL0Buffer Done	ATLBuffer Full	ITL1Buffer Full	ITL0Buffer Full
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 57. HcBufferStatus register: bit description

Bit	Symbol	Description
15 to 6	-	reserved
5	ATLBufferDone	**0 —** ATL Buffer not read by HC yet **1 —** ATL Buffer read by HC
4	ITL1BufferDone	**0 —** ITL1 Buffer not read by HC yet **1 —** ITL1 Buffer read by HC
3	ITL0BufferDone	**0 —** ITL0 Buffer not read by HC yet **1 —** ITL0 Buffer read by HC
2	ATLBufferFull	**0 —** ATL Buffer is empty **1 —** ATL Buffer is full
1	ITL1BufferFull	**0 —** ITL1 Buffer is empty **1 —** ITL1 Buffer is full
0	ITL0BufferFull	**0 —** ITL0 Buffer is empty **1 —** ITL0 Buffer is full

10.6.4 HcReadBackITL0Length register (R: 2DH)

This registers value stands for the current number of data bytes inside an ITL0 buffer to be read back by the microprocessor. The HCD must set the HcTransferCounter equivalent to this value before reading back the ITL0 buffer RAM.

Code: 2DH — read

Table 58. HcReadBackITL0Length register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	RdITL0BufferLength[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	RdITL0BufferLength[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 59. HcReadBackITL0Length register: bit description

Bit	Symbol	Description
15 to 0	RdITL0BufferLength[15:0]	The number of bytes for ITL0 data to be read back by the microprocessor.

10.6.5 HcReadBackITL1Length register (R: 2EH)

This registers value stands for the current number of data bytes inside the ITL1 buffer to be read back by the microprocessor. The HCD must set the HcTransferCounter equivalent to this value before reading back the ITL1 buffer RAM.

Code: 2EH — read

Table 60. HcReadBackITL1Length register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	RdITL1BufferLength[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	RdITL1BufferLength[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 61. HcReadBackITL1Length register: bit description

Bit	Symbol	Description
15 to 0	RdITL1BufferLength[15:0]	The number of bytes for ITL1 data to be read back by the microprocessor.

10.6.6 HcITLBufferPort register (R/W: 40H/C0H)

This is the ITL buffer RAM read/write port. The bits 15:8 contain the data byte that comes from the ITL buffer RAMs even address. The bits 7:0 contain the data byte that comes from the ITL buffer RAMs odd address.

Code: 40H — read

Code: C0H — write

Table 62. HcITLBufferPort register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	DataWord[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	DataWord[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 63. HcITLBufferPort register: bit description

Bit	Symbol	Description
15 to 0	DataWord[15:0]	Read/write ITL buffer RAMs two data bytes.

The HCD must set the byte count into the HcTransferCounter register and check the HcBufferStatus register before reading from or writing to the buffer. The HCD must write the command (40H to read, C0H to write) once only, and then read or write both bytes of the data word. After every read/write, the pointer of ITL buffer RAM will be automatically increased by two to point to the next data word until it reaches the value of the HcTransferCounter register; otherwise, an internal EOT signal is not generated to set bit 2 (AllEOTInterrupt) of the HcμPInterrupt register and update the HcBufferStatus register.

The HCD must take care of the fact that the internal buffer RAM is organized in bytes. The HCD must write the byte count into the HcTransferCounter register, but the HCD reads or writes the buffer RAM by 16 bits (by 1 word).

10.6.7 HcATLBufferPort register (R/W: 41H/C1H)

This is the ATL buffer RAM read/write port. Bits 15 to 8 contain the data byte that comes from the Acknowledged Transfer List (ATL) buffer RAMs odd address. Bits 7 to 0 contain the data byte that comes from the ATL buffer RAMs even address.

Code: 41H — read

Code: C1H — write

Table 64. HcATLBufferPort register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	DataWord[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	DataWord[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 65. HcATLBufferPort register: bit description

Bit	Symbol	Description
15 to 0	DataWord[15:0]	Read/write ATL buffer RAMs two data bytes.

The HCD must set the byte count into the HcTransferCounter register and check the HcBufferStatus register before reading from or writing to the buffer. The HCD must write the command (41H to read, C1H to write) once only, and then read or write both bytes of the data word. After every read/write, the pointer of ATL buffer RAM will be automatically increased by two to point to the next data word until it reaches the value of the HcTransferCounter register; otherwise, an internal EOT signal is not generated to set bit 2 (AllEOTInterrupt) of the HcµPInterrupt register and update the HcBufferStatus register.

The HCD must take care of the difference: the internal buffer RAM is organized in bytes, so the HCD must write the byte count into the HcTransferCounter register, but the HCD reads or writes the buffer RAM by 16 bits (by 1 data word).

11. Power supply

The SAA1160A can operate at either 5 V or 3.3 V.

When using 5 V as the SAA1160As power supply input, only V_{CC} (pin 56) can be connected to the 5 V power supply. An application with a 5 V power supply input is shown in Figure 29. The SAA1160A has an internal DC/DC regulator to provide 3.3 V for its internal core. This internal 3.3 V can also be obtained from $V_{REG(3V3)}$ (pin 58).

When using 3.3 V as the power supply input, the internal DC/DC regulator will be bypassed. All four power supply pins (V_{CC}, $V_{REG(3V3)}$, V_{HOLD1} and V_{HOLD2}) can be used as power supply input.

It is recommended that you connect all four power supply pins to the 3.3 V power supply, as shown in Figure 30. If, however, you have board space (routing area) constraints, you must connect at least V_{CC} and $V_{REG(3V3)}$ to the 3.3 V power supply.

For both 3.3 V and 5 V operation, all four power supply pins should be connected to a decoupling capacitor.

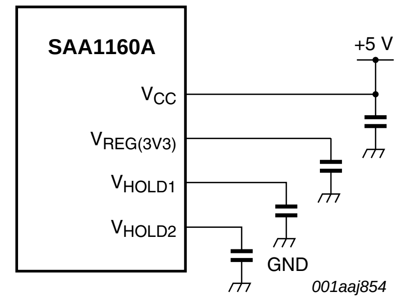


Fig 29. Using a 5 V supply

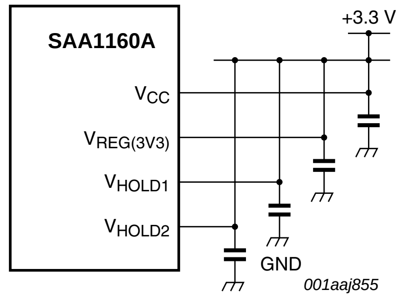


Fig 30. Using a 3.3 V supply

12. Crystal oscillator

The SAA1160A has a crystal oscillator designed for a 6 MHz parallel-resonant crystal (fundamental). A typical circuit is shown in Figure 31. Alternatively, an external clock signal of 6 MHz can be applied to input XTAL1, while leaving output XTAL2 open. See Figure 32. The 6 MHz oscillator frequency is multiplied to 48 MHz by an internal PLL.

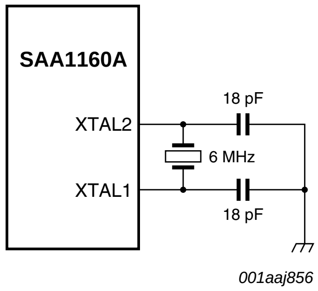


Fig 31. Oscillator circuit with external crystal

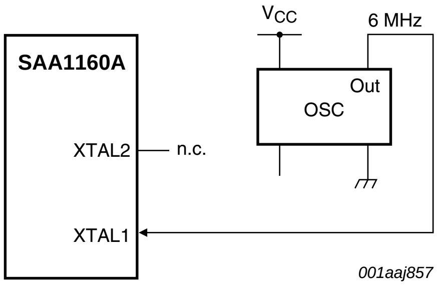
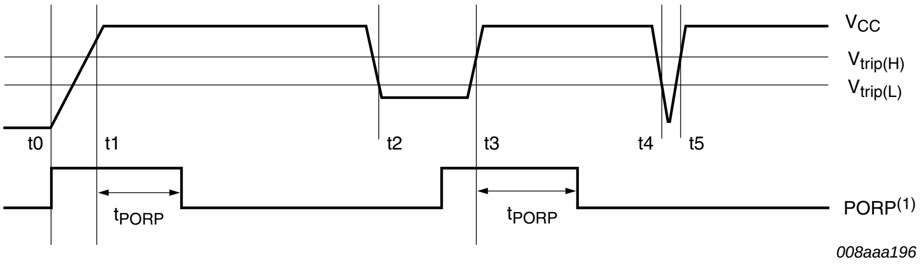


Fig 32. Oscillator circuit using external oscillator

13. Reset

13.1 Power-On Reset (POR)

Figure 33 shows a possible curve of V_{CC} with dips at t2 to t3 and t4 to t5. The internal Power-On Reset Pulse (PORP) starts with a HIGH at t0. At t1, the detector will see the passing of the trip level $V_{trip(H)}$ (typically 2.03 V) and a delay element will add another t_{PORP} (200 ns to 1000 ns) before the PORP drops to LOW. If the dip at t4 to t5 is too short (less than 11 μs), the PORP will not react and will remain LOW. A HIGH on PORP will be generated whenever V_{CC} drops below $V_{trip(L)}$ (typically 1.5 V) for more than 11 μs.



(1) PORP = internal power-on reset pulse.

Fig 33. Internal POR timing

13.2 External reset

Ensure that before asserting the RESET_N pin, signal CS_N is forced HIGH and signal A0 is forced LOW. Both signals must not toggle while RESET_N is active.

The RESET_N pin can be either connected to V_{CC} (using the internal POR circuit) or externally controlled (by the microprocessor, ASIC, and so on).

An externally controlled RESET_N has to be active as long as the external clock (XTAL1) is not stable. Further 3 ms are required to ensure stabilization of internal clocks.

14. Limiting values

Table 66. Absolute maximum ratings

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
$V_{CC(5V0)}$	supply voltage to pin V_{CC}			–0.5	+6.0	V
$V_{CC(3V3)}$	supply voltage to pin $V_{REG(3V3)}$			–0.5	+4.6	V
V_I	input voltage			–0.5	+6.0	V
I_{lu}	latch-up current	$V_I < 0$ or $V_I > V_{CC}$		-	100	mA
V_{esd}	electrostatic discharge voltage	$I_{LI} < 1$ μA	[1]	–2000	+2000	V
T_{stg}	storage temperature			–60	+150	°C

[1] Equivalent to discharging a 100 pF capacitor via a 1.5 kΩ resistor [Human Body Model (HBM)].

15. Recommended operating conditions

Table 67. Operating conditions

D+ is the USB ports positive data pin; D– is the USB ports negative data pin.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{CC}	supply voltage	with internal regulator		4.0	5.0	5.5	V
		internal regulator bypass		3.0	3.3	3.6	V
V_I	input voltage		[1]	0	V_{CC}	5.5	V
$V_{I(AI/O)}$	input voltage on analog I/O pins D+ and D–			0	-	3.6	V
$V_{O(od)}$	open-drain output pull-up voltage			0	-	V_{CC}	V
T_{amb}	ambient temperature			–40	-	+85	°C

[1] Maximum value is 5 V tolerant.

16. Static characteristics

Table 68. Static characteristics; supply pins

V_{CC} = 3.0 V to 3.6 V or 4.0 V to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; typical values at T_{amb} = 25 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{CC} = 5 V							
$V_{REG(3V3)}$	internal regulator output		[1]	3.0	3.3	3.6	V
I_{CC}	operating supply current			-	47	-	mA
$I_{CC(susp)}$	suspend supply current	see Section 9.9.1		-	-	2500	μA
V_{CC} = 3.3 V							
I_{CC}	operating supply current			-	50	-	mA

[1] In the suspend mode, the minimum voltage is 2.7 V.

Table 69. Static characteristics: digital pins

V_{CC} = 3.0 V to 3.6 V or 4.0 V to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Input levels							
V_{IL}	LOW-level input voltage			-	-	0.8	V
V_{IH}	HIGH-level input voltage			2.0	-	-	V
Schmitt trigger inputs							
$V_{th(LH)}$	positive-going threshold voltage			1.4	-	1.9	V
$V_{th(HL)}$	negative-going threshold voltage			0.9	-	1.5	V
V_{hys}	hysteresis voltage			0.4	-	0.7	V
Output levels							
V_{OL}	LOW-level output voltage	I_{OL} = 4 mA		-	-	0.4	V
		I_{OL} = 20 μA	[1]	-	-	0.1	V
V_{OH}	HIGH-level output voltage	I_{OH} = 4 mA	[1]	2.4	-	-	V
		I_{OH} = 20 μA	[1]	$V_{REG(3V3)}$ − 0.1	-	-	V
Leakage current							
I_{LI}	input leakage current		[2]	−5	-	+5	μA
C_{IN}	pin capacitance	pin to GND		-	-	5	pF
Open-drain outputs							
I_{OZ}	OFF-state output current			−5	-	+5	μA

[1] Not applicable for open-drain outputs.

[2] The maximum and minimum values are applicable to transistor input only. The value will be different if internal pull-up or pull-down resistors are used.

Table 70. Static characteristics: analog I/O pins D+ and D−

V_{CC} = 3.0 V to 3.6 V or 4.0 V to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; D+ is the USB ports positive data pin; D− is the USB ports negative data pin; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Input levels						
V_{DI}	differential input sensitivity	$\lvert V_{I(D+)} - V_{I(D-)}\rvert$	0.2	-	-	V
V_{CM}	differential common mode voltage	includes V_{DI} range	0.8	-	$V_{REG(3V3)} - 0.9$	V
V_{IL}	LOW-level input voltage		-	-	0.8	V
V_{IH}	HIGH-level input voltage		2.0	-	-	V
Output levels						
V_{OL}	LOW-level output voltage	R_L = 1.5 kΩ to +3.6 V	-	-	0.3	V
V_{OH}	HIGH-level output voltage	R_L = 15 kΩ to GND	$V_{REG(3V3)} - 0.5$	-	3.6	V
Leakage current						
I_{LZ}	OFF-state leakage current		−10	-	+10	μA
Capacitance						
C_{IN}	transceiver capacitance	pin to GND	-	-	10	pF
Resistance						
R_{PD}	pull-down resistance on HCs D+/D−	enable internal resistors	10	-	20	kΩ
Z_{DRV}	driver output impedance	steady-state drive [1]	28	-	44	Ω
Z_{INP}	input impedance		10	-	-	MΩ

[1] Includes external resistors of 18 Ω ± 1 % on each of pins H_DM1, H_DM2, H_DP1 and H_DP2.

17. Dynamic characteristics

Table 71. Dynamic characteristics

V_{CC} = 3.0 V to 3.6 V or 4.0 V to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Crystal oscillator						
f_{XTAL}	crystal frequency		-	6	-	MHz
R_S	series resistance		-	-	100	Ω
C_{LOAD}	load capacitance	C_{x1}, C_{x2} = 22 pF	-	18	-	pF
External clock input						
t_J	external clock jitter		-	-	500	ps
t_{DUTY}	clock duty cycle		45	50	55	%
t_{CR}, t_{CF}	rise time and fall time		-	-	3	ns

Table 72. Dynamic characteristics: analog I/O pins D+ and D−

V_{CC} = 3.0 V to 3.6 V or 4.0 V to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; C_L = 50 pF; full-speed mode only; D+ is the USB ports positive data pin; D− is the USB ports negative data pin; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Driver characteristics							
t_{FR}	rise time	10 % to 90 % of $\|V_{OH} - V_{OL}\|$		4	-	20	ns
t_{FF}	fall time	90 % to 10 % of $\|V_{OH} - V_{OL}\|$		4	-	20	ns
FRFM	differential rise/fall time matching (t_{FR}/t_{FF})		[1]	90	-	111.11	%
V_{CRS}	output signal crossover voltage		[1]	1.3	-	2.0	V

[1] Excluding the first transition from Idle state.

17.1 Programmed I/O timing

Table 73. Dynamic characteristics: programmed interface timing

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{AS}	address set-up time before WR_N HIGH		5	-	-	ns
t_{AH}	address hold time after WR_N HIGH		8	-	-	ns
Read timing						
t_{SHSL}	first RD_N/WR_N after A0 HIGH		300	-	-	ns
t_{SLRL}	CS_N LOW to RD_N LOW		0	-	-	ns
t_{RHSH}	RD_N HIGH to CS_N HIGH		0	-	-	ns
t_{RLRH}	RD_N LOW pulse width		33	-	-	ns
t_{RHRL}	RD_N HIGH to next RD_N LOW		110	-	-	ns
T_{RC}	RD_N cycle		143	-	-	ns
t_{RHDZ}	RD_N data hold time		3	-	22	ns
t_{RLDV}	RD_N LOW to data valid		-	-	32	ns
Write timing						
t_{WL}	WR_N LOW pulse width		26	-	-	ns
t_{WHWL}	WR_N HIGH to next WR_N LOW		110	-	-	ns
T_{WC}	WR_N cycle		136	-	-	ns
t_{SLWL}	CS_N LOW to WR_N LOW		0	-	-	ns
t_{WHSH}	WR_N HIGH to CS_N HIGH		0	-	-	ns
t_{WDSU}	WR_N data set-up time		5	-	-	ns
t_{WDH}	WR_N data hold time		8	-	-	ns

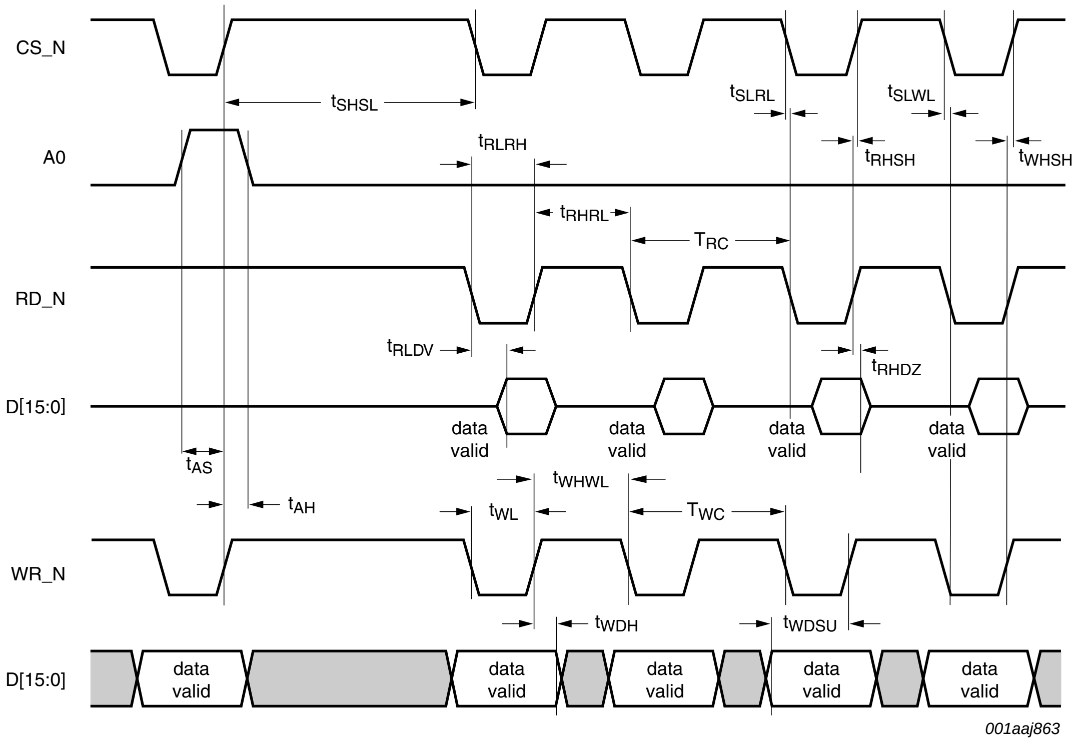


Fig 34. Programmed interface timing

17.2 DMA timing

17.2.1 Single-cycle DMA timing

Table 74. Dynamic characteristics: single-cycle DMA timing

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Read/write timing						
t_{RLRH}	RD_N pulse width		33	-	-	ns
t_{RLDV}	read process data set-up time		26	-	-	ns
t_{RHDZ}	read process data hold time		0	-	20	ns
t_{WSU}	write process data set-up time		5	-	-	ns
t_{WHD}	write process data hold time		0	-	-	ns
t_{AHRH}	DACK_N HIGH to DREQ HIGH		72	-	-	ns
t_{ALRL}	DACK_N LOW to DREQ LOW		-	-	21	ns
T_{DC}	DREQ cycle	[1]	-	-	-	ns
t_{SHAH}	RD_N/WR_N HIGH to DACK_N HIGH		0	-	-	ns
t_{RHAL}	DREQ HIGH to DACK_N LOW		0	-	-	ns
t_{DS}	DREQ pulse spacing		146	-	-	ns

[1] $T_{DC} = t_{RHAL} + t_{DS} + t_{ALRL}$.

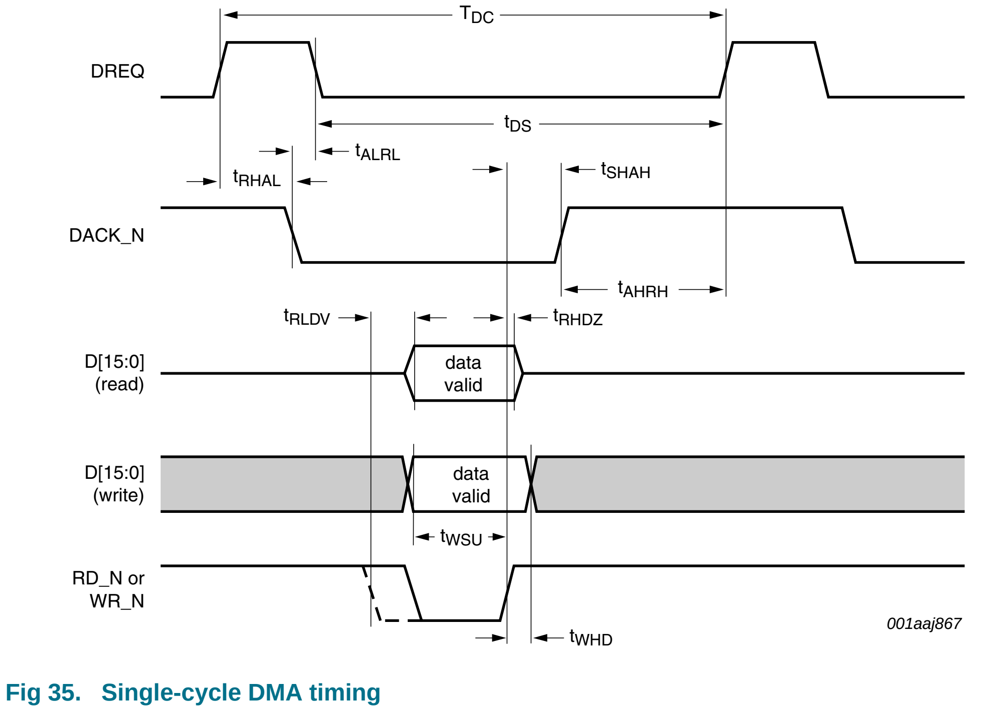


Fig 35. Single-cycle DMA timing

17.2.2 Burst mode DMA timing

Table 75. Dynamic characteristics: burst mode DMA timing

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Read/write timing (for 4-cycle and 8-cycle burst mode)						
t_{RLRH}	WR_N/RD_N LOW pulse width		42	-	-	ns
t_{RHRL}	WR_N/RD_N HIGH to next WR_N/RD_N LOW		60	-	-	ns
T_{RC}	WR_N/RD_N cycle		102	-	-	ns
t_{SLRL}	RD_N/WR_N LOW to DREQ LOW		22	-	64	ns
t_{SHAH}	RD_N/WR_N HIGH to DACK_N HIGH		0	-	-	ns
t_{SLAL}	DREQ HIGH to DACK_N LOW		0	-	-	ns
T_{DC}	DREQ cycle	[1]	-	-	-	ns
$t_{DS(read)}$	DREQ pulse spacing (read)	4-cycle burst mode	105	-	-	ns
$t_{DS(read)}$	DREQ pulse spacing (read)	8-cycle burst mode	150	-	-	ns
$t_{DS(write)}$	DREQ pulse spacing (write)	4-cycle burst mode	72	-	-	ns
$t_{DS(write)}$	DREQ pulse spacing (write)	8-cycle burst mode	167	-	-	ns
t_{RLIS}	RD_N/WR_N LOW to EOT LOW		0	-	-	ns

[1] $T_{DC} = t_{SLAL} + (4 \text{ or } 8)T_{RC} + t_{DS}$.

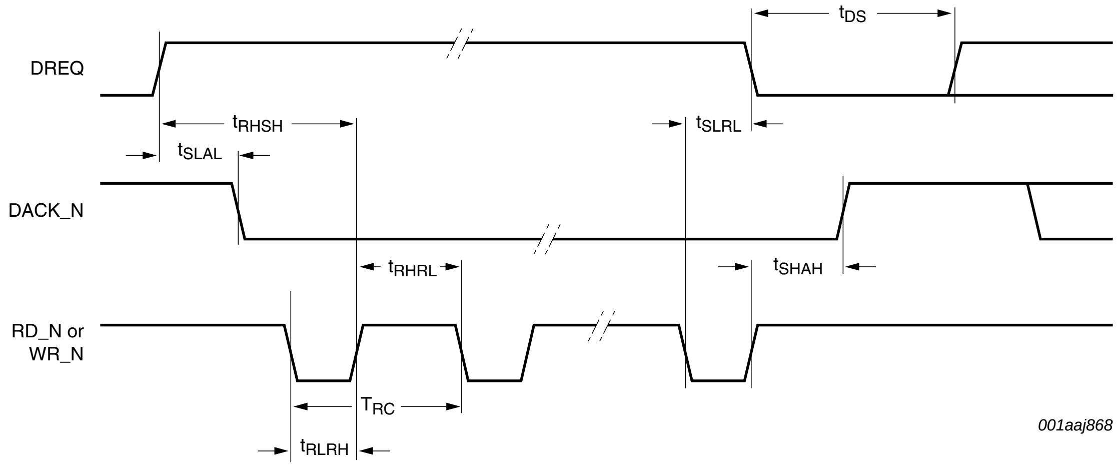


Fig 36. Burst mode DMA timing

17.2.3 External EOT timing for single-cycle DMA

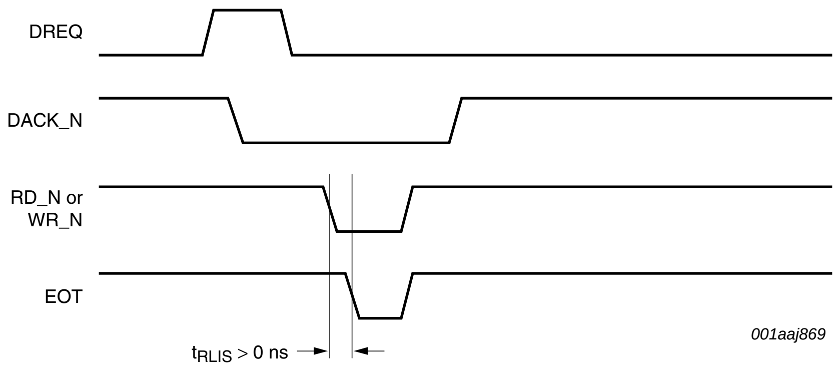


Fig 37. External EOT timing for single-cycle DMA

17.2.4 External EOT timing for burst mode DMA

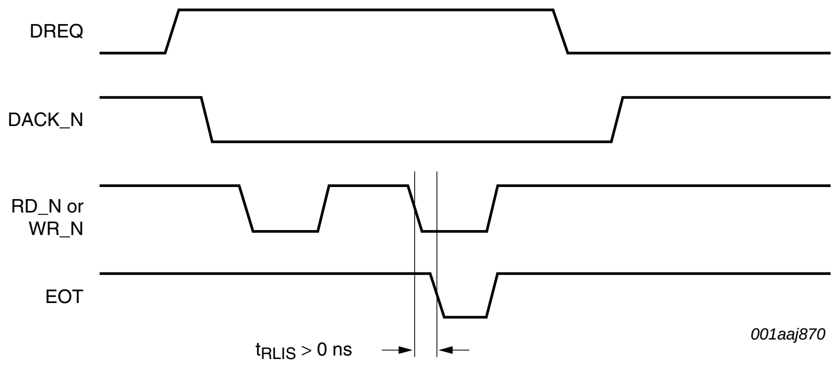
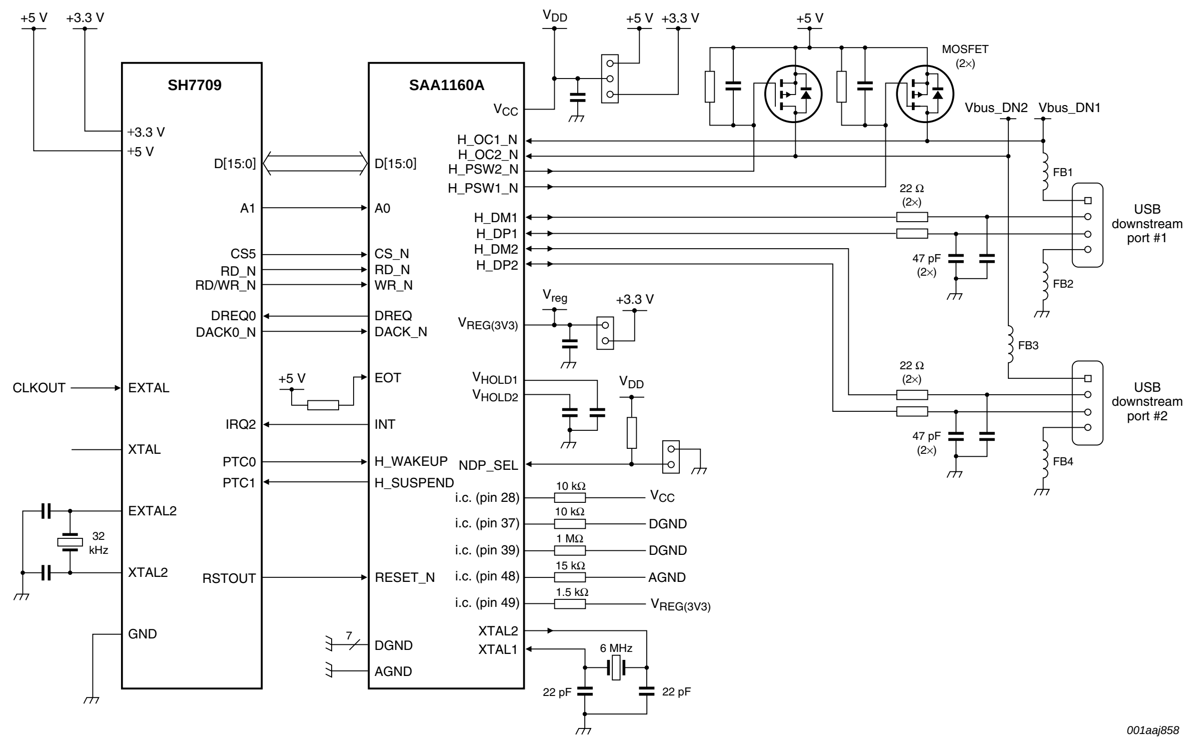


Fig 38. External EOT timing for burst mode DMA

18. Application information

18.1 Typical interface circuit



For MOSFET, R_{DSon} = 150 mΩ.

Fig 39. Typical interface circuit to Hitachi SH-3 (SH7709) RISC processor

18.2 Interfacing an SAA1160A to an SH7709 RISC processor

This section shows a typical interface circuit between the SAA1160A and a RISC processor. The Hitachi SH-3 series RISC processor SH7709 is used as the example. The main SAA1160A signals to be taken into consideration for connecting to an SH7709 RISC processor are:

- A 16-bit data bus: D[15:0] for the SAA1160A. The SAA1160A is ‘little endian’ compatible.
- The address line A0 is needed for a complete addressing of the SAA1160A internal registers:
  - A0 = 0 will select the Data Port of the Host Controller
  - A0 = 1 will select the Command Port of the Host Controller
- The CS_N line is used for chip selection of the SAA1160A in a certain address range of the RISC system. This signal is active LOW.
- RD_N and WR_N are common read and write signals. These signals are active LOW.
- There is a DMA channel standard control line: DREQ and DACK_N. The DREQ signal has programmable active levels.
- An interrupt line INT is used by the HC. It has programmable level/edge and polarity (active HIGH or LOW).
- The internal 15 kΩ pull-down resistors are used for the HCs two USB downstream ports.
- The RESET_N signal is active LOW.

Remark: SH7709s system clock input is for reference only. Refer to SH7709s specification for its actual use.

The SAA1160A can work under either 3.3 V or 5.0 V power supply; however, its internal core works at 3.3 V. When using 3.3 V as the power supply input, the internal DC/DC regulator will be bypassed. It is best to connect all four power supply pins (V_{CC}, $V_{REG(3V3)}$, V_{HOLD1} and V_{HOLD2}) to the 3.3 V power supply (for more information, see Section 11). All of the SAA1160As I/O pins are 5 V tolerant. This feature allows the SAA1160A the flexibility to be used in an embedded system under either a 3.3 V or a 5 V power supply.

A typical SH7709 interface circuit is shown in Figure 39.

18.3 Typical software model

This section shows a typical software requirement for an embedded system that incorporates the SAA1160A. The software model for a Digital Still Camera (DSC) is used as the example for illustration (as shown in Figure 40). The host stack provides API for Class driver and device driver, both of which provide API for application tasks for host function.

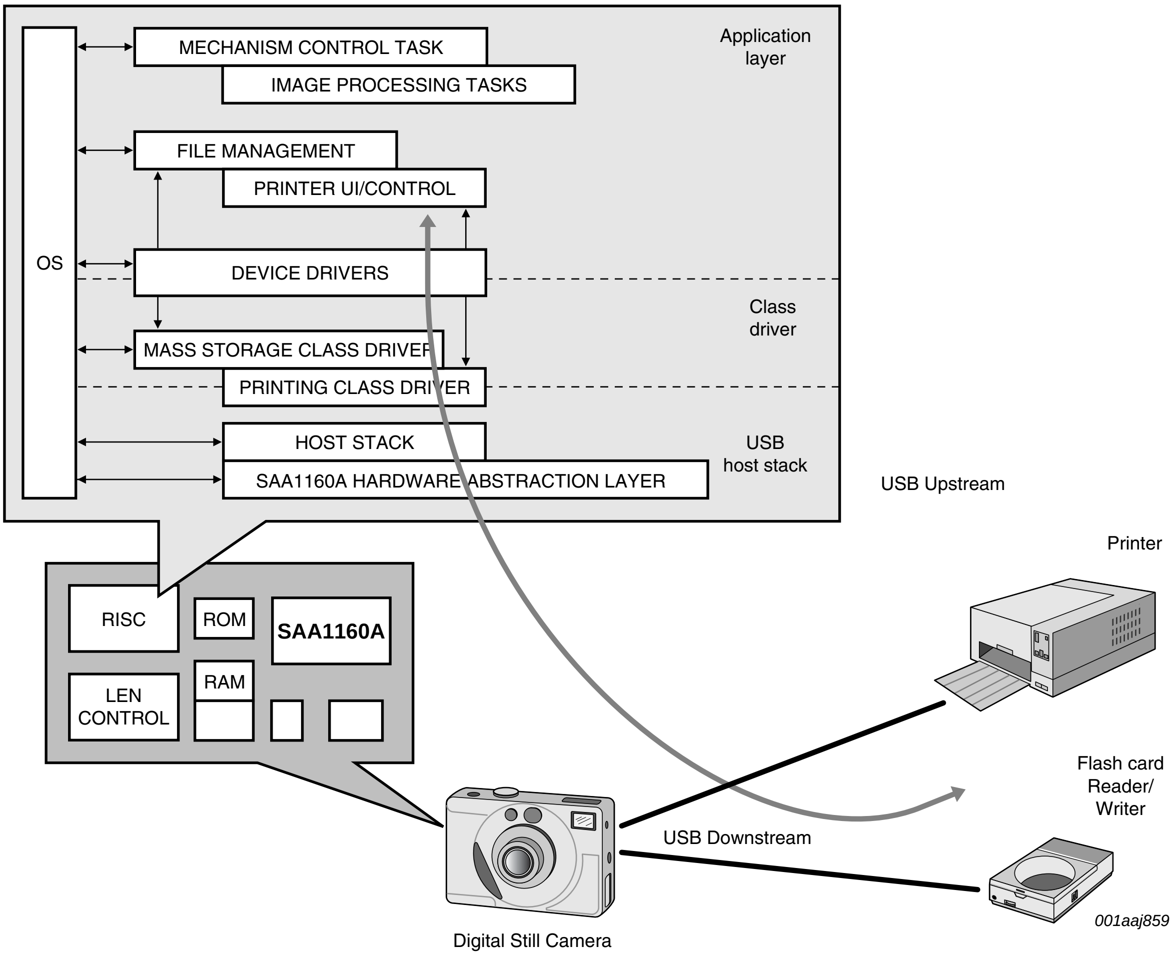
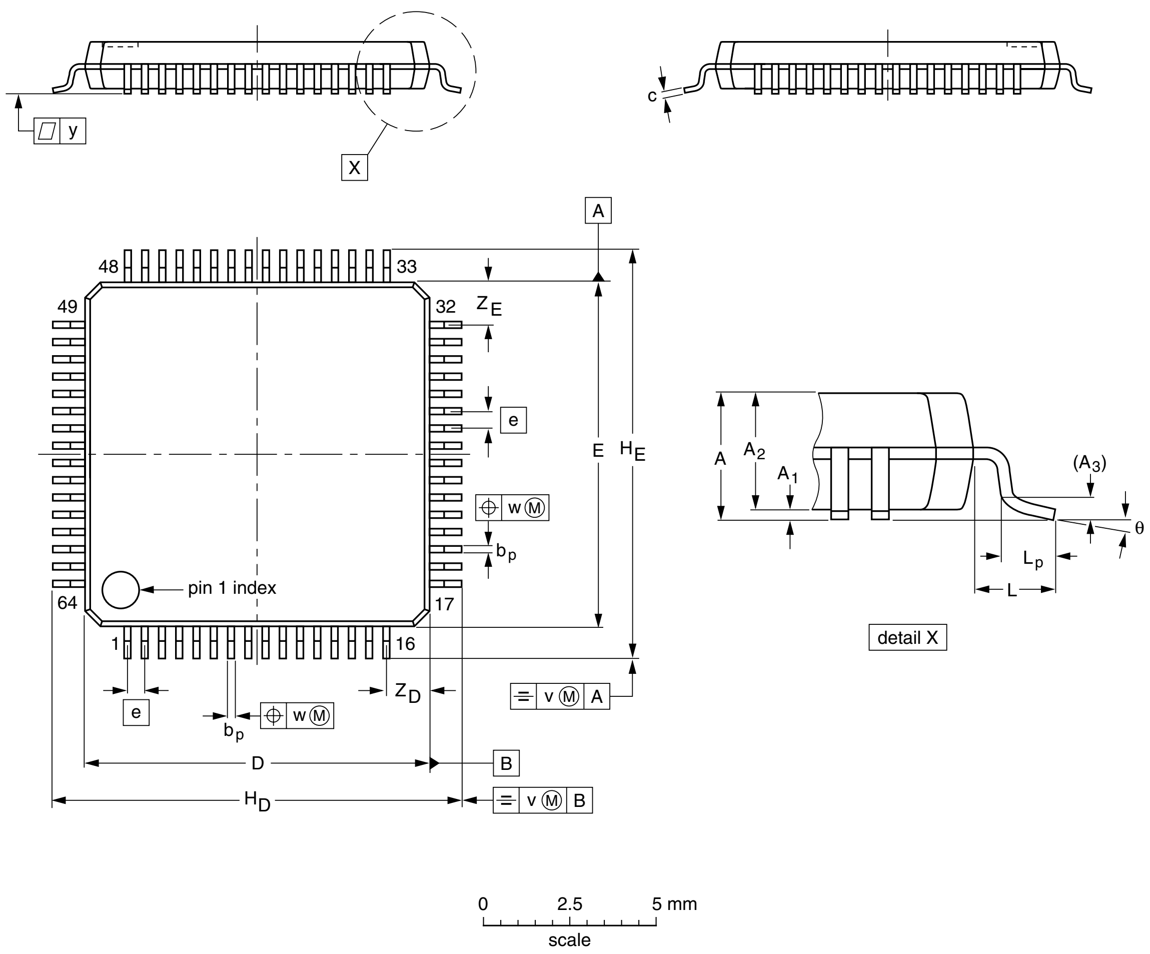


Fig 40. The SAA1160A software model for DSC application

19. Package outline

LQFP64: plastic low profile quad flat package; 64 leads; body 10 x 10 x 1.4 mm **SOT314-2**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.20 0.05	1.45 1.35	0.25	0.27 0.17	0.18 0.12	10.1 9.9	10.1 9.9	0.5	12.15 11.85	12.15 11.85	1	0.75 0.45	0.2	0.12	0.1	1.45 1.05	1.45 1.05	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT314-2	136E10	MS-026				~~00-01-19~~ 03-02-25

Fig 41. Package outline SOT314-2 (LQFP64)

20. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 “Surface mount reflow soldering description”*.

20.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

20.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

20.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

20.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 42) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 76 and 77

Table 76. SnPb eutectic process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm^3)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

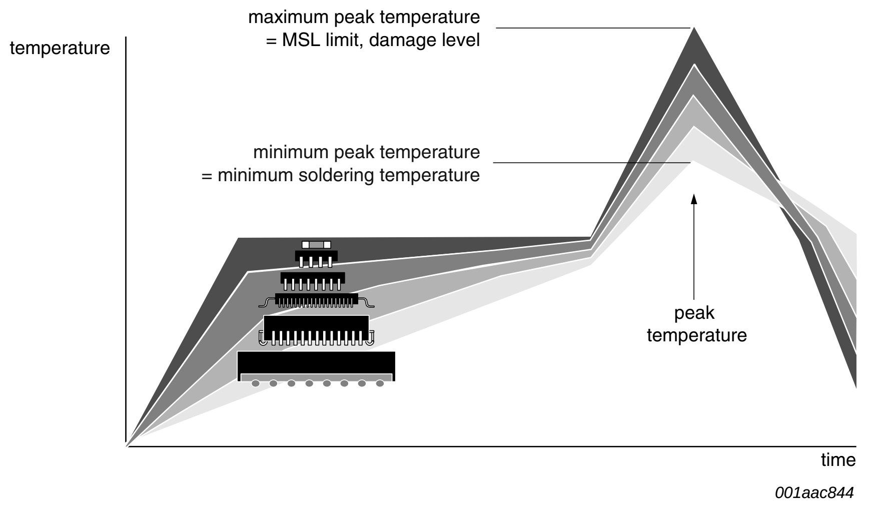
Table 77. Lead-free process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm^3)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 42.

All information provided in this document is subject to legal disclaimers.



MSL: Moisture Sensitivity Level

Fig 42. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

21. Appendix

21.1 Erratum 1: Bandwidth limitation due to buffer lock zone

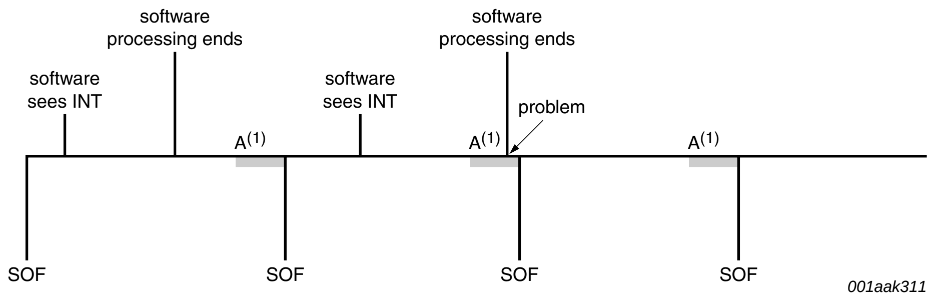
The ATLBufferDone and ITL1BufferDone or ITL0BufferDone bits are never set but the ATLBufferFull and ITL1BufferFull or ITL0BufferFull bits are always set.

21.1.1 Problem description

In certain bandwidth-intensive applications and in certain system scenarios, there is a possibility that the ATLBufferFull and ITL1BufferFull or ITL0BufferFull bits in the HcBufferStatus register are always set, and ATLBufferDone and ITL1BufferDone or ITL0BufferDone bits are never set.

The SAA1160A is designed so that whatever is written to the ATL or ITL buffer RAM in the present millisecond is processed in the next millisecond frame and appears on the USB bus. Once an interrupt occurs, software starts to process PTDs and if there are any pending PTDs from the software, the Host Controller Driver (HCD) writes them to the SAA1160A buffer. If the processor is slow or USB Interrupt Service Routine (ISR) has a low priority in the system, there will be a delay in processing the SOF interrupt. This delay causes the HCD to write the new PTD to the SAA1160A buffer RAM (or to read the PTD from the buffer) up to the SOF boundary. If the ATL or ITL PTD completion occurs in the buffer lock zone, then the ATLBufferDone or ITL1BufferDone/ITL0BufferDone bit is locked and always shown as zero, and the ATLBufferFull or ITL1BufferFull/ITL0BufferFull bit is always set as full; see Figure 43.

The buffer lock zone region is 500 ns from the boundary of SOF (1 ms to 500 ns). Within this zone, the ATL and ITL buffers are locked, and the HCD must not allow the ATL and ITL PTD completion (PTD header and payload) to occur in this zone. This is applicable for both ATL/ITL read and write. The buffer lock zone is indicated in Figure 43.



(1) Bars labeled A indicate buffer lock zone.

Fig 43. Buffer lock zone

The cut-off time for the SAA1160A core to process the new PTD is not at the start of the new frame but about 500 ns before the SOF. Therefore, if the HCD read/write to the ATL or ITL buffer is completed (PTD completion) in the buffer lock zone, the SAA1160A core logic gets confused. This results in the ATLBufferFull or ITL1BufferFull/ITL0BufferFull bit getting permanently set. But the ATLBufferDone or ITL1BufferDone/ITL0BufferDone bit never gets set.

21.1.2 Implications

The implication is serious in bandwidth intensive applications and slow processor systems. Once the preceding condition is met, the ATLBufferFull and ITL1BufferFull or ITL0BufferFull bits are always set and the ATLBufferDone and ITL1BufferDone or ITL0BufferDone bits are never set. When the ATLBufferFull bit is set, then HCD cannot perform any control, bulk or interrupt transfer. When the ITL0BufferFull bit is set, then isochronous transfers continue to occur with a reduced bandwidth until ITL1BufferFull is also locked forever, and vice versa.

21.1.3 Workaround

The workaround is to ensure that the ATL or ITL buffer read/write completion does not occur in the buffer lock zone.

Consider a platform with an I/O access timing (read/write cycle time) of 500 ns and the application needs to transfer 600 bytes in a single ATL PTD. Then, the total time taken by the HCD to write 600 bytes to the SAA1160A is 150 μs [(600 / 2) × 500]. Therefore, the HCD must ensure that the processing of ATL starts with at least 200 μs (200 μs is an approximate value, this can be 150 μs + 500 ns) of the frame remaining in the SOF.

In the SAA1160A ISR, before processing ATL PTDs (control, bulk and interrupt), read the HcFmRemaining register and if the frame remaining value is more than 200 μs, then process only ATL PTDs. Otherwise, skip the ATL processing. See the pseudo code.

```
#define 200_MICROSECONDS  0x95F /* 0x2edf (1ms) /5 */

fnuSOFHandler{
     fnvIsp1161HcorRead(ohci,uHcFmRemaining, &uData1);

     if(uData1 > 200_MICROSECONDS)
          /* Make sure uItlByteCount is a multiple of 4 */
          fnvIsp1161ItlWrite(pbyHcAItlBuff, uItlByteCount);
     else
          /* Exit without writing to SAA1160A FIFO RAM */
}
```

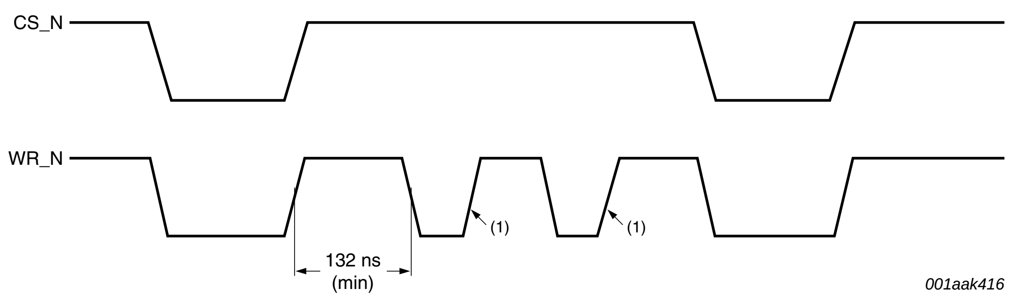
Remark: The I/O access timing is platform dependent.

The PTD payload size depends on the application. The ISO handling can also be implemented as shown earlier.

21.2 Erratum 2: Data corruption during write operations

21.2.1 Problem description

After a data write operation to any auxiliary register, the SAA1160A requires a 132 ns (minimum) delay before a write assertion can be issued for other devices. This must be fulfilled even after CS_N is deasserted; see Figure 44.



(1) These writes are not for the SAA1160A because CS_N is deasserted.

Fig 44. Timing

21.2.2 Implications

If the timing requirements in Figure 44 are not handled this will lead to data corruption.

21.2.3 Workaround

Ensure that the system can handle the write timing requirements as given in Figure 44. If the system is really fast and needs immediate write accesses to other devices, then it is better to qualify WR_N with respect to CS_N and provide the resultant write signal to the SAA1160A; see Figure 45.

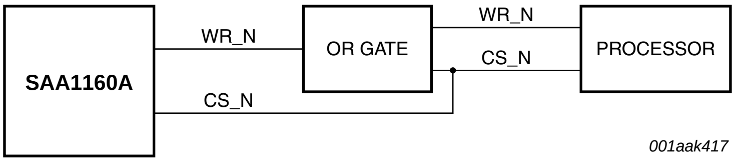


Fig 45. Workaround

Remark: Take into consideration the propagation delay of the logics.

22. Revision history

Table 78. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
SAA1160A_1	20100520	Product data sheet	-	-

23. Legal information

23.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term ‘short data sheet’ is explained in section “Definitions”.

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

23.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

23.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors’ aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer’s own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer’s sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer’s applications and products planned, as well as for the planned application and use of customer’s third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer’s applications or products, or the application or use by customer’s third party customer(s). Customer is responsible for doing all necessary testing for the customer’s applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer’s third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer’s general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

23.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

24. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

25. Contents

continued >>

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section ‘Legal information’.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 20 May 2010
Document identifier: SAA1160A_1