ISP1160

Embedded Universal Serial Bus Host Controller

Rev. 03 — 27 February 2003   **Product data**

1. General description

The ISP1160 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 ISP1160 provides two downstream ports. Each downstream port has its own 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 ISP1160 is well suited for embedded systems and portable devices that require a USB host. The ISP1160 brings high flexibility to the systems that have it built-in. For example, a system that has the ISP1160 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

- 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™, StrongARM®
- Maximum data transfer rate of 15 Mbyte/s between microprocessor and HC
- Supports single-cycle and burst mode DMA operations
- Built-in FIFO buffer RAM for HC (4 kbytes)
- Endpoints with double buffering to increase throughput and ease real-time data transfer 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 to +85 °C
- Available in two LQFP64 packages (SOT314-2 and SOT414-1).

PHILIPS

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

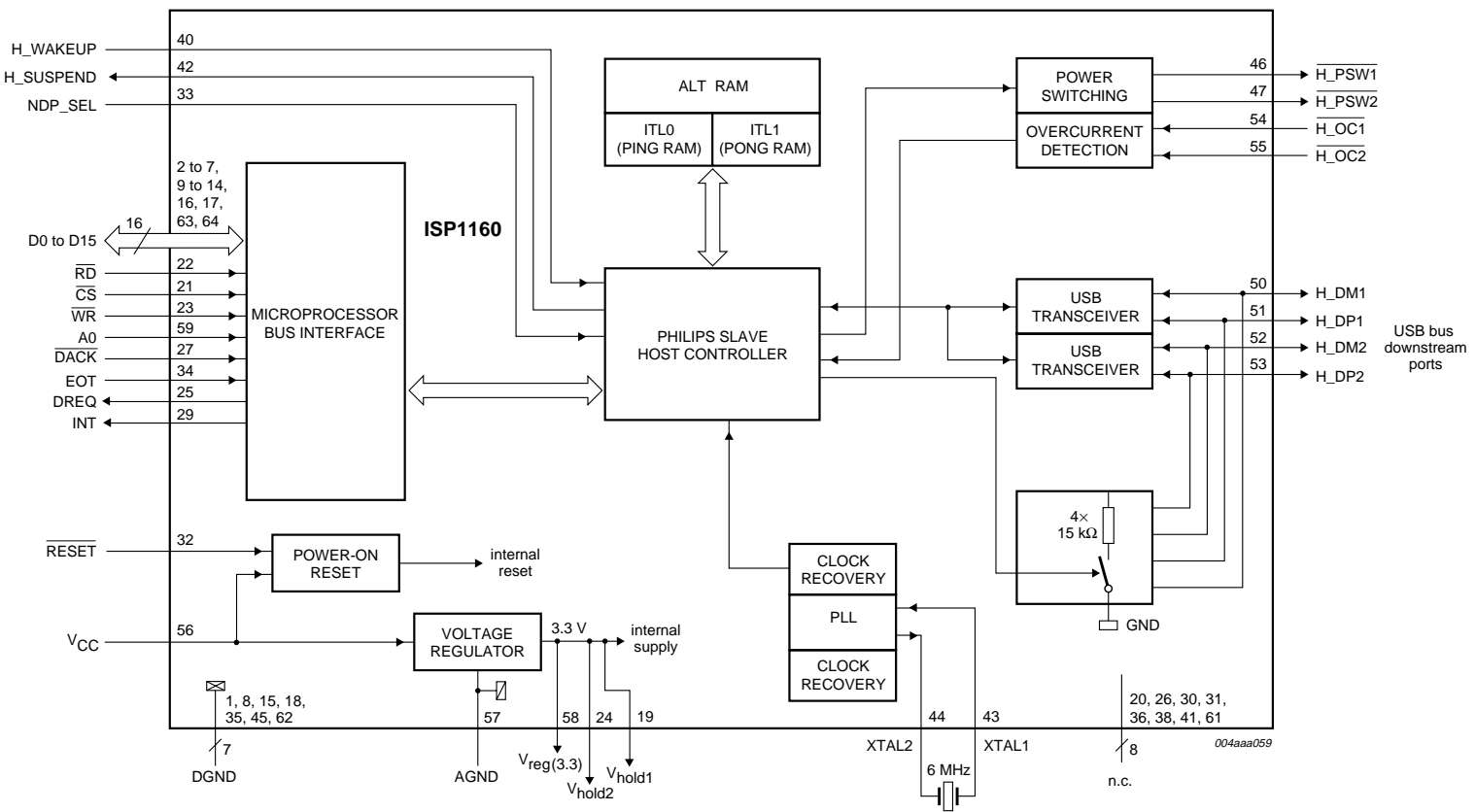| Type number	Package		
	Name	**Description**	**Version**
ISP1160BD	LQFP64	plastic low profile quad flat package; 64 leads; body 10 x 10 x 1.4 mm	SOT314-2
ISP1160BM	LQFP64	plastic low profile quad flat package; 64 leads; body 7 x 7 x 1.4 mm	SOT414-1

5. Block diagram

Fig 1.   Block diagram.
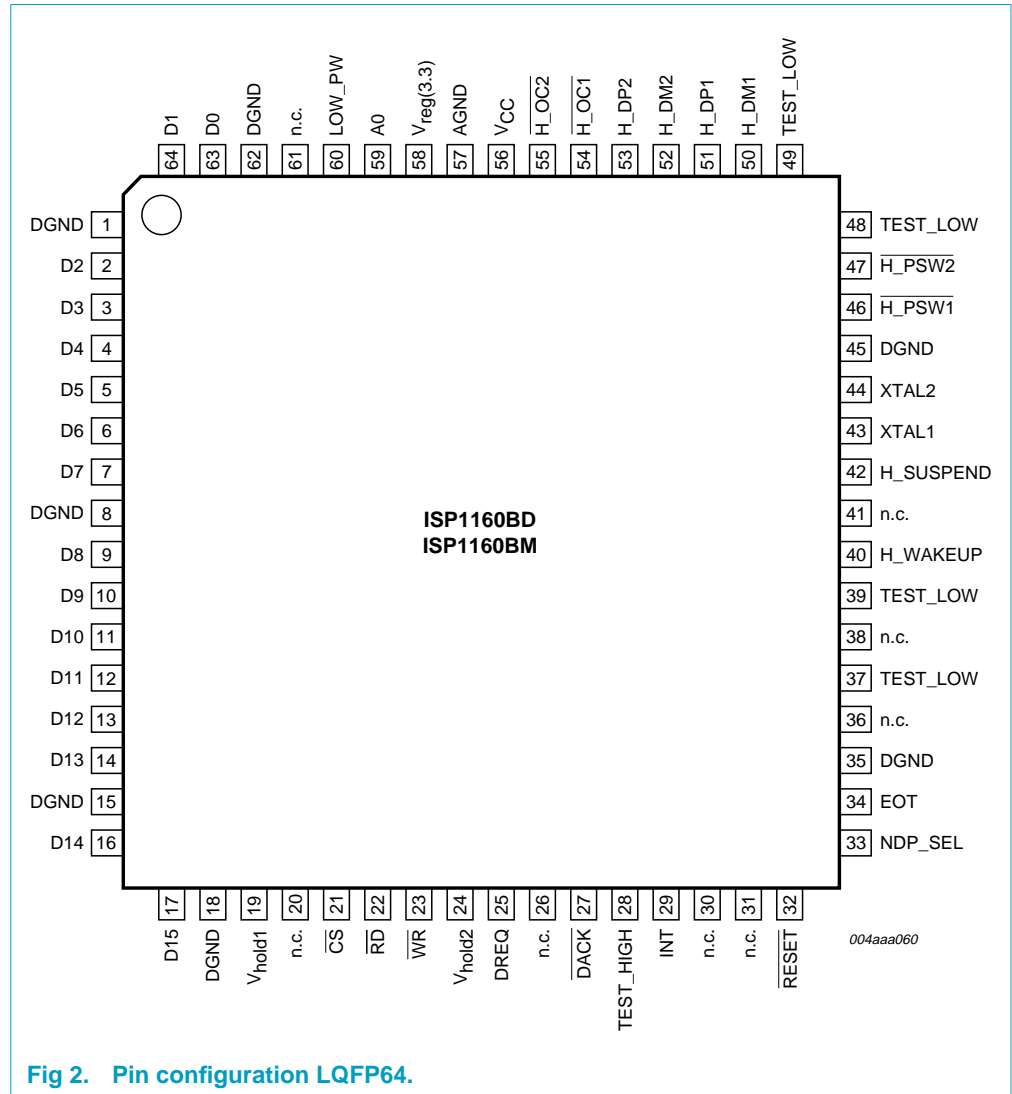
6. Pinning information

6.1 Pinning

Fig 2. Pin configuration LQFP64.

6.2 Pin description

Table 2:   Pin description for LQFP64

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

9397 750 10765

Product data   **Rev. 03 — 27 February 2003**   **4 of 89**

Table 2: **Pin description for LQFP64**…*continued*

Symbol[1]	Pin	Type	Description
D5	5	I/O	bit 5 of bidirectional data; slew-rate controlled; TTL input; three-state output
D6	6	I/O	bit 6 of bidirectional data; slew-rate controlled; TTL input; three-state output
D7	7	I/O	bit 7 of bidirectional data; slew-rate controlled; TTL input; three-state output
DGND	8	-	digital ground
D8	9	I/O	bit 8 of bidirectional data; slew-rate controlled; TTL input; three-state output
D9	10	I/O	bit 9 of bidirectional data; slew-rate controlled; TTL input; three-state output
D10	11	I/O	bit 10 of bidirectional data; slew-rate controlled; TTL input; three-state output
D11	12	I/O	bit 11 of bidirectional data; slew-rate controlled; TTL input; three-state output
D12	13	I/O	bit 12 of bidirectional data; slew-rate controlled; TTL input; three-state output
D13	14	I/O	bit 13 of bidirectional data; slew-rate controlled; TTL input; three-state output
DGND	15	-	digital ground
D14	16	I/O	bit 14 of bidirectional data; slew-rate controlled; TTL input; three-state output
D15	17	I/O	bit 15 of bidirectional data; slew-rate controlled; TTL input; three-state output
DGND	18	-	digital ground
V_{hold1}	19	-	voltage holding pin 1; this pin is internally connected to the $V_{reg(3.3)}$ and V_{hold2} pins. When pin V_{CC} is connected to 5 V, this pin will output 3.3 V and therefore, it must not be connected to 5 V. When pin V_{CC} is connected to 3.3 V, this pin can either be connected to 3.3 V or left unconnected. In **all** cases, this pin must be decoupled to DGND.
n.c.	20	-	not connected; leave this pin open
$\overline{CS}$	21	I	chip select input
$\overline{RD}$	22	I	read strobe input
$\overline{WR}$	23	I	write strobe input
V_{hold2}	24	-	voltage holding pin 2; this pin is internally connected to the $V_{reg(3.3)}$ and V_{hold1} pins. When pin V_{CC} is connected to 5 V, this pin will output 3.3 V and therefore, it must not be connected to 5 V. When pin V_{CC} is connected to 3.3 V, this pin can either be connected to 3.3 V or left unconnected. In **all** cases, this pin must be decoupled to DGND.
DREQ	25	O	HC's DMA request output (programmable polarity); signals to the DMA controller that the ISP1160 wants to start a DMA transfer; see Section 10.4.1
n.c.	26	-	not connected; leave this pin open

Table 2: **Pin description for LQFP64**…*continued*

Symbol[1]	Pin	Type	Description
$\overline{\text{DACK}}$	27	I	HC's DMA acknowledge input; when not in use, this pin must be connected to V_{CC} via an external 10 kΩ resistor
TEST_HIGH	28	-	this pin must be connected to V_{CC}
INT	29	O	HC's 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
$\overline{\text{RESET}}$	32	I	reset input (Schmitt trigger); a LOW level produces an asynchronous reset (internal pull-up resistor)
NDP_SEL	33	I	number of downstream ports: **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 (internal pull-up resistor)
EOT	34	I	DMA master device to inform the ISP1160 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
TEST_LOW	37	-	this pin must be connected to DGND
n.c.	38	O	not connected; leave this pin open
TEST_LOW	39	-	this pin must be connected to DGND
H_WAKEUP	40	I	HC's wake-up input; generates a remote wake-up from 'suspend' state (active HIGH); when not in use, this pin must be connected to DGND via an external 10 kΩ resistor (internal pull-up resistor)
n.c.	41	-	not connected; leave this pin open
H_SUSPEND	42	O	HC's suspend state indicator output; active HIGH
XTAL1	43	I	crystal input; connected directly to a 6 MHz crystal; when it is connected to an external clock oscillator, leave pin XTAL2 open
XTAL2	44	O	crystal output; connected directly to a 6 MHz crystal; when pin XTAL1 is connected to an external clock oscillator, leave this pin open
DGND	45	-	digital ground
H_PSW1	46	O	power switching control output for downstream port 1; open-drain output
H_PSW2	47	O	power switching control output for downstream port 2; open-drain output
TEST_LOW	48	-	this pin must be connected to DGND via an external 100 kΩ resistor
TEST_LOW	49	-	this pin must be connected to DGND via an external 100 kΩ resistor

Table 2:   Pin description for LQFP64…*continued*

Symbol[1]	Pin	Type	Description
H_DM1	50	AI/O	USB D− data line for HC's downstream port 1
H_DP1	51	AI/O	USB D+ data line for HC's downstream port 1
H_DM2	52	AI/O	USB D− data line for HC's downstream port 2; when not in use, leave this pin open
H_DP2	53	AI/O	USB D+ data line for HC's downstream port 2; when not in use, leave this pin open
$\overline{\text{H\_OC1}}$	54	I	overcurrent sensing input for HC's downstream port 1
$\overline{\text{H\_OC2}}$	55	I	overcurrent sensing input for HC's downstream port 2
V_{CC}	56	-	digital power supply voltage input (3.0 to 3.6 V or 4.75 to 5.25 V). This pin has been connected to the internal 3.3 V regulator input. When connected to 5 V, the internal regulator will output 3.3 V to pins $V_{reg(3.3)}$, V_{hold1} and V_{hold2}. When connected to 3.3 V, it will bypass the internal regulator.
AGND	57	-	analog ground
$V_{reg(3.3)}$	58	-	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, this pin should also be connected to 3.3 V.
A0	59	I	address input; selects command (A0 = 1) or data (A0 = 0)
LOW_PW	60	I	if low-current consumption (range of μs) is needed during suspend, connect this pin to address A1; otherwise, connect 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; three-state output
D1	64	I/O	bit 1 of bidirectional data; slew-rate controlled; TTL input; three-state output

[1]   Symbol names with an overscore (e.g. $\overline{\text{NAME}}$) represent active LOW signals.

9397 750 10765

Product data   **Rev. 03 — 27 February 2003**   **7 of 89**

7. Functional description

7.1 PLL clock multiplier

A 6 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 over-sampling principle. It is able to track jitter and frequency drift as specified in *Universal Serial Bus Specification Rev. 2.0.*

7.3 Analog transceivers

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 Philips Serial Interface Engine (SIE)

The Philips 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/serial conversion, bit (de)stuffing, CRC checking/generation, Packet IDentifier (PID) verification/generation, address recognition and handshake evaluation/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 ISP1160 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 ISP1160 occupies only two I/O ports or two memory locations of a microprocessor. External microprocessors can read or write the ISP1160's internal control registers and FIFO buffer RAM through the Programmed I/O (PIO) operating mode. Figure 3 shows the Programmed I/O interface between a microprocessor and the ISP1160.
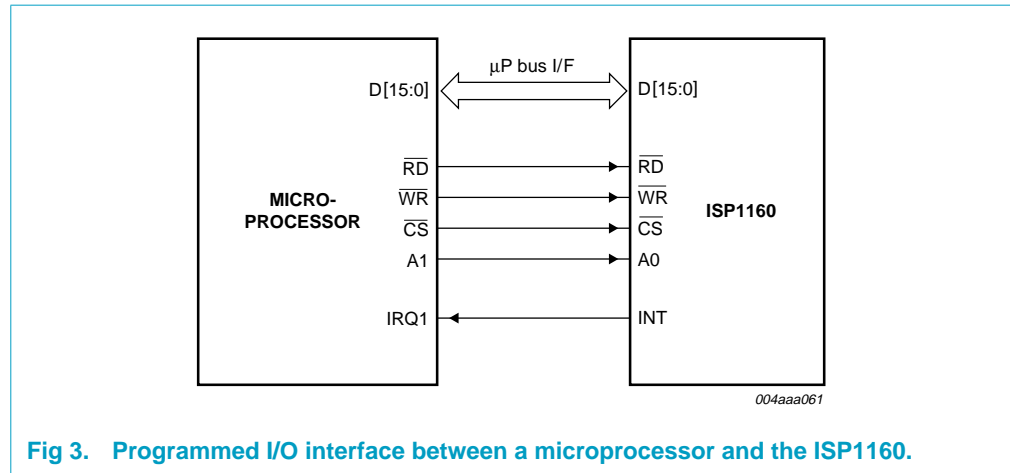
9397 750 10765

Product data   **Rev. 03 — 27 February 2003**   **8 of 89**

Fig 3.   Programmed I/O interface between a microprocessor and the ISP1160.

8.2 DMA mode

The ISP1160 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 microprocessor's system memory and the ISP1160's internal FIFO buffer RAM. Note: the DMA operation must be controlled by the external microprocessor system's DMA controller (Master). Figure 4 shows the DMA interface between a microprocessor system and the ISP1160. The ISP1160 provides a DMA channel controlled by DREQ for $\overline{DACK}$ signals for the DMA transfer between a microprocessor's system memory and the ISP1160 HC's 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 ISP1160 provides an internal EOT signal to terminate the DMA transfer as well. Setting the HcDMAConfiguration register (21H - Read, A1H - Write) enables the ISP1160's HC internal DMA counter for the DMA transfer. When the DMA counter reaches the value set in the HcTransferCounter (22H - Read, A2H - Write) register to be used as the byte count of the DMA transfer, the internal EOT signal will be generated to terminate the DMA transfer.
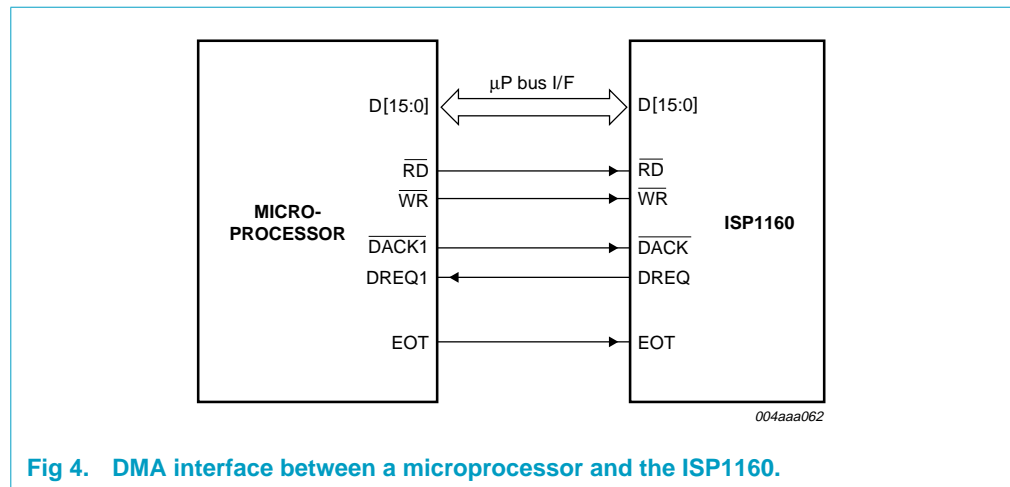

Fig 4.   DMA interface between a microprocessor and the ISP1160.

9397 750 10765

Product data   **Rev. 03 — 27 February 2003**   **9 of 89**

8.3 Microprocessor read/write the ISP1160's internal control registers by PIO mode

8.3.1 I/O port addressing

Table 3 shows the ISP1160's I/O port addressing. Complete decoding of the I/O port address should include the chip select signal $\overline{CS}$ and the address line A0. However, the direction of access of I/O ports is controlled by the $\overline{RD}$ and $\overline{WR}$ signals. When $\overline{RD}$ is LOW, the microprocessor reads data from the ISP1160's data port. When $\overline{WR}$ 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	$\overline{CS}$	[A0] (Bin)	Access	Data bus width (bits)	Description
0	0	0	R/W	16	HC data port
1	0	1	W	16	HC command port

Figure 5 illustrates how an external microprocessor accesses the ISP1160's 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.
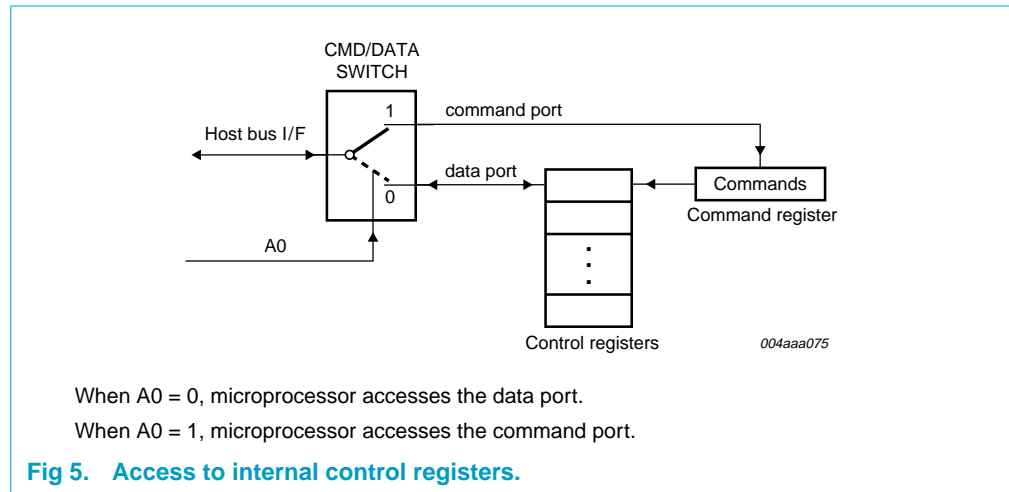
8.3.2 Register access phases

The ISP1160's 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 ISP1160 to the next register to be accessed. A command is 8 bits long. On a microprocessor's 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 ISP1160. 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 a chip's ID, which is saved in the HC's HcChipID register (27H, read only) where its command code is 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 ISP1160's HC command port

2. The microprocessor reads the data word of the chip's ID (6110H) from the ISP1160's HC data port.
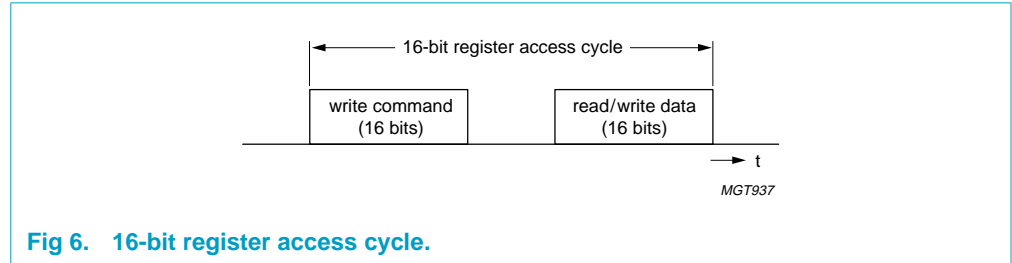

Fig 6.   16-bit register access cycle.

Most of the ISP1160's internal control registers are 16-bit wide. Some of the internal control registers, however, are 32-bit wide. Figure 7 shows how the ISP1160's 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 should first read or write the lower 16-bit data, followed by the upper 16-bit data.
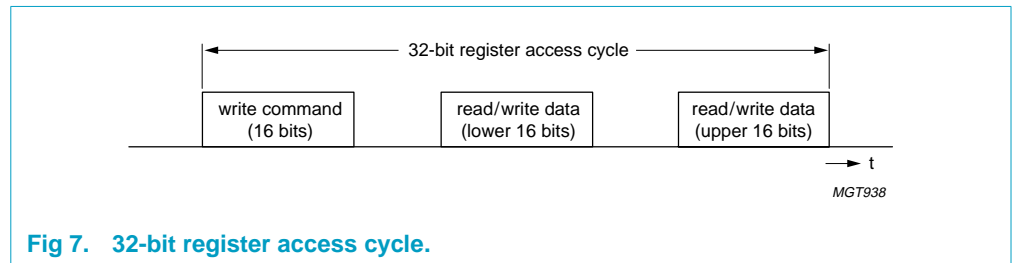

Fig 7.   32-bit register access cycle.

To further describe the complete access cycles of ISP1160's internal control registers, the status of some pin signals of the microprocessor bus interface is shown in Figure 8.

Fig 8.   Accessing the ISP1160 HC control registers.

8.4 Microprocessor read/write the ISP1160's internal FIFO buffer RAM by PIO mode

Since the ISP1160's internal memory is structured as a FIFO buffer RAM, the FIFO buffer RAM is mapped to dedicated register fields. Therefore, accessing the ISP1160's internal FIFO buffer RAM is just like accessing the internal control registers in multiple data phases.
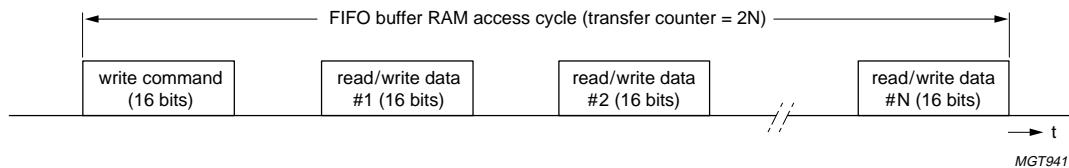

Fig 9. The ISP1160's internal FIFO buffer RAM access cycle.

Figure 9 shows a complete access cycle of the ISP1160's internal FIFO buffer RAM. For a write cycle, the microprocessor first writes the FIFO buffer RAM's command code to the command port, and then writes the data words one by one to the data port until half of the transfer's byte count is reached. The HcTransferCounter register (22H - Read, A2H - Write) is used to specify the byte count of a FIFO buffer RAM's 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 Microprocessor read/write the ISP1160's internal FIFO buffer RAM by DMA mode

The DMA interface between a microprocessor and the ISP1160 is shown in Figure 4.

When doing a DMA transfer, at the beginning of every burst the ISP1160 outputs a DMA request to the microprocessor via pin DREQ. After receiving this signal, the microprocessor will reply with a DMA acknowledge to the ISP1160 via pin $\overline{\text{DACK}}$, and at the same time, do the DMA transfer through the data bus. In the DMA mode, the microprocessor must still issue a $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal to the ISP1160's $\overline{\text{RD}}$ or $\overline{\text{WR}}$ pin. The ISP1160 will repeat the DMA cycles until it receives an EOT signal to terminate the DMA transfer.

The ISP1160 supports both external and internal EOT signals. The external EOT signal is received as input from the ISP1160's EOT pin: it generally comes from the external microprocessor. The internal EOT signal is generated by the ISP1160 internally.

To select either, set the DMA configuration registers. For example, for the HC, setting bit 2 of the HcDMAConfiguration register (21H - Read, A1H - Write) to logic 1 will enable the DMA counter for DMA transfer. When the DMA counter reaches the value of HcTransferCounter register, the internal EOT signal will be generated to terminate the DMA transfer.

The ISP1160 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.
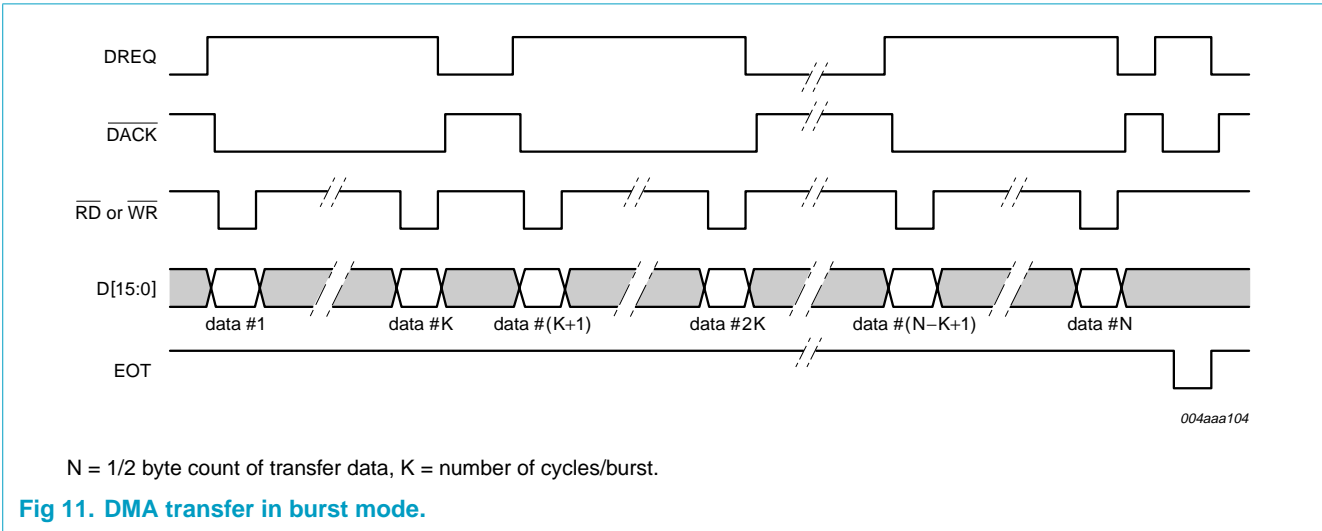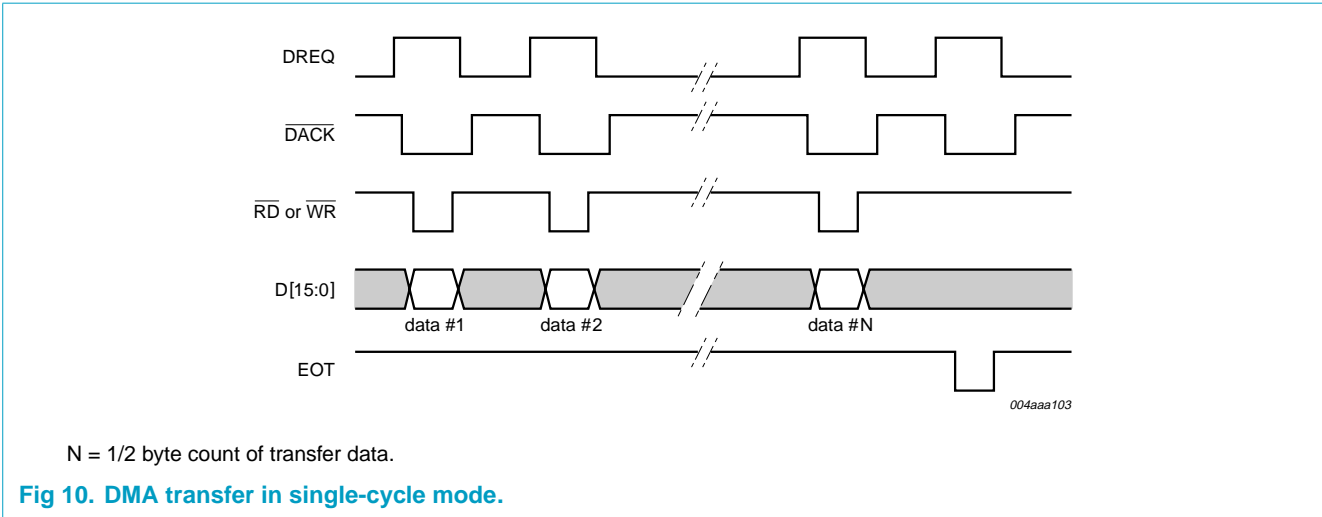
In both figures, the DMA transfer is configured such that DREQ is active HIGH and $\overline{\text{DACK}}$ is active LOW.

8.6 Interrupts

The ISP1160 has an interrupt request pin INT.

8.6.1 Pin configuration

The interrupt output signals have four configuration modes:

- Level trigger, active LOW
- Level trigger, active HIGH
- Edge trigger, active LOW
- Edge trigger, active HIGH.

Figure 12 shows these four interrupt configuration modes. They are programmable through register settings, which are also used to disable or enable the signals.
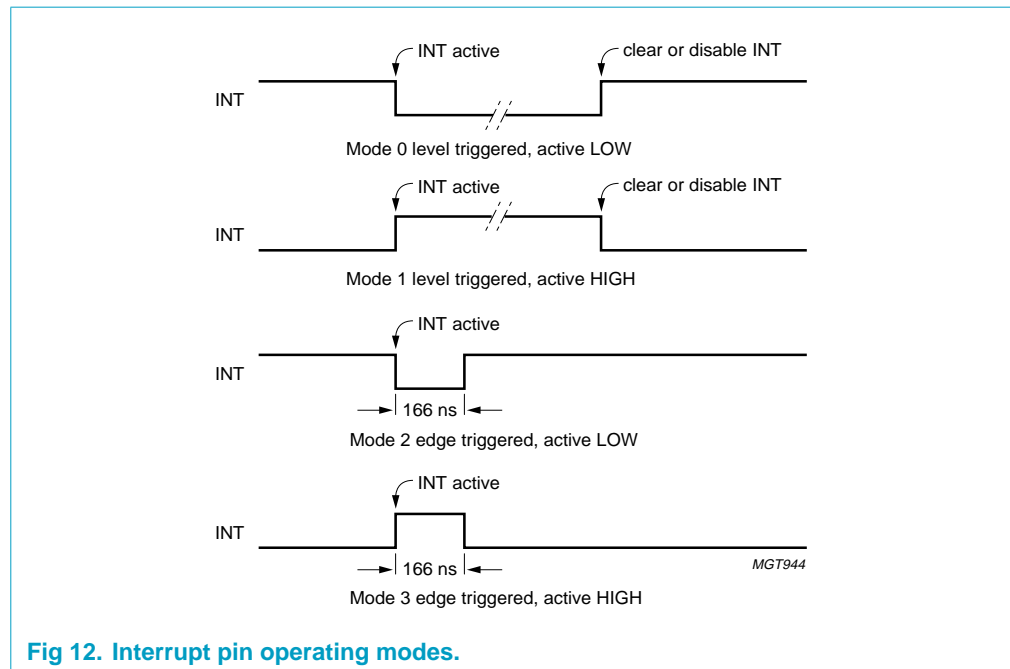
Fig 12. Interrupt pin operating modes.

8.6.2 Interrupt output pin (INT)

To program the four configuration modes of the HC's interrupt output signal (INT), set bits 1 and 2 of the HcHardwareConfiguration register (20H - Read, A0H - Write). Bit 0 is used as the master enable setting for pin INT.

INT has many interrupt events. The relationship between pin INT and its interrupt events is shown as in Figure 13.
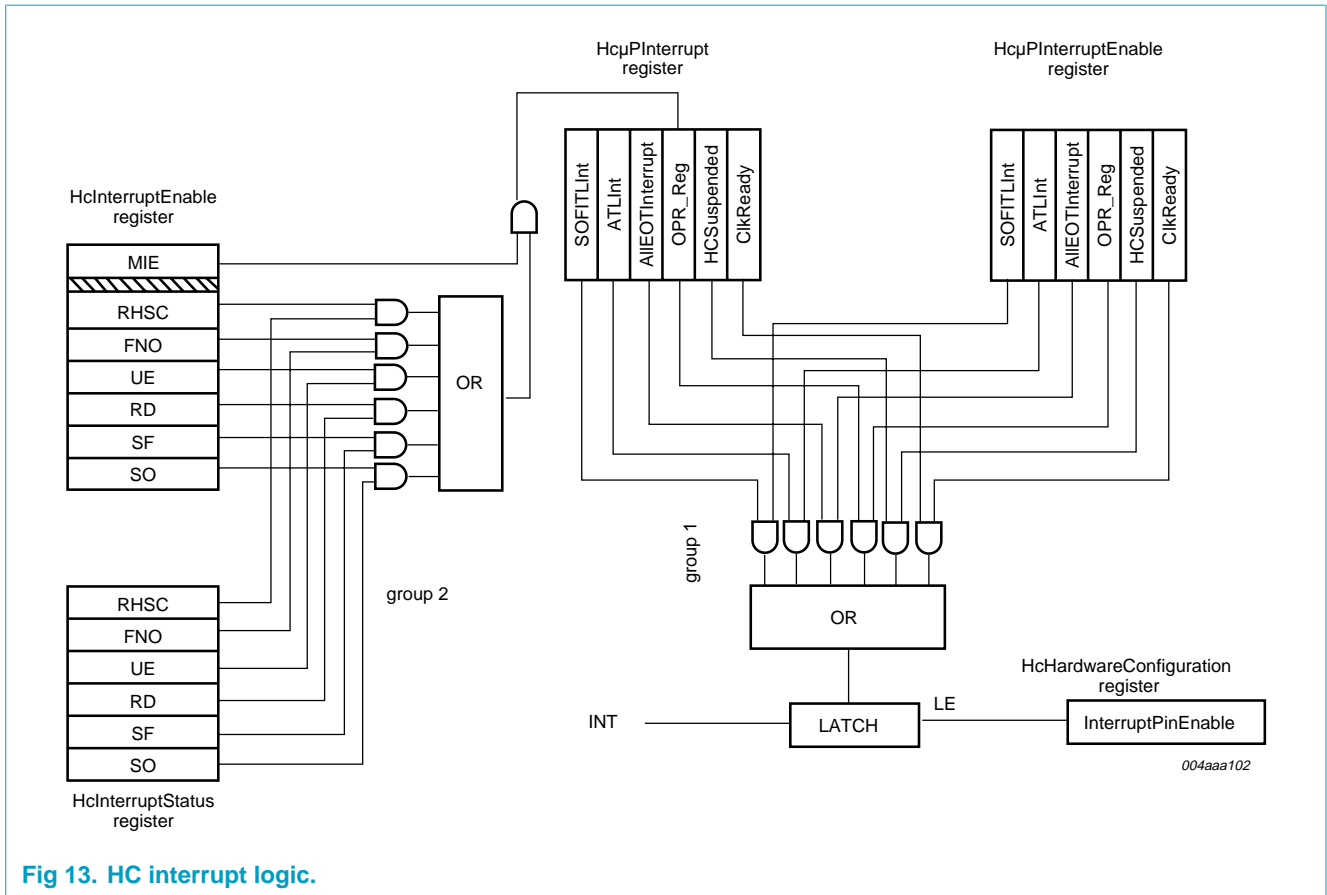
Fig 13. HC interrupt logic.

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 ANDed 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 ISP1160 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 reenable 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 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 reenable 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. Philips slave Host Controller (HC)

9.1 HC's four USB states

The ISP1160's USB HC has four USB states – USB Operational, USB Reset, USB Suspend and USB Resume – that define the HC's USB signalling and bus states responsibilities. The signals are visible to the HC (software) Driver via the ISP1160 USB HC's control registers.

9397 750 10765

Product data Rev. 03 — 27 February 2003 16 of 89
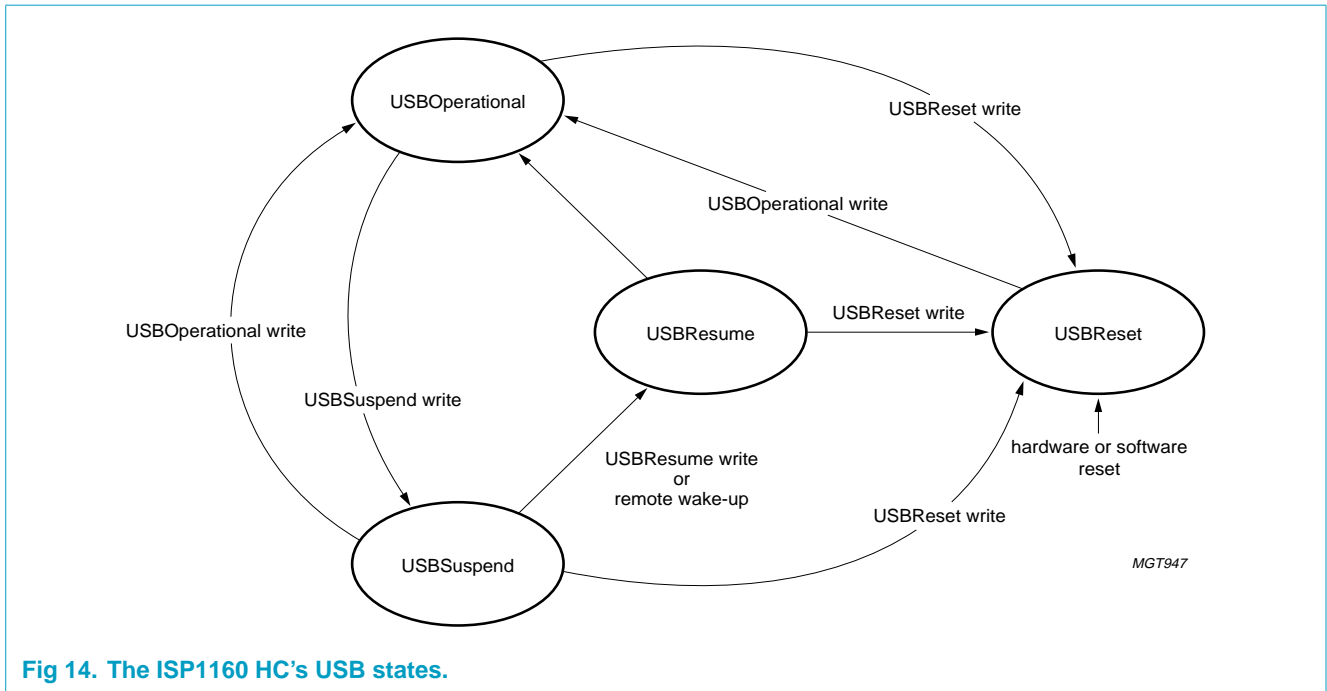
Fig 14. The ISP1160 HC's USB states.

USB states are reflected in the HostControllerFunctionalState field of the HcControl register (01H - Read, 81H - Write), which is located at bits 7 and 6 of the register.

The HC Driver (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 - Read, 82H - Write).

9.2 Generating USB traffic

USB traffic can be generated only when the ISP1160 USB HC is under the USB Operational 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 ISP1160 USB HC's register usage.

MGT948
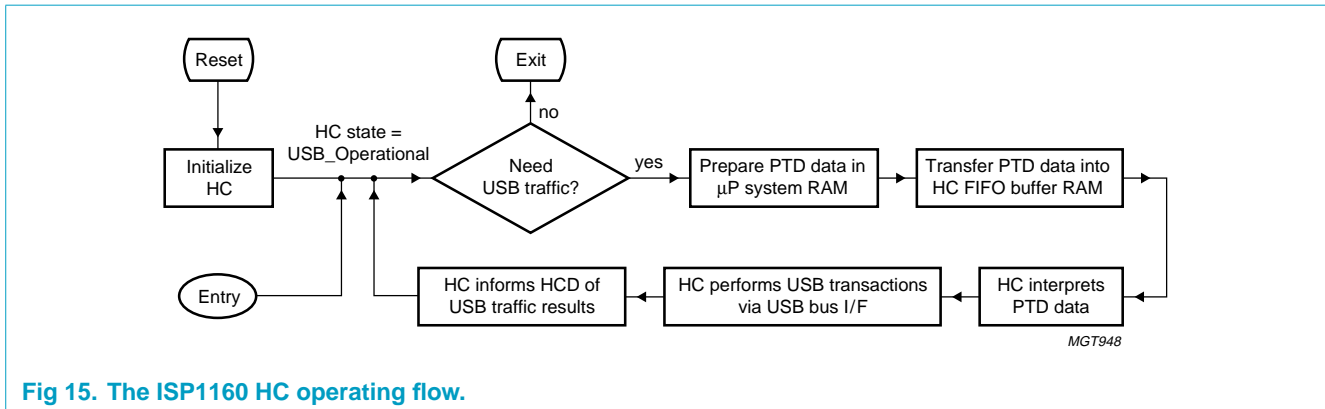
Fig 15. The ISP1160 HC operating flow.

Description of the flow diagram:

- **Reset**
  This includes hardware reset by pin $\overline{\text{RESET}}$ and software reset by the HcSoftwareReset command (A9H). The reset function will clear all the HC's internal control registers to their reset status. After reset, the HCD must initialize the ISP1160 USB HC by setting some registers.

- **Initialize HC**
  It includes:

  - Setting the physical size for the HC's internal FIFO buffer RAM by setting the HcITLBufferLength register (2AH - Read, AAH - Write) and the HcATLBufferLength register (2BH - Read, ABH - Write)

  - Setting the HcHardwareConfiguration register according to requirements

  - Clearing interrupt events, if required

  - Enabling interrupt events, if required

  - Setting the HcFmInterval register (0DH - Read, 8DH - Write)

  - Setting the HC's Root Hub registers

  - Setting the HcControl register to move the HC into USB Operational state

  See also Section 9.5.

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

- **Need USB traffic**

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

  - Connecting to or disconnecting from downstream ports

  - Issuing the Resume signal to the HC

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

- **Prepare PTD data in microprocessor's system RAM**

  The communication channel between the HCD and the ISP1160's USB HC is in the form of Philips 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 microprocessor's system RAM. For the ISP1160's USB HC, it is the ISP1160's internal FIFO buffer RAM.

  The HCD prepares PTD data in the microprocessor's system RAM for transfer to the ISP1160's HC internal FIFO buffer RAM.

- **Transfer PTD data into HC's FIFO buffer RAM**
  When PTD data is ready in the microprocessor's system RAM, the HCD must transfer the PTD data from the microprocessor's system RAM into the ISP1160's internal FIFO buffer RAM.

- **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.

- **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.

- **HC informs HCD the USB traffic results**
  The USB transaction status and the feedback from the specified USB device endpoint will be put back into the ISP1160's 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 Philips Transfer Descriptor (PTD) data structure provides a communication channel between the HCD and the ISP1160's USB HC. 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.
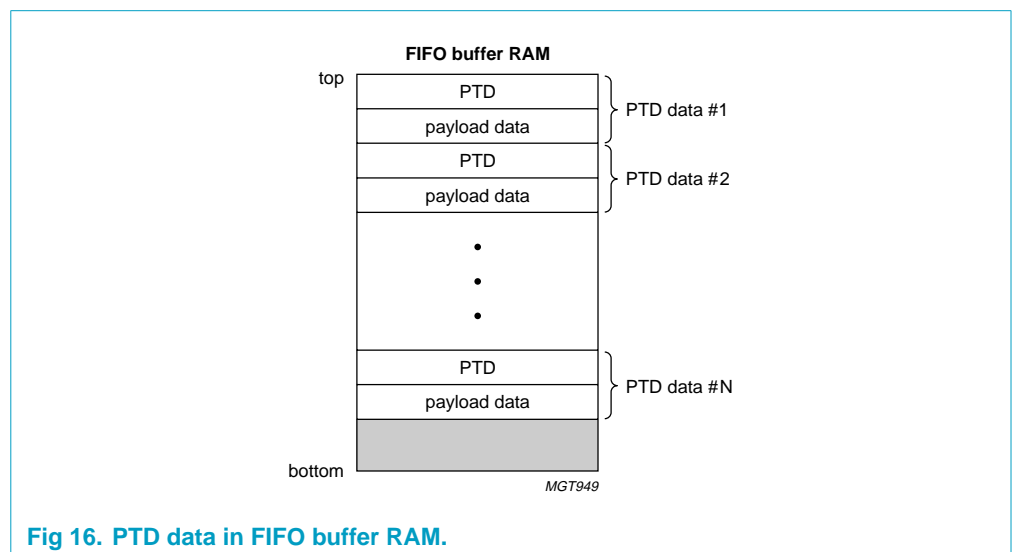
Fig 16. PTD data in FIFO buffer RAM.

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 firmware, 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. PTD data is DWORD (double-word or 4-byte) 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 payload data's actual size. A PTD is an 8-byte data structure that is very important for HCD programming.

Table 4:   Philips 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		B5_5	reserved	DirectionPID[1:0]		TotalBytes[9:8]	
Byte 6	Format	FunctionAddress[6:0]						
Byte 7	reserved							

Table 5:   Philips Transfer Descriptor (PTD): bit description

Symbol	Access	Description
ActualBytes[9:0]	R/W	Contains the number of bytes that were transferred for this PTD.

Table 5:   Philips Transfer Descriptor (PTD): bit description…*continued*

Symbol	Access				Description
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 MaximumPacketSize field of ED) 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 should 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 (PTD)	R				Last PTD of a list (ITL or ATL). A logic 1 indicates that the PTD is the last PTD.
(Low) Speed	R				Speed of the endpoint: **S = 0 —** full speed **S = 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.

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **21 of 89**

Table 5:   Philips 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		This bit is logic 0 at power-on reset. When this feature is not used, software used in the ISP1160 is the same for ISP1161 and ISP1161A. When this bit is set to logic 1 in this PTD for interrupt endpoint transfer, only 1 PTD USB transaction will be sent out in 1 ms.
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 HC's 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 HC's internal FIFO buffer RAM has a physical size of 4 kbytes. 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 (ISO) 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 another ITL buffer at the same time. This architecture can improve the ISO transfer performance.

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

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

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.

- ATL buffer length = 0H, ITL buffer length = 800H.
  This will use the internal FIFO buffer RAM for only ISO transfers.
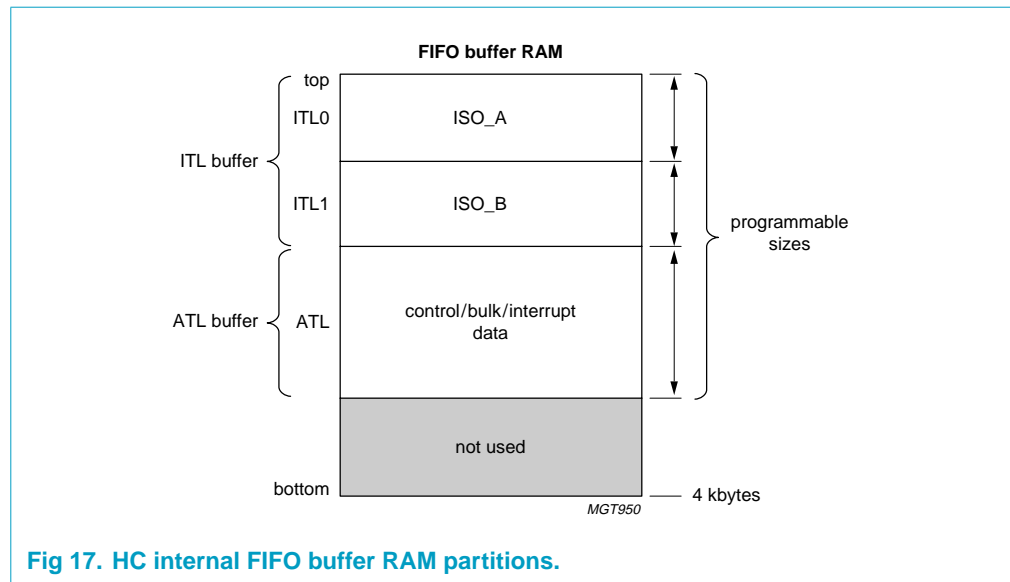
Fig 17. HC internal FIFO buffer RAM partitions.

The actual requirement for the buffer RAM need not reach 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 bytes (20 ISO packets of 64 bytes). Total RAM size needed for this is $20 \times 8 + 1280 = 1440$ bytes.

- Maximum number of packets for different endpoints sent during one USB frame is 150 (150 ISO packets of 1 byte). Total RAM size needed is:

  $150 \times 8 + 150 \times 1 = 1350$ bytes.

- The Ping buffer RAM (ITL0) and the Pong buffer RAM (ITL1) have a maximum size of 2 kbytes 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 - Read only) indicates that the buffer is empty, the microprocessor can write data into the buffer. When the HcBufferStatus register indicates that the buffer is full, that is, data is ready on the buffer, 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 Mbyte/s. In the DMA operation, the single-cycle or multi-cycle burst modes are supported. For the multi-cycle burst mode, 1, 4 or 8 cycles per burst is supported for the ISP1160.

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, some 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 should be put into ITL buffer RAM, the ISP1160 takes care of the Ping-Pong action for the ITL buffer RAM access.
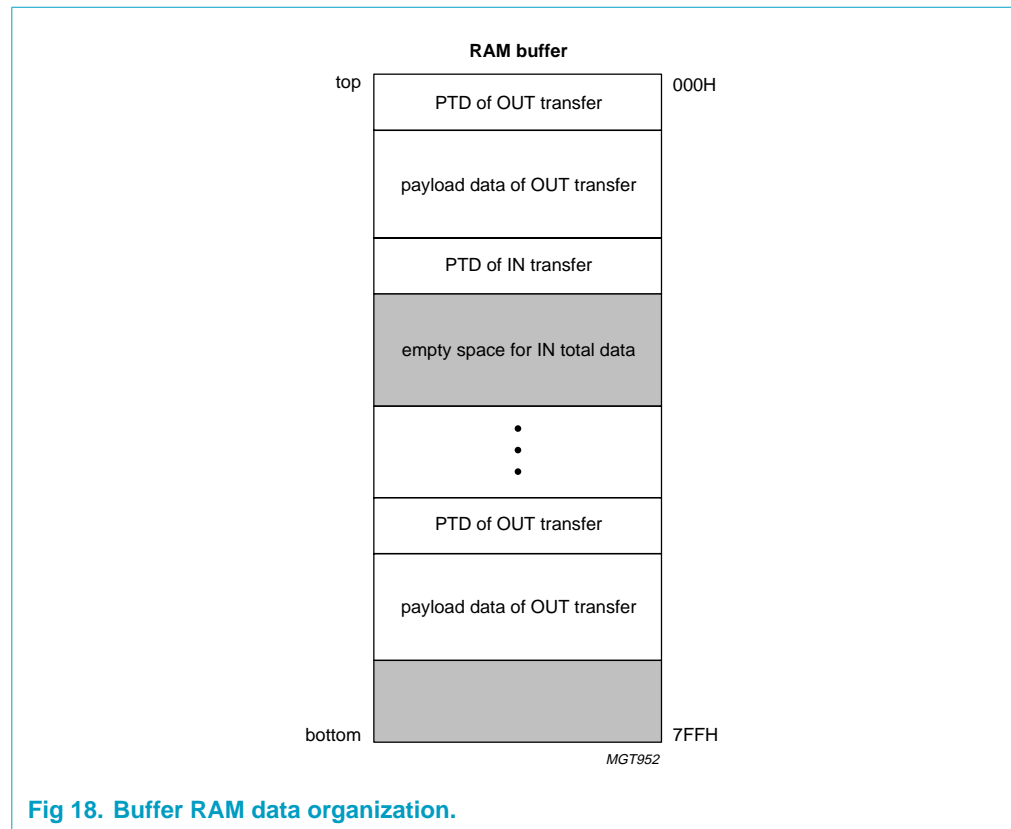
Fig 18. Buffer RAM data organization.

The PTD data (PTD header and its payload data) is a structure of DWORD alignment. This means that the memory address is organized in steps of 4 bytes. 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 bytes 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.
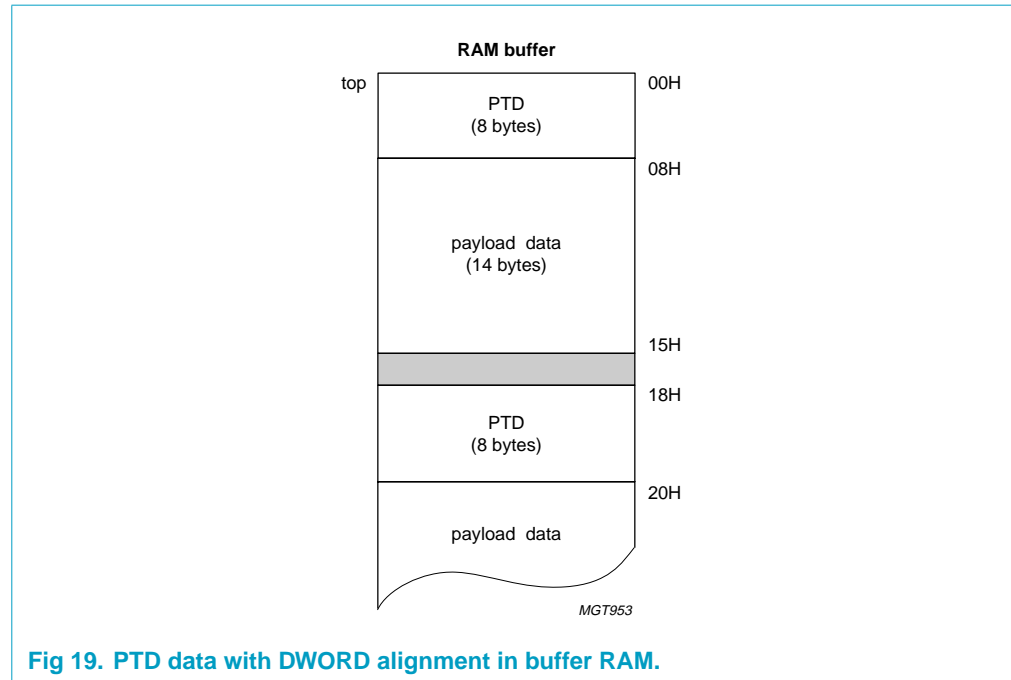
Fig 19. PTD data with DWORD alignment in buffer RAM.

9.4.3 Operation & C Program Example

Figure 20 shows the block diagram for internal FIFO buffer RAM operations by the PIO mode. The ISP1160 provides one register as the access port for each buffer RAM. For the ITL buffer RAM, the access port is the ITLBufferPort register (40H - Read, C0H - Write). For the ATL buffer RAM, the access port is the ATLBufferPort register (41H - Read, C1H - Write). The buffer RAM is an array of bytes (8 bits) 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 higher byte in the access port register corresponds to the other 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.

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **25 of 89**

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 80 (decimal) that will be set into the HcTransferCounter register as 50H. The data consists of four types of PTD data:

1. The first PTD header (IN) is 8 bytes, followed by 16 bytes of space reserved for its payload data;

2. The second PTD header (IN) is also 8 bytes, followed by 8 bytes of space reserved for its payload data;

3. The third PTD header (OUT) is 8 bytes, followed by 16 bytes of payload data with values beginning from 0H to FH incrementing by 1;

4. The fourth PTD header (OUT) is also 8 bytes, followed by 8 bytes of payload data with values beginning from 0H to EH incrementing by 2.

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 that MaxPacketSize.

Table 6 shows the results after running this program.

```
//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[0x28]=
{

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,

                 0x0800,0x1808,0x0408,0x0005, //PTD header for OUT token #2

                 0x0200,0x0604,0x0a08,0x0e0c //Payload data for OUT token #2
                 };
                 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,0x50);

                 wCount = 0x28; //Get word count outport
                 (HcCmdPort,0x00c1); //Command for ATL buffer write

                 //write 80 (0x50) 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);
                  }
```

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **27 of 89**

Table 6: **Run results of the C program example**

Observed items	HC not initialized and not under Operational state	HC initialized and under Operational 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

However, if communication with a peripheral USB device is desired, the device should be connected to the downstream port and pass enumeration.
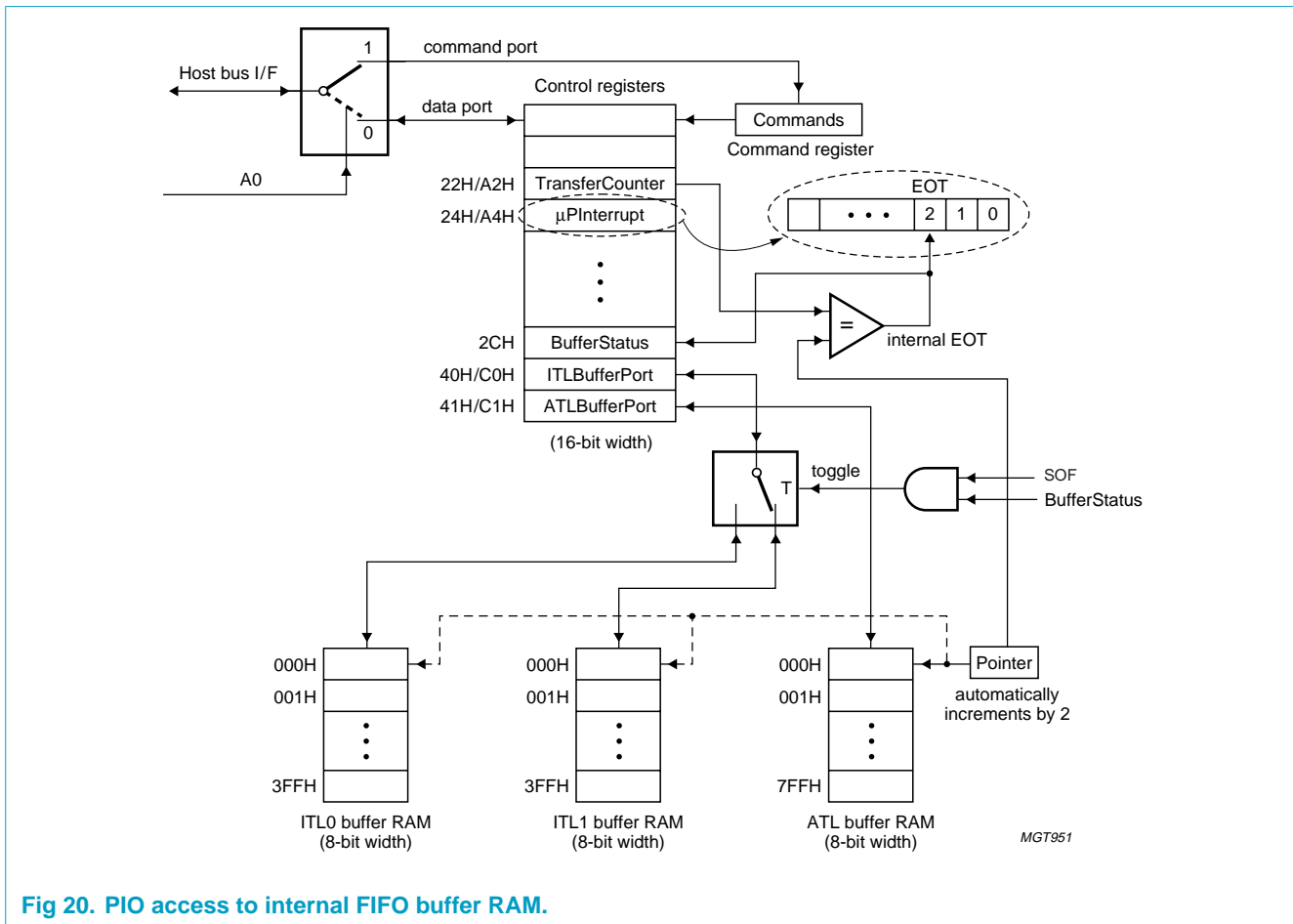
Fig 20. PIO access to internal FIFO buffer RAM.

9.5 HC's 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 - Read, 8DH - Write) and the HcLSThreshold register (11H - Read, 91H - 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 - Read, AAH - Write) is programmed.

If a USB frame contains both ISO and AT 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 by the microprocessor, after which the interrupt is cleared by the microprocessor.

Next the programmable AT Interrupt (ATLint is set in the HcμPInterrupt register) is issued, which triggers reading and writing of the ITL 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, an active bit and a 'TotalBytes of transfer' field are introduced (see Table 5). The active bit is cleared only if all data of the Philips Transfer Descriptor (PTD) are transferred or if a transaction at that endpoint contained a fatal error. If all PTD of the ATL are serviced once and the frame is not over yet, the HC starts looking for a PTD with the active bit 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 Host Controller Driver has also to take care of the BufferStatus register (2CH, Read only) for the ITL buffer RAM operations. After the Host Controller Driver writes ISO data into ITL buffer RAM, the ITL0BufferFull or ITL1BufferFull bit (depends if 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 Host Controller Driver can clear buffer status bits by a read of the ITL buffer RAM, and 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, in the first frame, for the HCD doing a write of ISO-A data into the ITL0 buffer. This will cause the BufferStatus register to show that the ITL0 buffer is full by setting the ITL0BufferFull bit to logic 1. At this stage the Host Controller Driver cannot write ISO data into the ITL0 buffer RAM again.

In the second frame, the Host Controller 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 ITL0 buffer at least two bytes (one word) 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 bit will be cleared automatically, but the ITL0BufferFull bit remains at logic 1 and the ITL0BufferFull bit will be unable to be cleared. This condition will disable the HCD from writing ISO data into the ITL0

buffer again and ISO processing be unable to be carried on. This also applies to the ITL1 buffer because the ITL0 and ITL1 are Ping-Pong structured buffers. To recover from this state the power-on reset or software reset will have to be applied on the HC.

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.
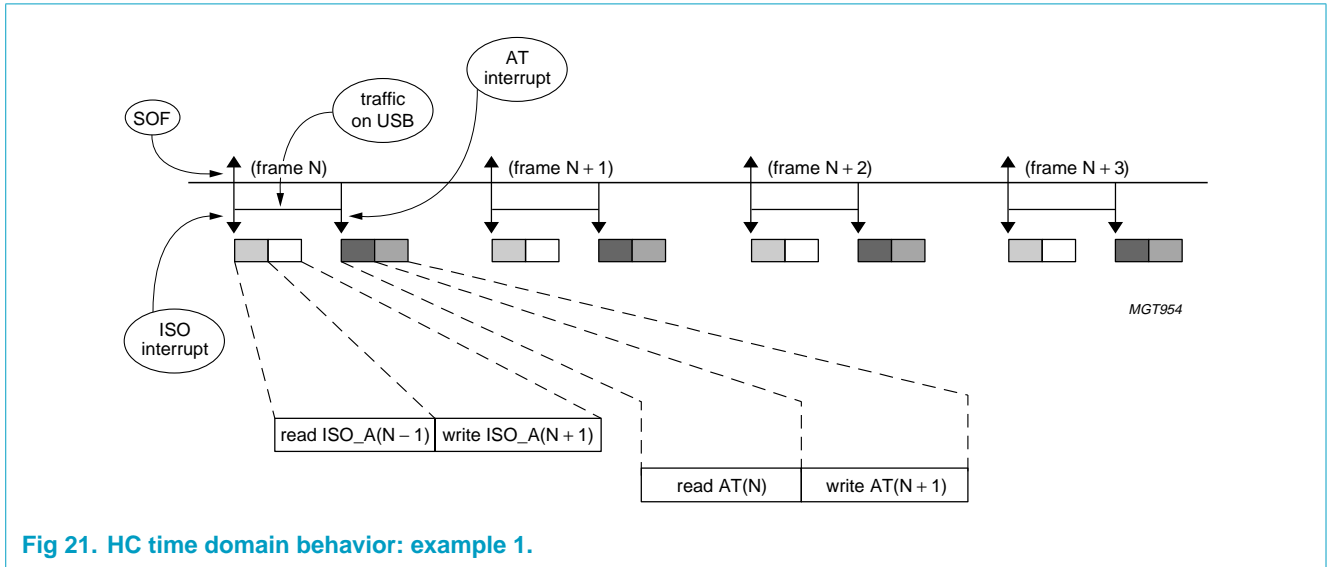
Fig 21. HC time domain behavior: example 1.

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.
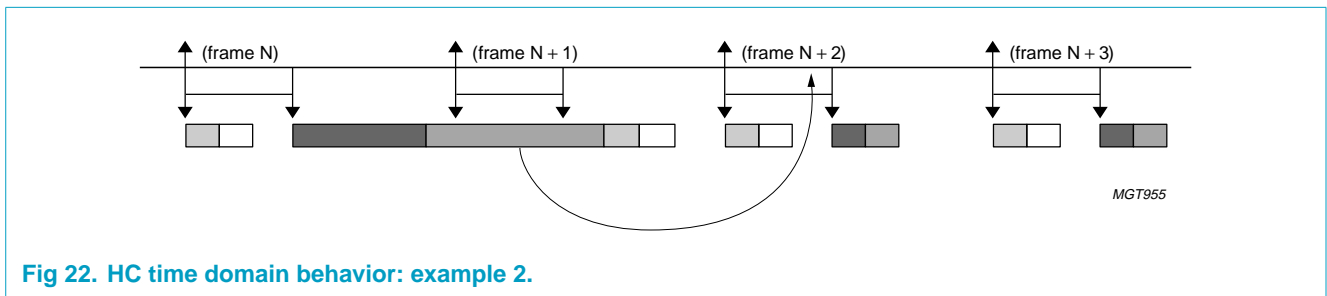
Fig 22. HC time domain behavior: example 2.

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 the exact timing data is corrupted or not). The HC should not raise an AT interrupt in frame N + 1.
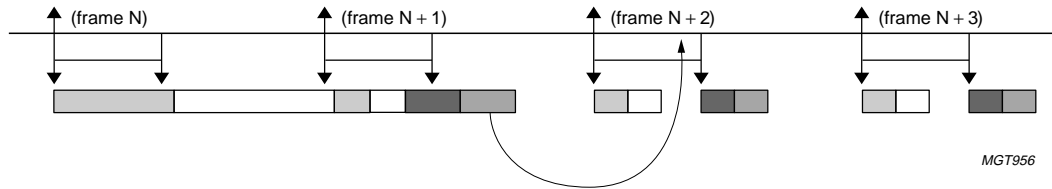
Fig 23. HC time domain behavior: example 3.

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 bytes. 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 ISP1160 for the two downstream ports: two resistors for each port. These resistors are software selectable by programming bit 12 of the HcHardwareConfiguration register (20H - Read, A0H - Write). When bit 12 is cleared to logic 0, it means that external 15 kΩ pull-down resistors should be used. Bit 12 is set to logic 1 that indicates the internal built in 15 kΩ pull-down resistors will be used instead of external ones. See Figure 24.

This feature is a cost-saving option. However, the power-on reset default value is logic 0. If you want to use the internal resistors, the HCD must check this bit status after every reset, because a reset action will clear this bit regardless of it being a hardware reset or a software reset.

Using either internal or external 15 kΩ resistors.

Fig 24. Use of 15 kΩ pull-down resistors on downstream ports.
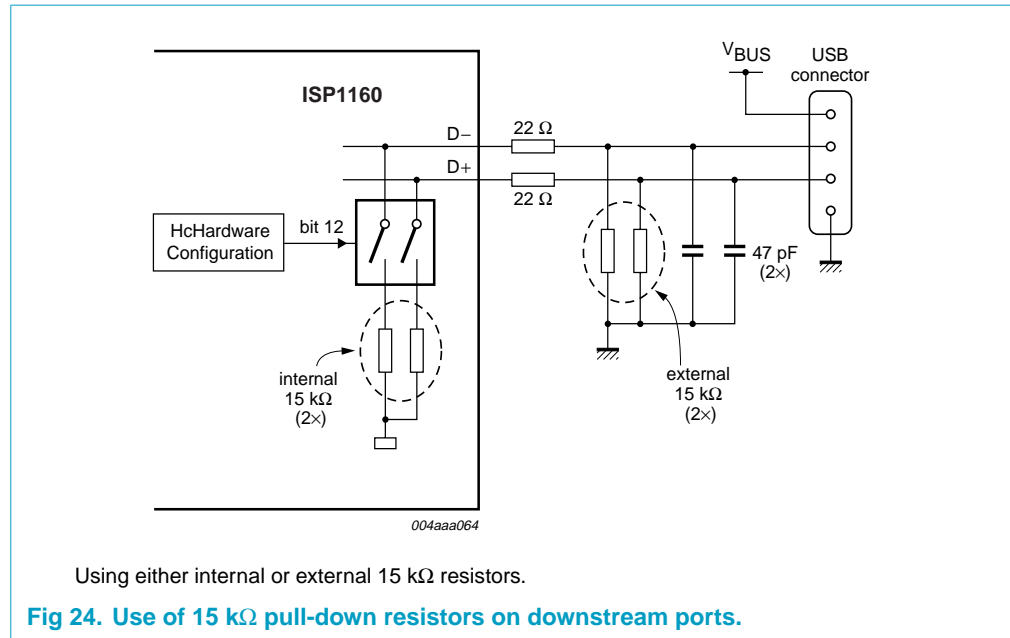
9.8 Overcurrent detection and power switching control

A downstream port provides 5 V power supply to V_{BUS}. The ISP1160 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. $\overline{H\_PSW1}$ and $\overline{H\_PSW2}$ are power switching control output pins (active LOW, open-drain) for downstream port 1 and 2, respectively. $\overline{H\_OC1}$ and $\overline{H\_OC2}$ are overcurrent detection input pins for downstream ports 1 and 2, respectively. Let $\overline{H\_PSWn}$ denote either $\overline{H\_PSW1}$ or $\overline{H\_PSW2}$ and $\overline{H\_OCn}$ denote either $\overline{H\_OC1}$ or $\overline{H\_OC2}$.

Figure 25 shows the ISP1160 downstream port power management scheme.

Fig 25. Downstream port power management scheme.
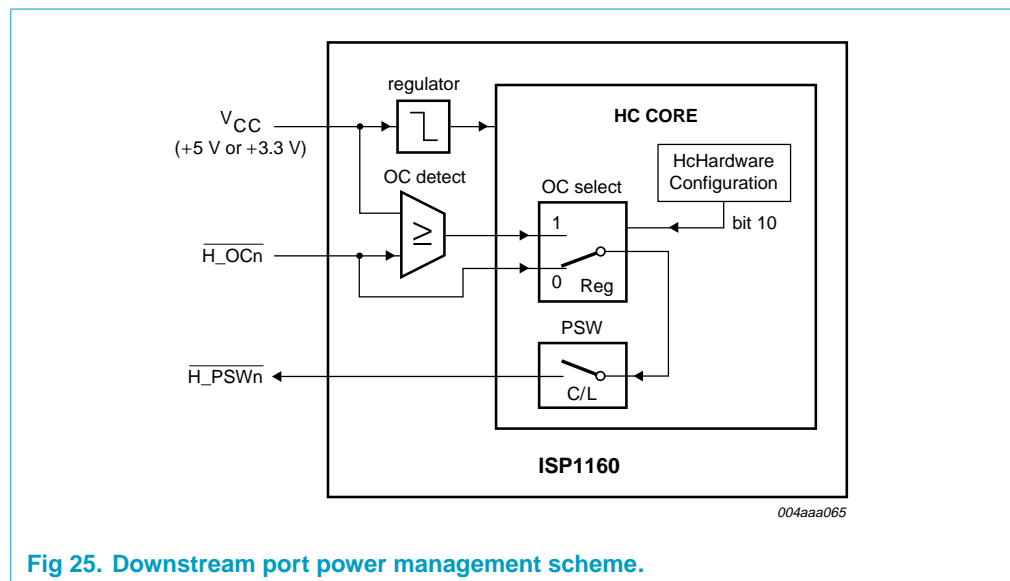
9.8.1 Using internal OC detection circuit

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, where H_DMn denotes either pin H_DM1 or H_DM2, while H_DPn denotes either pin H_DP1 or H_DP2. 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 $\overline{\text{H\_OCn}}$ detects an overcurrent status on a downstream port, $\overline{\text{H\_PSWn}}$ 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, $\overline{\text{H\_PSWn}}$ will output LOW, a logic 0 to turn on the 5 V power supply to the downstream port V_{BUS}.

In general applications, we can use a P-channel MOSFET as the power switch for V_{BUS}. Connect the 5 V power supply to the drain pole of the P-channel MOSFET, V_{BUS} to the source pole, and $\overline{\text{H\_PSWn}}$ to the gate pole. We call the voltage drop (ΔV) across the drain and source poles the overcurrent trip voltage. For the internal overcurrent detection circuit, a voltage comparator has been designed-in, with a nominal voltage threshold of 75 mV. Therefore, when the overcurrent trip voltage (ΔV) exceeds the voltage threshold, $\overline{\text{H\_PSWn}}$ will output a HIGH level, logic 1 to turn off the P-channel MOSFET. If the P-channel MOSFET has a R_{DSon} of 150 mΩ, the overcurrent threshold will be 500 mA. The selection of a P-channel MOSFET with a different R_{DSon} will result in a different overcurrent threshold.

9397 750 10765

Product data

Rev. 03 — 27 February 2003

33 of 89
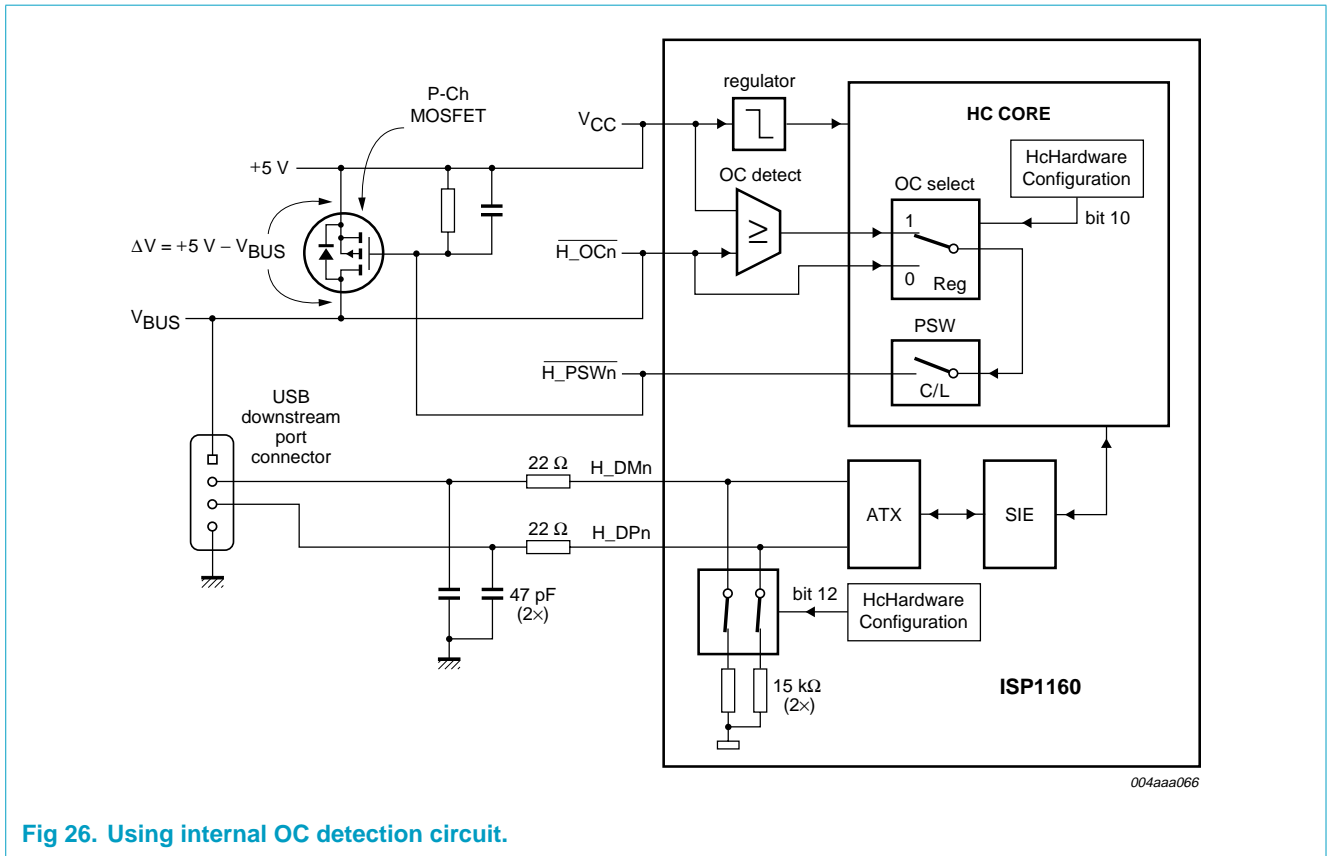
Fig 26. Using internal OC detection circuit.

9.8.2 Using external OC detection circuit

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

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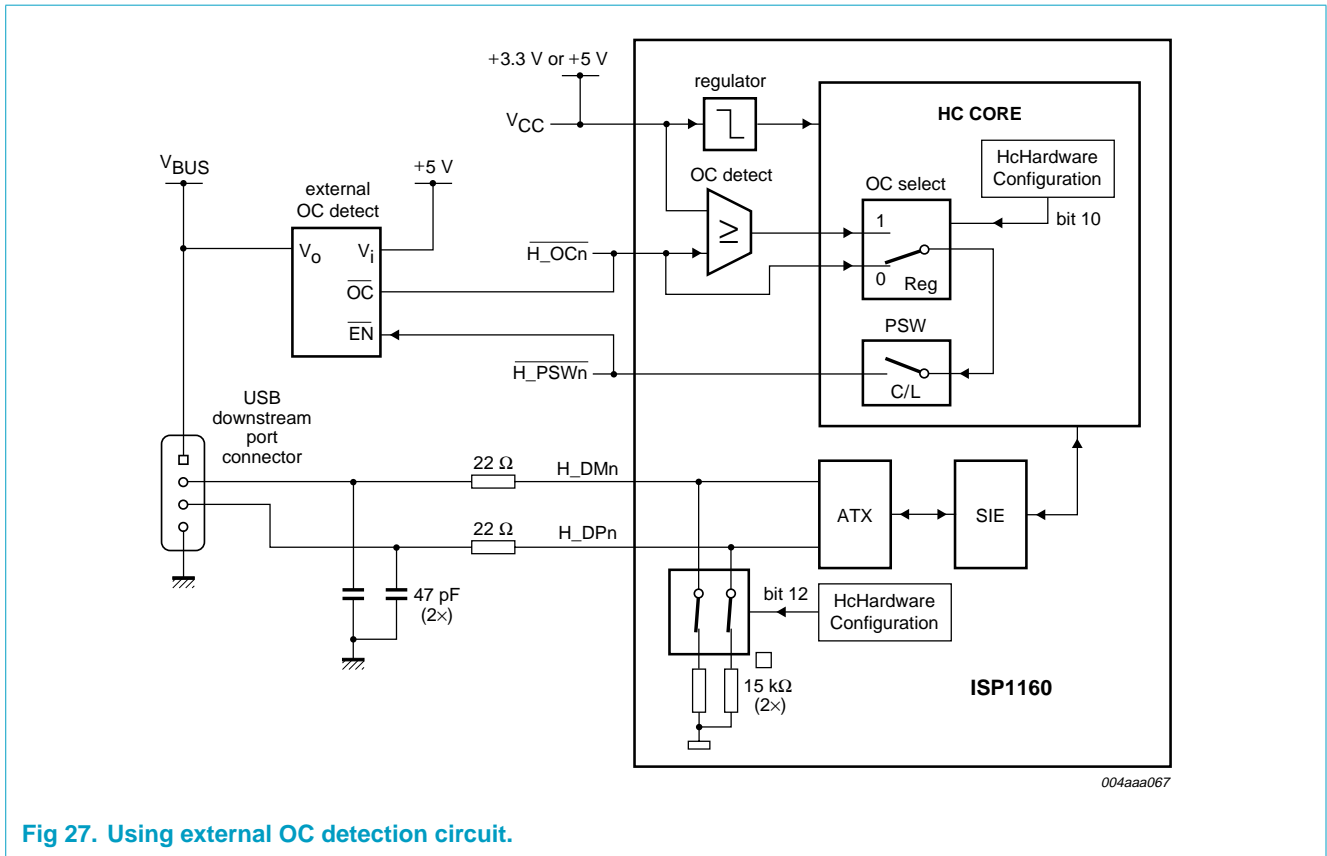
Fig 27. Using external OC detection circuit.

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 - Read, 81H - Write). See Figure 14 for the HC's flow of USB states changes.

Fig 28. The ISP1160 suspend and resume clock scheme.

9397 750 10765

Product data

Rev. 03 — 27 February 2003

35 of 89
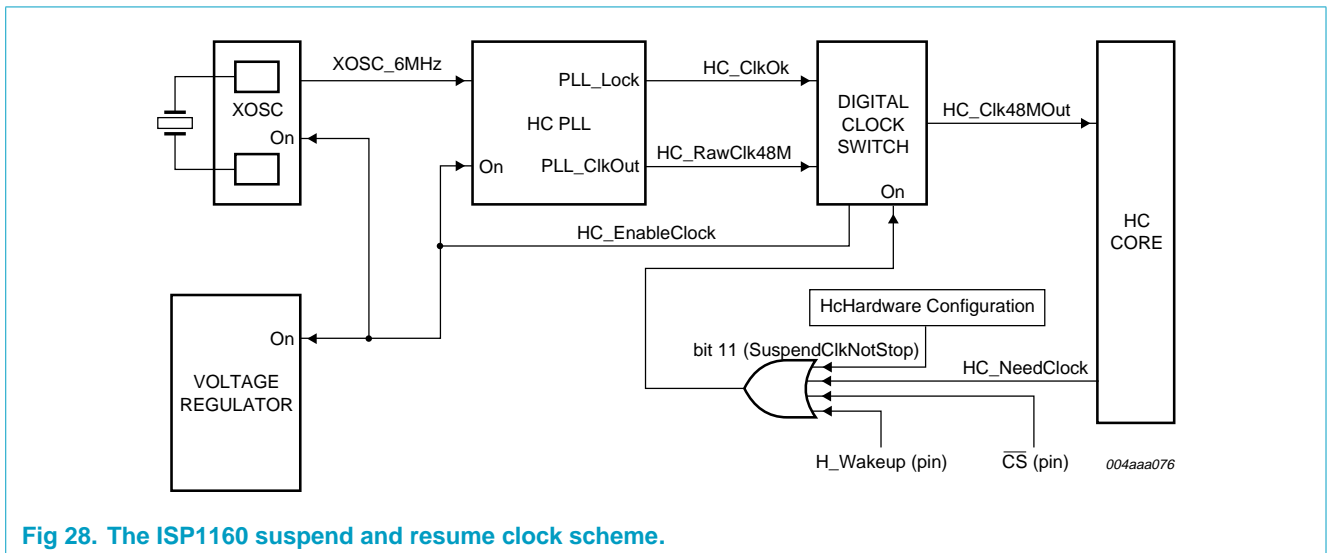
When the ISP1160 is in a suspended state, it will consume considerably less power by turning off the internal 48 MHz clock, PLL and crystal, and setting the internal regulator to power-down mode. The ISP1160 suspend and resume clock scheme is shown in Figure 28.

Pin H_SUSPEND is the sensing output pin for HC's suspended state. When the HC goes into SUSPEND 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 RESET state or OPERATIONAL state (refer to the HcControl register bits 7 to 6, 01H - Read, 81H - Write). By setting bit 11, SuspendClkNotStop, of the HcHardwareConfiguration register (20H - Read, A0H - Write), you can also define such that when the HC goes into SUSPEND state, its internal clock is stopped or kept running. After the HC enters the SUSPEND state for 1.3 ms, the internal clock will be stopped if bit SuspendClkNotStop is logic 0.

9.9.2 HC wake-up from suspended state

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

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 SUSPEND state, it can be woken up by sending a HIGH level pulse to pin H_WAKEUP. This will turn on the HC's internal clock, and set bit 6, ClkReady, of the HcμPInterrupt register (24H - Read, A4H - Write). Under the SUSPEND 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 OPERATIONAL 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 SUSPEND state.

Wake-up by pin $\overline{CS}$ (software wake-up): During the SUSPEND state, an external microprocessor issues the chip select signal through pin $\overline{CS}$ to the ISP1160. This method of access to ISP1160 internal registers is a software wake-up.

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 SUSPEND state to RESUME State. This will also set the ResumeDetected bit, bit 3 of the HcInterruptStatus register (03H - Read, 83H - Write).

No matter which method is used to wake up the HC from SUSPEND state, you must enable the corresponding interrupt bits before the HC goes into SUSPEND 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 bits). These operational registers are made compatible to OpenHCI (Host Controller Interface) operational registers. This makes a provision that the OpenHCI HCD can be ported to the ISP1160 easily.

Reserved bits may be defined in future releases of this specification. To ensure interoperability, the HCD that does not use a reserved field must not assume that the 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 read 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 written, bits written to reserved fields must be logic 0.

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

Table 7:   HC registers summary

Address (Hex)		Register	Width	Functionality
Read	Write			
00	N/A	HcRevision	32	HC control and status registers
01	81	HcControl	32	
02	82	HcCommandStatus	32	
03	83	HcInterruptStatus	32	
04	84	HcInterruptEnable	32	
05	85	HcInterruptDisable	32	
0D	8D	HcFmInterval	32	HC frame counter registers
0E	N/A	HcFmRemaining	32	
0F	N/A	HcFmNumber	32	
11	91	HcLSThreshold	32	
12	92	HcRhDescriptorA	32	HC Root Hub registers
13	93	HcRhDescriptorB	32	
14	94	HcRhStatus	32	
15	95	HcRhPortStatus[1]	32	
16	96	HcRhPortStatus[2]	32	
20	A0	HcHardwareConfiguration	16	HC DMA and interrupt control registers
21	A1	HcDMAConfiguration	16	
22	A2	HcTransferCounter	16	
24	A4	HcµPInterrupt	16	
25	A5	HcµPInterruptEnable	16	

Table 7:   HC registers summary…*continued*

Address (Hex)		Register	Width	Functionality
Read	**Write**			
27	N/A	HcChipID	16	HC miscellaneous registers
28	A8	HcScratch	16	
N/A	A9	HcSoftwareReset	16	
2A	AA	HcITLBufferLength	16	HC buffer RAM control registers
2B	AB	HcATLBufferLength	16	
2C	N/A	HcBufferStatus	16	
2D	N/A	HcReadBackITL0Length	16	
2E	N/A	HcReadBackITL1Length	16	
40	C0	HcITLBufferPort	16	
41	C1	HcATLBufferPort	16	

10.1 HC control and status registers

10.1.1 HcRevision register (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	1	0	1	0	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. For example, a value of 11H corresponds to version 1.1. All HC implementations that are compliant with this specification will have a value of 10H.

Code (Hex): 00 — read only

10.1.2 HcControl register (01H—Read, 81H—Write)

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

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.

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **39 of 89**

Table 11: **HcControl register: bit description**…*continued*

Bit	Symbol	Description
8	-	reserved
7 to 6	HCFS	**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 may be changed by the HC only when in the USBSUSPEND state. The HC may 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

Code (Hex): 01 — read

Code (Hex): 81 — write

10.1.3 HcCommandStatus register (02H—Read, 82H—Write)

The HcCommandStatus register is used by the HC to receive commands issued by the HCD, and it also reflects the HC's 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.

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 HCD to initiate a software reset of 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; e.g., the InterruptRouting field of HcControl, 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 s. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports.

Code (Hex): 02 — read

Code (Hex): 82 — write

10.1.4   HcInterruptStatus register (03H—Read, 83H—Write)

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 becomes set, a hardware interrupt is generated if the interrupt is enabled in the HcInterruptEnable register (see Section 10.1.5) and the MasterInterruptEnable bit is set. The HCD can clear specific bits in this register by writing logic 1 to the bit positions to be cleared, but cannot set any of these bits. The HC can set bits in this register, but cannot clear these bits.

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[NumberofDownstreamPort] has changed.
5	FNO	**FrameNumberOverflow:** This bit is set when the MSB of HcFmNumber (bit 15) changes value, from logic 0 to 1 or from logic 1 to 0.
4	UE	**UnrecoverableError:** This bit is set when the HC detects a system error not related to USB. The HC should 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. PHCI: 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. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets 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.

Code (Hex): 03 — read

Code (Hex): 83 — write

10.1.5   HcInterruptEnable register (04H—Read, 84H—Write)

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. When these three conditions occur:

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

Then a hardware interrupt is requested on the host bus.

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.

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
3	RD	**0** — ignore **1** — enable interrupt generation due to Resume Detect

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **43 of 89**

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

Bit	Symbol	Description
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

Code (Hex): 04 — read

Code (Hex): 84 — write

10.1.6 HcInterruptDisable register (05H—Read, 85H—Write)

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.

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

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **44 of 89**

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

Code (Hex): 05 — read

Code (Hex): 85 — write

10.2 HC frame counter registers

10.2.1 HcFmInterval register (0DH—Read, 8DH—Write)

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 over the present one at each SOF. This provides the programmability necessary for the HC to synchronize with an external clocking resource and to adjust any unknown local clock offset.

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 nominal value is set to be 11 999. The HCD must store the current value of this field before resetting the HC. By setting the HostControllerReset bit 0 field of HcCommandStatus register will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon completing the reset sequence.

Code (Hex): 0D — read

Code (Hex): 8D — write

10.2.2   HcFmRemaining register (0EH—Read only)

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

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 HcFmInterval whenever FrameRemaining reaches 0. This bit is used by the Host Controller Driver (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 HcFmInterval 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.

Code (Hex): 0E — read

10.2.3 HcFmNumber register (0FH—Read only)

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.

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 will be rolled over to 0H after FFFFH. When the USBOPERATIONAL state is entered, this will be incremented automatically. HC will set the StartofFrame in HcInterruptStatus.

Code (Hex): 0F — read

10.2.4 HcLSThreshold register (11H—Read, 91H—Write)

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.

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 $\geq$ this field. The value is calculated by the HCD, which considers transmission and set-up overhead.

Code (Hex): 11 — read

Code (Hex): 91 — write

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 a functionally 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 Hub's Device, Configuration, Interface, and Endpoint Descriptors are maintained only in the HCD as well as some static fields of the Class Descriptor. The HCD also maintains and decodes the Root Hub's device address as well as other trivial operations that they are better suited to software than to hardware.

The Root Hub registers are developed to maintain the similarity of bit organization and operation to typical hubs found in the system.

Four registers are defined as follows:

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

Each register is read and written as a Dword. These registers are only written during initialization to correspond with the system implementation. The HcRhDescriptorA and HcRhDescriptorB registers should be implemented such that they are writeable regardless of the HCs USB states. HcRhStatus and HcRhPortStatus must be writeable during the USBOPERATIONAL state.

10.3.1 HcRhDescriptorA register (12H—Read, 92H—Write)

The HcRhDescriptorA register is the first register of two describing the characteristics of the Root Hub. Reset values are implementation-specific (IS). 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 the registers HcRhDescriptorA and HcRhDescriptorB.

Remark: IS denotes an implementation-specific reset value for that field.

Table 28: **HcRhDescriptorA register: bit description**

Bit	31	30	29	28	27	26	25	24
Symbol	POTPGT[7:0]							
Reset	IS	IS	IS	IS	IS	IS	IS	IS
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	IS	IS	0	IS	IS
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]	
Reset	0	0	0	0	0	0	IS	IS
Access	R	R	R	R	R	R	R	R

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. It is implementation-specific (IS). The unit of time is 2 ms. The duration is calculated as POTPGT $\times$ 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 should reflect 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:** These bits are used to specify whether power switching is supported or ports are always powered. It is implementation-specific. 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. It is implementation-specific. 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 the 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. It is implementation-specific. The maximum number of ports supported is 2.

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **50 of 89**

Code (Hex): 12 — read

Code (Hex): 92 — write

10.3.2 HcRhDescriptorB register (13H—Read, 93H—Write)

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. Reset values are implementation-specific (IS).

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	IS	IS	IS
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	IS	IS	IS
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 port's 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

Code (Hex): 13 — read

Code (Hex): 93 — write

10.3.3 HcRhStatus register (14H—Read, 94H—Write)

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

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			CCIC	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	CCIC	**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.
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 PortPowerControlMask bit is not set. Writing a logic 0 has no effect.

Code (Hex): 14 — read

Code (Hex): 94 — write

10.3.4   HcRhPortStatus[1:2]
([1]:15H—Read, 95H—Write, [2]: 16H—Read, 96H—Write)

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.

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 re-synchronization 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

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

Bit	Symbol	Description
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 re-evaluate 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 the DeviceRemovable[NDP] bit 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	(read) **LowSpeedDeviceAttached:** This bit indicates the speed of the device attached to this port. When set, a low-speed device is attached to this port. When clear, a full-speed device is attached to this port. This field is valid only when the CurrentConnectStatus is set. **0 —** full-speed device attached **1 —** low-speed device attached (write) **ClearPortPower:** The HCD clears the PortPowerStatus bit 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	(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 the PortPowerControlMask[NDP] bit 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
		(write) **SetPortPower:** The HCD writes a logic 1 to set the PortPowerStatus bit. 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	(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
		(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	(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
		(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	(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 (write) **SetPortSuspend:** The HCD sets the PortSuspendStatus bit 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	(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, if it is not already, at the completion of a port reset when ResetStatusChange is set or port suspend when SuspendStatusChange is set. **0 —** port is disabled **1 —** port is enabled (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	(read) **CurrentConnectStatus:** This bit reflects the current state of the downstream port. **0 —** no device connected **1 —** device connected (write) **ClearPortEnable:** The HCD writes a logic 1 to this bit to clear the PortEnableStatus bit. 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]).

Code (Hex): [1] = 15, [2] = 16 — read

Code (Hex): [1] = 95, [2] = 96 — write

10.4 HC DMA and interrupt control registers

10.4.1 HcHardwareConfiguration register (20H—Read, A0H—Write)

Table 36:   HcHardwareConfiguration register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved			2_Down stream 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_DownstreamPort15Kresistor Sel	**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 can not 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. $\overline{\text{DACK}}$ is used with read and write signals. Power-up value
		1 — reserved
7	EOTInputPolarity	**0** — active LOW. Power-up value
		1 — active HIGH
6	DACKInputPolarity	**0** — active LOW. Power-up value
		1 — reserved
5	DREQOutputPolarity	**0** — active LOW
		1 — active HIGH. Power-up value
4 to 3	DataBusWidth[1:0]	**01** — 16 bits
		Others — reserved

Table 37:   HcHardwareConfiguration register: bit description…*continued*

Bit	Symbol	Description
2	InterruptOutputPolarity	**0 —** active LOW. Power-up value
		1 — active HIGH
1	InterruptPinTrigger	**0 —** interrupt is level-triggered. Power-up value
		1 — interrupt is edge-triggered
0	InterruptPinEnable	**0 —** power-up value
		1 — global interrupt pin INT is enabled

Code (Hex): 20 — read

Code (Hex): A0 — write

Remark:

1. Bit 0, InterruptPinEnable, is used as pin INT's master interrupt enable. This bit should be used together with the register HcμPInterruptEnable to enable pin INT.
2. Bits 4:3 are fixed at the value 01B for the ISP1160.

10.4.2   HcDMAConfiguration register (21H—Read, 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]	**00B —** single-cycle burst DMA
		01B — 4-cycle burst DMA
		10B — 8-cycle burst DMA
		11B — reserved

Table 39: HcDMAConfiguration register: bit description…*continued*

Bit	Symbol	Description
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	DMACounter Select	**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	DMARead WriteSelect	**0 —** read from the ISP1160 HC's FIFO buffer RAM
		1 — write to the ISP1160 HC's FIFO buffer RAM

Code (Hex): 21 — read

Code (Hex): A1 — write

10.4.3 HcTransferCounter register (22H—Read, A2H—Write)

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 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 AllEOInterrupt, of the HcµPInterrupt register, and also update the HcBufferStatus register.

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.

Code (Hex): 22 — read

Code (Hex): A2 — write

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **60 of 89**

10.4.4 HcµPInterrupt register (24H—Read, A4H—Write)

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 - Read) is read, the bits that are active will not be reset, until logic 1 is written to the bits in this register (A4H - Write) to clear it.

The bits in this register are cleared only when you write to this register indicating the bits to be cleared. To clear all the enabled bits in this register, the HCD must write FFH to this register.

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 and no USB activity is sent from the microprocessor for each ms. 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

Table 43: HcμPInterrupt register: bit description…continued

Bit	Symbol	Description
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. The time for this interrupt depends on the number of clocks bit set for USB activities in each ms.
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 microprocessor to get ISO data to or from the HC. For more information, see the 6th paragraph in Section 9.5.

Code (Hex): 24 — read

Code (Hex): A4 — write

10.4.5 HcμPInterruptEnable register (25H—Read, A5H—Write)

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.

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 register's 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. The time for this interrupt depends on the number of clock bits set for USB activities in each ms.
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)

Code (Hex): 25 — read

Code (Hex): A5 — write

10.5 HC miscellaneous registers

10.5.1 HcChipID register (27H—Read only)

Read this register to get the ID of the ISP1160 silicon chip. The high byte stands for the product name (here 61H stands for the ISP1160). The low byte indicates the revision number of the product including engineering samples.

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	0
Access	R	R	R	R	R	R	R	R

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **63 of 89**

Table 47:   HcChipID register: bit description

Bit	Symbol	Description
15 to 0	ChipID[15:0]	ISP1160's chip ID

Code (Hex): 27 — read

10.5.2   HcScratch register (28H—Read, A8H—Write)

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

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

Code (Hex): 28 — read

Code (Hex): A8 — write

10.5.3   HcSoftwareReset register (A9H—Write)

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.

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.

Code (Hex): A9 — write

10.6 HC buffer RAM control registers

10.6.1 HcITLBufferLength register (2AH—Read, AAH—Write)

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 kbytes, then ITL0 and ITL1 would be allocated 2 kbytes each.

Must follow the formula:

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

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

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

Code (Hex): 2A — read

Code (Hex): AA — write

10.6.2 HcATLBufferLength register (2BH—Read, ABH—Write)

Write to this register to assign ATL buffer size.

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

Code (Hex): 2B — read

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **65 of 89**

Code (Hex): AB — 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.

10.6.3 HcBufferStatus register (2CH—Read only)

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	ATLBuffer Done	**0** — ATL Buffer not read by HC yet
		1 — ATL Buffer read by HC
4	ITL1Buffer Done	**0** — ITL1 Buffer not read by HC yet
		1 — ITL1 Buffer read by HC
3	ITL0Buffer Done	**0** — 1TL0 Buffer not read by HC yet
		1 — 1TL0 Buffer read by HC
2	ATLBuffer Full	**0** — ATL Buffer is empty
		1 — ATL Buffer is full
1	ITL1Buffer Full	**0** — 1TL1 Buffer is empty
		1 — 1TL1 Buffer is full
0	ITL0Buffer Full	**0** — ITL0 Buffer is empty
		1 — ITL0 Buffer is full

Code (Hex): 2C — read

10.6.4 HcReadBackITL0Length register (2DH—Read only)

This register's 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.

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **66 of 89**

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

Code (Hex): 2D — read

10.6.5 HcReadBackITL1Length register (2EH—Read only)

This register's 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.

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

Code (Hex): 2E — read

10.6.6 HcITLBufferPort register (40H—Read, C0H—Write)

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

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 RAM's two data bytes.

Code (Hex): 40 — read

Code (Hex): C0 — write

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 for read, C0H for write) once only, and then read or write all the data bytes in 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 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 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 (41H—Read, C1H—Write)

This is the ATL buffer RAM read/write port. The bits 15:8 contain the data byte that comes from the Acknowledged Transfer List (ATL) buffer RAM's odd address. The bits 7:0 contain the data byte that comes from the ATL buffer RAM's even address.

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 RAM's two data bytes.

Code (Hex): 41 — read

Code (Hex): C1 — write

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 for read, C1H for write) once only, and then read or write all the data bytes in 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 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 word).

11. Power supply

The ISP1160 can operate at either 5 V or 3.3 V.

When using 5 V as ISP1160's 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 ISP1160 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 $V_{reg(3.3)}$ (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}, $V_{reg(3.3)}$, 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 the V_{CC} and the $V_{reg(3.3)}$ 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.
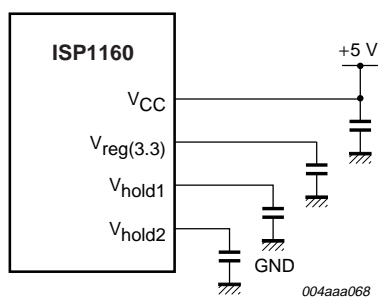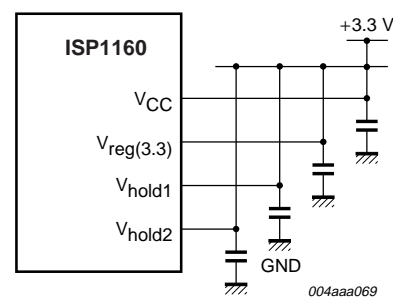

Fig 29. Using a 5 V supply.

Fig 30. Using a 3.3 V supply.

9397 750 10765

Product data

Rev. 03 — 27 February 2003

69 of 89

12. Crystal oscillator

The ISP1160 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.
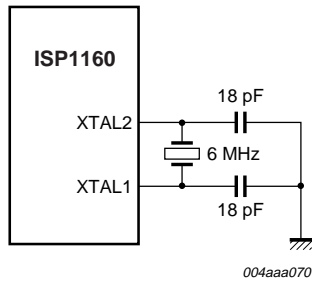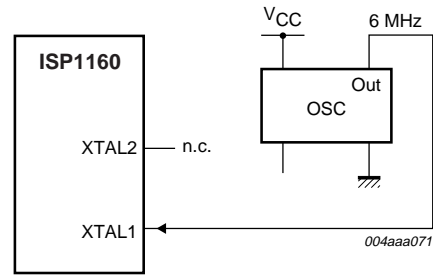
Fig 31. Oscillator circuit with external crystal.

Fig 32. Oscillator circuit using external oscillator.

13. Limiting values

Table 66: Absolute maximum ratings
In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{CC(5V)}$	supply voltage to pin V_{CC}		−0.5	+6.0	V
$V_{CC(3.3V)}$	supply voltage to pin $V_{reg(3.3)}$		−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	$I_{LI} < 1\ \mu A$ [1]	-	±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).

14. Recommended operating conditions

Table 67: Recommended operating conditions

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		0	V_{CC}	5.5[1]	V
$V_{I(A\ I/O)}$	input voltage on analog I/O pins (D+, 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] 5 V tolerant.

15. Static characteristics

Table 68:   Static characteristics; supply pins

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CC} = 5 V						
$V_{reg(3.3)}$	internal regulator output	[1]	3.0	3.3	3.6	V
I_{CC}	operating supply current		-	47	-	mA
$I_{CC(susp)}$	suspend supply current		-	40	500	µA
V_{CC} = 3.3 V						
I_{CC}	operating supply current		-	50	-	mA
$I_{CC(susp)}$	suspend supply current		-	150	500	µA

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

Table 69:   Static characteristics: digital pins

V_{CC} = 3.0 to 3.6 V or 4.0 to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 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	-	-	0.1	V
V_{OH}	HIGH-level output voltage	I_{OH} = 4 mA [1]	2.4	-	-	V
		I_{OH} = 20 µA	$V_{reg(3.3)}$ − 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	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+, D−)**

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

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Input levels							
V_{DI}	differential input sensitivity	$\|V_{I(D+)} - V_{I(D-)}\|$	[1]	0.2	-	-	V
V_{CM}	differential common mode voltage	includes V_{DI} range		0.8	-	2.5	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		2.8	-	3.6	V
Leakage current							
I_{LZ}	OFF-state leakage current			-	-	±10	µA
Capacitance							
C_{IN}	transceiver capacitance	pin to GND		-	-	10	pF
Resistance							
R_{PD}	pull-down resistance on HC's DP/DM	enable internal resistors		10	-	20	kΩ
Z_{DRV}	driver output impedance	steady-state drive	[2]	29	-	44	Ω
Z_{INP}	input impedance			10	-	-	MΩ

[1] D+ is the USB positive data pin; D− is the USB negative data pin.

[2] Includes external resistors of 18 Ω ±1% on both H_D+ and H_D−.

16. Dynamic characteristics

Table 71: Dynamic characteristics

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Reset						
$t_{W(RESET)}$	pulse width on input $\overline{RESET}$	crystal oscillator running	160	-	-	ms
		crystal oscillator stopped [1]	-		-	ms
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

[1]   Dependent on the crystal oscillator start-up time.

Table 72: Dynamic characteristics: analog I/O pins (D+, D−)[1]

V_{CC} = 3.0 to 3.6 V or 4.0 to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 to +85 °C; C_L = 50 pF; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Driver characteristics						
t_{FR}	rise time	C_L = 50 pF; 10% to 90% of $|V_{OH} - V_{OL}|$	4	-	20	ns
t_{FF}	fall time	C_L = 50 pF; 90% to 10% of $|V_{OH} - V_{OL}|$	4	-	20	ns
FRFM	differential rise/fall time matching (t_{FR}/t_{FF})	[2]	90	-	111.11	%
V_{CRS}	output signal crossover voltage	[2][3]	1.3	-	2.0	V

[1]   Test circuit; see Figure 40.

[2]   Excluding the first transition from Idle state.

[3]   Characterized only, not tested. Limits guaranteed by design.

16.1 Timing symbols

Table 73: Legend for timing characteristics

Symbol	Description
Time symbols	
t	time
T	cycle time (periodic signal)
Signal names	
A	address;
	DMA acknowledge ($\overline{\text{DACK}}$)
C	clock;
	command
D	data input;
	data
E	chip enable
G	output enable
I	instruction (program memory content);
	input (general)
P	program store enable ($\overline{\text{PSEN}}$, active LOW);
	propagation delay
Q	data output
R	read signal ($\overline{\text{RD}}$, active LOW);
	read (action);
	DMA request (DREQ)
S	chip select
W	write signal ($\overline{\text{WR}}$, active LOW);
	write (action);
	pulse width
U	undefined
Y	output (general)
Logic levels	
H	logic HIGH
L	logic LOW
P	stop, not active (OFF)
S	start, active (ON)
V	valid logic level
X	invalid logic level
Z	high-impedance (floating, three-state)

9397 750 10765

Product data   **Rev. 03 — 27 February 2003**   **75 of 89**

16.2 Programmed I/O timing

16.2.1 Programmed I/O timing

Table 74: Dynamic characteristics: programmed interface timing

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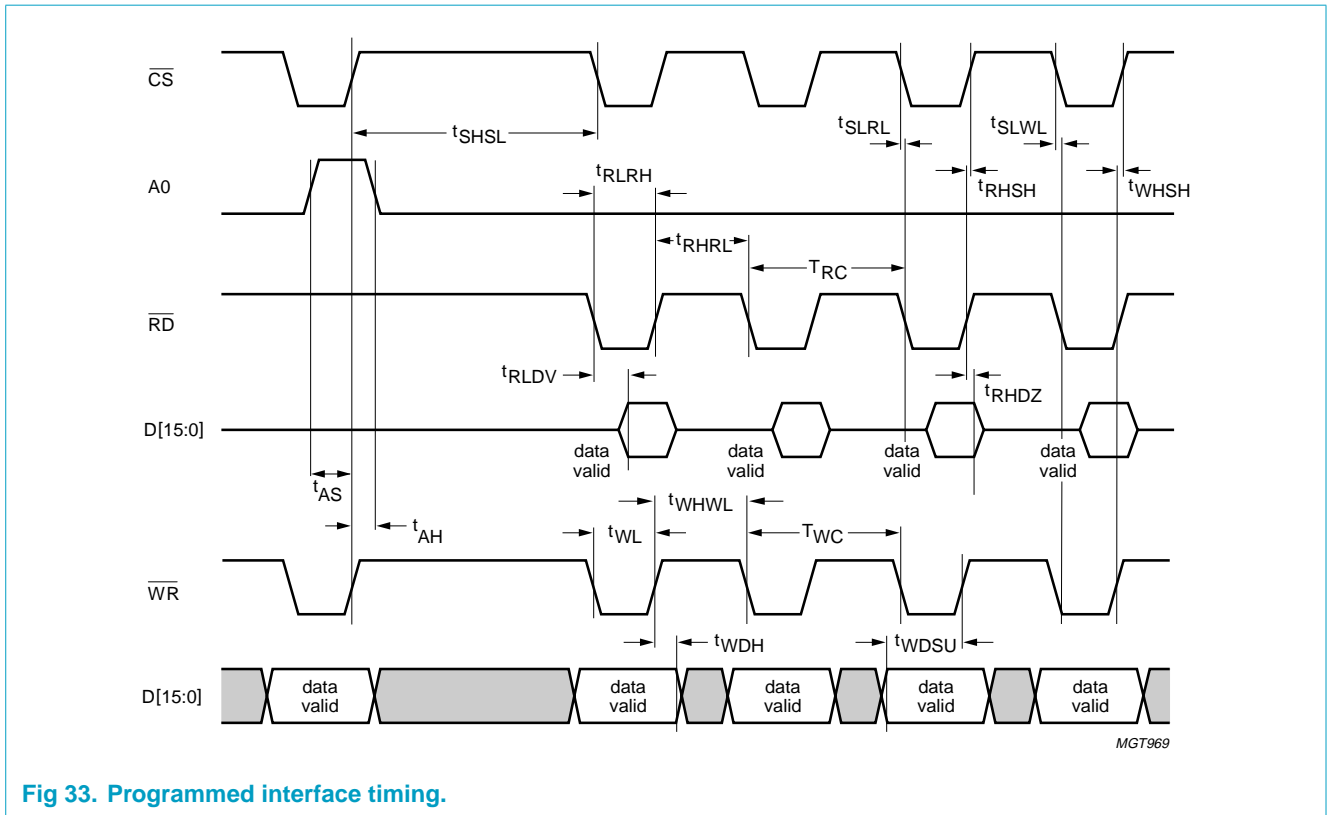
Fig 33. Programmed interface timing.

16.3 DMA timing

16.3.1 Single-cycle DMA timing

Table 75: Dynamic characteristics: single-cycle DMA timing

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Read/write timing						
t_{RLRH}	$\overline{RD}$ 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}	$\overline{DACK}$ HIGH to DREQ HIGH		72	-	-	ns
t_{ALRL}	$\overline{DACK}$ LOW to DREQ LOW		-	-	21	ns
T_{DC}	DREQ cycle	[1]	-	-	-	ns
t_{SHAH}	$\overline{RD}/\overline{WR}$ HIGH to $\overline{DACK}$ HIGH		0	-	-	ns
t_{RHAL}	DREQ HIGH to $\overline{DACK}$ LOW		0	-	-	ns
t_{DS}	DREQ pulse spacing		146	-	-	ns

[1] $t_{RHAL} + t_{DS} + t_{ALRL}$
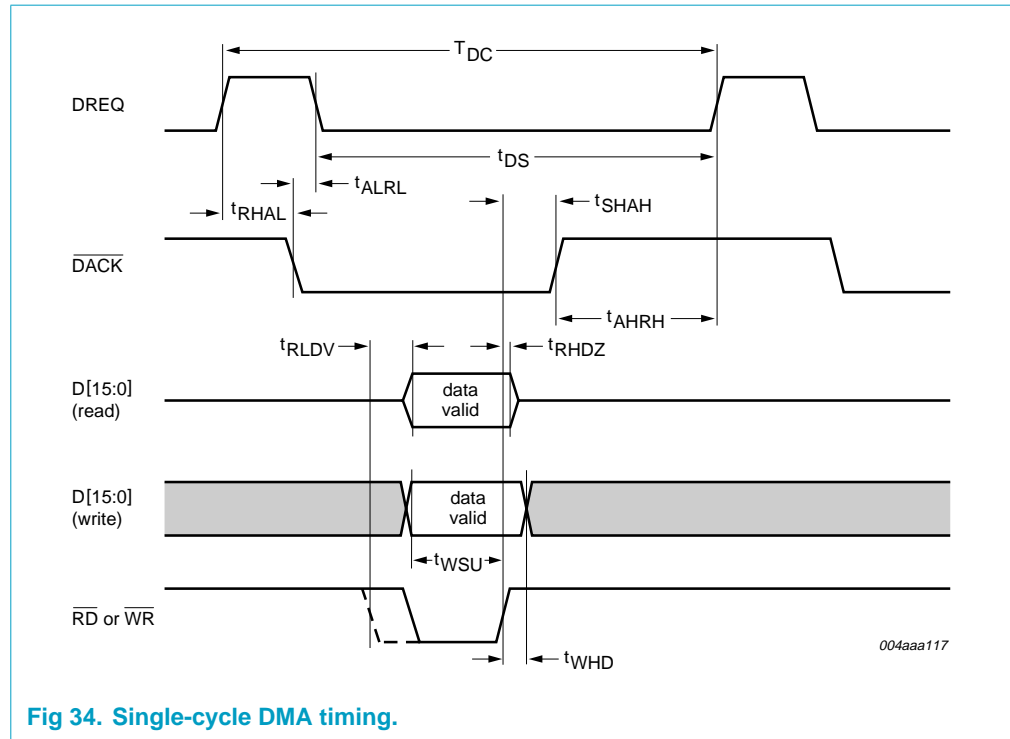
Fig 34. Single-cycle DMA timing.

16.3.2 Burst mode DMA timing

Table 76: 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}	$\overline{WR}/\overline{RD}$ LOW pulse width		42	-	-	ns
t_{RHRL}	$\overline{WR}/\overline{RD}$ HIGH to next $\overline{WR}/\overline{RD}$ LOW		60	-	-	ns
T_{RC}	$\overline{WR}/\overline{RD}$ cycle		102	-	-	ns
t_{SLRL}	$\overline{RD}/\overline{WR}$ LOW to DREQ LOW		22	-	64	ns
t_{SHAH}	$\overline{RD}/\overline{WR}$ HIGH to $\overline{DACK}$ HIGH		0	-	-	ns
t_{SLAL}	DREQ HIGH to $\overline{DACK}$ 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}	$\overline{RD}/\overline{WR}$ LOW to EOT LOW		0	-	-	ns

[1] t_{SLAL} + (4 or 8)T_{RC} + t_{DS}
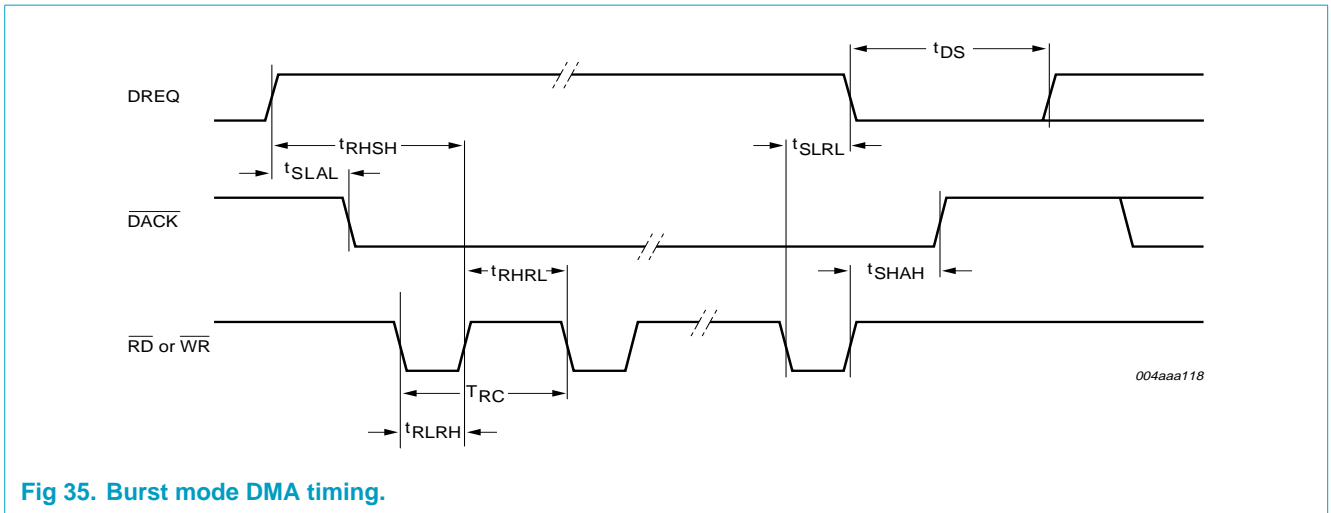
Fig 35. Burst mode DMA timing.

16.3.3 External EOT timing for single-cycle DMASETUP
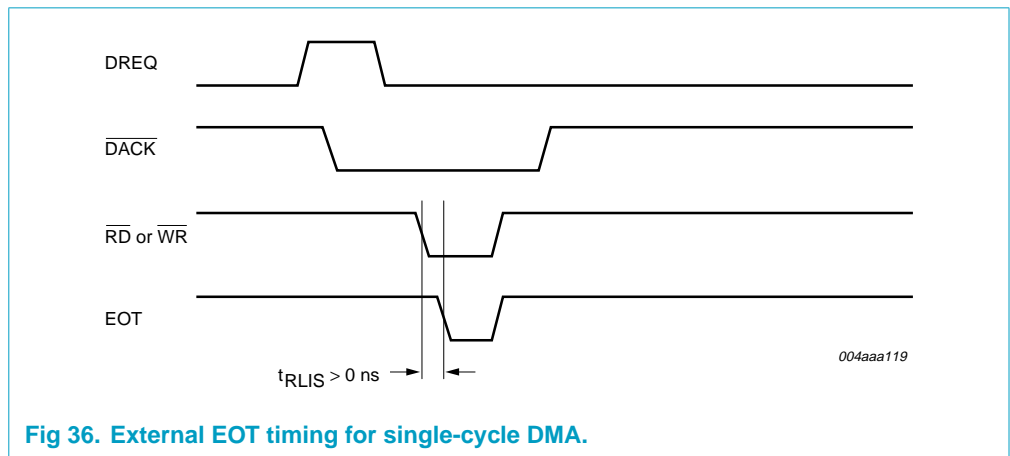
Fig 36. External EOT timing for single-cycle DMA.

16.3.4 External EOT timing for burst mode DMA
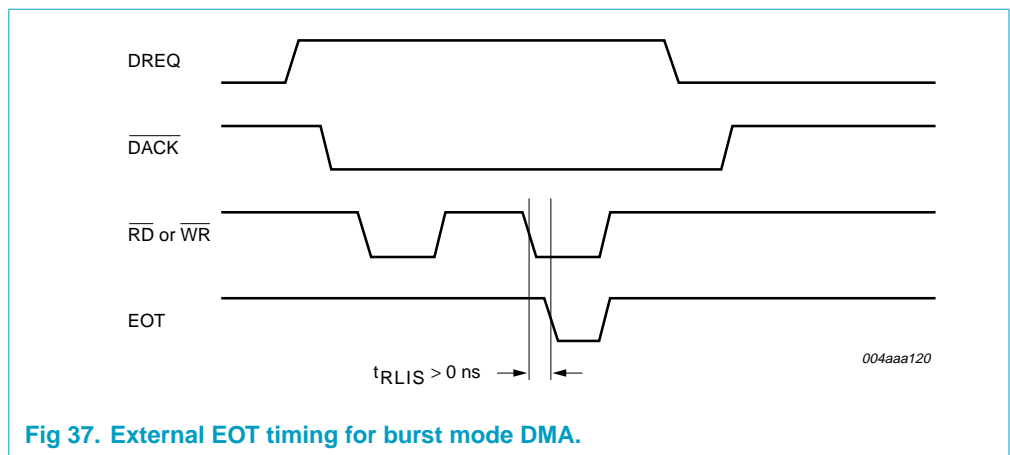
Fig 37. External EOT timing for burst mode DMA.

17. Application information

17.1 Typical interface circuit

For MOSFET, R_{DSon} = 150 mΩ.

Fig 38. Typical interface circuit to Hitachi SH-3 (SH7709) RISC processor.
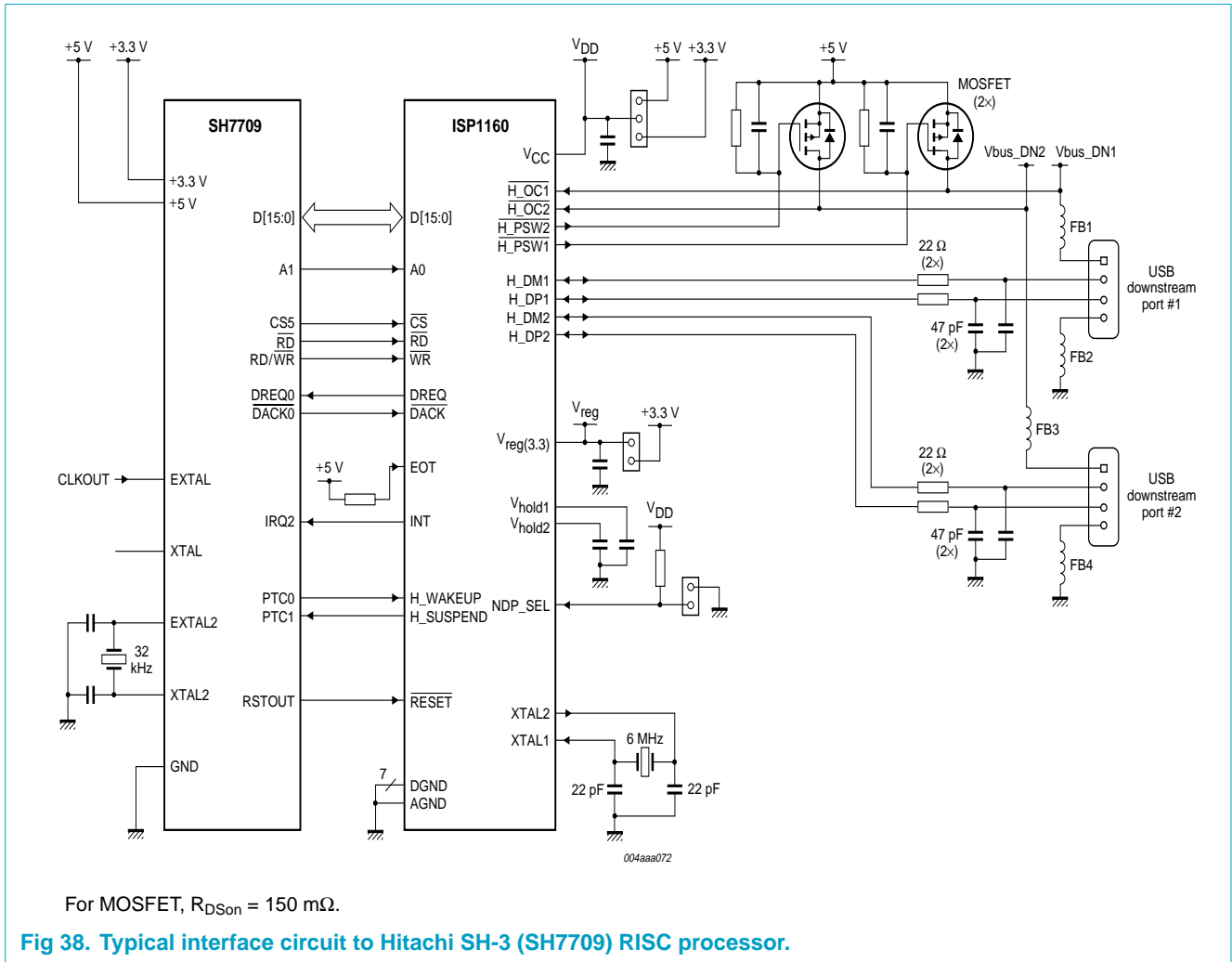
17.2 Interfacing a ISP1160 to a SH7709 RISC processor

This section shows a typical interface circuit between the ISP1160 and a RISC processor. The Hitachi SH-3 series RISC processor SH7709 is used as the example. The main ISP1160 signals to be taken into consideration for connecting to a SH7709 RISC processor are:

• A 16-bit data bus: D15 to D0 for the ISP1160. The ISP1160 is 'little endian' compatible.

• The address line A0 is needed for a complete addressing of the ISP1160 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 $\overline{CS}$ line is used for chip selection of the ISP1160 in a certain address range of the RISC system. This signal is active LOW.
- $\overline{RD}$ and $\overline{WR}$ are common read and write signals. These signals are active LOW.
- There is a DMA channel standard control line:
  - DREQ and $\overline{DACK}$

  (the channel is used by the host controller). The DREQ signal has programmable active levels.
- An interrupt line INT is used by the host controller. 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 $\overline{RESET}$ signal is active LOW.

Remark: SH7709's system clock input is for reference only. Please refer to SH7709's specification for its actual use.

The ISP1160 can work under either 3.3 V or 5.0 V power supply; however, its internal core actually 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}, $V_{reg(3.3)}$, V_{hold1} and V_{hold2}) to the 3.3 V power supply (for more information, see Section 11). All of the ISP1160's I/O pins are 5 V-tolerant. This feature allows the ISP1160 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 38.

17.3 Typical software model

This section shows a typical software requirement for an embedded system that incorporates the ISP1160. The software model for a Digital Still Camera (DSC) is used as the example for illustration (as shown in Figure 39). The host stack provides API for Class driver and device driver, both of which provide API for application tasks for host function.
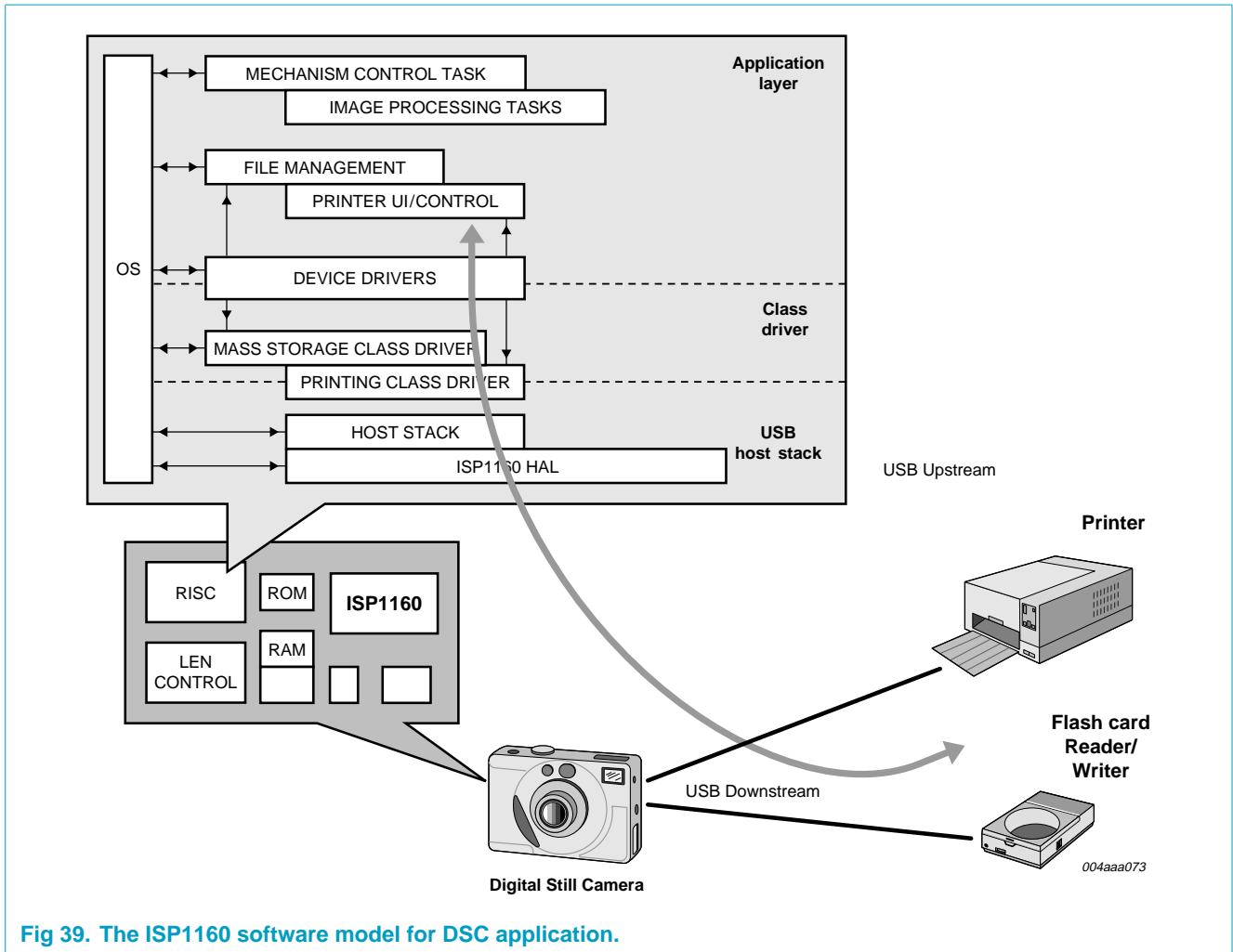
Fig 39. The ISP1160 software model for DSC application.

18. Test information

The dynamic characteristics of the analog I/O ports (D+ and D−) as listed in Table 72 were determined using the circuit shown in Figure 40.
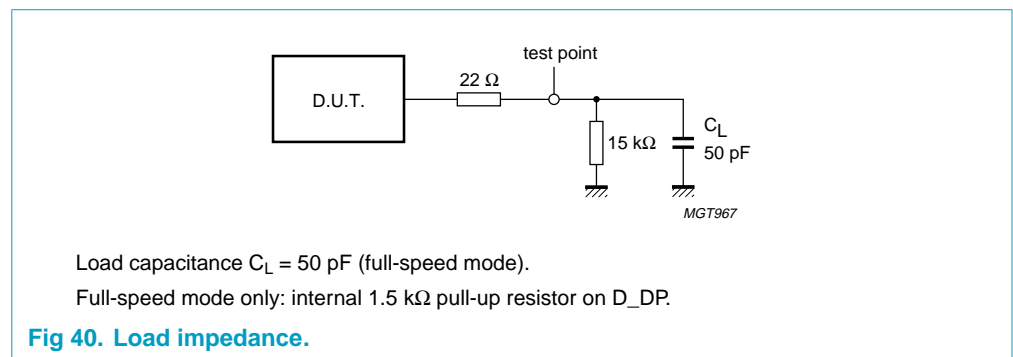

Load capacitance C_L = 50 pF (full-speed mode).

Full-speed mode only: internal 1.5 kΩ pull-up resistor on D_DP.

Fig 40. Load impedance.

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.	A₁	A₂	A₃	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	1.60	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	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	EIAJ			
SOT314-2	136E10	MS-026				~~99-12-27~~ 00-01-19
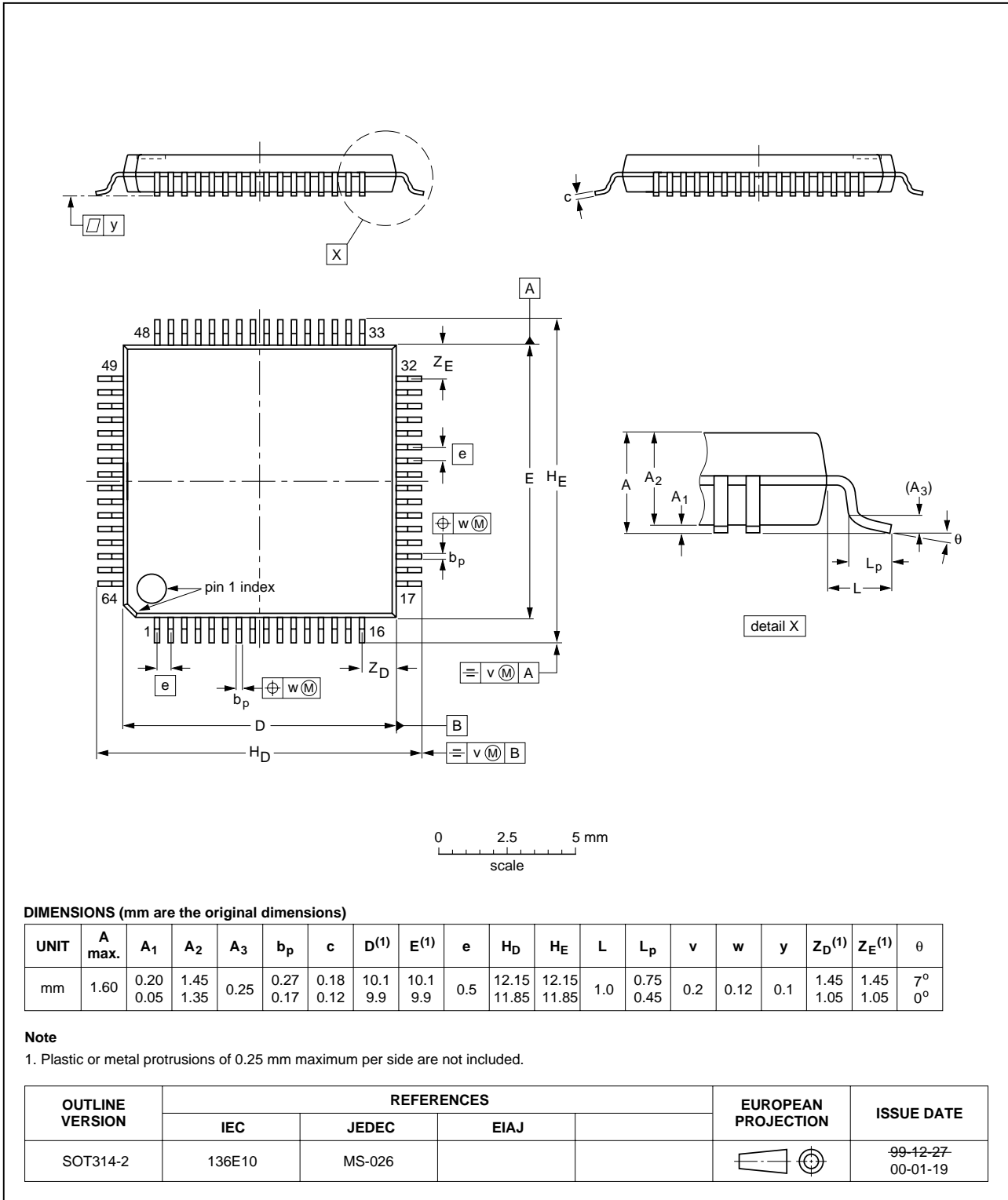
Fig 41. LQFP64 (SOT314-2) package outline.

9397 750 10765

Product data   **Rev. 03 — 27 February 2003**   **83 of 89**

LQFP64: plastic low profile quad flat package; 64 leads; body 7 x 7 x 1.4 mm

SOT414-1

DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A₁	A₂	A₃	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	0.15 0.05	1.45 1.35	0.25	0.23 0.13	0.20 0.09	7.1 6.9	7.1 6.9	0.4	9.15 8.85	9.15 8.85	1.0	0.75 0.45	0.2	0.08	0.08	0.64 0.36	0.64 0.36	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	EIAJ			
SOT414-1	136E06	MS-026				99-12-27 00-01-19

Fig 42. LQFP64 (SOT414-1) package outline.

20. Soldering

20.1 Introduction to soldering surface mount packages

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *Data Handbook IC26; Integrated Circuit Packages* (document order number 9398 652 90011).

There is no soldering method that is ideal for all surface mount IC packages. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

20.2 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several methods exist for reflowing; for example, convection or convection/infrared heating in a conveyor type oven. Throughput times (preheating, soldering and cooling) vary between 100 and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 to 250 °C. The top-surface temperature of the packages should preferably be kept:

- below 220 °C for all the BGA packages and packages with a thickness $\geq$ 2.5mm and packages with a thickness <2.5 mm and a volume $\geq$350 mm^3 so called thick/large packages
- below 235 °C for packages with a thickness <2.5 mm and a volume <350 mm^3 so called small/thin packages.

20.3 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  – larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;
  – smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

The footprint must incorporate solder thieves at the downstream end.

- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time is 4 seconds at 250 °C. A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

20.4 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C.

When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

20.5 Package related soldering information

Table 77: **Suitability of surface mount IC packages for wave and reflow soldering methods**

Package[1]	Soldering method	
	Wave	**Reflow**[2]
BGA, LBGA, LFBGA, SQFP, TFBGA, VFBGA	not suitable	suitable
DHVQFN, HBCC, HBGA, HLQFP, HSQFP, HSOP, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable[3]	suitable
PLCC[4], SO, SOJ	suitable	suitable
LQFP, QFP, TQFP	not recommended[4][5]	suitable
SSOP, TSSOP, VSO, VSSOP	not recommended[6]	suitable

[1] For more detailed information on the BGA packages refer to the *(LF)BGA Application Note* (AN01026); order a copy from your Philips Semiconductors sales office.

[2] All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the *Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods*.

[3] These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.

[4] If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.

[5] Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.

[6] Wave soldering is suitable for SSOP and TSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.

9397 750 10765

Product data **Rev. 03 — 27 February 2003** **86 of 89**

21. Revision history

Table 78:   Revision history

Rev	Date	CPCN	Description
03	20030227	-	**Product data (9397 750 10765)**
			Modifications:
			• Table 2: updated description for pins 32, 33, 40 and 60.
			• Section 9.4.3: updated the paragraph "In all PTDs, we have assigned device address as 5 and endpoint 1. ActualBytes is always zero....."
			• Updated Figure 24, Figure 26 and Figure 27.
			• Removed Chapter Reset (old Section 11).
			• Table 67: upgraded to Chapter level.
			• Table 69: added table note 2.
			• Table 71: $t_{W(\overline{RESET})}$: changed from 10 µs to 160 µs.
			• Table 74: updated the following:
			– description for t_{AS} and t_{AH}
			– changed the value of t_{RLDV}.
			• Updated Figure 33.
			• Updated Section 25.
02	20020912	-	**Product data (9397 750 09628)**
01	20020104	-	**Objective data (9397 750 09161)**

22. Data sheet status

Level	Data sheet status[1]	Product status[2][3]	Definition
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

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

[2]   The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

[3]   For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

23. Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

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

24. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors

customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

25. Trademarks

ARM7 and ARM9 — are trademarks of ARM Ltd.
GoodLink — is a trademark of Koninklijke Philips Electronics N.V.
Hitachi — is a registered trademark of Hitachi Ltd.
MIPS-based — is a trademark of MIPS Technologies, Inc.
SoftConnect — is a trademark of Koninklijke Philips Electronics N.V.
StrongARM — is a registered trademark of ARM Ltd.
SuperH — is a trademark of Hitachi Ltd.

Contact information

For additional information, please visit **http://www.semiconductors.philips.com**.
For sales office addresses, send e-mail to: **sales.addresses@www.semiconductors.philips.com**.

Fax: +31 40 27 24825

Contents

PHILIPS

Let's make things better.