ISP1183

Low-power USB peripheral controller with DMA

Rev. 04 — 29 September 2009 Product data sheet

1. General description

The ISP1183 is a Universal Serial Bus (USB) Peripheral Controller that complies with *Universal Serial Bus Specification Rev. 2.0*, supporting data transfer at full-speed (12 Mbit/s). It provides full-speed USB communication capacity to microcontroller or microprocessor-based systems. The ISP1183 communicates with the system's microcontroller or microprocessor through a fast general-purpose parallel interface.

The ISP1183 supports fully autonomous, multiconfigurable Direct Memory Access (DMA) operation.

The modular approach to implementing a USB Peripheral Controller allows designer to select the optimum system microcontroller from the wide variety available. The ability to reuse existing architecture and firmware investments shortens development time, eliminates risks and reduces costs. The result is fast and efficient development of the most cost-effective USB peripheral solution.

The ISP1183 supports I/O voltage range of 1.65 V to 3.6 V enabling it to be directly interfaced to battery-operated devices, such as mobile phones. The ISP1183 is ideally suited for battery-operated (low power) application in many portable peripherals such as mobile phones, Personal Digital Assistants (PDAs) and MP3 players. This device can be used in bus-powered or hybrid-powered applications. Also, more number of endpoints in the ISP1183 enable the device to be used in applications such as multifunctional printers, other than standard applications such as printers, communication devices, scanners, external mass storage devices and digital still cameras.

2. Features

- Complies with *Universal Serial Bus Specification Rev. 2.0* and most device class specifications
- Complies with ACPI, OnNow and USB power management requirements
- Supports data transfer at full-speed (12 Mbit/s)
- High performance USB Peripheral Controller with integrated Serial Interface Engine (SIE), FIFO memory, transceiver and 3.3 V voltage regulator
- High-speed (11.1 MB/s or 90 ns read/write cycle) parallel interface
- Fully autonomous and multiconfiguration DMA operation
- Up to 14 programmable USB endpoints with 2 fixed control IN/OUT endpoints
- Integrated physical 2462 bytes of multiconfiguration FIFO memory
- Endpoints with double buffering to increase throughput and ease real-time data transfer
- Seamless interface with most microcontrollers and microprocessors
- Bus-powered capability with low power consumption and low suspend current

- Software controlled connection to the USB bus (SoftConnect[1])
- Supports internal power-on and low-voltage reset circuit
- Supports software reset
- Hybrid-powered capability with low-power consumption required from the system
- V_{BUS} indication
- 6 MHz crystal oscillator input with integrated PLL for low EMI
- Supports I/O voltage range of 1.65 V to 3.6 V
- Operation over the extended USB bus voltage range (4.0 V to 5.5 V) with 3.3 V tolerant I/O pads
- Operating temperature range of −40 °C to +85 °C
- Full-scan design with high fault coverage
- Available in HVQFN32 lead-free and halogen-free package

3. Applications

- Battery-operated device, for example:
  - Mobile phone
  - MP3 player
  - Personal Digital Assistant (PDA)
- Communication device, for example:
  - Router
  - Modem
- Digital camera
- Mass storage device, for example:
  - Zip drive
- Printer
- Scanner

4. Ordering information

Table 1. Ordering information

Commercial product code	Package description	Packing	Minimum sellable quantity
ISP1183BSTM	HVQFN32; 32 terminals; body 5 × 5 × 0.85 mm	13 inch tape and reel non-dry pack	6000 pieces
ISP1183BSFA	HVQFN32; 32 terminals; body 5 × 5 × 0.85 mm	single tray non-dry pack	490 pieces

1. SoftConnect is a trademark of ST-Ericsson.

5. Block diagram

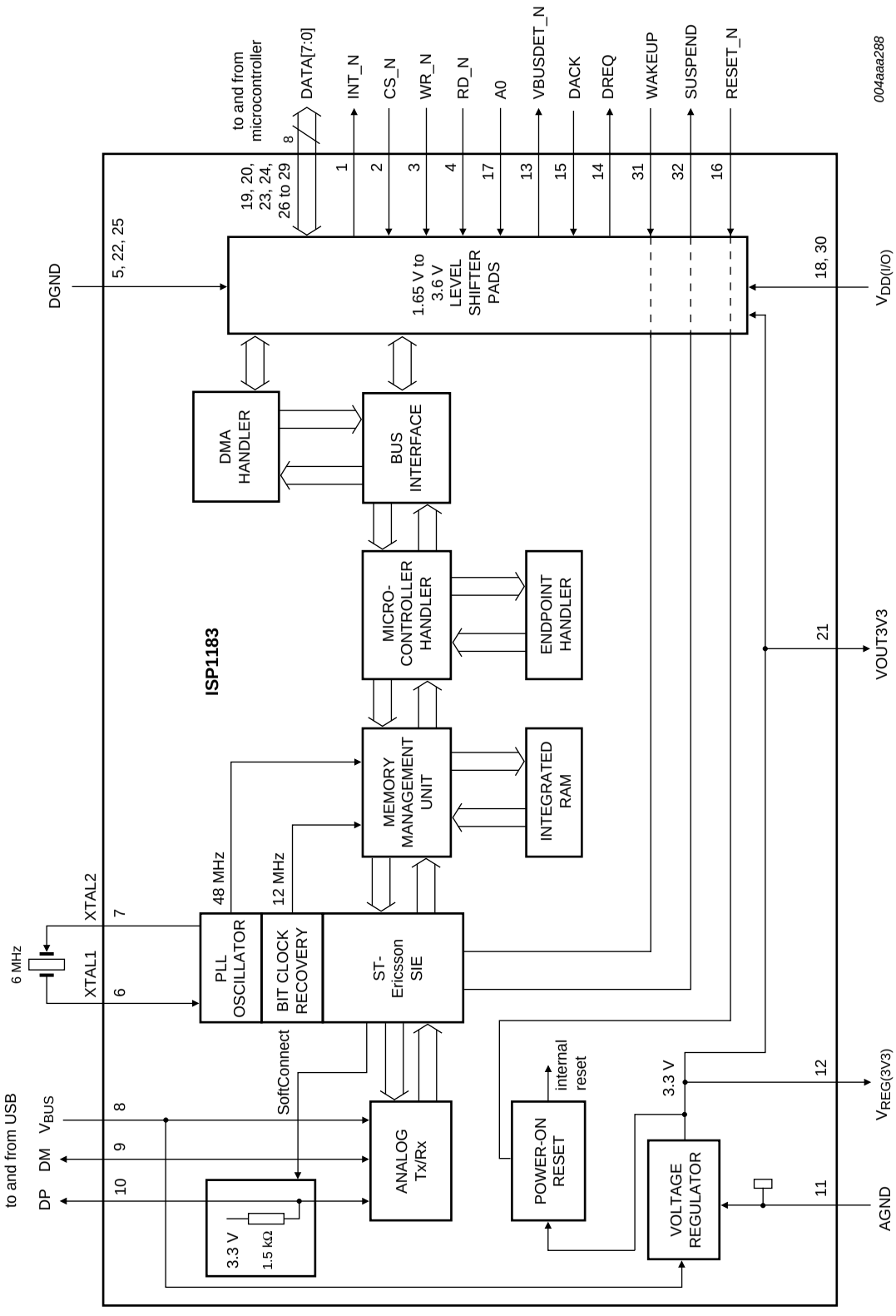


Fig 1. Block diagram

6. Pinning information

6.1 Pinning

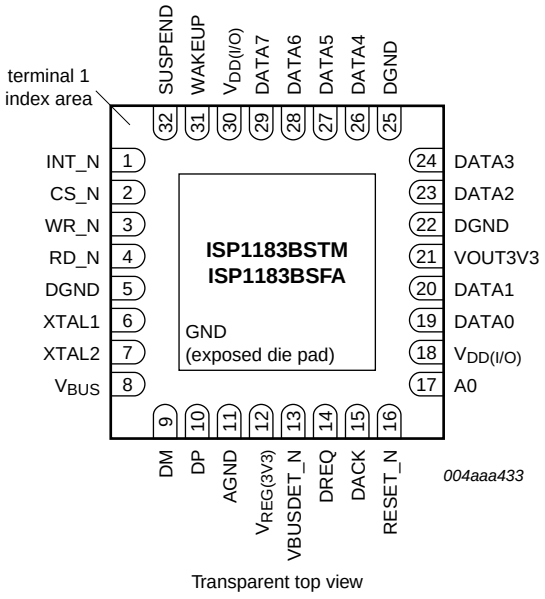


Fig 2. Pin configuration HVQFN32

6.2 Pin description

Table 2. Pin description

Symbol[1]	Pin	Type[2]	Description
INT_N	1	O	interrupt output; active LOW 3.3 V tolerant I/O pad
CS_N	2	I	chip select input 3.3 V tolerant I/O pad
WR_N	3	I	write strobe input 3.3 V tolerant I/O pad
RD_N	4	I	read strobe input 3.3 V tolerant I/O pad
DGND	5	-	digital ground supply
XTAL1	6	I	crystal oscillator input (6 MHz); connect a fundamental parallel-resonant crystal or an external clock source (leave pin XTAL2 unconnected)
XTAL2	7	O	crystal oscillator output (6 MHz); connect a fundamental parallel-resonant crystal; leave this pin open when using an external clock source on pin XTAL1
V_{BUS}	8	I	V_{BUS} sensing input and power supply input; see Section 7.11
DM	9	AI/O	USB D– line connection (analog)
DP	10	AI/O	USB D+ line connection (analog)

Table 2. Pin description *...continued*

Symbol[1]	Pin	Type[2]	Description
AGND	11	-	analog ground supply
$V_{REG(3V3)}$	12	-	regulated supply voltage (3.3 V ± 10 %) from the internal regulator; used to connect a 0.1 μF decoupling capacitor and pull-up resistor on pin DP **Remark:** Cannot be used to supply external devices.
VBUSDET_N	13	O	V_{BUS} indicator output (active LOW); see Table 3
DREQ	14	O	DMA request output (4 mA; programmable polarity, see Table 21); signals to the DMA controller that the ISP1183 wants to start a DMA transfer 3.3 V tolerant I/O pad
DACK	15	I	DMA acknowledge input (programmable polarity, see Table 21); used by the DMA controller to signal the start of a DMA transfer requested by the ISP1183; when not in use, connect this pin to ground through a 10 kΩ resistor When the RESET_N pin is LOW, ensure that the DACK and WAKEUP pins are LOW. Otherwise, the device will enter test mode. 3.3 V tolerant I/O pad
RESET_N	16	I	reset input (Schmitt trigger); a LOW level produces an asynchronous reset When the RESET_N pin is LOW, ensure that the DACK and WAKEUP pins are LOW. Otherwise, the device will enter test mode. 3.3 V tolerant I/O pad
A0	17	I	address input; selects command (A0 = HIGH) or data (A0 = LOW) 3.3 V tolerant I/O pad
$V_{DD(I/O)}$	18	-	I/O power supply; add a decoupling capacitor of 0.1 μF (1.65 V to 3.6 V); see Section 7.11
DATA0	19	I/O	data bit 0 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
DATA1	20	I/O	data bit 1 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
VOUT3V3	21	-	3.3 V output voltage; internally connected to the regulator output; connect to a decoupling capacitor of 0.1 μF
DGND	22		digital ground supply
DATA2	23	I/O	data bit 2 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
DATA3	24	I/O	data bit 3 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
DGND	25	-	digital ground supply
DATA4	26	I/O	data bit 4 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
DATA5	27	I/O	data bit 5 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad

Table 2. Pin description *…continued*

Symbol[1]	Pin	Type[2]	Description
DATA6	28	I/O	data bit 6 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
DATA7	29	I/O	data bit 7 input and output bidirectional (4 mA), 3.3 V tolerant I/O pad
$V_{DD(I/O)}$	30	-	I/O power supply; add a decoupling capacitor of 0.1 μF
WAKEUP	31	I	wake-up input (edge triggered, LOW to HIGH); generates a remote wake-up from the suspend state; when not in use, connect this pin to ground through a 10 kΩ resistor When the RESET_N pin is LOW, ensure that the DACK and WAKEUP pins are LOW. Otherwise, the device will enter test mode. 3.3 V tolerant I/O pad
SUSPEND	32	O	suspend state indicator output (4 mA) 3.3 V tolerant I/O pad
GND	exposed die pad	-	ground supply; down bonded to the exposed die pad (heat sink); to be connected to the DGND during the PCB layout

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

[2] I = input; O = output; I/O = input/output; AI/O = analog input/output.

7. Functional description

The ISP1183 is a full-speed USB Peripheral Controller with up to 14 configurable endpoints. It has a fast general-purpose parallel interface for communication with many types of microcontrollers and microprocessors. It supports an 8-bit data bus with separate address and data. The block diagram is given in Figure 1.

The ISP1183 has 2462 bytes of internal FIFO memory that is shared among enabled USB endpoints. The type and FIFO size of each endpoint can individually be configured, depending on the required packet size. Isochronous and bulk endpoints are double-buffered for increased data throughput.

The ISP1183 requires two supply voltages. The core voltage is supplied from V_{BUS} through an internal regulator, which transforms +5.0 V to +3.3 V when V_{BUS} is powered. The I/O interface voltage is supplied from $V_{DD(I/O)}$, which can be 1.65 V to 3.6 V.

The ISP1183 operates on a 6 MHz oscillator frequency.

7.1 Analog transceiver

The transceiver is compliant with *Universal Serial Bus Specification Rev. 2.0*. It directly interfaces to the USB cable through external termination resistors.

7.2 ST-Ericsson SIE

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

7.3 MMU and integrated RAM

The Memory Management Unit (MMU) and the integrated RAM provide the conversion between the USB speed (full-speed: 12 Mbit/s bursts) and the parallel interface to the microcontroller (maximum 11.1 MB/s). This allows the microcontroller to read and write USB packets at its own speed.

7.4 SoftConnect

The connection to USB is accomplished by pulling pin DP (for full-speed USB devices) to HIGH through a 1.5 kΩ pull-up resistor. In the ISP1183, by default, the 1.5 kΩ pull-up resistor is integrated on-chip. The connection is established by a command sent from the external or system microcontroller. This allows the system microcontroller to complete its initialization sequence before deciding to establish connection with USB. Re-initialization of the USB connection can also be performed without disconnecting the cable.

Remark: The tolerance of the internal resistors is 25 %. This is higher than the 5 % tolerance specified by *Universal Serial Bus Specification Rev. 2.0*. The overall voltage specification for the connection, however, can still be met with a good margin. The decision to make use of this feature lies with the USB equipment designer.

7.5 Bit clock recovery

The bit clock recovery circuit recovers the clock from the incoming USB data stream using a 4 × over-sampling principle. It can track jitter and frequency drift as specified in *Universal Serial Bus Specification Rev. 2.0*.

7.6 Voltage regulator

A 5 V-to-3.3 V voltage regulator is integrated on-chip to supply the analog transceiver and internal logic. This voltage is available at pin $V_{REG(3V3)}$ to supply an external 1.5 kΩ pull-up resistor on pin DP. Alternatively, the ISP1183 provides the SoftConnect technology through an integrated 1.5 kΩ pull-up resistor (see Section 7.4).

7.7 PLL clock multiplier

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

7.8 PIO and DMA interfaces

A generic Parallel I/O (PIO) interface is defined for speed and ease-of-use. It also allows direct interfacing to most microcontrollers. To a microcontroller, the ISP1183 appears as a memory device with an 8-bit data bus and a 1-bit address bus. The ISP1183 supports non-multiplexed address and data buses.

The ISP1183 can also be configured as a Direct Memory Access (DMA) slave device to allow more efficient data transfer. One of the 14 endpoint FIFOs may directly transfer data to or from the local shared memory. The DMA interface can independently be configured from the PIO interface.

It can be directly interfaced to microprocessors or microcontrollers with an I/O supply voltage as low as 1.65 V.

7.9 V_{BUS} indicator

The ISP1183 indicates the availability of V_{BUS} using the V_{BUS} pin. When V_{BUS} is available (at pin V_{BUS}), pin VBUSDET_N will output LOW. When V_{BUS} is not available (at pin V_{BUS}), pin VBUSDET_N will output HIGH. Pin VBUSDET_N will change from HIGH-to-LOW level in approximately 2.5 ms to 3.5 ms. See Section 17.

7.10 Operation modes

The ISP1183 can be operated in several operation modes as given in Table 3.

Table 3. ISP1183 operation modes

Pin name	Plug-out state	Dead state	Reset state	Plug-in state	Normal state
V_{BUS}	0 V	X	5 V	5 V	5 V
$V_{DD(I/O)}$	1.8 V	0 V	1.8 V	1.8 V	1.8 V
WAKEUP	X	X	L	L	L
RESET_N	X	X	L	H	H
INT_N	H	L[1]	H	H	-[2]

Table 3. ISP1183 operation modes *...continued*

Pin name	Plug-out state	Dead state	Reset state	Plug-in state	Normal state
SUSPEND	H	L[1]	L	L	L
VBUSDET_N	H	L[1]	L[3]	H → L[4]	L
DATA	Hi-Z	L[1]	Hi-Z	Hi-Z	-

[1] Not driven to LOW. There is, however, no current flow through pads because no I/O supply voltage is available. Therefore, no potential will develop at the output.

[2] During the normal operation, when V_{BUS} is available, pin SUSPEND is LOW. If there is no activity on the USB bus for 3 ms or more, a suspend interrupt is generated on pin INT_N. On receiving the suspend interrupt, the external processor issues a GOSUSP command to the device. Once the GOSUSP command is issued by the processor, the device starts to prepare itself to go to suspend mode. During suspend, to reduce the power consumption, internal clocks can be shut down. Once the device is completely ready to go into suspend mode, it will assert pin SUSPEND to HIGH and go into suspend mode. The typical time between the issuing of the GOSUSP command to the device and the device asserting pin SUSPEND to HIGH is approximately 2 ms.

[3] Independent of the external reset. Depends only on the power-on reset.

[4] On connecting the USB cable (V_{BUS}), pin VBUSDET_N will change from HIGH level to LOW level in approximately 2.5 ms to 3.5 ms.

7.11 Power supply

The ISP1183 is powered from a single supply voltage, ranging from 4.0 V to 5.5 V. An integrated voltage regulator provides a 3.3 V supply voltage for the internal logic and the USB transceiver. This voltage is available at pin $V_{REG(3V3)}$ to connect an external pull-up resistor on USB connection pin DP. See Figure 3.

The ISP1183 can also be operated from a 3.0 V to 3.6 V supply, as shown in Figure 4. In this case, the internal voltage regulator is disabled and pin $V_{REG(3V3)}$ must be connected to V_{BUS}. For details, see Section 17.

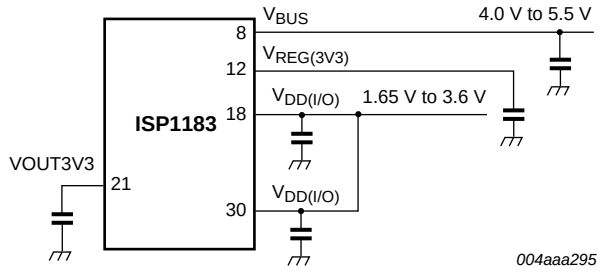


Fig 3. ISP1183 with a 4.0 V to 5.5 V supply

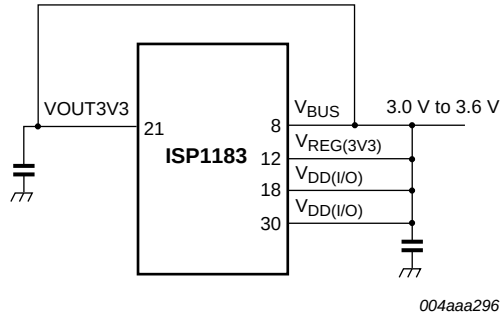


Fig 4. ISP1183 with a 3.0 V to 3.6 V supply

7.12 Crystal oscillator

The ISP1183 has a crystal oscillator designed for a 6 MHz parallel-resonant crystal (fundamental). A typical circuit is shown in Figure 5. Alternatively, an external clock signal of 6 MHz can be applied to input XTAL1, while leaving output XTAL2 open.

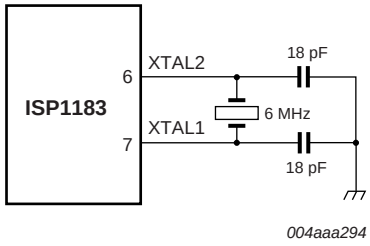


Fig 5. Typical oscillator circuit

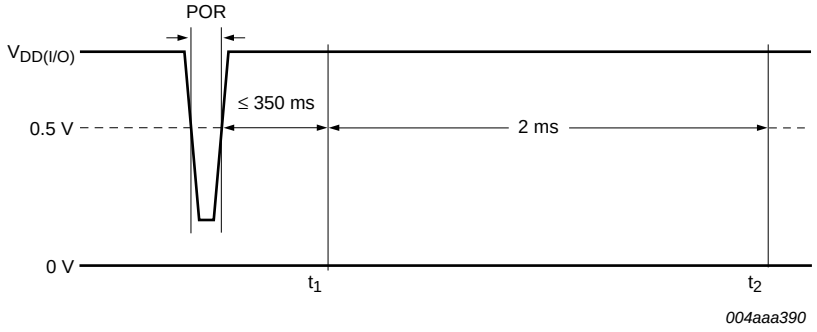
The 6 MHz oscillator frequency is multiplied to 48 MHz by an internal PLL.

In the suspend state, the crystal oscillator and the PLL are switched off to save power. The oscillator operation is controlled by using bit CLKRUN in the Hardware Configuration register. CLKRUN switches the oscillator on and off.

7.13 Power-on reset

The ISP1183 has an internal Power-On Reset (POR) circuit. The clock signal normally requires 3 ms to 4 ms to stabilize.

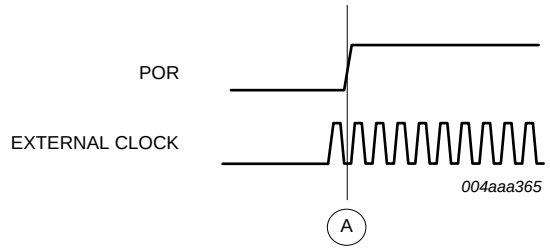
The triggering voltage of the POR circuit is 0.5 V nominal. A POR is automatically generated when $V_{DD(I/O)}$ goes below the trigger voltage for a duration longer than 50 μs.



t_1: clock is running

t_2: registers are accessible

Fig 6. POR timing



Stable external clock available at A.

Fig 7. Clock with respect to the external POR

A hardware reset disables all USB endpoints and clears all Endpoint Configuration Registers (ECRs), except for the control endpoint that is fixed and always enabled. Section 9.3 explains how to initialize and re-initialize endpoints.

8. Interrupts

Figure 8 shows the interrupt logic of the ISP1183. Each of the indicated USB events is logged in a status bit of the Interrupt register. Corresponding bits in the Interrupt Enable register determine whether an event will generate an interrupt.

Interrupts can be masked globally using bit INTENA of the Mode register (see Table 18).

The signaling mode of output INT_N is controlled by bit INTLVL of the Hardware Configuration register (see Table 20). Default settings after reset is level mode. When pulse mode is selected, a pulse of 166 ns is generated when the OR-ed combination of all interrupt bits changes from logic 0 to logic 1.

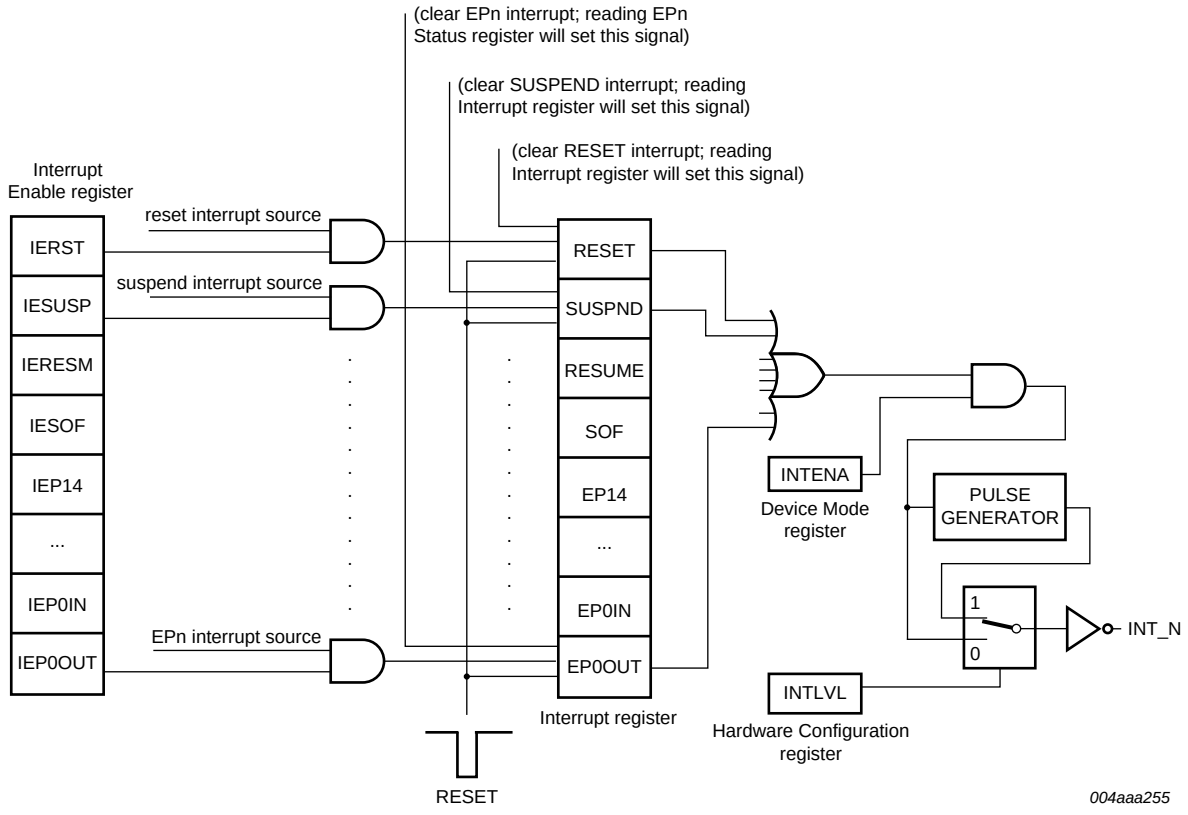


Fig 8. Interrupt logic

Bits SUSPND, RESET, RESUME, SP_EOT, EOT and SOF are cleared when the Interrupt register is read. The endpoint bits (EP0OUT to EP14) are cleared when the associated Endpoint Status register is read.

Bit BUSTATUS follows the USB bus status exactly, allowing firmware to get the current bus status when reading the Interrupt register.

SETUP and OUT token interrupts are generated after the ISP1183 has acknowledged the associated data packet. In bulk transfer mode, the ISP1183 will issue interrupts for every ACK received for an OUT token or transmitted for an IN token.

In isochronous mode, an interrupt is issued on each packet transaction. Firmware is responsible for timing synchronization with the host. This can be done using the Pseudo Start-Of-Frame (PSOF) interrupt, enabled using bit IEPSOF in the Interrupt Enable register. If a Start-Of-Frame (SOF) is lost, PSOF interrupts are generated every 1 ms. This allows firmware to keep data transfer synchronized with the host. After three missed SOF events, the ISP1183 will enter the suspend state.

An alternative way of handling the isochronous data transfer is to enable both the SOF and PSOF interrupts, and disable the interrupt for each isochronous endpoint.

9. Endpoint description

Each USB device is logically composed of several independent endpoints. An endpoint acts as a terminus of a communication flow between the host and the device. At design time, each endpoint is assigned a unique number (endpoint identifier, see Table 4). The combination of the device address (given by the host during enumeration), the endpoint number, and the transfer direction allows each endpoint to be uniquely referenced.

The ISP1183 has 16 endpoints: endpoint 0 (control IN and OUT) plus 14 configurable endpoints, which can individually be defined as interrupt, bulk or isochronous: IN or OUT. Each enabled endpoint has an associated FIFO, which can be accessed either using the PIO or DMA interface.

9.1 Endpoint access

Table 4 lists endpoint access modes and programmability. All endpoints support I/O mode access. Endpoints 1 to 14 also support DMA access. FIFO DMA access is selected and enabled through bits EPDIX[3:0] of the DMA Configuration register and bit DMAEN of the DMA Function and Scratch register. A detailed description of the DMA operation is given in Section 10.

Table 4. Endpoint access and programmability

Endpoint identifier	FIFO size (bytes)[1]	Double buffering	I/O mode access	DMA mode access	Endpoint type
0	64 (fixed)	no	yes	no	control OUT[2]
0	64 (fixed)	no	yes	no	control IN[2]
1 to 14	programmable	supported	supported	supported	programmable

[1] The total amount of FIFO storage allocated to enabled endpoints must not exceed 2462 bytes.

[2] IN: input for the USB host (ISP1183 transmits); OUT: output from the USB host (ISP1183 receives). The data flow direction is determined by bit EPDIR in the Endpoint Configuration register.

9.2 Endpoint FIFO size

The FIFO size determines the maximum packet size that the hardware can support for a given endpoint. Only enabled endpoints are allocated space in the shared FIFO storage, disabled endpoints have zero bytes. Table 5 lists programmable FIFO sizes.

The following bits in the Endpoint Configuration Register (ECR) affect FIFO allocation:

- Endpoint enable bit (FIFOEN)
- Size bits of an enabled endpoint (FFOSZ[3:0])
- Isochronous bit of an enabled endpoint (FFOISO)

Remark: Register changes that affect the allocation of the shared FIFO storage among endpoints must not be made while valid data is present in any FIFO of enabled endpoints. Such changes will render all FIFO contents undefined.

Table 5. Programmable FIFO size

FFOSZ[3:0]	Non-isochronous	Isochronous
0000	8 bytes	16 bytes
0001	16 bytes	32 bytes
0010	32 bytes	48 bytes
0011	64 bytes	64 bytes
0100	reserved	96 bytes
0101	reserved	128 bytes
0110	reserved	160 bytes
0111	reserved	192 bytes
1000	reserved	256 bytes
1001	reserved	320 bytes
1010	reserved	384 bytes
1011	reserved	512 bytes
1100	reserved	640 bytes
1101	reserved	768 bytes
1110	reserved	896 bytes
1111	reserved	1023 bytes

Each programmable FIFO can independently be configured through its ECR. The total physical size of all enabled endpoints (IN plus OUT), however, must not exceed 2462 bytes.

Table 6 shows an example of a configuration fitting in the maximum available space of 2462 bytes. The total number of logical bytes in the example is 1311. The physical storage capacity used for double buffering is managed by the device hardware and is transparent to the user.

Table 6. Memory configuration example

Physical size (bytes)	Logical size (bytes)	Endpoint description
64	64	control IN (64-byte fixed)
64	64	control OUT (64-byte fixed)
2046	1023	double-buffered 1023-byte isochronous endpoint
16	16	16-byte interrupt OUT
16	16	16-byte interrupt IN
128	64	double-buffered 64-byte bulk OUT
128	64	double-buffered 64-byte bulk IN

9.3 Endpoint initialization

In response to standard USB request Set Interface, firmware must program all 16 ECRs of the ISP1183 in sequence (see Table 4), whether endpoints are enabled or not. The hardware will then automatically allocate FIFO storage space.

If all endpoints have successfully been configured, firmware must return an empty packet to the control IN endpoint to acknowledge success to the host. If there are errors in the endpoint configuration, firmware must stall the control IN endpoint.

When reset by hardware or through the USB bus, the ISP1183 disables all endpoints and clears all ECRs, except for the control endpoint, which is fixed and always enabled.

Endpoint initialization can be done at any time. It is, however, valid only after enumeration.

9.4 Endpoint I/O mode access

When an endpoint event occurs (a packet is transmitted or received), the associated endpoint interrupt bits (EPn) of the Interrupt Register (IR) are set by the SIE. Firmware then responds to the interrupt and selects the endpoint for processing.

The endpoint interrupt bit is cleared when the Endpoint Status Register (ESR) is read. The ESR also contains information on the status of the endpoint buffer.

For an OUT (= receive) endpoint, the packet length and the packet data can be read from the ISP1183 by using the Read Buffer command. When the whole packet is read, firmware sends a Clear Buffer command to enable the reception of new packets.

For an IN (= transmit) endpoint, the packet length and data to be sent can be written to the ISP1183 by using the Write Buffer command. When the whole packet is written to the buffer, firmware sends a Validate Buffer command to enable data transmission to the host.

9.5 Special actions on control endpoints

Control endpoints require special firmware actions. The arrival of a SETUP packet flushes the IN buffer, and disables the Validate Buffer and Clear Buffer commands for the control IN and OUT endpoints. The microcontroller must re-enable these commands by sending an Acknowledge Setup command to both control endpoints.

This ensures that the last SETUP packet stays in the buffer and that no packets can be sent back to the host until the microcontroller has explicitly acknowledged that it has seen the SETUP packet.

10. DMA transfer

Direct Memory Access (DMA) is a method to transfer data from one location to another in a computer system, without intervention of the CPU. Many implementations of DMA exist. The ISP1183 supports two methods:

- **8237 compatible mode**: based on the DMA subsystem of the IBM personal computers (PC, AT and all its successors and clones); this architecture uses the Intel 8237 DMA controller and has separate address spaces for memory and I/O
- **DACK-only mode**: based on the DMA implementation in some embedded RISC processors, which has a single address space for both memory and I/O

The ISP1183 supports DMA transfer for all 14 configurable endpoints (see Table 4). Only one endpoint can be selected at a time for DMA transfer. The DMA operation of the ISP1183 can be interleaved with normal I/O mode access to other endpoints.

The following features are supported:

- Single-cycle or burst transfers (up to 16 bytes per cycle)
- Programmable transfer direction (read or write)
- Programmable signal levels on pins DREQ and DACK

10.1 Selecting an endpoint for DMA transfer

The target endpoint for DMA access is selected through bits EPDIX[3:0] in the DMA Configuration register, see Table 7. The transfer direction (read or write) is automatically set by bit EPDIR in the associated ECR, to match the selected endpoint type (OUT endpoint: read; IN endpoint: write).

Asserting input DACK automatically selects the endpoint specified in the DMA Configuration register, regardless of the current endpoint used for the I/O mode access.

Table 7. Endpoint selection for the DMA transfer

Endpoint identifier	EPDIX[3:0]	Transfer direction	
		EPDIR = 0	EPDIR = 1
1	0010	OUT: read	IN: write
2	0011	OUT: read	IN: write
3	0100	OUT: read	IN: write
4	0101	OUT: read	IN: write
5	0110	OUT: read	IN: write
6	0111	OUT: read	IN: write
7	1000	OUT: read	IN: write
8	1001	OUT: read	IN: write
9	1010	OUT: read	IN: write
10	1011	OUT: read	IN: write
11	1100	OUT: read	IN: write
12	1101	OUT: read	IN: write
13	1110	OUT: read	IN: write
14	1111	OUT: read	IN: write

10.2 8237 compatible mode

8237 compatible DMA mode is selected by clearing bit DAKOLY in the Hardware Configuration register (see Table 20). The pin functions for this mode are shown in Table 8.

Table 8. 8237 compatible mode: pin functions

Symbol	Description	I/O	Function
DREQ	DMA request	O	ISP1183 requests a DMA transfer
DACK	DMA acknowledge	I	DMA controller confirms the transfer
RD_N	read strobe	I	instructs the ISP1183 to put data on the bus
WR_N	write strobe	I	instructs the ISP1183 to get data from the bus

The DMA subsystem of an IBM-compatible PC is based on the Intel 8237 DMA controller. It operates as a 'fly-by' DMA controller: data is not stored in the DMA controller, but it is transferred between an I/O port and a memory address. A typical example of the ISP1183 in 8237-compatible DMA mode is given in Figure 9.

The 8237 has two control signals for each DMA channel: DREQ (DMA request) and DACK_N (DMA acknowledge). General control signals are HRQ (hold request) and HLDA (hold acknowledge). The bus operation is controlled using MEMR_N (memory read), MEMW_N (memory write), IOR_N (I/O read) and IOW_N (I/O write).

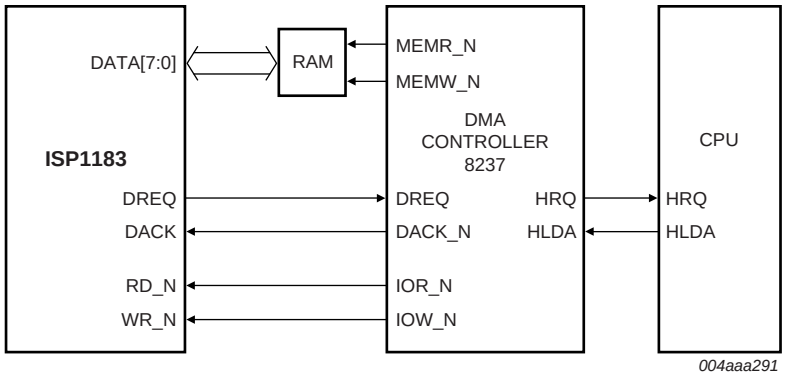


Fig 9. ISP1183 in 8237-compatible DMA mode

The following example shows the steps that occur in a typical DMA transfer:

1. The ISP1183 receives a data packet in one of its endpoint FIFOs. The packet must be transferred to memory address 1234h.
2. The ISP1183 asserts the DREQ signal requesting the 8237 for a DMA transfer.
3. The 8237 asks the CPU to release the bus by asserting the HRQ signal.
4. After completing the current instruction cycle, CPU places bus control signals (MEMR_N, MEMW_N, IOR_N and IOW_N) and address lines in 3-state and asserts HLDA to inform the 8237 that it has control of the bus.
5. The 8237 sets its address lines to 1234h and activates the MEMW_N and IOR_N control signals.
6. The 8237 asserts DACK_N to inform the ISP1183 that it will start a DMA transfer.

7. The ISP1183 places the byte or word to be transferred on data bus lines because its RD_N signal was asserted by the 8237.
8. The 8237 waits one DMA clock period and then de-asserts MEMW_N and IOR_N. This latches and stores the byte or word at the desired memory location. It also informs the ISP1183 that data on bus lines has been transferred.
9. The ISP1183 de-asserts the DREQ signal to indicate to the 8237 that DMA is no longer needed. In single-cycle mode, this is done after each byte or word; in burst mode following the last transferred byte or word of the DMA cycle.
10. The 8237 de-asserts the DACK_N output, indicating that the ISP1183 must stop placing data on the bus.
11. The 8237 places bus control signals (MEMR_N, MEMW_N, IOR_N and IOW_N) and address lines in 3-state and de-asserts the HRQ signal, informing the CPU that it has released the bus.
12. The CPU acknowledges control of the bus by de-asserting HLDA. After activating bus control lines (MEMR_N, MEMW_N, IOR_N and IOW_N) and address lines, the CPU resumes the execution of instructions.

For a typical bulk transfer, the preceding process is repeated 64 times, once for each byte. After each byte, the Address register in the DMA controller is incremented and the byte counter is decremented.

10.3 DACK-only mode

DACK-only DMA mode is selected by setting bit DAKOLY in the Hardware Configuration register (see Table 20). The pin functions for this mode are shown in Table 9. A typical example of the ISP1183 in DACK-only DMA mode is given in Figure 10.

Table 9. DACK-only mode: pin functions

Symbol	Description	I/O	Function
DREQ	DMA request	O	ISP1183 requests a DMA transfer
DACK	DMA acknowledge	I	DMA controller confirms the transfer; also functions as data strobe
RD_N	read strobe	I	not used
WR_N	write strobe	I	not used

In DACK-only mode, the ISP1183 uses the DACK signal as data strobe. Input signals RD_N and WR_N are ignored. This mode is used in CPU systems that have a single address space for memory and I/O access. Such systems have no separate MEMW_N and MEMR_N signals: the RD_N and WR_N signals are also used as memory data strobes.

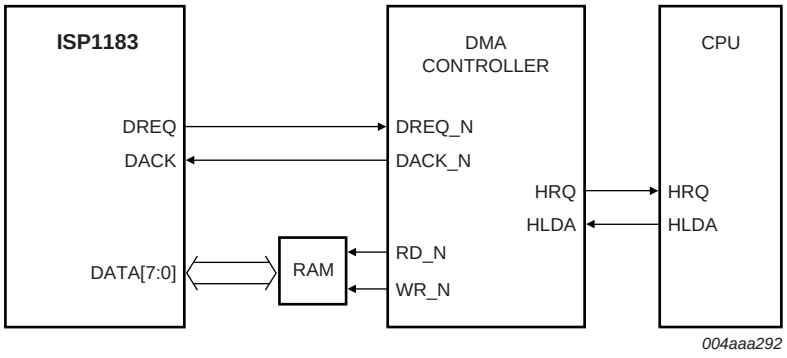


Fig 10. ISP1183 in DACK-only DMA mode

10.4 EOT conditions

10.4.1 Bulk endpoints

A DMA transfer to or from a bulk endpoint can be terminated by any of the following conditions (for bit names, see the DMA Function and Scratch register in Table 30 and the DMA Configuration register in Table 32):

- The DMA transfer completes as programmed in the DMA Counter register (CNTREN = 1).
- A short packet is received on an enabled OUT endpoint (SHORTP = 1).
- DMA operation is disabled by clearing bit DMAEN.

10.4.1.1 DMA Counter register

An End-Of-Transfer (EOT) from the DMA Counter register is enabled by setting bit CNTREN in the DMA Configuration register. The ISP1183 has a 16-bit DMA Counter register, which specifies the number of bytes to be transferred. When DMA is enabled (DMAEN = 1), the internal DMA counter is loaded with the value from the DMA Counter register. When the internal counter completes the transfer as programmed in the DMA counter, an EOT condition is generated and the DMA operation stops.

10.4.1.2 Short packet

Normally, the transfer byte count must be set though a control endpoint before any DMA transfer occurs. When a short packet has been enabled as EOT indicator (SHORTP = 1), the transfer size is determined by the presence of a short packet in data. This mechanism permits the use of a fully autonomous data transfer protocol.

When reading from an OUT endpoint, reception of a short packet at an OUT token will stop the DMA operation after transferring the data bytes of this packet.

Table 10. Summary of EOT conditions for a bulk endpoint

EOT condition	OUT endpoint	IN endpoint
DMA Counter register	transfer completes as programmed in the DMA Counter register	transfer completes as programmed in the DMA Counter register
Short packet	short packet is received and transferred	counter reaches zero in the middle of the buffer
DMAEN bit in the DMA Function and Scratch register	DMAEN = 0[1]	DMAEN = 0[1]

[1] The DMA transfer stops. No interrupt, however, is generated.

10.4.2 Isochronous endpoints

A DMA transfer to or from an isochronous endpoint can be terminated by any of the following conditions (for bit names, see the DMA Function and Scratch register in Table 30 and the DMA Configuration register in Table 32):

- The DMA transfer completes as programmed in the DMA Counter register (CNTREN = 1).
- DMA operation is disabled by clearing bit DMAEN.

Table 11. Recommended EOT usage for isochronous endpoints

EOT condition	OUT endpoint	IN endpoint
DMA Counter register zero	do not use	preferred
Clear DMAEN bit	preferred	do not use

11. Suspend and resume

11.1 Suspend conditions

The ISP1183 detects a USB suspend status when a constant idle state is present on the USB bus for more than 3 ms.

The bus-powered devices that are suspended must not consume more than 500 μA of current. This is achieved by shutting down power to system components or supplying them with a reduced voltage.

The steps leading up to the suspend status are:

1. On detecting a wake-up-to-suspend transition, the ISP1183 sets bit SUSPND in the Interrupt register. This will generate an interrupt if bit IESUSP in the Interrupt Enable register is set.
2. When firmware detects a suspend condition, it must prepare all system components for the suspend state:
   a. All signals connected to the ISP1183 must enter appropriate states to meet the power consumption requirements of the suspend state.
   b. All input pins of the ISP1183 must have a CMOS LOW or HIGH level.
3. In the interrupt service routine, firmware must check the current status of the USB bus. When bit BUSTATUS in the Interrupt register is logic 0, the USB bus has left suspend mode and the process must be aborted. Otherwise, the next step can be executed.
4. To meet suspend current requirements for a bus-powered device, internal clocks must be switched off by clearing bit CLKRUN in the Hardware Configuration register.
5. When firmware has set and cleared bit GOSUSP in the Mode register, the ISP1183 enters the suspend state. In powered-off application, the ISP1183 asserts output SUSPEND and switches off internal clocks after 2 ms.

Figure 11 shows a typical timing diagram.

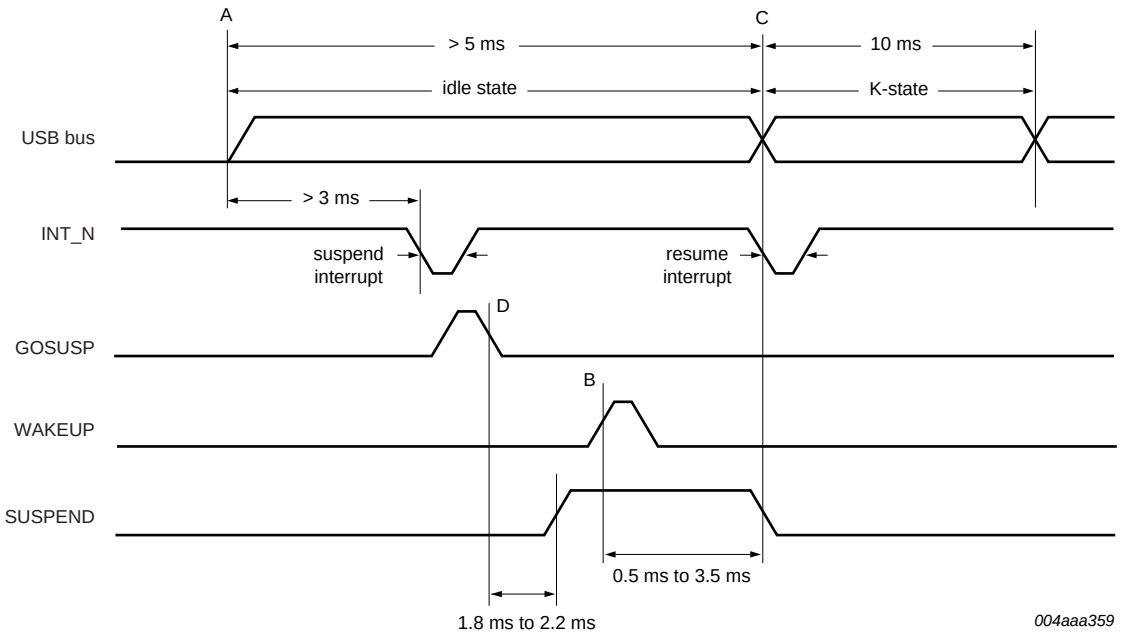


Fig 11. Suspend and resume timing

In Figure 11:

- **A**: indicates the point at which the USB bus enters the idle state.
- **B**: indicates resume condition, which can be a 20 ms K-state on the USB bus, a HIGH level on pin WAKEUP, or a LOW level on pin CS_N.
- **C**: indicates remote wake-up. The ISP1183 will drive a K-state on the USB bus for 10 ms after pin WAKEUP goes HIGH or pin CS_N goes LOW.
- **D**: after detecting the suspend interrupt, set and clear bit GOSUSP in the Mode register.

11.1.1 Powered-off application

Figure 12 shows a typical bus-powered modem application using the ISP1183. The SUSPEND output switches off power to the microcontroller and other external circuits during the suspend state. The ISP1183 is woken up through the USB bus (global resume) or by the ring detection circuit on the telephone line.

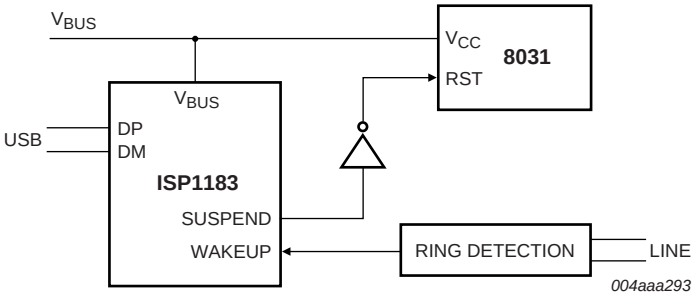


Fig 12. SUSPEND and WAKEUP signals in a powered-off modem application

11.2 Resume conditions

A wake-up from the suspend state is initiated either by the USB host or by the application:

- **USB host**: drives a K-state on the USB bus (global resume).
- **Application**: remote wake-up through a HIGH level on input WAKEUP or a LOW level on input CS_N (if enabled using bit WKUPCS in the Hardware Configuration register). Wake-up on CS_N will work only if V_{BUS} is present.

The steps of a wake-up sequence are:

1. The internal oscillator and the PLL multiplier are re-enabled. When stabilized, clock signals are routed to all internal circuits of the ISP1183.
2. The SUSPEND output is de-asserted, and bit RESUME in the Interrupt register is set. This will generate an interrupt if bit IERESUME in the Interrupt Enable register is set.
3. Maximum 15 ms after starting the wake-up sequence, the ISP1183 resumes its normal functionality.
4. In case of a remote wake-up, the ISP1183 drives a K-state on the USB bus for 10 ms.
5. Following the de-assertion of output SUSPEND, the application restores itself and other system components to normal operating mode.
6. After wake-up, the internal registers of the ISP1183 are write-protected to prevent corruption by inadvertent writing during power-up of external components. Firmware must send an Unlock Device command to the ISP1183 to restore its full functionality. For details, see Section 12.4.2.

11.3 Control bits in suspend and resume

Table 12. Summary of control bits

Register	Bit	Function
Interrupt	SUSPND	a transition from awake to the suspend state was detected
	BUSTATUS	monitors USB bus status (logic 1 = suspend); used when interrupt is serviced
	RESUME	a transition from suspend to the resume state was detected
Interrupt Enable	IESUSP	enables output INT_N to signal the suspend state
	IERESUME	enables output INT_N to signal the resume state
Mode	SOFTCT	enables SoftConnect pull-up resistor to the USB bus
	GOSUSP	a HIGH-to-LOW transition enables the suspend state
Hardware Configuration	EXTPUL	selects internal (SoftConnect) or external pull-up resistor
	WKUPCS	enables wake-up on LOW level of input CS_N
Unlock	all	sending data AA37h unlocks internal registers for writing after a resume

12. Commands and registers

The functions and registers of the ISP1183 are accessed using commands, which consist of a command code, followed by optional data bytes (read or write action). An overview of the available commands and registers is given in Table 13.

A complete access consists of two phases:

1. **Command phase**: when address pin A0 = HIGH, the ISP1183 interprets the data on the lower byte of bus pins D[7:0] as a command code. Commands without a data phase are immediately executed.
2. **Data phase (optional)**: when address pin A0 = LOW, the ISP1183 transfers the data on the bus to or from a register or endpoint FIFO. Multi-byte registers are accessed least significant byte or word first.

Table 13. Command and register overview

Name	Destination	Code	Transaction	Reference
Initialization commands				
Write Control OUT Configuration	Endpoint Configuration register endpoint 0 OUT	20h	write 1 byte	Section 12.1.1 on page 26
Write Control IN Configuration	Endpoint Configuration register endpoint 0 IN	21h	write 1 byte	
Write Endpoint n Configuration (n = 1 to 14)	Endpoint Configuration register endpoints 1 to 14	22h to 2Fh	write 1 byte	
Read Control OUT Configuration	Endpoint Configuration register endpoint 0 OUT	30h	read 1 byte	
Read Control IN Configuration	Endpoint Configuration register endpoint 0 IN	31h	read 1 byte	
Read Endpoint n Configuration (n = 1 to 14)	Endpoint Configuration register endpoints 1 to 14	32h to 3Fh	read 1 byte	
Write or read Device Address	Address register	B6h/B7h	write or read 1 byte	Section 12.1.2 on page 27
Write or read Mode register	Mode register	B8h/B9h	write or read 1 byte	Section 12.1.3 on page 28
Write or read Hardware Configuration	Hardware Configuration register	BAh/BBh	write or read 2 bytes	Section 12.1.4 on page 28
Write or read Interrupt Enable register	Interrupt Enable register	C2h/C3h	write or read 4 bytes	Section 12.1.5 on page 29
Reset Device	resets all registers	F6h	-	Section 12.1.6 on page 31

Table 13. Command and register overview *…continued*

Name	Destination	Code	Transaction	Reference
Data flow commands				
Write Control OUT Buffer	illegal: endpoint is read-only	(00h)	-	Section 12.2.1 on page 31
Write Control IN Buffer	FIFO endpoint 0 IN	01h	N ≤ 64 bytes	
Write Endpoint n Buffer (n = 1 to 14)	FIFO endpoints 1 to 14 (IN endpoints only)	02h to 0Fh	isochronous: N ≤ 1023 bytes interrupt or bulk: N ≤ 64 bytes	
Read Control OUT Buffer	FIFO endpoint 0 OUT	10h	N ≤ 64 bytes	
Read Control IN Buffer	illegal: endpoint is write-only	(11h)	-	
Read Endpoint n Buffer (n = 1 to 14)	FIFO endpoints 1 to 14 (OUT endpoints only)	12h to 1Fh	isochronous: N ≤ 1023 bytes interrupt or bulk: N ≤ 64 bytes	
Stall Control OUT Endpoint	endpoint 0 OUT	40h	-	Section 12.2.3 on page 33
Stall Control IN Endpoint	endpoint 0 IN	41h	-	
Stall Endpoint n (n = 1 to 14)	endpoints 1 to 14	42h to 4Fh	-	
Read Control OUT Status	Endpoint Status register endpoint 0 OUT	50h	read 1 byte	Section 12.2.2 on page 32
Read Control IN Status	Endpoint Status register endpoint 0 IN	51h	read 1 byte	
Read Endpoint n Status (n = 1 to 14)	Endpoint Status register n endpoints 1 to 14	52h to 5Fh	read 1 byte	
Validate Control OUT Buffer	illegal: IN endpoints only[1]	(60h)	-	Section 12.2.4 on page 33
Validate Control IN Buffer	FIFO endpoint 0 IN	61h	-	
Validate Endpoint n Buffer (n = 1 to 14)	FIFO endpoints 1 to 14 (IN endpoints only)[1]	62h to 6Fh	-	
Clear Control OUT Buffer	FIFO endpoint 0 OUT	70h	-	Section 12.2.5 on page 34
Clear Control IN Buffer	illegal[2]	(71h)	-	
Clear Endpoint n Buffer (n = 1 to 14)	FIFO endpoints 1 to 14 (OUT endpoints only)[2]	72h to 7Fh	-	
Unstall Control OUT Endpoint	endpoint 0 OUT	80h	-	Section 12.2.3 on page 33
Unstall Control IN Endpoint	endpoint 0 IN	81h	-	
Unstall Endpoint n (n = 1 to 14)	endpoints 1 to 14	82h to 8Fh	-	
Check Control OUT Status[3]	Endpoint Status Image register endpoint 0 OUT	D0h	read 1 byte	Section 12.2.6 on page 34
Check Control IN Status[3]	Endpoint Status Image register endpoint 0 IN	D1h	read 1 byte	
Check Endpoint n Status (n = 1 to 14)[3]	Endpoint Status Image register n endpoints 1 to 14	D2h to DFh	read 1 byte	
Acknowledge Setup	endpoint 0 IN and OUT	F4h	-	Section 12.2.7 on page 35

Table 13. Command and register overview *…continued*

Name	Destination	Code	Transaction	Reference
DMA commands				
Write or read DMA Function and Scratch register	DMA Function and Scratch register	B2h/B3h	write or read 2 bytes	Section 12.3.1 on page 35
Write or read DMA Configuration	DMA Configuration register	F0h/F1h	write or read 2 bytes	Section 12.3.2 on page 36
Write or read DMA Counter	DMA Counter register	F2h/F3h	write or read 2 bytes	Section 12.3.3 on page 36
General commands				
Read Control OUT Error Code	Error Code register endpoint 0 OUT	A0h	read 1 byte	Section 12.4.1 on page 37
Read Control IN Error Code	Error Code register endpoint 0 IN	A1h	read 1 byte	
Read Endpoint n Error Code (n = 1 to 14)	Error Code register endpoints 1 to 14	A2h to AFh	read 1 byte	
Unlock Device	all registers with write access	B0h	write 2 bytes	Section 12.4.2 on page 38
Read Frame Number	Frame Number register	B4h	read 1 byte or 2 bytes	Section 12.4.3 on page 39
Read Chip ID	Chip ID register	B5h	read 2 bytes	Section 12.4.4 on page 39
Read Interrupt register	Interrupt register	C0h	read 4 bytes	Section 12.4.5 on page 40

[1] Validating an OUT endpoint buffer causes unpredictable behavior of the ISP1183.

[2] Clearing an IN endpoint buffer causes unpredictable behavior of the ISP1183.

[3] Reads a copy of the Status register: executing this command does not clear any status bits or interrupt bits.

12.1 Initialization commands

Initialization commands are used during the enumeration process of the USB network. These commands are used to configure and enable embedded endpoints. They also set the USB assigned address of the ISP1183 and perform device reset.

12.1.1 Endpoint Configuration register (R/W: 30h to 3Fh/20h to 2Fh)

This command accesses the Endpoint Configuration Register (ECR) of the target endpoint. It defines the endpoint type (isochronous or bulk/interrupt), direction (OUT/IN), FIFO size and buffering scheme. It also enables the endpoint FIFO. The register bit allocation is shown in Table 14. A bus reset will disable all endpoints.

The allocation of FIFO memory takes place only after all 16 endpoints have been configured in sequence (from endpoint 0 OUT to endpoint 14). Although control endpoints have fixed configurations, they must be included in the initialization sequence and configured with their default values (see Table 4). Automatic FIFO allocation starts when endpoint 14 is configured.

Remark: If any change is made to an endpoint configuration that affects the allocated memory (size, enable/disable), the FIFO memory contents of all endpoints become invalid. Therefore, all valid data must be removed from enabled endpoints before changing the configuration.

Code: 20h to 2Fh — write (control OUT, control IN, endpoints 1 to 14)

Code: 30h to 3Fh — read (control OUT, control IN, endpoints 1 to 14)

Transaction — write or read 1 byte

Table 14. Endpoint Configuration register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	FIFOEN	EPDIR	DBLBUF	FFOISO	FFOSZ[3:0]			
Reset[1][2]	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

[1] The reset value of the control OUT endpoint is fixed as 83h for the Endpoint Configuration register.

[2] The reset value of the control IN endpoint is fixed as C3h for the Endpoint Configuration register.

Table 15. Endpoint Configuration register: bit description

Bit	Symbol	Description
7	FIFOEN	Logic 1 indicates an enabled FIFO with allocated memory. Logic 0 indicates a disabled FIFO (no bytes allocated).
6	EPDIR	This bit defines the endpoint direction (0 = OUT, 1 = IN). It also determines the DMA transfer direction (0 = read, 1 = write).
5	DBLBUF	Logic 1 indicates that this endpoint has double buffering.
4	FFOISO	Logic 1 indicates an isochronous endpoint. Logic 0 indicates a bulk or interrupt endpoint.
3 to 0	FFOSZ[3:0]	This field specifies the FIFO size according to Table 5.

12.1.2 Address register (R/W: B7h/B6h)

This command sets the USB assigned address in the Address register and enables the USB device. The Address register bit allocation is shown in Table 16.

A USB bus reset sets the device address to 00h (internally) and enables the device. The value of the Address register (accessible by the microcontroller) is not altered by the bus reset. In response to the standard USB request (Set Address), firmware must issue a Write Device Address command, followed by sending an empty packet to the host. The new device address is activated when the host acknowledges the empty packet.

Code: B6h/B7h — write or read Address register

Transaction — write or read 1 byte

Table 16. Address register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	DEVEN	DEVADR[6:0]						
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 17. Address register: bit description

Bit	Symbol	Description
7	DEVEN	Logic 1 enables the device.
6 to 0	DEVADR[6:0]	This field specifies the USB device address.

12.1.3 Mode register (R/W: B9h/B8h)

This command accesses the ISP1183 Mode register, which consists of 1 byte (bit allocation: see Table 18). In 16-bit bus mode, the upper byte is ignored.

The Mode register controls the DMA bus width, resume and suspend modes, interrupt activity and SoftConnect operation. It can be used to enable debug mode, in which all errors and Not Acknowledge (NAK) conditions will generate an interrupt.

Code: B8h/B9h — write or read Mode register

Transaction — write or read 1 byte

Table 18. Mode register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved	reserved	GOSUSP	reserved	INTENA	DBGMOD	reserved	SOFTCT
Reset	0[1]	0	0	0	0[1]	0[1]	0[1]	0[1]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] Unchanged by a bus reset.

Table 19. Mode register: bit description

Bit	Symbol	Description
7	reserved	This bit should be always written as logic 0.
6	-	reserved
5	GOSUSP	Writing logic 1, followed by logic 0 will activate suspend mode.
4	-	reserved
3	INTENA	Logic 1 enables all interrupts. Bus reset value: unchanged.
2	DBGMOD	Logic 1 enables debug mode, in which all NAKs and errors will generate an interrupt. Logic 0 selects normal operation, in which interrupts are generated on every ACK (bulk endpoints) or after every data transfer (isochronous endpoints). Bus reset value: unchanged.
1	-	reserved
0	SOFTCT	Logic 1 enables SoftConnect (see Section 7.4). This bit is ignored if EXTPUL = 1 in the Hardware Configuration register (see Table 20). Bus reset value: unchanged.

12.1.4 Hardware Configuration register (R/W: BBh/BAh)

This command accesses the Hardware Configuration register that consists of 2 bytes. The first (lower) byte contains the device configuration and control values, the second (upper) byte holds clock control bits and the clock division factor. The bit allocation is given in Table 20. A bus reset will not change any of the programmed bit values.

The Hardware Configuration register controls the connection to the USB bus, clock activity and power supply during the suspend state, output clock frequency, DMA operating mode and pin configurations (polarity, signaling mode).

Code: BAh/BBh — write or read Hardware Configuration register

Transaction — write or read 2 bytes

Table 20. Hardware Configuration register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved	EXTPUL	reserved	CLKRUN	reserved			
Reset	0	0	1	0	0	0	1	1
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	DAKOLY	DRQPOL	DAKPOL	reserved	WKUPCS	reserved	INTLVL	reserved
Reset	0	1	0	0	0	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21. Hardware Configuration register: bit description

Bit	Symbol	Description
15	-	reserved
14	EXTPUL	Logic 1 indicates that an external 1.5 kΩ pull-up resistor is used on pin DP and that SoftConnect is not used. Bus reset value: unchanged.
13	-	reserved
12	CLKRUN	Logic 1 indicates that internal clocks are always running, even during the suspend state. Logic 0 switches off the internal oscillator and PLL, when they are not needed. During the suspend state, this bit must be made logic 0 to meet suspend current requirements. The clock is stopped after a delay of approximately 2 ms, following the setting of bit GOSUSP in the Mode register. Bus reset value: unchanged.
11 to 8	-	reserved
7	DAKOLY	Logic 1 selects DACK-only DMA mode. Logic 0 selects 8237 compatible DMA mode. Bus reset value: unchanged.
6	DRQPOL	Selects the DREQ signal polarity (0 = active LOW, 1 = active HIGH). Bus reset value: unchanged.
5	DAKPOL	Selects the DACK signal polarity (0 = active LOW, 1 = active HIGH). Bus reset value: unchanged.
4	reserved	This bit should be always written as logic 0.
3	WKUPCS	Logic 1 enables remote wake-up through a LOW level on input CS_N (For wake-up on CS_N to work, V_{BUS} must be present). Bus reset value: unchanged.
2	-	reserved
1	INTLVL	Selects interrupt signaling mode on output INT_N (0 = level, 1 = pulsed). In pulsed mode, an interrupt produces 166 ns pulse. For details, see Section 11. Bus reset value: unchanged.
0	reserved	This bit should be always written as logic 0.

12.1.5 Interrupt Enable register (R/W: C3h/C2h)

This command individually enables or disables interrupts from all endpoints, as well as interrupts caused by events on the USB bus (SOF, SOF lost, EOT, suspend, resume, reset). A bus reset will not change any of the programmed bit values.

The command accesses the Interrupt Enable register that consists of 4 bytes. The bit allocation is given in Table 22.

Code: C2h/C3h — write or read Interrupt Enable register

Transaction — write or read 4 bytes

Table 22. Interrupt Enable 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	IEP14	IEP13	IEP12	IEP11	IEP10	IEP9	IEP8	IEP7
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	IEP6	IEP5	IEP4	IEP3	IEP2	IEP1	IEP0IN	IEP0OUT
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	SP_IEEOT	IEPSOF	IESOF	IEEOT	IESUSP	IERESM	IERST
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 23. Interrupt Enable register: bit description

Bit	Symbol	Description
31 to 24	-	reserved; must write logic 0
23	IEP14	Logic 1 enables interrupts from endpoint 14.
22	IEP13	Logic 1 enables interrupts from endpoint 13.
21	IEP12	Logic 1 enables interrupts from endpoint 12.
20	IEP11	Logic 1 enables interrupts from endpoint 11.
19	IEP10	Logic 1 enables interrupts from endpoint 10.
18	IEP9	Logic 1 enables interrupts from endpoint 9.
17	IEP8	Logic 1 enables interrupts from endpoint 8.
16	IEP7	Logic 1 enables interrupts from endpoint 7.
15	IEP6	Logic 1 enables interrupts from endpoint 6.
14	IEP5	Logic 1 enables interrupts from endpoint 5.
13	IEP4	Logic 1 enables interrupts from endpoint 4.
12	IEP3	Logic 1 enables interrupts from endpoint 3.
11	IEP2	Logic 1 enables interrupts from endpoint 2.
10	IEP1	Logic 1 enables interrupts from endpoint 1.
9	IEP0IN	Logic 1 enables interrupts from the control IN endpoint.
8	IEP0OUT	Logic 1 enables interrupts from the control OUT endpoint.
7	-	reserved
6	SP_IEEOT	Logic 1 enables interrupt on detection of a short packet.
5	IEPSOF	Logic 1 enables 1 ms interrupts on detection of pseudo SOF.
4	IESOF	Logic 1 enables interrupt on SOF detection.
3	IEEOT	Logic 1 enables interrupt on EOT detection.

Table 23. Interrupt Enable register: bit description …continued

Bit	Symbol	Description
2	IESUSP	Logic 1 enables interrupt on detection of a suspend state.
1	IERESM	Logic 1 enables interrupt on detection of a resume state.
0	IERST	Logic 1 enables interrupt on detection of a bus reset.

12.1.6 Reset Device (F6h)

This command resets the ISP1183 in the same way as an external hardware reset through input RESET_N. All registers are initialized to their reset values.

Code: F6h — reset the device

Transaction — none

12.2 Data flow commands

Data flow commands are used to manage the data transmission between USB endpoints and the system microcontroller. Much of the data flow is initiated through an interrupt to the microcontroller. Data flow commands are used to access endpoints and determine whether endpoint FIFOs contain valid data.

Remark: The IN buffer of an endpoint contains input data for the host. The OUT buffer receives output data from the host.

12.2.1 Endpoint Buffer (R/W: 10h, 12h to 1Fh/01h to 0Fh)

This command accesses endpoint FIFO buffers for reading or writing. First, the buffer pointer is reset to the beginning of the buffer. Following the command, a maximum of (N + 2) bytes can be written or read, N representing the size of the endpoint buffer. After each read or write action, the buffer pointer is automatically incremented by one (8-bit bus width).

In DMA access, the first two bytes (the packet length) are skipped: transfers start at the third byte of the endpoint buffer. When reading, the ISP1183 can detect the last byte through the EOP condition. When writing to a bulk or interrupt endpoint, the endpoint buffer must be completely filled before sending data to the host.

Remark: Reading data after a Write Endpoint Buffer command or writing data after a Read Endpoint Buffer command data will cause unpredictable behavior of the ISP1183.

Code: 01h to 0Fh — write (control IN, endpoints 1 to 14)

Code: 10h, 12h to 1Fh — read (control OUT, endpoints 1 to 14)

Transaction — write or read maximum (N + 2) bytes (isochronous endpoint: $N \leq 1023$, bulk or interrupt endpoint: $N \leq 64$)

The data in the endpoint FIFO must be organized as shown in Table 24. Examples of endpoint FIFO access are given in Table 25.

Table 24. Endpoint FIFO organization

Byte # (8-bit bus)	Description
0	packet length (lower byte)
1	packet length (upper byte)
2	data byte 1
3	data byte 2
:	:
(N + 1)	data byte N

Table 25. Example of endpoint FIFO access

A0	Phase	Bus lines	Byte #	Description
HIGH	command	D[7:0]	-	command code (00h to 1Fh)
LOW	data	D[7:0]	0	packet length (lower byte)
LOW	data	D[7:0]	1	packet length (upper byte)
LOW	data	D[7:0]	2	data byte 1
LOW	data	D[7:0]	3	data byte 2
LOW	data	D[7:0]	4	data byte 3
LOW	data	D[7:0]	5	data byte 4
:	:	:	:	:

Remark: There is no protection against writing or reading past a buffer's boundary, against writing into an OUT buffer, or reading from an IN buffer. Any of these actions could cause an incorrect operation. Data residing in an OUT buffer is meaningful only after a successful transaction. Exception: during DMA access of a double-buffered endpoint, the buffer pointer automatically points to the secondary buffer after reaching the end of the primary buffer.

12.2.2 Endpoint Status register (R: 50h to 5Fh)

This command reads the status of an endpoint FIFO. The command accesses the Endpoint Status register, the bit allocation of which is shown in Table 26. Reading the Endpoint Status register will clear the interrupt bit set for the corresponding endpoint in the Interrupt register (see Table 46).

All bits of the Endpoint Status register are read-only. Bit EPSTAL is controlled by the Stall or Unstall commands and by the reception of a SETUP token (see Section 12.2.3).

Code: 50h to 5Fh — read (control OUT, control IN, endpoints 1 to 14)

Transaction — read 1 byte

Table 26. Endpoint Status register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	EPSTAL	EPFULL1	EPFULL0	DATA_PID	OVER WRITE	SETUPT	CPUBUF	reserved
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 27. Endpoint Status register: bit description

Bit	Symbol	Description
7	EPSTAL	This bit indicates whether the endpoint is stalled or not (1 = stalled, 0 = not stalled). Set by a Stall Endpoint command. Cleared by an Unstall Endpoint command. The endpoint is automatically unstalled on receiving a SETUP token.
6	EPFULL1	Logic 1 indicates that the secondary endpoint buffer is full.
5	EPFULL0	Logic 1 indicates that the primary endpoint buffer is full.
4	DATA_PID	This bit indicates the data PID of the next packet (0 = DATA0 PID, 1 = DATA1 PID).
3	OVERWRITE	This bit is set by hardware. Logic 1 indicates that a new set-up packet has overwritten the previous set-up information, before it was acknowledged or before the endpoint was stalled. This bit is cleared by reading, if writing the set-up data has finished. Firmware must check this bit before sending an Acknowledge Setup command or stalling the endpoint. On reading logic 1, firmware must stop ongoing set-up actions and wait for a new set-up packet.
2	SETUPT	Logic 1 indicates that the buffer contains a set-up packet.
1	CPUBUF	This bit indicates which buffer is currently selected for CPU access (0 = primary buffer, 1 = secondary buffer).
0	-	reserved

12.2.3 Stall or Unstall Endpoint (40h to 4Fh/80h to 8Fh)

These commands are used to stall or unstall an endpoint. The commands modify the content of the Endpoint Status register (see Table 26).

A stalled control endpoint is automatically unstalled when it receives a SETUP token, regardless of the packet content. If the endpoint should stay in its stalled state, the microcontroller can re-stall it with the Stall Endpoint command.

When a stalled endpoint is unstalled (either by the Unstall Endpoint command or by receiving a SETUP token), it is also re-initialized. This flushes the buffer: if it is an OUT buffer, it waits for a DATA0 PID; if it is an IN buffer, it writes a DATA0 PID.

Code: 40h to 4Fh — stall (control OUT, control IN, endpoints 1 to 14)

Code: 80h to 8Fh — unstall (control OUT, control IN, endpoints 1 to 14)

Transaction — none

Remark: When unstalling a stalled endpoint, issue the unstall command two times. The first unstall command will update the Endpoint Status register in RAM. The second unstall command will reset buffer pointers.

12.2.4 Validate Endpoint Buffer (61h to 6Fh)

This command signals the presence of valid data for transmission to the USB host, by setting the Buffer Full flag of the selected IN endpoint. This indicates that the data in the buffer is valid and can be sent to the host, when the next IN token is received. For a double-buffered endpoint, this command switches the current FIFO for CPU access.

Remark: For special aspects of the control IN endpoint, see Section 9.5.

Code: 61h to 6Fh — validate endpoint buffer (control IN, endpoints 1 to 14)

Transaction — none

12.2.5 Clear Endpoint Buffer (70h, 72h to 7Fh)

This command unlocks and clears the buffer of the selected OUT endpoint, allowing the reception of new packets. Reception of a complete packet causes the Buffer Full flag of an OUT endpoint to be set. Any subsequent packets are refused by returning a NAK condition, until the buffer is unlocked using this command. For a double-buffered endpoint, this command switches the current FIFO for CPU access.

Remark: For special aspects of the control OUT endpoint, see Section 9.5.

Code: 70h, 72h to 7Fh — clear endpoint buffer (control OUT, endpoints 1 to 14)

Transaction — none

12.2.6 Check Endpoint Status (D0h to DFh)

This command checks the status of the selected endpoint FIFO without clearing any status or interrupt bits. The command accesses the Endpoint Status Image register, which contains a copy of the Endpoint Status register. The bit allocation of the Endpoint Status Image register is shown in Table 28.

Code: D0h to DFh — check status (control OUT, control IN, endpoints 1 to 14)

Transaction — write or read 1 byte

Table 28. Endpoint Status Image register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	EPSTAL	EPFULL1	EPFULL0	DATA_PID	OVER WRITE	SETUPT	CPUBUF	reserved
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 29. Endpoint Status Image register: bit description

Bit	Symbol	Description
7	EPSTAL	This bit indicates whether the endpoint is stalled or not (1 = stalled, 0 = not stalled).
6	EPFULL1	Logic 1 indicates that the secondary endpoint buffer is full.
5	EPFULL0	Logic 1 indicates that the primary endpoint buffer is full.
4	DATA_PID	This bit indicates the data PID of the next packet (0 = DATA0 PID, 1 = DATA1 PID).
3	OVER WRITE	This bit is set by hardware. Logic 1 indicates that a new set-up packet has overwritten the previous set-up information, before it was acknowledged or before the endpoint was stalled. This bit is cleared by reading, if writing the set-up data has finished. Firmware must check this bit before sending an Acknowledge Setup command or stalling the endpoint. On reading logic 1, firmware must stop ongoing set-up actions and wait for a new set-up packet.

Table 29. Endpoint Status Image register: bit description …continued

Bit	Symbol	Description
2	SETUPT	Logic 1 indicates that the buffer contains a set-up packet.
1	CPUBUF	This bit indicates which buffer is currently selected for CPU access (0 = primary buffer, 1 = secondary buffer).
0	-	reserved

12.2.7 Acknowledge Setup (F4h)

This command acknowledges to the host that a SETUP packet was received. The arrival of a SETUP packet disables the Validate Buffer and Clear Buffer commands for the control IN and OUT endpoints. The microcontroller needs to re-enable these commands by sending an Acknowledge Setup command, see Section 9.5.

Code: F4h — acknowledge setup

Transaction — none

12.3 DMA commands

12.3.1 DMA Function and Scratch register (R/W: B3h/B2h)

This command accesses the 16-bit DMA Function and Scratch register, which can be used by firmware to save and restore information. For example, the device status before powering down in the suspend state. The register bit allocation is given in Table 30.

Code: B2h/B3h — write or read DMA Function and Scratch register

Transaction — write or read 2 bytes

Table 30. DMA Function and Scratch register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	DMAEN	reserved		SFIRH[4:0]				
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	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	SFIRL[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 31. DMA Function and Scratch register: bit description

Bit	Symbol	Description
15	DMAEN	Writing logic 1 enables DMA function.
14 to 13	-	reserved; must be logic 0
12 to 8	SFIRH[4:0]	Scratch Information register (high byte)
7 to 0	SFIRL[7:0]	Scratch Information register (low byte)

12.3.2 DMA Configuration register (R/W: F1h/F0h)

This command defines the DMA configuration of the ISP1183 and enables or disables DMA transfers. The command accesses the DMA Configuration register, which consists of 2 bytes. The bit allocation is given in Table 32. A bus reset will clear bit DMAEN (DMA disabled), all other bits remain unchanged.

Code: F0h/F1h — write or read DMA Configuration

Transaction — write or read 2 bytes

Table 32. DMA Configuration register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	CNTREN	SHORTP	reserved					
Reset	0[1]	0[1]	0[1]	0[1]	0[1]	0[1]	0[1]	0[1]
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	EPDIX[3:0]				DMA START	reserved	BURSTL[1:0]	
Reset	0[1]	0[1]	0[1]	0[1]	0	0	0[1]	0[1]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] Unchanged by a bus reset.

Table 33. DMA Configuration register: bit description

Bit	Symbol	Description
15	CNTREN	Logic 1 enables the generation of an EOT condition, when the DMA Counter register reaches zero. Bus reset value: unchanged.
14	SHORTP	Logic 1 enables short or empty packet mode. When receiving (OUT endpoint) a short or empty packet, an EOT condition is generated. When transmitting (IN endpoint), this bit should be cleared. Bus reset value: unchanged.
13 to 8	-	reserved
7 to 4	EPDIX[3:0]	Indicates the destination endpoint for DMA, see Table 7.
3	DMASTART	Writing logic 1 starts a DMA transfer. Logic 0 forces the end of an ongoing DMA transfer. Reading this bit indicates whether DMA is started (0 = DMA stopped, 1 = DMA started). This bit is cleared by a bus reset.
2	-	reserved
1 to 0	BURSTL[1:0]	Selects the DMA burst length: **00 —** single-cycle mode (1 byte) **01 —** burst mode (4 bytes) **10 —** burst mode (8 bytes) **11 —** burst mode (16 bytes) Bus reset value: unchanged.

12.3.3 DMA Counter register (R/W: F3h/F2h)

This command accesses the DMA Counter register, which consists of 2 bytes. The bit allocation is given in Table 34. Writing to the register sets the number of bytes for a DMA transfer. Reading the register returns the number of remaining bytes in the current transfer. A bus reset will not change programmed bit values.

The internal DMA counter is automatically reloaded from the DMA Counter register when DMA is re-enabled (DMAEN = 1). For details, see Section 12.3.2.

Code: F2h/F3h — write or read DMA Counter register

Transaction — write or read 2 bytes

Table 34. DMA Counter register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	DMACRH[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	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	DMACRL[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 35. DMA Counter register: bit description

Bit	Symbol	Description
15 to 8	DMACRH[7:0]	DMA Counter register (high byte)
7 to 0	DMACRL[7:0]	DMA Counter register (low byte)

12.4 General commands

12.4.1 Endpoint Error Code (R: A0h to AFh)

This command returns the status of the last transaction of the selected endpoint, as stored in the Error Code register. Each new transaction overwrites the previous status information. The bit allocation of the Error Code register is shown in Table 36.

Code: A0h to AFh — read error code (control OUT, control IN, endpoints 1 to 14)

Transaction — read 1 byte

Table 36. Error Code register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	UNREAD	DATA01	reserved	ERROR[3:0]				RTOK
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 37. Error Code register: bit description

Bit	Symbol	Description
7	UNREAD	Logic 1 indicates that a new event occurred before the previous status was read.
6	DATA01	This bit indicates the PID type of the last successfully received or transmitted packet (0 = DATA0 PID, 1 = DATA1 PID).
5	-	reserved
4 to 1	ERROR[3:0]	Error code. For error description, see Table 38.
0	RTOK	Logic 1 indicates that data was successfully received or transmitted.

Table 38. Transaction error codes

Error code (binary)	Description
0000	no error
0001	PID encoding error; bits 7 to 4 are not the inverse of bits 3 to 0
0010	PID unknown; encoding is valid, but PID does not exist
0011	unexpected packet; packet is not of the expected type (token, data, or acknowledge), or is a SETUP token to a non-control endpoint
0100	token CRC error
0101	data CRC error
0110	time-out error
0111	babble error
1000	unexpected end-of-packet
1001	sent or received NAK (Not AcKnowledge)
1010	sent stall; a token was received, but the endpoint was stalled
1011	overflow; the received packet was larger than the available buffer space
1100	sent empty packet (ISO only)
1101	bit stuffing error
1110	sync error
1111	wrong (unexpected) toggle bit in DATA PID; data was ignored

12.4.2 Unlock Device (B0h)

This command unlocks the ISP1183 from write-protection mode after a resume. In the suspend state, all registers and FIFOs are write-protected to prevent data corruption by external devices during a resume. Also, the register access for reading is possible only after the Unlock Device command is executed.

After waking up from the suspend state, firmware must unlock registers and FIFOs using this command, by writing the unlock code (AA37h) into the Lock register (8-bit bus: lower byte first). The bit allocation of the Lock register is given in Table 39.

Code: B0h — unlock the device

Transaction — write 2 bytes (unlock code)

Table 39. Lock register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	UNLOCKH[7:0] = AAh							
Reset	1	0	1	0	1	0	1	0
Access	W	W	W	W	W	W	W	W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	UNLOCKL[7:0] = 37h							
Reset	0	0	1	1	0	1	1	1
Access	W	W	W	W	W	W	W	W

Table 40. Lock register: bit description

Bit	Symbol	Description
15 to 0	UNLOCK[15:0]	Sending data AA37h unlocks internal registers and FIFOs for writing, following a resume.

12.4.3 Frame Number register (R: B4h)

This command returns the frame number of the last successfully received SOF. It is followed by reading one or two bytes from the Frame Number register, containing the frame number (lower byte first). The Frame Number register is shown in Table 41.

Remark: After a bus reset, the value of the Frame Number register is undefined.

Code: B4h — read frame number

Transaction — read 1 byte or 2 bytes

Table 41. Frame Number register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved					SOFRH[2:0]		
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	SOFRL[7:0]							
Reset[1]	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

[1] Reset value undefined after a bus reset.

Table 42. Frame Number register: bit description

Bit	Symbol	Description
15 to 11	-	reserved
10 to 8	SOFRH[2:0]	SOF frame number (upper byte)
7 to 0	SOFRL[7:0]	SOF frame number (lower byte)

Table 43. Example of Frame Number register access

A0	Phase	Bus lines	Byte #	Description
HIGH	command	D[7:0]	-	command code (B4h)
LOW	data	D[7:0]	0	frame number (lower byte)
LOW	data	D[7:0]	1	frame number (upper byte)

12.4.4 Chip ID register (R: B5h)

This command reads the chip identification code and hardware version number. Firmware must check this information to determine supported functions and features. This command accesses the Chip ID register, which is shown in Table 44.

Code: B5h — read chip ID

Transaction — read 2 bytes

Table 44. Chip ID register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	CHIPIDH[7:0]							
Reset	82h							
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	CHIPIDL[7:0]							
Reset	11h							
Access	R	R	R	R	R	R	R	R

Table 45. Chip ID register: bit description

Bit	Symbol	Description
15 to 8	CHIPIDH[7:0]	chip ID code (82h)
7 to 0	CHIPIDL[7:0]	silicon version (11h)

12.4.5 Interrupt register (R: C0h)

This command indicates the sources of interrupts as stored in the 4-byte Interrupt register. Each individual endpoint has its own interrupt bit. The bit allocation of the Interrupt register is shown in Table 46. Bit BUSTATUS verifies the current bus status in the interrupt service routine. Interrupts are enabled through the Interrupt Enable register, see Section 12.1.5.

While reading the interrupt register, read all the 4 bytes completely.

Code: C0h — read Interrupt register

Transaction — read 4 bytes

Table 46. Interrupt 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	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP7
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	EP6	EP5	EP4	EP3	EP2	EP1	EP0IN	EP0OUT
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	BUSTATUS	SP_EOT	PSOF	SOF	EOT	SUSPND	RESUME	RESET
Reset	0[1]	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

[1] The reset value of this bit depends on the current USB bus status. If the bus is idle, the reset value will be 1.

Table 47. Interrupt register: bit description

Bit	Symbol	Description
31 to 24	-	reserved
23	EP14	Logic 1 indicates the interrupt source: endpoint 14.
22	EP13	Logic 1 indicates the interrupt source: endpoint 13.
21	EP12	Logic 1 indicates the interrupt source: endpoint 12.
20	EP11	Logic 1 indicates the interrupt source: endpoint 11.
19	EP10	Logic 1 indicates the interrupt source: endpoint 10.
18	EP9	Logic 1 indicates the interrupt source: endpoint 9.
17	EP8	Logic 1 indicates the interrupt source: endpoint 8.
16	EP7	Logic 1 indicates the interrupt source: endpoint 7.
15	EP6	Logic 1 indicates the interrupt source: endpoint 6.
14	EP5	Logic 1 indicates the interrupt source: endpoint 5.
13	EP4	Logic 1 indicates the interrupt source: endpoint 4.
12	EP3	Logic 1 indicates the interrupt source: endpoint 3.
11	EP2	Logic 1 indicates the interrupt source: endpoint 2.
10	EP1	Logic 1 indicates the interrupt source: endpoint 1.
9	EP0IN	Logic 1 indicates the interrupt source: control IN endpoint.
8	EP0OUT	Logic 1 indicates the interrupt source: control OUT endpoint.
7	BUSTATUS	It monitors the current USB bus status (0 = awake, 1 = suspend).
6	SP_EOT	Logic 1 indicates that an EOT interrupt has occurred for a short packet.
5	PSOF	Logic 1 indicates that an interrupt is issued every 1 ms because of the pseudo SOF; after three missed SOFs, the suspend state is entered.
4	SOF	Logic 1 indicates that a SOF condition was detected.
3	EOT	Logic 1 indicates that an internal EOT condition was generated by the DMA counter reaching zero.
2	SUSPND	Logic 1 indicates that an awake to suspend change of state was detected on the USB bus.
1	RESUME	Logic 1 indicates that a resume state was detected.
0	RESET	Logic 1 indicates that a bus reset condition was detected.

13. Limiting values

Table 48. Limiting values

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

Symbol	Parameter	Conditions		Min	Max	Unit
V_{BUS}	bus supply voltage			–0.5	+6.0	V
$V_{DD(I/O)}$	I/O supply voltage			–0.5	+4.6	V
V_I	input voltage	digital		–0.5	$V_{DD(I/O)}$ + 0.5	V
I_{lu}	latch-up current	V_I < 0 V or V_I > V_{BUS}		-	100	mA
V_{esd}	electrostatic discharge voltage	I_{LI} < 1 μA	[1]	–2000	+2000	V
T_{stg}	storage temperature			–60	+150	°C
P_{tot}	total power dissipation	V_{BUS} = 5.5 V		-	100	mW

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

14. Recommended operating conditions

Table 49. Recommended operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{BUS}	bus supply voltage	with regulator	4.0	5.0	5.5	V
$V_{DD(I/O)}$	I/O supply voltage		1.65	-	3.6	V
V_I	input voltage		0	-	$V_{DD(I/O)}$	V
$V_{O(I/O)}$	output I/O voltage		0	-	$V_{DD(I/O)}$	V
$V_{I(AI/O)}$	input voltage on analog I/O pins DP and DM		0	-	3.6	V
$V_{O(od)}$	open-drain output pull-up voltage		0	-	V_{BUS}	V
T_{amb}	ambient temperature		–40	-	+85	°C

15. Static characteristics

Table 50. Static characteristics: supply pins

V_{BUS} = 4.0 V to 5.5 V; $V_{DD(I/O)}$ = 1.65 V to 3.6 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$V_{REG(3V3)}$	3.3 volt regulator voltage	V_{BUS} = 4.0 V to 5.5 V	[1][2]	3.0	3.3	3.6	V
I_{CC}	supply current	V_{BUS} = 5.0 V; T_{amb} = 25 °C		-	19	-	mA
$I_{CC(susp)}$	suspend supply current	V_{BUS} = 5.0 V; T_{amb} = 25 °C	[3]	-	-	250	µA
$I_{ref(static)}$	$V_{DD(I/O)}$ static I/O supply current	suspend or no V_{BUS}		-	-	10	µA
I_{ref}	$V_{DD(I/O)}$ operating I/O supply current			-	-	3.5	mA

[1] For 3.3 V operation, pin $V_{REG(3V3)}$ must be connected to pin V_{BUS}.

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

[3] External loading is not included.

Table 51. Static characteristics: digital pins

V_{BUS} = 4.0 V to 5.5 V; $V_{DD(I/O)}$ = 1.65 V to 3.6 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IL(I/O)}$	LOW-level input voltage		-	-	$0.2V_{DD(I/O)}$	V
$V_{IH(I/O)}$	HIGH-level input voltage		$0.7V_{DD(I/O)}$	-	-	V
V_{OL}	LOW-level output voltage		-	-	$0.22V_{DD(I/O)}$	V
V_{OH}	HIGH-level output voltage		$0.8V_{DD(I/O)}$	-	-	V
I_{LI}	input leakage current		−1	-	+1	µA
C_i	input capacitance		-	-	10	pF
Z_i	input impedance		2	-	-	MΩ

Table 52. Static characteristics: analog I/O pins DP and DM

V_{BUS} = 4.0 V to 5.5 V; $V_{DD(I/O)}$ = 1.65 V to 3.6 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.[1]

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Input levels							
V_{DI}	differential input sensitivity	$\|V_{I(DP)} - V_{I(DM)}\|$		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 ground		2.8	-	3.6	V
Leakage current							
I_{LZ}	OFF-state leakage current			−10	-	+10	μA
Capacitance							
C_{IN}	transceiver capacitance	pin to ground		-	-	20	pF
Resistance							
R_{PU}	pull-up resistance	SoftConnect = on	[2]	1	-	2	kΩ
Z_{DRV}	driver output impedance	steady-state drive	[3]	29	-	44	Ω
Z_{INP}	input impedance			10	-	-	MΩ
Termination							
V_{TERM}	termination voltage for upstream facing port pull-up	R_{PU}	[4][5]	3.0	-	3.6	V

[1] DP is the USB positive data pin; DM is the USB negative data pin.

[2] Pull-up resistance on DP.

[3] Includes external resistors of 22 Ω ± 1 % on both DP and DM.

[4] This voltage is available at pin $V_{REG(3V3)}$.

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

16. Dynamic characteristics

Table 53. Dynamic characteristics

V_{BUS} = 4.0 V to 5.5 V; $V_{DD(I/O)}$ = 1.65 V to 3.6 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Reset							
$t_{W(RESET\_N)}$	external RESET_N pulse width	crystal oscillator running		50	-	-	µs
		crystal oscillator stopped	[1]	-	3	-	ms
Crystal oscillator							
f_{XTAL1}	frequency on pin XTAL1			-	6	-	MHz

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

Table 54. Dynamic characteristics: analog I/O pins DP and DM

V_{BUS} = 4.0 V to 5.5 V; $V_{DD(I/O)}$ = 1.65 V to 3.6 V; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; C_L = 50 pF; R_{PU} = 1.5 kΩ on DP to V_{TERM}; unless otherwise specified.[1]

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 time/fall time matching	t_{FR}/t_{FF}	[2]	90	-	111.11	%
V_{CRS}	output signal crossover voltage		[2][3]	1.3	-	2.0	V
Data source timing							
t_{FEOPT}	source SE0 interval of EOP		[3]	160	-	175	ns
t_{FDEOP}	source jitter for differential transition to SE0 transition		[3]	−2	-	+5	ns
Receiver timing							
t_{JR1}	receiver jitter to next transition		[3]	−18.5	-	+18.5	ns
t_{JR2}	receiver jitter for paired transitions		[3]	−9	-	+9	ns
t_{FEOPR}	receiver SE0 interval of EOP	accepted as EOP	[3]	82	-	-	ns
t_{FST}	width of SE0 interval during differential transition	rejected as EOP	[3]	-	-	14	ns

[1] Test circuit: see Figure 27.

[2] Excluding the first transition from Idle state.

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

16.1 Timing

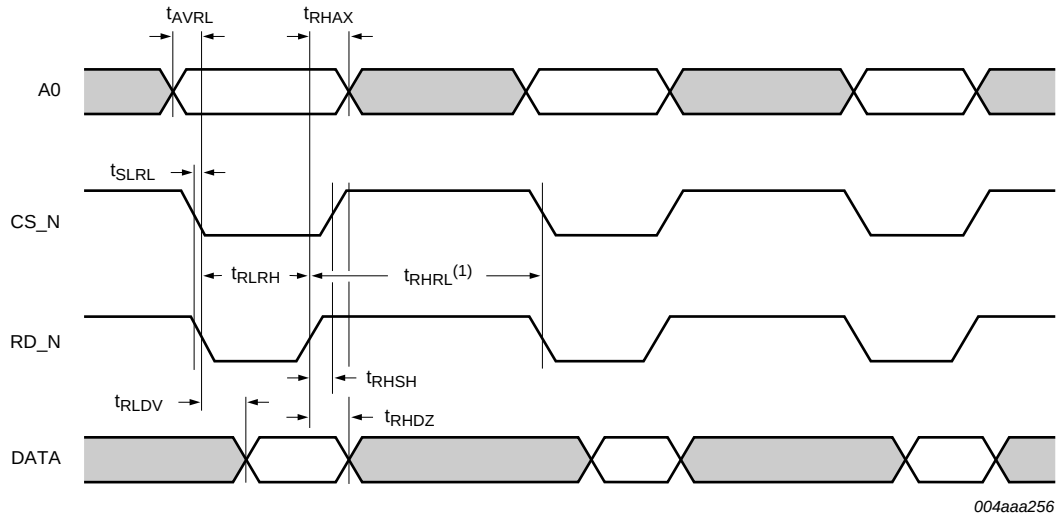
16.1.1 Parallel I/O timing

Table 55. Dynamic characteristics: parallel interface timing

V_{BUS} = $V_{REG(3V3)}$ = 2.7 V to 3.9 V; $V_{DD(I/O)}$ = 1.8 V; unless otherwise specified.

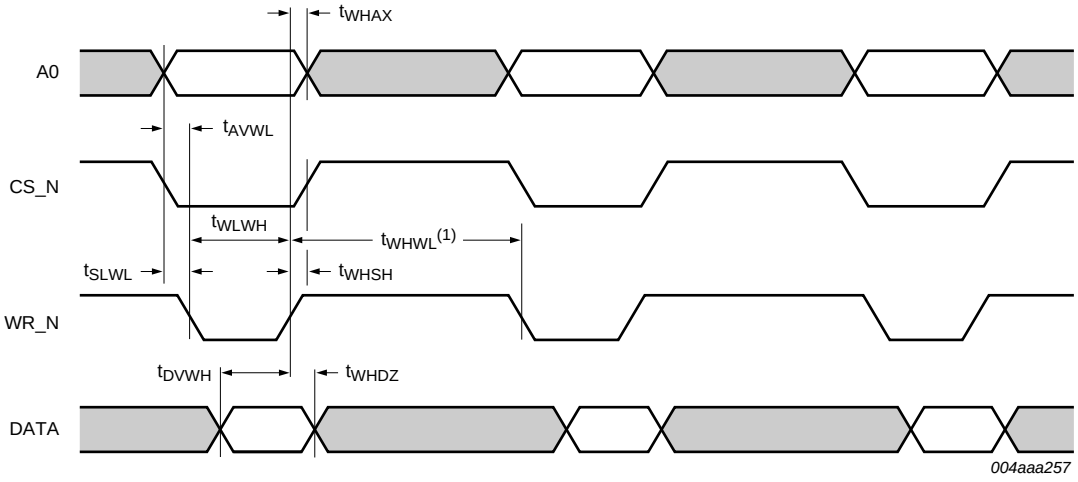
Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Read timing (see Figure 13)						
t_{RHAX}	address hold time after RD_N HIGH	C_L = 30 pF	0	-	-	ns
t_{AVRL}	address setup time before RD_N LOW		0	-	-	ns
t_{RHDZ}	data outputs high-impedance time after RD_N HIGH		0	-	-	ns
t_{RHSH}	chip deselect time after RD_N HIGH		−2	-	-	ns
t_{RHRL}	RD_N LOW after RD_N HIGH		65	-	-	ns
t_{RLRH}	RD_N pulse width		25	-	-	ns
t_{SLRL}	CS_N time before RD_N LOW		0	-	-	ns
t_{RLDV}	data valid time after RD_N LOW		-	-	20	ns
t_{RC} (t_{RHRL} + t_{RLRH})	read cycle time		90	-	-	ns
Write timing (see Figure 14)						
t_{WHAX}	address hold time after WR_N HIGH		1	-	-	ns
t_{AVWL}	address setup time before WR_N LOW		0	-	-	ns
t_{SLWL}	CS_N time before WR_N LOW		0	-	-	ns
t_{WL} (t_{WHWL} + t_{WLWH})	write cycle time	[1]	90/180	-	-	ns
t_{WLWH}	WR_N pulse width		22	-	-	ns
t_{WHWL}	WR_N LOW after WR_N HIGH	[1]	68/158	-	-	ns
t_{WHSH}	chip deselect time after WR_N HIGH		0	-	-	ns
t_{DVWH}	data setup time before WR_N HIGH		2	-	-	ns
t_{WHDZ}	data hold time after WR_N HIGH		1	-	-	ns

[1] The minimum value for data flow commands (see Table 13) is 180 ns.



(1) If required, CS_N can be kept permanently asserted. There is no need to de-assert and assert in between the read and write cycles.

Fig 13. Parallel interface read timing



(1) If required, CS_N can be kept permanently asserted. There is no need to de-assert and assert in between the read and write cycles.

Fig 14. Parallel interface write timing

16.1.2 Access cycle timing

Table 56. Dynamic characteristics: access cycle timing

V_{BUS} = $V_{REG(3V3)}$ = 2.7 V to 3.9 V; $V_{DD(I/O)}$ = 1.8 V; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Write command + write data (see Figure 15 and Figure 16)							
$T_{cy(WC-WD)}$	cycle time for write command, then write data	C_L = 30 pF	[1]	100	-	-	ns
$T_{cy(WD-WD)}$	cycle time for write data			90	-	-	ns

Table 56. Dynamic characteristics: access cycle timing *...continued*

V_{BUS} = $V_{REG(3V3)}$ = 2.7 V to 3.9 V; $V_{DD(I/O)}$ = 1.8 V; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$T_{cy(WD-WC)}$	cycle time for write data, then write command		90	-	-	ns
Write command + read data (see Figure 17 and Figure 18)						
$T_{cy(WC-RD)}$	cycle time for write command, then read data	[1]	100	-	-	ns
$T_{cy(RD-RD)}$	cycle time for read data		90	-	-	ns
$T_{cy(RD-WC)}$	cycle time for read data, then write command		90	-	-	ns

[1] The minimum value for data flow commands (see Table 13) is 180 ns.

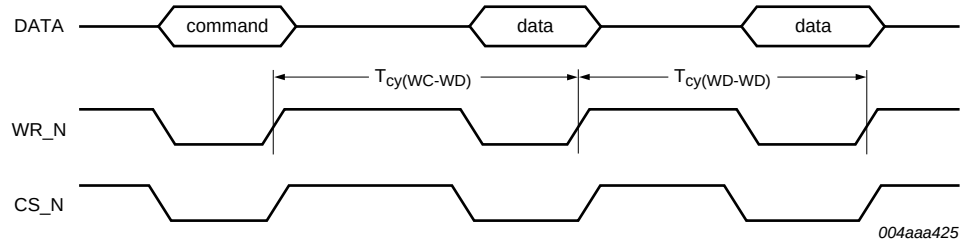
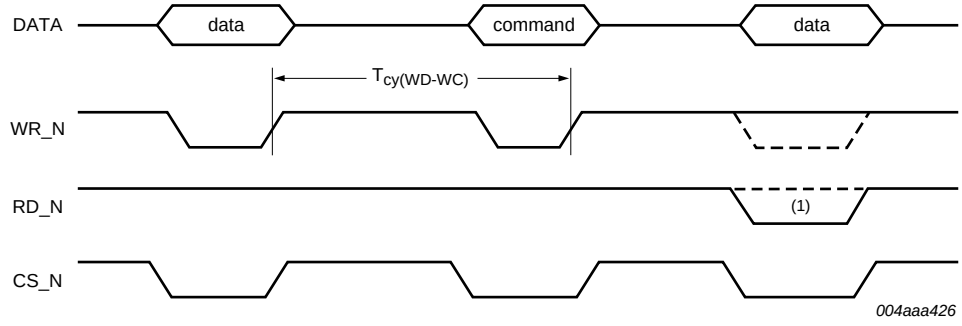


Fig 15. Write command + write data cycle timing



(1) Example: read data.

Fig 16. Write data + write command cycle timing

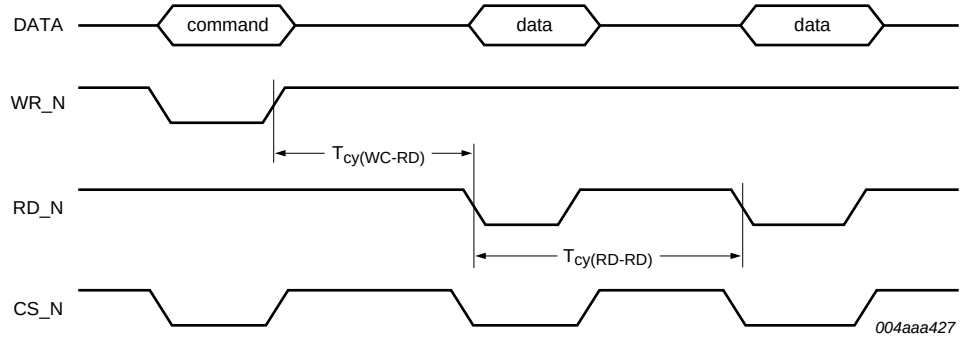
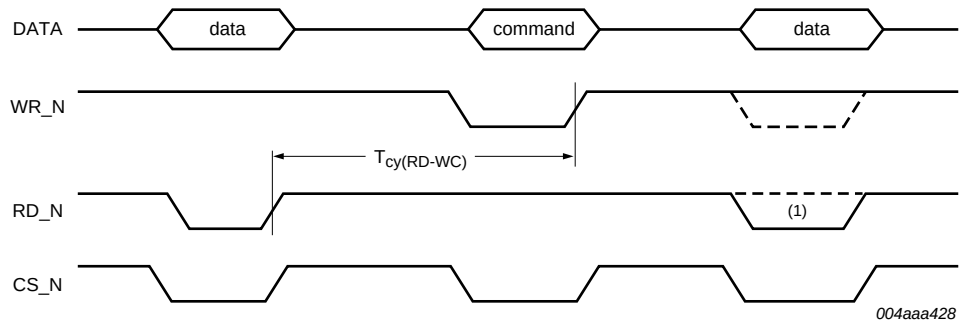


Fig 17. Write command + read data cycle timing



(1) Example: read data.

Fig 18. Read data + write command cycle timing

16.1.3 DMA timing: single-cycle mode

Table 57. Dynamic characteristics: single-cycle DMA timing

V_{BUS} = $V_{REG(3V3)}$ = 2.7 V to 3.9 V; $V_{DD(I/O)}$ = 1.8 V; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
8237 compatible mode (see Figure 19)						
t_{ASRP}	DREQ off after DACK on		-		40	ns
$T_{cy(DREQ)}$	cycle time signal DREQ		90		-	ns
Read in DACK-only mode (see Figure 20)						
t_{ASRP}	DREQ off after DACK on		-		40	ns
t_{ASAP}	DACK pulse width		25		-	ns
t_{ASAP} + t_{APRS}	DREQ on after DACK off		90		-	ns
t_{ASDV}	data valid after DACK on		-		22	ns
t_{APDZ}	data hold after DACK off		-		3	ns
Write in DACK-only mode (see Figure 21)						
t_{ASRP}	DREQ off after DACK on		-		40	ns
t_{ASAP} + t_{APRS}	DREQ on after DACK off		90		-	ns
t_{DVAP}	data setup before DACK off		5		-	ns
t_{APDZ}	data hold after DACK off		3		-	ns

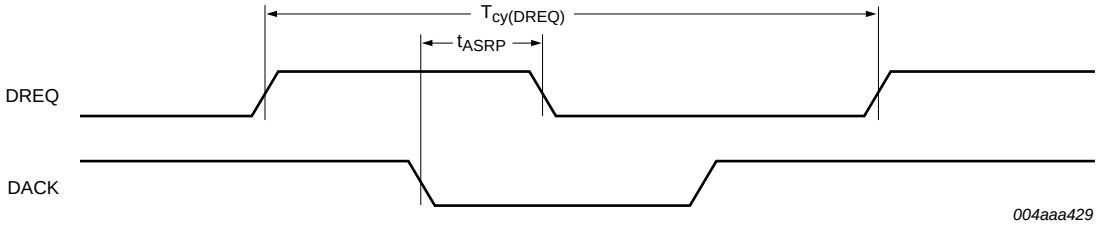


Fig 19. DMA timing in 8237 compatible mode

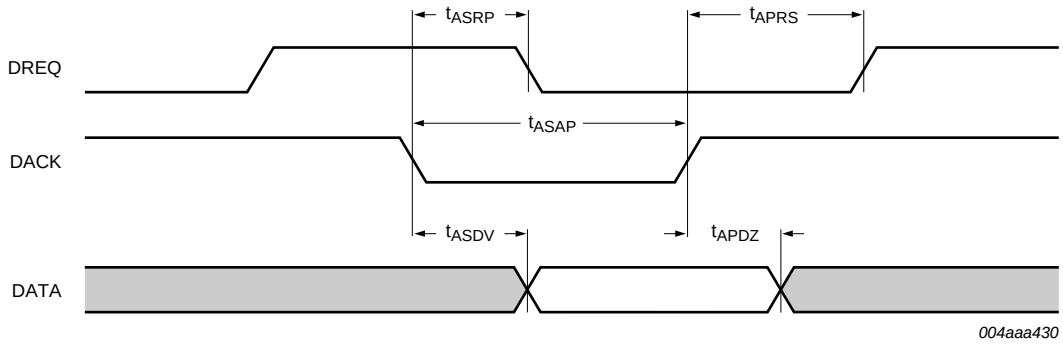


Fig 20. DMA read timing in DACK-only mode

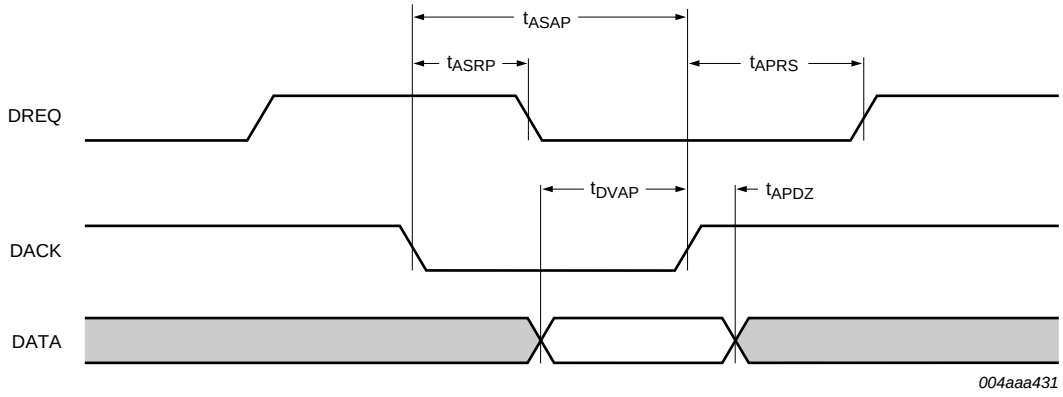


Fig 21. DMA write timing in DACK-only mode

16.1.4 DMA timing: burst mode

Table 58. Dynamic characteristics: burst mode DMA timing

V_{BUS} = $V_{REG(3V3)}$ = 2.7 V to 3.9 V; $V_{DD(I/O)}$ = 1.8 V; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Burst (see Figure 22)						
t_{RSIH}	input RD_N or WR_N HIGH after DREQ on		22		-	ns
t_{ILRP}	DREQ off after input RD_N or WR_N LOW		-		60	ns
t_{IHAP}	DACK off after input RD_N or WR_N HIGH		0		-	ns
t_{IHIL}	DMA burst repeat interval (input RD_N or WR_N HIGH to LOW)		90		-	ns

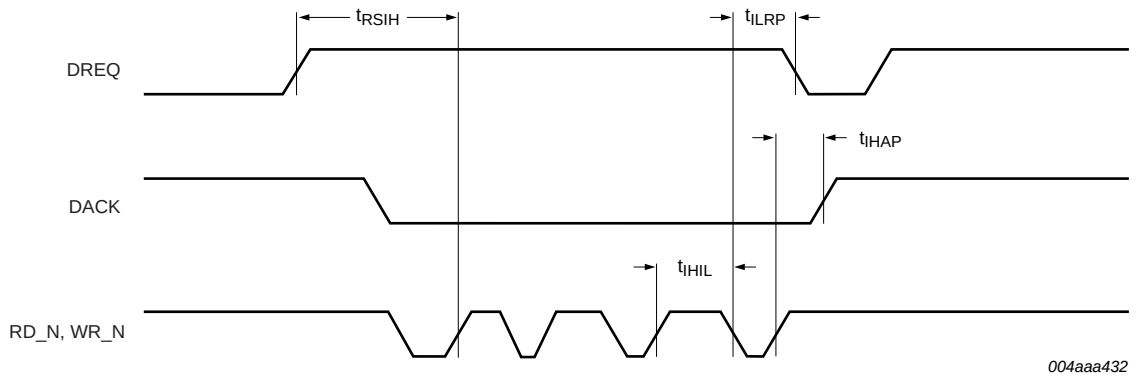


Fig 22. Burst mode DMA timing

17. Application information

17.1 Bus-powered mode

In bus-powered mode, pin VBUSDET_N is not necessary. See Figure 23.

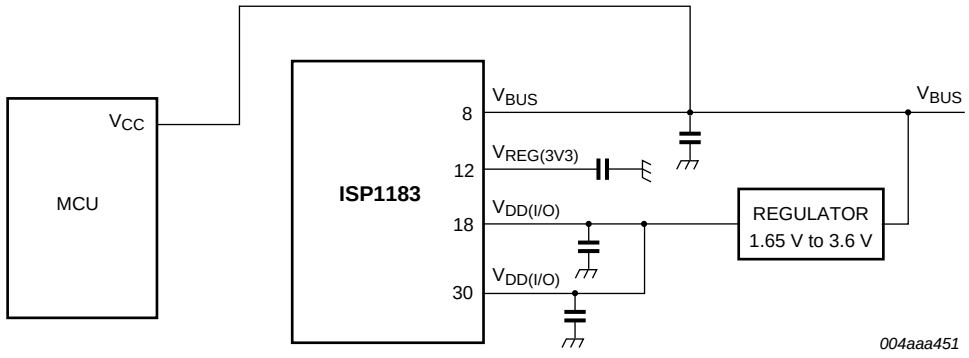


Fig 23. Bus-powered mode

17.2 Hybrid-powered mode

In this mode:

- When the USB cable is pulled out, pin VBUSDET_N goes HIGH, thereby indicating to the microcontroller that USB is disconnected. See Figure 24.
- When the USB cable is plugged in, pin VBUSDET_N goes LOW. This indicates to the microcontroller that the USB cable is plugged in. The microcontroller can then prepare to reconfigure all registers of the ISP1183. See Figure 24.

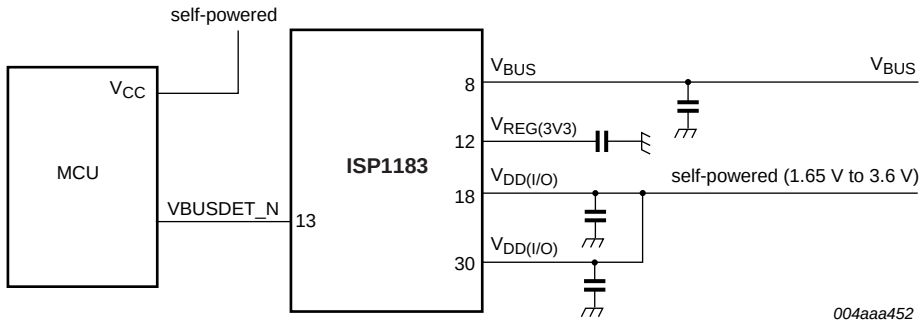


Fig 24. Hybrid-powered mode

17.3 Self-powered mode

In self-powered mode, pin VBUSDET_N cannot be used. The V_{BUS} sensing can be done in the following two ways:

- Connecting V_{BUS} to the microprocessor; see Figure 25.
  - When VBUSDET goes LOW, the microprocessor clears bit SOFTCT.
  - When VBUSDET goes HIGH, the microprocessor sets bit SOFTCT.
- Connecting transistor switching; see Figure 26.
  - When V_{BUS} is HIGH, $V_{REG(3V3)}$ will bypass to pull up DP. This indicates that the device is connected.
  - When V_{BUS} is LOW, pull up DP is switched off. This indicates that the device is disconnected.

Remark: The above implementation is necessary to comply with USB-IF requirements.

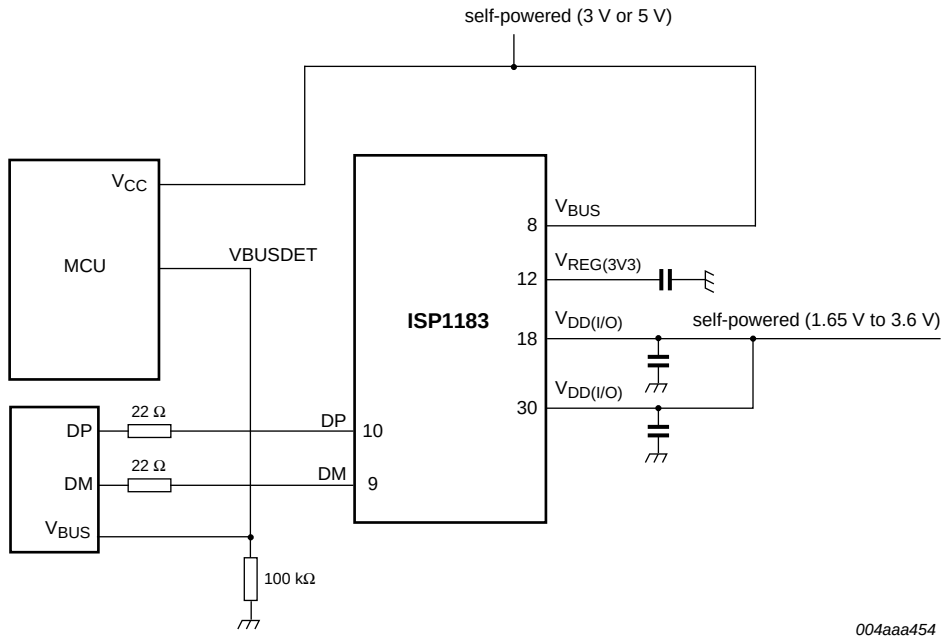


Fig 25. V_{BUS} connected to the microprocessor

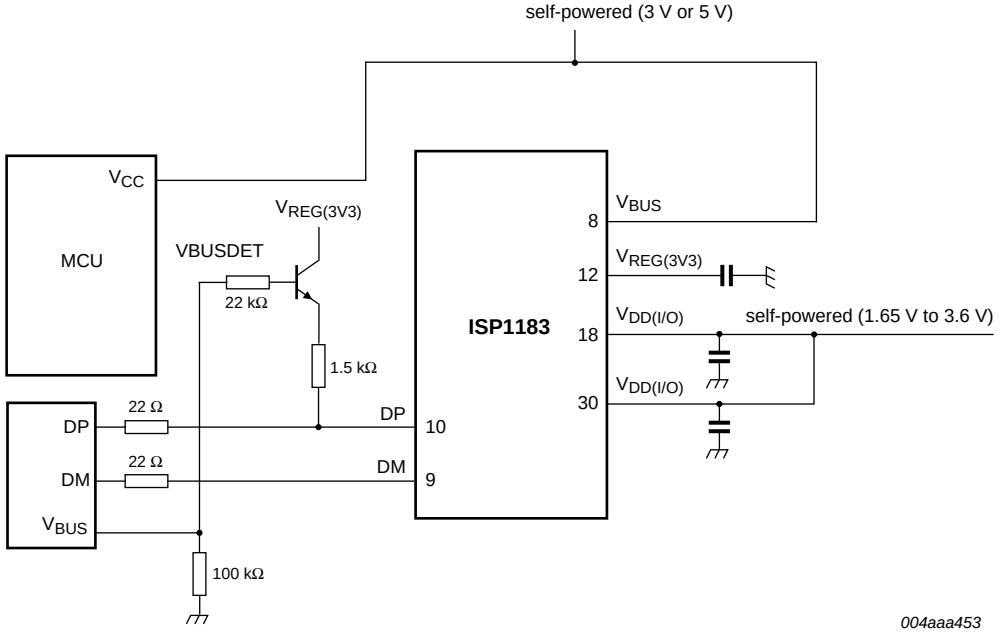
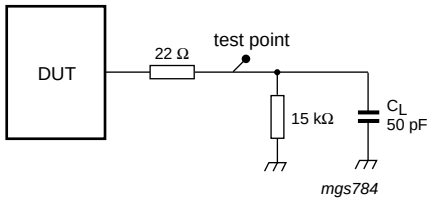


Fig 26. Transistor switching

18. Test information

The dynamic characteristics of the analog I/O ports (DP and DM) as listed in Table 54 were determined using the circuit shown in Figure 27.



Load capacitance:

(1) C_L = 50 pF (full-speed mode)

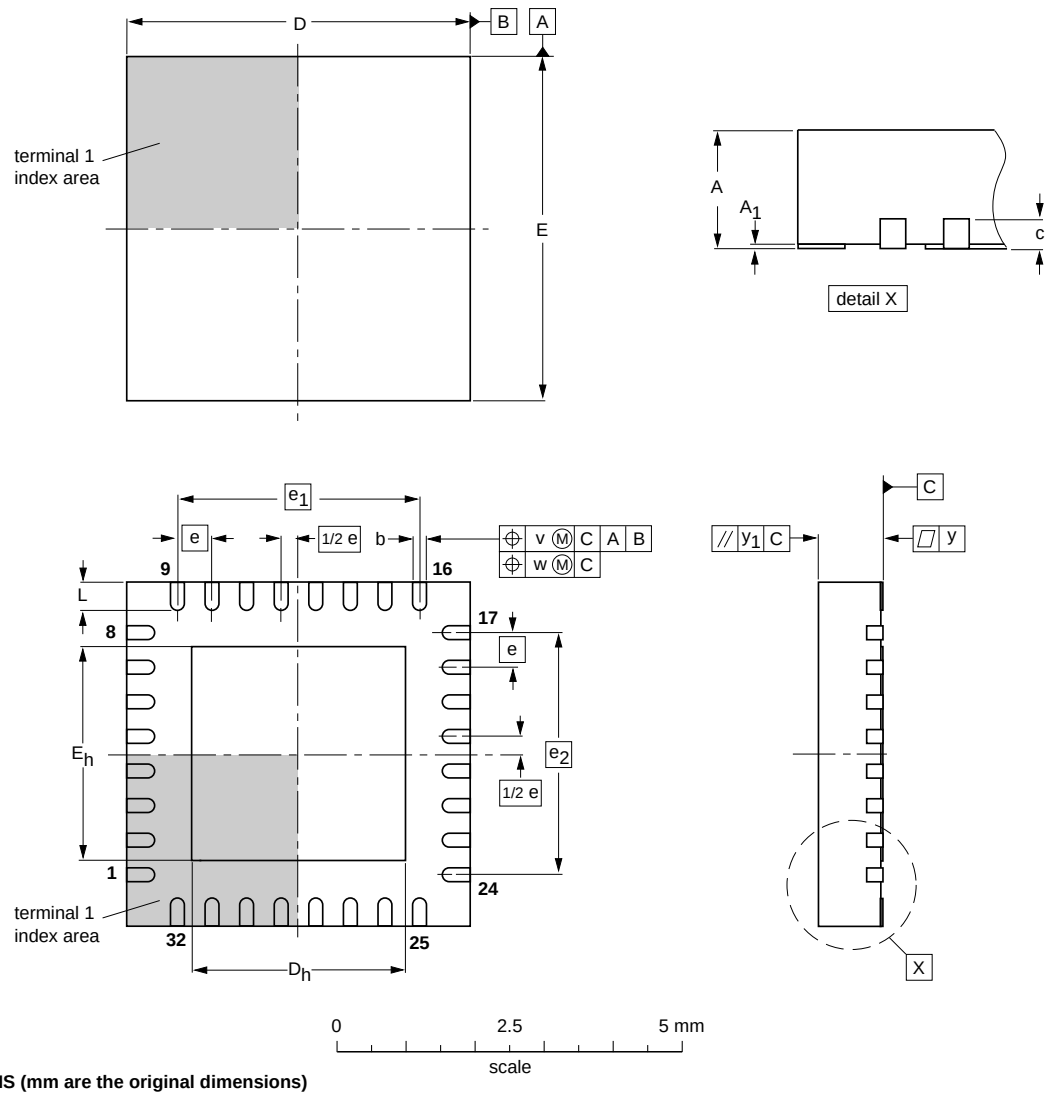
Speed:

(1) Full-speed mode only: internal 1.5 kΩ pull-up resistor on DP

Fig 27. Load impedance for the DP and DM pins

19. Package outline

HVQFN32: plastic thermal enhanced very thin quad flat package; no leads; 32 terminals; body 5 x 5 x 0.85 mm **SOT617-1**



DIMENSIONS (mm are the original dimensions)

UNIT	$A^{(1)}$ max.	A_1	b	c	$D^{(1)}$	D_h	$E^{(1)}$	E_h	e	e_1	e_2	L	v	w	y	y_1
mm	1	0.05 0.00	0.30 0.18	0.2	5.1 4.9	3.25 2.95	5.1 4.9	3.25 2.95	0.5	3.5	3.5	0.5 0.3	0.1	0.05	0.05	0.1

Note

1. Plastic or metal protrusions of 0.075 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT617-1	- - -	MO-220	- - -			~~01-08-08~~ 02-10-18

Fig 28. Package outline SOT617-1 (HVQFN32)

20. Abbreviations

Table 59. Abbreviations

Acronym	Description
ACK	Acknowledge
ACPI	Advanced Configuration and Power Interface
AT	Advanced Technology
CMOS	Complementary Metal-Oxide Semiconductor
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
ECR	Endpoint Configuration Register
EMI	ElectroMagnetic Interference
EOP	End-Of-Packet
EOT	End-Of-Transfer
ESR	Endpoint Status Register
FIFO	First In, First Out
I/O	Input/Output
IR	Interrupt Register
ISO	Isochronous
MMU	Memory Management Unit
NAK	Not Acknowledge
PCB	Printed Circuit Board
PID	Packet Identifier
PIO	Parallel I/O
PLL	Phase-Locked Loop
POR	Power-On Reset
PSOF	Pseudo Start-Of-Frame
RISC	Reduced Instruction Set Computer
SIE	Serial Interface Engine
SOF	Start-Of-Frame
USB	Universal Serial Bus

21. Revision history

Table 60. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
ISP1183_4	20090929	Product data sheet	-	ISP1183_3
Modifications:	• Rebranded to the ST-Ericsson template. • Section 4 “Ordering information”: updated. • Removed soldering information.			
ISP1183_3	20090120	Product data sheet	-	ISP1183_2
ISP1183_2	20070607	Product data sheet	-	ISP1183-01
ISP1183-01 (9397 750 11804)	20040224	Product data	-	-

22. Tables

23. Figures

24. Contents

Contact information at www.stericsson.com under Contacts

www.stericsson.com