ST-NXP Wireless

IMPORTANT NOTICE

Dear customer,

As from August 2nd 2008, the wireless operations of NXP have moved to a new company, ST-NXP Wireless.

As a result, the following changes are applicable to the attached document.

- **Company name - Philips Semiconductors** is replaced with **ST-NXP Wireless**.

- **Copyright** - the copyright notice at the bottom of each page "© Koninklijke Philips Electronics N.V. 200x. All rights reserved", shall now read: "© ST-NXP Wireless 200x - All rights reserved".

- **Web site** - http://www.semiconductors.philips.com is replaced with http://www.stnwireless.com

- **Contact information** - the list of sales offices previously obtained by sending an email to sales.addresses@www.semiconductors.philips.com, is now found at http://www.stnwireless.com under Contacts.

If you have any questions related to the document, please contact our nearest sales office. Thank you for your cooperation and understanding.

ST-NXP Wireless

ISP1181A

Full-speed Universal Serial Bus peripheral controller

Rev. 05 — 08 December 2004 **Product data**

1. General description

The ISP1181A 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 ISP1181A communicates with the system's microcontroller or microprocessor through a high-speed general-purpose parallel interface.

The ISP1181A supports fully autonomous, multi-configurable Direct Memory Access (DMA) operation.

The modular approach to implementing a USB peripheral controller allows the 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 ISP1181A is ideally suited for application in many personal computer peripherals such as printers, communication devices, scanners, external mass storage (Zip® drive) devices and digital still cameras. It offers an immediate cost reduction for applications that currently use SCSI implementations.

2. Features

- Complies with *Universal Serial Bus Specification Rev. 2.0* and most Device Class specifications
- 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 Mbyte/s or 90 ns read/write cycle) parallel interface
- Fully autonomous and multi-configuration DMA operation
- Up to 14 programmable USB endpoints with 2 fixed control IN/OUT endpoints
- Integrated physical 2462 bytes of multi-configuration FIFO memory
- Endpoints with double buffering to increase throughput and ease real-time data transfer
- Seamless interface with most microcontrollers/microprocessors
- Bus-powered capability with low power consumption and low 'suspend' current
- 6 MHz crystal oscillator input with integrated PLL for low EMI
- Controllable LazyClock (100 kHz ± 50 %) output during 'suspend'
- Software controlled connection to the USB bus (SoftConnect™)
- Good USB connection indicator that blinks with traffic (GoodLink™)

PHILIPS

- Clock output with programmable frequency (up to 48 MHz)
- Complies with the ACPI™, OnNow™ and USB power management requirements
- Internal power-on and low-voltage reset circuit, with possibility of a software reset
- Operation over the extended USB bus voltage range (4.0 V to 5.5 V) with 5 V tolerant I/O pads
- Operating temperature range –40 °C to +85 °C
- Full-scan design with high fault coverage
- Available in TSSOP48 and HVQFN48 packages.

3. Applications

- Personal Digital Assistant (PDA)
- Digital camera
- Communication device, for example:
  - Router
  - Modem
- Mass storage device, for example:
  - Zip drive
- Printer
- Scanner.

4. Ordering information

Table 1: Ordering information

Type number	Package		
	Name	Description	Version
ISP1181ADGG	TSSOP48	Plastic thin shrink small outline package; 48 leads; body width 6.1 mm	SOT362-1
ISP1181ABS	HVQFN48	Plastic thermal enhanced very thin quad flat package; no leads; 48 terminals; body 7 × 7 × 0.85 mm	SOT619-2

5. Block diagram

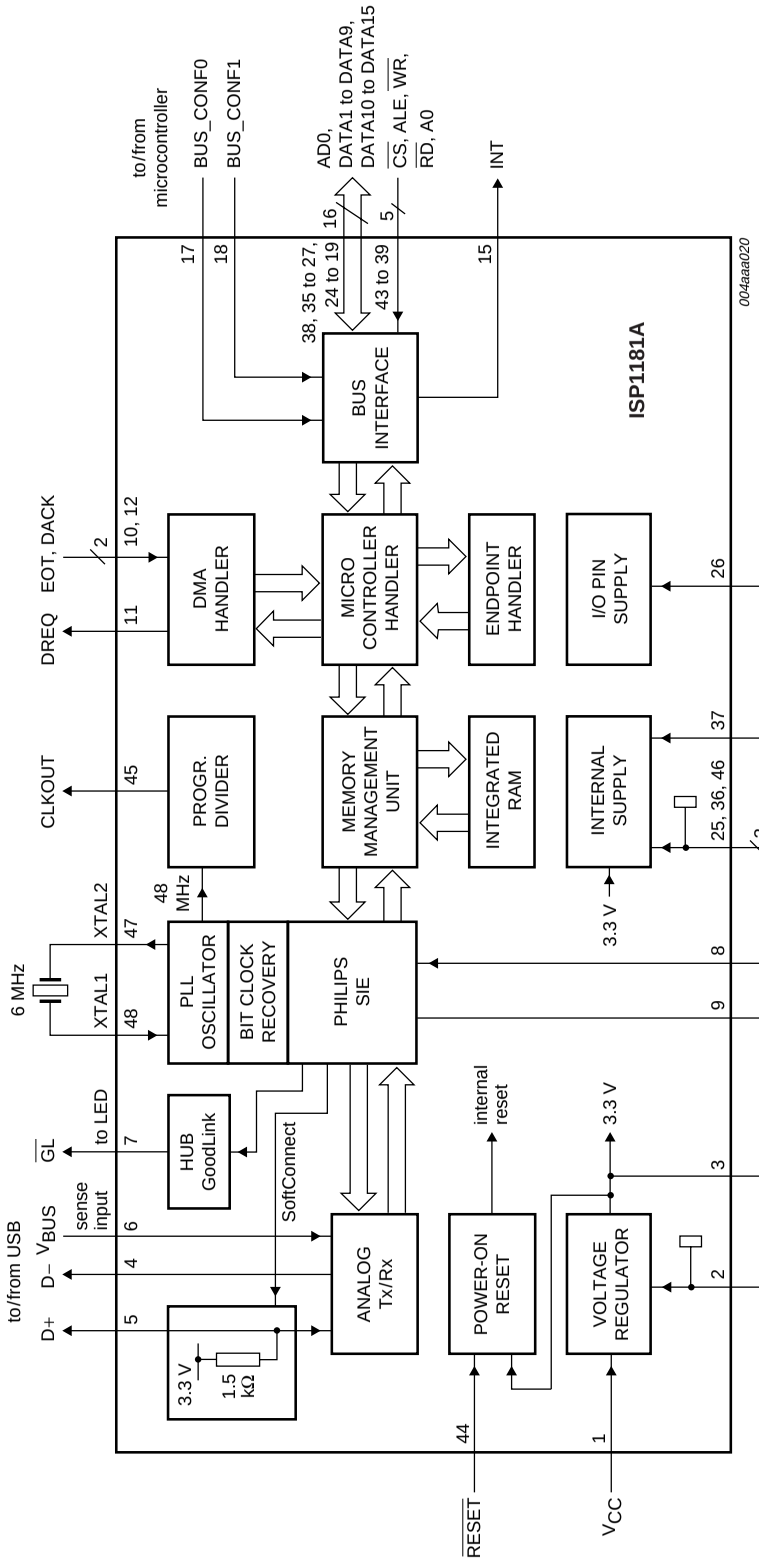


Fig 1. Block diagram.

6. Pinning information

6.1 Pinning

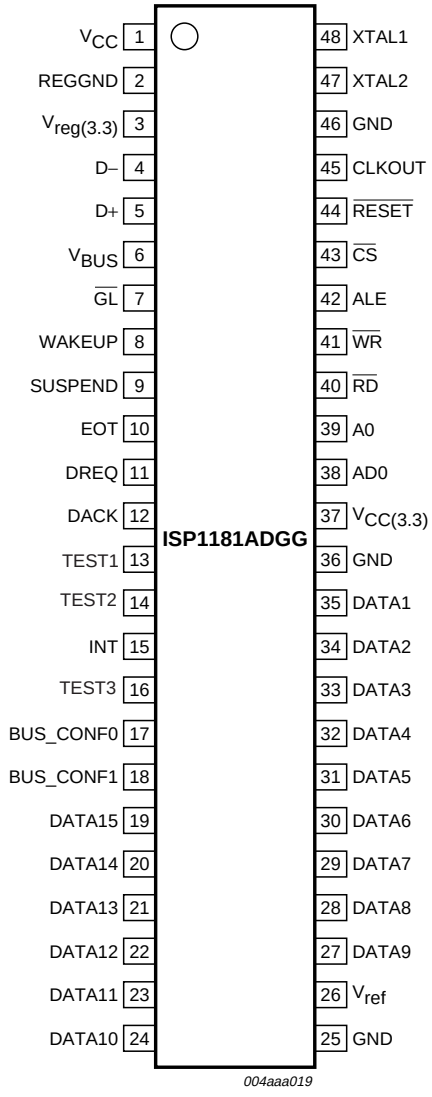


Fig 2. Pin configuration TSSOP48.

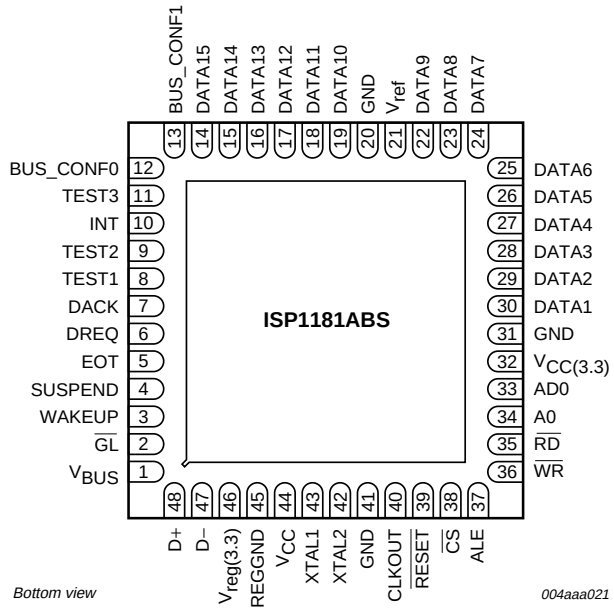


Fig 3. Pin configuration HVQFN48.

6.2 Pin description

Table 2: Pin description

Symbol[1]	Pin		Type	Description
	TSSOP48	HVQFN48		
V_{CC}	1	44	-	supply voltage (3.3 V or 5.0 V)
REGGND	2	45	-	voltage regulator ground supply
$V_{reg(3.3)}$	3	46	-	regulated supply voltage (3.3 V ± 10 %) from internal regulator; used to connect decoupling capacitor and pull-up resistor on D+ line; **Remark:** Cannot be used to supply external devices.
D–	4	47	AI/O	USB D– connection (analog)
D+	5	48	AI/O	USB D+ connection (analog)
V_{BUS}	6	1	I	V_{BUS} sensing input
$\overline{GL}$	7	2	O	GoodLink LED indicator output (open-drain, 8 mA); the LED is default ON, blinks OFF upon USB traffic; to connect an LED use a series resistor of 470 Ω (V_{CC} = 5.0 V) or 330 Ω (V_{CC} = 3.3 V)
WAKEUP	8	3	I	wake-up input (edge triggered, LOW to HIGH); generates a remote wake-up from ‘suspend’ state
SUSPEND	9	4	O	‘suspend’ state indicator output (4 mA); used as power switch control output (active LOW) for powered-off application

Table 2: Pin description *...continued*

Symbol[1]	Pin		Type	Description
	TSSOP48	HVQFN48		
EOT	10	5	I	End-Of-Transfer input (programmable polarity, see Table 21); used by the DMA controller to force the end of a DMA transfer to the ISP1181A
DREQ	11	6	O	DMA request output (4 mA; programmable polarity, see Table 21); signals to the DMA controller that the ISP1181A wants to start a DMA transfer
DACK	12	7	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 ISP1181A
TEST1	13	8	I	test input; this pin must be connected to V_{CC} via an external 10 kΩ resistor
TEST2	14	9	I	test input; this pin must be connected to V_{CC} via an external 10 kΩ resistor
INT	15	10	O	interrupt output; programmable polarity (active HIGH or LOW) and signalling (level or pulse); see Table 21
TEST3	16	11	O	test output; this pin is used for test purposes only
BUS_CONF0	17	12	I	bus configuration selector; see Table 3
BUS_CONF1	18	13	I	bus configuration selector; see Table 3
DATA15	19	14	I/O	bit 15 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA14	20	15	I/O	bit 14 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA13	21	16	I/O	bit 13 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA12	22	17	I/O	bit 12 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA11	23	18	I/O	bit 11 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA10	24	19	I/O	bit 10 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
GND	25	20	-	ground supply
V_{ref}	26	21	-	I/O pin reference voltage (3.3 V); no connection if V_{CC} = 5.0 V
DATA9	27	22	I/O	bit 9 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA8	28	23	I/O	bit 8 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA7	29	24	I/O	bit 7 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA6	30	25	I/O	bit 6 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)

Table 2: Pin description *...continued*

Symbol[1]	Pin		Type	Description
	TSSOP48	HVQFN48		
DATA5	31	26	I/O	bit 5 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA4	32	27	I/O	bit 4 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA3	33	28	I/O	bit 3 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA2	34	29	I/O	bit 2 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
DATA1	35	30	I/O	bit 1 of D[15:0]; bidirectional data line (slew-rate controlled output, 4 mA)
GND	36	31	-	ground supply
$V_{CC(3.3)}$	37	32	-	supply voltage (3.0 V to 3.6 V); leave this pin unconnected when using V_{CC} = 5.0 V
AD0	38	33	I/O	multiplexed bidirectional address and data line; represents address A0 or bit 0 of D[15:0] in conjunction with input ALE; level-sensitive input or slew-rate controlled output (4 mA) **Address phase**: a HIGH-to-LOW transition on input ALE latches the level on this pin as address A0 (1 = command, 0 = data) **Data phase**: during reading this pin outputs bit D[0]; during writing the level on this pin is latched as bit D[0]
A0	39	34	I	address input; selects command (A0 = 1) or data (A0 = 0); in a multiplexed address/data bus configuration this pin is not used and must be tied LOW (connect to GND)
$\overline{RD}$	40	35	I	read strobe input
$\overline{WR}$	41	36	I	write strobe input
ALE	42	37	I	address latch enable input; a HIGH-to-LOW transition latches the level on pin AD0 as address information in a multiplexed address/data bus configuration; must be tied LOW (connect to GND) for a separate address/data bus configuration
$\overline{CS}$	43	38	I	chip select input
$\overline{RESET}$	44	39	I	reset input (Schmitt trigger); a LOW level produces an asynchronous reset; connect to V_{CC} for power-on reset (internal POR circuit)
CLKOUT	45	40	O	programmable clock output (2 mA)

Table 2: Pin description *...continued*

Symbol[1]	Pin		Type	Description
	TSSOP48	HVQFN48		
GND	46	41	-	ground supply
XTAL2	47	42	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
XTAL1	48	43	I	crystal oscillator input (6 MHz); connect a fundamental parallel-resonant crystal or an external clock source (leave pin XTAL2 unconnected)

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

7. Functional description

The ISP1181A 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 or microprocessors. It supports different bus configurations (see Table 3) and local DMA transfers of up to 16 bytes per cycle. The block diagram is given in Figure 1.

The ISP1181A has 2462 bytes of internal FIFO memory, which is shared among the enabled USB endpoints. The type and FIFO size of each endpoint can be individually configured, depending on the required packet size. Isochronous and bulk endpoints are double-buffered for increased data throughput.

The ISP1181A requires a single supply voltage of 3.3 V or 5.0 V and has an internal 3.3 V voltage regulator for powering the analog USB transceiver. It supports bus-powered operation.

The ISP1181A operates on a 6 MHz oscillator frequency. A programmable clock output is available up to 48 MHz. During ‘suspend’ state the 100 kHz ± 50 % LazyClock frequency can be output.

7.1 Analog transceiver

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

7.2 Philips Serial Interface Engine (SIE)

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

7.3 Memory Management Unit (MMU) and integrated RAM

The MMU and the integrated RAM provide the conversion between the USB speed (12 Mbit/s bursts) and the parallel interface to the microcontroller (max. 12 Mbyte/s). This allows the microcontroller to read and write USB packets at its own speed.

7.4 SoftConnect

The connection to the USB is accomplished by bringing D+ (for full-speed USB peripherals) HIGH through a 1.5 kΩ pull-up resistor. In the ISP1181A, the 1.5 kΩ pull-up resistor is integrated on-chip and is not connected to V_{CC} by default. The connection is established by a command sent from the external/system microcontroller. This allows the system microcontroller to complete its initialization sequence before deciding to establish connection with the USB. Reinitialization of the USB connection can also be performed without disconnecting the cable.

The ISP1181A will check for USB V_{BUS} availability before the connection can be established. V_{BUS} sensing is provided through pin V_{BUS}.

V_{BUS} sensing prevents the peripheral from wake-up when V_{BUS} is not present. Without V_{BUS} sensing, any activity or noise on (D+, D-) might wake up the peripheral. With V_{BUS} sensing, (D+, D-) is decoupled when no V_{BUS} is present. Therefore, even if there is noise on the (D+, D-) lines, it is not taken into account. This ensures that the peripheral remains in the suspend state.

Remark: Note that the tolerance of the internal resistors is 25 %. This is higher than the 5 % tolerance specified by the USB specification. However, the overall voltage specification for the connection can still be met with a good margin. The decision to make use of this feature lies with the USB equipment designer.

7.5 GoodLink

Indication of a good USB connection is provided at pin $\overline{GL}$ through GoodLink technology. During enumeration, the LED indicator will blink on momentarily. When the ISP1181A has been successfully enumerated (the peripheral address is set), the LED indicator will remain permanently on. Upon each successful packet transfer (with ACK) to and from the ISP1181A, the LED will blink off for 100 ms. During ‘suspend’ state, the LED will remain off.

This feature provides a user-friendly indication of the status of the USB peripheral, the connected hub, and the USB traffic. It is a useful field diagnostics tool for isolating faulty equipment. It can therefore help to reduce field support and hotline overhead.

7.6 Bit clock recovery

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

7.7 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 $V_{reg(3.3)}$ to supply an external 1.5 kΩ pull-up resistor on the D+ line. Alternatively, the ISP1181A provides SoftConnect technology via an integrated 1.5 kΩ pull-up resistor (see Section 7.4).

7.8 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.

7.9 Parallel I/O (PIO) and Direct Memory Access (DMA) interface

A generic PIO interface is defined for speed and ease-of-use. It also allows direct interfacing to most microcontrollers. To a microcontroller, the ISP1181A appears as a memory device with an 8/16-bit data bus and a 1-bit address line. The ISP1181A supports both multiplexed and non-multiplexed address and data buses.

The ISP1181A can also be configured as a DMA slave device to allow more efficient data transfer. One of the 14 endpoint FIFOs may directly transfer data to/from the local shared memory. The DMA interface can be configured independently from the PIO interface.

8. Modes of operation

The ISP1181A has four bus configuration modes, selected via pins BUS_CONF1 and BUS_CONF0:

Mode 0 16-bit I/O port shared with 16-bit DMA port

Mode 1 reserved

Mode 2 8-bit I/O port shared with 8-bit DMA port

Mode 3 reserved.

The bus configurations for each of these modes are given in Table 3. Typical interface circuits for each mode are given in Section 21.1.

Table 3: Bus configuration modes

Mode	BUS_CONF[1:0]		PIO width	DMA width		Description
				DMAWD = 0	DMAWD = 1	
0	0	0	D[15:1], AD0	-	D[15:1], AD0	multiplexed address/data on pin AD0; bus is shared by 16-bit I/O port and 16-bit DMA port
1	0	1	reserved	reserved	reserved	reserved
2	1	0	D[7:1], AD0	D[7:1], AD0	-	multiplexed address/data on pin AD0; bus is shared by 8-bit I/O port and 8-bit DMA port
3	1	1	reserved	reserved	reserved	reserved

9. Endpoint descriptions

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

The ISP1181A has 16 endpoints: endpoint 0 (control IN and OUT) plus 14 configurable endpoints, which can be individually defined as interrupt/bulk/isochronous, IN or OUT. Each enabled endpoint has an associated FIFO, which can be accessed either via the parallel I/O interface or via DMA.

9.1 Endpoint access

Table 4 lists the 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 via bits EPIDX[3:0] and DMAEN of the DMA Configuration 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	programmable	supported	supported	supported	programmable
2	programmable	supported	supported	supported	programmable
3	programmable	supported	supported	supported	programmable
4	programmable	supported	supported	supported	programmable
5	programmable	supported	supported	supported	programmable
6	programmable	supported	supported	supported	programmable
7	programmable	supported	supported	supported	programmable
8	programmable	supported	supported	supported	programmable
9	programmable	supported	supported	supported	programmable
10	programmable	supported	supported	supported	programmable
11	programmable	supported	supported	supported	programmable
12	programmable	supported	supported	supported	programmable
13	programmable	supported	supported	supported	programmable
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 (ISP1181A transmits); OUT: output from the USB host (ISP1181A receives). The data flow direction is determined by bit EPDIR in the Endpoint Configuration Register.

9.2 Endpoint FIFO size

The size of the FIFO 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 the 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 the 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 be configured independently via its ECR, but the total physical size of all enabled endpoints (IN plus OUT) 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 peripheral 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 the standard USB request, Set Interface, the firmware must program all 16 ECRs of the ISP1181A in sequence (see Table 4), whether the endpoints are enabled or not. The hardware will then automatically allocate FIFO storage space.

If all endpoints have been configured successfully, the 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, the firmware must stall the control IN endpoint.

When reset by hardware or via the USB bus, the ISP1181A 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; however, it is 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) will be set by the SIE. The firmware then responds to the interrupt and selects the endpoint for processing.

The endpoint interrupt bit will be cleared by reading the Endpoint Status Register (ESR). The ESR also contains information on the status of the endpoint buffer.

For an OUT (= receive) endpoint, the packet length and packet data can be read from ISP1181A using the Read Buffer command. When the whole packet has been read, the 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 ISP1181A using the Write Buffer command. When the whole packet has been written to the buffer, the 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 needs to 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 central processor (CPU). Many different implementations of DMA exist. The ISP1181A 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 ISP1181A supports DMA transfer for all 14 configurable endpoints (see Table 4). Only one endpoint at a time can be selected for DMA transfer. The DMA operation of the ISP1181A 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)
- Multiple End-Of-Transfer (EOT) sources: external pin, internal conditions, short/empty packet
- Programmable signal levels on pins DREQ, DACK and EOT.

10.1 Selecting an endpoint for DMA transfer

The target endpoint for DMA access is selected via bits EPDIX[3:0] in the DMA Configuration Register, as shown in 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 I/O mode access.

Table 7: Endpoint selection for DMA transfer

Endpoint identifier	EPIDX[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

Table 7: Endpoint selection for DMA transfer…*continued*

Endpoint identifier	EPIDX[3:0]	Transfer direction	
		EPDIR = 0	EPDIR = 1
12	1101	OUT: read	IN: write
13	1110	OUT: read	IN: write
14	1111	OUT: read	IN: write

10.2 8237 compatible mode

The 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	ISP1181A requests a DMA transfer
DACK	DMA acknowledge	I	DMA controller confirms the transfer
EOT	end of transfer	I	DMA controller terminates the transfer
$\overline{RD}$	read strobe	I	instructs ISP1181A to put data on the bus
$\overline{WR}$	write strobe	I	instructs ISP1181A 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: the 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 ISP1181A in 8237 compatible DMA mode is given in Figure 4.

The 8237 has two control signals for each DMA channel: DREQ (DMA Request) and $\overline{DACK}$ (DMA Acknowledge). General control signals are HRQ (Hold Request) and HLDA (Hold Acknowledge). The bus operation is controlled via $\overline{MEMR}$ (Memory Read), $\overline{MEMW}$ (Memory Write), $\overline{IOR}$ (I/O read) and $\overline{IOW}$ (I/O write).

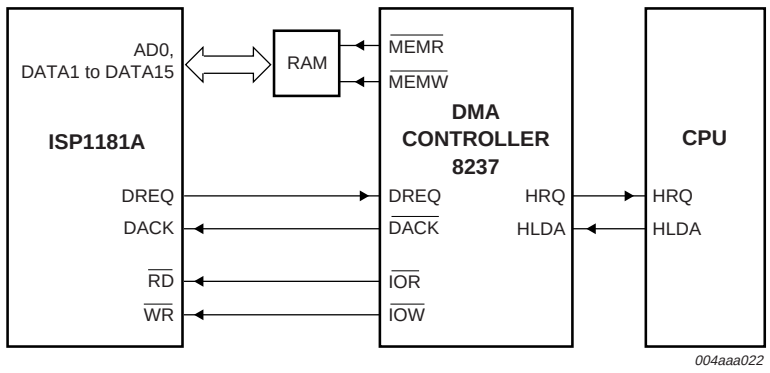


Fig 4. ISP1181A in 8237 compatible DMA mode.

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

1. ISP1181A receives a data packet in one of its endpoint FIFOs; the packet must be transferred to memory address 1234H.
2. ISP1181A 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, the CPU places the bus control signals ($\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$) and the address lines in three-state and asserts HLDA to inform the 8237 that it has control of the bus.
5. The 8237 now sets its address lines to 1234H and activates the $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$ control signals.
6. The 8237 asserts $\overline{\text{DACK}}$ to inform the ISP1181A that it will start a DMA transfer.
7. The ISP1181A now places the byte or word to be transferred on the data bus lines, because its $\overline{\text{RD}}$ signal was asserted by the 8237.
8. The 8237 waits one DMA clock period and then de-asserts $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$. This latches and stores the byte or word at the desired memory location. It also informs the ISP1181A that the data on the bus lines has been transferred.
9. The ISP1181A 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 $\overline{\text{DACK}}$ output indicating that the ISP1181A must stop placing data on the bus.
11. The 8237 places the bus control signals ($\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$) and the address lines in three-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 the bus control lines ($\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ and $\overline{\text{IOW}}$) and the address lines, the CPU resumes the execution of instructions.

For a typical bulk transfer the above 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. When using 16-bit DMA, the number of transfers is 32, and address incrementing and byte counter decrementing is done by 2 for each word.

10.3 DACK-only mode

The 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 ISP1181A in DACK-only DMA mode is given in Figure 5.

Table 9: DACK-only mode: pin functions

Symbol	Description	I/O	Function
DREQ	DMA request	O	ISP1181A requests a DMA transfer
DACK	DMA acknowledge	I	DMA controller confirms the transfer; also functions as data strobe
EOT	End-Of-Transfer	I	DMA controller terminates the transfer
$\overline{\text{RD}}$	read strobe	I	not used
$\overline{\text{WR}}$	write strobe	I	not used

In DACK-only mode the ISP1181A uses the DACK signal as data strobe. Input signals $\overline{RD}$ and $\overline{WR}$ 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 $\overline{MEMW}$ and $\overline{MEMR}$ signals: the $\overline{RD}$ and $\overline{WR}$ signals are also used as memory data strobes.

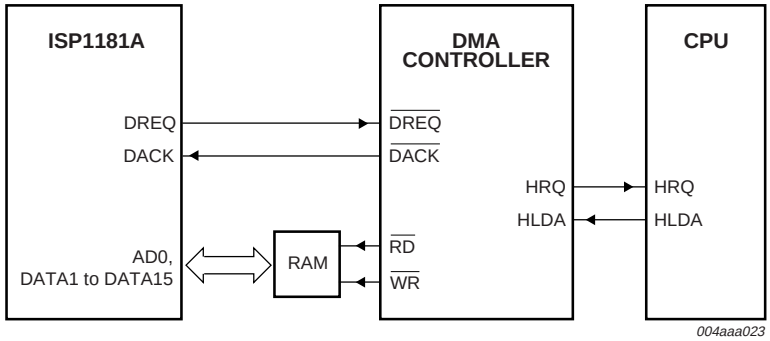


Fig 5. ISP1181A in DACK-only DMA mode.

10.4 End-Of-Transfer conditions

10.4.1 Bulk endpoints

A DMA transfer to/from a bulk endpoint can be terminated by any of the following conditions (bit names refer to the DMA Configuration Register, see Table 24):

- An external End-Of-Transfer signal occurs on input EOT
- 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.

External EOT: When reading from an OUT endpoint, an external EOT will stop the DMA operation and **clear any remaining data** in the current FIFO. For a double-buffered endpoint the other (inactive) buffer is not affected.

When writing to an IN endpoint, an EOT will stop the DMA operation and the data packet in the FIFO (even if it is smaller than the maximum packet size) will be sent to the USB host at the next IN token.

DMA Counter Register: An EOT from the DMA Counter Register is enabled by setting bit CNTREN in the DMA Configuration Register. The ISP1181A 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.

Short packet: Normally, the transfer byte count must be set via a control endpoint before any DMA transfer takes place. 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 the 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
EOT input	EOT is active	EOT is active
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 DMA Configuration Register	DMAEN = 0[1]	DMAEN = 0[1]

[1] The DMA transfer stops. However, no interrupt is generated.

10.4.2 Isochronous endpoints

A DMA transfer to/from an isochronous endpoint can be terminated by any of the following conditions (bit names refer to the DMA Configuration Register, see Table 24):

- An external End-Of-Transfer signal occurs on input EOT
- The DMA transfer completes as programmed in the DMA Counter register (CNTREN = 1)
- An End-Of-Packet (EOP) signal is detected
- DMA operation is disabled by clearing bit DMAEN.

Table 11: Recommended EOT usage for isochronous endpoints

EOT condition	OUT endpoint	IN endpoint
EOT input active	do not use	preferred
DMA Counter Register zero	do not use	preferred
End-Of-Packet	preferred	do not use

11. Suspend and resume

11.1 Suspend conditions

The ISP1181A 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 suspend status are as follows:

1. On detecting a wakeup-to-suspend transition, the ISP1181A sets bit SUSPND in the Interrupt register. This will generate an interrupt if bit IESUSP in the Interrupt Enable register is set.
2. When the firmware detects a suspend condition, it must prepare all system components for the suspend state:
   a. All signals connected to the ISP1181A must enter appropriate states to meet the power consumption requirements of the suspend state.
   b. All input pins of the ISP1181A must have a CMOS LOW or HIGH level.
3. In the interrupt service routine, the 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 the suspend mode and the process must be aborted. Otherwise, the next step can be executed.
4. To meet the suspend current requirements for a bus-powered device, the internal clocks must be switched off by clearing bit CLKRUN in the Hardware Configuration register.
5. When the firmware has set and cleared bit GOSUSP in the Mode register, the ISP1181A enters the suspend state. In powered-off application, the ISP1181A asserts output SUSPEND and switches off the internal clocks after 2 ms.

Figure 6 shows a typical timing diagram.

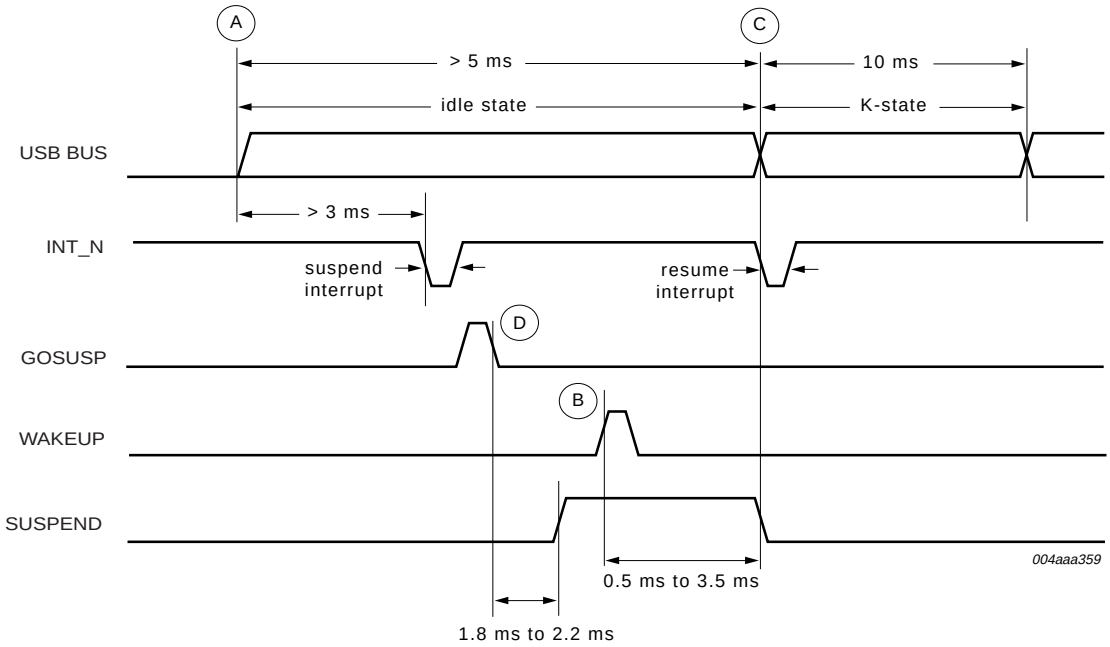


Fig 6. Suspend and resume timing.

In Figure 6:

- **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 $\overline{CS}$.
- **C**: indicates remote wake-up. The ISP1181A will drive a K-state on the USB bus for 10 ms after pin WAKEUP goes HIGH or pin $\overline{CS}$ goes LOW.
- **D**: after detecting the suspend interrupt, set and clear bit GOSUSP in the Mode register.

11.1.1 Powered-off application

Figure 7 shows a typical bus-powered modem application using the ISP1181A. The SUSPEND output switches off power to the microcontroller and other external circuits during the suspend state. The ISP1181A is woken up through the USB bus (global resume) or by the ring detection circuit on the telephone line.

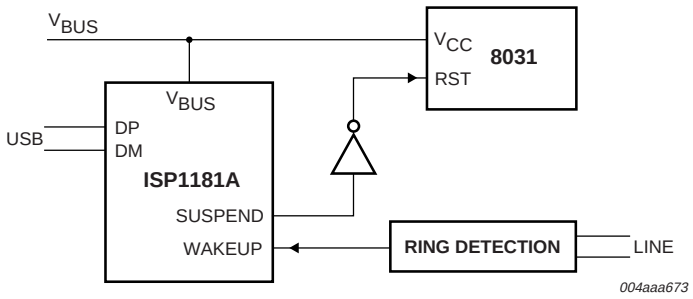


Fig 7. SUSPEND and WAKEUP signals in a powered-off modem application.

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 $\overline{CS}$ (if enabled using bit WKUPCS in the Hardware Configuration register). Wake-up on $\overline{CS}$ will work only if V_{BUS} is present.

The steps of a wake-up sequence are as follows:

1. The internal oscillator and the PLL multiplier are re-enabled. When stabilized, the clock signals are routed to all internal circuits of the ISP1181A.
2. The SUSPEND output is deasserted, 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 ISP1181A resumes its normal functionality.
4. In case of a remote wake-up, the ISP1181A drives a K-state on the USB bus for 10 ms.
5. Following the deassertion of output SUSPEND, the application restores itself and other system components to the normal operating mode.
6. After wake-up, the internal registers of the ISP1181A are write-protected to prevent corruption by inadvertent writing during power-up of external components. The firmware must send an Unlock Device command to the ISP1181A to restore its full functionality.

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

Table 12: Summary of control bits...*continued*

Register	Bit	Function
Interrupt Enable	IESUSP	enables output INT to signal the suspend state
	IERESUME	enables output INT to signal the resume state
Mode	SOFTCT	enables SoftConnect pull-up resistor to 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 $\overline{CS}$
	PWROFF	selects powered-off mode during the suspend state
Unlock	all	sending data AA37H unlocks the internal registers for writing after a resume

12. Commands and registers

The functions and registers of ISP1181A are accessed via 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 bit A0 = 1, the ISP1181A interprets the data on the lower byte of the bus bits D[7:0] as a command code. Commands without a data phase are executed immediately.
2. **Data phase (optional)**: when address bit A0 = 0, the ISP1181A transfers the data on the bus to or from a register or endpoint FIFO. Multi-byte registers are accessed least significant byte/word first.

The following applies for register or FIFO access in 16-bit bus mode:

- The upper byte (bits D15 to D8) in command phase or the undefined byte in data phase are ignored.
- The access of registers is word-aligned: byte access is not allowed.
- If the packet length is odd, the upper byte of the last word in an IN endpoint buffer is **not** transmitted to the host. When reading from an OUT endpoint buffer, the upper byte of the last word must be ignored by the firmware. The packet length is stored in the first 2 bytes of the endpoint buffer.

Table 13: Command and register summary

Name	Destination	Code (Hex)	Transaction[1]
Initialization commands			
Write Control OUT Configuration	Endpoint Configuration Register endpoint 0 OUT	20	write 1 byte[2]
Write Control IN Configuration	Endpoint Configuration Register endpoint 0 IN	21	write 1 byte[2]
Write Endpoint n Configuration (n = 1 to 14)	Endpoint Configuration Register endpoint 1 to 14	22 to 2F	write 1 byte[2][3]
Read Control OUT Configuration	Endpoint Configuration Register endpoint 0 OUT	30	read 1 byte[2]

Table 13: Command and register summary *...continued*

Name	Destination	Code (Hex)	Transaction[1]
Read Control IN Configuration	Endpoint Configuration Register endpoint 0 IN	31	read 1 byte[2]
Read Endpoint n Configuration (n = 1 to 14)	Endpoint Configuration Register endpoint 1 to 14	32 to 3F	read 1 byte[2]
Write/Read Device Address	Address Register	B6/B7	write/read 1 byte[2]
Write/Read Mode Register	Mode Register	B8/B9	write/read 1 byte[2]
Write/Read Hardware Configuration	Hardware Configuration Register	BA/BB	write/read 2 bytes
Write/Read Interrupt Enable Register	Interrupt Enable Register	C2/C3	write/read 4 bytes
Write/Read DMA Configuration	DMA Configuration Register	F0/F1	write/read 2 bytes
Write/Read DMA Counter	DMA Counter Register	F2/F3	write/read 2 bytes
Reset Device	resets all registers	F6	-
Data flow commands			
Write Control OUT Buffer	illegal: endpoint is read-only	(00)	-
Write Control IN Buffer	FIFO endpoint 0 IN	01	N ≤ 64 bytes
Write Endpoint n Buffer (n = 1 to 14)	FIFO endpoint 1 to 14 (IN endpoints only)	02 to 0F	isochronous: N ≤ 1023 bytes interrupt/bulk: N ≤ 64 bytes
Read Control OUT Buffer	FIFO endpoint 0 OUT	10	N ≤ 64 bytes
Read Control IN Buffer	illegal: endpoint is write-only	(11)	-
Read Endpoint n Buffer (n = 1 to 14)	FIFO endpoint 1 to 14 (OUT endpoints only)	12 to 1F	isochronous: N ≤ 1023 bytes[4] interrupt/bulk: N ≤ 64 bytes
Stall Control OUT Endpoint	Endpoint 0 OUT	40	-
Stall Control IN Endpoint	Endpoint 0 IN	41	-
Stall Endpoint n (n = 1 to 14)	Endpoint 1 to 14	42 to 4F	-
Read Control OUT Status	Endpoint Status Register endpoint 0 OUT	50	read 1 byte[2]
Read Control IN Status	Endpoint Status Register endpoint 0 IN	51	read 1 byte[2]
Read Endpoint n Status (n = 1 to 14)	Endpoint Status Register n endpoint 1 to 14	52 to 5F	read 1 byte[2]
Validate Control OUT Buffer	illegal: IN endpoints only[5]	(60)	-
Validate Control IN Buffer	FIFO endpoint 0 IN[5]	61	-[3]
Validate Endpoint n Buffer (n = 1 to 14)	FIFO endpoint 1 to 14 (IN endpoints only)[5]	62 to 6F	-[3]
Clear Control OUT Buffer	FIFO endpoint 0 OUT	70	-[3]
Clear Control IN Buffer	illegal[6]	(71)	-
Clear Endpoint n Buffer (n = 1 to 14)	FIFO endpoint 1 to 14 (OUT endpoints only)[6]	72 to 7F	[3]
Unstall Control OUT Endpoint	Endpoint 0 OUT	80	-
Unstall Control IN Endpoint	Endpoint 0 IN	81	-

Table 13: Command and register summary *...continued*

Name	Destination	Code (Hex)	Transaction[1]
Unstall Endpoint n (n = 1 to 14)	Endpoint 1 to 14	82 to 8F	-
Check Control OUT Status[7]	Endpoint Status Image Register endpoint 0 OUT	D0	read 1 byte[2]
Check Control IN Status[7]	Endpoint Status Image Register endpoint 0 IN	D1	read 1 byte[2]
Check Endpoint n Status (n = 1 to 14)[7]	Endpoint Status Image Register n endpoint 1 to 14	D2 to DF	read 1 byte[2]
Acknowledge Setup	Endpoint 0 IN and OUT	F4	-[3]
General commands			
Read Control OUT Error Code	Error Code Register endpoint 0 OUT	A0	read 1 byte[2]
Read Control IN Error Code	Error Code Register endpoint 0 IN	A1	read 1 byte[2]
Read Endpoint n Error Code (n = 1 to 14)	Error Code Register endpoint 1 to 14	A2 to AF	read 1 byte[2]
Unlock Device	all registers with write access	B0	write 2 bytes
Write/Read Scratch Register	Scratch Register	B2/B3	write/read 2 bytes
Read Frame Number	Frame Number Register	B4	read 1 or 2 bytes
Read Chip ID	Chip ID Register	B5	read 2 bytes
Read Interrupt Register	Interrupt Register	C0	read 4 bytes

[1] With N representing the number of bytes, the number of words for 16-bit bus width is: (N + 1) DIV 2.

[2] When accessing an 8-bit register in 16-bit mode, the upper byte is invalid.

[3] In 8-bit bus mode this command requires more time to complete than other commands. See Table 58.

[4] During isochronous transfer in 16-bit mode, because N ≤ 1023, the firmware must take care of the upper byte.

[5] Validating an OUT endpoint buffer causes unpredictable behavior of ISP1181A.

[6] Clearing an IN endpoint buffer causes unpredictable behavior of ISP1181A.

[7] 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 the embedded endpoints. They also serve to set the USB assigned address of ISP1181A and to perform a device reset.

12.1.1 Write/Read Endpoint Configuration

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

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

Code (Hex): 20 to 2F — write (control OUT, control IN, endpoint 1 to 14)

Code (Hex): 30 to 3F — read (control OUT, control IN, endpoint 1 to 14)

Transaction — write/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	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15: Endpoint Configuration Register: bit description

Bit	Symbol	Description
7	FIFOEN	A logic 1 indicates an enabled FIFO with allocated memory. A 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	A logic 1 indicates that this endpoint has double buffering.
4	FFOISO	A logic 1 indicates an isochronous endpoint. A logic 0 indicates a bulk or interrupt endpoint.
3 to 0	FFOSZ[3:0]	Selects the FIFO size according to Table 5

12.1.2 Write/Read Device Address

This command is used to set the USB assigned address in the Address Register and enable 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 micro) is not altered by the bus reset. In response to the standard USB request Set Address the 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 (Hex): B6/B7 — write/read Address Register

Transaction — write/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	A logic 1 enables the device.
6 to 0	DEVADR[6:0]	This field specifies the USB device address.

12.1.3 Write/Read Mode Register

This command is used to access the ISP1181A 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, where all errors and Not Acknowledge (NAK) conditions will generate an interrupt.

Code (Hex): B8/B9 — write/read Mode Register

Transaction — write/read 1 byte

Table 18: Mode Register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	DMAWD	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	DMAWD	A logic 1 selects 16-bit DMA bus width (bus configuration modes 0 and 2). A logic 0 selects 8-bit DMA bus width. Bus reset value: unchanged.
6	-	reserved
5	GOSUSP	Writing a logic 1 followed by a logic 0 will activate ‘suspend’ mode.
4	-	reserved
3	INTENA	A logic 1 enables all interrupts. Bus reset value: unchanged.
2	DBGMOD	A logic 1 enables debug mode. where all NAKs and errors will generate an interrupt. A logic 0 selects normal operation, where interrupts are generated on every ACK (bulk endpoints) or after every data transfer (isochronous endpoints). Bus reset value: unchanged.
1	-	reserved
0	SOFTCT	A 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 Write/Read Hardware Configuration

This command is used to access the Hardware Configuration Register, which consists of 2 bytes. The first (lower) byte contains the device configuration and control values, the second (upper) byte holds the 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 'suspend' state, output clock frequency, DMA operating mode and pin configurations (polarity, signalling mode).

Code (Hex): BA/BB — write/read Hardware Configuration Register

Transaction — write/read 2 bytes

Table 20: Hardware Configuration Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved	EXTPUL	NOLAZY	CLKRUN	CLKDIV[3:0]			
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	EOTPOL	WKUPCS	PWROFF	INTLVL	INTPOL
Reset	0	1	0	0	0	0	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	A logic 1 indicates that an external 1.5 kΩ pull-up resistor is used on pin D+ and that SoftConnect is not used. Bus reset value: unchanged.
13	NOLAZY	A logic 1 disables output on pin CLKOUT of the LazyClock frequency (100 kHz ± 50 %) during 'suspend' state. A logic 0 causes pin CLKOUT to switch to LazyClock output after approximately 2 ms delay, following the setting of bit GOSUSP in the Mode Register. Bus reset value: unchanged.
12	CLKRUN	A logic 1 indicates that the internal clocks are always running, even during 'suspend' state. A logic 0 switches off the internal oscillator and PLL, when they are not needed. During 'suspend' state this bit must be made logic 0 to meet the 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	CLKDIV[3:0]	This field specifies the clock division factor N, which controls the clock frequency on output CLKOUT. The output frequency in MHz is given by $48/(N+1)$. The clock frequency range is 3 MHz to 48 MHz (N = 0 to 15). with a reset value of 12 MHz (N = 3). The hardware design guarantees no glitches during frequency change. Bus reset value: unchanged.
7	DAKOLY	A logic 1 selects DACK-only DMA mode. A logic 0 selects 8237 compatible DMA mode. Bus reset value: unchanged.

Table 21: Hardware Configuration Register: bit description…*continued*

Bit	Symbol	Description
6	DRQPOL	Selects DREQ signal polarity (0 = active LOW, 1 = active HIGH). Bus reset value: unchanged.
5	DAKPOL	Selects DACK signal polarity (0 = active LOW, 1 = active HIGH). Bus reset value: unchanged.
4	EOTPOL	Selects EOT signal polarity (0 = active LOW, 1 = active HIGH). Bus reset value: unchanged.
3	WKUPCS	A logic 1 enables remote wake-up via a LOW level on input $\overline{CS}$ (For wake-up on $\overline{CS}$ to work, V_{BUS} must be present.). Bus reset value: unchanged.
2	PWROFF	A logic 1 enables powering-off during 'suspend' state. Output SUSPEND is configured as a power switch control signal for external devices (HIGH during 'suspend'). This value should always be initialized to logic 1. Bus reset value: unchanged.
1	INTLVL	Selects the interrupt signalling mode on output INT (0 = level, 1 = pulsed). In pulsed mode an interrupt produces an 166 ns pulse. See Section 13 for details. Bus reset value: unchanged.
0	INTPOL	Selects INT signal polarity (0 = active LOW, 1 = active HIGH). Bus reset value: unchanged.

12.1.5 Write/Read Interrupt Enable Register

This command is used to individually enable/disable 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, which consists of 4 bytes. The bit allocation is given in Table 22.

Code (Hex): C2/C3 — write/read Interrupt Enable Register

Transaction — write/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		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 to 10	IEP14 to IEP1	A logic 1 enables interrupts from the indicated endpoint.
9	IEP0IN	A logic 1 enables interrupts from the control IN endpoint.
8	IEP0OUT	A logic 1 enables interrupts from the control OUT endpoint.
7, 6	-	reserved
5	IEPSOF	A logic 1 enables 1 ms interrupts upon detection of Pseudo SOF.
4	IESOF	A logic 1 enables interrupt upon SOF detection.
3	IEEOT	A logic 1 enables interrupt upon EOT detection.
2	IESUSP	A logic 1 enables interrupt upon detection of ‘suspend’ state.
1	IERESM	A logic 1 enables interrupt upon detection of a ‘resume’ state.
0	IERST	A logic 1 enables interrupt upon detection of a bus reset.

12.1.6 Write/Read DMA Configuration

This command defines the DMA configuration of ISP1181A and enables/disables DMA transfers. The command accesses the DMA Configuration Register, which consists of 2 bytes. The bit allocation is given in Table 24. A bus reset will clear bit DMAEN (DMA disabled), all other bits remain unchanged.

Code (Hex): F0/F1 — write/read DMA Configuration

Transaction — write/read 2 bytes

Table 24: 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]				DMAEN	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 25: DMA Configuration Register: bit description

Bit	Symbol	Description
15	CNTREN	A logic 1 enables the generation of an EOT condition, when the DMA Counter Register reaches zero. Bus reset value: unchanged.
14	SHORTP	A logic 1 enables short/empty packet mode. When receiving (OUT endpoint) a short/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	DMAEN	Writing a logic 1 enables DMA transfer, a logic 0 forces the end of an ongoing DMA transfer. Reading this bit indicates whether DMA is enabled (0 = DMA stopped, 1 = DMA enabled). 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.

12.1.7 Write/Read DMA Counter

This command accesses the DMA Counter Register, which consists of 2 bytes. The bit allocation is given in Table 26. 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 the programmed bit values.

The internal DMA counter is automatically reloaded from the DMA Counter Register when DMA is re-enabled (DMAEN = 1). See Section 12.1.6 for more details.

Code (Hex): F2/F3 — write/read DMA Counter Register

Transaction — write/read 2 bytes

Table 26: 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 27: 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)

12.1.8 Reset Device

This command resets the ISP1181A in the same way as an external hardware reset via input $\overline{\text{RESET}}$. All registers are initialized to their 'reset' values.

Code (Hex): F6 — reset the device

Transaction — none

12.2 Data flow commands

Data flow commands are used to manage the data transmission between the USB endpoints and the system microcontroller. Much of the data flow is initiated via an interrupt to the microcontroller. The data flow commands are used to access the endpoints and determine whether the 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 Write/Read Endpoint Buffer

This command is used to access 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. For 16-bit access the maximum number of words is (M + 1), with M given by (N + 1) DIV 2. After each read/write action the buffer pointer is automatically incremented by 1 (8-bit bus width) or by 2 (16-bit bus width).

In DMA access the first 2 bytes or the first word (the packet length) are skipped: transfers start at the third byte or the second word of the endpoint buffer. When reading, the ISP1181A can detect the last byte/word via the EOP condition. When writing to a bulk/interrupt endpoint, the endpoint buffer must be completely filled before sending the data to the host. Exception: when a DMA transfer is stopped by an external EOT condition, the current buffer content (full or not) is sent 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 ISP1181A.

Code (Hex): 01 to 0F — write (control IN, endpoint 1 to 14)

Code (Hex): 10, 12 to 1F — read (control OUT, endpoint 1 to 14)

Transaction — write/read maximum N + 2 bytes (isochronous endpoint: N ≤ 1023, bulk/interrupt endpoint: N ≤ 32)

The data in the endpoint FIFO must be organized as shown in Table 28. Examples of endpoint FIFO access are given in Table 29 (8-bit bus) and Table 30 (16-bit bus).

Table 28: Endpoint FIFO organization

Byte # (8-bit bus)	Word # (16-bit bus)	Description
0	0 (lower byte)	packet length (lower byte)
1	0 (upper byte)	packet length (upper byte)
2	1 (lower byte)	data byte 1
3	1 (upper byte)	data byte 2
…	…	…
(N + 1)	M = (N + 1) DIV 2	data byte N

Table 29: Example of endpoint FIFO access (8-bit bus width)

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

Table 30: Example of endpoint FIFO access (16-bit bus width)

A0	Phase	Bus lines	Word #	Description
1	command	D[7:0]	-	command code (00H to 1FH)
		D[15:8]	-	ignored
0	data	D[15:0]	0	packet length
0	data	D[15:0]	1	data word 1 (data byte 2, data byte 1)
0	data	D[15:0]	2	data word 2 (data byte 4, data byte 3)
…	…	…	…	…

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 are only meaningful 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 Read Endpoint Status

This command is used to read the status of an endpoint FIFO. The command accesses the Endpoint Status Register, the bit allocation of which is shown in Table 31. Reading the Endpoint Status Register will clear the interrupt bit set for the corresponding endpoint in the Interrupt Register (see Table 48).

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

Code (Hex): 50 to 5F — read (control OUT, control IN, endpoint 1 to 14)

Transaction — read 1 byte

Table 31: 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 32: 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 to logic 1 by a Stall Endpoint command, cleared to logic 0 by an Unstall Endpoint command. The endpoint is automatically unstalled upon reception of a SETUP token.
6	EPFULL1	A logic 1 indicates that the secondary endpoint buffer is full.
5	EPFULL0	A logic 1 indicates that the primary endpoint buffer is full.
4	DATA_PID	This bit indicates the data PID of the next packet (0 = DATA PID, 1 = DATA1 PID).
3	OVERWRITE	This bit is set by hardware, a logic 1 indicating that a new Setup packet has overwritten the previous setup information, before it was acknowledged or before the endpoint was stalled. This bit is cleared by reading, if writing the setup data has finished. Firmware must check this bit before sending an Acknowledge Setup command or stalling the endpoint. Upon reading a logic 1 the firmware must stop ongoing setup actions and wait for a new Setup packet.
2	SETUPT	A logic 1 indicates that the buffer contains a Setup 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 Endpoint/Unstall Endpoint

These commands are used to stall or unstall an endpoint. The commands modify the content of the Endpoint Status Register (see Table 31).

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 restall 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 reinitialized. This flushes the buffer: if it is an OUT buffer it waits for a DATA 0 PID, if it is an IN buffer it writes a DATA 0 PID.

Code (Hex): 40 to 4F — stall (control OUT, control IN, endpoint 1 to 14)

Code (Hex): 80 to 8F — unstall (control OUT, control IN, endpoint 1 to 14)

Transaction — none

12.2.4 Validate Endpoint Buffer

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 (Hex): 61 to 6F — validate endpoint buffer (control IN, endpoint 1 to 14)

Transaction — none

12.2.5 Clear Endpoint Buffer

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 (Hex): 70, 72 to 7F — clear endpoint buffer (control OUT, endpoint 1 to 14)

Transaction — none

12.2.6 Check Endpoint Status

This command is used to check 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 33.

Code (Hex): D0 to DF — check status (control OUT, control IN, endpoint 1 to 14)

Transaction — write/read 1 byte

Table 33: 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 34: 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	A logic 1 indicates that the secondary endpoint buffer is full.
5	EPFULL0	A 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).

Table 34: Endpoint Status Image Register: bit description…*continued*

Bit	Symbol	Description
3	OVERWRITE	This bit is set by hardware, a logic 1 indicating that a new Setup packet has overwritten the previous setup information, before it was acknowledged or before the endpoint was stalled. This bit is cleared by reading, if writing the setup data has finished. Firmware must check this bit before sending an Acknowledge Setup command or stalling the endpoint. Upon reading a logic 1 the firmware must stop ongoing setup actions and wait for a new Setup packet.
2	SETUPT	A logic 1 indicates that the buffer contains a Setup 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

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 (Hex): F4 — acknowledge setup

Transaction — none

12.3 General commands

12.3.1 Read Endpoint Error Code

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 35.

Code (Hex): A0 to AF — read error code (control OUT, control IN, endpoint 1 to 14)

Transaction — read 1 byte

Table 35: 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 36: Error Code Register: bit description

Bit	Symbol	Description
7	UNREAD	A 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).

Table 36: Error Code Register: bit description…*continued*

Bit	Symbol	Description
5	-	reserved
4 to 1	ERROR[3:0]	Error code. For error description, see Table 37.
0	RTOK	A logic 1 indicates that data was received or transmitted successfully.

Table 37: 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

12.3.2 Unlock Device

This command unlocks the ISP1181A from write-protection mode after a ‘resume’. In ‘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 ‘suspend’ state, the firmware must unlock the registers and FIFOs via 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 38.

Code (Hex): B0 — unlock the device

Transaction — write 2 bytes (unlock code)

Table 38: 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 39: Lock Register: bit description

Bit	Symbol	Description
15 to 0	UNLOCK[15:0]	Sending data AA37H unlocks the internal registers and FIFOs for writing, following a ‘resume’.

12.3.3 Write/Read Scratch Register

This command accesses the 16-bit Scratch Register, which can be used by the firmware to save and restore information, for example, the device status before powering down in ‘suspend’ state. The register bit allocation is given in Table 40.

Code (Hex): B2/B3 — write/read Scratch Register

Transaction — write/read 2 bytes

Table 40: Scratch Information Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved	SFIRH[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
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 41: Scratch Information Register: bit description

Bit	Symbol	Description
15	-	reserved; must be logic 0
14 to 8	SFIRH[6:0]	Scratch Information Register (high byte)
7 to 0	SFIRL[7:0]	Scratch Information Register (low byte)

12.3.4 Read Frame Number

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 42.

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

Code (Hex): B4 — read frame number

Transaction — read 1 or 2 bytes

Table 42: Frame Number Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved					SOFRH[2:0]		
Reset[1]	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 43: 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 44: Example of Frame Number Register access (8-bit bus width)

A0	Phase	Bus lines	Byte #	Description
1	command	D[7:0]	-	command code (B4H)
0	data	D[7:0]	0	frame number (lower byte)
0	data	D[7:0]	1	frame number (upper byte)

Table 45: Example of Frame Number Register access (16-bit bus width)

A0	Phase	Bus lines	Word #	Description
1	command	D[7:0]	-	command code (B4H)
		D[15:8]	-	ignored
0	data	D[15:0]	0	frame number

12.3.5 Read Chip ID

This command reads the chip identification code and hardware version number. The firmware must check this information to determine the supported functions and features. This command accesses the Chip ID Register, which is shown in Table 46.

Code (Hex): B5 — read chip ID

Transaction — read 2 bytes

Table 46: Chip ID Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	CHIPIDH[7:0]							
Reset	81H							
Access	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Symbol	CHIPIDL[7:0]							
Reset	41H							
Access	R	R	R	R	R	R	R	R

Table 47: Chip ID Register: bit description

Bit	Symbol	Description
15 to 8	CHIPIDH[7:0]	chip ID code (81H)
7 to 0	CHIPIDL[7:0]	silicon version (41H, with 41H representing the BCD encoded version number)

12.3.6 Read Interrupt Register

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 48. Bit BUSTATUS is used to verify the current bus status in the interrupt service routine. Interrupts are enabled via the Interrupt Enable Register, see Section 12.1.5.

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

Code (Hex): C0 — read interrupt register

Transaction — read 4 bytes

Table 48: 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	reserved	PSOF	SOF	EOT	SUSPND	RESUME	RESET
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 49: Interrupt Register: bit description

Bit	Symbol	Description
31 to 24	-	reserved
23 to 10	EP14 to EP1	A logic 1 indicates the interrupt source(s): endpoint 14 to 1.
9	EP0IN	A logic 1 indicates the interrupt source: control IN endpoint.

Table 49: Interrupt Register: bit description...*continued*

Bit	Symbol	Description
8	EP0OUT	A logic 1 indicates the interrupt source: control OUT endpoint.
7	BUSTATUS	It monitors the current USB bus status (0 = awake, 1 = suspend).
6	-	reserved
5	PSOF	A logic 1 indicates that an interrupt is issued every 1 ms because of the Pseudo SOF; after 3 missed SOFs 'suspend' state is entered.
4	SOF	A logic 1 indicates that a SOF condition was detected.
3	EOT	A logic 1 indicates that an internal EOT condition was generated by the DMA Counter reaching zero.
2	SUSPND	A logic 1 indicates that an 'awake' to 'suspend' change of state was detected on the USB bus.
1	RESUME	A logic 1 indicates that a 'resume' state was detected.
0	RESET	A logic 1 indicates that a bus reset condition was detected.

13. Interrupts

Figure 8 shows the interrupt logic of the ISP1181A. 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 or not an event will generate an interrupt.

Interrupts can be masked globally by means of the INTENA bit of the Mode Register (see Table 19).

The active level and signalling mode of the INT output is controlled by the INTPOL and INTLVL bits of the Hardware Configuration Register (see Table 21). Default settings after reset are active LOW and 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.

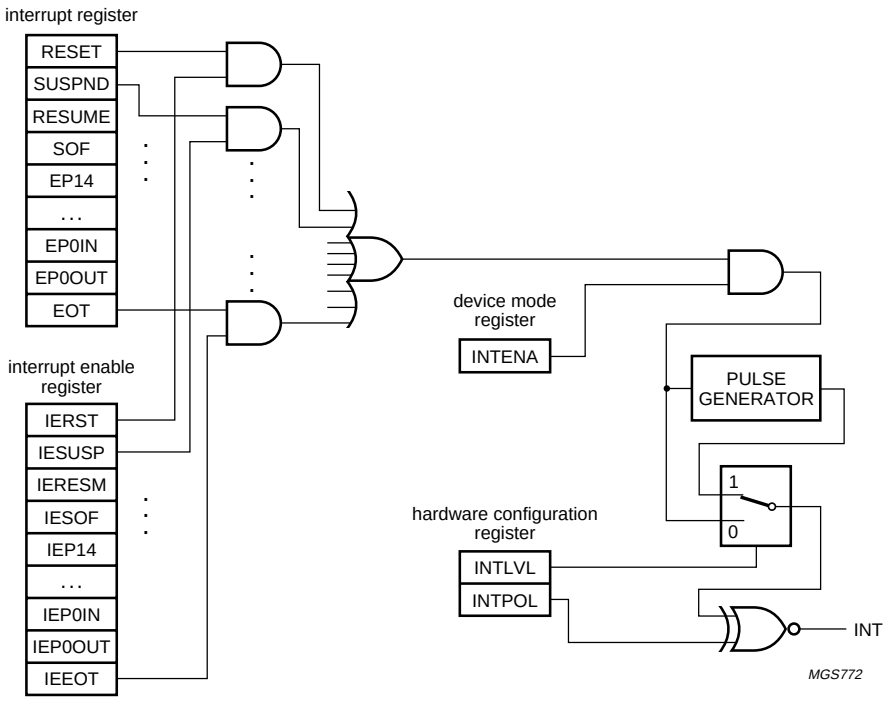


Fig 8. Interrupt logic.

Bits RESET, RESUME, EOT and SOF are cleared upon reading the Interrupt Register. The endpoint bits (EP0OUT to EP14) are cleared by reading the associated Endpoint Status Register.

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

SETUP and OUT token interrupts are generated after ISP1181A has acknowledged the associated data packet. In bulk transfer mode, the ISP1181A will issue interrupts for every ACK received for an OUT token or transmitted for an IN token.

In isochronous mode, an interrupt is issued upon each packet transaction. The firmware must take care of timing synchronization with the host. This can be done via the Pseudo Start-Of-Frame (PSOF) interrupt, enabled via bit IEPSOF in the Interrupt Enable Register. If a Start-Of-Frame is lost, PSOF interrupts are generated every 1 ms. This allows the firmware to keep data transfer synchronized with the host. After 3 missed SOF events the ISP1181A will enter ‘suspend’ state.

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

14. Power supply

The ISP1181A 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 $V_{reg(3.3)}$ for connecting an external pull-up resistor on USB connection D+. See Figure 9.

The ISP1181A can also be operated from a 3.0 V to 3.6 V supply, as shown in Figure 10. In this case, the internal voltage regulator is disabled and pin $V_{reg(3.3)}$ must be connected to V_{CC}.

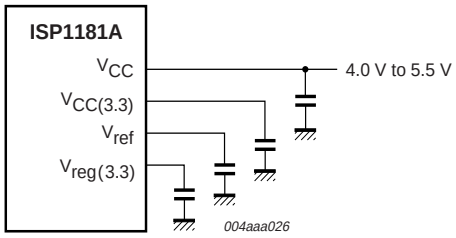


Fig 9. ISP1181A with a 4.0 V to 5.5 V supply.

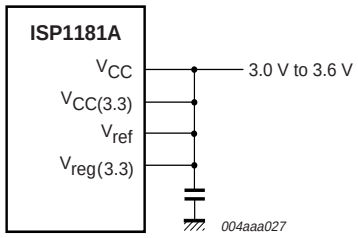


Fig 10. ISP1181A with a 3.0 V to 3.6 V supply.

15. Crystal oscillator and LazyClock

The ISP1181A has a crystal oscillator designed for a 6 MHz parallel-resonant crystal (fundamental). A typical circuit is shown in Figure 11. Alternatively, an external clock signal of 6 MHz can be applied to input XTAL1, while leaving output XTAL2 open.

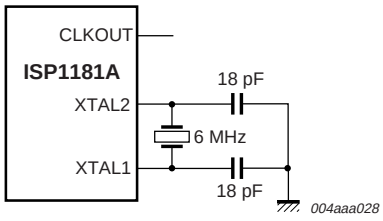


Fig 11. Typical oscillator circuit.

The 6 MHz oscillator frequency is multiplied to 48 MHz by an internal PLL. This frequency is used to generate a programmable clock output signal at pin CLKOUT, ranging from 3 MHz to 48 MHz.

In ‘suspend’ state the normal CLKOUT signal is not available, because the crystal oscillator and the PLL are switched off to save power. Instead, the CLKOUT signal can be switched to the LazyClock frequency of 100 kHz ± 50 %.

The oscillator operation and the CLKOUT frequency are controlled via the Hardware Configuration Register, as shown in Figure 12. The following bits are involved:

- CLKRUN switches the oscillator on and off
- CLKDIV[3:0] is the division factor determining the normal CLKOUT frequency
- NOLAZY controls the LazyClock signal output during ‘suspend’ state.

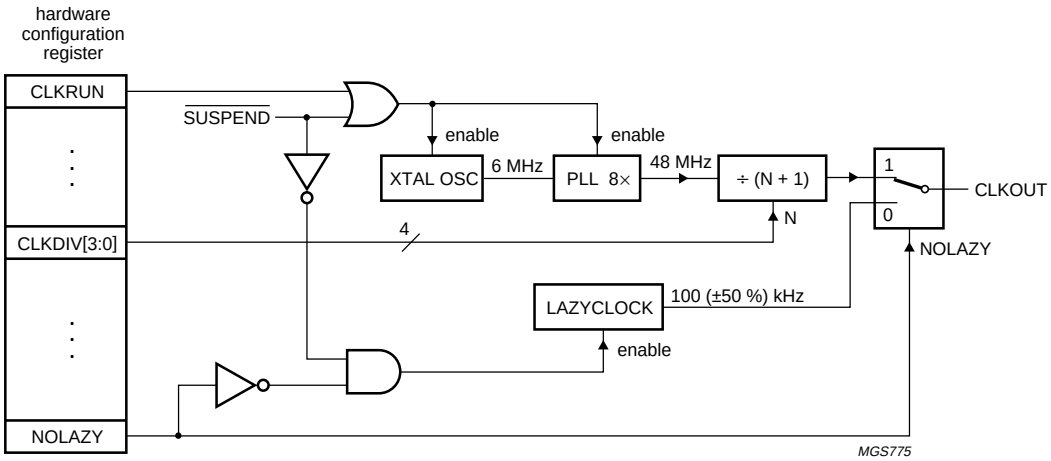
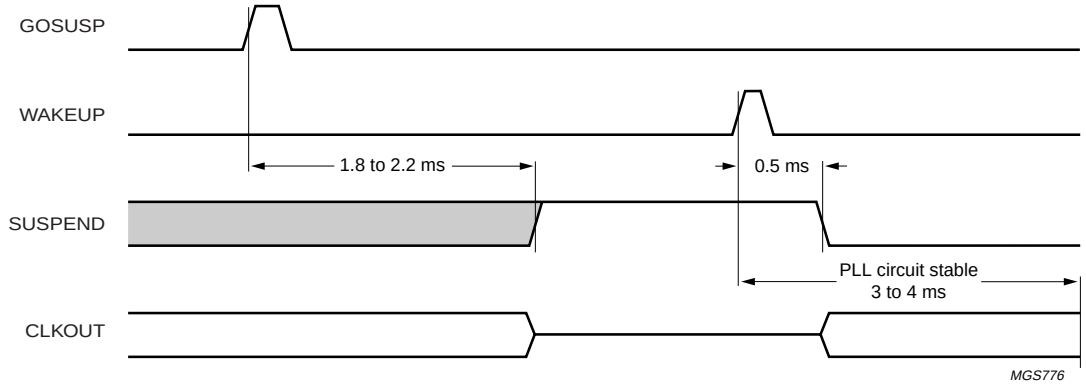


Fig 12. Oscillator and LazyClock logic.

When ISP1181A enters ‘suspend’ state (by setting and clearing bit GOSUSP in the Mode Register), outputs SUSPEND and CLKOUT change state after approximately 2 ms delay. When NOLAZY = 0, the clock signal on output CLKOUT does not stop, but changes to the 100 kHz ± 50 % LazyClock frequency.

When resuming from ‘suspend’ state by a positive pulse on input WAKEUP, output SUSPEND is cleared and the clock signal on CLKOUT is restarted after a 0.5 ms delay. The timing of the CLKOUT signal at ‘suspend’ and ‘resume’ is given in Figure 13.



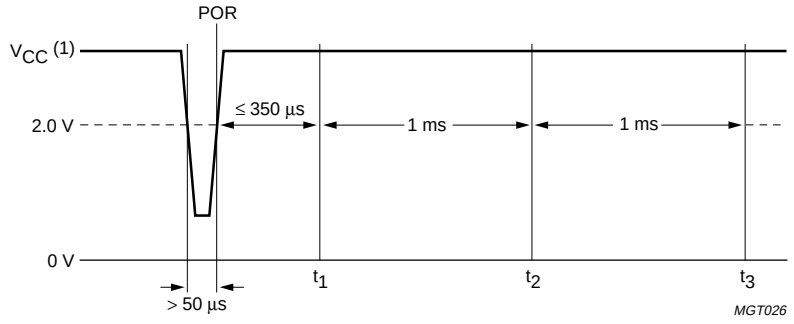
If enabled, the 100 kHz ± 50 % LazyClock frequency will be output on pin CLKOUT during ‘suspend’ state.

Fig 13. CLKOUT signal timing at ‘suspend’ and ‘resume’.

16. Power-on reset

The ISP1181A has an internal power-on reset (POR) circuit. Input pin $\overline{\text{RESET}}$ can be directly connected to V_{CC}. The clock signal on output CLKOUT starts 0.5 ms after power-on and normally requires 3 to 4 ms to stabilize.

The triggering voltage of the POR circuit is 2.0 V nominal. A POR is automatically generated when V_{CC} goes below the trigger voltage for a duration longer than 50 µs.



t_1: clock is running

t_2: BUS_CONF pins are sampled

t_3: registers are accessible

(1) Supply voltage (5 V or 3.3 V), connected externally to pin $\overline{\text{RESET}}$.

Fig 14. Power-on reset timing.

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

17. Limiting values

Table 50: Absolute maximum ratings

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

Symbol	Parameter	Conditions		Min	Max	Unit
V_{CC}	supply voltage			–0.5	+6.0	V
V_I	input voltage			–0.5	V_{CC} + 0.5	V
$I_{latchup}$	latch-up current	$V_I < 0$ or $V_I > V_{CC}$		-	100	mA
V_{esd}	electrostatic discharge voltage	$I_{LI} < 1$ µA	[1]	-	±2000	V
T_{stg}	storage temperature			–60	+150	°C
P_{tot}	total power dissipation	V_{CC} = 5.5 V		-	165	mW

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

18. Recommended operating conditions

Table 51: Recommended operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CC}	supply voltage	with regulator	4.0	5.0	5.5	V
		without regulator	3.0	3.3	3.6	V
V_I	input voltage		0	-	V_{CC}	V
$V_{I(AI/O)}$	input voltage on analog I/O pins (D+/D–)		0	-	3.6	V
$V_{O(od)}$	open-drain output pull-up voltage		0	-	V_{CC}	V
T_{amb}	ambient temperature		–40	-	+85	°C

19. Static characteristics

Table 52: Static characteristics; supply pins

V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$V_{reg(3.3)}$	regulated supply voltage	V_{CC} = 4.0 V to 5.5 V	[1]	3.0[2]	3.3	3.6	V
I_{CC}	operating supply current	V_{CC} = 5.0 V; T_{amb} = 25 °C		-	26	-	mA
		V_{CC} = 3.3 V; T_{amb} = 25 °C		-	22	-	mA
$I_{CC(susp)}$	suspend supply current	V_{CC} = 5.0 V; T_{amb} = 25 °C		-	-	20[3]	µA
		V_{CC} = 3.3 V; T_{amb} = 25 °C		-	-	70[3]	µA

[1] For 3.3 V operation, pin $V_{reg(3.3)}$ must be connected to pin $V_{CC(3.3)}$.

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

[3] External loading is not included.

Table 53: Static characteristics: digital pins

V_{CC} = 3.3 V ± 10 % or 5.0 V ± 10 %; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Input levels							
V_{IL}	LOW-level input voltage			-	-	0.8	V
V_{IH}	HIGH-level input voltage			2.0	-	-	V
Schmitt trigger inputs							
$V_{th(LH)}$	positive-going threshold voltage			1.4	-	1.9	V
$V_{th(HL)}$	negative-going threshold voltage			0.9	-	1.5	V
V_{hys}	hysteresis voltage			0.4	-	0.7	V
Output levels							
V_{OL}	LOW-level output voltage	I_{OL} = rated drive		-	-	0.4	V
		I_{OL} = 20 µA		-	-	0.1	V
V_{OH}	HIGH-level output voltage	I_{OH} = rated drive	[1]	2.4	-	-	V
Leakage current							
I_{LI}	input leakage current			-	-	±5	µA
Open-drain outputs							
I_{OZ}	OFF-state output current			-	-	±5	µA

[1] Not applicable for open-drain outputs.

Table 54: Static characteristics: analog I/O pins (D+, D–)[1]

V_{CC} = 3.3 V ±10 % or 5.0 V ±10 %; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Input levels						
V_{DI}	differential input sensitivity	$\|V_{I(D+)} - V_{I(D-)}\|$	0.2	-	-	V
V_{CM}	differential common mode voltage	includes V_{DI} range	0.8	-	2.5	V
V_{IL}	LOW-level input voltage		-	-	0.8	V
V_{IH}	HIGH-level input voltage		2.0	-	-	V
Output levels						
V_{OL}	LOW-level output voltage	R_L = 1.5 kΩ to +3.6 V	-	-	0.3	V
V_{OH}	HIGH-level output voltage	R_L = 15 kΩ to GND	2.8	-	3.6	V
Leakage current						
I_{LZ}	OFF-state leakage current		-	-	±10	µA
Capacitance						
C_{IN}	transceiver capacitance	pin to GND	-	-	20	pF
Resistance						
R_{PU}	pull-up resistance on D+	SoftConnect = ON	1	-	2	kΩ
Z_{DRV}[2]	driver output impedance	steady-state drive	29	-	44	Ω
Z_{INP}	input impedance		10	-	-	MΩ
Termination						
V_{TERM}[3]	termination voltage for upstream port pull-up (R_{PU})		3.0[4]	-	3.6	V

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

[2] Includes external resistors of 22 Ω ±1 % on both D+ and D–.

[3] This voltage is available at pin $V_{reg(3.3)}$.

[4] In ‘suspend’ mode the minimum voltage is 2.7 V.

20. Dynamic characteristics

Table 55: Dynamic characteristics

V_{CC} = 3.3 V ± 10 % or 5.0 V ± 10 %; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

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

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

Table 56: Dynamic characteristics: analog I/O pins (D+, D−)[1]

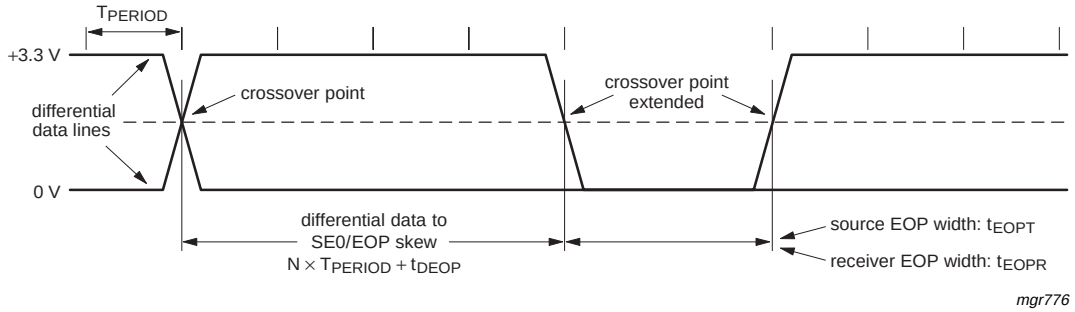
V_{CC} = 3.3 V ± 10 % or 5.0 V ± 10 %; V_{GND} = 0 V; T_{amb} = −40 °C to +85 °C; C_L = 50 pF; R_{PU} = 1.5 kΩ on D+ to V_{TERM}; unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Driver characteristics							
t_{FR}	rise time	C_L = 50 pF; 10 % to 90 % of $\lvert V_{OH} - V_{OL}\rvert$		4	-	20	ns
t_{FF}	fall time	C_L = 50 pF; 90 % to 10 % of $\lvert V_{OH} - V_{OL}\rvert$		4	-	20	ns
FRFM	differential rise/fall time matching (t_{FR}/t_{FF})		[2]	90	-	111.11	%
V_{CRS}	output signal crossover voltage		[2][3]	1.3	-	2.0	V
Data source timing							
t_{FEOPT}	source EOP width	see Figure 15	[3]	160	-	175	ns
t_{FDEOP}	source differential data-to-EOP transition skew	see Figure 15	[3]	−2	-	+5	ns
Receiver timing							
t_{JR1}	receiver data jitter tolerance for consecutive transitions	see Figure 16	[3]	−18.5	-	+18.5	ns
t_{JR2}	receiver data jitter tolerance for paired transitions	see Figure 16	[3]	−9	-	+9	ns
t_{FEOPR}	receiver SE0 width	accepted as EOP; see Figure 15	[3]	82	-	-	ns
t_{FST}	width of SE0 during differential transition	rejected as EOP; see Figure 17	[3]	-	-	14	ns

[1] Test circuit: see Figure 33.

[2] Excluding the first transition from Idle state.

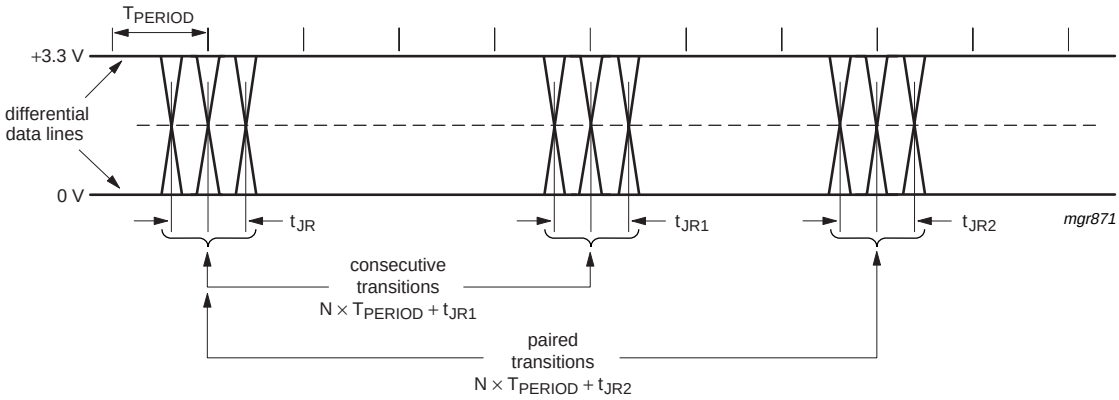
[3] Characterized only, not tested. Limits guaranteed by design.



T_{PERIOD} is the bit duration corresponding with the USB data rate.

Full-speed timing symbols have a subscript prefix 'F', low-speed timings a prefix 'L'.

Fig 15. Source differential data-to-EOP transition skew and EOP width.



T_{PERIOD} is the bit duration corresponding with the USB data rate.

Fig 16. Receiver differential data jitter.

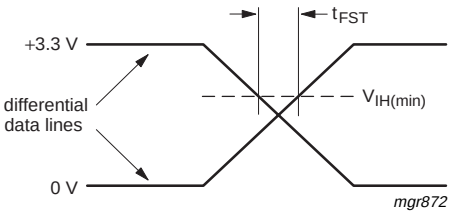


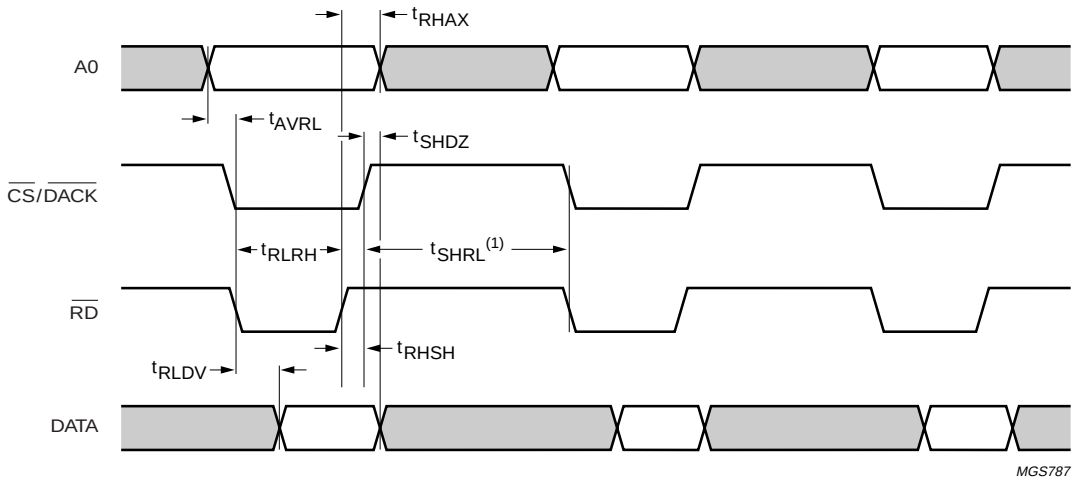
Fig 17. Receiver SE0 width tolerance.

20.1 Parallel I/O timing

Table 57: Dynamic characteristics: parallel interface timing

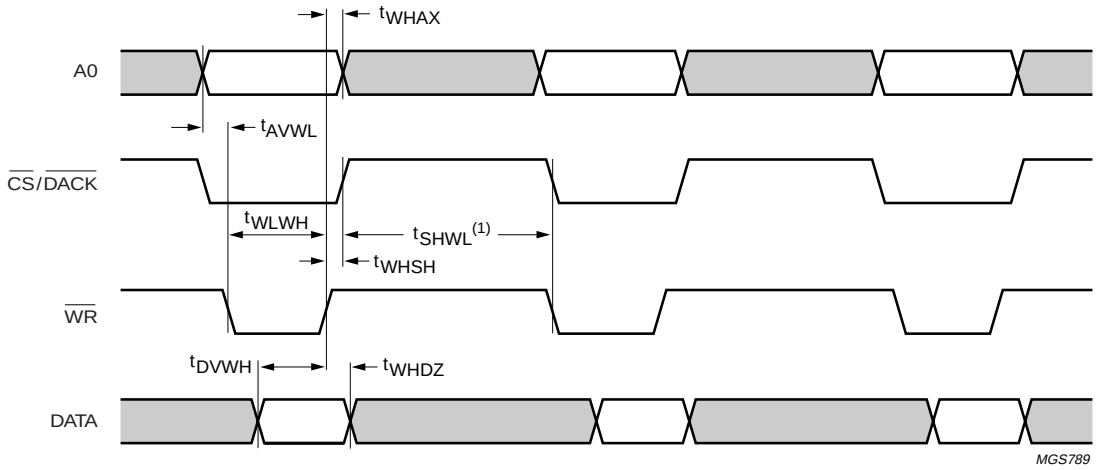
Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Read timing (see Figure 18)							
t_{RHAX}	address hold time after $\overline{RD}$ HIGH		0	-	0	-	ns
t_{AVRL}	address setup time before $\overline{RD}$ LOW		0	-	0	-	ns
t_{SHDZ}	data outputs high-impedance time after $\overline{CS}$ HIGH		-	3	-	3	ns
t_{RHSH}	chip deselect time after $\overline{RD}$ HIGH		0	-	0	-	ns
t_{RLRH}	$\overline{RD}$ pulse width		25	-	25	-	ns
t_{RLDV}	data valid time after $\overline{RD}$ LOW		-	22	-	22	ns
t_{SHRL}	$\overline{CS}$ HIGH until next ISP1181A $\overline{RD}$		60	-	120	-	ns
$t_{SHRL}+t_{RLRH}$	read cycle time		90	-	180	-	ns
Write timing (see Figure 19)							
t_{WHAX}	address hold time after $\overline{WR}$ HIGH		1	-	1	-	ns
t_{AVWL}	address setup time before $\overline{WR}$ LOW		0	-	0	-	ns
t_{SHWL}	$\overline{CS}$ HIGH until next ISP1181A $\overline{WR}$		60	-	120	-	ns
$t_{SHWL}+t_{WLWH}$	write cycle time		90/180[1]	-	180	-	ns
t_{WLWH}	$\overline{WR}$ pulse width		22	-	22	-	ns
t_{WHSH}	chip deselect time after $\overline{WR}$ HIGH		0	-	0	-	ns
t_{DVWH}	data setup time before $\overline{WR}$ HIGH		5	-	5	-	ns
t_{WHDZ}	data hold time after $\overline{WR}$ HIGH		3	-	3	-	ns
ALE timing (see Figure 20)							
t_{LH}	ALE pulse width		20	-	20	-	ns
t_{AVLL}	address setup time before ALE LOW		10	-	10	-	ns
t_{LLAX}	address hold time after ALE LOW	reading	0	10	0	10	ns
		writing	0	-	0	-	ns

[1] Commands Acknowledge Setup, Clear Buffer, Validate Buffer and Write Endpoint Configuration require 180 ns to complete.



(1) For t_{SHRL}, both $\overline{CS}$ and $\overline{RD}$ must be de-asserted.

Fig 18. Parallel interface read timing (I/O and 8237 compatible DMA).



(1) For t_{SHWL}, both $\overline{CS}$ and $\overline{WR}$ must be de-asserted.

Fig 19. Parallel interface write timing (I/O and 8237 compatible DMA).

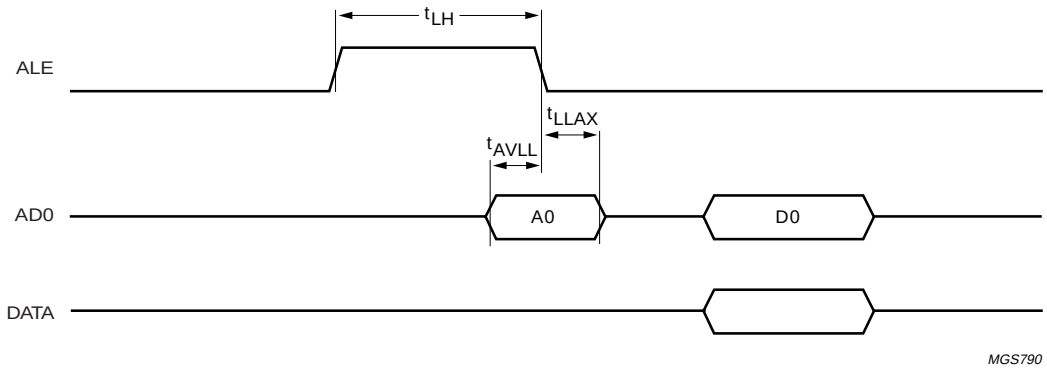


Fig 20. ALE timing.

20.2 Access cycle timing

Table 58: Dynamic characteristics: access cycle timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Write command + write data (see Figure 21 and Figure 22)							
$T_{cy(WC-WD)}$	cycle time for write command, then write data		100[1]	-	205	-	ns
$T_{cy(WD-WD)}$	cycle time for write data		90	-	205	-	ns
$T_{cy(WD-WC)}$	cycle time for write data, then write command		90	-	205	-	ns
Write command + read data (see Figure 23 and Figure 24)							
$T_{cy(WC-RD)}$	cycle time for write command, then read data		100[1]	-	205	-	ns
$T_{cy(RD-RD)}$	cycle time for read data		90	-	205	-	ns
$T_{cy(RD-WC)}$	cycle time for read data, then write command		90	-	205	-	ns

[1] Commands Acknowledge Setup, Clear Buffer, Validate Buffer and Write Endpoint Configuration require 180 ns to complete.

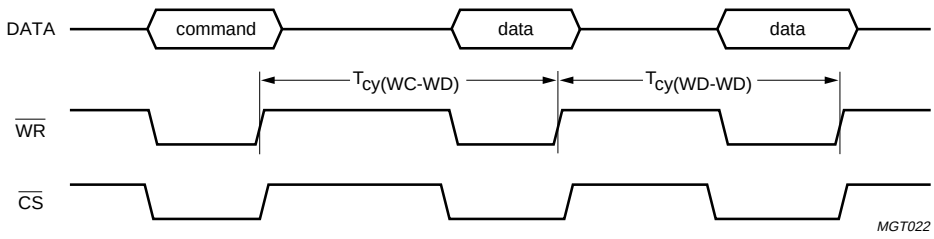
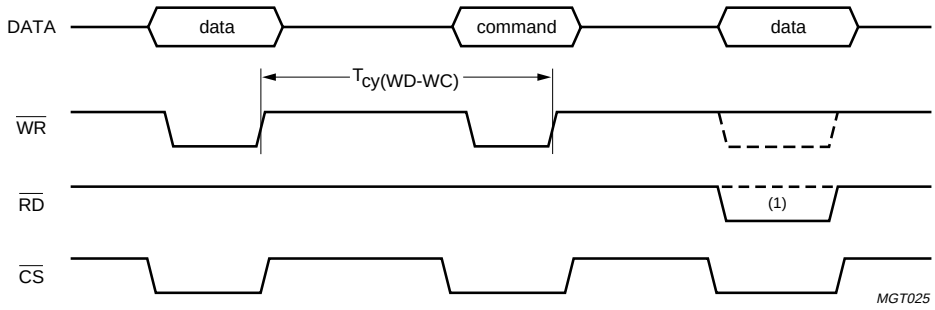


Fig 21. Write command + write data cycle timing.



(1) Example: read data.

Fig 22. Write data + write command cycle timing.

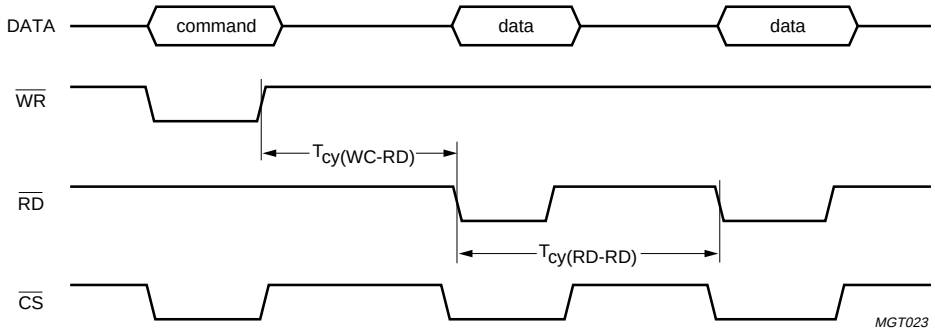
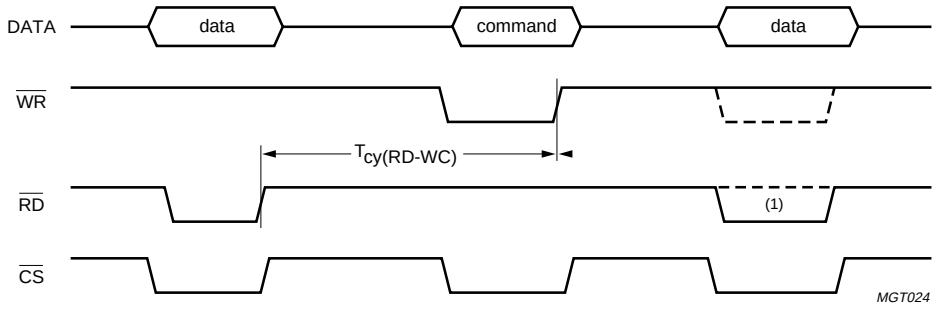


Fig 23. Write command + read data cycle timing.



(1) Example: read data.

Fig 24. Read data + write command cycle timing.

20.3 DMA timing: single-cycle mode

Table 59: Dynamic characteristics: single-cycle DMA timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
8237 compatible mode (see Figure 25)							
t_{ASRP}	DREQ off after DACK on		-	40	-	40	ns
$T_{cy(DREQ)}$	cycle time signal DREQ		90	-	180	-	ns
Read in DACK-only mode (see Figure 26)							
t_{ASRP}	DREQ off after DACK on		-	40	-	40	ns
t_{ASAP}	DACK pulse width		25	-	25	-	ns
t_{ASAP} + t_{APRS}	DREQ on after DACK off		90	-	180	-	ns
t_{ASDV}	data valid after DACK on		-	22	-	22	ns
t_{APDZ}	data hold after DACK off		-	3	-	3	ns
Write in DACK-only mode (see Figure 27)							
t_{ASRP}	DREQ off after DACK on		-	40	-	40	ns
t_{ASAP} + t_{APRS}	DREQ on after DACK off		90	-	180	-	ns
t_{DVAP}	data setup before DACK off		5	-	5	-	ns
t_{APDZ}	data hold after DACK off		3	-	3	-	ns
Single-cycle EOT (see Figure 28)							
t_{RSIH}	input $\overline{RD}$/$\overline{WR}$ HIGH after DREQ on		22	-	22	-	ns
t_{IHAP}	DACK off after input $\overline{RD}$/$\overline{WR}$ HIGH		0	-	0	-	ns
t_{EOT}	EOT pulse width	EOT on; DACK on; $\overline{RD}$/$\overline{WR}$ LOW	22	-	22	-	ns
t_{RLIS}	input EOT on after $\overline{RD}$ LOW		-	22	-	89	ns
t_{WLIS}	input EOT on after $\overline{WR}$ LOW		-	22	-	89	ns

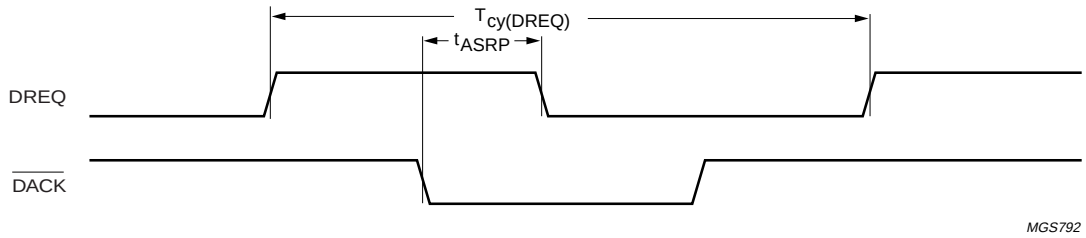


Fig 25. DMA timing in 8237 compatible mode.

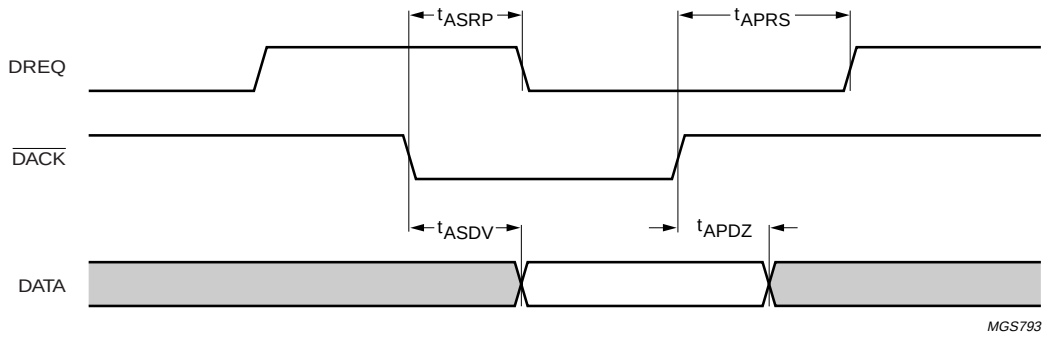


Fig 26. DMA read timing in DACK-only mode.

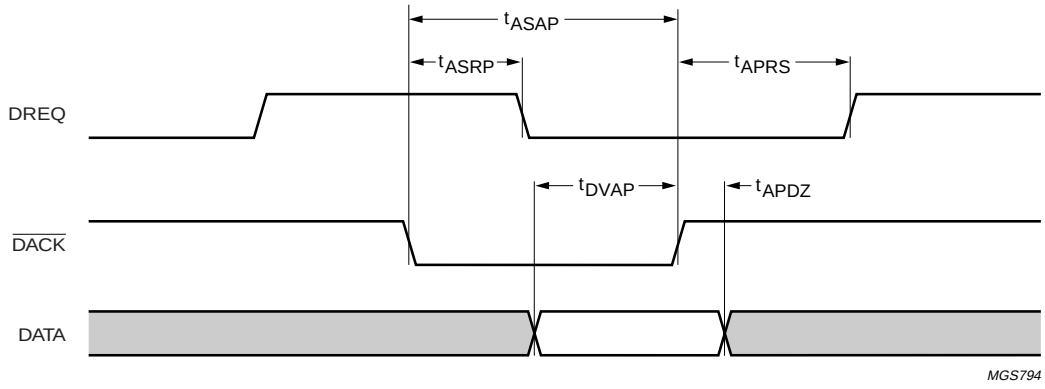
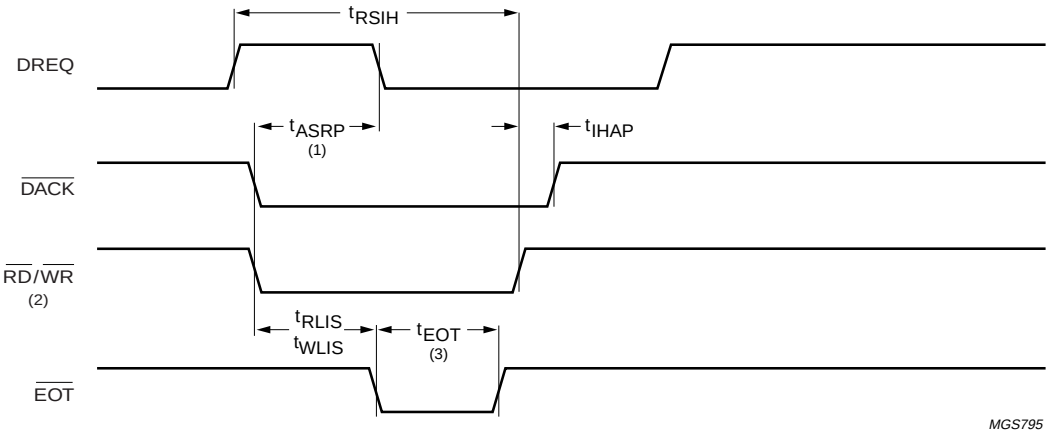


Fig 27. DMA write timing in DACK-only mode.



(1) t_{ASRP} starts from $\overline{DACK}$ or $\overline{RD}/\overline{WR}$ going LOW, whichever occurs later.

(2) The $\overline{RD}/\overline{WR}$ signals are not used in DACK-only DMA mode.

(3) The EOT condition is considered valid if $\overline{DACK}$, $\overline{RD}/\overline{WR}$ and $\overline{EOT}$ are all active (= LOW).

Fig 28. EOT timing in single-cycle DMA mode.

20.4 DMA timing: burst mode

Table 60: Dynamic characteristics: burst mode DMA timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Burst (see Figure 29)							
t_{RSIH}	input $\overline{RD}/\overline{WR}$ HIGH after DREQ on		22	-	22	-	ns
t_{ILRP}	DREQ off after input $\overline{RD}/\overline{WR}$ LOW		-	60	-	60	ns
t_{IHAP}	DACK off after input $\overline{RD}/\overline{WR}$ HIGH		0	-	0	-	ns
t_{IHIL}	DMA burst repeat interval (input $\overline{RD}/\overline{WR}$ HIGH to LOW)		90	-	180	-	ns
Burst EOT (see Figure 30)							
t_{EOT}	EOT pulse width	EOT on; DACK on; $\overline{RD}/\overline{WR}$ LOW	22	-	22	-	ns
t_{ISRP}	DREQ off after input EOT on		-	40	-	40	ns
t_{RLIS}	input EOT on after $\overline{RD}$ LOW		-	22	-	89	ns
t_{WLIS}	input EOT on after $\overline{WR}$ LOW		-	22	-	89	ns

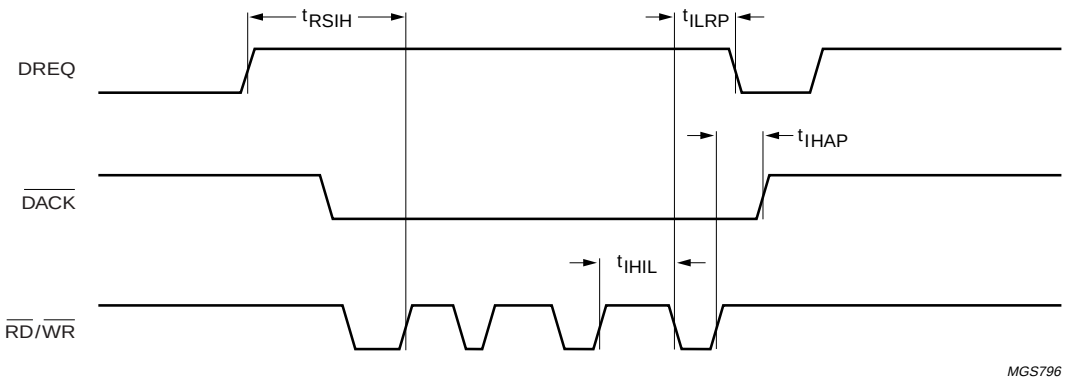
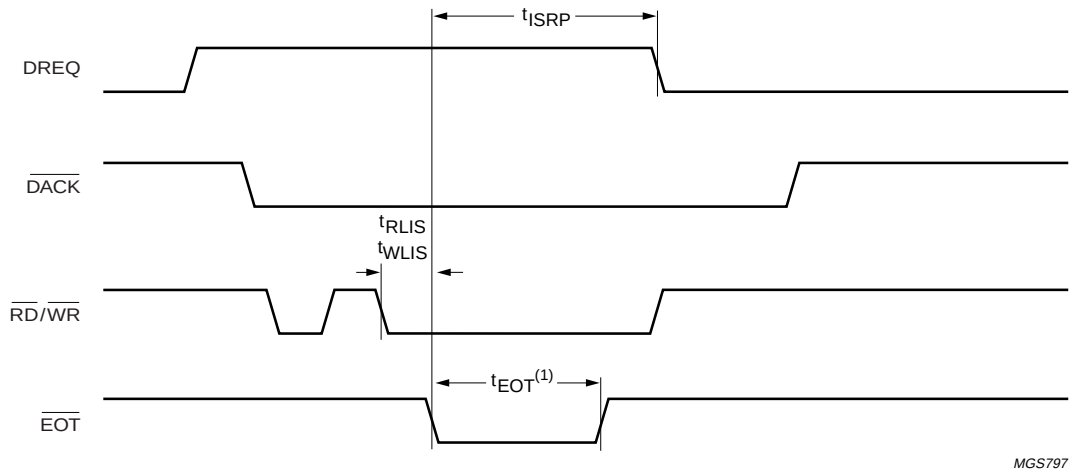


Fig 29. Burst mode DMA timing.

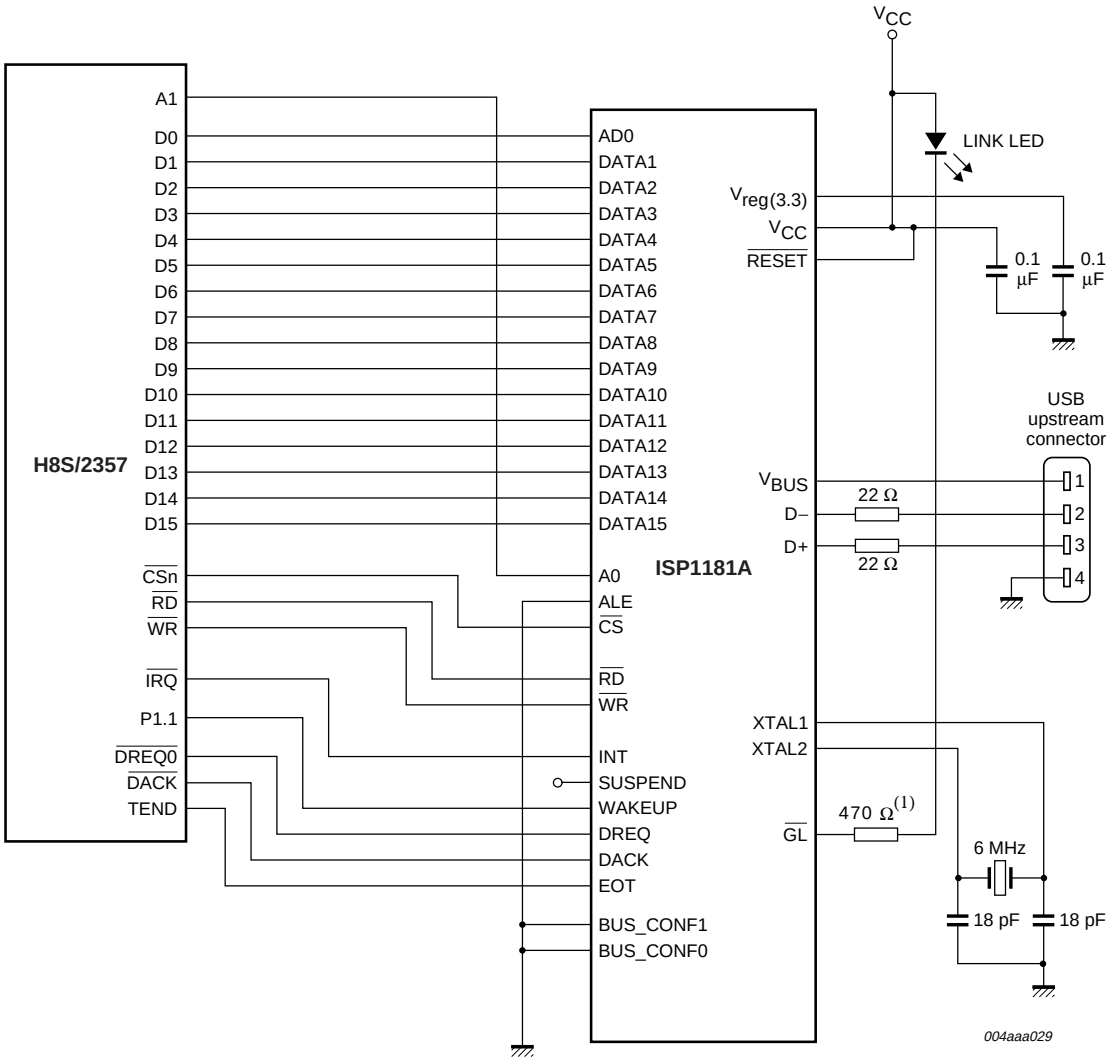


(1) The EOT condition is considered valid if $\overline{DACK}$, $\overline{RD}/\overline{WR}$ and $\overline{EOT}$ are all active (= LOW).

Fig 30. EOT timing in burst mode DMA.

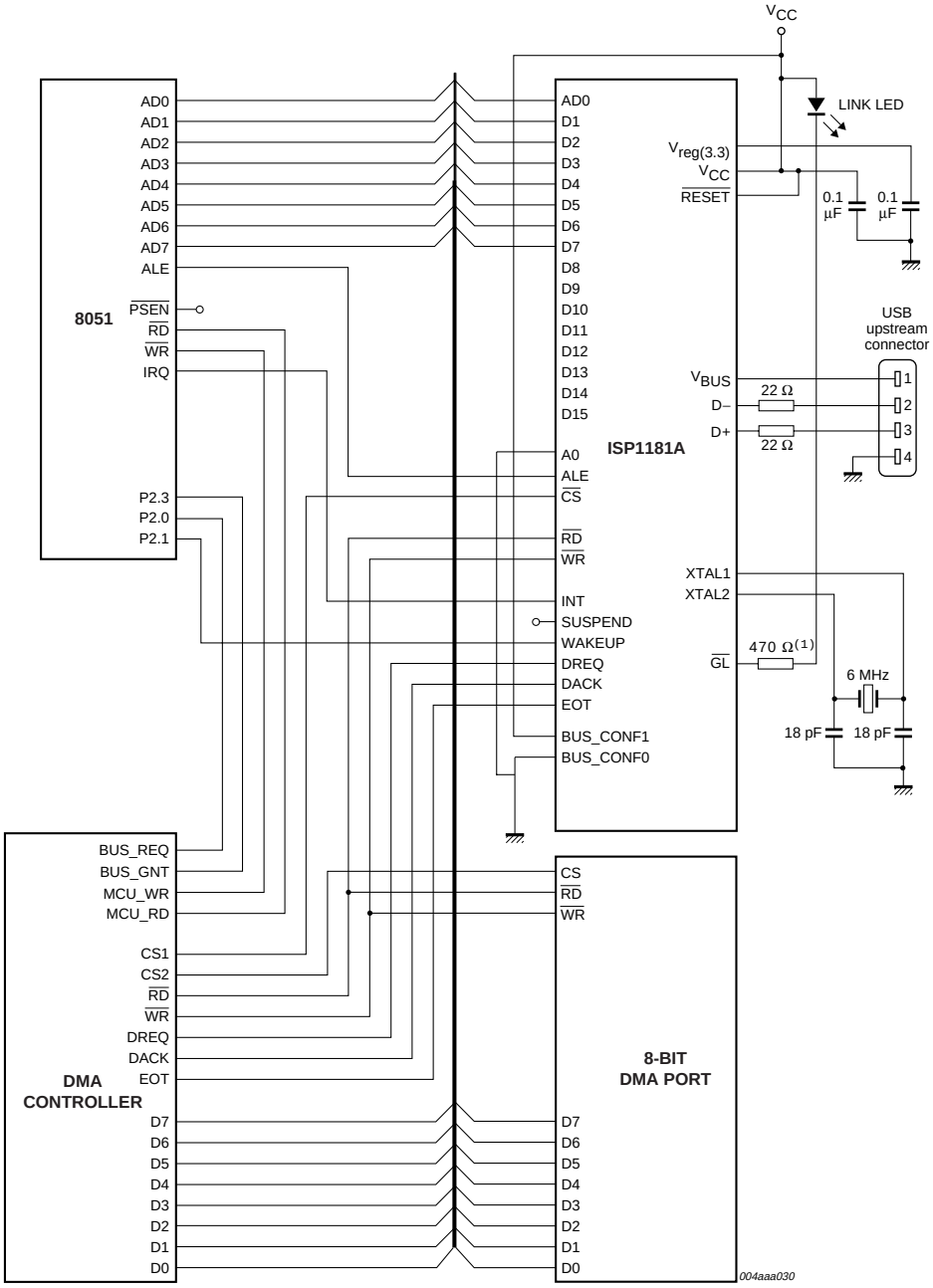
21. Application information

21.1 Typical interface circuits



(1) Use of a 470 Ω resistor assumes that V_{CC} is 5.0 V.

Fig 31. Typical interface circuit for bus configuration mode 0 (shared ports: 16-bit PIO, 16-bit DMA).



(1) Use of a 470 Ω resistor assumes that V_{CC} is 5.0 V.

Fig 32. Typical interface circuit for bus configuration mode 2 (shared ports: 8-bit PIO, 8-bit DMA).

21.2 Interfacing ISP1181A with an H8S/2357 microcontroller

This section gives a summary of the ISP1181A interface with a H8S/2357 (or compatible) microcontroller. Aspects discussed are: interrupt handling, address mapping, DMA and I/O port usage for suspend and remote wake-up control. A typical interface circuit is shown in Figure 31.

21.2.1 Interrupt handling

- **ISP1181A:** program the Hardware Configuration register to select an active LOW level for output INT (INTPOL = 0, see Table 20)
- **H8S/2357**: program the IRQ Sense Control Register (ISCRH and ISCRL) to specify low-level sensing for the IRQ input.

21.2.2 Address mapping in H8S/2357

The H8S/2357 bus controller partitions its 16 Mbyte address space into eight areas (0 to 7) of 2 Mbyte each. The bus controller will activate one of the outputs $\overline{CS0}$ to $\overline{CS7}$ when external address space for the associated area is accessed.

The ISP1181A can be mapped to any address area, allowing easy interfacing when the ISP1181A is the only peripheral in that area. If in the example circuit for bus configuration mode 0 (see Figure 31) the ISP1181A is mapped to address FFFF08H (in area 7), output $\overline{CS7}$ of the H8S/2357 can be directly connected to input $\overline{CS}$ of the ISP1181A.

The external bus specifications, bus width, number of access states and number of program wait states can be programmed for each address area. The recommended settings of H8S/2357 for interfacing the ISP1181A are:

- 8-bit bus in Bus Width Control Register (ABWCR)
- enable wait states in Access State Control Register (ASTCR)
- 1 program wait state in the Wait Control Register (WCRH and WCRL).

21.2.3 Using DMA

The ISP1181A can be configured for several methods of DMA with the H8S/2357 and other devices. The interface circuit in Figure 31 shows an example of the ISP1181A working with the H8S/2357 in single-address DACK-only DMA mode. External devices are not shown.

For single-address DACK-only mode, firmware must program the following settings:

- **ISP1181A**:
  - program the DMA Counter register with the total transfer byte count
  - program the Hardware Configuration Register to select active level LOW for DREQ and DACK
  - select the target endpoint and transfer direction
  - select DACK-only mode and enable DMA transfer.

21.2.4 Using H8S/2357 I/O Ports

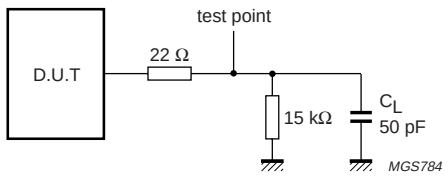
In the interface circuit of Figure 31 pin P1.1 of the H8S/2357 is configured as a general purpose output port. This pin drives the ISP1181A's WAKEUP input to generate a remote wake-up.

The H8S/2357 has 3 registers to configure port 1: Port 1 Data Direction Register (P1DDR), Port 1 Data Register (P1DR) and Port 1 Register (PORT1). Only registers P1DDR and P1DR must be configured, register PORT1 is only used to read the actual levels on the port pins.

- **H8S/2357:**
  - select pin P1.1 to be an output in register P1DDR
  - program the desired bit value for P1.1 in register P1DR.

22. Test information

The dynamic characteristics of the analog I/O ports (D+ and D–) as listed in Table 56, were determined using the circuit shown in Figure 33.



Load capacitance:

C_L = 50 pF (full-speed mode)

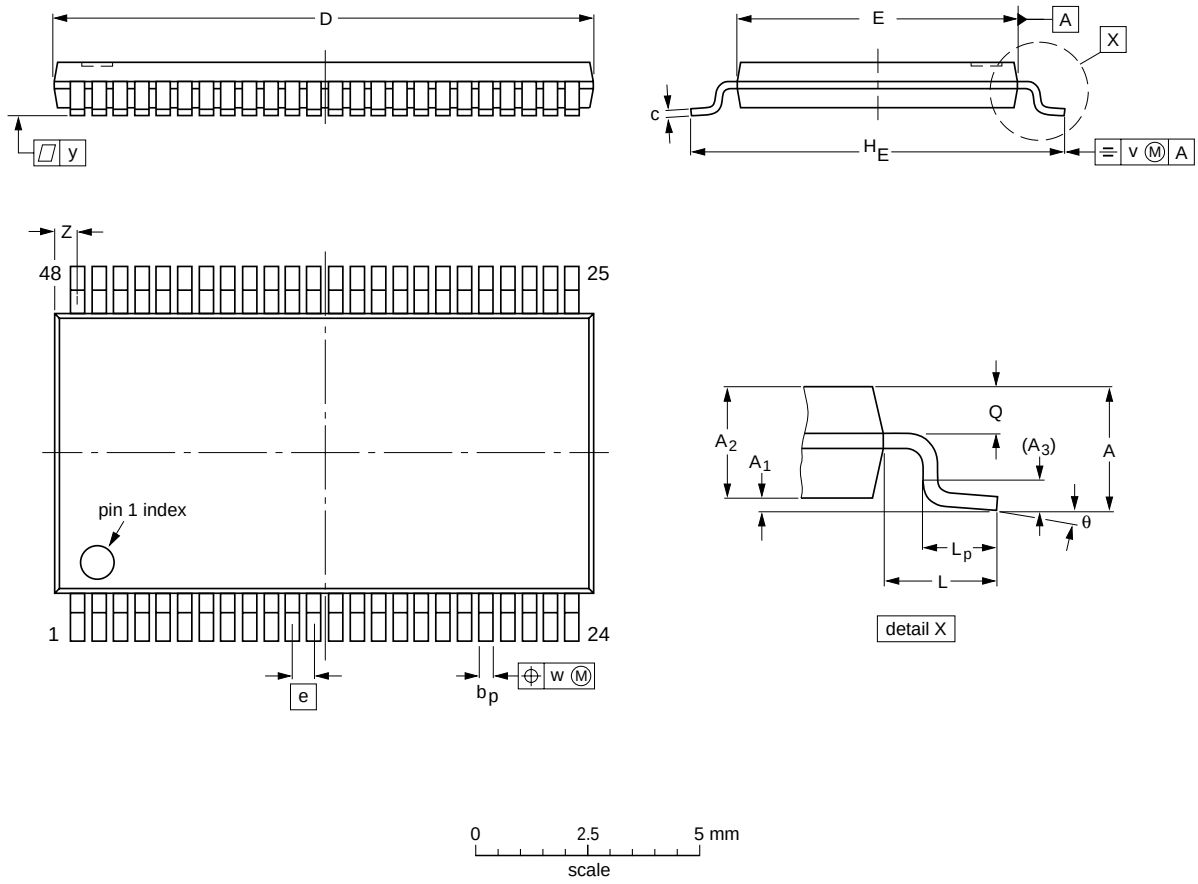
Speed:

full-speed mode only: internal 1.5 kΩ pull-up resistor on D+

Fig 33. Load impedance for D+ and D– pins.

23. Package outline

TSSOP48: plastic thin shrink small outline package; 48 leads; body width 6.1 mm **SOT362-1**



DIMENSIONS (mm are the original dimensions).

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	Z	θ
mm	1.2	0.15 0.05	1.05 0.85	0.25	0.28 0.17	**0.2** **0.1**	12.6 12.4	6.2 6.0	0.5	8.3 7.9	1	0.8 0.4	0.50 0.35	0.25	0.08	0.1	0.8 0.4	8^o 0^o

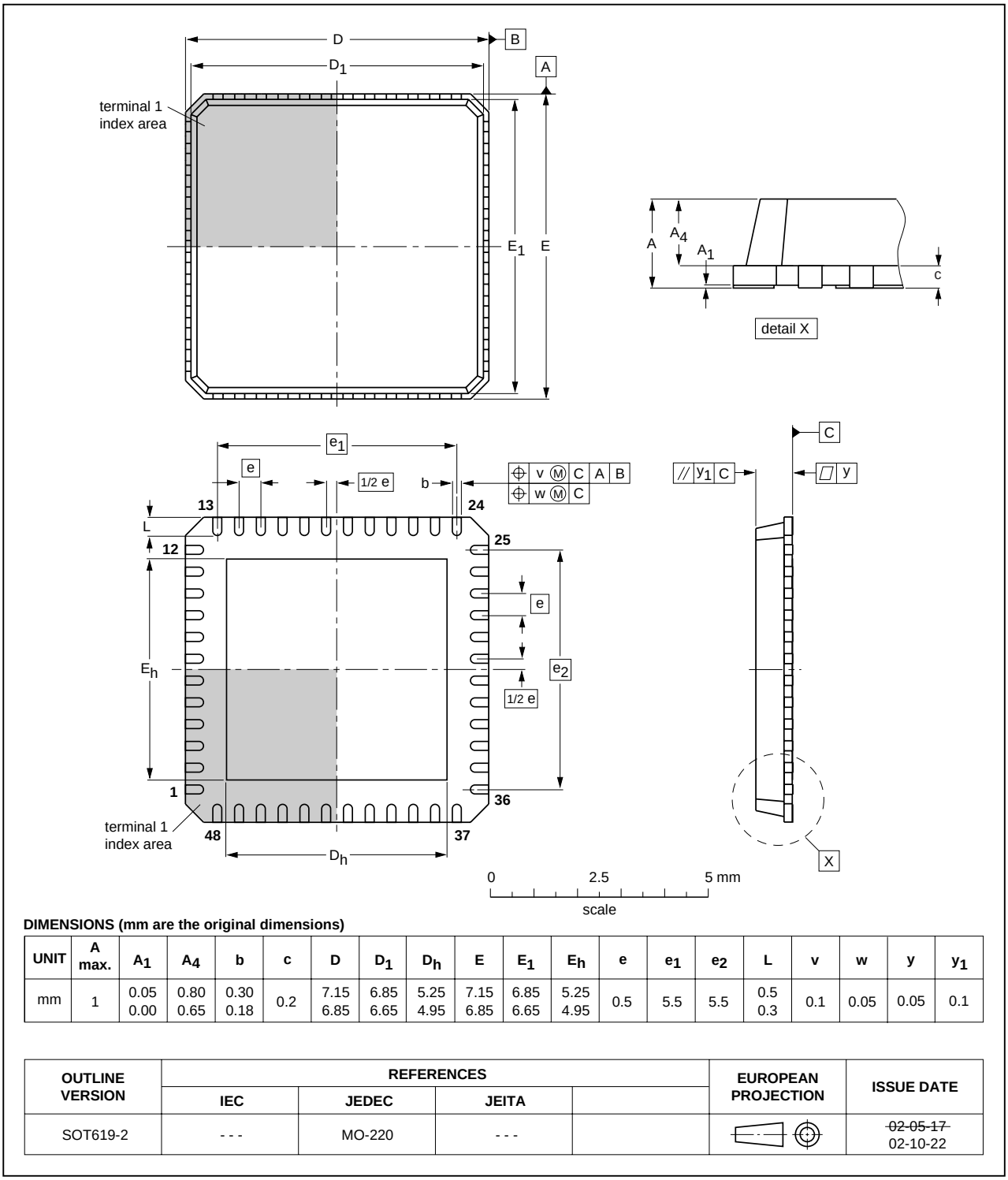
Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT362-1		MO-153				~~99-12-27~~ 03-02-19

Fig 34. TSSOP48 package outline.

HVQFN48: plastic thermal enhanced very thin quad flat package; no leads; 48 terminals; body 7 x 7 x 0.85 mm **SOT619-2**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_4	b	c	D	D_1	D_h	E	E_1	E_h	e	e_1	e_2	L	v	w	y	y_1
mm	1	0.05 0.00	0.80 0.65	0.30 0.18	0.2	7.15 6.85	6.85 6.65	5.25 4.95	7.15 6.85	6.85 6.65	5.25 4.95	0.5	5.5	5.5	0.5 0.3	0.1	0.05	0.05	0.1

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT619-2	- - -	MO-220	- - -			~~02-05-17~~ 02-10-22

Fig 35. HVQFN48 package outline.

24. Soldering

24.1 Introduction to soldering surface mount packages

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

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

24.2 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement. Driven by legislation and environmental forces the worldwide use of lead-free solder pastes is increasing.

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

Typical reflow peak temperatures range from 215 to 270 °C depending on solder paste material. The top-surface temperature of the packages should preferably be kept:

- below 225 °C (SnPb process) or below 245 °C (Pb-free process)
  - for all BGA, HTSSON..T and SSOP..T packages
  - for packages with a thickness ≥ 2.5 mm
  - for packages with a thickness < 2.5 mm and a volume ≥ 350 mm^3 so called thick/large packages.
- below 240 °C (SnPb process) or below 260 °C (Pb-free process) for packages with a thickness < 2.5 mm and a volume < 350 mm^3 so called small/thin packages.

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

24.3 Wave soldering

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

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

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

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.

- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;
  - smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

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

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

Typical dwell time of the leads in the wave ranges from 3 to 4 seconds at 250 °C or 265 °C, depending on solder material applied, SnPb or Pb-free respectively.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

24.4 Manual soldering

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

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

24.5 Package related soldering information

Table 61: Suitability of surface mount IC packages for wave and reflow soldering methods

Package[1]	Soldering method	
	Wave	Reflow[2]
BGA, HTSSON..T[3], LBGA, LFBGA, SQFP, SSOP..T[3], TFBGA, USON, VFBGA	not suitable	suitable
DHVQFN, HBCC, HBGA, HLQFP, HSO, HSOP, HSQFP, HSSON, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable[4]	suitable
PLCC[5], SO, SOJ	suitable	suitable
LQFP, QFP, TQFP	not recommended[5][6]	suitable
SSOP, TSSOP, VSO, VSSOP	not recommended[7]	suitable
CWQCCN..L[8], PMFP[9], WQCCN..L[8]	not suitable	not suitable

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

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

[3] These transparent plastic packages are extremely sensitive to reflow soldering conditions and must on no account be processed through more than one soldering cycle or subjected to infrared reflow soldering with peak temperature exceeding 217 °C ± 10 °C measured in the atmosphere of the reflow oven. The package body peak temperature must be kept as low as possible.

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

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

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

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

[8] Image sensor packages in principle should not be soldered. They are mounted in sockets or delivered pre-mounted on flex foil. However, the image sensor package can be mounted by the client on a flex foil by using a hot bar soldering process. The appropriate soldering profile can be provided on request.

[9] Hot bar soldering or manual soldering is suitable for PMFP packages.

25. Revision history

Table 62: Revision history

Rev	Date	CPCN	Description
05	**20041208**	**200412002**	**Product data (9397 750 13959)** Modifications: • Updated terminology from device to peripheral, where applicable • Updated to the current document style • Figure 3 “Pin configuration HVQFN48.”: made CS active LOW • Section 9.2 “Endpoint FIFO size”: removed “(512 bytes for non-isochronous FIFOs)” from the second last paragraph • Section 11 “Suspend and resume”: updated the complete section • Section 12.1.4 “Write/Read Hardware Configuration”: changed bit name from CKDIV to CLKDIV • Table 32 “Endpoint Status Register: bit description”: changed the description for bit 7 from “The endpoint automatically resumes upon reception of a SETUP token.” to “The endpoint is automatically unstalled upon reception of a SETUP token.” • Section 12.2.3 “Stall Endpoint/Unstall Endpoint”: updated second and third paragraphs: – second paragraph: changed “A stalled control endpoint automatically resumes when it receives a SETUP token, regardless of the packet content.” to “A stalled control endpoint is automatically unstalled when it receives a SETUP token, regardless of the packet content.” – third paragraph: changed “When a stalled endpoint resumes (either by the Unstall Endpoint command or by receiving a SETUP token), it is also re-initialized.” to “When a stalled endpoint is unstalled (either by the Unstall Endpoint command or by receiving a SETUP token), it is also re-initialized.” • Created a separate section for “Recommended operating conditions” • Removed old Section 19.1 “Timing symbols” • Table 57 “Dynamic characteristics: parallel interface timing”: updated values of parameters t_{RHAX} and t_{WHAX}. Added parameters t_{SHRL} and t_{SHWL}.
04	**20020716**	**-**	**Product data (9397 750 09613)**
03	**20020108**	**-**	**Product data (9397 750 09007)**
02	**20010919**	**-**	**Preliminary data (9397 750 08734)**
01	**20010725**	**-**	**Objective data (9397 750 08602)**

26. Data sheet status

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

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

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

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

27. Definitions

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

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

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

28. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

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

29. Trademarks

ACPI — is an open industry specification for PC power management, co-developed by Intel Corp., Microsoft Corp. and Toshiba
GoodLink — is a trademark of Koninklijke Philips Electronics N.V.
OnNow — is a trademark of Microsoft Corp.
SoftConnect — is a trademark of Koninklijke Philips Electronics N.V.
Zip — is a registered trademark of Iomega Corp.

Contact information

For additional information, please visit **http://www.semiconductors.philips.com**.
For sales office addresses, send e-mail to: **sales.addresses@www.semiconductors.philips.com**. **Fax: +31 40 27 24825**

9397 750 13959

Contents

Date of release: 08 December 2004 **Document order number: 9397 750 13959**

PHILIPS

Let's make things better.