ISP1181

Full-speed Universal Serial Bus interface device

Rev. 01 — 13 March 2000 **Objective specification**

1. General description

The ISP1181 is a Universal Serial Bus (USB) interface device which complies with *Universal Serial Bus Specification Rev. 1.1*. It provides full-speed USB communication capacity to microcontroller or microprocessor-based systems. The ISP1181 communicates with the system's microcontroller or microprocessor through a high-speed general-purpose parallel interface.

The fully autonomous Direct Memory Access (DMA) operation - auto download, auto repeat, auto execution - removes the need for the device to re-enable or re-initialize the DMA operation every time.

The modular approach to implementing a USB interface device allows the designer to select the optimum system microcontroller from the wide variety available. The ability to re-use 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 ISP1181 is ideally suited for application in many personal computer peripherals, such as printers, 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. 1.1* and most Device Class specifications
- High performance USB interface device with integrated Serial Interface Engine (SIE), FIFO memory, transceiver and 3.3 V voltage regulator
- Interrupt endpoint can be configured in 'rate feedback' mode
- 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 with integrated PLL for low EMI

- Controllable LazyClock (24 kHz) output during ‘suspend’
- Software controlled connection to the USB bus (SoftConnect™)
- Good USB connection indicator that blinks with traffic (GoodLink™)
- 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 to 5.5 V) with 5 V tolerant I/O pads
- Operating temperature range –40 to +85 °C
- 8 kV in-circuit ESD protection for lower cost of external components
- Full-scan design with high fault coverage
- Available in a TSSOP48 package.

3. Applications

- Personal digital assistant (PDA)
- Digital camera
- Communication device, e.g.
  - router
  - modem
- Printer
- Scanner.

4. Ordering information

Table 1: Ordering information

Type number	Package		
	Name	Description	Version
ISP1181DGG	TSSOP48	Plastic thin shrink small outline package; 48 leads; body width 6.1 mm	SOT362-1

5. Block diagram

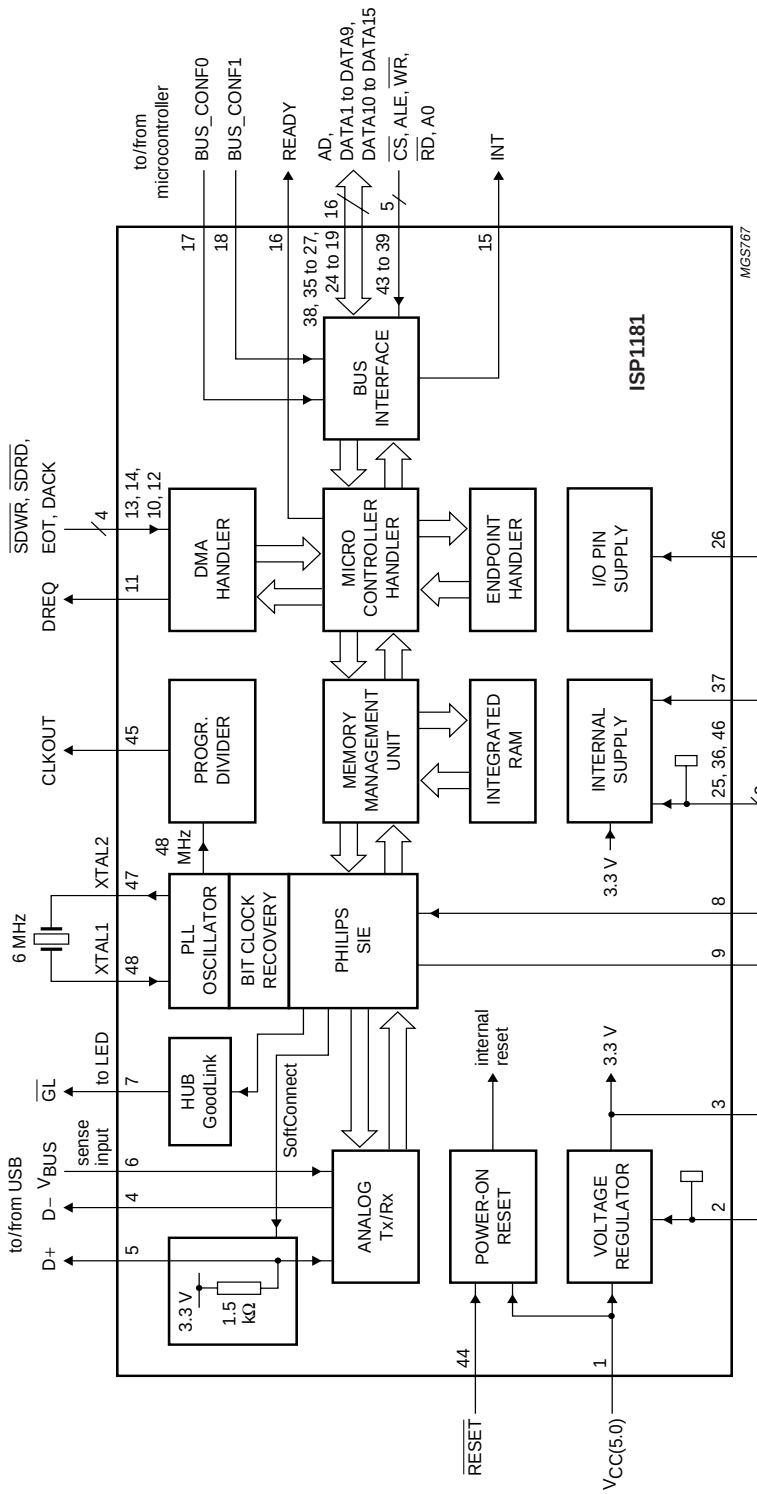


Fig 1. Block diagram.

6. Pinning information

6.1 Pinning

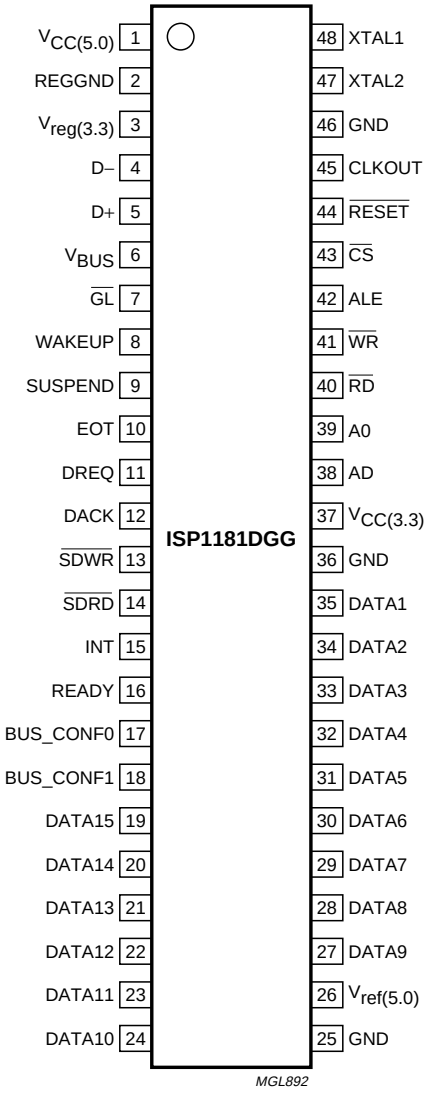


Fig 2. Pin configuration TSSOP48.

6.2 Pin description

Table 2: Pin description for TSSOP48

Symbol [1]	Pin	Type	Description
$V_{CC(5.0)}$	1	-	supply voltage (3.0 to 5.5 V)
REGGND	2	-	voltage regulator ground supply
$V_{reg(3.3)}$	3	-	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	AI/O	USB D– connection (analog)
D+	5	AI/O	USB D+ connection (analog)
V_{BUS}	6	I	V_{BUS} sensing input
$\overline{GL}$	7	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 330 Ω series resistor;
WAKEUP	8	I	wake-up input (edge triggered, LOW to HIGH); generates a remote wake-up from ‘suspend’ state
SUSPEND	9	O	‘suspend’ state indicator output (4 mA); used as power switch control output (active LOW) for powered-off application or as resume signal to the CPU (active HIGH) for powered-on application
EOT	10	I	End-Of-Transfer input (programmable polarity, see Table 23); used by the DMA controller to force the end of a DMA transfer by the ISP1181
DREQ	11	O	DMA request output (4 mA; programmable polarity, see Table 23); signals to the DMA controller that the ISP1181 wants to start a DMA transfer
DACK	12	I	DMA acknowledge input (programmable polarity, see Table 23); used by the DMA controller to signal the start of a DMA transfer requested by the ISP1181
$\overline{SDWR}$	13	I	DMA write strobe input; used only in bus configuration mode 1 (separate PIO and DMA ports)
$\overline{SDRD}$	14	I	DMA read strobe input; used only in bus configuration mode 1 (separate PIO and DMA ports)
INT	15	O	interrupt output; programmable polarity (active HIGH or LOW) and signalling (level or pulse); see Table 23
READY	16	O	I/O ready output; a LOW level indicates that ISP1181 is processing a previous command or data and is not ready for the next PIO command or data transfer; a HIGH level signals that ISP1181 will complete a PIO data transfer; applies only to a PIO port or a PIO port shared with a DMA port
BUS_CONF1	17	I	bus configuration selector; see Table 3
BUS_CONF0	18	I	bus configuration selector; see Table 3
DATA15	19	I/O	bit 15 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA14	20	I/O	bit 14 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)

Table 2: Pin description for TSSOP48

Symbol [1]	Pin	Type	Description
DATA13	21	I/O	bit 13 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA12	22	I/O	bit 12 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA11	23	I/O	bit 11 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA10	24	I/O	bit 10 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
GND	25	-	ground supply
$V_{ref(5.0)}$	26	-	I/O pin reference voltage (3.0 to 5.5 V)
DATA9	27	I/O	bit 9 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA8	28	I/O	bit 8 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA7	29	I/O	bit 7 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA6	30	I/O	bit 6 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA5	31	I/O	bit 5 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA4	32	I/O	bit 4 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA3	33	I/O	bit 3 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA2	34	I/O	bit 2 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
DATA1	35	I/O	bit 1 of D[15:0]; bi-directional data line (slew-rate controlled output, 4 mA)
GND	36	-	ground supply
$V_{CC(3.3)}$	37	-	supply voltage (3.0 to 3.6 V); leave this pin unconnected when using the internal regulator
AD	38	I/O	multiplexed bi-directional 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	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 HIGH (connect to V_{CC} or $V_{reg(3.3)}$)
$\overline{RD}$	40	I	read strobe input
$\overline{WR}$	41	I	write strobe input

Table 2: Pin description for TSSOP48

Symbol [1]	Pin	Type	Description
ALE	42	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 DGND) for a separate address/data bus configuration
$\overline{CS}$	43	I	chip select input
$\overline{RESET}$	44	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	O	programmable clock output (2 mA)
GND	46	-	ground supply
XTAL2	47	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	I	crystal oscillator input (6 MHz); connect a fundamental parallel-resonant crystal or an external clock source (leaving pin XTAL2 is unconnected)

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

7. Functional description

The ISP1181 is a full-speed USB interface device 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 ISP1181 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. Interrupt IN endpoints can be configured in rate-feedback mode.

The ISP1181 requires a single supply voltage of 3.0 to 5.5 V and has an internal 3.3 V voltage regulator for powering the analog USB transceiver. It supports bus-powered operation.

The ISP1181 operates on a 6 MHz oscillator frequency. A programmable clock output is available up to 48 MHz. During ‘suspend’ state the 24 kHz LazyClock frequency can be output.

7.1 Analog transceiver

The transceiver is compliant with *Universal Serial Bus Specification Rev. 1.1*. 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 high-speed USB devices) HIGH through a 1.5 kΩ pull-up resistor. In the ISP1181 the 1.5 kΩ pull-up resistor is integrated on-chip and is not connected to V_{CC} by default. The connection is established through a command sent by the external/system microcontroller. This allows the system microcontroller to complete its initialization sequence before deciding to establish connection with the USB. Re-initialization of the USB connection can also be performed without disconnecting the cable.

The ISP1181 will check for USB V_{BUS} availability before the connection can be established. V_{BUS} sensing is provided through pin V_{BUS}.

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 V_{SE} 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 ISP1181 has been successfully enumerated (the device address is set), the LED indicator will remain permanently on. Upon each successful packet transfer (with ACK) to and from the ISP1181 the LED will blink off for 100 ms. During ‘suspend’ state the LED will remain off.

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

A register bit can be set to stop the GoodLink LED blinking in traffic (see Table 20). The LED indicator will then be permanently on.

7.6 Bit clock recovery

The bit clock recovery circuit recovers the clock from the incoming USB data stream using a 4× over-sampling principle. It is able to track jitter and frequency drift as specified by the *USB Specification Rev. 1.1*.

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 ISP1181 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 ISP1181 appears as a memory device with an 8/16-bit data bus and an 1-bit address bus. The ISP1181 supports both multiplexed and non-multiplexed address and data buses.

The ISP1181 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 ISP1181 has four bus configuration modes, selected via pins BUS_CONF1 and BUSCONF0:

Mode 0 16-bit I/O port shared with 8-bit or 16-bit DMA port

Mode 1 separate 8-bit I/O port and 8-bit DMA port

Mode 2 8-bit I/O port shared with 8-bit or 16-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 20.1.

Table 3: Bus configuration modes

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

9. Endpoint descriptions

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

The ISP1181 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)	Double buffering	I/O mode access	DMA mode access	Endpoint type
0	64 (fixed)	no	yes	no	control OUT [1]
0	64 (fixed)	no	yes	no	control IN [1]
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] IN: input for the USB host (ISP1181 transmits); OUT: output from the USB host (ISP1181 receives).

[2] The data flow direction is determined by bit EPDIR in the Endpoint Configuration Register.

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

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	interrupt IN 8 bytes, rate feedback mode	256 bytes
1001	interrupt IN 16 bytes, rate feedback mode	320 bytes
1010	interrupt IN 32 bytes, rate feedback mode	384 bytes
1011	interrupt IN 64 bytes, rate feedback mode	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 (512 bytes for non-isochronous FIFOs).

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

Table 6: Memory configuration example

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

9.3 Endpoint initialization

In response to the standard USB request Set Interface, the firmware must program all 16 ECRs of the ISP1181 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 ISP1181 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 ISP1181 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 ISP1181 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 ISP1181 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 ISP1181 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 ISP1181 can be interleaved with normal I/O mode access to other endpoints.

The following features are supported:

- Single-cycle or burst transfers (up tot 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
- Automatic DMA counter reload and transfer restart following 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

Table 7: Endpoint selection for DMA transfer

Endpoint identifier	EPIDX[3:0]	Transfer direction	
		EPDIR = 0	EPDIR = 1
8	1001	OUT: read	IN: write
9	1010	OUT: read	IN: write
10	1011	OUT: read	IN: write
11	1100	OUT: read	IN: write
12	1101	OUT: read	IN: write
13	1110	OUT: read	IN: write
14	1111	OUT: read	IN: write

10.2 8237 compatible mode

The 8237 compatible DMA mode is selected by clearing bit DAKOLY in the Hardware Configuration Register (see Table 22). 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	ISP1181 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 ISP1181 to put data on the bus
$\overline{WR}$	write strobe	I	instructs ISP1181 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 ISP1181 in 8237 compatible DMA mode is given in Figure 3.

The 8237 has two control signals for each DMA channel: DRQ (DMA Request) and $\overline{DACK}$ (DMA Acknowledge). General control signals are HRQ (Hold Request), HLDA (Hold Acknowledge) and $\overline{EOP}$ (End-Of-Process). 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).

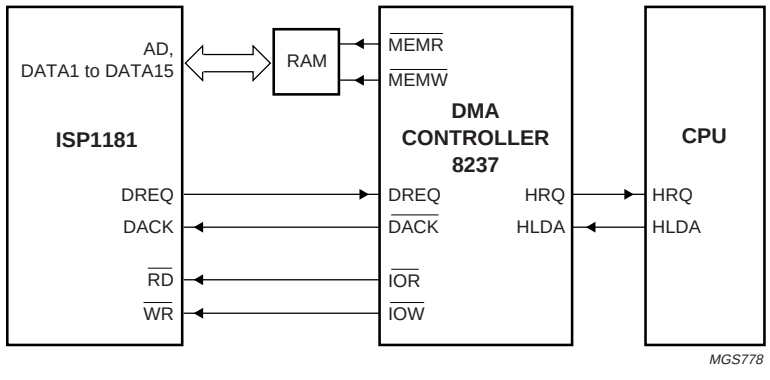


Fig 3. ISP1181 in 8237 compatible DMA mode.

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

1. ISP1181 receives a data packet in one of its endpoint FIFOs; the packet must be transferred to memory address 1234H.
2. ISP1181 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 ISP1181 that it will start a DMA transfer.
7. The ISP1181 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 ISP1181 that the data on the bus lines has been transferred.
9. The ISP1181 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 ISP1181 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 22). The pin functions for this mode are shown in Table 9. A typical example of ISP1181 in DACK-only DMA mode is given in Figure 4.

Table 9: DACK-only mode: pin functions

Symbol	Description	I/O	Function
DREQ	DMA request	O	ISP1181 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 ISP1181 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.

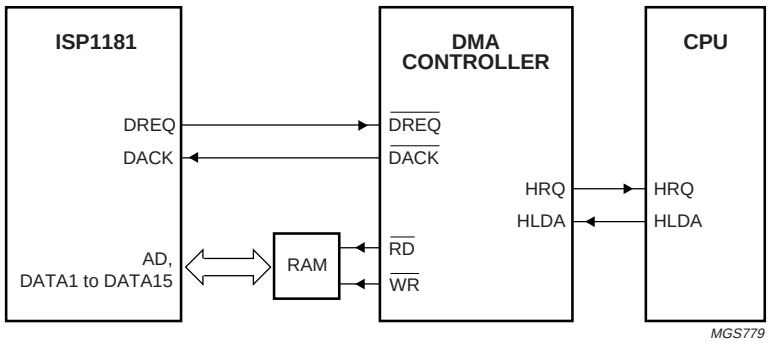


Fig 4. ISP1181 in DACK-only DMA mode.

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 26):

- An external End-Of-Transfer signal occurs on input EOT
- The internal DMA Counter Register reaches zero (CNTREN = 1)
- A short/empty 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 zero: An EOT from the DMA Counter Register is enabled by setting bit CNTREN in the DMA Configuration Register. The ISP1181 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 reaches zero an EOT condition is generated and the DMA operation stops.

Short/empty packet: Normally, the transfer byte count must be set via a control endpoint before any DMA transfer takes place. When a short/empty packet has been enabled as EOT indicator (SHORTP = 1), the transfer size is determined by the presence of a short/empty 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/empty packet at an OUT token will stop the DMA operation after transferring the data bytes of this packet.

When writing to an IN endpoint, a short packet transferred at an IN token will stop the DMA operation after all bytes have been transferred. If the number of bytes in the buffer is zero, ISP1181 will automatically send an empty 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	counter reaches zero	counter reaches zero
Short packet	short packet is received and transferred	counter reaches zero in the middle of the buffer
Empty packet	empty packet is received and transferred	empty packet is automatically appended when needed [1]
DMAEN bit in DMA Configuration Register	DMAEN = 0	DMAEN = 0

[1] If short/empty packet EOT is enabled (SHORTP = 1 in DMA Configuration Register) and DMA Counter Register is zero.

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 26):

- An external End-Of-Transfer signal occurs on input EOT
- The internal DMA Counter Register reaches zero (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

10.4.3 DMA auto-restart

If the AUTOLD bit in the DMA Configuration Register is set, the DMA operation will automatically restart when the last transfer has been completed. First the internal DMA counter is reloaded from of the DMA Counter Register. Output DREQ is then asserted to request a new DMA transfer for an IN endpoint, or when the buffer of an OUT endpoint buffer has been filled.

11. Suspend and resume

11.1 Suspend conditions

The ISP1181 detects a USB ‘suspend’ status in the following cases:

- A J-state is present on the USB bus for 3 ms
- V_{BUS} is lost (weak pull-up/down on D+ and D–)
- SoftConnect is disabled by clearing bit SOFTCT in the Mode Register, with external pull-ups disabled by EXTPUL = 0 in the Hardware Configuration Register. In this situation ISP1181 is effectively disconnected from the USB bus.

ISP1181 will remain in ‘suspend’ state for at least 5 ms, before responding to external wake-up events such as global resume, bus traffic, wake-up on $\overline{CS}$ or WAKEUP. The typical timing is shown in Figure 5.

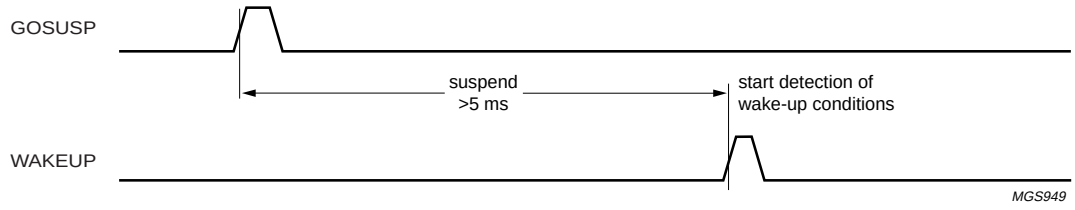


Fig 5. Typical suspend timing.

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

ISP1181 can either be in powered-on or powered-off mode during ‘suspend’ state. This is controlled by bit PWROFF in the Hardware Configuration Register. A full explanation of these modes is given in Section 11.1.1 and Section 11.1.2.

The steps leading up to ‘suspend’ status are as follows:

1. Upon detection of a ‘wake-up’ to ‘suspend’ transition ISP1181 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 ‘suspend’ state:
   a. All signals connected to ISP1181 must enter appropriate states to meet the power consumption requirements of ‘suspend’ state.
   b. All input pins of ISP1181 must have a CMOS logic 0 or logic 1 level. Pin settings differ for powered-on and powered-off application.
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 ‘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 the GOSUSP bit in the Mode Register, the ISP1181 enters ‘suspend’ state. In powered-off application, the ISP1181 asserts output SUSPEND and switches off the internal clocks after 2 ms.

11.1.1 Powered-on application

In powered-on application (PWROFF = 0 in the Hardware Configuration Register) the power supply of the CPU and other parts of the circuit is not switched off. The CPU is normally placed in low-power mode. The SUSPEND output of ISP1181 is normally HIGH and pulses LOW for 10 ms upon a ‘resume’ condition. This signal can be used to wake up the CPU. The signal timing is shown in Figure 6.

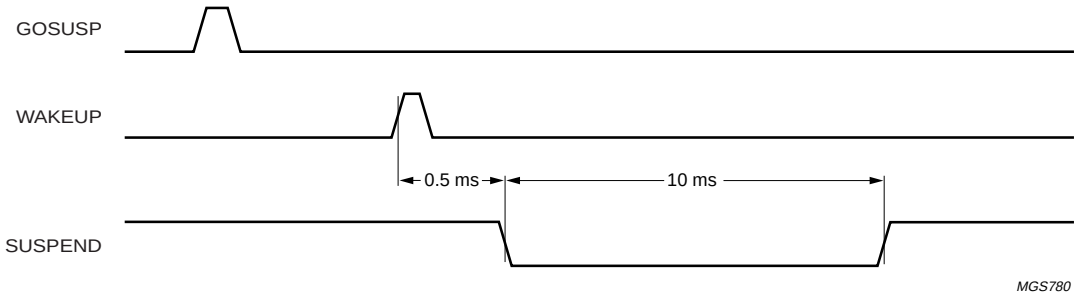


Fig 6. Suspend and resume timing for powered-on application.

In powered-on application ISP1181 drives its output pins, while the inputs are driven by the application. Bi-directional pins are placed in three-state and driven HIGH or LOW by the application. A summary of appropriate pin states is given in Table 12.

Table 12: Pin states in powered-on application

Pin	Type	Appropriate state
A0	I/O (three-state)	externally driven [1] to logic 0 or logic 1
DATA[15:0]	I/O (three-state)	depends on state of inputs $\overline{RD}$ and $\overline{CS}$
SUSPEND	O	ISP1181 drives logic 1
WAKEUP	I	externally driven to logic 1
INT	O (three-state)	ISP1181 drives logic 0 or logic 1
$\overline{RESET}$	I	externally driven to logic 1
$\overline{CS}$	I	externally driven to logic 0 or logic 1 (default: logic 1)
$\overline{RD}$	I	externally driven to logic 0 or logic 1 (default: logic 1)
$\overline{WR}$	I	externally driven to logic 1
XTAL1	I	externally driven to logic 1, if external oscillator is used
CLKOUT	O (three-state)	ISP1181 drives logic 0

[1] ‘Externally driven’ refers to logic outside the ISP1181.

The USB connections D+ and D– remain powered and logically connected to the USB bus. If a crystal oscillator is used, powering down during ‘suspend’ is managed by the internal logic of ISP1181. When using an external oscillator on pin XTAL1, a stable logic 1 level must be applied during ‘suspend’ state.

Figure 7 shows a typical bus-powered modem application using ISP1181 in powered-on mode. The SUSPEND output is connected to the reset input (RST) of the 8031 microcontroller via an external inverter. This allows a ‘resume’ condition to wake up the 8031 from power-down mode. The ISP1181 is woken up via the USB bus (global resume) or by the ring detection circuit on the telephone line.

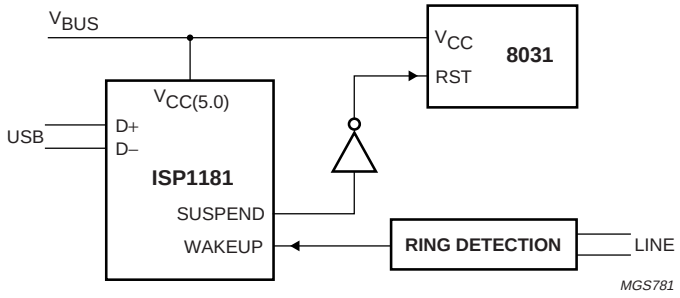


Fig 7. SUSPEND and WAKEUP signals in a powered-on modem application.

11.1.2 Powered-off application

In powered-off application (PWROFF = 1 in the Hardware Configuration Register) the supply of the CPU and other parts of the circuit is removed during ‘suspend’ state. The SUSPEND output is active HIGH during ‘suspend’ state, making it suitable as a power switch control signal, e.g. for an external oscillator.

Input pins of ISP1181 are pulled to ground via the pin buffers. Outputs are made three-state to prevent current flowing in the application. Bi-directional pins are made three-state and must be pulled to ground externally by the application. The power supply of external pull-ups must also be removed to reduce power consumption.

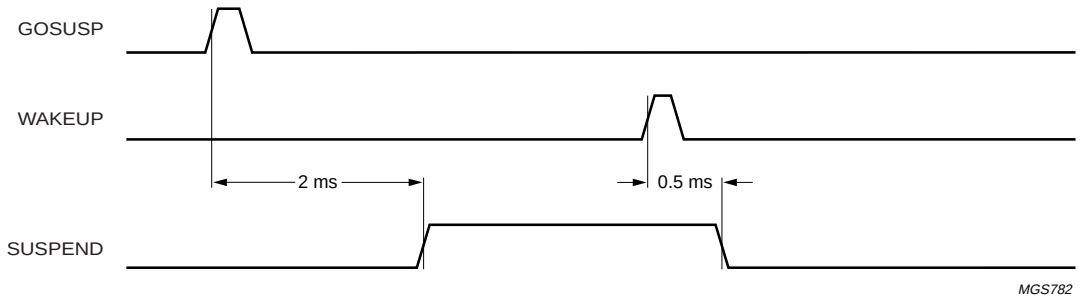


Fig 8. Suspend and resume timing for powered-off application.

Table 13: Pin states in powered-off application

Pin	Type	Appropriate state
A0	I/O (three-state)	powered off; internally connected to ground (logic 0)
DATA[15:0]	I/O (three-state)	powered off; internally connected to ground (logic 0)
SUSPEND	O	ISP1181 drives logic 1
WAKEUP	I	powered off; internally connected to ground (logic 0)
INT	O (three-state)	powered off; internally connected to ground (logic 0)
$\overline{\text{RESET}}$	I	externally driven [1] to logic 1
$\overline{\text{CS}}$	I	powered off; internally connected to ground (logic 0)
$\overline{\text{RD}}$	I	powered off; internally connected to ground (logic 0)
$\overline{\text{WR}}$	I	powered off; internally connected to ground (logic 0)
XTAL1	I	powered off; internally connected to ground (logic 0)
CLKOUT	O (three-state)	ISP1181 drives logic 0

[1] ‘Externally driven’ refers to logic outside the ISP1181.

When external components are powered-off, it is possible that interface signals $\overline{\text{RD}}$, $\overline{\text{WR}}$ and $\overline{\text{CS}}$ have unknown values immediately after leaving ‘suspend’ state. To prevent corruption of its internal registers, ISP1181 enables a locking mechanism once suspend is enabled.

After wake-up from suspend’ state, all internal registers except the Unlock Register are write-protected. A special unlock operation is needed to re-enable write access. This prevents data corruption during power-up of external components.

Figure 9 shows a typical bus-powered modem application using ISP1181 in powered-off mode. The SUSPEND output is used to switch off power to the microcontroller and other external circuits during ‘suspend’ state. The ISP1181 is woken up via the USB bus (global resume) or by the ring detection circuit on the telephone line.

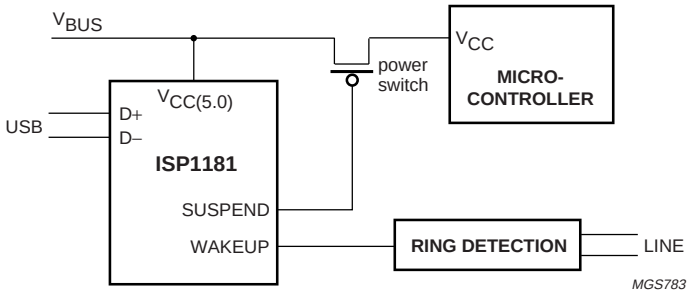


Fig 9. SUSPEND and WAKEUP signals in a powered-off modem application.

11.2 Resume conditions

For both application modes (powered-on and powered-off) wake-up from ‘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 via a HIGH level on input WAKEUP or a LOW level on input $\overline{CS}$ (if enabled via bit WKUPCS in the Hardware Configuration Register).

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 ISP1181.
2. The SUSPEND output is de-asserted and the RESUME bit 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 ISP1181 resumes its normal functionality.
4. In case of a remote wake-up ISP1181 drives a K-state on the USB bus for 10 ms.
5. Following the de-assertion of output SUSPEND, the application restores itself and other system components to normal operating mode.
6. After wake-up the internal registers of ISP1181 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 ISP1181 to restore its full functionality. See Section 12.3.2 for more details.

11.3 Control bits in suspend and resume

Table 14: Summary of control bits

Register	Bit	Function
Interrupt	SUSPND	a transition from ‘awake’ to ‘suspend’ state was detected
	BUSTATUS	monitors USB bus status (logic 1 = suspend); used when interrupt is serviced
Interrupt Enable	IESUSP	enables output INT to signal ‘suspend’ state
Mode	SOFTCT	enables SoftConnect pull-up resistor to USB bus
	GOSUSP	a HIGH-to-LOW transition enables ‘suspend’ state
	SNDRSU	a HIGH-to-LOW transition enables sending a 10 ms resume signal (K-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 ‘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 ISP1181 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 15.

A complete access consists of two phases:

1. **Command phase**: when address bit A0 = 1, the ISP1181 interprets the data on the lower byte of the bus (bits D7 to D0) as a command code. Commands without a data phase are executed immediately.
2. **Data phase (optional)**: when address bit A0 = 0, the ISP1181 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 15: 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/word [6]
Write Control IN Configuration	Endpoint Configuration Register endpoint 0 IN	21	write 1 byte/word [6]
Write Endpoint n Configuration (n = 1 to 14)	Endpoint Configuration Register endpoint 1 to 14	22 to 2F	write 1 byte/word [6] [3]
Read Control OUT Configuration	Endpoint Configuration Register endpoint 0 OUT	30	read 1 byte/word [6]
Read Control IN Configuration	Endpoint Configuration Register endpoint 0 IN	31	read 1 byte/word [6]
Read Endpoint n Configuration (n = 1 to 14)	Endpoint Configuration Register endpoint 1 to 14	32 to 3F	read 1 byte/word [6]
Write/Read Device Address	Address Register	B6/B7	write/read 1 byte/word [6]
Write/Read Mode Register	Mode Register	B8/B9	write/read 1 byte/word [6]
Write/Read Hardware Configuration	Hardware Configuration Register	BA/BB	write/read 1 byte/word [6]
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 1 byte/word [6]
Write/Read DMA Counter	DMA Counter Register	F2/F3	write/read 2 bytes
Reset Device	resets all registers	F6	none

Table 15: Command and register summary

Name	Destination	Code (Hex)	Transaction [1]
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 [7] interrupt/bulk: N ≤ 64 bytes
Write Control OUT Status	Endpoint Status Register endpoint 0 OUT	40	write 1 byte/word [6]
Write Control IN Status	Endpoint Status Register endpoint 0 IN	41	write 1 byte/word [6]
Write Endpoint n Status (n = 1 to 14)	Endpoint Status Register n endpoint 1 to 14	42 to 4F	write 1 byte/word [6]
Read Control OUT Status	Endpoint Status Register endpoint 0 OUT	50	read 1 byte/word [6]
Read Control IN Status	Endpoint Status Register endpoint 0 IN	51	read 1 byte/word [6]
Read Endpoint n Status (n = 1 to 14)	Endpoint Status Register n endpoint 1 to 14	52 to 5F	read 1 byte/word [6]
Validate Control OUT Buffer	illegal: IN endpoints only [2]	(60)	-
Validate Control IN Buffer	FIFO endpoint 0 IN [2]	61	none [3]
Validate Endpoint n Buffer (n = 1 to 14)	FIFO endpoint 1 to 14 (IN endpoints only) [2]	62 to 6F	none [3]
Clear Control OUT Buffer	FIFO endpoint 0 OUT	70	none [3]
Clear Control IN Buffer	illegal [4]	(71)	-
Clear Endpoint n Buffer (n = 1 to 14)	FIFO endpoint 1 to 14 (OUT endpoints only) [4]	72 to 7F	none [3]
Check Control OUT Status [5]	Endpoint Status Image Register endpoint 0 OUT	D0	read 1 byte/word [6]
Check Control IN Status [5]	Endpoint Status Image Register endpoint 0 IN	D1	read 1 byte/word [6]
Check Endpoint n Status (n = 1 to 14) [5]	Endpoint Status Image Register n endpoint 1 to 14	D2 to DF	read 1 byte/word [6]
Acknowledge Setup	Endpoint 0 IN and OUT	F4	none [3]

Table 15: Command and register summary

Name	Destination	Code (Hex)	Transaction [1]
General commands			
Read Control OUT Error Code	Error Code Register endpoint 0 OUT	A0	read 1 byte/word [6]
Read Control IN Error Code	Error Code Register endpoint 0 IN	A1	read 1 byte/word [6]
Read Endpoint n Error Code (n = 1 to 14)	Error Code Register endpoint 1 to 14	A2 to AF	read 1 byte/word [6]
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 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] Validating an OUT endpoint buffer causes unpredictable behaviour of ISP1181.

[3] In 8-bit bus mode this command requires more time to complete than other commands. See Table 60.

[4] Clearing an IN endpoint buffer causes unpredictable behaviour of ISP1181.

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

[6] In 8-bit mode, the upper byte is invalid.

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

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 ISP1181 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 16. 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 16: 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 17: 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 18.

A USB bus reset sets the device address to 00H and enables the device. 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 18: 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 19: 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 ISP1181 Mode Register, which consists of 1 byte (bit allocation: see Table 19). In 16-bit bus mode the upper byte is ignored.

The Mode Register controls the DMA bus width, resume and suspend modes, interrupt activity, GoodLink signalling 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 20: Mode Register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	DMAWD	SNDRSU	GOSUSP	reserved	INTENA	DBGMOD	DISGLBL	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 21: 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	SNDRSU	Writing a logic 1 followed by a logic 0 will generate an upstream ‘resume’ signal of 10 ms duration, after a 5 ms delay.
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	DISGLBL	A logic 1 disables GoodLInk LED blinking on USB traffic. The LED will be continuously on ($\overline{GL}$ = LOW) after successful enumeration. Bus reset value: unchanged.
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 22). 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 22. 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 22: Hardware Configuration Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved	EXTPUL	NOLAZY	CLKRUN	CKDIV[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	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23: 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 (24 kHz) 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	CKDIV[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 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.
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 $\overline{CS}$. Bus reset value: unchanged.

Table 23: Hardware Configuration Register: bit description

Bit	Symbol	Description
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'). 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 83 ms 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 24.

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

Transaction — write/read 4 bytes

Table 24: 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	reserved	IENOSOF	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 25: 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	IENOSOF	A logic 1 enables 1 ms interrupts upon loss of 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 ISP1181 and enables/disables DMA transfers. The command accesses the DMA Configuration Register, which consists of 2 bytes. The bit allocation is given in Table 26. A bus reset will clear bits DMAEN and AUTOLD (DMA and auto-restart disabled), all other bits remain unchanged.

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

Transaction — write/read 2 bytes

Table 26: DMA Configuration Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	CNTREN	SHORTP	reserved	reserved	reserved	reserved	reserved	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	AUTOLD	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 27: 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) an empty packet is appended when needed. Bus reset value: unchanged.
13 to 8	-	reserved
7 to 4	EPDIX[3:0]	Indicates the destination endpoint for DMA, see Table 7.

Table 27: DMA Configuration Register: bit description

Bit	Symbol	Description
3	DMAEN	Writing a logic 1 enables DMA transfer, a logic 0 forces the end of an ongoing DMA transfer and generates an EOT interrupt. Reading this bit indicates whether DMA is enabled (0 = DMA stopped, 1 = DMA enabled). This bit is cleared by a bus reset.
2	AUTOLD	A logic 1 enables automatic restarting of DMA transfers. This bit is cleared by a bus reset.
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 28. 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) or upon completion of a DMA transfer, when auto-restart is enabled (AUTOLD = 1). See Section 12.1.6 for more details.

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

Transaction — write/read 2 bytes

Table 28: 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 29: 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 ISP1181 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 ISP1181 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 behaviour of ISP1181.

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: $N \leq 1023$, bulk/interrupt endpoint: $N \leq 32$)

The data in the endpoint FIFO must be organized as shown in Table 30. Examples of endpoint FIFO access are given in Table 31 (8-bit bus) and Table 32 (16-bit bus).

Table 30: 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 31: 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 32: 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 Write/Read Endpoint Status

This command is used to read the status of the endpoint FIFO or stall the endpoint by writing. The command accesses the Endpoint Status Register, the bit allocation of which is shown in Table 33.

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

When a stalled endpoint is unstalled (either by the Set Endpoint Status command or by receiving a SETUP token), it is also re-initialized. This flushes the buffer: in and 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 — write (control OUT, control IN, endpoint 1 to 14)

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

Transaction — write/read 1 byte

Table 33: Endpoint Status Register: bit allocation

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

Table 34: Endpoint Status Register: bit description

Bit	Symbol	Description
7	EPSTAL	Writing a logic 1 will stall the endpoint. The endpoint is automatically unstalled upon reception of a SETUP token. Reading 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	-	reserved
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 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

12.2.4 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

12.2.5 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 35.

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

Transaction — write/read 1 byte

Table 35: Endpoint Status Image Register: bit allocation

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

Table 36: 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	-	reserved
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

12.2.6 Acknowledge Setup

This command is acknowledges to the host that a Setup packet was received. It re-enables the Validate Buffer and Clear Buffer commands for the control IN and control OUT endpoints. These commands are disabled automatically when a Setup packet is received, see Section 9.5.

Remark: The Acknowledge Setup command must be sent to **both** control endpoints (IN and OUT).

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 37.

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

Transaction — read 1 byte

Table 37: 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 38: 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	Indicates the PID type of the last successfully received packet (0 = DATA0 PID, 1 = DATA1 PID).
5	-	reserved
4 to 1	ERROR[3:0]	Error code. For error description, see Table 39.
0	RTOK	A logic 1 indicates that data was received or transmitted successfully.

Table 39: 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

Table 39: Transaction error codes

Error code (Binary)	Description
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 ISP1181 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’. Register access for reading is not blocked.

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 40.

Code (Hex): B0 — unlock the device

Transaction — write 2 bytes (unlock code)

Table 40: 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 41: Error Code 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, e.g. the device status before powering down in 'suspend' state. The register bit allocation is given in Table 42.

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

Transaction — write/read 2 bytes

Table 42: Scratch Information Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	SFIRH[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	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 43: Scratch Information Register: bit description

Bit	Symbol	Description
15 to 8	SFIRH[7:0]	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 44.

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 44: Frame Number Register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved	reserved	reserved	reserved	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 45: 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 46: 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 47.

Code (Hex): B5 — read chip ID

Transaction — read 2 bytes

Table 47: Chip ID Register: bit allocation

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

Table 48: Scratch Information Register: bit description

Bit	Symbol	Description
15 to 8	CHIPIDH[7:0]	Chip ID Register (high byte)
7 to 0	CHIPIDL[7:0]	Chip ID Register (low byte)

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 49. 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.

Code (Hex): C0 — read interrupt register

Transaction — read 4 bytes

Table 49: Interrupt Register: bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved	reserved	reserved	reserved	reserved	reserved	reserved	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	NOSOF	SOF	EOT	SUSPND	RESUME	RESET
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 50: 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
8	EP0OUT	A logic 1 indicates the interrupt source: control OUT endpoint
7	BUSTATUS	Monitors the current USB bus status (0 = awake, 1 = suspend).
6	-	reserved
5	NOSOF	A logic 1 indicates that an SOF was lost; interrupt is issued every 1 ms; 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 10 shows the interrupt logic of the ISP1181. 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 21).

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 23). Default settings after reset are active LOW and level mode. When pulse mode is selected, a pulse of 83 ns is generated when the OR-ed combination of all interrupt bits changes from logic 0 to logic 1.

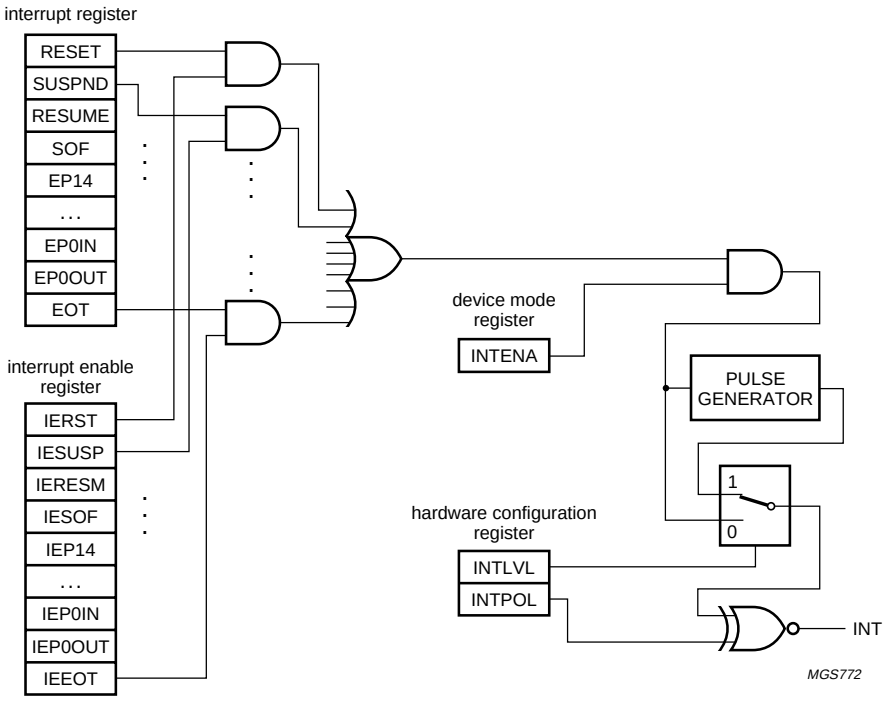


Fig 10. Interrupt logic.

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 ISP1181 has acknowledged the associated data packet. In bulk transfer mode, the ISP1181 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 loss of Start-Of-Frame (NOSOF) interrupt, enabled via bit IENOSOF in the Interrupt Enable Register. If a Start-Of-Frame is lost, NOSOF interrupts are generated every 1 ms. This allows the firmware to keep data transfer synchronized with the host. After 3 missed SOF events the ISP1181 will enter ‘suspend’ state.

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

14. Power supply

The ISP1181 is powered from a single supply voltage, ranging from 4.0 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 11.

The ISP1181 can also be operated from a 3.0 to 3.6 V supply, as shown in Figure 12. In that case the internal voltage regulator is disabled and pin $V_{reg(3.3)}$ must be connected to V_{CC}.

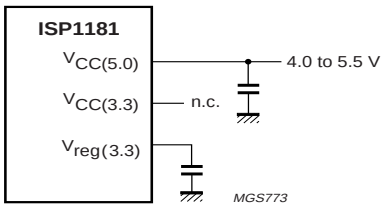


Fig 11. ISP1181 with a 4.0 to 5.5 V supply.

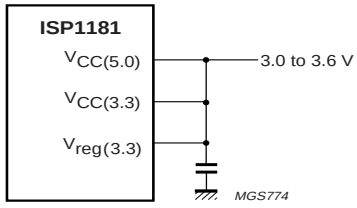


Fig 12. ISP1181 with a 3.0 to 3.6 V supply.

15. Crystal oscillator and LazyClock

The ISP1181 has a crystal oscillator designed for a 6 MHz parallel-resonant crystal (fundamental). A typical circuit is shown in Figure 13. Alternatively, an external clock signal of 6 MHz can be applied to input XTAL1, while leaving output XTAL2 open.

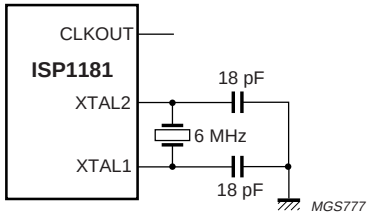


Fig 13. Typical oscillator circuit.

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 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 24 kHz.

The oscillator operation and the CLKOUT frequency are controlled via the Hardware Configuration Register, as shown in Figure 14. 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.

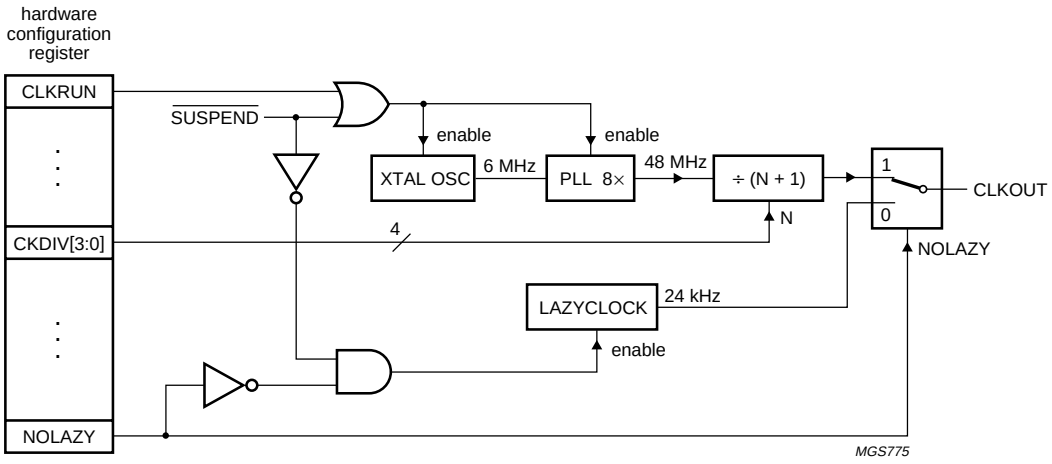
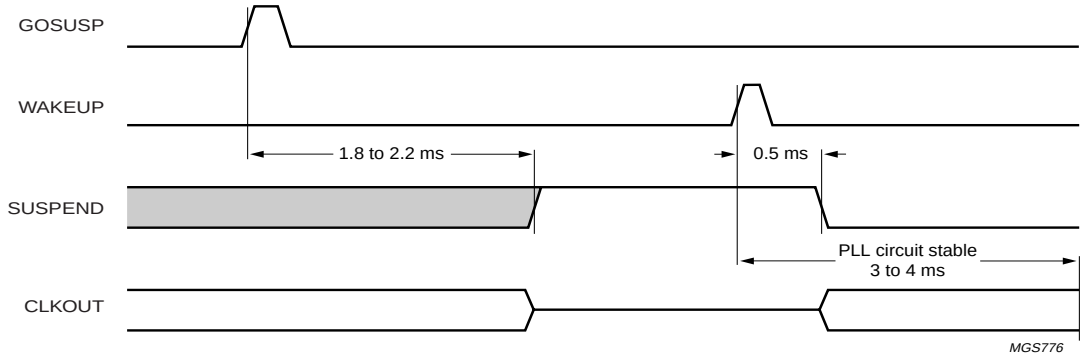


Fig 14. Oscillator and LazyClock logic.

When ISP1181 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 24 kHz LazyClock frequency.

When resuming from ‘suspend’ state by a positive pulse on input WAKEUP, output SUSPEND is cleared and the clock signal on CLKOUT restarted after a 0.5 ms delay. The timing of the CLKOUT signal at ‘suspend’ and ‘resume’ is given in Figure 15.



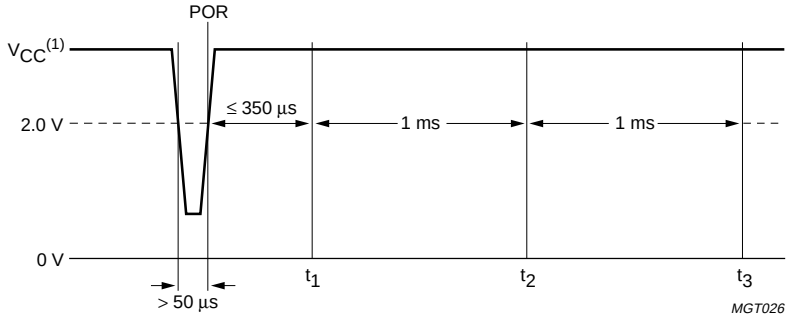
If enabled, the 24 kHz LazyClock frequency will be output on pin CLKOUT during 'suspend' state.

Fig 15. CLKOUT signal timing at 'suspend' and 'resume'.

16. Power-on reset

The ISP1181 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 16. 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 51: 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
$I_{latchup}$	latchup current	$V_I < 0$ or $V_I > V_{CC}$	-	200	mA
V_{esd}	electrostatic discharge voltage	$I_{LI} < 15$ µA [1] [2]	-	±4000 [3]	V
T_{stg}	storage temperature		–60	+150	°C
P_{tot}	total power dissipation		-	<tbf>	mW

[1] Equivalent to discharging a 100 pF capacitor via a 1.5 kΩ resistor.

[2] Values are given for device only; in-circuit $V_{esd(max)}$ = ±8000 V.

[3] For open-drain pins $V_{esd(max)}$ = ±2000 V.

Table 52: Recommended operating conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	supply voltage		4.0	5.5	V
V_I	input voltage		0	5.5	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}	operating ambient temperature		–40	+85	°C

18. Static characteristics

Table 53: Static characteristics; supply pins

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{reg(3.3)}$	regulated supply voltage		3.0 [1]	3.3	3.6	V
I_{CC}	operating supply current		-	<tbf>	-	mA
$I_{CC(susp)}$	suspend supply current	1.5 kΩ pull-up on upstream port D+ (pin DP0)	-	-	<tbf>	µA
		no pull-up on upstream port D+ (pin DP0)	-	-	<tbf>	µA

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

Table 54: Static characteristics: digital pins

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Input levels						
V_{IL}	LOW-level input voltage		-	-	0.8	V
V_{IH}	HIGH-level input voltage		2.0	-	-	V

Table 54: Static characteristics: digital pins

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
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 (open drain outputs)	I_{OL} = rated drive	-	-	0.4	V
		I_{OL} = 20 µA	-	-	0.1	V
Leakage current						
I_{LI}	input leakage current		-	-	±5	µA
Open-drain outputs						
I_{OZ}	OFF-state output current		-	-	±5	µA

Table 55: Static characteristics: analog I/O pins (D+, D–) [1]

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Input levels						
V_{DI}	differential input sensitivity	$\|V_{I(D+)} - V_{I(D-)}\|$	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 +3.6V	-	-	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	1.1	-	1.9	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.

19. Dynamic characteristics

Table 56: Dynamic characteristics

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

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

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

Table 57: Dynamic characteristics: analog I/O pins (D+, D−)

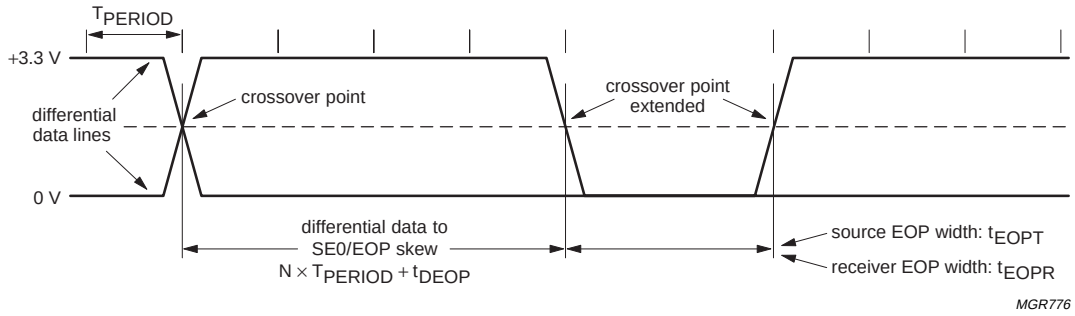
V_{CC} = 4.0 to 5.5 V; V_{GND} = 0 V; T_{amb} = −40 to +85 °C; C_L = 50 pF; R_{PU} = 1.5 kΩ on D+ to V_{TERM}.; unless otherwise specified.

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 17 [3]	160	-	175	ns
t_{FDEOP}	source differential data-to-EOP transition skew	see Figure 17 [3]	−2	-	+5	ns
Receiver timing						
t_{JR1}	receiver data jitter tolerance for consecutive transitions	see Figure 18 [3]	−18.5	-	+18.5	ns
t_{JR2}	receiver data jitter tolerance for paired transitions	see Figure 18 [3]	−9	-	+9	ns
t_{FEOPR}	receiver SE0 width	accepted as EOP; see Figure 17 [3]	82	-	-	ns
t_{FST}	width of SE0 during differential transition	rejected as EOP; see Figure 19 [3]	-	-	14	ns

[1] Test circuit: see Figure 38.

[2] Excluding the first transition from Idle state.

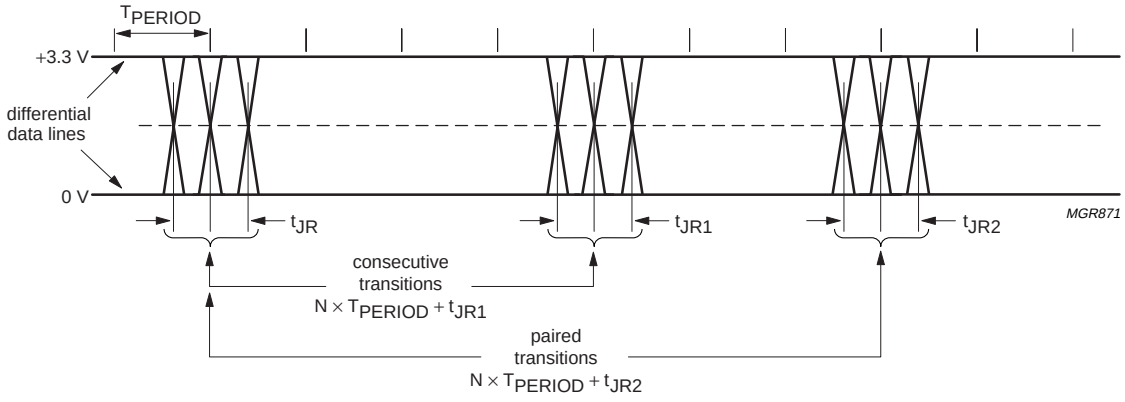
[3] Characterized only, not tested. Limits guaranteed by design.



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 17. Source differential data-to-EOP transition skew and EOP width.



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

Fig 18. Receiver differential data jitter.

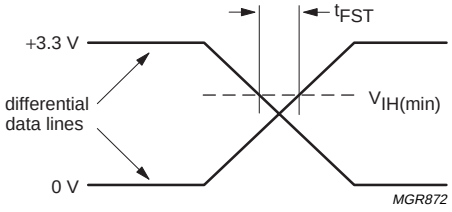


Fig 19. Receiver SE0 width tolerance.

19.1 Timing symbols

Table 58: Legend for timing characteristics

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

19.2 Parallel I/O timing

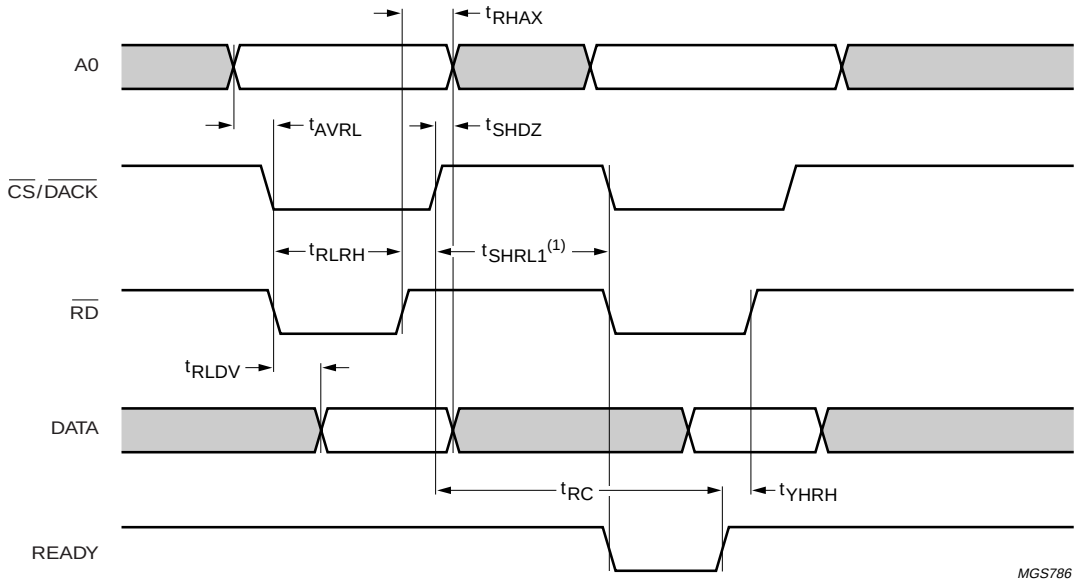
Table 59: Dynamic characteristics: parallel interface timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Read timing (see Figure 20 and Figure 21)							
t_{RHAX}	address hold after $\overline{RD}$ HIGH		3	-	3	-	ns
t_{AVRL}	address setup before $\overline{RD}$ LOW		0	-	0	-	ns
t_{SHDZ}	data outputs high-impedance after $\overline{CS}$ HIGH		-	3	-	3	ns
t_{RLRH}	$\overline{RD}$ pulse width		25	-	25	-	ns
t_{RLDV}	data valid after $\overline{RD}$ LOW		-	22	-	22	ns
t_{SHRL1}	read interval after $\overline{CS}$ HIGH [1]	READY pulsing	22	-	22	-	ns
t_{SHRL2}	read interval after $\overline{CS}$ HIGH [1]	READY = HIGH	90	-	180	-	ns
t_{YHRH}	output READY HIGH before $\overline{RD}$ HIGH		0	-	0	-	ns
t_{RC}	read cycle duration		-	90	-	180	ns
Write timing (see Figure 22 and Figure 23)							
t_{WHAX}	address hold after $\overline{WR}$ HIGH		3	-	3	-	ns
t_{AVWL}	address setup before $\overline{WR}$ LOW		0	-	0	-	ns
t_{SHWL1}	write interval after $\overline{CS}$ HIGH [2]	READY pulsing	22	-	22	-	ns
t_{SHWL2}	write interval after $\overline{CS}$ HIGH [2]	READY = HIGH	90/180 [3]	-	180	-	ns
t_{WLWH}	$\overline{WR}$ pulse width		22	-	22	-	ns
t_{WHSH}	chip deselect after $\overline{WR}$ HIGH		0	-	0	-	ns
t_{DVWH}	data setup before $\overline{WR}$ HIGH		5	-	5	-	ns
t_{WHDZ}	data hold after $\overline{WR}$ HIGH		3	-	3	-	ns
t_{WC}	write cycle duration		-	90/180 [3]	-	180	ns
ALE timing (see Figure 24)							
t_{LH}	ALE pulse width		20	-	20	-	ns
t_{AVLL}	address setup before ALE LOW		10	-	10	-	ns
t_{LLAX}	address hold after ALE LOW	reading	0	10	0	10	ns
		writing	0	-	0	-	ns

[1] Measured from $\overline{CS}$ going HIGH to $\overline{CS}$ and $\overline{RD}$ both going LOW.

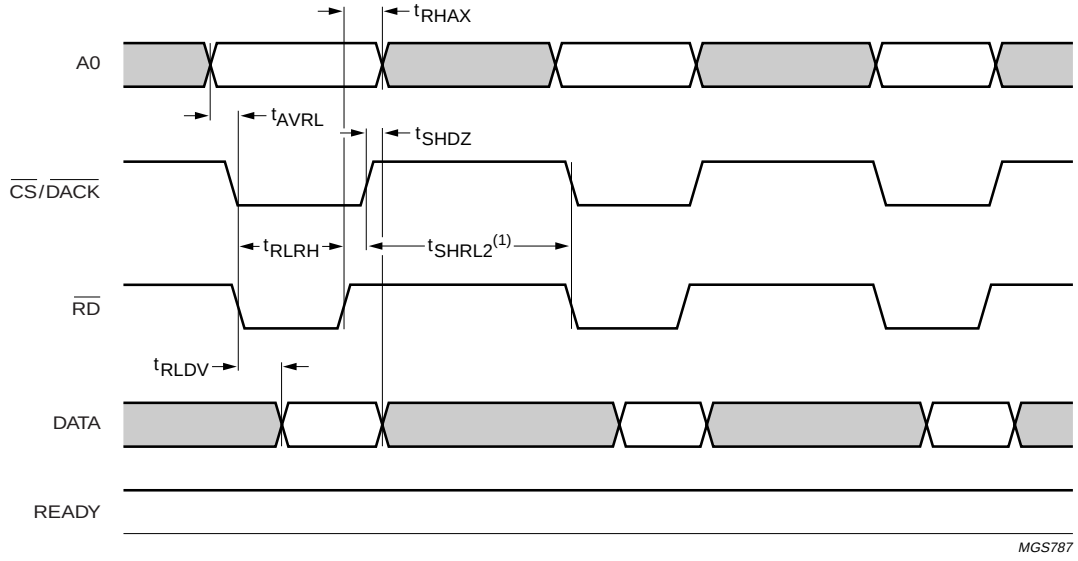
[2] Measured from $\overline{CS}$ going HIGH to $\overline{CS}$ and $\overline{WR}$ both going LOW.

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



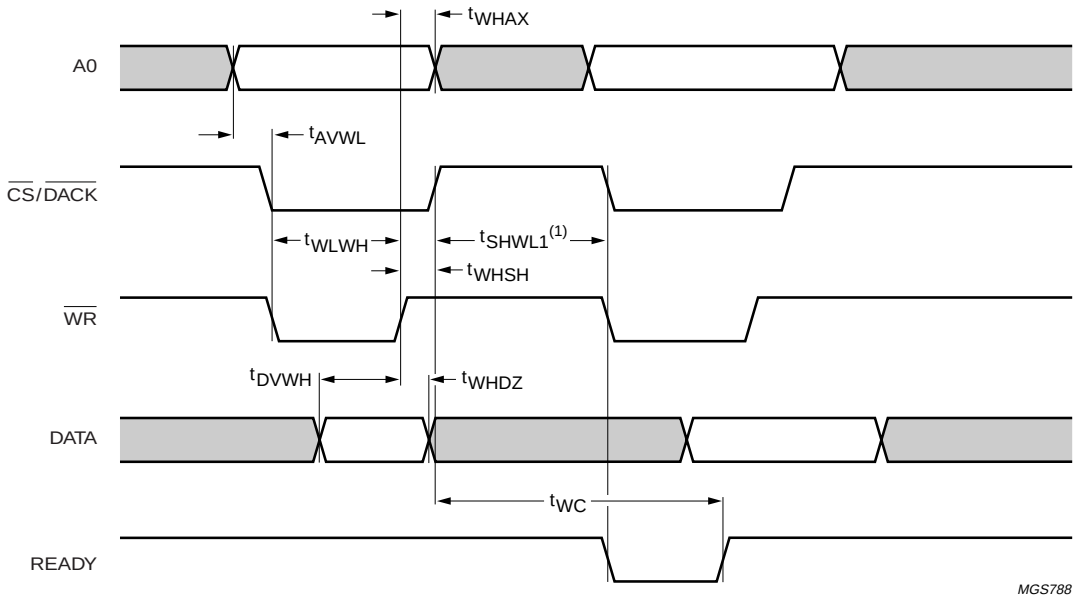
(1) For t_{SHRL} both $\overline{CS}$ and $\overline{RD}$ must be LOW.

Fig 20. Parallel interface read timing (I/O and 8237 compatible DMA) with READY.



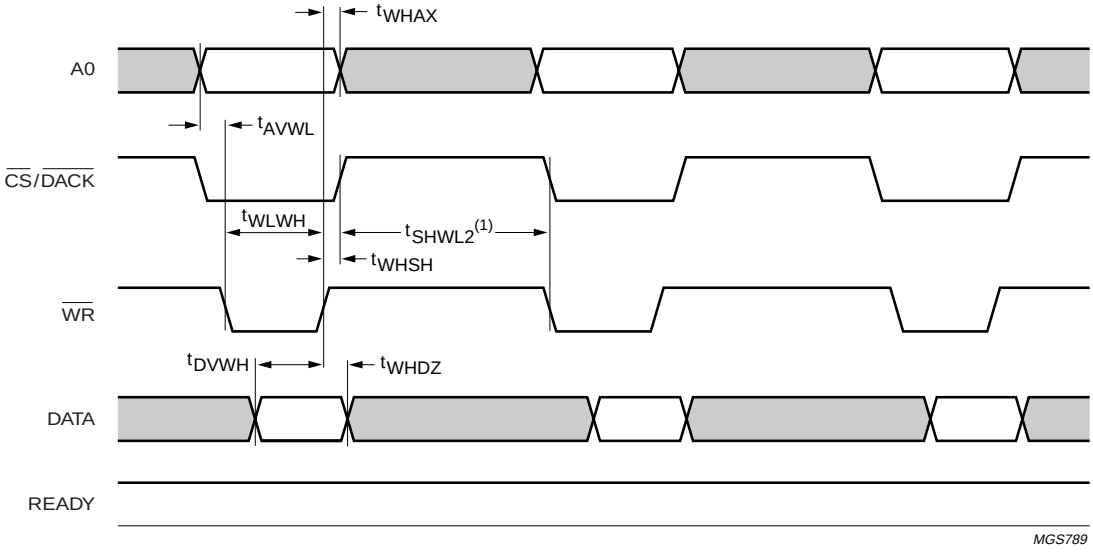
(1) For t_{SHRL} both $\overline{CS}$ and $\overline{RD}$ must be LOW.

Fig 21. Parallel interface read timing (I/O and 8237 compatible DMA) without READY.



(1) For t_{SHRL} both $\overline{CS}$ and $\overline{WR}$ must be LOW.

Fig 22. Parallel interface write timing (I/O and 8237 compatible DMA) with READY.



(1) For t_{SHRL} both $\overline{CS}$ and $\overline{WR}$ must be LOW.

Fig 23. Parallel interface write timing (I/O and 8237 compatible DMA) without READY.

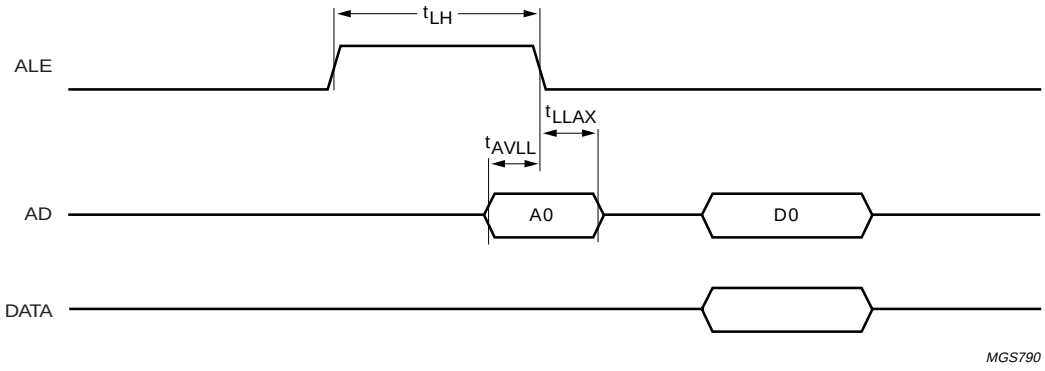


Fig 24. ALE timing.

19.3 Access cycle timing

Table 60: Dynamic characteristics: access cycle timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min [1]	Max	Min [1]	Max	
Write command + write data (see Figure 25 and Figure 26)							
$T_{cy(WC-WD)}$	cycle time for write command, then write data		100 [2]	-	200	-	ns
$T_{cy(WD-WD)}$	cycle time for write data		90	-	180	-	ns
$T_{cy(WD-WC)}$	cycle time for write data, then write command		90	-	180	-	ns
Write command + read data (see Figure 27 and Figure 28)							
$T_{cy(WC-RD)}$	cycle time for write command, then read data		100 [2]	-	200	-	ns
$T_{cy(RD-RD)}$	cycle time for read data		90	-	180	-	ns
$T_{cy(RD-WC)}$	cycle time for read data, then write command		90	-	180	-	ns

[1] If the access cycle time is less than specified, the READY signal will be LOW until the internal processing has finished.

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

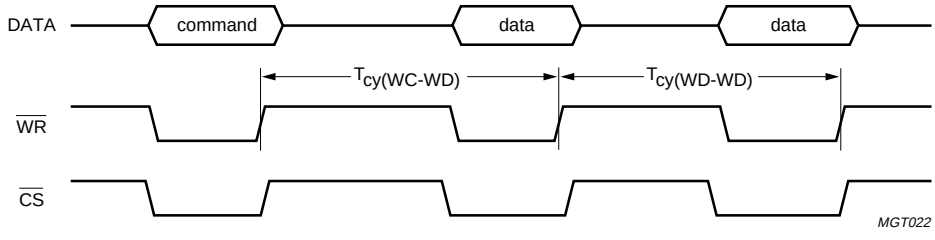
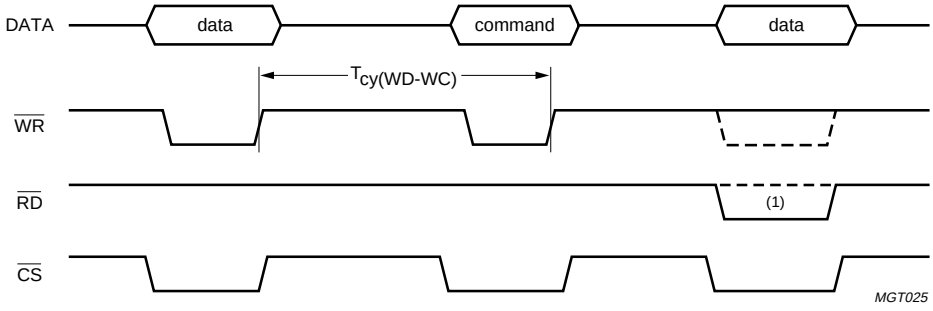


Fig 25. Write command + write data cycle timing.



(1) Example: read data.

Fig 26. Write data + write command cycle timing.

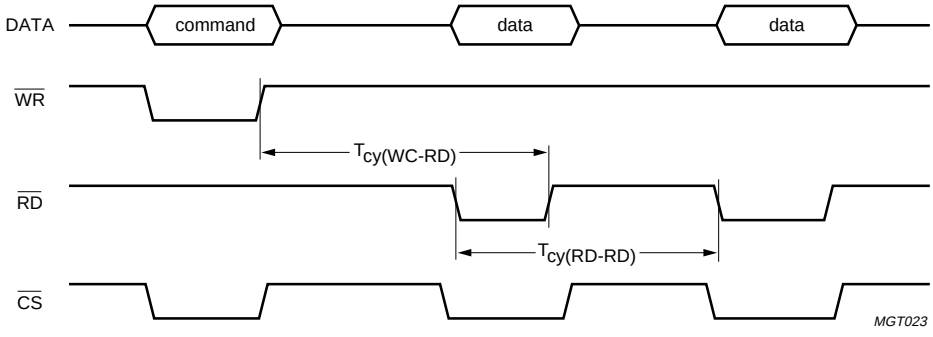
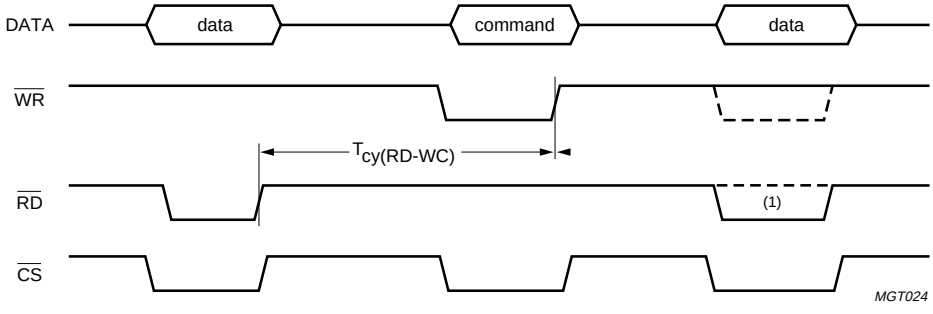


Fig 27. Write command + read data cycle timing.



(1) Example: read data.

Fig 28. Read data + write command cycle timing.

19.4 DMA timing: single-cycle mode

Table 61: 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 29)							
t_{ASRP}	DREQ off after DACK on		-	40	-	40	ns
t_{APRS}	DREQ on after DACK off		-	22	-	22	ns

Table 61: Dynamic characteristics: single-cycle DMA timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Read in DACK-only mode (see Figure 30)							
t_{ASRP}	DREQ off after DACK on		-	22	-	22	ns
t_{APRS}	DREQ on after DACK off		-	22	-	22	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 31)							
t_{ASRP}	DREQ off after DACK on		-	22	-	22	ns
t_{APRS}	DREQ on after DACK off		-	22	-	22	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 32)							
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

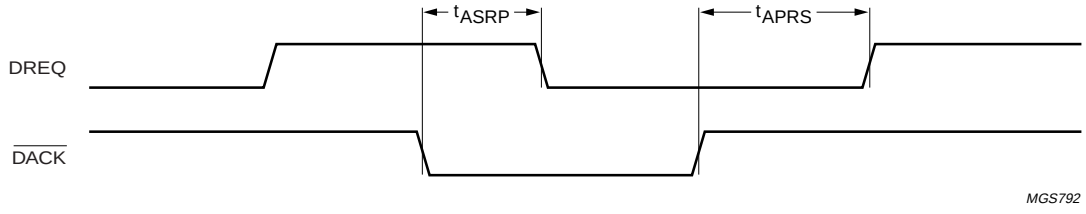


Fig 29. DMA timing in 8237 compatible mode.

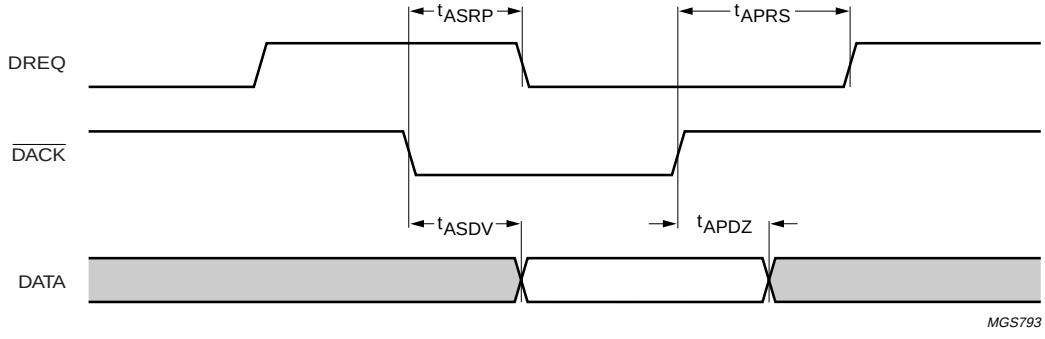


Fig 30. DMA read timing in DACK-only mode.

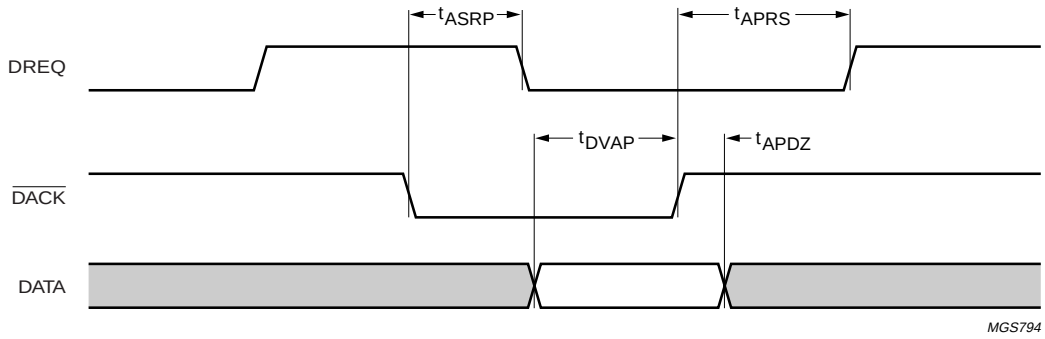
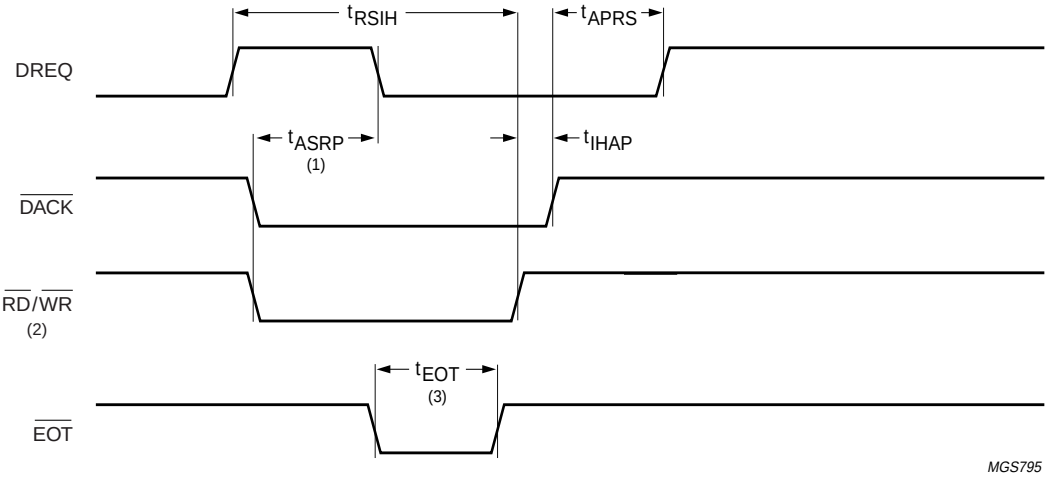


Fig 31. DMA write timing in DACK-only mode.



(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 8237 compatible DMA mode.

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

Fig 32. EOT timing in single-cycle DMA mode.

19.5 DMA timing: burst mode

Table 62: Dynamic characteristics: burst mode DMA timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Burst (see Figure 33)							
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

Table 62: Dynamic characteristics: burst mode DMA timing

Symbol	Parameter	Conditions	8-bit bus		16-bit bus		Unit
			Min	Max	Min	Max	
Burst EOT (see Figure 34)							
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

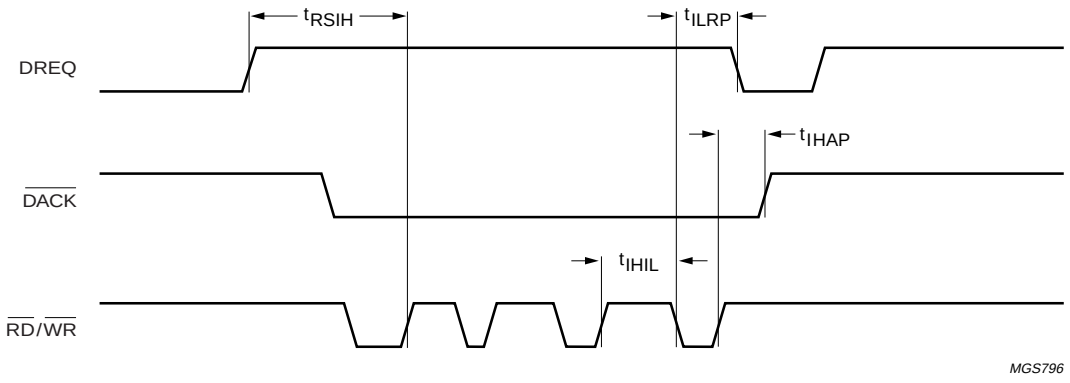
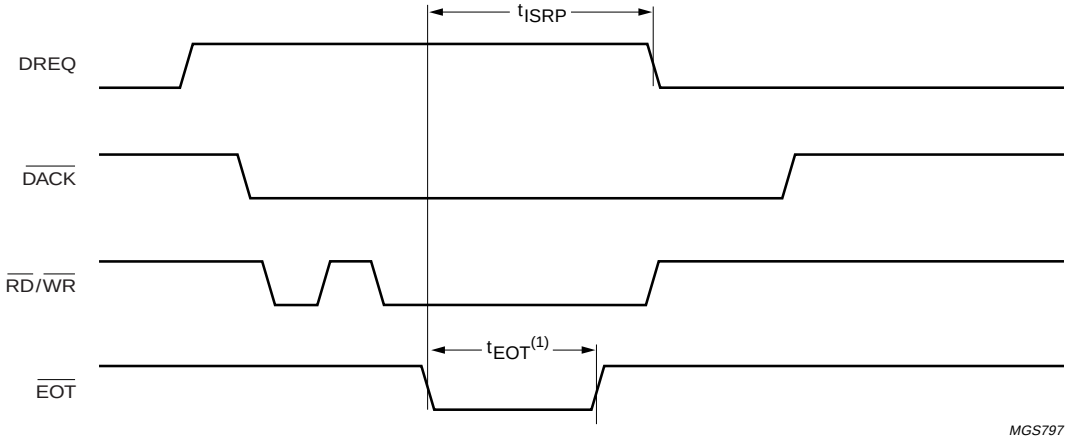


Fig 33. Burst mode DMA timing.



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

Fig 34. EOT timing in burst mode DMA.

20. Application information

20.1 Typical interface circuits

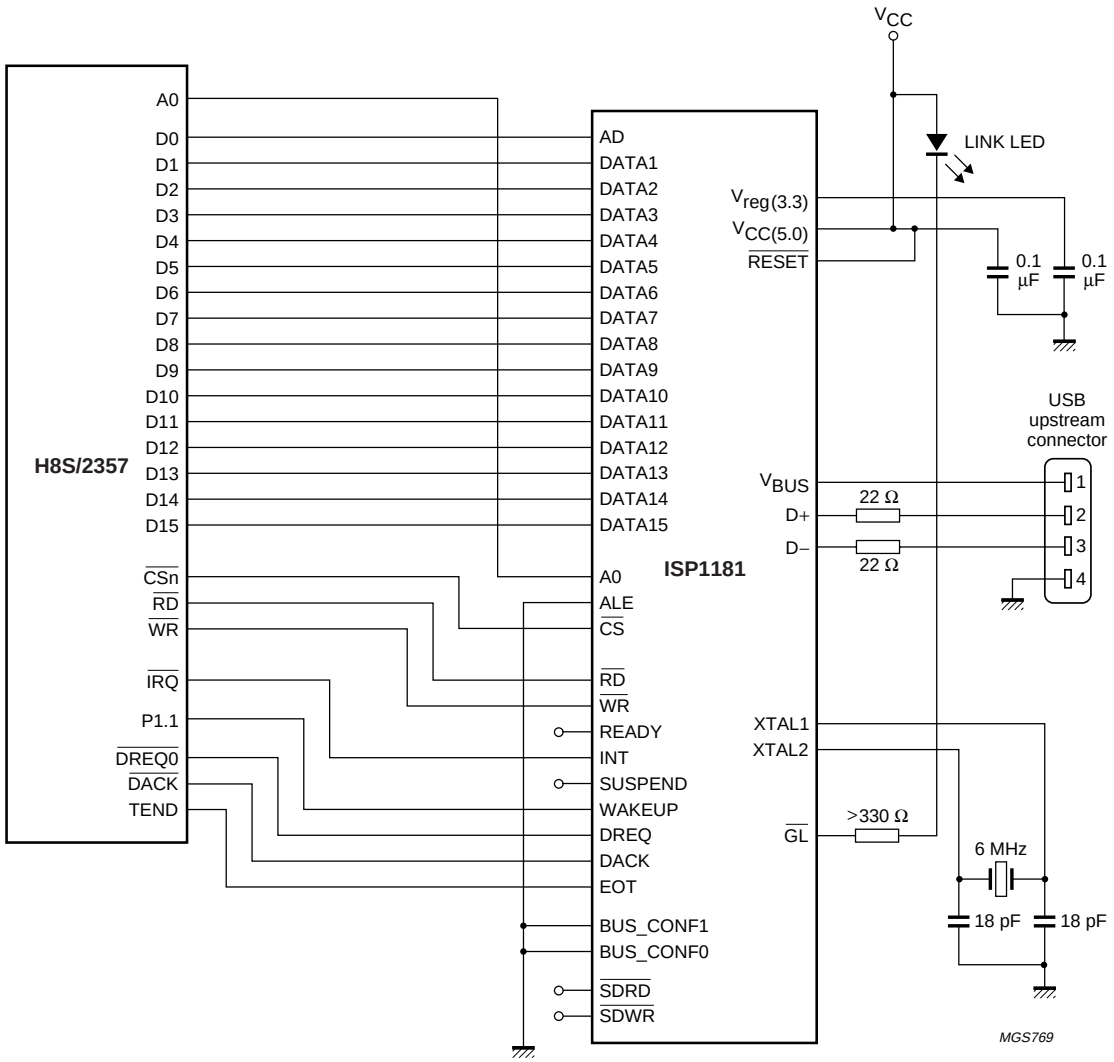


Fig 35. Typical interface circuit for bus configuration mode 0 (shared ports: 16-bit PIO, 8-bit or 16-bit DMA).

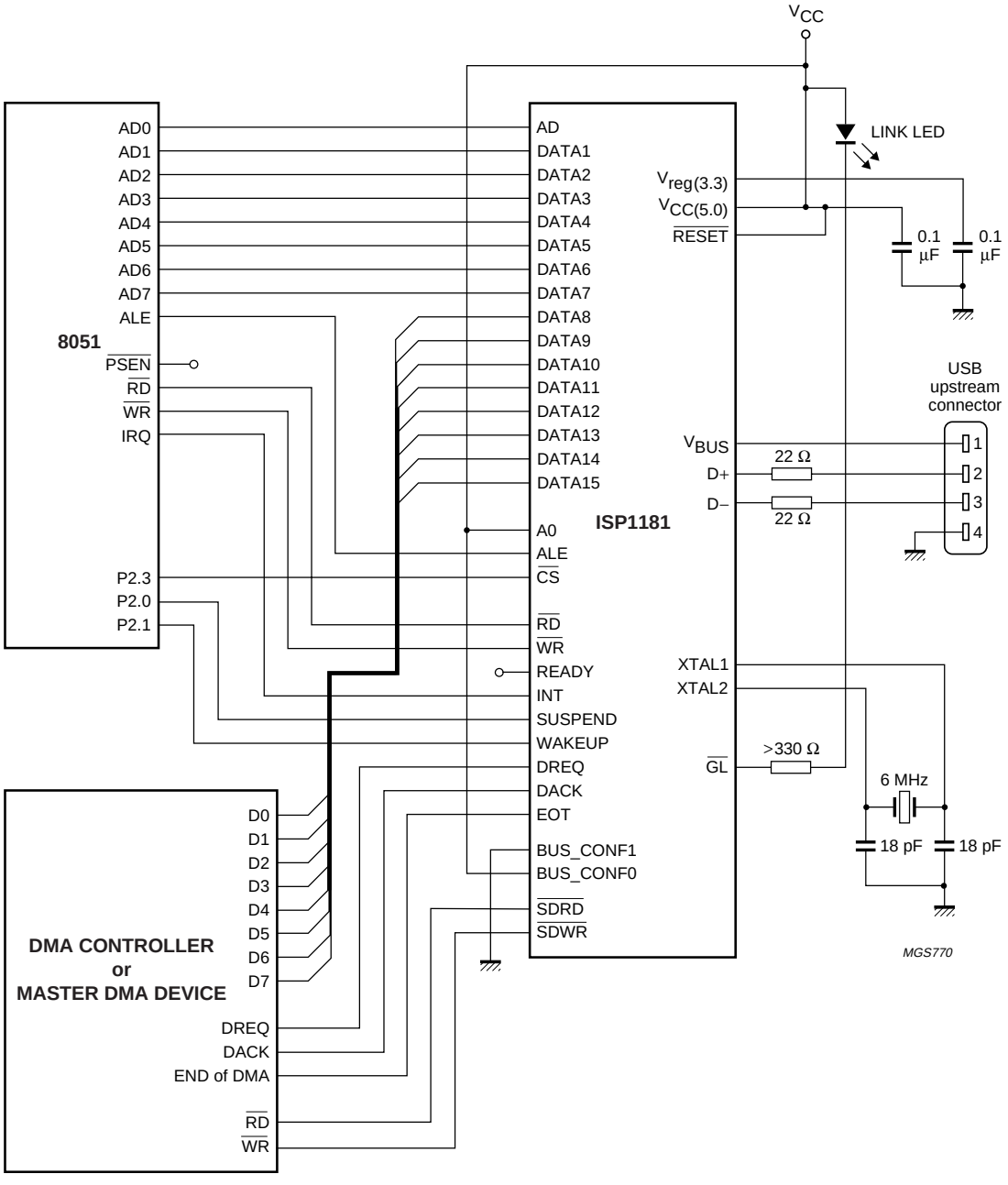


Fig 36. Typical interface circuit for bus configuration mode 1 (separate ports: 8-bit PIO and 8-bit DMA)

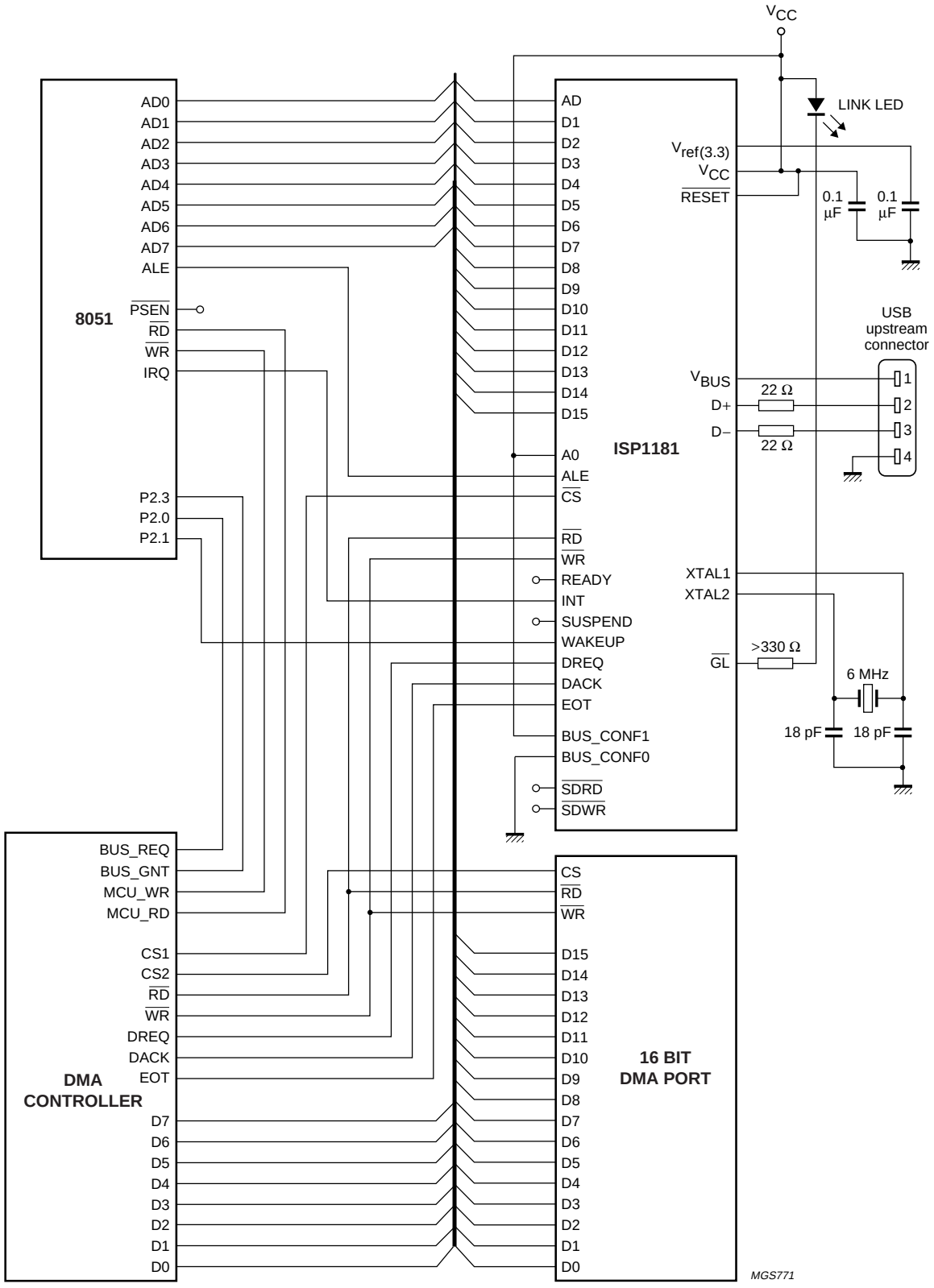


Fig 37. Typical interface circuit for bus configuration mode 2 (shared ports: 8-bit PIO, 8-bit or 16-bit DMA).

20.2 Interfacing ISP1181 with an H8S/2357 microcontroller

This section gives a summary of the ISP1181 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 35.

20.2.1 Interrupt Handling

- **ISP1181**: program the Hardware Configuration register to select an active LOW level for output INT (INTPOL = 0, see Table 22)
- **H8S/2357**: program the IRQ Sense Control Register (ISCRH and ISCRL) to specify low-level sensing for the IRQ input.

20.2.2 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 ISP1181 can be mapped to any address area, allowing easy interfacing when the ISP1181 is the only device in that area. If in the example circuit for bus configuration mode 0 (see Figure 35) the ISP1181 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 ISP1181.

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 ISP1181 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).

20.2.3 Using DMA

The ISP1181 can be configured for several methods of DMA with the H8S/2357 and other devices. The interface circuit in Figure 35 shows an example of the ISP1181 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:

- **ISP1181**:
  - 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.

20.2.4 Using H8S2357 I/O Ports

In the interface circuit of Figure 35 pin P1.1 of the H8S/2357 is configured as a general purpose output port. This pin drives the ISP1181's WAKEUP input to generate a remote wake-up.

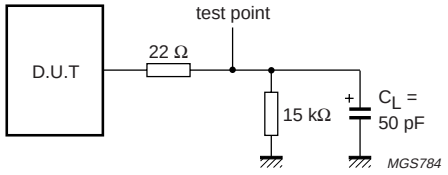
The H8S/2357 has 3 registers to configure port 1: Port 1 Data Direction Register (P1DDR), Port 1 Data Register (P1DR) and Port 1 Register (PORT1). Only registers P1DDR and P1DR must be configured, register PORT1 is only used to read the actual levels on the port pins.

For single

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

21. Test information

The dynamic characteristics of the analog I/O ports (D+ and D–) as listed in Table 57, were determined using the circuit shown in Figure 38.



Load capacitance:

C_L = 50 pF (full-speed mode)

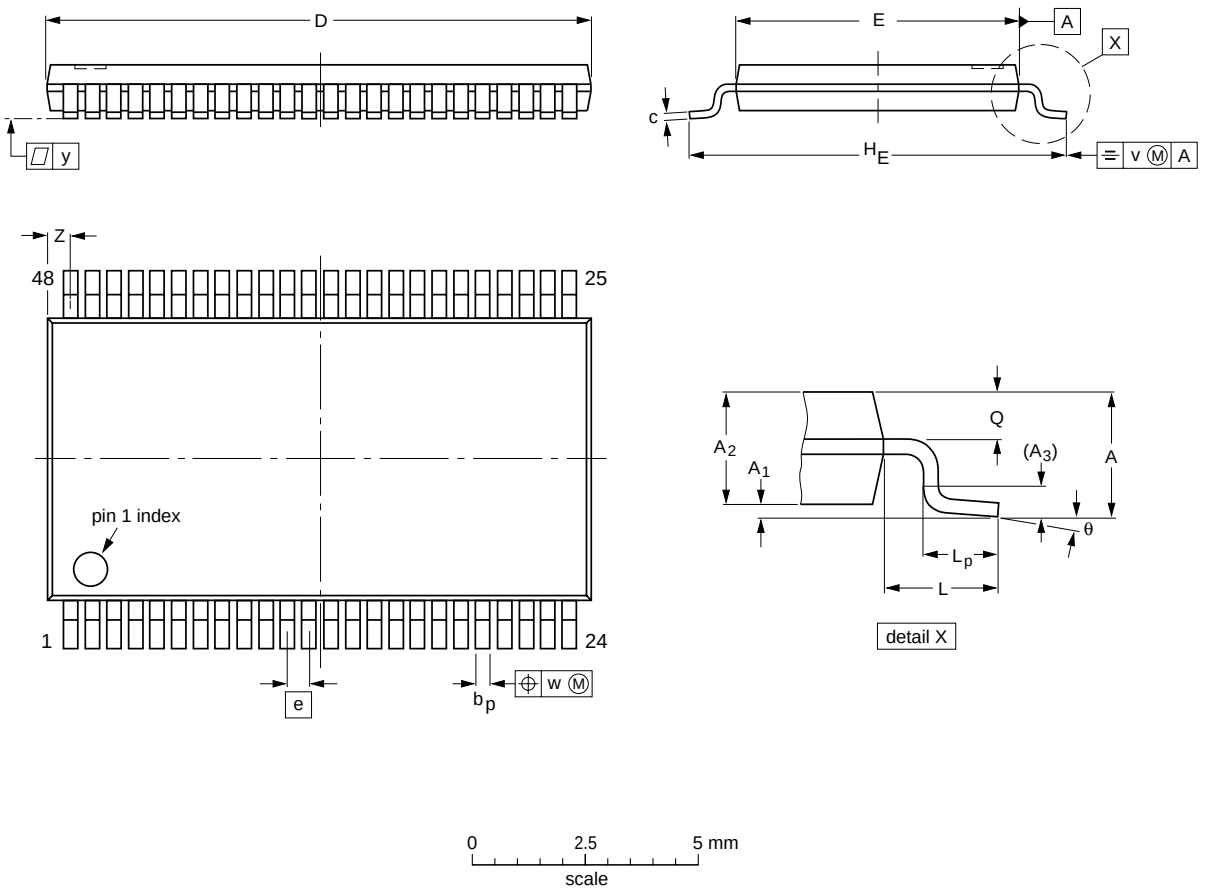
Speed:

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

Fig 38. Load impedance for D+ and D– pins.

22. 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	EIAJ				
SOT362-1		MO-153					~~95-02-10~~ 99-12-27

Fig 39. TSSOP48 package outline.

23. Soldering

23.1 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 is not always suitable for surface mount ICs, or for printed-circuit boards with high population densities. In these situations reflow soldering is often used.

23.2 Reflow soldering

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

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

Typical reflow peak temperatures range from 215 to 250 °C. The top-surface temperature of the packages should preferable be kept below 230 °C.

23.3 Wave soldering

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

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

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

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

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

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

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

23.4 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.

23.5 Package related soldering information

Table 63: Suitability of surface mount IC packages for wave and reflow soldering methods

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

[1] 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*.

[2] These packages are not suitable for wave soldering as a solder joint between the printed-circuit board and heatsink (at bottom version) can not be achieved, and as solder may stick to the heatsink (on top version).

[3] 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.

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

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

24. Revision history

Table 64: Revision history

Rev	Date	CPCN	Description
01	20000313		Objective specification; initial version.

25. Data sheet status

Datasheet status	Product status	Definition [1]
Objective specification	Development	This data sheet contains the design target or goal specifications for product development. Specification may change in any manner without notice.
Preliminary specification	Qualification	This data sheet contains preliminary data, and supplementary data will be published at a later date. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Product specification	Production	This data sheet contains final specifications. Philips Semiconductors reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

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

26. 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.

27. 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, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. 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.

28. 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 Royal Philips Electronics

OnNow — is a trademark of Microsoft Corp.

SMBus — is a bus specification for PC power management, developed by Intel Corp. based on the I^2C-bus from Royal Philips Electronics

SoftConnect — is a trademark of Royal Philips Electronics

Philips Semiconductors - a worldwide company

Argentina: see South America
Australia: Tel. +61 2 9704 8141, Fax. +61 2 9704 8139
Austria: Tel. +43 160 101, Fax. +43 160 101 1210
Belarus: Tel. +375 17 220 0733, Fax. +375 17 220 0773
Belgium: see The Netherlands
Brazil: see South America
Bulgaria: Tel. +359 268 9211, Fax. +359 268 9102
Canada: Tel. +1 800 234 7381
China/Hong Kong: Tel. +852 2 319 7888, Fax. +852 2 319 7700
Colombia: see South America
Czech Republic: see Austria
Denmark: Tel. +45 3 288 2636, Fax. +45 3 157 0044
Finland: Tel. +358 961 5800, Fax. +358 96 158 0920
France: Tel. +33 14 099 6161, Fax. +33 14 099 6427
Germany: Tel. +49 40 23 5360, Fax. +49 402 353 6300
Hungary: see Austria
India: Tel. +91 22 493 8541, Fax. +91 22 493 8722
Indonesia: see Singapore
Ireland: Tel. +353 17 64 0000, Fax. +353 17 64 0200
Israel: Tel. +972 36 45 0444, Fax. +972 36 49 1007
Italy: Tel. +39 039 203 6838, Fax +39 039 203 6800
Japan: Tel. +81 33 740 5130, Fax. +81 3 3740 5057
Korea: Tel. +82 27 09 1412, Fax. +82 27 09 1415
Malaysia: Tel. +60 37 50 5214, Fax. +60 37 57 4880
Mexico: Tel. +9-5 800 234 7381
Middle East: see Italy

For all other countries apply to: Philips Semiconductors,
International Marketing & Sales Communications,
Building BE, P.O. Box 218, 5600 MD EINDHOVEN,
The Netherlands, Fax. +31 40 272 4825

Netherlands: Tel. +31 40 278 2785, Fax. +31 40 278 8399
New Zealand: Tel. +64 98 49 4160, Fax. +64 98 49 7811
Norway: Tel. +47 22 74 8000, Fax. +47 22 74 8341
Philippines: Tel. +63 28 16 6380, Fax. +63 28 17 3474
Poland: Tel. +48 22 5710 000, Fax. +48 22 5710 001
Portugal: see Spain
Romania: see Italy
Russia: Tel. +7 095 755 6918, Fax. +7 095 755 6919
Singapore: Tel. +65 350 2538, Fax. +65 251 6500
Slovakia: see Austria
Slovenia: see Italy
South Africa: Tel. +27 11 471 5401, Fax. +27 11 471 5398
South America: Tel. +55 11 821 2333, Fax. +55 11 829 1849
Spain: Tel. +34 33 01 6312, Fax. +34 33 01 4107
Sweden: Tel. +46 86 32 2000, Fax. +46 86 32 2745
Switzerland: Tel. +41 14 88 2686, Fax. +41 14 81 7730
Taiwan: Tel. +886 22 134 2865, Fax. +886 22 134 2874
Thailand: Tel. +66 27 45 4090, Fax. +66 23 98 0793
Turkey: Tel. +90 216 522 1500, Fax. +90 216 522 1813
Ukraine: Tel. +380 44 264 2776, Fax. +380 44 268 0461
United Kingdom: Tel. +44 208 730 5000, Fax. +44 208 754 8421
United States: Tel. +1 800 234 7381
Uruguay: see South America
Vietnam: see Singapore
Yugoslavia: Tel. +381 11 3341 299, Fax. +381 11 3342 553

Internet: http://www.semiconductors.philips.com

(SCA69)

Contents

Date of release: 13 March 2000 **Document order number: 9397 750 06896**

PHILIPS

Let's make things better.