ST-NXP Wireless

IMPORTANT NOTICE

Dear customer,

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

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

- **Company name - NXP B.V.** is replaced with **ST-NXP Wireless**.

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

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

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

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

ST-NXP Wireless

ISP1761

Hi-Speed Universal Serial Bus On-The-Go controller

Rev. 05 — 13 March 2008 Product data sheet

1. General description

The ISP1761 is a single-chip Hi-Speed Universal Serial Bus (USB) On-The-Go (OTG) Controller integrated with advanced NXP slave host controller and the NXP ISP1582 peripheral controller.

The Hi-Speed USB host controller and peripheral controller comply to Ref. 1 “Universal Serial Bus Specification Rev. 2.0” and support data transfer speeds of up to 480 Mbit/s. The Enhanced Host Controller Interface (EHCI) core implemented in the host controller is adapted from Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”. The OTG controller adheres to Ref. 3 “On-The-Go Supplement to the USB Specification Rev. 1.3”.

The ISP1761 has three USB ports. Port 1 can be configured to function as a downstream port, an upstream port or an OTG port; ports 2 and 3 are always configured as downstream ports. The OTG port can switch its role from host to peripheral, and peripheral to host. The OTG port can become a host through the Host Negotiation Protocol (HNP) as specified in the OTG supplement.

2. Features

- Compliant with Ref. 1 “Universal Serial Bus Specification Rev. 2.0”; supporting data transfer at high-speed (480 Mbit/s), full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s)
- Integrated Transaction Translator (TT) for Original USB (full-speed and low-speed) peripheral support
- Three USB ports that support three operational modes:
  - Mode 1: Port 1 is an OTG controller port, and ports 2 and 3 are host controller ports
  - Mode 2: Ports 1, 2 and 3 are host controller ports
  - Mode 3: Port 1 is a peripheral controller port, and ports 2 and 3 are host controller ports
- Supports OTG Host Negotiation Protocol (HNP) and Session Request Protocol (SRP)
- Multitasking support with virtual segmentation feature (up to four banks)
- High-speed memory controller (variable latency and SRAM external interface)
- Directly addressable memory architecture
- Generic processor interface to most CPUs, such as Hitachi SH-3 and SH-4, NXP XA, Intel StrongARM, NEC and Toshiba MIPS, Freescale DragonBall and PowerPC Reduced Instruction Set Computer (RISC) processors
- Configurable 32-bit and 16-bit external memory data bus
- Supports Programmed I/O (PIO) and Direct Memory Access (DMA)
- Slave DMA implementation on CPU interface to reduce the host system’s CPU load

- Separate IRQ, DREQ and DACK lines for the host controller and the peripheral controller
- Integrated multi-configuration FIFO
- Double-buffering scheme increases throughput and facilitates real-time data transfer
- Integrated Phase-Locked Loop (PLL) with external 12 MHz crystal for low EMI
- Tolerant I/O for low voltage CPU interface (1.65 V to 3.3 V)
- 3.3 V-to-5.0 V external power supply input
- Integrated 5.0 V-to-1.8 V or 3.3 V-to-1.8 V voltage regulator (internal 1.8 V for low-power core)
- Internal power-on reset or low-voltage reset and block-dedicated software reset
- Supports suspend and remote wake-up
- Built-in overcurrent circuitry (analog overcurrent protection)
- Hybrid-power mode: $V_{CC(5V0)}$ (can be switched off), $V_{CC(I/O)}$ (permanent)
- Target total current consumption:
  - Normal operation; one port in high-speed active: I_{CC} < 100 mA when the internal charge pump is not used
  - Suspend mode: $I_{CC(susp)}$ < 150 μA at ambient temperature of +25 °C
- Available in LQFP128 and TFBGA128 packages
- Host controller-specific features
  - High performance USB host with integrated Hi-Speed USB transceivers; supports high-speed, full-speed and low-speed
  - EHCI core is adapted from Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”
  - Configurable power management
  - Integrated TT for Original USB peripheral support on all three ports
  - Integrated 64 kB high-speed memory (internally organized as 8 k × 64 bit)
  - Additional 2.5 kB separate memory for TT
  - Individual or global overcurrent protection with built-in sense circuits
  - Built-in overcurrent circuitry (digital or analog overcurrent protection)
- OTG controller-specific features
  - OTG transceiver: fully integrated; adheres to Ref. 3 “On-The-Go Supplement to the USB Specification Rev. 1.3”
  - Supports HNP and SRP for OTG dual-role devices
  - HNP: status and control registers for software implementation
  - SRP: status and control registers for software implementation
  - Programmable timers with high resolution (0.01 ms to 80 ms) for HNP and SRP
  - Supports external source of V_{BUS}
- Peripheral controller-specific features
  - High-performance USB peripheral controller with integrated Serial Interface Engine (SIE), FIFO memory and transceiver
  - Complies with Ref. 1 “Universal Serial Bus Specification Rev. 2.0” and most device class specifications
  - Supports auto Hi-Speed USB mode discovery and Original USB fallback capabilities
  - Supports high-speed and full-speed on the peripheral controller
  - Bus-powered or self-powered capability with suspend mode

- Slave DMA, fully autonomous and supports multiple configurations
- Seven IN endpoints, seven OUT endpoints and one fixed control IN and OUT endpoint
- Integrated 8 kB memory
- Software-controllable connection to the USB bus, SoftConnect

3. Applications

The ISP1761 can be used to implement a dual-role USB device in any application, USB host or USB peripheral, depending on the cable connection. If the dual-role device is connected to a typical USB peripheral, it behaves like a typical USB host. The dual-role device can also be connected to a PC or any other USB host and behave like a typical USB peripheral.

3.1 Host/peripheral roles

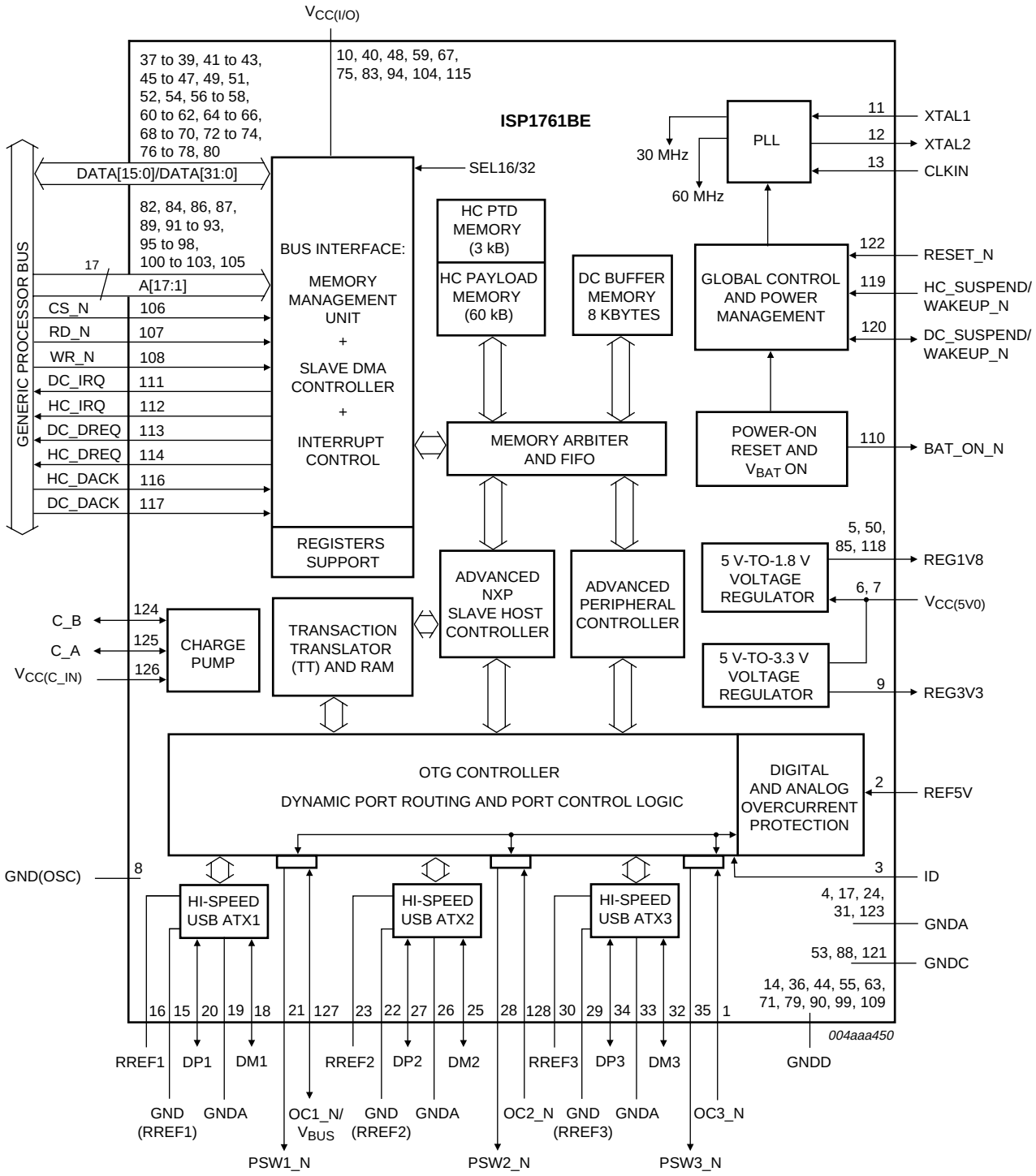
- Mobile phone to/from:
  - Mobile phone: exchange contact information
  - Digital still camera: e-mail pictures or upload pictures to the web
  - MP3 player: upload/download/broadcast music
  - Mass storage: upload/download files
  - Scanner: scan business cards
- Digital still camera to/from:
  - Digital still camera: exchange pictures
  - Mobile phone: e-mail pictures, upload pictures to the web
  - Printer: print pictures
  - Mass storage: store pictures
- Printer to/from:
  - Digital still camera: print pictures
  - Scanner: print scanned image
  - Mass storage: print files stored in a device
- MP3 player to/from:
  - MP3 player: exchange songs
  - Mass storage: upload/download songs
- Oscilloscope to/from:
  - Printer: print screen image
- Personal digital assistant to/from:
  - Personal digital assistant: exchange files
  - Printer: print files
  - Mobile phone: upload/download files
  - MP3 player: upload/download songs
  - Scanner: scan pictures
  - Mass storage: upload/download files
  - Global Positioning System (GPS): obtain directions, mapping information
  - Digital still camera: upload pictures
  - Oscilloscope: configure oscilloscope

4. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
ISP1761BE	LQFP128	plastic low profile quad flat package; 128 leads; body 14 × 20 × 1.4 mm	SOT425-1
ISP1761ET	TFBGA128	plastic thin fine-pitch ball grid array package; 128 balls; body 9 × 9 × 0.8 mm	SOT857-1

5. Block diagram



This figure shows the LQFP pinout. For the TFBGA ballout, see Table 2.

All ground pins should normally be connected to a common ground plane.

Fig 1. Block diagram

6. Pinning information

6.1 Pinning

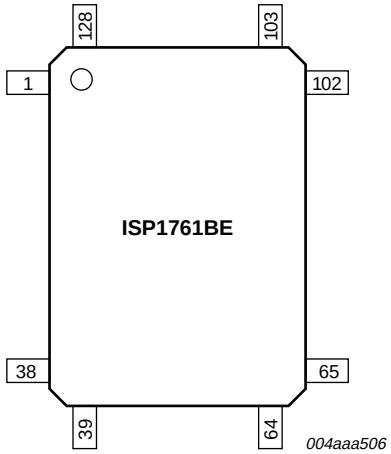


Fig 2. Pin configuration (LQFP128); top view

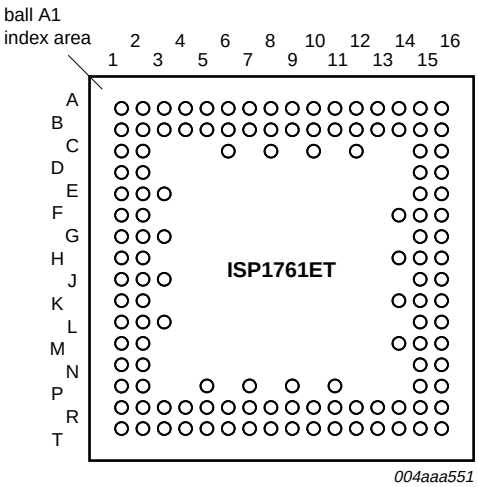


Fig 3. Pin configuration (TFBGA128); top view

6.2 Pin description

Table 2. Pin description

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
OC3_N	1	C2	AI/I	port 3 analog (5 V input) and digital overcurrent input; if not used, connect to $V_{CC(I/O)}$ through a 10 kΩ resistor input, 5 V tolerant
REF5V	2	A2	AI	5 V reference input for analog OC detector; connect a 100 nF decoupling capacitor
ID	3	B2	I	ID input to detect the default host or peripheral setting when port 1 is in OTG mode; pull-up to 3.3 V through a 4.7 kΩ resistor input, 3.3 V tolerant
GNDA	4	A1	-	analog ground
REG1V8	5	B1	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor; for details on additional capacitor placement, see Section 7.7
$V_{CC(5V0)}$	6	C1	P	input to internal regulators (3.0 V-to-5.5 V); connect a 100 nF decoupling capacitor; see Section 7.7
$V_{CC(5V0)}$	7	D2	P	input to internal regulators (3.0 V-to-5.5 V); connect a 100 nF decoupling capacitor; see Section 7.7
GND(OSC)	8	E3	-	oscillator ground
REG3V3	9	D1	P	regulator output (3.3 V); for decoupling only; connect a 100 nF capacitor and a 4.7 µF-to-10 µF capacitor; see Section 7.7
$V_{CC(I/O)}$	10	E2	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
XTAL1	11	E1	AI	12 MHz crystal connection input; connect to ground if an external clock is used
XTAL2	12	F2	AO	12 MHz crystal connection output
CLKIN	13	F1	I	12 MHz oscillator or clock input; when not in use, connect to $V_{CC(I/O)}$
GNDD	14	G3	-	digital ground
GND(RREF1)	15	G2	-	RREF1 ground
RREF1	16	G1	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF1 ground
GNDA[4]	17	H2	-	analog ground
DM1	18	H1	AI/O	downstream data minus port 1
GNDA	19	J3	-	analog ground
DP1	20	J2	AI/O	downstream data plus port 1
PSW1_N	21	J1	OD	power switch port 1, active LOW output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
GND(RREF2)	22	K2	-	RREF2 ground
RREF2	23	K1	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF2 ground
GNDA[5]	24	L3	-	analog ground
DM2	25	L1	AI/O	downstream data minus port 2
GNDA	26	L2	-	analog ground
DP2	27	M2	AI/O	downstream data plus port 2

Table 2. Pin description *...continued*

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
PSW2_N	28	M1	OD	power switch port 2, active LOW output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
GND(RREF3)	29	N2	-	RREF3 ground
RREF3	30	N1	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF3 ground
GNDA[6]	31	P2	-	analog ground
DM3	32	P1	AI/O	downstream data minus port 3
GNDA	33	R2	-	analog ground
DP3	34	R1	AI/O	downstream data plus port 3
PSW3_N	35	T1	OD	power switch port 3, active LOW output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
GNDD	36	T2	-	digital ground
DATA0	37	R3	I/O	data bit 0 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA1	38	T3	I/O	data bit 1 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA2	39	R4	I/O	data bit 2 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	40	T4	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
DATA3	41	P5	I/O	data bit 3 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA4	42	T5	I/O	data bit 4 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA5	43	R5	I/O	data bit 5 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	44	T6	-	digital ground
DATA6	45	R6	I/O	data bit 6 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA7	46	P7	I/O	data bit 7 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA8	47	T7	I/O	data bit 8 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	48	R7	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7

Table 2. Pin description *...continued*

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
DATA9	49	T8	I/O	data bit 9 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
REG1V8	50	R8	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor; for details on additional capacitor placement, see Section 7.7
DATA10	51	P9	I/O	data bit 10 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA11	52	T9	I/O	data bit 11 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDC	53	R9	-	core ground
DATA12	54	T10	I/O	data bit 12 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	55	R10	-	digital ground
DATA13	56	P11	I/O	data bit 13 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA14	57	T11	I/O	data bit 14 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA15	58	R11	I/O	data bit 15 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	59	T12	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
DATA16	60	R12	I/O	data bit 16 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA17	61	T13	I/O	data bit 17 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA18	62	R13	I/O	data bit 18 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	63	R14	-	digital ground
DATA19	64	T14	I/O	data bit 19 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA20	65	T15	I/O	data bit 20 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
DATA21	66	R15	I/O	data bit 21 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	67	P15	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
DATA22	68	T16	I/O	data bit 22 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA23	69	R16	I/O	data bit 23 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA24	70	P16	I/O	data bit 24 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	71	N16	-	digital ground
DATA25	72	N15	I/O	data bit 25 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA26	73	M15	I/O	data bit 26 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA27	74	M16	I/O	data bit 27 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	75	M14	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
DATA28	76	L16	I/O	data bit 28 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA29	77	L15	I/O	data bit 29 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA30	78	K16	I/O	data bit 30 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	79	K15	-	digital ground
DATA31	80	K14	I/O	data bit 31 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
TEST	81	J16	-	connect to ground
A1	82	H16	I	address pin 1 input, 3.3 V tolerant
$V_{CC(I/O)}$	83	J15	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7

Table 2. Pin description *...continued*

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
A2	84	H15	I	address pin 2 input, 3.3 V tolerant
REG1V8	85	G16	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor and a 4.7 µF-to-10 µF capacitor; see Section 7.7
A3	86	H14	I	address pin 3 input, 3.3 V tolerant
A4	87	F16	I	address pin 4 input, 3.3 V tolerant
GNDC	88	G15	-	core ground
A5	89	F15	I	address pin 5 input, 3.3 V tolerant
GNDD	90	E16	-	digital ground
A6	91	F14	I	address pin 6 input, 3.3 V tolerant
A7	92	E15	I	address pin 7 input, 3.3 V tolerant
A8	93	D16	I	address pin 8 input, 3.3 V tolerant
$V_{CC(I/O)}$	94	D15	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
A9	95	C16	I	address pin 9 input, 3.3 V tolerant
A10	96	C15	I	address pin 10 input, 3.3 V tolerant
A11	97	B16	I	address pin 11 input, 3.3 V tolerant
A12	98	B15	I	address pin 12 input, 3.3 V tolerant
GNDD	99	A16	-	digital ground
A13	100	A15	I	address pin 13 input, 3.3 V tolerant
A14	101	B14	I	address pin 14 input, 3.3 V tolerant
A15	102	A14	I	address pin 15 input, 3.3 V tolerant
A16	103	A13	I	address pin 16 input, 3.3 V tolerant
$V_{CC(I/O)}$	104	B13	P	digital voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
A17	105	C12	I	address pin 17 input, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
CS_N	106	A12	I	chip select assertion indicates the ISP1761 being accessed; active LOW input, 3.3 V tolerant
RD_N	107	B12	I	read enable; active LOW input, 3.3 V tolerant
WR_N	108	B11	I	write enable; active LOW input, 3.3 V tolerant
GNDD	109	A11	-	digital ground
BAT_ON_N	110	C10	OD	to indicate the presence of a minimum 3.3 V on pins 6 and 7 (open-drain); connect to $V_{CC(I/O)}$ through a 10 kΩ pull-up resistor output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
DC_IRQ	111	A10	O	peripheral controller interrupt signal output 4 mA drive, 3.3 V tolerant
HC_IRQ	112	B10	O	host controller interrupt signal output 4 mA drive, 3.3 V tolerant
DC_DREQ	113	A9	O	DMA controller request for the peripheral controller output 4 mA drive, 3.3 V tolerant
HC_DREQ	114	B9	O	DMA controller request for host controller output 4 mA drive, 3.3 V tolerant
$V_{CC(I/O)}$	115	C8	P	digital voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.7
HC_DACK	116	A8	I	host controller DMA request acknowledgment; when not in use, connect to $V_{CC(I/O)}$ through a 10 kΩ pull-up resistor input, 3.3 V tolerant
DC_DACK	117	B8	I	peripheral controller DMA request acknowledgment; when not in use, connect to $V_{CC(I/O)}$ through a 10 kΩ pull-up resistor input, 3.3 V tolerant
REG1V8	118	B7	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor; for details on additional capacitor placement, see Section 7.7
HC_SUSPEND /WAKEUP_N	119	A7	I/OD	host controller suspend and wake-up; 3-state suspend output (active LOW) and wake-up input circuits are connected together • HIGH = output is 3-state; the ISP1761 is in suspend mode • LOW = output is LOW; the ISP1761 is not in suspend mode connect to $V_{CC(I/O)}$ through an external 10 kΩ pull-up resistor output pad, open-drain, 4 mA output drive, 3.3 V tolerant
DC_SUSPEND /WAKEUP_N	120	C6	I/OD	peripheral controller suspend and wake-up; 3-state suspend output (active LOW) and wake-up input circuits are connected together • HIGH = output is 3-state; the ISP1761 is in suspend mode • LOW = output is LOW; the ISP1761 is not in suspend mode connect to $V_{CC(I/O)}$ through an external 10 kΩ pull-up resistor output pad, open-drain, 4 mA output drive, 3.3 V tolerant
GNDC	121	A6	-	core ground

Table 2. Pin description *...continued*

Symbol[1][2]	Pin		Type[3]	Description
	LQFP128	TFBGA128		
RESET_N	122	B6	I	external power-up reset; active LOW; when reset is asserted, it is expected that bus signals are idle, that is, not toggling input, 3.3 V tolerant **Remark:** During reset, ensure that all the input pins to the ISP1761 are not toggling and are in their inactive states.
GNDA	123	B5	-	analog ground
C_B	124	A5	AI/O	charge pump capacitor input; connect a 220 nF capacitor between this pin and pin 125
C_A	125	B4	AI/O	charge pump capacitor input; connect a 220 nF capacitor between this pin and pin 124
$V_{CC(C\_IN)}$	126	A4	P	charge pump input; connect to 3.3 V
OC1_N/V_{BUS}	127	B3	(AI/O)(I)	This pin has multiple functions: • Input: Port 1 OC1_N detection when port 1 is configured for host functionality and an external power switch is used; if not used, connect to $V_{CC(I/O)}$ through a 10 kΩ resistor; the 10 kΩ resistor is usually required by the open-drain output of the power-switch flag pin • Output: V_{BUS} out when internal charge pump is used and port 1 is configured for the OTG functionality; maximum 50 mA current capability; the overcurrent protection in this case is ensured by the internal charge pump current limitation; only for port 1 • Input: V_{BUS} input detection when port 1 is defined for the peripheral functionality 5 V tolerant
OC2_N	128	A3	AI/I	port 2 analog (5 V input) and digital overcurrent input; if not used, connect to $V_{CC(I/O)}$ through a 10 kΩ resistor input, 5 V tolerant

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

[2] All ground pins should normally be connected to a common ground plane.

[3] I = input only; O = output only; I/O = digital input/output; OD = open-drain output; AI/O = analog input/output; AI = analog input; P = power; (AI/O)(I) = analog input/output digital input; AI/I = analog input digital input.

[4] For port 1.

[5] For port 2.

[6] For port 3.

7. Functional description

7.1 ISP1761 internal architecture: advanced NXP slave host controller and hub

The EHCI block and the Hi-Speed USB hub block are the main components of the advanced NXP slave host controller.

The EHCI is the latest generation design, with improved data bandwidth. The EHCI in the ISP1761 is adapted from Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

The internal Hi-Speed USB hub block replaces the companion host controller block used in the original architecture of a Peripheral Component Interconnect (PCI) Hi-Speed USB host controller to handle full-speed and low-speed modes. The hardware architecture in the ISP1761 is simplified to help reduce cost and development time, by eliminating the additional work involved in implementing the OHCI software required to support full-speed and low-speed modes.

Figure 4 shows the internal architecture of the ISP1761. The ISP1761 implements an EHCI that has an internal port, the root hub port (not available externally), on which the internal hub is connected. The three external ports are always routed to the internal hub. The internal hub is a Hi-Speed USB hub including the TT.

Remark: The root hub must be enabled and the internal hub must be enumerated. Enumerate the internal hub as if it is externally connected. For details, refer to Ref. 5 “Interfacing the ISP176x to the Intel PXA25x processor (AN10037)”.

At the host controller reset and initialization, the internal root hub port will be polled until a new connection is detected, showing the connection of the internal hub.

The internal Hi-Speed USB hub is enumerated using a sequence similar to a standard Hi-Speed USB hub enumeration sequence, and the polling on the root hub is stopped because the internal Hi-Speed USB hub will never be disconnected. When enumerated, the internal hub will report the three externally available ports.

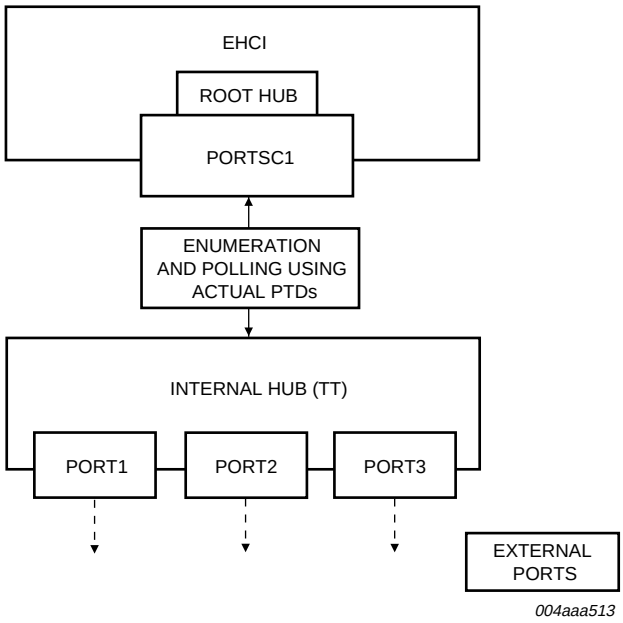


Fig 4. Internal hub

7.1.1 Internal clock scheme and port selection

Figure 5 shows the internal clock scheme of the ISP1761. The ISP1761 has three ports.

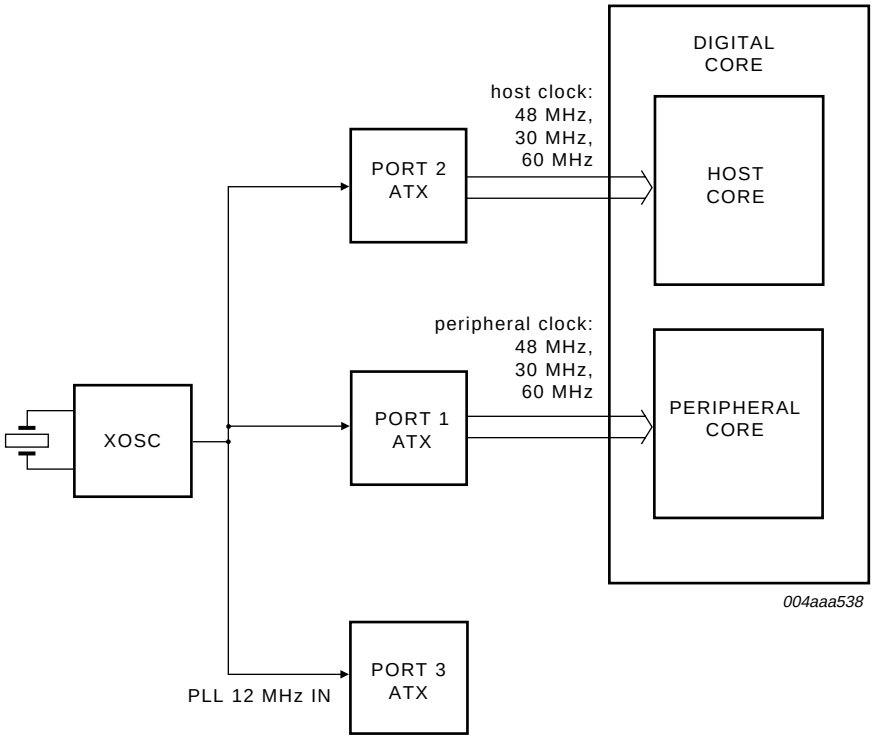


Fig 5. ISP1761 clock scheme

Port 2 does not need to be enabled using software if only port 1 or port 3 is used. No port needs to be disabled by external pull-up resistors, if not used. The DP and DM of the unused ports need not be externally pulled HIGH because there are internal pull-down resistors on each port that are enabled by default.

Table 3 lists the various port connection scenarios.

Table 3. Port connection scenarios

Port configuration	Port 1	Port 2	Port 3
One port (port 1)	DP and DM are routed to USB connector	DP and DM are not connected (left open)	DP and DM are not connected (left open)
One port (port 2)	DP and DM are not connected (left open)	DP and DM are routed to USB connector	DP and DM are not connected (left open)
One port (port 3)	DP and DM are not connected (left open)	DP and DM are not connected (left open)	DP and DM are routed to USB connector
Two ports (ports 1 and 2)	DP and DM are routed to USB connector	DP and DM are routed to USB connector	DP and DM are not connected (left open)
Two ports (ports 2 and 3)	DP and DM are not connected (left open)	DP and DM are routed to USB connector	DP and DM are routed to USB connector
Two ports (ports 1 and 3)	DP and DM are routed to USB connector	DP and DM are not connected (left open)	DP and DM are routed to USB connector
Three ports (ports 1, 2 and 3)	DP and DM are routed to USB connector	DP and DM are routed to USB connector	DP and DM are routed to USB connector

7.2 Host controller buffer memory block

7.2.1 General considerations

The internal addressable host controller buffer memory is 63 kB. The 63 kB effective memory size is the result of subtracting the size of the registers (1 kB) from the total addressable memory space defined by the ISP1761 (64 kB). This is an optimized value to achieve the highest performance with minimal cost.

The ISP1761 is a slave host controller. This means that it does not need access to the local bus of the system to transfer data from the system memory to the ISP1761 internal memory, unlike the case of the original PCI Hi-Speed USB host controllers. Therefore, correct data must be transferred to both the Philips Transfer Descriptor (PTD) area and the payload area by PIO (using CPU access) or programmed DMA.

The ‘slave-host’ architecture ensures better compatibility with most of the processors present in the market today because not all processors allow a ‘bus-master’ on the local bus. It also allows better load balancing of the processor’s local bus because only the internal bus arbiter of the processor controls the transfer of data dedicated to USB. This prevents the local bus from being busy when other more important transfers may be in the queue; and therefore achieving a ‘linear’ system data flow that has less impact on other processes running at the same time.

The considerations mentioned are also the main reason for implementing the pre-fetching technique, instead of using a READY signal. The resulting architecture avoids ‘freezing’ of the local bus, by asserting READY, enhancing the ISP1761 memory access time, and avoiding introduction of programmed additional wait states. For details, see Section 7.3.

The total amount of memory allocated to the payload determines the maximum transfer size specified by a PTD, a larger internal memory size results in less CPU interruption for transfer programming. This means less time spent in context switching, resulting in better CPU usage.

A larger buffer also implies that a larger amount of data can be transferred. This transfer, however, can be done over a longer period of time, to maintain the overall system performance. Each transfer of the USB data on the USB bus can span up to a few milliseconds before requiring further CPU intervention for data movement.

The internal architecture of the ISP1761 allows a flexible definition of the memory buffer for optimization of the data transfer on the CPU extension bus and the USB. It is possible to implement different data transfer schemes, depending on the number and type of USB devices present. For example: push-pull; data can be written to half of the memory while data in the other half is being accessed by the host controller and sent on the USB bus. This is useful especially when a high-bandwidth ‘continuous or periodic’ data flow is required.

Through an analysis of the hardware and software environment regarding the usual data flow and performance requirements of most embedded systems, NXP has determined the optimal size for the internal buffer as approximately 64 kB.

7.2.2 Structure of the ISP1761 host controller memory

The 63 kB of internal memory consists of the PTD area and the payload area.

The PTD memory zone is divided into three dedicated areas for each main type of USB transfer: Isochronous (ISO), Interrupt (INT) and Asynchronous Transfer List (ATL). As shown in Table 4, the PTD areas for ISO, INT and ATL are grouped at the beginning of the memory, occupying the address range 0400h to 0FFFh, following the register address space. The payload or data area occupies the next memory address range 1000h to FFFFh, meaning that 60 kB of memory are allocated for the payload data.

A maximum of 32 PTD areas and their allocated payload areas can be defined for each type of transfer. The structure of a PTD is similar for every transfer type and consists of eight Double Words (DWs) that must be correctly programmed for a correct USB data transfer. The reserved bits of a PTD must be set to logic 0. A detailed description of the PTD structure can be found in Section 8.5.

The transfer size specified by the PTD determines the contiguous USB data transfer that can be performed without any CPU intervention. The respective payload memory area must be equal to the transfer size defined. The maximum transfer size is flexible and can be optimized, depending on the number and nature of USB devices or PTDs defined and their respective MaxPacketSize.

The CPU will program the DMA to transfer the necessary data in the payload memory. The next CPU intervention will be required only when the current transfer is completed and DMA programming is necessary to transfer the next data payload. This is normally signaled by the IRQ that is generated by the ISP1761 on completing the current PTD, meaning all the data in the payload area was sent on the USB bus. The external IRQ signal is asserted according to the settings in the IRQ Mask OR or IRQ MASK AND registers, see Section 8.4.

The RAM is structured in blocks of PTDs and payloads so that while the USB is executing on an active transfer-based PTD, the processor can simultaneously fill up another block area in the RAM. A PTD and its payload can then be updated on-the-fly without stopping or delaying any other USB transaction or corrupting the RAM data.

Some of the design features are:

- The address range of the internal RAM buffer is from 0400h to FFFFh.
- The internal memory contains isochronous, interrupt and asynchronous PTDs, and respective defined payloads.
- All accesses to the internal memory are double-word aligned.
- Internal memory address range calculation:

  Memory address = (CPU address – 0400h) (shift right >> 3). Base address is 0400h.

Table 4. Memory address

Memory map	CPU address	Memory address
ISO	0400h to 07FFh	0000h to 007Fh
INT	0800h to 0BFFh	0080h to 00FFh
ATL	0C00h to 0FFFh	0100h to 017Fh
Payload	1000h to FFFFh	0180h to 1FFFh

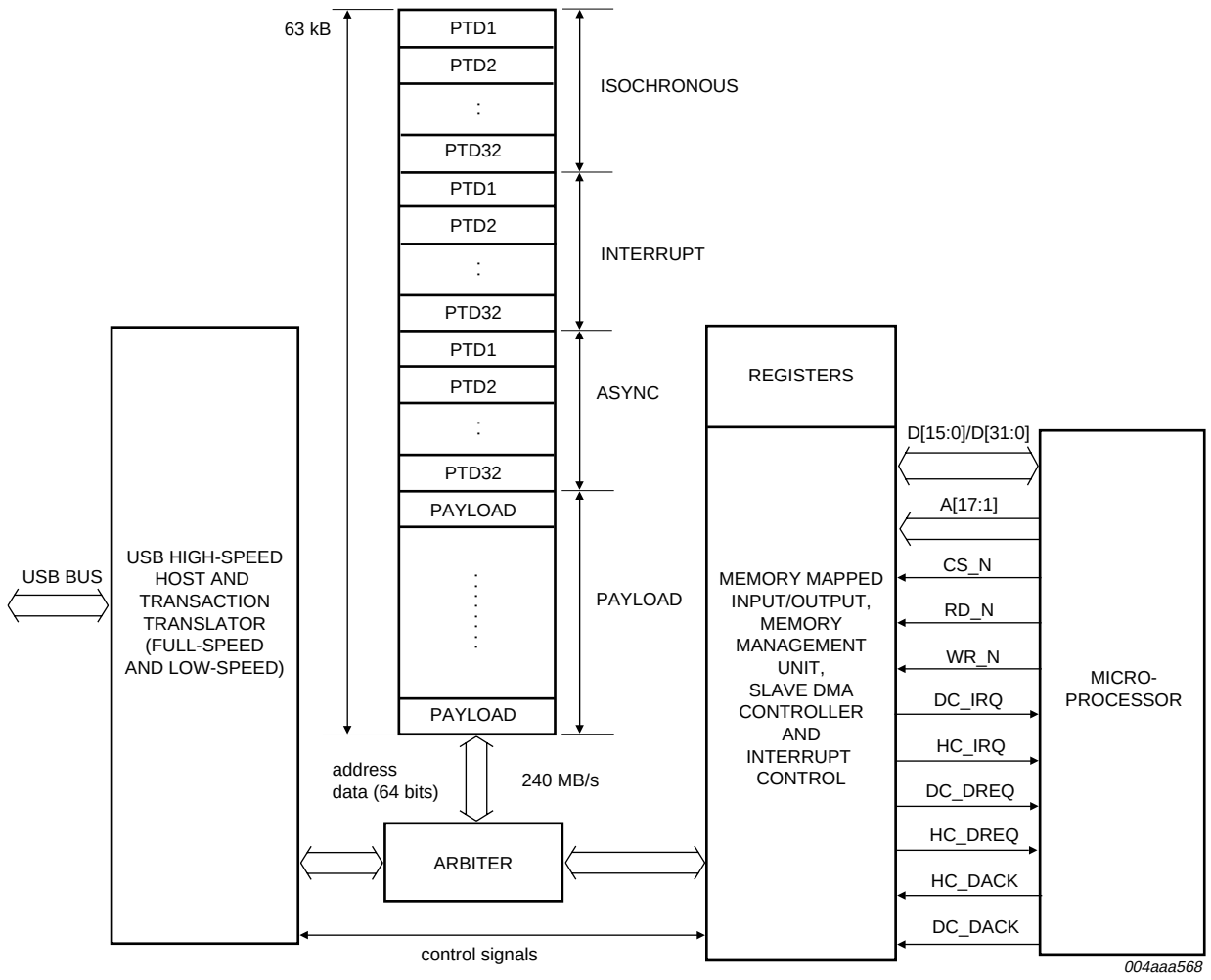


Fig 6. Memory segmentation and access block diagram

Both the CPU interface logic and the USB host controller require access to the internal ISP1761 RAM at the same time. The internal arbiter controls these accesses to the internal memory, organized internally on a 64-bit data bus width, allowing a maximum bandwidth of 240 MB/s. This bandwidth avoids any bottleneck on accesses both from the CPU interface and the internal USB host controller.

7.3 Accessing the ISP1761 host controller memory: PIO and DMA

The CPU interface of the ISP1761 can be configured for a 16-bit or 32-bit data bus width.

When the ISP1761 is configured for a 16-bit data bus width, the upper unused 16 data lines must be pulled up to $V_{CC(I/O)}$. This can be achieved by connecting DATA[31:16] lines together to a single 10 kΩ pull-up resistor. The 16-bit or 32-bit data bus width configuration is done by programming bit 8 of the HW Mode Control register. This will determine the register and memory access types in both PIO and DMA modes to all internal blocks: host controller, peripheral controller and OTG controller. All accesses must be word-aligned for 16-bit mode and double-word aligned for 32-bit mode, where one word = 16 bits. When accessing the host controller registers in 16-bit mode, the register

access must always be completed using two subsequent accesses. In the case of a DMA transfer, the 16-bit or 32-bit data bus width configuration will determine the number of bursts that will complete a certain transfer length.

In PIO mode, CS_N, WR_N and RD_N are used to access registers and memory. In DMA mode, the data validation is performed by DACK, instead of CS_N, together with the WR_N and RD_N signals. The DREQ signal will always be asserted as soon as the ISP1761 DMA is enabled.

7.3.1 PIO mode access, memory read cycle

The following method is implemented to reduce the read access timing in a memory read:

- The Memory register contains the starting address and the bank selection to read from the memory. Before every new read cycle of the same or different banks, an appropriate value is written to this register.
- Once a value is written to this register, the address is stored in the FIFO of that bank and is then used to pre-fetch data for the memory read of that bank.

  For every subsequent read operation executed at a contiguous address, the address pointer corresponding to that bank is automatically incremented to pre-fetch the next data to be sent to the CPU.

  Memory read accesses for multiple banks can be interleaved. In this case, the FIFO block handles the multiplexing of appropriate data to the CPU.
- The address written to the Memory register is incremented and used to successively pre-fetch data from the memory irrespective of the value on the address bus for each bank, until a new value for a bank is written to the Memory register. This is valid only when the address refers to the memory space (400h to FFFFh).

  For example, consider the following sequence of operations:

  – Write the starting (read) address 4000h and bank1 = 01 to the Memory register. When RD_N is asserted for three cycles with A[17:16] = 01, the returned data corresponds to addresses 4000h, 4004h and 4008h.

    Remark: Once 4000h is written to the Memory register for bank1, the bank select value determines the successive incremental addresses used to fetch data. That is, the fetching of data is independent of the address on A[15:0] lines.

  – Write the starting (read) address 4100h and bank2 = 10 to the Memory register. When RD_N is asserted for four cycles with A[17:16] = 10, the returned data corresponds to addresses 4100h, 4104h, 4108h and 410Ch.

    Consequently, the RD_N assertion with A[17:16] = 01 will return data from 400Ch because the bank1 read stopped there in the previous cycle. Also, RD_N assertions with A[17:16] = 10 will now return data from 4110h because the bank2 read stopped there in the previous cycle.

7.3.2 PIO mode access, memory write cycle

The PIO memory write access is similar to a normal memory access. It is not necessary to set the pre-fetching address before a write cycle to memory.

The ISP1761 internal write address will not automatically be incremented during consecutive write accesses, unlike in a series of ISP1761 memory read cycles. The memory write address must be incremented before every access.

7.3.3 PIO mode access, register read cycle

The PIO register read access is similar to a general register access. It is not necessary to set a pre-fetching address before a register read.

The ISP1761 register read address will not automatically be incremented during consecutive read accesses, unlike in a series of ISP1761 memory read cycles. The ISP1761 register read address must be correctly specified before every access.

7.3.4 PIO mode access, register write cycle

The PIO register write access is similar to a general register access. It is not necessary to set a pre-fetching address before a register write.

The ISP1761 register write address will not automatically be incremented during consecutive write accesses, unlike in a series of ISP1761 memory read cycles. The ISP1761 register write address must be correctly specified before every access.

7.3.5 DMA mode, read and write operations

The internal ISP1761 host controller DMA is a slave DMA. The host system processor or DMA must ensure the data transfer to or from the ISP1761 memory.

The ISP1761 DMA supports a DMA burst length of 1, 4, 8 and 16 cycles for both the 16-bit and 32-bit data bus width. DREQ will be asserted at the beginning of the first burst of a DMA transfer and will be de-asserted on the last cycle, RD_N or WR_N active pulse, of that burst. It will be reasserted shortly after the DACK de-assertion, as long as the DMA transfer counter was not reached. DREQ will be de-asserted on the last cycle when the DMA transfer counter is reached and will not be reasserted until the DMA reprogramming is performed. Both the DREQ and DACK signals are programmable as active LOW or active HIGH, according to system requirements.

The DMA start address must be initialized in the respective register, and the subsequent transfers will automatically increment the internal ISP1761 memory address. A register or memory access or access to other system memory can occur in between DMA bursts, whenever the bus is released because DACK is de-asserted, without affecting the DMA transfer counter or the current address.

Any memory area can be accessed by the system's DMA at any starting address because there are no predefined memory blocks. The DMA transfer must start on a word or double word address, depending on whether the data bus width is set to 16-bit or 32-bit. DMA is the most efficient method to initialize the payload area, to reduce the CPU usage and overall system loading.

The ISP1761 does not implement EOT to signal the end of a DMA transfer. If programmed, an interrupt may be generated by the ISP1761 at the end of the DMA transfer.

The slave DMA of the ISP1761 will issue a DREQ to the DMA controller of the system to indicate that it is programmed for transfer and data is ready. The system DMA controller may also start a transfer without the need of the DREQ, if the ISP1761 memory is available for the data transfer and the ISP1761 DMA programming is completed.

It is also possible that the system's DMA will perform a memory-to-memory type of transfer between the system memory and the ISP1761 memory. The ISP1761 will be accessed in PIO mode. Consequently, memory read operations must be preceded by initializing the Memory register (address 033Ch), as described in Section 7.3.1. No IRQ will be generated by the ISP1761 on completing the DMA transfer but an internal processor interrupt may be generated to signal that the DMA transfer is completed. This is mainly useful in implementing the double-buffering scheme for data transfer to optimize the USB bandwidth.

The ISP1761 DMA programming involves:

- Set the active levels of signals DREQ and DACK in the HW Mode Control register.
- The DMA Start Address register contains the first memory address at which the data transfer will start. It must be word-aligned in 16-bit data bus mode and double word aligned in 32-bit data bus mode.
- The programming of the HcDMAConfiguration register specifies:
  - The type of transfer that will be performed: read or write.
  - The burst size, expressed in bytes, is specified, regardless of the data bus width. For the same burst size, a double number of cycles will be generated in 16-bit mode data bus width as compared to 32-bit mode.
  - The transfer length, expressed in number of bytes, defines the number of bursts. The DREQ will be de-asserted and asserted to generate the next burst, as long as there are bytes to be transferred. At the end of a transfer, the DREQ will be de-asserted and an IRQ can be generated if DMAEOTINT (bit 3 in the HcInterrupt register) is set. The maximum DMA transfer size is equal to the maximum memory size. The transfer size can be an odd or even number of bytes, as required. If the transfer size is an odd number of bytes, the number of bytes transferred by the system's DMA is equal to the next multiple of two for the 16-bit data bus width or four for the 32-bit data bus width. For a write operation, however, only the specified odd number of bytes in the ISP1761 memory will be affected.
  - Enable ENABLE_DMA (bit 1) of the HcDMAConfiguration register to determine the assertion of DREQ immediately after setting the bit.

After programming the preceding parameters, the system's DMA may be enabled, waiting for the DREQ to start the transfer or immediate transfer may be started.

The programming of the system's DMA must match the programming of the ISP1761 DMA parameters. Only one DMA transfer may take place at a time. A PIO mode data transfer may occur simultaneously with a DMA data transfer, in the same or a different memory area.

7.4 Interrupts

The ISP1761 will assert the IRQ according to the source or event in the HcInterrupt register. The main steps to enable the IRQ assertion are:

1. Set GLOBAL_INTR_EN (bit 0) in the HW Mode Control register.
2. Define the IRQ active as level or edge in INTR_LEVEL (bit 1) of the HW Mode Control register.

3. Define the IRQ polarity as active LOW or active HIGH in INTR_POL (bit 2) of the HW Mode Control register. These settings must match IRQ settings of the host processor.

   By default, interrupt is level-triggered and active LOW.

4. Program the individual Interrupt Enable bits in the HcInterruptEnable register. The software will need to clear the Interrupt Status bits in the HcInterrupt register before enabling individual interrupt enable bits.

Additional IRQ characteristics can be adjusted in the Edge Interrupt Count register, as necessary, applicable only when IRQ is set to be edge-active; a pulse of a defined width is generated every time IRQ is active.

Bits 15 to 0 of the Edge Interrupt Count register define the IRQ pulse width. The maximum pulse width that can be programmed is FFFFh, corresponding to a 1 ms pulse width. This setting is necessary for certain processors that may require a different minimum IRQ pulse width from the default value. The default IRQ pulse width set at power-on is approximately 500 ns.

Bits 31 to 24 of the Edge Interrupt Count register define the minimum interval between two interrupts to avoid frequent interrupts to the CPU. The default value of 00h attributed to these bits determines the normal IRQ generation, without any delay. When a delay is programmed and the IRQ becomes active after the respective delay, several IRQ events may have already occurred.

All the interrupt events are represented by the respective bits allocated in the HcInterrupt register. There is no mechanism to show the order or the moment occurrence of an interrupt.

The asserted bits in the HcInterrupt register can be cleared by writing back the same value to the HcInterrupt register. This means that writing logic 1 to each of the set bits will reset that corresponding bits to the initial inactive state.

The IRQ generation rules that apply according to the preceding settings are:

- If an event of interrupt occurs but the respective bit in the Interrupt Enable register is not set, then the respective HcInterrupt register bit is set but the interrupt signal is not asserted.

  An interrupt will be generated when interrupt is enabled and the respective bit in the Interrupt Enable register is set.
- For a level trigger, an interrupt signal remains asserted until the processor clears the HcInterrupt register by writing logic 1 to clear the HcInterrupt register bits that are set.
- If an interrupt is made edge-sensitive and is asserted, writing to clear the HcInterrupt register will not have any effect because the interrupt will be asserted for a prescribed amount of clock cycles.
- The clock stopping mechanism does not affect the generation of an interrupt. This is useful during the suspend and resume cycles, when an interrupt is generated to signal a wake-up event.

The IRQ generation can also be conditioned by programming the IRQ Mask OR and IRQ Mask AND registers.

With the help of the IRQ Mask AND and IRQ Mask OR registers for each type of transfer (ISO, INT and bulk), software can determine which PTDs get priority and an interrupt will be generated when the AND or OR conditions are met. The PTDs that are set will wait until the respective bits of the remaining PTDs are set and then all PTDs generate an interrupt request to the CPU together.

The registers definition shows that the AND or OR conditions are applicable to the same category of PTDs: ISO, INT and ATL.

When an IRQ is generated, the PTD Done Map registers and the respective V bits will show which PTDs were completed.

The rules that apply to the IRQ Mask AND or IRQ Mask OR settings are:

- The OR mask has a higher priority over the AND mask. An IRQ is generated if bit n of done map is set and the corresponding bit n of the OR mask register is set.
- If the OR mask for any done bit is not set, then the AND mask comes into picture. An IRQ is generated if all the corresponding done bits of the AND Mask register are set. For example: If bits 2, 4 and 10 are set in the AND Mask register, an IRQ is generated only if bits 2, 4, 10 of the done map are set.
- If using the IRQ interval setting for the bulk PTD, an interrupt will only occur at the regular time interval as programmed in the ATL Done Timeout register. Even if an interrupt event occurs before the time-out of the register, no IRQ will be generated until the time is up.

For an example on using the IRQ Mask AND or IRQ Mask OR registers, without the ATL Done Timeout register, see Table 5.

The AND function: activate the IRQ only if PTDs 1, 2 and 4 are done.

The OR function: if any of the PTDs 7, 8 or 9 are done, an IRQ for each of the PTD will be raised.

Table 5. Using the IRQ Mask AND or IRQ Mask OR registers

PTD	AND register	OR register	Time	PTD done	IRQ
1	1	0	1 ms	1	-
2	1	0	-	1	-
3	0	0	-	-	-
4	1	0	3 ms	1	active because of AND
5	0	0	-	-	-
6	0	0	-	-	-
7	0	1	5 ms	1	active because of OR
8	0	1	6 ms	1	active because of OR
9	0	1	7 ms	1	active because of OR

7.5 Phase-Locked Loop (PLL) clock multiplier

The internal PLL requires a 12 MHz input, which can be a 12 MHz crystal or a 12 MHz clock already existing in the system with a precision better than 50 ppm. This allows the use of a low-cost 12 MHz crystal that also minimizes ElectroMagnetic Interference (EMI). When an external crystal is used, make sure the CLKIN pin is connected to $V_{CC(I/O)}$.

The PLL block generates all the main internal clocks required for normal functionality of various blocks: 30 MHz, 48 MHz and 60 MHz.

No external components are required for the PLL operation.

7.6 Power management

The ISP1761 implements a flexible power management scheme, allowing various power saving stages.

The usual powering scheme implies programming EHCI registers and the internal Hi-Speed USB (USB 2.0) hub in the same way it is done in a PCI Hi-Speed USB host controller with a Hi-Speed USB hub attached.

While the ISP1761 is set in suspend mode, main internal clocks will be stopped to ensure minimum power consumption. An internal LazyClock of 100 kHz $\pm$ 40 % will continue running. This allows initiating a resume on one of these events:

- External USB device connect or disconnect
- CS_N signal asserted when the ISP1761 is accessed
- Driving the HC_SUSPEND/WAKEUP_N pin to a LOW logical level will wake up the host controller, and driving the DC_SUSPEND/WAKEUP_N pin to a LOW logical level will wake up the peripheral controller

The HC_SUSPEND/WAKEUP_N and DC_SUSPEND/WAKEUP_N pins are bidirectional. These pins must be connected to the GPIO pins of a processor.

The awake state can be verified by reading the LOW level of this pin. If the level is HIGH, it means that the ISP1761 is in the suspend state.

HC_SUSPEND/WAKEUP_N and DC_SUSPEND/WAKEUP_N require pull-up resistors because in the ISP1761 suspended state these pins become 3-state and can be pulled down, driving them externally by switching the processor's GPIO lines to output mode to generate the ISP1761 wake-up.

The HC_SUSPEND/WAKEUP_N and DC_SUSPEND/WAKEUP_N pins are 3-state output and also input to the internal wake-up logic.

When in suspend mode, the ISP1761 internal wake-up circuitry will sense the status of the HC_SUSPEND/WAKEUP_N and DC_SUSPEND/WAKEUP_N pins:

- If the pins remain pulled-up, no wake-up will be generated because a HIGH is sensed by the internal wake-up circuit.
- If the pins are externally pulled LOW, for example, by the GPIO lines or just a test by jumpers, the input to the wake-up circuitry becomes LOW and the wake-up is internally initiated.

The resume state has a clock-off count timer defined by bits 31 to 16 of the Power Down Control register. The default value of this timer is 10 ms, meaning that the resume state will be maintained for 10 ms. If during this time, the RUN/STOP bit in the USBCMD register is set to logic 1, the host controller will go into a permanent resume; the normal functional state. If the RUN/STOP bit is not set during the time determined by the clock-off

count, the ISP1761 will switch back to suspend mode after the specified time. The maximum delay that can be programmed in the clock-off count field is approximately 500 ms.

Additionally, the Power Down Control register allows ISP1761 internal blocks to disable for lower power consumption as defined in Section 8.3.11.

The lowest suspend current, $I_{CC(susp)}$, that can be achieved is approximately 150 µA at ambient temperature of +25 °C. The suspend current will increase with the increase in temperature, with approximately 300 µA at 40 °C and up to a typical 1 mA at 85 °C. The system is not in suspend mode when its temperature increases above 40 °C. Therefore, even a 1 mA current consumption by the ISP1761 in suspend mode can be considered negligible. In normal environmental conditions, when the system is in suspend mode, the maximum ISP1761 temperature is approximately 40 °C, determined by the ambient temperature. Therefore, the ISP1761 maximum suspend current will be below 300 µA. An alternative solution to achieve a very low suspend current is to completely switch off the $V_{CC(5V0)}$ power input by using an external PMOS transistor, controlled by one of the GPIO pins of the processor. This is possible because the ISP1761 can be used in hybrid mode, which allows only the $V_{CC(I/O)}$ powered on to avoid loading of the system bus.

When the ISP1761 power is always on, the time from wake-up to suspend will be approximately 100 ms.

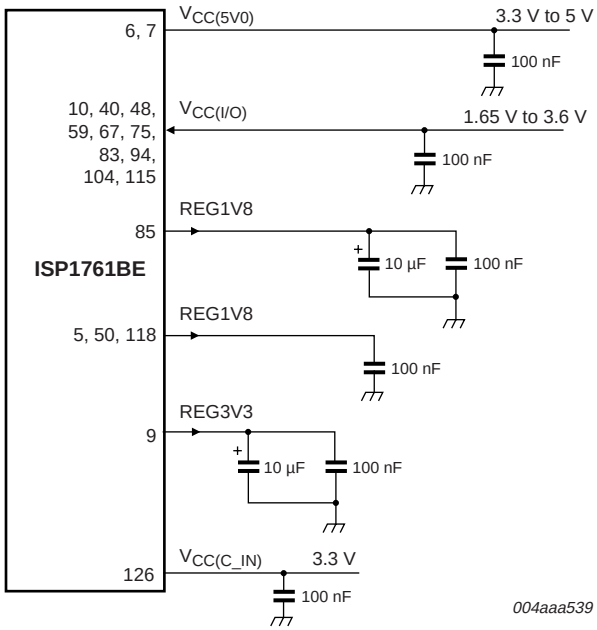
It is necessary to wait for the CLKREADY interrupt assertion before programming the ISP1761 because internal clocks are stopped during deep sleep suspend and restarted after the first wake-up event. The occurrence of the CLKREADY interrupt means that internal clocks are running and the normal functionality is achieved.

It is estimated that the CLKREADY interrupt will be generated less than 100 µs after the wake-up event, if the power to the ISP1761 was on during suspend.

If the ISP1761 is used in hybrid mode and $V_{CC(5V0)}$ is off during suspend, a 2 ms reset pulse is required when the power is switched back on, before the resume programming sequence starts. This will ensure that the internal clocks are running and all logics reach a stable initial state.

7.7 Power supply

Figure 7 shows the ISP1761 power supply connection.

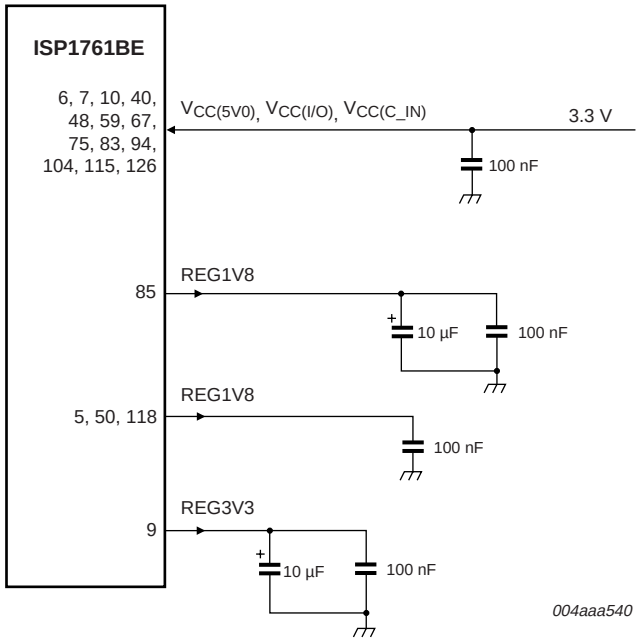


This figure shows the LQFP pinout. For the TFBGA ballout, see Table 2.

A 4.7 µF-to-10 µF electrolytic or tantalum capacitor is required on any one of the pins 5, 50 or 118. All the electrolytic or tantalum capacitors must be of LOW ESR type (0.2 Ω to 2 Ω).

Fig 7. ISP1761 power supply connection

Figure 8 shows the most commonly used power supply connection.



This figure shows the LQFP pinout. For the TFBGA ballout, see Table 2.

A 4.7 µF-to-10 µF electrolytic or tantalum capacitor is required on any one of the pins 5, 50 or 118. All the electrolytic or tantalum capacitors must be of LOW ESR type (0.2 Ω to 2 Ω).

Fig 8. Most commonly used power supply connection

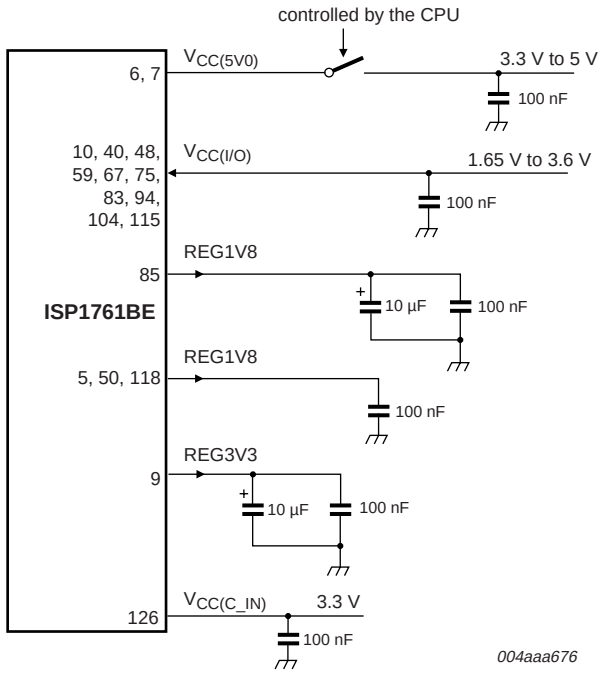
7.7.1 Hybrid mode

Table 6 shows the description of hybrid mode.

Table 6. Hybrid mode

Voltage	Status
$V_{CC(5V0)}$	off
$V_{CC(I/O)}$	on

In hybrid mode (see Figure 9), $V_{CC(5V0)}$ can be switched off using an external PMOS transistor, controlled using one of the GPIO pins of the processor. This helps to reduce the suspend current, $I_{CC(I/O)}$, below 100 µA. If the ISP1761 is used in hybrid mode and $V_{CC(5V0)}$ is off during suspend, a 2 ms reset pulse is required when power is switched back on, before the resume programming sequence starts.



This figure shows the LQFP pinout. For the TFBGA ballout, see Table 2.

A 4.7 µF-to-10 µF electrolytic or tantalum capacitor is required on any one of the pins 5, 50 or 118. All the electrolytic or tantalum capacitors must be of LOW ESR type (0.2 Ω to 2 Ω).

Fig 9. Hybrid mode

Table 7 shows the status of output pins during hybrid mode.

Table 7. Pin status during hybrid mode

Pins	$V_{CC(I/O)}$	$V_{CC(5V0)}$	Status
DATA[31:0], A[17:1], TEST, HC_IRQ, DC_IRQ, HC_DREQ, DC_DREQ, HC_DACK, DC_DACK, HC_SUSPEND/WAKEUP_N, DC_SUSPEND/WAKEUP_N	on	on	normal
	on	off	high-Z
	off	X	undefined
CS_N, RESET_N, RD_N, WR_N	on	X	input
	off	X	undefined

7.8 Overcurrent detection

The ISP1761 can implement a digital or analog overcurrent detection scheme. Bit 15 of the HW Mode Control register can be programmed to select the analog or digital overcurrent detection. An analog overcurrent detection circuit is integrated on-chip. The main features of this circuit are self reporting, automatic resetting, low-trip time and low cost. This circuit offers an easy solution at no extra hardware cost on the board. The port power will automatically be disabled by the ISP1761 on an overcurrent event occurrence, by de-asserting the PSWn_N signal without any software intervention.

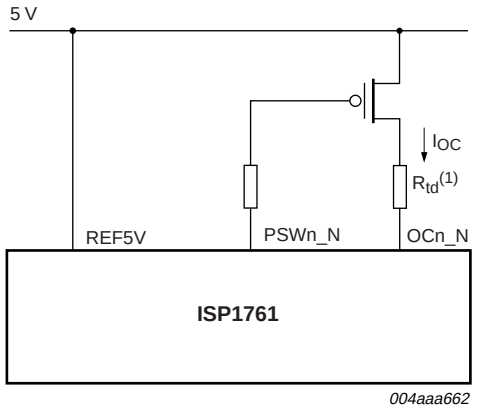
When using the integrated analog overcurrent detection, the range of the overcurrent detection voltage for the ISP1761 is 45 mV to 100 mV. Calculation of the external components must be based on the 45 mV value, with the actual overcurrent detection threshold usually positioned in the middle of the interval.

For an overcurrent limit of 500 mA per port, a PMOS transistor with R_{DSON} of approximately 100 mΩ is required. If a PMOS transistor with a lower R_{DSON} is used, the analog overcurrent detection can be adjusted using a series resistor; see Figure 10.

$\Delta V_{PMOS} = \Delta V_{OC(TRIP)} = \Delta V_{TRIP(intrinsic)} - (I_{OC(nom)} \times R_{td})$, where:

ΔV_{PMOS} = voltage drop on PMOS

$I_{OC(nom)}$ = 1 µA



(1) R_{td} is optional.

Fig 10. Adjusting analog overcurrent detection limit (optional)

The digital overcurrent scheme requires using an external power switch with integrated overcurrent detection, such as LM3526, MIC2526 (2 ports) or LM3544 (4 ports). These devices are controlled by PSWn_N signals corresponding to each port. In the case of overcurrent occurrence, these devices will assert OCn_N signals. On OCn_N assertion, the ISP1761 cuts off the port power by de-asserting PSWn_N. The external integrated power switch will also automatically cut off the port power in the case of an overcurrent event, by implementing a thermal shutdown. An internal delay filter will prevent false overcurrent reporting because of in-rush currents when plugging a USB device. Because of this internal delay, as soon as OCn_N is asserted, PSWn_N will switch off the external PMOS in less than 15 ms.

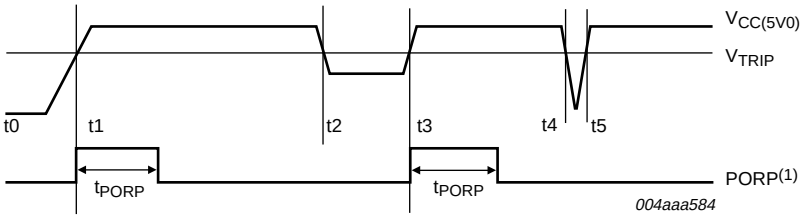
Remark: If port 1 is used in OTG mode or as a dual-role device, the analog overcurrent detection must be used, same on all three ports, because the same bit (bit 15 of the HW Mode Control register) determines the overcurrent detection type.

7.9 Power-On Reset (POR)

When $V_{CC(I/O)}$ is directly connected to the RESET_N pin, the internal POR pulse width, t_{PORP}, will typically be 800 ns. The pulse is started when $V_{CC(5V0)}$ rises above V_{TRIP} of 1.2 V.

To give a better view of the functionality, Figure 11 shows a possible curve of $V_{CC(5V0)}$ with dips at t2 to t3 and t4 to t5. If the dip at t4 to t5 is too short, that is, < 11 µs, the internal POR pulse will not react and will remain LOW. The internal POR starts with a 1 at t0. At t1, the detector will see the passing of the trip level and a delay element will add another t_{PORP} before it drops to 0.

The internal POR pulse will be generated whenever $V_{CC(5V0)}$ drops below V_{TRIP} for more than 11 µs.

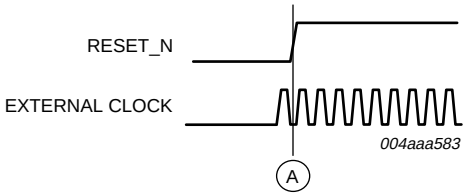


(1) PORP = Power-On Reset Pulse.

Fig 11. Internal power-on reset timing

The recommended RESET input pulse length at power-on must be at least 2 ms to ensure that internal clocks are stable.

The RESET_N pin can be either connected to $V_{CC(I/O)}$, using the internal POR circuit or externally controlled by the microcontroller, ASIC, and so on. Figure 12 shows the availability of the clock with respect to the external POR.



Stable external clock is available at A.

Fig 12. Clock with respect to the external power-on reset

8. Host controller

Table 8 shows the bit description of the registers.

- All registers range from 0000h to 03FFh. These registers can be read or written as double word, that is 32-bit data.
- Operational registers range from 0000h to 01FFh. Host controller-specific and OTG controller-specific registers range from 0300h to 03FFh. Peripheral controller-specific registers range from 0200h to 02FFh.
- 17 address lines (15/14 addresses, necessary to address up to 64 kB range on a 16-bit/32-bit data bus configuration + additional 2 addresses for bank select/virtual segmentation for memory address access time improvement). A0 is not defined because 8-bit access is not implemented.

Table 8. Host controller-specific register overview

Address	Register	Reset value	References
EHCI capability registers			
0000h	CAPLENGTH	20h	Section 8.1.1 on page 33
0002h	HCIVERSION	0100h	Section 8.1.2 on page 33
0004h	HCSPARAMS	0000 0011h	Section 8.1.3 on page 33
0008h	HCCPARAMS	0000 0086h	Section 8.1.4 on page 34
EHCI operational registers			
0020h	USBCMD	0008 0B00h	Section 8.2.1 on page 35
0024h	USBSTS	0000 0000h	Section 8.2.2 on page 36
0028h	USBINTR	0000 0000h	Section 8.2.3 on page 37
002Ch	FRINDEX	0000 0000h	Section 8.2.4 on page 37
0060h	CONFIGFLAG	0000 0000h	Section 8.2.5 on page 38
0064h	PORTSC1	0000 2000h	Section 8.2.6 on page 39
0130h	ISO PTD Done Map	0000 0000h	Section 8.2.7 on page 40
0134h	ISO PTD Skip Map	FFFF FFFFh	Section 8.2.8 on page 41
0138h	ISO PTD Last PTD	0000 0000h	Section 8.2.9 on page 41
0140h	INT PTD Done Map	0000 0000h	Section 8.2.10 on page 41
0144h	INT PTD Skip Map	FFFF FFFFh	Section 8.2.11 on page 41
0148h	INT PTD Last PTD	0000 0000h	Section 8.2.12 on page 42
0150h	ATL PTD Done Map	0000 0000h	Section 8.2.13 on page 42
0154h	ATL PTD Skip Map	FFFF FFFFh	Section 8.2.14 on page 42
0158h	ATL PTD Last PTD	0000 0000h	Section 8.2.15 on page 43
Configuration registers			
0300h	HW Mode Control	0000 0100h	Section 8.3.1 on page 43
0304h	HcChipID	0001 1761h	Section 8.3.2 on page 45
0308h	HcScratch	0000 0000h	Section 8.3.3 on page 45
030Ch	SW Reset	0000 0000h	Section 8.3.4 on page 45
0330h	HcDMAConfiguration	0000 0000h	Section 8.3.5 on page 46
0334h	HcBufferStatus	0000 0000h	Section 8.3.6 on page 47
0338h	ATL Done Timeout	0000 0000h	Section 8.3.7 on page 48

Table 8. Host controller-specific register overview …continued

Address	Register	Reset value	References
033Ch	Memory	0000 0000h	Section 8.3.8 on page 48
0340h	Edge Interrupt Count	0000 000Fh	Section 8.3.9 on page 49
0344h	DMA Start address	0000 0000h	Section 8.3.10 on page 50
0354h	Power Down Control	03E8 1BA0h	Section 8.3.11 on page 51
Interrupt registers			
0310h	HcInterrupt	0000 0000h	Section 8.4.1 on page 53
0314h	HcInterruptEnable	0000 0000h	Section 8.4.2 on page 55
0318h	ISO IRQ Mask OR	0000 0000h	Section 8.4.3 on page 57
031Ch	INT IRQ Mask OR	0000 0000h	Section 8.4.4 on page 57
0320h	ATL IRQ Mask OR	0000 0000h	Section 8.4.5 on page 57
0324h	ISO IRQ Mask AND	0000 0000h	Section 8.4.6 on page 58
0328h	INT IRQ Mask AND	0000 0000h	Section 8.4.7 on page 58
032Ch	ATL IRQ Mask AND	0000 0000h	Section 8.4.8 on page 58

8.1 EHCI capability registers

8.1.1 CAPLENGTH register

The bit description of the Capability Length (CAPLENGTH) register is given in Table 9.

Table 9. CAPLENGTH - Capability Length register (address 0000h) bit description

Bit	Symbol	Access	Value	Description
7 to 0	CAPLENGTH [7:0]	R	20h	**Capability Length**: This is used as an offset. It is added to the register base to find the beginning of the operational register space.

8.1.2 HCIVERSION register

Table 10 shows the bit description of the Host Controller Interface Version Number (HCIVERSION) register.

Table 10. HCIVERSION - Host Controller Interface Version Number register (address 0002h) bit description

Bit	Symbol	Access	Value	Description
15 to 0	HCIVERSION [15:0]	R	0100h	**Host Controller Interface Version Number**: It contains a BCD encoding of the version number of the interface to which the host controller interface conforms.

8.1.3 HCSPARAMS register

The Host Controller Structural Parameters (HCSPARAMS) register is a set of fields that are structural parameters. The bit allocation is given in Table 11.

Table 11. HCSPARAMS - Host Controller Structural Parameters register (address 0004h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Bit	23	22	21	20	19	18	17	16
Symbol	DPN[3:0]				reserved			P_INDICATOR
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	N_CC[3:0]				N_PCC[3:0]			
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	PRR	reserved		PPC	N_PORTS[3:0]			
Reset	0	0	0	1	0	0	0	1
Access	R	R	R	R	R	R	R	R

Table 12. HCSPARAMS - Host Controller Structural Parameters register (address 0004h) bit description

Bit	Symbol	Description[1]
31 to 24	-	reserved; write logic 0
23 to 20	DPN[3:0]	**Debug Port Number**: This field identifies which of the host controller ports is the debug port.
19 to 17	-	reserved; write logic 0
16	P_INDICATOR	**Port Indicators**: This bit indicates whether the ports support port indicator control.
15 to 12	N_CC[3:0]	**Number of Companion Controller**: This field indicates the number of companion controllers associated with this Hi-Speed USB host controller.
11 to 8	N_PCC[3:0]	**Number of Ports per Companion Controller**: This field indicates the number of ports supported per companion host controller.
7	PRR	**Port Routing Rules**: This field indicates the method used to map ports to the companion controllers.
6 to 5	-	reserved; write logic 0
4	PPC	**Port Power Control**: This field indicates whether the host controller implementation includes port power control.
3 to 0	N_PORTS[3:0]	**N_Ports**: This field specifies the number of physical downstream ports implemented on this host controller.

[1] For details on register bit description, refer to Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

8.1.4 HCCPARAMS register

The Host Controller Capability Parameters (HCCPARAMS) register is a 4 bytes register, and the bit allocation is given in Table 13.

Table 13. HCCPARAMS - Host Controller Capability Parameters register (address 0008h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Bit	23	22	21	20	19	18	17	16
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	EECP[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	IST[3:0]				reserved	ASPC	PFLF	reserved
Reset	1	0	0	0	0	1	1	0
Access	R	R	R	R	R	R	R	R

Table 14. HCCPARAMS - Host Controller Capability Parameters register (address 0008h) bit description

Bit	Symbol	Description[1]
31 to 16	-	reserved; write logic 0
15 to 8	EECP[7:0]	**EHCI Extended Capabilities Pointer**: Default = implementation dependent. This optional field indicates the existence of a capabilities list.
7 to 4	IST[3:0]	**Isochronous Scheduling Threshold**: Default = implementation dependent. This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule.
3	-	reserved; write logic 0
2	ASPC	**Asynchronous Scheduling Park Capability**: Default = implementation dependent. If this bit is set to logic 1, the host controller supports the park feature for high-speed Transfer Descriptors in the asynchronous schedule.
1	PFLF	**Programmable Frame List Flag**: Default = implementation dependent. If this bit is cleared, the system software must use a frame list length of 1024 elements with this host controller. If PFLF is set, the system software can specify and use a smaller frame list and configure the host through the USBCMD register FLS field.
0	-	reserved; write logic 0

[1] For details on register bit description, refer to Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

8.2 EHCI operational registers

8.2.1 USBCMD register

The USB Command (USBCMD) register indicates the command to be executed by the serial host controller. Writing to this register causes a command to be executed. Table 15 shows the USBCMD register bit allocation.

Table 15. USBCMD - USB Command register (address 0020h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]							
Reset	0	0	0	0	1	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved[1]							
Reset	0	0	0	0	1	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	LHCR	reserved[1]					HCRESET	RS
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

[1] The reserved bits should always be written with the reset value.

Table 16. USBCMD - USB Command register (address 0020h) bit description

Bit	Symbol	Description[1]
31 to 8	-	reserved
7	LHCR	**Light Host Controller Reset** (optional): If implemented, it allows the driver software to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers. If not implemented, a read of this field will always return logic 0.
6 to 2	-	reserved
1	HCRESET	**Host Controller Reset**: This control bit is used by the software to reset the host controller.
0	RS	**Run/Stop**: 1 = Run, 0 = Stop. When set, the host controller executes the schedule.

[1] For details on register bit description, refer to Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

8.2.2 USBSTS register

The USB Status (USBSTS) register indicates pending interrupts and various states of the host controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software clears the register bits by writing ones to them. The bit allocation is given in Table 17.

Table 17. USBSTS - USB Status register (address 0024h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17	16
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved[1]							
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[1]				FLR	PCD	reserved[1]	
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

[1] The reserved bits should always be written with the reset value.

Table 18. USBSTS - USB Status register (address 0024h) bit description

Bit	Symbol	Description[1]
31 to 4	-	reserved; write logic 0
3	FLR	**Frame List Rollover**: The host controller sets this bit to logic 1 when the frame list index rolls over from its maximum value to zero.
2	PCD	**Port Change Detect**: The host controller sets this bit to logic 1 when any port, where the PO bit is cleared, has a change to a one or a FPR bit changes to a one as a result of a J-K transition detected on a suspended port.
1 to 0	-	reserved

[1] For details on register bit description, refer to Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

8.2.3 USBINTR register

The USB Interrupt (USBINTR) register is a read or write register located at 0028h. All the bits in this register are reserved.

8.2.4 FRINDEX register

The Frame Index (FRINDEX) register is used by the host controller to index into the periodic frame list. The register updates every 125 µs (once each microframe). Bits n to 3 are used to select a particular entry in the periodic frame list during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by the system software in the FLS (Frame List Size) field of the USBCMD register. This register must be written as a double word. A word-only write (16-bit mode) produces undefined results. A write to this register while the RS (Run/Stop) bit is set produces undefined results. Writes to this register also affect the SOF value. The bit allocation is given in Table 19.

Table 19. FRINDEX - Frame Index register (address: 002Ch) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	23	22	21	20	19	18	17	16
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved[1]		FRINDEX[13:8]					
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	FRINDEX[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

[1] The reserved bits should always be written with the reset value.

Table 20. FRINDEX - Frame Index register (address: 002Ch) bit description

Bit	Symbol	Description[1]
31 to 14	-	reserved
13 to 0	FRINDEX [13:0]	**Frame Index**: Bits in this register are used for the frame number in the SOF packet and as the index into the frame list. The value in this register increments at the end of each time frame. For example, microframe.

[1] For details on register bit description, refer to Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

8.2.5 CONFIGFLAG register

The bit allocation of the Configure Flag (CONFIGFLAG) register is given in Table 21.

Table 21. CONFIGFLAG - Configure Flag register (address 0060h) bit allocation

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

[1] The reserved bits should always be written with the reset value.

Table 22. CONFIGFLAG - Configure Flag register (address 0060h) bit description

Bit	Symbol	Description[1]
31 to 1	-	reserved
0	CF	**Configure Flag**: The host software sets this bit as the last action when it is configuring the host controller. This bit controls the default port-routing control logic.

[1] For details on register bit description, refer to Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

8.2.6 PORTSC1 register

The Port Status and Control (PORTSC) register (bit allocation: Table 23) is in the power well. It is reset by hardware only when the auxiliary power is initially applied or in response to a host controller reset. The initial conditions of a port are:

- No peripheral connected
- Port disabled

If the port has power control, software cannot change the state of the port until it sets port power bits. Software must not attempt to change the state of the port until the power is stable on the port (maximum delay is 20 ms from the transition).

Table 23. PORTSC1 - Port Status and Control 1 register (address 0064h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]				PTC[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
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	PIC[1:0]		PO	PP	LS[1:0]		reserved[1]	PR
Reset	0	0	1	0	0	0	0	0
Access	R	R	R/W	R/W	R/W	R/W	R/W	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	SUSP	FPR	reserved[1]			PED	ECSC	ECCS
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

[1] The reserved bits should always be written with the reset value.

Table 24. PORTSC1 - Port Status and Control 1 register (address 0064h) bit description

Bit	Symbol	Description[1]
31 to 20	-	reserved
19 to 16	PTC[3:0]	**Port Test Control**: When this field is zero, the port is not operating in test mode. A non-zero value indicates that it is operating in test mode indicated by the value.
15 to 14	PIC[1:0]	**Port Indicator Control**: Writing to this field has no effect if the P_INDICATOR bit in the HCSPARAMS register is logic 0. For a description on how these bits are implemented, refer to Ref. 1 "Universal Serial Bus Specification Rev. 2.0".[2]
13	PO	**Port Owner**: This bit unconditionally goes to logic 0 when the configured bit in the CONFIGFLAG register makes a logic 0 to logic 1 transition. This bit unconditionally goes to logic 1 whenever the configured bit is logic 0.
12	PP	**Port Power**: The function of this bit depends on the value of the PPC (Port Power Control) field in the HCSPARAMS register.
11 to 10	LS[1:0]	**Line Status**: This field reflects the current logical levels of the DP (bit 11) and DM (bit 10) signal lines.
9	-	reserved
8	PR	**Port Reset**: Logic 1 means the port is in the reset state. Logic 0 means the port is not in reset.[2]
7	SUSP	**Suspend**: Logic 1 means the port is in the suspend state. Logic 0 means the port is not suspended.[2]
6	FPR	**Force Port Resume**: Logic 1 means resume detected or driven on the port. Logic 0 means no resume (K-state) detected or driven on the port.[2]
5 to 3	-	reserved
2	PED	**Port Enabled/Disabled**: Logic 1 means enable. Logic 0 means disable.[2]
1	ECSC	**Connect Status Change**: Logic 1 means change in ECCS. Logic 0 means no change.[2]
0	ECCS	**Current Connect Status**: Logic 1 indicates a peripheral is present on the port. Logic 0 indicates no peripheral is present.[2]

[1] For details on register bit description, refer to Ref. 2 "Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0".

[2] These fields read logic 0, if the PP (Port Power) bit in register PORTSC 1 is logic 0.

8.2.7 ISO PTD Done Map register

The bit description of the register is given in Table 25.

Table 25. ISO PTD Done Map register (address 0130h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ISO_PTD_DONE_MAP[31:0]	R	0000 0000h	**ISO PTD Done Map**: Done map for each of the 32 PTDs for the ISO transfer

This register represents a direct map of the done status of the 32 PTDs. The bit corresponding to a certain PTD will be set to logic 1 as soon as that PTD execution is completed. Reading the Done Map register will clear all the bits that are set to logic 1, and the next reading will reflect the updated status of new executed PTDs.

8.2.8 ISO PTD Skip Map register

Table 26 shows the bit description of the register.

Table 26. ISO PTD Skip Map register (address 0134h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ISO_PTD_SKIP_MAP[31:0]	R/W	FFFF FFFFh	**ISO PTD Skip Map**: Skip map for each of the 32 PTDs for the ISO transfer

When a bit in the PTD Skip Map is set to logic 1 that PTD will be skipped although its V bit may be set. The information in that PTD is not processed. For example, NextPTDPointer will not affect the order of processing of PTDs. The Skip bit must not normally be set on the position indicated by NextPTDPointer.

8.2.9 ISO PTD Last PTD register

Table 27 shows the bit description of the ISO PTD Last PTD register.

Table 27. ISO PTD Last PTD register (address 0138h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ISO_PTD_LAST_PTD[31:0]	R/W	0000 0000h	**ISO PTD last PTD**: Last PTD of the 32 PTDs. **1h —** One PTD in ISO **2h —** Two PTDs in ISO **4h —** Three PTDs in ISO

Once the LastPTD bit corresponding to a PTD is set, this will be the last PTD processed (checking V = 1) in that PTD category. Subsequently, the process will restart with the first PTD of that group. This is useful to reduce the time in which all the PTDs, the respective memory space, would be checked, especially if only a few PTDs are defined. The LastPTD bit must normally be set to a higher position than any other position indicated by the NextPTDPointer from an active PTD.

8.2.10 INT PTD Done Map register

The bit description of the register is given in Table 28.

Table 28. INT PTD Done Map register (address 0140h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	INT_PTD_DONE_MAP[31:0]	R	0000 0000h	**INT PTD Done Map**: Done map for each of the 32 PTDs for the INT transfer

This register represents a direct map of the done status of the 32 PTDs. The bit corresponding to a certain PTD will be set to logic 1 as soon as that PTD execution is completed. Reading the Done Map register will clear all the bits that are set to logic 1, and the next reading will reflect the updated status of new executed PTDs.

8.2.11 INT PTD Skip Map register

Table 29 shows the bit description of the INT PTD Skip Map register.

Table 29. INT PTD Skip Map register (address 0144h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	INT_PTD_SKIP_MAP[31:0]	R/W	FFFF FFFFh	**INT PTD Skip Map**: Skip map for each of the 32 PTDs for the INT transfer

When a bit in the PTD Skip map is set to logic 1 that PTD will be skipped although its V bit may be set. The information in that PTD is not processed. For example, NextPTDPointer will not affect the order of processing of PTDs. The Skip bit should not normally be set on the position indicated by NextPTDPointer.

8.2.12 INT PTD Last PTD register

The bit description of the register is given in Table 30.

Table 30. INT PTD Last PTD register (address 0148h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	INT_PTD_LAST_PTD[31:0]	R/W	0000 0000h	**INT PTD Last PTD**: Last PTD of the 32 PTDs. **1h —** One PTD in INT **2h —** Two PTDs in INT **3h —** Three PTDs in INT

Once the LastPTD bit corresponding to a PTD is set, this will be the last PTD processed (checking V = 1) in that PTD category. Subsequently, the process will restart with the first PTD of that group. This is useful to reduce the time in which all the PTDs, the respective memory space, would be checked, especially if only a few PTDs are defined. The LastPTD bit must normally be set to a higher position than any other position indicated by the NextPTDPointer from an active PTD.

8.2.13 ATL PTD Done Map register

Table 31 shows the bit description of the ATL PTD Done Map register.

Table 31. ATL PTD Done Map register (address 0150h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ATL_PTD_DONE_MAP[31:0]	R	0000 0000h	**ATL PTD Done Map**: Done map for each of the 32 PTDs for the ATL transfer

This register represents a direct map of the done status of the 32 PTDs. The bit corresponding to a certain PTD will be set to logic 1 as soon as that PTD execution is completed. Reading the Done Map register will clear all the bits that are set to logic 1, and the next reading will reflect the updated status of new executed PTDs.

8.2.14 ATL PTD Skip Map register

The bit description of the register is given in Table 32.

Table 32. ATL PTD Skip Map register (address 0154h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ATL_PTD_SKIP_MAP[31:0]	R/W	FFFF FFFFh	**ATL PTD Skip Map**: Skip map for each of the 32 PTDs for the ATL transfer

When a bit in the PTD Skip map is set to logic 1 that PTD will be skipped although its V bit may be set. The information in that PTD is not processed. For example, NextPTDPointer will not affect the order of processing of PTDs. The Skip bit should not normally be set on the position indicated by NextPTDPointer.

8.2.15 ATL PTD Last PTD register

The bit description of the ATL PTD Last PTD register is given in Table 33.

Table 33. ATL PTD Last PTD register (address 0158h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ATL_PTD_LAST_PTD[31:0]	R/W	0000 0000h	**ATL PTD Last PTD**: Last PTD of the 32 PTDs. **1h —** One PTD in ATL **2h —** Two PTDs in ATL **4h —** Three PTDs in ATL

Once the LastPTD bit corresponding to a PTD is set, this will be the last PTD processed (checking V = 1) in that PTD category. Subsequently, the process will restart with the first PTD of that group. This is useful to reduce the time in which all the PTDs, the respective memory space, would be checked, especially if only a few PTDs are defined. The LastPTD bit must normally be set to a higher position than any other position indicated by the NextPTDPointer from an active PTD.

8.3 Configuration registers

8.3.1 HW Mode Control register

Table 34 shows the bit allocation of the register.

Table 34. HW Mode Control - Hardware Mode Control register (address 0300h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	ALL_ATX_RESET	reserved[1]						
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]							
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	ANA_DIGI_OC	reserved[1]			DEV_DMA	COMN_IRQ	COMN_DMA	DATA_BUS_WIDTH
Reset	0	0	0	0	0	0	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	reserved	DACK_POL	DREQ_POL	reserved[1]		INTR_POL	INTR_LEVEL	GLOBAL_INTR_EN
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

[1] The reserved bits should always be written with the reset value.

Table 35. HW Mode Control - Hardware Mode Control register (address 0300h) bit description

Bit	Symbol	Description
31	ALL_ATX_RESET	**All ATX Reset**: For debugging purposes (not used normally). **1 —** Enable reset, then write back logic 0 **0 —** No reset
30 to 16	-	reserved; write logic 0
15	ANA_DIGI_OC	**Analog Digital Overcurrent**: This bit selects analog or digital overcurrent detection on pins OC1_N/V_{BUS}, OC2_N and OC3_N. **0 —** Digital overcurrent **1 —** Analog overcurrent
14 to 12	-	reserved; write logic 0
11	DEV_DMA	**Device DMA**: When this bit and bit 9 are set, DC_DREQ and DC_DACK peripheral signals are selected on the HC_DREQ and HC_DACK pins.
10	COMN_INT	**Common IRQ**: When this bit is set, DC_IRQ will be generated on the HC_IRQ pin.
9	COMN_DMA	**Common DMA**: When this bit and bit 11 are set, the DC_DREQ and DC_DACK peripheral signals are routed to the HC_DREQ and HC_DACK pins.
8	DATA_BUS_WIDTH	**Data Bus Width**: **0 —** Defines a 16-bit data bus width **1 —** Sets a 32-bit data bus width **Remark:** Setting this bit will affect all the controllers on the chip: host controller, peripheral controller and OTG controller.
7	-	reserved; write logic 0
6	DACK_POL	**DACK Polarity**: **1 —** Indicates that the DACK input is active HIGH **0 —** Indicates active LOW
5	DREQ_POL	**DREQ Polarity**: **1 —** Indicates that the DREQ output is active HIGH **0 —** Indicates active LOW
4 to 3	-	reserved; write logic 0

Table 35. HW Mode Control - Hardware Mode Control register (address 0300h) bit description *...continued*

Bit	Symbol	Description
2	INTR_POL	**Interrupt Polarity**: **0 —** Active LOW **1 —** Active HIGH
1	INTR_LEVEL	**Interrupt Level**: **0 —** INT is level triggered. **1 —** INT is edge triggered. A pulse of certain width is generated.
0	GLOBAL_INTR_EN	**Global Interrupt Enable**: This bit must be set to logic 1 to enable IRQ signal assertion. **0 —** IRQ assertion disabled. IRQ will never be asserted, regardless of other settings or IRQ events. **1 —** IRQ assertion enabled. IRQ will be asserted according to the HcInterruptEnable register, and events setting and occurrence.

8.3.2 HcChipID register

Read this register to get the ID of the ISP1761. This upper word of the register contains the hardware version number and the lower word contains the chip ID. Table 36 shows the bit description of the register.

Table 36. HcChipID - Host Controller Chip Identifier register (address 0304h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	CHIPID[31:0]	R	0001 1761h	**Chip ID**: This register represents the hardware version number (0001h) and the chip ID (1761h) for the host controller.

8.3.3 HcScratch register

This register is for testing and debugging purposes only. The value read back must be the same as the value that was written. The bit description of this register is given in Table 37.

Table 37. HcScratch - Host Controller Scratch register (address 0308h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	SCRATCH[31:0]	R/W	0000 0000h	**Scratch**: For testing and debugging purposes

8.3.4 SW Reset register

Table 38 shows the bit allocation of the register.

Table 38. SW Reset - Software Reset register (address 030Ch) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]							
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[1]						RESET_HC	RESET_ALL
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

[1] The reserved bits should always be written with the reset value.

Table 39. SW Reset - Software Reset register (address 030Ch) bit description

Bit	Symbol	Description
31 to 2	-	reserved; write logic 0
1	RESET_HC	**Reset Host Controller**: Reset only host controller-specific registers (only registers with address below 300h). **0 —** No reset **1 —** Enable reset
0	RESET_ALL	**Reset All**: Reset all host controller and CPU interface registers. **0 —** No reset **1 —** Enable reset

8.3.5 HcDMAConfiguration register

The bit allocation of the HcDMAConfiguration register is given in Table 40.

Table 40. HcDMAConfiguration - Host Controller Direct Memory Access Configuration register (address 0330h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	DMA_COUNTER[23:16]							
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	DMA_COUNTER[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	DMA_COUNTER[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	reserved[1]				BURST_LEN[1:0]		ENABLE_DMA	DMA_READ_WRITE_SEL
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

[1] The reserved bits should always be written with the reset value.

Table 41. HcDMAConfiguration - Host Controller Direct Memory Access Configuration register (address 0330h) bit description

Bit	Symbol	Description
31 to 8	DMA_COUNTER[23:0]	**DMA Counter**: The number of bytes to be transferred (read or write). **Remark:** Different number of bursts will be generated for the same transfer length programmed in 16-bit and 32-bit modes because DMA_COUNTER is in number of bytes.
7 to 4	-	reserved
3 to 2	BURST_LEN[1:0]	**DMA Burst Length**: **00 —** Single DMA burst **01 —** 4-cycle DMA burst **10 —** 8-cycle DMA burst **11 —** 16-cycle DMA burst
1	ENABLE_DMA	**Enable DMA**: **0 —** Terminate DMA **1 —** Enable DMA
0	DMA_READ_WRITE_SEL	**DMA Read or Write Select**: Indicates if the DMA operation is a write or read to or from the ISP1761. **0 —** DMA write to the ISP1761 internal RAM is set **1 —** DMA read from the ISP1761 internal RAM

8.3.6 HcBufferStatus register

The HcBufferStatus register is used to indicate the HC that a particular PTD buffer (that is, ATL, INT and ISO) contains at least one PTD that must be scheduled. Once software sets the Buffer Filled bit of a particular transfer in the HcBufferStatus register, the HC will start traversing through PTD headers that are not marked for skipping and are valid PTDs.

Remark: Software can set these bits during the initialization.

Table 42 shows the bit allocation of the HcBufferStatus register.

Table 42. HcBufferStatus - Host Controller Buffer Status register (address 0334h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]							
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[1]					ISO_BUF_FILL	INT_BUF_FILL	ATL_BUF_FILL
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

[1] The reserved bits should always be written with the reset value.

Table 43. HcBufferStatus - Host Controller Buffer Status register (address 0334h) bit description

Bit	Symbol	Description
31 to 3	-	reserved
2	ISO_BUF_FILL	**ISO Buffer Filled**: **1 —** Indicates one of the ISO PTDs is filled, and the ISO PTD area will be processed. **0 —** Indicates there is no PTD in this area. Therefore, processing of ISO PTDs will be completely skipped.
1	INT_BUF_FILL	**INT Buffer Filled**: **1 —** Indicates one of the INT PTDs is filled, and the INT PTD area will be processed. **0 —** Indicates there is no PTD in this area. Therefore, processing of INT PTDs will be completely skipped.
0	ATL_BUF_FILL	**ATL Buffer Filled**: **1 —** Indicates one of the ATL PTDs is filled, and the ATL PTD area will be processed. **0 —** Indicates there is no PTD in this area. Therefore, processing of ATL PTDs will be completely skipped.

8.3.7 ATL Done Timeout register

The bit description of the ATL Done Timeout register is given in Table 44.

Table 44. ATL Done Timeout register (address 0338h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ATL_DONE_TIMEOUT[31:0]	R/W	0000 0000h	**ATL Done Timeout**: This register determines the ATL done time-out interrupt. This register defines the time-out in milliseconds after which the ISP1761 asserts the INT line, if enabled. It is applicable to ATL done PTDs only.

8.3.8 Memory register

The Memory register contains the base memory read address and the respective bank. This register needs to be set only before a first memory read cycle. Once written, the address will be latched for the bank and will be incremented for every read of that bank until a new address for that bank is written to change the address pointer.

The bit description of the register is given in Table 45.

Table 45. Memory register (address 033Ch) bit allocation

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

[1] The reserved bits should always be written with the reset value.

Table 46. Memory register (address 033Ch) bit description

Bit	Symbol	Description
31 to 18	-	reserved
17 to 16	MEM_BANK_SEL[1:0]	**Memory Bank Select**: Up to four memory banks can be selected. For details on internal memory read description, see Section 7.3.1. Applicable to PIO mode memory read or write data transfers only.
15 to 0	START_ADDR_MEM_READ [15:0]	**Start Address for Memory Read Cycles**: The start address for a series of memory read cycles at incremental addresses in a contiguous space. Applicable to PIO mode memory read data transfers only.

8.3.9 Edge Interrupt Count register

Table 47 shows the bit allocation of the register.

Table 47. Edge Interrupt Count register (address 0340h) bit allocation

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

[1] The reserved bits should always be written with the reset value.

Table 48. Edge Interrupt Count register (address 0340h) bit description

Bit	Symbol	Description
31 to 24	MIN_WIDTH[7:0]	**Minimum Width**: Indicates the minimum width between two edge interrupts in µSOFs (1 µSOF = 125 µs). This is not valid for level interrupts. A count of zero means that interrupts occur as and when an event occurs.
23 to 16	-	reserved
15 to 0	NO_OF_CLK[15:0]	**Number of Clocks**: Count in number of clocks that the edge interrupt must be kept asserted on the interface. 16 clocks of 60 MHz on POR if this register has a value of 0000h. The default IRQ pulse width is approximately 500 ns.

8.3.10 DMA Start Address register

This register defines the start address select for the DMA read and write operations. See Table 49 for bit allocation.

Table 49. DMA Start Address register (address 0344h) bit allocation

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

[1] The reserved bits should always be written with the reset value.

Table 50. DMA Start Address register (address 0344h) bit description

Bit	Symbol	Description
31 to 16	-	reserved
15 to 0	START_ADDR_DMA[15:0]	**Start Address for DMA**: The start address for DMA read or write cycles.

8.3.11 Power Down Control register

This register is used to turn off power to internal blocks of the ISP1761 to obtain maximum power savings. Table 51 shows the bit allocation of the register.

Table 51. Power Down Control register (address 0354h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	CLK_OFF_COUNTER[15:8]							
Reset	0	0	0	0	0	0	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	CLK_OFF_COUNTER[7:0]							
Reset	1	1	1	0	1	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved[1]			PORT3_PD	PORT2_PD	VBATDET_PWR	reserved[1]	
Reset	0	0	0	1	1	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	reserved[1]		BIASEN	VREG_ON	OC3_PWR	OC2_PWR	OC1_PWR	HC_CLK_EN
Reset	1	0	1	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 52. Power Down Control register (address 0354h) bit description

Bit[1]	Symbol	Description
31 to 16	CLK_OFF_COUNTER [15:0]	**Clock Off Counter**: Determines the wake-up status duration after any wake-up event before the ISP1761 goes back into suspend mode. This time-out is applicable only if, during the given interval, the host controller is not programmed back to normal functionality. **03E8h —** The default value. It determines the default wake-up interval of 10 ms. A value of zero implies that the host controller never wakes up on any of the events. This may be useful when using the ISP1761 as a peripheral to save power by permanently programming the host controller in suspend. **FFFFh —** The maximum value. It determines a maximum wake-up time of 500 ms. The setting of this register is based on the 100 kHz ± 40 % LazyClock frequency. It is a multiple of 10 µs period. **Remark:** In 16-bit mode, the default value is 17E8h. A write operation to these bits with any value fixes the clock off counter at 1400h. This value is equivalent to a fixed wake-up time of 50 ms.
15 to 13	-	reserved
12	PORT3_PD	**Port 3 Pull-Down**: Controls port 3 pull-down resistors. **0 —** Port 3 internal pull-down resistors are not connected. **1 —** Port 3 internal pull-down resistors are connected.
11	PORT2_PD	**Port 2 Pull-Down**: Controls port 2 pull-down resistors. **0 —** Port 2 internal pull-down resistors are not connected. **1 —** Port 2 internal pull-down resistors are connected.
10	VBATDET_PWR	**V_{BAT} Detector Powered**: Controls the power to the V_{BAT} detector. **0 —** V_{BAT} detector is powered or enabled in suspend. **1 —** V_{BAT} detector is not powered or disabled in suspend.
9 to 6	-	reserved; write reset value
5	BIASEN	**Bias Circuits Powered**: Controls the power to internal bias circuits. **0 —** Internal bias circuits are not powered in suspend. **1 —** Internal bias circuits are powered in suspend.
4	VREG_ON	**V_{REG} Powered**: Enables or disables the internal 3.3 V and 1.8 V regulators when the ISP1761 is in suspend. **0 —** Internal regulators are normally powered in suspend. **1 —** Internal regulators switch to low power mode (in suspend mode).
3	OC3_PWR	**OC3_N Powered**: Controls the powering of the overcurrent detection circuitry for port 3. **0 —** Overcurrent detection is powered on or enabled during suspend. **1 —** Overcurrent detection is powered off or disabled during suspend. This may be useful when connecting a faulty device while the system is in standby.

Table 52. Power Down Control register (address 0354h) bit description *...continued*

Bit[1]	Symbol	Description
2	OC2_PWR	**OC2_N Powered**: Controls the powering of the overcurrent detection circuitry for port 2. **0 —** Overcurrent detection is powered on or enabled during suspend. **1 —** Overcurrent detection is powered off or disabled during suspend. This may be useful when connecting a faulty device while the system is in standby.
1	OC1_PWR	**OC1_N Powered**: Controls the powering of the overcurrent detection circuitry for port 1. **0 —** Overcurrent detection is powered on or enabled during suspend. **1 —** Overcurrent detection is powered off or disabled during suspend. This may be useful when connecting a faulty device while the system is in standby.
0	HC_CLK_EN	**Host Controller Clock Enabled**: Controls internal clocks during suspend. **0 —** Clocks are disabled during suspend. This is the default value. Only the LazyClock of 100 kHz ± 40 % will be left running in suspend if this bit is logic 0. If clocks are stopped during suspend, CLKREADY IRQ will be generated when all clocks are running stable. **1 —** All clocks are enabled even in suspend.

[1] For a 32-bit operation, the default wake-up counter value is 10 µs. For a 16-bit operation, the wake-up counter value is 50 ms. In the 16-bit operation, read and write back the same value on initialization.

8.4 Interrupt registers

8.4.1 HcInterrupt register

The bits of this register indicate the interrupt source, defining the events that determined the INT generation. Clearing the bits that were set because of the events listed is done by writing back logic 1 to the respective position. All bits must be reset before enabling new interrupt events. These bits will be set, regardless of the setting of bit GLOBAL_INTR_EN in the HW Mode Control register. Table 53 shows the bit allocation of the HcInterrupt register.

Table 53. HcInterrupt - Host Controller Interrupt register (address 0310h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved[1]					OTG_IRQ	ISO_IRQ	ATL_IRQ
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	INT_IRQ	CLK READY	HCSUSP	reserved[1]	DMAEOT INT	reserved[1]	SOFITLINT	reserved[1]
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

[1] The reserved bits should always be written with the reset value.

Table 54. HcInterrupt - Host Controller Interrupt register (address 0310h) bit description

Bit	Symbol	Description
31 to 11	-	reserved; write reset value
10	OTG_IRQ	**OTG_IRQ**: Indicates that an OTG event occurred. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No OTG event **1 —** OTG event occurred For details, see Section 7.4.
9	ISO_IRQ	**ISO IRQ**: Indicates that an ISO PTD was completed, or the PTDs corresponding to the bits set in the ISO IRQ Mask AND or ISO IRQ Mask OR register bits combination were completed. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No ISO PTD event occurred **1 —** ISO PTD event occurred For details, see Section 7.4.
8	ATL_IRQ	**ATL IRQ**: Indicates that an ATL PTD was completed, or the PTDs corresponding to the bits set in the ATL IRQ Mask AND or ATL IRQ Mask OR register bits combination were completed. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No ATL PTD event occurred **1 —** ATL PTD event occurred For details, see Section 7.4.
7	INT_IRQ	**INT IRQ**: Indicates that an INT PTD was completed, or the PTDs corresponding to the bits set in the INT IRQ Mask AND or INT IRQ Mask OR register bits combination were completed. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No INT PTD event occurred **1 —** INT PTD event occurred For details, see Section 7.4.
6	CLKREADY	**Clock Ready**: Indicates that internal clock signals are running stable. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No CLKREADY event has occurred **1 —** CLKREADY event occurred

Table 54. HcInterrupt - Host Controller Interrupt register (address 0310h) bit description *...continued*

Bit	Symbol	Description
5	HCSUSP	**Host Controller Suspend**: Indicates that the host controller has entered suspend mode. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** The host controller did not enter suspend mode. **1 —** The host controller entered suspend mode. If the Interrupt Service Routine (ISR) accesses the ISP1761, it will wake up for the time specified in bits 31 to 16 of the Power Down Control register.
4	-	reserved; write reset value
3	DMAEOTINT	**DMA EOT Interrupt**: Indicates the DMA transfer completion. The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No DMA transfer is completed **1 —** DMA transfer is completed
2	-	reserved; write reset value; value is zero just after reset and changes to one after a short while
1	SOFITLINT	**SOT ITL Interrupt**: The IRQ line will be asserted if the respective enable bit in the HcInterruptEnable register is set. **0 —** No SOF event has occurred **1 —** An SOF event has occurred
0	-	reserved; write reset value; value is zero just after reset and changes to one after a short while

8.4.2 HcInterruptEnable register

This register allows enabling or disabling of the IRQ generation because of various events as described in Table 55.

Table 55. HcInterruptEnable - Host Controller Interrupt Enable register (address 0314h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	reserved[1]					OTG_IRQ_E	ISO_IRQ_E	ATL_IRQ_E
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	INT_IRQ_E	CLKREADY_E	HCSUSP_E	reserved[1]	DMAEOTINT_E	reserved[1]	SOFITLINT_E	reserved[1]
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

[1] The reserved bits should always be written with the reset value.

Table 56. HcInterruptEnable - Host Controller Interrupt Enable register (address 0314h) bit description

Bit	Symbol	Description
31 to 11	-	reserved; write reset value
10	OTG_IRQ_E	**OTG_IRQ Enable**: Controls the IRQ assertion because of events present in the OTG Interrupt Latch register. **0 —** No IRQ will be asserted **1 —** IRQ will be asserted For details, see Section 7.4.
9	ISO_IRQ_E	**ISO IRQ Enable**: Controls the IRQ assertion when one or more ISO PTDs matching the ISO IRQ Mask AND or ISO IRQ Mask OR register bits combination are completed. **0 —** No IRQ will be asserted when ISO PTDs are completed **1 —** IRQ will be asserted For details, see Section 7.4.
8	ATL_IRQ_E	**ATL IRQ Enable**: Controls the IRQ assertion when one or more ATL PTDs matching the ATL IRQ Mask AND or ATL IRQ Mask OR register bits combination are completed. **0 —** No IRQ will be asserted when ATL PTDs are completed **1 —** IRQ will be asserted For details, see Section 7.4.
7	INT_IRQ_E	**INT IRQ Enable**: Controls the IRQ assertion when one or more INT PTDs matching the INT IRQ Mask AND or INT IRQ Mask OR register bits combination are completed. **0 —** No IRQ will be asserted when INT PTDs are completed **1 —** IRQ will be asserted For details, see Section 7.4.
6	CLKREADY_E	**Clock Ready Enable**: Enables the IRQ assertion when internal clock signals are running stable. Useful after wake-up. **0 —** No IRQ will be generated after a CLKREADY_E event **1 —** IRQ will be generated after a CLKREADY_E event
5	HCSUSP_E	**Host Controller Suspend Enable**: Enables the IRQ generation when the host controller enters suspend mode. **0 —** No IRQ will be generated when the host controller enters suspend mode **1 —** IRQ will be generated when the host controller enters suspend mode

Table 56. HcInterruptEnable - Host Controller Interrupt Enable register (address 0314h) bit description *...continued*

Bit	Symbol	Description
4	-	reserved; write reset value
3	DMAEOTINT_E	**DMA EOT Interrupt Enable**: Controls assertion of IRQ on the DMA transfer completion. **0 —** No IRQ will be generated when a DMA transfer is completed **1 —** IRQ will be asserted when a DMA transfer is completed
2	-	reserved; write reset value
1	SOFITLINT_E	**SOT ITL Interrupt Enable**: Controls the IRQ generation at every SOF occurrence. **0 —** No IRQ will be generated on SOF occurrence **1 —** IRQ will be asserted at every SOF
0	-	reserved; write reset value

8.4.3 ISO IRQ MASK OR register

Each bit of this register corresponds to one of the 32 ISO PTDs defined, and is a hardware IRQ mask for each PTD done map. See Table 57 for bit description. For details, see Section 7.4.

Table 57. ISO IRQ Mask OR register (address 0318h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ISO_IRQ_MASK_OR[31:0]	R/W	0000 0000h	**ISO IRQ Mask OR**: Represents a direct map for ISO PTDs 31 to 0. **0 —** No OR condition defined between ISO PTDs. **1 —** The bits corresponding to certain PTDs are set to logic 1 to define a certain OR condition.

8.4.4 INT IRQ MASK OR register

Each bit of this register (see Table 58) corresponds to one of the 32 INT PTDs defined, and is a hardware IRQ mask for each PTD done map. For details, see Section 7.4.

Table 58. INT IRQ Mask OR register (address 031Ch) bit description

Bit	Symbol	Access	Value	Description
31 to 0	INT_IRQ_MASK_OR[31:0]	R/W	0000 0000h	**INT IRQ Mask OR**: Represents a direct map for INT PTDs 31 to 0. **0 —** No OR condition defined between INT PTDs 31 to 0. **1 —** The bits corresponding to certain PTDs are set to logic 1 to define a certain OR condition.

8.4.5 ATL IRQ MASK OR register

Each bit of this register corresponds to one of the 32 ATL PTDs defined, and is a hardware IRQ mask for each PTD done map. See Table 59 for bit description. For details, see Section 7.4.

Table 59. ATL IRQ Mask OR register (address 0320h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ATL_IRQ_MASK_OR[31:0]	R/W	0000 0000h	**ATL IRQ Mask OR**: Represents a direct map for ATL PTDs 31 to 0. **0 —** No OR condition defined between ATL PTDs. **1 —** The bits corresponding to certain PTDs are set to logic 1 to define a certain OR condition.

8.4.6 ISO IRQ MASK AND register

Each bit of this register corresponds to one of the 32 ISO PTDs defined, and is a hardware IRQ mask for each PTD done map. For details, see Section 7.4.

Table 60 provides the bit description of the register.

Table 60. ISO IRQ Mask AND register (address 0324h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ISO_IRQ_MASK_AND[31:0]	R/W	0000 0000h	**ISO IRQ Mask AND**: Represents a direct map for ISO PTDs 31 to 0. **0 —** No AND condition defined between ISO PTDs. **1 —** The bits corresponding to certain PTDs are set to logic 1 to define a certain AND condition between the 32 INT PTDs.

8.4.7 INT IRQ MASK AND register

Each bit of this register (see Table 61) corresponds to one of the 32 INT PTDs defined, and is a hardware IRQ mask for each PTD done map. For details, see Section 7.4.

Table 61. INT IRQ MASK AND register (address 0328h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	INT_IRQ_MASK_AND[31:0]	R/W	0000 0000h	**INT IRQ Mask AND**: Represents a direct map for INT PTDs 31 to 0. **0 —** No OR condition defined between INT PTDs. **1 —** The bits corresponding to certain PTDs are set to logic 1 to define a certain AND condition between the 32 INT PTDs.

8.4.8 ATL IRQ MASK AND register

Each bit of this register corresponds to one of the 32 ATL PTDs defined, and is a hardware IRQ mask for each PTD done map. For details, see Section 7.4.

Table 62 shows the bit description of the register.

Table 62. ATL IRQ MASK AND register (address 032Ch) bit description

Bit	Symbol	Access	Value	Description
31 to 0	ATL_IRQ_MASK_AND [31:0]	R/W	0000 0000h	**ATL IRQ Mask AND**: Represents a direct map for ATL PTDs 31 to 0. **0 —** No OR condition defined between ATL PTDs. **1 —** The bits corresponding to certain PTDs are set to logic 1 to define a certain AND condition between the 32 ATL PTDs.

8.5 Philips Transfer Descriptor (PTD)

The standard EHCI data structures as described in Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0” are optimized for the bus master operation that is managed by the hardware state machine.

The PTD structures of the ISP1761 are translations of EHCI data structures that are optimized for the ISP1761. It, however, still follows the basic EHCI architecture. This optimized form of EHCI data structures is necessary because the ISP1761 is a slave host controller and has no bus master capability.

EHCI manages schedules in two lists: periodic and asynchronous. Data structures are designed to provide the maximum flexibility required by USB, minimize memory traffic, and reduce hardware and software complexity. The ISP1761 controller executes transactions for devices by using a simple shared-memory schedule. This schedule consists of data structures organized into three lists.

qISO — Isochronous transfer

qINTL — Interrupt transfer

qATL — Asynchronous transfer; for the control and bulk transfers

The system software maintains two lists for the host controller: periodic and asynchronous.

The ISP1761 has a maximum of 32 ISO, 32 INTL and 32 ATL PTDs. These PTDs are used as channels to transfer data from the shared memory to the USB bus. These channels are allocated and de-allocated on receiving the transfer from the core USB driver.

Multiple transfers are scheduled to the shared memory for various endpoints by traversing the next link pointer provided by endpoint data structures, until it reaches the end of the endpoint list. There are three endpoint lists: one for ISO endpoints, and the other for INTL and ATL endpoints. If the schedule is enabled, the host controller executes the ISO schedule, followed by the INTL schedule, and then the ATL schedule.

These lists are traversed and scheduled by the software according to the EHCI traversal rule. The host controller executes scheduled ISO, INTL and ATL PTDs. The completion of a transfer is indicated to the software by the interrupt that can be grouped under various PTDs by using the AND or OR registers that are available for each schedule type: ISO, INTL and ATL. These registers are simple logic registers to decide the completion status of group and individual PTDs. When the logical conditions of the Done bit is true in the shared memory, it means that PTD has completed.

There are four types of interrupts in the ISP1761: ISO, INTL, ATL and SOF. The latency can be programmed in multiples of μSOF (125 μs).

The NextPTD pointer is a feature that allows the ISP1761 to jump unused and skip PTDs. This will improve the PTD transversal latency time. The NextPTD pointer is not meant for same or single endpoint. The NextPTD works only in forward direction.

The NextPTD traversal rules defined by the ISP1761 hardware are:

1. Start the PTD memory vertical traversal, considering the skip and LastPTD information, as follows.
2. If the current PTD is active and not done, perform the transaction.
3. Follow the NextPTD pointer as specified in bits 4 to 0 of DW4.
4. If combined with LastPTD, the LastPTD setting must be at a higher address than the NextPTD specified. So both are set in a logical manner.

5. If combined with skip, the skip must not be set (logically) on the same position corresponding to NextPTD, pointed by the NextPTD pointer.
6. If PTD is set for skip, it will be neglected and the next vertical PTD will be considered.
7. If the skipped PTD already has a setting including a NextPTD pointer that will not be taken into consideration, the behavior will be just as described in the preceding step.

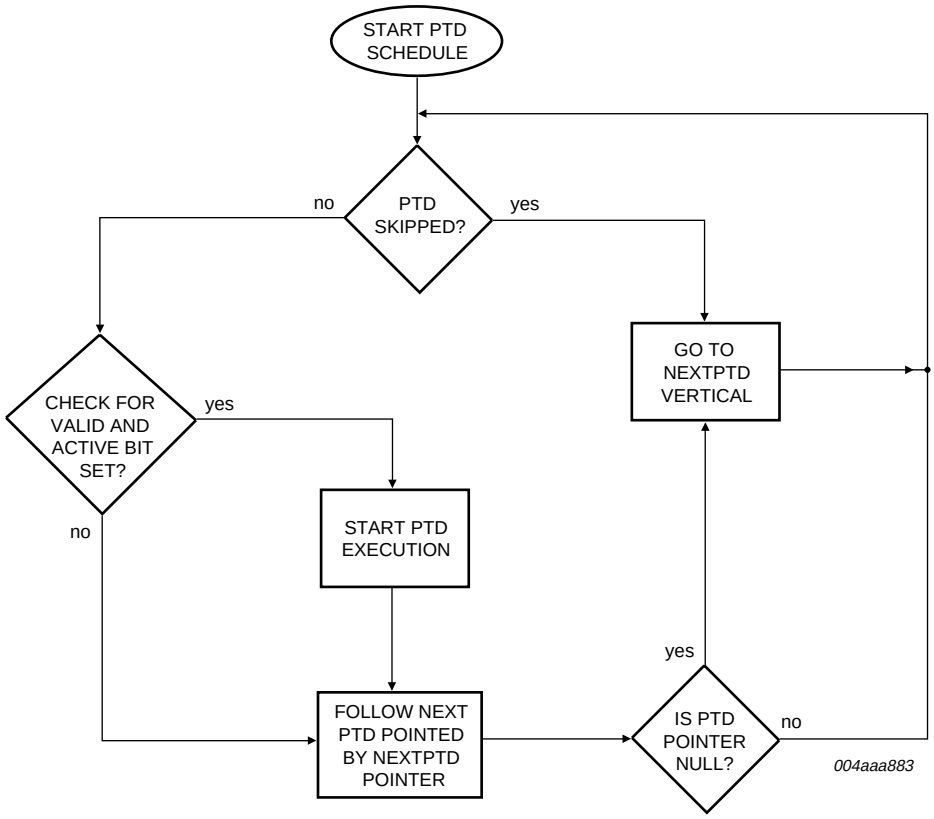


Fig 13. NextPTD traversal rule

8.5.1 High-speed bulk IN and OUT

Table 63 shows the bit allocation of the high-speed bulk IN and OUT, asynchronous Transfer Descriptor.

Table 63. High-speed bulk IN and OUT: bit allocation

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DW7	reserved																															
DW5	reserved																															
DW3	A	H	B	X	[1]	P	DT	Cerr [1:0]		NakCnt[3:0]				reserved				NrBytesTransferred[14:0] (32 kB – 1 B for high-speed)														
DW1	reserved																	S	EPType [1:0]		Token [1:0]		DeviceAddress[6:0]							EndPt[3:1]		

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW6	reserved																															
DW4	reserved																										J	NextPTDPointer[4:0]				
DW2	reserved				RL[3:0]			[1]	DataStartAddress[15:0]														reserved									
DW0	[2]	Mult [1:0]		MaxPacketLength[10:0]									NrBytesToTransfer[14:0]																	[1]		V

[1] Reserved.

[2] EndPt[0].

Table 64. High-speed bulk IN and OUT: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 32	reserved	-	-	-
DW6				
31 to 0	reserved	-	-	-
DW5				
63 to 32	reserved	-	-	-
DW4				
31 to 6	reserved	-	0	not applicable for asynchronous TD.
5	J	**SW —** writes	-	**Jump**: **0 —** To increment the PTD pointer. **1 —** To enable the next PTD branching.
4 to 0	NextPTDPointer [4:0]	**SW —** writes	-	**Next PTD Counter**: Next PTD branching assigned by the PTD pointer.
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: This bit corresponds to the Halt bit of the Status field of TD.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to the Babble Detected bit in the Status field of iTD, siTD or TD. **1 —** When babbling is detected, A and V are set to 0.
60	X	**HW —** writes	-	**Error**: This bit corresponds to the Transaction Error bit in the Status field of iTD, siTD or TD (Exec_Trans, the signal name is xacterr). **0 —** No PID error. **1 —** If there are PID errors, this bit is set active. The A and V bits are also set to inactive. This transaction is retried three times.
		SW — writes	-	**0 —** Before scheduling.
59	reserved	-	-	-
58	P	**SW —** writes **HW —** updates	-	**Ping**: For high-speed transactions, this bit corresponds to the Ping state bit in the Status field of a TD. **0 —** Ping is not set. **1 —** Ping is set. For the first time, software sets the Ping bit to 0. For the successive asynchronous TD, software sets the bit in asynchronous TD based on the state of the bit for the previous asynchronous TD of the same transfer, that is: • The current asynchronous TD is completed with the Ping bit set. • The next asynchronous TD will have its Ping bit set by the software.
57	DT	**HW —** updates **SW —** writes	-	**Data Toggle**: This bit is filled by software to start a PTD. If NrBytesToTransfer[14:0] is not complete, software needs to read this value and then write back the same value to continue.

Table 64. High-speed bulk IN and OUT: bit description *...continued*

Bit	Symbol	Access	Value	Description
56 to 55	Cerr[1:0]	**HW —** writes **SW —** writes	-	**Error Counter**: This field corresponds to the Cerr[1:0] field in TD. The default value of this field is zero for isochronous transactions. **00 —** The transaction will not retry. **11 —** The transaction will retry three times. Hardware will decrement these values.
54 to 51	NakCnt[3:0]	**HW —** writes **SW —** writes	-	**NAK Counter**: This field corresponds to the NakCnt field in TD. Software writes for the initial PTD launch. The V bit is reset if NakCnt decrements to zero and RL is a non-zero value. It reloads from RL if transaction is ACK-ed.
50 to 47	reserved	-	-	-
46 to 32	NrBytes Transferred [14:0]	**HW —** writes **SW —** writes 0000	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction. If Mult[1:0] is greater than one, it is possible to store intermediate results in this field.
DW2				
31 to 29	reserved	-	-	Set to 0 for asynchronous TD.
28 to 25	RL[3:0]	**SW —** writes	-	**Reload**: If RL is set to 0h, hardware ignores the NakCnt value. RL and NakCnt are set to the same value before a transaction.
24	reserved	-	-	Always 0 for asynchronous TD.
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the direct CPU address. RAM address = (CPU address – 400h) / 8
7 to 0	reserved	-	-	-
DW1				
63 to 47	reserved	-	-	Always 0 for asynchronous TD.
46	S	**SW —** writes	-	This bit indicates whether a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Transaction type**: **00 —** Control **10 —** Bulk
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: Identifies the token Packet Identifier (PID) for this transaction: **00 —** OUT **01 —** IN **10 —** SETUP **11 —** PING (written by hardware only).
41 to 35	DeviceAddress [6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.

Table 64. High-speed bulk IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
30 to 29	Mult[1:0]	**SW —** writes	-	**Multiplier**: This field is a multiplier used by the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. Set this field to 01b. You can also set it to 11b and 10b depending on your application. 00b is undefined.
28 to 18	MaxPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from an endpoint in a single data packet. The maximum packet size for a bulk transfer is 512 bytes. The maximum packet size for the isochronous transfer is also variable at any whole number.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB – 1 B).
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**Valid**: **0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

8.5.2 High-speed isochronous IN and OUT

Table 65 shows the bit allocation of the high-speed isochronous IN and OUT, isochronous Transfer Descriptor (iTD).

Table 65. High-speed isochronous IN and OUT: bit allocation

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DW7	ISOIN_7[11:0]												ISOIN_6[11:0]												ISOIN_5[11:4]							
DW5	ISOIN_2[7:0]								ISOIN_1[11:0]												ISOIN_0[11:0]											
DW3	A	H	B	reserved														NrBytesTransferred[14:0] (32 kB – 1 B for high-speed)														
DW1	reserved																	S	EP Type [1:0]		Token [1:0]		DeviceAddress[6:0]							EndPt[3:1]		

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW6	ISOIN_5[3:0]				ISOIN_4[11:0]												ISOIN_3[11:0]												ISOIN_2[11:8]			
DW4	Status7[2:0]			Status6[2:0]			Status5[2:0]			Status4[2:0]			Status3[2:0]			Status2[2:0]			Status1[2:0]			Status0[2:0]			µSA[7:0]							
DW2	reserved							DataStartAddress[15:0]																	µFrame[7:0]							
DW0	[2]	Mult [1:0]		MaxPacketLength[10:0]									NrBytesToTransfer[14:0]																	[1]		V

[1] Reserved.

[2] EndPt[0].

Table 66. High-speed isochronous IN and OUT: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 52	ISOIN_7[11:0]	**HW** — writes	-	Bytes received during µSOF7, if µSA[7] is set to 1 and frame number is correct.
51 to 40	ISOIN_6[11:0]	**HW** — writes	-	Bytes received during µSOF6, if µSA[6] is set to 1 and frame number is correct.
39 to 32	ISOIN_5[11:4]	**HW** — writes	-	Bytes received during µSOF5 (bits 11 to 4), if µSA[5] is set to 1 and frame number is correct.
DW6				
31 to 28	ISOIN_5[3:0]	**HW** — writes	-	Bytes received during µSOF5 (bits 3 to 0), if µSA[5] is set to 1 and frame number is correct.
27 to 16	ISOIN_4[11:0]	**HW** — writes	-	Bytes received during µSOF4, if µSA[4] is set to 1 and frame number is correct.
15 to 4	ISOIN_3[11:0]	**HW** — writes	-	Bytes received during µSOF3, if µSA[3] is set to 1 and frame number is correct.
3 to 0	ISOIN_2[11:8]	**HW** — writes	-	Bytes received during µSOF2 (bits 11 to 8), if µSA[2] is set to 1 and frame number is correct.
DW5				
63 to 56	ISOIN_2[7:0]	**HW** — writes	-	Bytes received during µSOF2 (bits 7 to 0), if µSA[2] is set to 1 and frame number is correct.
55 to 44	ISOIN_1[11:0]	**HW** — writes	-	Bytes received during µSOF1, if µSA[1] is set to 1 and frame number is correct.
43 to 32	ISOIN_0[11:0]	**HW** — writes	-	Bytes received during µSOF0, if µSA[0] is set to 1 and frame number is correct.
DW4				
31 to 29	Status7[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF7
28 to 26	Status6[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF6
25 to 23	Status5[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF5
22 to 20	Status4[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF4
19 to 17	Status3[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF3
16 to 14	Status2[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF2
13 to 11	Status1[2:0]	**HW** — writes	-	ISO IN or OUT status at µSOF1
10 to 8	Status0[2:0]	**HW** — writes	-	Status of the payload on the USB bus for this µSOF after ISO has been delivered. **Bit 0 —** Transaction error (IN and OUT) **Bit 1 —** Babble (IN token only) **Bit 2 —** Underrun (OUT token only)
7 to 0	µSA[7:0]	**SW** — writes (0 → 1) **HW** — writes (1 → 0) After processing	-	**µSOF Active**: When the frame number of bits DW1[7:3] match the frame number of the USB bus, these bits are checked for 1 before they are sent for µSOF. For example: If µSA[7:0] = 1, 1, 1, 1, 1, 1, 1, 1: send ISO every µSOF of the entire millisecond. If µSA[7:0] = 0, 1, 0, 1, 0, 1, 0, 1: send ISO only on µSOF0, µSOF2, µSOF4 and µSOF6.

Table 66. High-speed isochronous IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
DW3				
63	A	**SW —** sets	-	**Active**: This bit is the same as the Valid bit.
62	H	**HW —** writes	-	**Halt**: Only one bit for the entire millisecond. When this bit is set, the Valid bit is reset. The device decides to stall an endpoint.
61	B	**HW —** writes	-	**Babble**: Not applicable here.
60 to 47	reserved	-	0	Set to 0 for isochronous.
46 to 32	NrBytes Transferred [14:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction. If Mult[1:0] is greater than one, it is possible to store intermediate results in this field. NrBytesTransferred[14:0] is 32 kB – 1 B per PTD.
DW2				
31 to 24	reserved	-	0	Set to 0 for isochronous.
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the direct CPU address. RAM address = (CPU address – 400h) / 8
7 to 0	µFrame[7:0]	**SW —** writes	-	**Bits 2 to 0 —** Don't care **Bits 7 to 3 —** Frame number that this PTD will be sent for ISO OUT or IN
DW1				
63 to 47	reserved	-	-	-
46	S	**SW —** writes	-	This bit indicates whether a split transaction has to be executed. **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Endpoint type**: **01 —** Isochronous
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: This field indicates the token PID for this transaction: **00 —** OUT **01 —** IN
41 to 35	Device Address[6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	Mult[1:0]	**SW —** writes	-	This field is a multiplier counter used by the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. For details, refer to Appendix D of Ref. 2 "Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0".

Table 66. High-speed isochronous IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
28 to 18	MaxPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from the endpoint in a single data packet. The maximum packet size for an isochronous transfer is 1024 bytes. The maximum packet size for the isochronous transfer is also variable at any whole number.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB – 1 B).
2 to 1	reserved	-	-	-
0	V	**HW —** resets **SW —** sets	-	**0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

8.5.3 High-speed interrupt IN and OUT

Table 67 shows the bit allocation of the high-speed interrupt IN and OUT, periodic Transfer Descriptor (pTD).

Table 67. High-speed interrupt IN and OUT: bit allocation

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DW7	INT_IN_7[11:0]												INT_IN_6[11:0]												INT_IN_5[11:4]							
DW5	INT_IN_2[7:0]								INT_IN_1[11:0]												INT_IN_0[11:0]											
DW3	A	H	reserved				DT	Cerr [1:0]	reserved									NrBytesTransferred[14:0] (32 kB – 1 B for high-speed)															
DW1	reserved																	S	EP Type [1:0]		Token [1:0]		DeviceAddress[6:0]							EndPt[3:1]		

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW6	INT_IN_5[3:0]				INT_IN_4[11:0]												INT_IN_3[11:0]												INT_IN_2[11:8]			
DW4	Status7[2:0]			Status6[2:0]			Status5[2:0]			Status4[2:0]			Status3[2:0]			Status2[2:0]			Status1[2:0]			Status0[2:0]			µSA[7:0]							
DW2	reserved								DataStartAddress[15:0]																µFrame[7:0]							
DW0	[2]		Mult [1:0]						MaxPacketLength[10:0]								NrBytesToTransfer[14:0]													[1]		V

[1] Reserved.

[2] EndPt[0].

Table 68. High-speed interrupt IN and OUT: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 52	INT_IN_7[11:0]	**HW —** writes	-	Bytes received during µSOF7, if µSA[7] is set to 1 and frame number is correct.
51 to 40	INT_IN_6[11:0]	**HW —** writes	-	Bytes received during µSOF6, if µSA[6] is set to 1 and frame number is correct.
39 to 32	INT_IN_5[11:4]	**HW —** writes	-	Bytes received during µSOF5 (bits 11 to 4), if µSA[5] is set to 1 and frame number is correct.
DW6				
31 to 28	INT_IN_5[3:0]	**HW —** writes	-	Bytes received during µSOF5 (bits 3 to 0), if µSA[5] is set to 1 and frame number is correct.
27 to 16	INT_IN_4[11:0]	**HW —** writes	-	Bytes received during µSOF4, if µSA[4] is set to 1 and frame number is correct.
15 to 4	INT_IN_3[11:0]	**HW —** writes	-	Bytes received during µSOF3, if µSA[3] is set to 1 and frame number is correct.
3 to 0	INT_IN_2[11:8]	**HW —** writes	-	Bytes received during µSOF2 (bits 11 to 8), if µSA[2] is set to 1 and frame number is correct.
DW5				
63 to 56	INT_IN_2[7:0]	**HW —** writes	-	Bytes received during µSOF2 (bits 7 to 0), if µSA[2] is set to 1 and frame number is correct.
55 to 44	INT_IN_1[11:0]	**HW —** writes	-	Bytes received during µSOF1, if µSA[1] is set to 1 and frame number is correct.
43 to 32	INT_IN_0[11:0]	**HW —** writes	-	Bytes received during µSOF0, if µSA[0] is set to 1 and frame number is correct.
DW4				
31 to 29	Status7[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF7
28 to 26	Status6[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF6
25 to 23	Status5[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF5
22 to 20	Status4[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF4
19 to 17	Status3[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF3
16 to 14	Status2[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF2
13 to 11	Status1[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF1
10 to 8	Status0[2:0]	**HW —** writes	-	Status of the payload on the USB bus for this µSOF after INT has been delivered. **Bit 0 —** Transaction error (IN and OUT) **Bit 1 —** Babble (IN token only) **Bit 2 —** Underrun (OUT token only)
7 to 0	µSA[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	When the frame number of bits DW2[7:3] match the frame number of the USB bus, these bits are checked for 1 before they are sent for µSOF. For example: When µSA[7:0] = 1, 1, 1, 1, 1, 1, 1, 1: send INT for every µSOF of the entire millisecond. When µSA[7:0] = 0, 1, 0, 1, 0, 1, 0, 1: send INT for µSOF0, µSOF2, µSOF4 and µSOF6. When µSA[7:0] = 1, 0, 0, 0, 1, 0, 0, 0 = send INT for every fourth µSOF.

Table 68. High-speed interrupt IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
DW3				
63	A	**HW —** writes **SW —** writes	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: Transaction is halted.
61 to 58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: Set the Data Toggle bit to start the PTD. Software writes the current transaction toggle value. Hardware writes the next transaction toggle value.
56 to 55	Cerr[1:0]	**HW —** writes **SW —** writes	-	**Error Counter**: This field corresponds to the Cerr[1:0] field in TD. The default value of this field is zero for isochronous transactions.
54 to 47	reserved	-	-	-
46 to 32	NrBytes Transferred [14:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction. If Mult[1:0] is greater than one, it is possible to store intermediate results in this field.
DW2				
31 to 24	reserved	-	-	-
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the direct CPU address. RAM address = (CPU address – 400h) / 8
7 to 0	µFrame[7:0]	**SW —** writes	-	Bits 7 to 3 represent the polling rate in milliseconds. The INT polling rate is defined as $2^{(b-1)}$ µSOF, where b is 1 to 9. When b is 1, 2, 3 or 4, use µSA to define polling because the rate is equal to or less than 1 ms. Bits 7 to 3 are set to 0. Polling checks µSA bits for µSOF rates. See Table 69.
DW1				
63 to 47	reserved	-	-	-
46	S	**SW —** writes	-	This bit indicates if a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Endpoint type**: **11 —** Interrupt
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: This field indicates the token PID for this transaction: **00 —** OUT **01 —** IN
41 to 35	DeviceAddress [6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by the buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.

Table 68. High-speed interrupt IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
30 to 29	Mult[1:0]	**SW —** writes	-	**Multiplier**: This field is a multiplier counter used by the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. Set this field to 01b. You can also set it to 11b and 10b depending on your application. 00b is undefined.
28 to 18	MaxPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from the endpoint in a single data packet.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB – 1 B).
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**Valid**: **0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

Table 69. Microframe description

b	Rate	µFrame[7:3]	µSA[7:0]
1	1 µSOF	0 0000	1111 1111
2	2 µSOF	0 0000	1010 1010 or 0101 0101
3	4 µSOF	0 0000	any 2 bits set
4	1 ms	0 0000	any 1 bit set
5	2 ms	0 0001	any 1 bit set
6	4 ms	0 0010 to 0 0011	any 1 bit set
7	8 ms	0 0100 to 0 0111	any 1 bit set
8	16 ms	0 1000 to 0 1111	any 1 bit set
9	32 ms	1 0000 to 1 1111	any 1 bit set

8.5.4 Start and complete split for bulk

Table 70 shows the bit allocation of Start Split (SS) and Complete Split (CS) for bulk, asynchronous Start Split and Complete Split (SS/CS) Transfer Descriptor.

Table 70. Start and complete split for bulk: bit allocation

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DW7	reserved																															
DW5	reserved																															
DW3	A	H	B	X	SC	[1]	DT	Cerr [1:0]		NakCnt[3:0]				reserved				NrBytesTransferred[14:0]														
DW1	HubAddress[6:0]							PortNumber[6:0]							SE[1:0]		[1]	S	EP Type [1:0]		Token [1:0]		DeviceAddress[6:0]							EndPt[3:1]		

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DW6	reserved																															
DW4	reserved																										J	NextPTDAddress[4:0]				
DW2	reserved			RL[3:0]				[1]	DataStartAddress[15:0]									reserved														
DW0	[2]	[1]	MaxPacketLength[10:0]												NrBytesToTransfer[14:0]															[1]		V

[1] Reserved.

[2] EndPt[0].

Table 71. Start and complete split for bulk: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 32	reserved	-	-	-
DW6				
31 to 0	reserved	-	-	-
DW5				
63 to 32	reserved	-	-	-
DW4				
31 to 6	reserved	-	-	-
5	J	**SW —** writes	-	**0 —** To increment the PTD pointer. **1 —** To enable the next PTD branching.
4 to 0	NextPTDPointer[4:0]	**SW —** writes	-	**Next PTD Pointer**: Next PTD branching assigned by the PTD pointer.
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: This bit corresponds to the Halt bit of the Status field of TD.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to the Babble Detected bit in the Status field of iTD, siTD or TD. **1 —** when babbling is detected, A and V are set to 0.
60	X	**HW —** writes	-	**Transaction Error**: This bit corresponds to the Transaction Error bit in the status field.
		SW — writes	-	**0 —** Before scheduling
59	SC	**SW —** writes 0 **HW —** updates	-	**Start/Complete**: **0 —** Start split **1 —** Complete split
58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: Set the Data Toggle bit to start for the PTD.
56 to 55	Cerr[1:0]	**HW —** updates **SW —** writes	-	**Error Counter**: This field contains the error count for asynchronous start and complete split (SS/CS) TD. When an error has no response or bad response, Cerr[1:0] will be decremented to zero and then Valid will be set to zero. A NAK or NYET will reset Cerr[1:0]. For details, refer to Section 4.12.1.2 of Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”. If retry has insufficient time at the beginning of a new SOF, the first PTD must be this retry. This can be accomplished if aperiodic PTD is not advanced.
54 to 51	NakCnt[3:0]	**HW —** writes **SW —** writes	-	**NAK Counter**: The V bit is reset if NakCnt decrements to zero and RL is a non-zero value. Not applicable to isochronous split transactions.
50 to 47	reserved	-	-	-

Table 71. Start and complete split for bulk: bit description *…continued*

Bit	Symbol	Access	Value	Description
46 to 32	NrBytes Transferred[14:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction.
DW2				
31 to 29	reserved	-	-	-
28 to 25	RL[3:0]	**SW —** writes	-	**Reload**: If RL is set to 0h, hardware ignores the NakCnt value. Set RL and NakCnt to the same value before a transaction. For full-speed and low-speed transactions, set this field to 0000b. Not applicable to isochronous start split and complete split.
24	reserved	-	-	-
23 to 8	DataStartAddress [15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the direct CPU address. RAM address = (CPU address – 400h) / 8
7 to 0	reserved	-	-	-
DW1				
63 to 57	HubAddress[6:0]	**SW —** writes	-	**Hub Address**: This indicates the hub address.
56 to 50	PortNumber[6:0]	**SW —** writes	-	**Port Number**: This indicates the port number of the hub or embedded TT.
49 to 48	SE[1:0]	**SW —** writes	-	This depends on the endpoint type and direction. It is valid only for split transactions. Table 72 applies to start split and complete split only.
47	reserved	-	-	-
46	S	**SW —** writes	-	This bit indicates whether a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Endpoint Type**: **00 —** Control **10 —** Bulk
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: This field indicates the PID for this transaction. **00 —** OUT **01 —** IN **10 —** SETUP
41 to 35	DeviceAddress[6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	reserved	-	-	-

Table 71. Start and complete split for bulk: bit description *…continued*

Bit	Symbol	Access	Value	Description
28 to 18	MaximumPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from an endpoint in a single data packet. The maximum packet size for full-speed is 64 bytes as defined in Ref. 1 “Universal Serial Bus Specification Rev. 2.0”.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field.
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**Valid**: **0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

Table 72. SE description

Bulk	Control	S	E	Remarks
I/O	I/O	1	0	low-speed
I/O	I/O	0	0	full-speed

8.5.5 Start and complete split for isochronous

Table 73 shows the bit allocation for start and complete split for isochronous, split isochronous Transfer Descriptor (siTD).

Table 73. Start and complete split for isochronous: bit allocation

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DW7	reserved																								ISO_IN_7[7:0]							
DW5	ISO_IN_2[7:0]								ISO_IN_1[7:0]								ISO_IN_0[7:0]								µSCS[7:0]							
DW3	A	H	B	X	SC	[1]	DT	reserved											NrBytesTransferred[11:0]													
DW1	HubAddress[6:0]							PortNumber[6:0]							reserved			S	EP Type [1:0]		Token [1:0]		DeviceAddress[6:0]							EndPt[3:1]		
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DW6	ISO_IN_6[7:0]								ISO_IN_5[7:0]								ISO_IN_4[7:0]								ISO_IN_3[7:0]							
DW4	Status7[2:0]			Status6[2:0]			Status5[2:0]		Status4[2:0]			Status3[2:0]			Status2[2:0]			Status1[2:0]			Status0[2:0]			µSA[7:0]							
DW2	reserved								DataStartAddress[15:0]																µFrame[7:0] (full-speed)							
DW0	[2]		[1]				TT_MPS_Len[10:0]								NrBytesToTransfer[14:0] (1 kB for full-speed)										[1]						V	

[1] Reserved.

[2] EndPt[0].

Table 74. Start and complete split for isochronous: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 40	reserved	-	-	-
39 to 32	ISO_IN_7[7:0]	**HW —** writes	-	Bytes received during μSOF7, if μSA[7] is set to 1 and frame number is correct.
DW6				
31 to 24	ISO_IN_6[7:0]	**HW —** writes	-	Bytes received during μSOF6, if μSA[6] is set to 1 and frame number is correct.
23 to 16	ISO_IN_5[7:0]	**HW —** writes	-	Bytes received during μSOF5, if μSA[5] is set to 1 and frame number is correct.
15 to 8	ISO_IN_4[7:0]	**HW —** writes	-	Bytes received during μSOF4, if μSA[4] is set to 1 and frame number is correct.
7 to 0	ISO_IN_3[7:0]	**HW —** writes	-	Bytes received during μSOF3, if μSA[3] is set to 1 and frame number is correct.
DW5				
63 to 56	ISO_IN_2[7:0]	**HW —** writes	-	Bytes received during μSOF2 (bits 7 to 0), if μSA[2] is set to 1 and frame number is correct.
55 to 48	ISO_IN_1[7:0]	**HW —** writes	-	Bytes received during μSOF1, if μSA[1] is set to 1 and frame number is correct.
47 to 40	ISO_IN_0[7:0]	**HW —** writes	-	Bytes received during μSOF0 if μSA[0] is set to 1 and frame number is correct.
39 to 32	μSCS[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	All bits can be set to one for every transfer. It specifies which μSOF the complete split needs to be sent. Valid only for IN. Start split and complete split active bits, μSA = 0000 0001 and μSCS = 0000 0100, will cause SS to execute in μFrame0 and CS in μFrame2.
DW4				
31 to 29	Status7[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF7
28 to 26	Status6[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF6
25 to 23	Status5[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF5
22 to 20	Status4[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF4
19 to 17	Status3[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF3
16 to 14	Status2[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF2
13 to 11	Status1[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF1
10 to 8	Status0[2:0]	**HW —** writes	-	Isochronous IN or OUT status of μSOF0 **Bit 0 —** Transaction error (IN and OUT) **Bit 1 —** Babble (IN token only) **Bit 2 —** Underrun (OUT token only)
7 to 0	μSA[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	Specifies which μSOF the start split needs to be placed. **For OUT token**: When the frame number of bits DW2[7:3] matches the frame number of the USB bus, these bits are checked for one before they are sent for the μSOF. **For IN token**: Only μSOF0, μSOF1, μSOF2 or μSOF3 can be set to 1. Nothing can be set for μSOF4 and above.

Table 74. Start and complete split for isochronous: bit description *…continued*

Bit	Symbol	Access	Value	Description
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: The Halt bit is set when any microframe transfer status has a stalled or halted condition.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to bit 1 of Status0 to Status7 for every microframe transfer status.
60	X	**HW —** writes	-	**Transaction Error**: This bit corresponds to bit 0 of Status0 to Status7 for every microframe transfer status.
59	SC	**SW —** writes 0 **HW —** updates	-	**Start/Complete**: **0 —** Start split **1 —** Complete split
58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: Set the Data Toggle bit to start for the PTD.
56 to 44	reserved	-	-	-
43 to 32	NrBytes Transferred [11:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction.
DW2				
31 to 24	reserved	-	-	-
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the CPU address.
7 to 0	µFrame[7:0]	**SW —** writes	-	Bits 7 to 3 determine which frame to execute.
DW1				
63 to 57	HubAddress [6:0]	**SW —** writes	-	**Hub Address**: This indicates the hub address.
56 to 50	PortNumber [6:0]	**SW —** writes	-	**Port Number**: This indicates the port number of the hub or embedded TT.
49 to 47	reserved	-	-	-
46	S	**SW —** writes	-	**Split**: This bit indicates whether a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Transaction type**: **01 —** Isochronous
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: Token PID for this transaction: **00 —** OUT **01 —** IN

Table 74. Start and complete split for isochronous: bit description *…continued*

Bit	Symbol	Access	Value	Description
41 to 35	Device Address[6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	reserved	-	-	-
28 to 18	TT_MPS_Len [10:0]	**SW —** writes	-	**Transaction Translator Maximum Packet Size Length**: This field indicates the maximum number of bytes that can be sent per start split depending on the number of total bytes needed. If the total bytes to be sent for the entire millisecond is greater than 188 bytes, this field should be set to 188 bytes for an OUT token and 192 bytes for an IN token. Otherwise, this field should be equal to the total bytes sent.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field. This field is restricted to 1023 bytes because in siTD the maximum allowable payload for a full-speed device is 1023 bytes. This field indirectly becomes the maximum packet size of the downstream device.
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

8.5.6 Start and complete split for interrupt

Table 75 shows the bit allocation of start and complete split for interrupt.

Table 75. Start and complete split for interrupt: bit allocation

Bit	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DW7	reserved																								INT_IN_7[7:0]							
DW5	INT_IN_2[7:0]								INT_IN_1[7:0]								INT_IN_0[7:0]								µSCS[7:0]							
DW3	A	H	B	X	SC	[1]	DT	Cerr [1:0]	reserved												NrBytesTransferred[11:0] (4 kB for full-speed and low-speed)											
DW1	HubAddress[6:0]`								PortNumber[6:0]						SE[1:0]		-	S	EP Type [1:0]		Token [1:0]		DeviceAddress[6:0]							EndPt[3:1]		
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DW6	INT_IN_6[7:0]								INT_IN_5[7:0]								INT_IN_4[7:0]								INT_IN_3[7:0]							
DW4	Status7[2:0]			Status6[2:0]			Status5[2:0]			Status4[2:0]			Status3[2:0]			Status2[2:0]			Status1[2:0]			Status0[2:0]			µSA[7:0]						
DW2	reserved								DataStartAddress[15:0]																µFrame[7:0] (full-speed and low-speed)							
DW0	[2]		[1]					MaxPacketLength[10:0]							NrBytesToTransfer[14:0] (4 kB for full-speed and low-speed)															[1]		V

[1] Reserved.

[2] EndPt[0].

Table 76. Start and complete split for interrupt: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 40	reserved	-	-	-
39 to 32	INT_IN_7[7:0]	**HW** — writes	-	Bytes received during μSOF7, if μSA[7] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
DW6				
31 to 24	INT_IN_6[7:0]	**HW** — writes	-	Bytes received during μSOF6, if μSA[6] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
23 to 16	INT_IN_5[7:0]	**HW** — writes	-	Bytes received during μSOF5, if μSA[5] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
15 to 8	INT_IN_4[7:0]	**HW** — writes	-	Bytes received during μSOF4, if μSA[4] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
7 to 0	INT_IN_3[7:0]	**HW** — writes	-	Bytes received during μSOF3, if μSA[3] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
DW5				
63 to 56	INT_IN_2[7:0]	**HW** — writes	-	Bytes received during μSOF2 (bits 7 to 0), if μSA[2] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
55 to 48	INT_IN_1[7:0]	**HW** — writes	-	Bytes received during μSOF1, if μSA[1] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
47 to 40	INT_IN_0[7:0]	**HW** — writes	-	Bytes received during μSOF0 if μSA[0] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
39 to 32	μSCS[7:0]	**SW** — writes ($0 \rightarrow 1$) **HW** — writes ($1 \rightarrow 0$) After processing	-	All bits can be set to one for every transfer. It specifies which μSOF the complete split needs to be sent. Valid only for IN. Start split and complete split active bits, μSA = 0000 0001 and μSCS = 0000 0100, will cause SS to execute in μFrame0 and CS in μFrame2.
DW4				
31 to 29	Status7[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF7
28 to 26	Status6[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF6
25 to 23	Status5[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF5
22 to 20	Status4[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF4
19 to 17	Status3[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF3
16 to 14	Status2[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF2
13 to 11	Status1[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF1
10 to 8	Status0[2:0]	**HW** — writes	-	Interrupt IN or OUT status of μSOF0 **Bit 0** — Transaction error (IN and OUT) **Bit 1** — Babble (IN token only) **Bit 2** — Underrun (OUT token only)

Table 76. Start and complete split for interrupt: bit description *...continued*

Bit	Symbol	Access	Value	Description
7 to 0	µSA[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	Specifies which µSOF the start split needs to be placed. **For OUT token**: When the frame number of bits DW1[7:3] matches the frame number of the USB bus, these bits are checked for one before they are sent for the µSOF. **For IN token**: Only µSOF0, µSOF1, µSOF2 or µSOF3 can be set to 1. Nothing can be set for µSOF4 and above.
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: The Halt bit is set when any microframe transfer status has a stalled or halted condition.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to bit 1 of Status0 to Status7 for every microframe transfer status.
60	X	**HW —** writes	-	**Transaction Error**: This bit corresponds to bit 0 of Status0 to Status7 for every microframe transfer status.
59	SC	**SW —** writes 0 **HW —** updates	-	**Start/Complete**: **0 —** Start split **1 —** Complete split
58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: For an interrupt transfer, set correct bit to start the PTD.
56 to 55	Cerr[1:0]	**HW —** writes **SW —** writes	-	**Error Counter**: This field corresponds to the Cerr[1:0] field in TD. **00 —** The transaction will not retry. **11 —** The transaction will retry three times. Hardware will decrement these values.
54 to 44	reserved	-	-	-
43 to 32	NrBytes Transferred [11:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction.
DW2				
31 to 24	reserved	-	-	-
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the CPU address.
7 to 0	µFrame[7:0]	**SW —** writes	-	Bits 7 to 3 is the polling rate in milliseconds. Polling rate is defined as $2^{(b-1)}$ µSOF; where b = 4 to 16. When b is 4, executed every millisecond. See Table 77.
DW1				
63 to 57	HubAddress [6:0]	**SW —** writes	-	**Hub Address**: This indicates the hub address.
56 to 50	PortNumber [6:0]	**SW —** writes	-	**Port Number**: This indicates the port number of the hub or embedded TT.
49 to 48	SE[1:0]	**SW —** writes	-	This depends on the endpoint type and direction. It is valid only for split transactions. Table 78 applies to start split and complete split only.

Table 76. Start and complete split for interrupt: bit description *...continued*

Bit	Symbol	Access	Value	Description
47	reserved	-	-	-
46	S	**SW** — writes	-	This bit indicates whether a split transaction has to be executed: **0** — High-speed transaction **1** — Split transaction
45 to 44	EPType[1:0]	**SW** — writes	-	Transaction type: **11** — Interrupt
43 to 42	Token[1:0]	**SW** — writes	-	Token PID for this transaction: **00** — OUT **01** — IN
41 to 35	DeviceAddress [6:0]	**SW** — writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW** — writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
DW0				
31	EndPt[0]	**SW** — writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	reserved	-	-	-
28 to 18	MaxPacket Length[10:0]	**SW** — writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from an endpoint in a single data packet. The maximum packet size for the full-speed and low-speed devices is 64 bytes as defined in Ref. 1 “Universal Serial Bus Specification Rev. 2.0”.
17 to 3	NrBytesTo Transfer[14:0]	**SW** — writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field. The maximum total number of bytes for this transaction is 4 kB.
2 to 1	reserved	-	-	-
0	V	**SW** — sets **HW** — resets	-	**0** — This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1** — Software updates to one when there is payload to be sent or received. The current PTD is active.

Table 77. Microframe description

b	Rate	μFrame[7:3]
5	2 ms	0 0001
6	4 ms	0 0010 or 0 0011
7	8 ms	0 0100 or 0 0111
8	16 ms	0 1000 or 0 1111
9	32 ms	1 0000 or 1 1111

Table 78. SE description

Interrupt	S	E	Remarks
I/O	1	0	low-speed
I/O	0	0	full-speed

9. OTG controller

9.1 Introduction

OTG is a supplement to the Hi-Speed USB specification that augments existing USB peripherals by adding to these peripherals limited host capability to support other targeted USB peripherals. It is primarily targeted at portable devices because it addresses concerns related to such devices, such as a small connector and low power. Non-portable devices, even standard hosts, can also benefit from OTG features.

The ISP1761 OTG controller is designed to perform all the tasks specified in the OTG supplement. It supports Host Negotiation Protocol (HNP) and Session Request Protocol (SRP) for dual-role devices. The ISP1761 uses software implementation of HNP and SRP for maximum flexibility. A set of OTG registers provides the control and status monitoring capabilities to support software HNP and SRP.

Besides the normal USB transceiver, timers and analog components required by OTG are also integrated on-chip. The analog components include:

- Built-in 3.3 V-to-5 V charge pump
- Voltage comparators
- Pull-up or pull-down resistors on data lines
- Charging or discharging resistors for V_{BUS}

9.2 Dual-role device

When port 1 of the ISP1761 is configured in OTG mode, it can be used as an OTG dual-role device. A dual-role device is a USB device that can function either as a host or as a peripheral.

The default role of the ISP1761 is controlled by the ID pin, which in turn is controlled by the type of plug connected to the micro-AB receptacle. If ID = LOW (micro-A plug connected), it becomes an A-device, which is a host by default. If ID = HIGH (micro-B plug connected), it becomes a B-device, which is a peripheral by default.

Both the A-device and the B-device work on a session base. A session is defined as the period of time in which devices exchange data. A session starts when V_{BUS} is driven and ends when V_{BUS} is turned off. Both the A-device and the B-device may start a session. During a session, the role of the host can be transferred back and forth between the A-device and the B-device any number of times by using HNP.

If the A-device wants to start a session, it turns on V_{BUS} by enabling the charge pump. The B-device detects that V_{BUS} has risen above the B_SESS_VLD level and assumes the role of a peripheral asserting its pull-up resistor on the DP line. The A-device detects the remote pull-up resistor and assumes the role of a host. Then, the A-device can communicate with the B-device as long as it wishes. When the A-device finishes communicating with the B-device, the A-device turns off V_{BUS} and both the devices finally go into the idle state. See Figure 15 and Figure 16.

If the B-device wants to start a session, it must initiate SRP by ‘data line pulsing’ and ‘V_{BUS} pulsing’. When the A-device detects any of these SRP events, it turns on its V_{BUS}. (Note: only the A-device is allowed to drive V_{BUS}.) The B-device assumes the role of a

peripheral, and the A-device assumes the role of a host. The A-device detects that the B-device can support HNP by getting the OTG descriptor from the B-device. The A-device will then enable the HNP hand-off by using SetFeature (b_hnp_enable) and then go into the suspend state. The B-device signals claiming the host role by de-asserting its pull-up resistor. The A-device acknowledges by going into the peripheral state. The B-device then assumes the role of a host and communicates with the A-device as long as it wishes. When the B-device finishes communicating with the A-device, both the devices finally go into the idle state. See Figure 15 and Figure 16.

9.3 Session Request Protocol (SRP)

As a dual-role device, the ISP1761 can initiate and respond to SRP. The B-device initiates SRP by data line pulsing, followed by V_{BUS} pulsing. The A-device can detect either data line pulsing or V_{BUS} pulsing.

9.3.1 B-device initiating SRP

The ISP1761 can initiate SRP by performing the following steps:

1. Detect initial conditions [read B_SESS_END and B_SE0_SRP (bits 7 and 8) of the OTG Status register].
2. Start data line pulsing [set DP_PULLUP (bit 0) of the OTG Control (set) register to logic 1].
3. Wait for 5 ms to 10 ms.
4. Stop data line pulsing [set DP_PULLUP (bit 0) of the OTG Control (clear) register to logic 0].
5. Start V_{BUS} pulsing [set VBUS_CHRG (bit 6) of the OTG Control (set) register to logic 1].
6. Wait for 10 ms to 20 ms.
7. Stop V_{BUS} pulsing [set VBUS_CHRG (bit 6) of the OTG Control (clear) register to logic 0].
8. Discharge V_{BUS} for about 30 ms [by using VBUS_DISCHRG (bit 5) of the OTG Control (set) register], optional.

The B-device must complete both data line pulsing and V_{BUS} pulsing within 100 ms.

9.3.2 A-device responding to SRP

The A-device must be able to respond to one of the two SRP events: data line pulsing or V_{BUS} pulsing. When data line pulsing is used, the ISP1761 can detect DP pulsing. This means that the peripheral-only device must initiate data line pulsing through DP. A dual-role device will always initiate data line pulsing through DP.

To enable the SRP detection through the V_{BUS} pulsing, set A_B_SESS_VLD (bit 1) in the OTG Interrupt Enable Fall and OTG Interrupt Enable Rise registers.

To enable the SRP detection through the DP pulsing, set DP_SRP (bit 2) in the OTG Interrupt Enable Rise register.

9.4 Host Negotiation Protocol (HNP)

HNP is used to transfer control of the host role between the default host (A-device) and the default peripheral (B-device) during a session. When the A-device is ready to give up its role as a host, it will condition the B-device using SetFeature (b_hnp_enable) and will go into suspend. If the B-device wants to use the bus at that time, it signals a disconnect to the A-device. Then, the A-device will take the role of a peripheral and the B-device will take the role of a host.

9.4.1 Sequence of HNP events

The sequence of events for HNP as observed on the USB bus is illustrated in Figure 14.

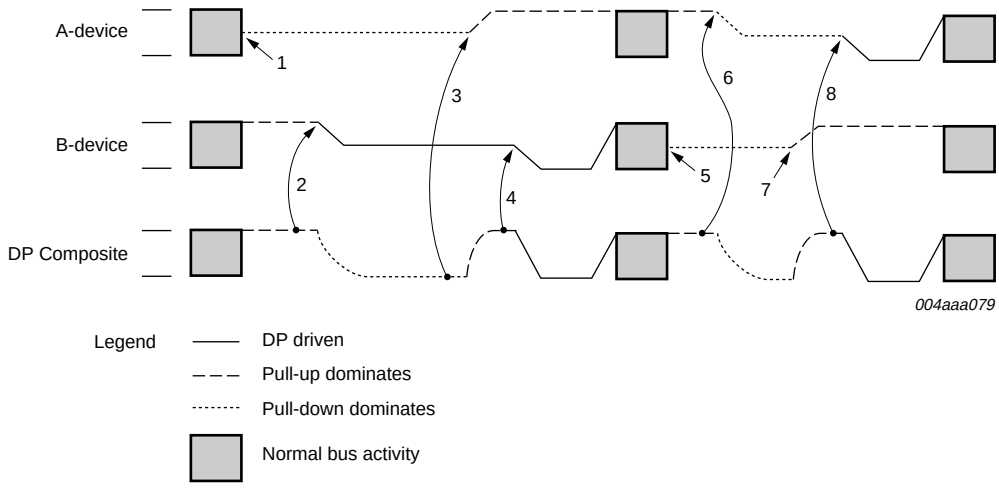


Fig 14. HNP sequence of events

As can be seen in Figure 14:

1. The A-device completes using the bus and stops all bus activity, that is, suspends the bus.
2. The B-device detects that the bus is idle for more than 5 ms and begins HNP by turning off the pull-up on DP. This allows the bus to discharge to the SE0 state.
3. The A-device detects SE0 on the bus and recognizes this as a request from the B-device to become a host. The A-device responds by turning on its DP pull-up within 3 ms of first detecting SE0 on the bus.
4. After waiting for 30 µs to ensure that the DP line is not HIGH because of the residual effect of the B-device pull-up, the B-device notices that the DP line is HIGH and the DM line is LOW, that is, J state. This indicates that the A-device has recognized the HNP request from the B-device. At this point, the B-device becomes a host and asserts bus reset to start using the bus. The B-device must assert the bus reset, that is, SE0, within 1 ms of the time that the A-device turns on its pull-up.
5. When the B-device completes using the bus, it stops all bus activities. Optionally, the B-device may turn on its DP pull-up at this time.

   Remark: The bus idle state will generate a DC suspend interrupt corresponding to the toggle of the SUSP bit in the DcInterrupt register (address: 218h), when accordingly enabled.

6. The A-device detects lack of bus activity for more than 3 ms and turns off its DP pull-up. Alternatively, if the A-device has no further need to communicate with the B-device, the A-device may turn off V_{BUS} and end the session.
7. The B-device turns on its pull-up.
8. After waiting 30 μs to ensure that the DP line is not HIGH because of the residual effect of the A-device pull-up, the A-device notices that the DP-line is HIGH and the DM line is LOW, indicating that the B-device is signaling a connect and is ready to respond as a peripheral. At this point, the A-device becomes a host and asserts the bus reset to start using the bus.

9.4.2 OTG state diagrams

Figure 15 and Figure 16 show state diagrams for the dual-role A-device and the dual-role B-device, respectively. For a detailed explanation, refer to Ref. 3 “On-The-Go Supplement to the USB Specification Rev. 1.3”.

The OTG state machine is implemented with software. The inputs to the state machine come from four sources: hardware signals from the USB bus, software signals from the application program, internal variables with the state machines, and timers:

- Hardware inputs: Include id, a_vbus_vld, a_sess_vld, b_sess_vld, b_sess_end, a_conn, b_conn, a_bus_suspend, b_bus_suspend, a_bus_resume, b_bus_resume, a_srp_det and b_se0_srp. All these inputs can be derived from the OTG Interrupt and OTG Status registers.
- Software inputs: Include a_bus_req, a_bus_drop and b_bus_req.
- Internal variables: Include a_set_b_hnp_en, b_hnp_enable and b_srp_done.
- Timers: The HNP state machine uses four timers: a_wait_vrise_tmr, a_wait_bcon_tmr, a_aidl_bdis_tmr and b_ase0_brst, tmr. All timers are started on entry to and reset on exit from their associated states. The ISP1761 provides a programmable timer that can be used as any of these four timers.

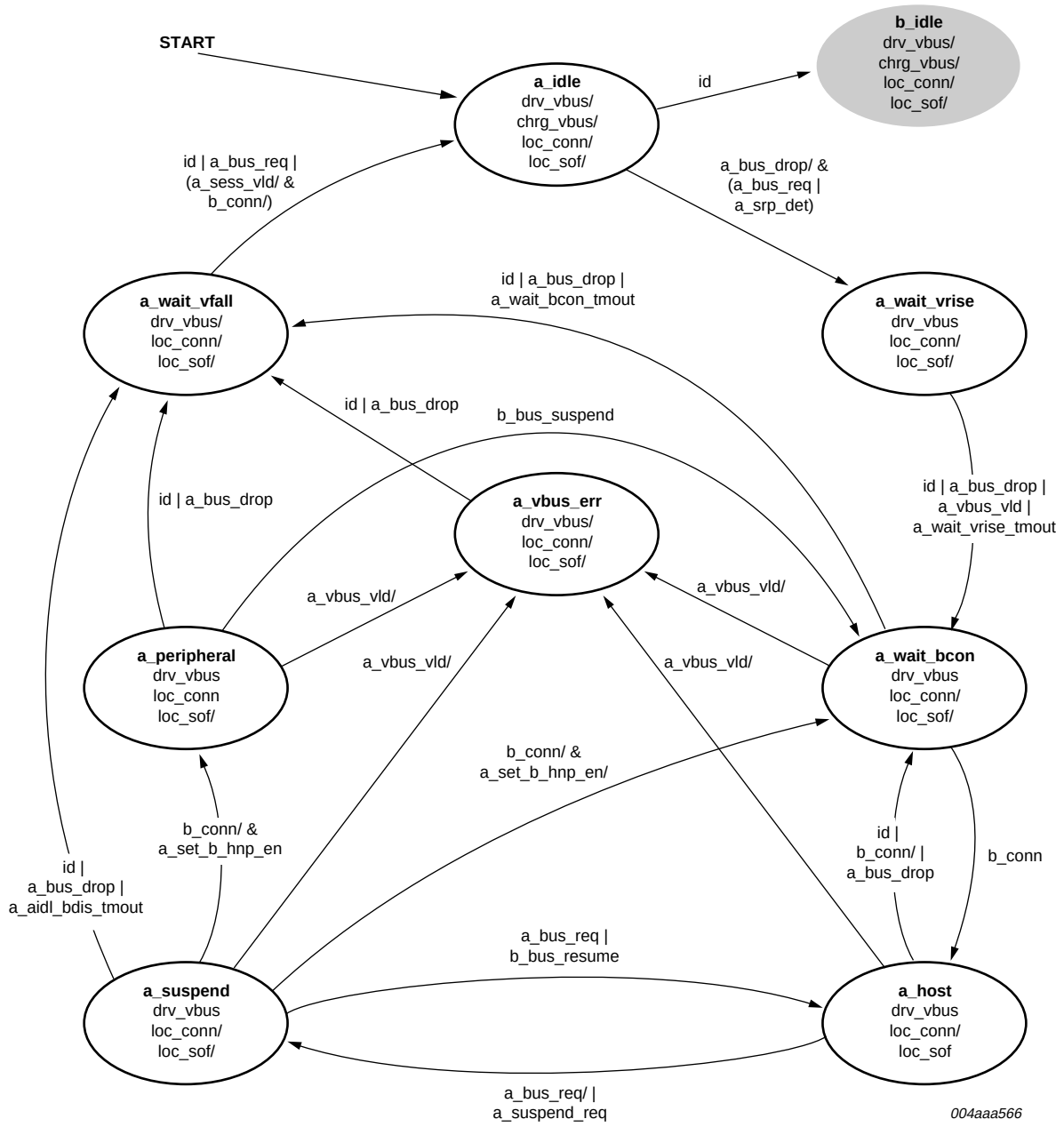


Fig 15. Dual-role A-device state diagram

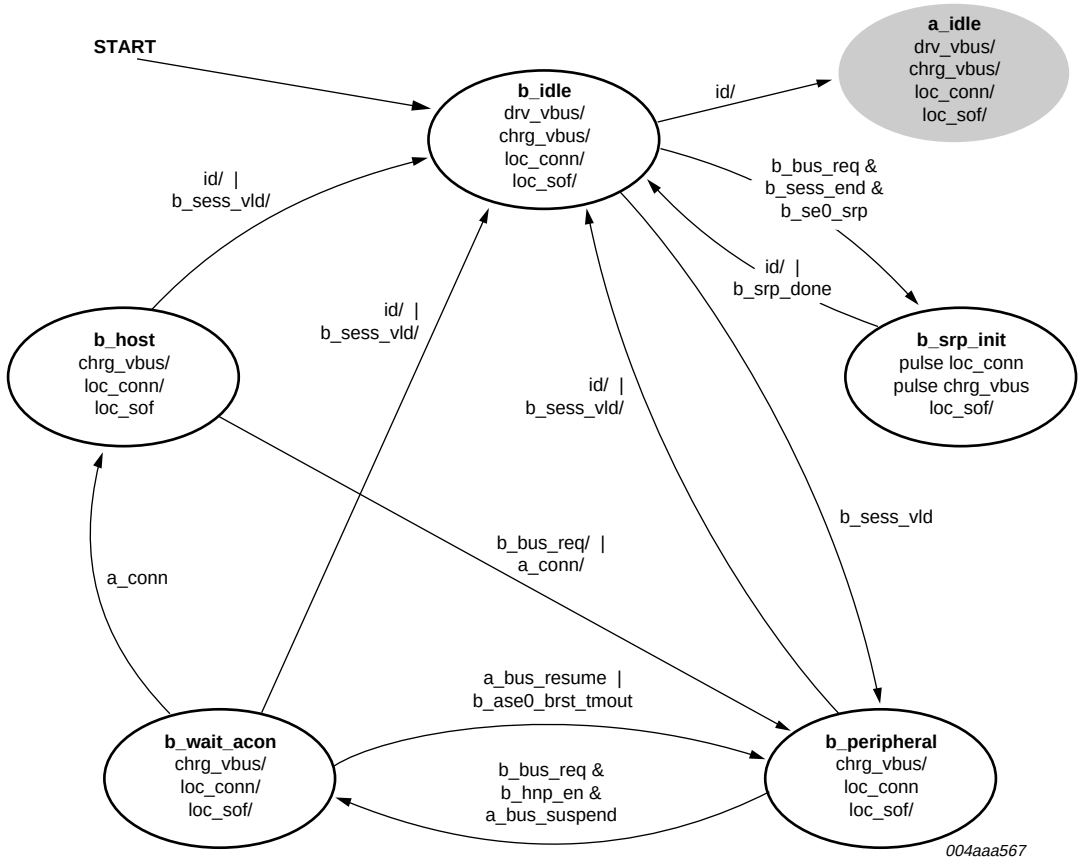


Fig 16. Dual-role B-device state diagram

9.4.3 HNP implementation and OTG state machine

The OTG state machine is the software behind all the OTG functionality. It is implemented in the microprocessor system that is connected to the ISP1761. The ISP1761 provides registers for all input status, the output control and timers to fully support the state machine transitions in Figure 15 and Figure 16. These registers include:

- OTG Control register: Provides control to V_{BUS} driving, charging or discharging, data line pull-up or pull-down, SRP detection, and so on.
- OTG Status register: Provides status detection on V_{BUS} and data lines including ID, V_{BUS} session valid, session end, overcurrent and bus status.
- OTG Interrupt Latch register: Provides interrupts for status change in OTG Interrupt Status register bits and the OTG Timer time-out event.
- OTG Interrupt Enable Fall and OTG Interrupt Enable Rise registers: Provide interrupt mask for OTG Interrupt Latch register bits.
- OTG Timer register: Provides 0.01 ms base programmable timer for use in the OTG state machine.

The following steps are required to enable an OTG interrupt:

1. Set the polarity and level-triggering or edge-triggering mode of the HW Mode Control register.

2. Set the corresponding bits of the OTG Interrupt Enable Rise and OTG Interrupt Enable Fall registers.
3. Set bit OTG_IRQ_E of the HcInterruptEnable register (bit 10).
4. Set bit GLOBAL_INTR_EN of the HW Mode Control register (bit 0).

When an interrupt is generated on HC_IRQ, perform these steps in the interrupt service routine to get the related OTG status:

1. Read the HcInterrupt register. If OTG_IRQ (bit 10) is set, then step 2.
2. Read the OTG Interrupt Latch register. If any of the bits 0 to 4 are set, then step 3.
3. Read the OTG Status register.

The OTG state machine routines are called when any of the inputs is changed. These inputs come from either OTG registers (hardware) or application program (software). The outputs of the state machine include control signals to the OTG register (for hardware) and states or error codes (for software).

The ISP1761 can be configured in OTG mode or in pure host or peripheral mode. Programming the ISP1761 in OTG mode is done by configuring bit 10 of the OTG control register. This will enable OTG-specific mechanisms controlled by the OTG control register bits.

When the OTG protocol is not implemented by the software, the ISP1761 can be used as a host or a peripheral. In this case, bit 10 of the OTG control register will be set to logic 0. The host or peripheral functionality is determined by bit 7 of the OTG Control register.

Programming of OTG registers is done by a SET and RESET scheme. An OTG register has two parts: a 16-bit SET and a 16-bit RESET. Writing logic 1 in a certain position to the SET-type dedicated 16-bit register part will set the respective bit to logic 1 while writing logic 1 to the RESET-type 16-bit dedicated register will change the corresponding bit to logic 0.

9.5 OTG controller registers

Table 79. OTG controller-specific register overview

Address	Register	Reset value	References
037Xh to 038Xh	OTG registers	-	-

Table 80. Address mapping of registers: 32-bit data bus mode

Address	Byte 3	Byte 2	Byte 1	Byte 0
Device ID registers				
0370h	Product ID (read only)		Vendor ID (read only)	
OTG Control register				
0374h	OTG Control (clear)		OTG Control (set)	
OTG Interrupt registers				
0378h	reserved		OTG Status (read only)	
037Ch	OTG Interrupt Latch (clear)		OTG Interrupt Latch (set)	
0380h	OTG Interrupt Enable Fall (clear)		OTG Interrupt Enable Fall (set)	
0384h	OTG Interrupt Enable Rise (clear)		OTG Interrupt Enable Rise (set)	

Table 80. Address mapping of registers: 32-bit data bus mode *...continued*

Address	Byte 3	Byte 2	Byte 1	Byte 0
OTG Timer register				
0388h	OTG Timer (Lower word: clear)		OTG Timer (Lower word: set)	
038Ch	OTG Timer (Higher word: clear)		OTG Timer (Higher word: set)	

Table 81. Address mapping of registers: 16-bit data bus mode

Address	Byte 1	Byte 0	Reference
Device ID registers			
0370h	Vendor ID (read only)		Section 9.5.1.1 on page 92
0372h	Product ID (read only)		Section 9.5.1.2 on page 92
OTG Control register			
0374h	OTG Control (set)		Section 9.5.2.1 on page 93
0376h	OTG Control (clear)		
OTG Interrupt registers			
0378h	OTG Status (read only)		Section 9.5.3.1 on page 94
037Ah	reserved		-
037Ch	OTG Interrupt Latch (set)		Section 9.5.3.2 on page 95
037Eh	OTG Interrupt Latch (clear)		
0380h	OTG Interrupt Enable Fall (set)		Section 9.5.3.3 on page 96
0382h	OTG Interrupt Enable Fall (clear)		
0384h	OTG Interrupt Enable Rise (set)		Section 9.5.3.4 on page 96
0386h	OTG Interrupt Enable Rise (clear)		
OTG Timer register			
0388h	OTG Timer (Lower word: set)		Section 9.5.4.1 on page 97
038Ah	OTG Timer (Lower word: clear)		
038Ch	OTG Timer (Higher word: set)		
038Eh	OTG Timer (Higher word: clear)		

9.5.1 Device Identification registers

9.5.1.1 Vendor ID register

Table 82 shows the bit description of the register.

Table 82. Vendor ID - Vendor Identifier (address 0370h) register: bit description

Bit	Symbol	Access	Value	Description
15 to 0	VENDOR_ID[15:0]	R	04CCh	NXP Semiconductors' Vendor ID

9.5.1.2 Product ID register (R: 0372h)

The bit description of the register is given in Table 83.

Table 83. Product ID - Product Identifier register (address 0372h) bit description

Bit	Symbol	Access	Value	Description
15 to 0	PRODUCT_ID[15:0]	R	1761h	Product ID of the ISP1761

9.5.2 OTG Control register

9.5.2.1 OTG Control register

Table 84 shows the bit allocation of the register.

Table 84. OTG Control register (address set: 0374h, clear: 0376h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]					OTG_DISABLE	OTG_SE0_EN	BDIS_ACON_EN
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	SW_SEL_HC_DC	VBUS_CHRG	VBUS_DISCHRG	VBUS_DRV	SEL_CP_EXT	DM_PULLDOWN	DP_PULLDOWN	DP_PULLUP
Reset	1	0	0	0	0	1	1	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

[1] The reserved bits should always be written with the reset value.

Table 85. OTG Control register (address set: 0374h, clear: 0376h) bit description

Bit[1]	Symbol	Description
15 to 11	-	reserved for future use
10	OTG_DISABLE	**0 —** OTG functionality enabled **1 —** OTG disabled; pure host or peripheral
9	OTG_SE0_EN	This bit is used by the host controller to send SE0 on remote connect. **0 —** No SE0 sent on remote connect detection **1 —** SE0 (bus reset) sent on remote connect detection **Remark:** This bit is normally set when the B-device goes into the B_WAIT_ACON state (recommended sequence: LOC_CONN = 0 → DELAY → 0 ms → OTG_SEQ_EN = 1 → SEL_HC_DC = 0) and is cleared when it comes out of the B_WAIT_ACON state.
8	BDIS_ACON_EN	Enables the A-device to connect if the B-device disconnect is detected
7	SW_SEL_HC_DC	In software HNP mode, this bit selects between the host controller and the peripheral controller. **0 —** Host controller connected to ATX **1 —** Peripheral controller connected to ATX This bit is set to logic 1 by hardware when there is an event corresponding to the BDIS_ACON interrupt. BDIS_ACON_EN is set and there is an automatic pull-up connection on remote disconnect.
6	VBUS_CHRG	Connect V_{BUS} to $V_{CC(I/O)}$ through a resistor
5	VBUS_DISCHRG	Discharge V_{BUS} to ground through a resistor
4	VBUS_DRV	Drive V_{BUS} to 5 V using the charge pump
3	SEL_CP_EXT	**0 —** Internal charge pump selected **1 —** External charge pump selected

Table 85. OTG Control register (address set: 0374h, clear: 0376h) bit description *...continued*

Bit[1]	Symbol	Description
2	DM_PULLDOWN	**DM pull-down**: **0 —** Disable **1 —** Enable
1	DP_PULLDOWN	**DP pull-down**: **0 —** Disable **1 —** Enable
0	DP_PULLUP	**0 —** The pull-up resistor is disconnected from the DP line. The data line pulsing is stopped. **1 —** An internal 1.5 kΩ pull-up resistor is present on the DP line. The data line pulsing is started. **Remark:** When port 1 is in peripheral mode or it plays the role of a peripheral while the OTG functionality is enabled, it depends on the setting of DP_PULLUP and the V_{BUS} sensing signal to connect the DP line to HIGH through a pull-up resister. V_{BUS} is an internal signal. When 5 V is present on the V_{BUS} pin, V_{BUS} = 1.

[1] To use port 1 as a host controller, write 0080 0018h to this register after power-on. To use port 1 as a peripheral controller, write 0006 0400h to this register after power-on.

9.5.3 OTG Interrupt registers

9.5.3.1 OTG Status register

This register indicates the current state of the signals that can generate an interrupt. The bit allocation of the register is given in Table 86.

Table 86. OTG Status register (address 0378h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved							B_SE0_SRP
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	B_SESS_END	reserved		RMT_CONN	ID	DP_SRP	A_B_SESS_VLD	VBUS_VLD
Reset	[1]	0	0	0	[1]	0	[1]	[1]
Access	R	R	R	R	R	R	R	R

[1] The reset value depends on the corresponding OTG status. For details, see Table 87.

Table 87. OTG Status register (address 0378h) bit description

Bit	Symbol	Description
15 to 9	-	reserved for future use
8	B_SE0_SRP	2 ms of SE0 detected in the B-idle state

Table 87. OTG Status register (address 0378h) bit description *...continued*

Bit	Symbol	Description
7	B_SESS_END	V_{BUS} < 0.8 V
6 to 5	-	reserved
4	RMT_CONN	Remote connect detection
3	ID	ID pin digital input
2	DP_SRP	DP asserted during SRP
1	A_B_SESS_VLD	A-session valid for the A-device. B-session valid for the B-device.
0	VBUS_VLD	A-device V_{BUS} valid comparator, indicates V_{BUS} > 4.4 V

9.5.3.2 OTG Interrupt Latch register

The OTG Interrupt Latch register indicates the source that generated the interrupt. The status of this register bits depends on the settings of the Interrupt Enable Fall and Interrupt Enable Rise registers, and the occurrence of the respective events.

The bit allocation of the register is given in Table 88.

Table 88. OTG Interrupt Latch register (address set: 037Ch, clear: 037Eh) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]						OTG_TMR_TIMEOUT	B_SE0_SRP
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R	R/S/C	R/S/C
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	B_SESS_END	BDIS_ACON	OTG_RESUME	RMT_CONN	ID	DP_SRP	A_B_SESS_VLD	VBUS_VLD
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

[1] The reserved bits should always be written with the reset value.

Table 89. OTG Interrupt Latch register (address set: 037Ch, clear: 037Eh) bit description

Bit	Symbol	Description
15 to 10	-	reserved for future use
9	OTG_TMR_TIMEOUT	OTG timer time-out
8	B_SE0_SRP	2 ms of SE0 detected in the B-idle state
7	B_SESS_END	V_{BUS} < 0.8 V
6	BDIS_ACON	Indicates that the BDIS_ACON event has occurred
5	OTG_RESUME	J → K resume change detected
4	RMT_CONN	Remote connect detection
3	ID	Indicates change on pin ID
2	DP_SRP	DP asserted during SRP
1	A_B_SESS_VLD	A-session valid for the A-device. B-session valid for the B-device.
0	VBUS_VLD	Indicates change in the VBUS_VLD status

9.5.3.3 OTG Interrupt Enable Fall register

Table 90 shows the bit allocation of this register that enables interrupts on transition from HIGH-to-LOW.

Table 90. OTG Interrupt Enable Fall register (address set: 0380h, clear: 0382h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]							B_SE0_SRP
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	B_SESS_END	reserved		RMT_CONN	ID	reserved	A_B_SESS_VLD	VBUS_VLD
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

[1] The reserved bits should always be written with the reset value.

Table 91. OTG Interrupt Enable Fall register (address set: 0380h, clear: 0382h) bit description

Bit	Symbol	Description
15 to 9	-	reserved for future use
8	B_SE0_SRP	IRQ asserted when the bus exits from at least 2 ms of the SE0 state
7	B_SESS_END	IRQ asserted when V_{BUS} > 0.8 V
6 to 5	-	reserved
4	RMT_CONN	IRQ asserted on RMT_CONN removal
3	ID	IRQ asserted on the ID pin transition from HIGH to LOW
2	-	reserved
1	A_B_SESS_VLD	IRQ asserted on removing A-session valid for the A-device or B-session valid for the B-device condition
0	VBUS_VLD	IRQ asserted on the falling edge of V_{BUS}

9.5.3.4 OTG Interrupt Enable Rise register

This register (see Table 92 for bit allocation) enables interrupts on transition from LOW-to-HIGH.

Table 92. OTG Interrupt Enable Rise register (address set: 0384h, clear: 0386h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]						OTG_TMR_TIMEOUT	B_SE0_SRP
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

Bit	7	6	5	4	3	2	1	0
Symbol	B_SESS_END	BDIS_ACON	OTG_RESUME	RMT_CONN	ID	DP_SRP	A_B_SESS_VLD	VBUS_VLD
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

[1] The reserved bits should always be written with the reset value.

Table 93. OTG Interrupt Enable Rise register (address set: 0384h, clear: 0386h) bit description

Bit	Symbol	Description
15 to 10	-	reserved
9	OTG_TMR_TIMEOUT	IRQ asserted on OTG timer time-out
8	B_SE0_SRP	IRQ asserted when at least 2 ms of SE0 is detected in the B-idle state
7	B_SESS_END	IRQ asserted when V_{BUS} is less than 0.8 V
6	BDIS_ACON	IRQ asserted on BDIS_ACON condition
5	OTG_RESUME	IRQ asserted on J-K resume
4	RMT_CONN	IRQ asserted on RMT_CONN
3	ID	IRQ asserted on the ID pin transition from LOW to HIGH
2	DP_SRP	IRQ asserted when DP is asserted during SRP
1	A_B_SESS_VLD	IRQ asserted on the A-session valid for the A-device or on the B-session valid for the B-device
0	VBUS_VLD	IRQ asserted on the rising edge of V_{BUS}

9.5.4 OTG Timer register

9.5.4.1 OTG Timer register

This is a 32-bit register organized as two 16-bit fields. These two fields have separate set and clear addresses. Table 94 shows the bit allocation of the register.

Table 94. OTG Timer register (address low word set: 0388h, low word clear: 038Ah; high word set: 038Ch, high word clear: 038Eh) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	START_TMR	reserved[1]						
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	TIMER_INIT_VALUE[23:16]							
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	TIMER_INIT_VALUE[15:8]							
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

Bit	7	6	5	4	3	2	1	0
Symbol	TIMER_INIT_VALUE[7:0]							
Reset	0	0	0	0	0	0	0	0
Access	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C

[1] The reserved bits should always be written with the reset value.

Table 95. OTG Timer register (address low word set: 0388h, low word clear: 038Ah; high word set: 038Ch, high word clear: 038Eh) bit description

Bit	Symbol	Description
31	START_TMR	This is the start/stop bit of the OTG timer. Writing logic 1 will cause the OTG timer to load TMR_INIT_VALUE into the counter and start to count. Writing logic 0 will stop the timer. This bit is automatically cleared when the OTG timer is timed out. **0 —** stop the timer **1 —** start the timer
30 to 24	-	reserved
23 to 0	TIMER_INIT_VALUE[23:0]	These bits define the initial value used by the OTG timer. The timer interval is 0.01 ms. Maximum time allowed is 167.772 s.

10. Peripheral controller

10.1 Introduction

The design of the peripheral controller in the ISP1761 is compatible with the NXP *ISP1582 Hi-Speed Universal Serial Bus peripheral controller* IC. The functionality of the peripheral controller in the ISP1761 is similar to the ISP1582 in 16-bit bus mode. In addition, the register sets are also similar, with only a few variations.

The USB protocol and data transfer operations of the peripheral controller are executed using external firmware. The external microcontroller or microprocessor can access the peripheral controller-specific registers through the local bus interface. The transfer of data between a microprocessor and the peripheral controller can be done in PIO mode or programmed DMA mode.

For details on general functional description of the peripheral controller, refer to the ISP1582 data sheet. For details on the software programming, refer to Ref. 6 “ISP1582/83 Firmware Programming Guide (AN10039)” and Ref. 4 “ISP1582/83 Control Pipe (AN10031)”.

10.1.1 Direct Memory Access (DMA)

The DMA controller of the ISP1761 is used to transfer data between the system memory and endpoints buffers. It is a slave DMA controller that requires an external DMA master to control the transfer.

10.1.1.1 DMA for the IN endpoint

When the internal DMA is enabled and at least one buffer is free, the DC_DREQ line is asserted. The external DMA controller then starts negotiating for control of the bus. As soon as it has access, it asserts the DC_DACK line and starts writing data. The burst length is programmable. When the number of bytes equal to the burst length has been written, the DC_DREQ line is de-asserted. As a result, the DMA controller de-asserts the DC_DACK line and releases the bus. At that moment, the whole cycle restarts for the next burst. When the buffer is full, the DC_DREQ line is de-asserted and the buffer is validated, which means that it is sent to the host at the next IN token. When the DMA transfer is terminated, the buffer is also validated, even if it is not full.

10.1.1.2 DMA for the OUT endpoint

When the internal DMA is enabled and at least one buffer is full, the DC_DREQ line is asserted. The external DMA controller then starts negotiating for control of the bus. As soon as it has access, it asserts the DC_DACK line and starts reading data. The burst length is programmable. When the number of bytes equal to the burst length has been read, the DC_DREQ line is de-asserted. As a result, the DMA controller de-asserts the DC_DACK line and releases the bus. At that moment, the whole cycle restarts for the next burst. When all the data is read, the DC_DREQ line is de-asserted and the buffer is cleared. This means that it can be overwritten when a new packet arrives.

10.1.1.3 DMA initialization

To reduce the power consumption, a controllable clock that drives DMA controller circuits is turned off, by default. If the DMA functionality is required by an application, DMACLKON (bit 9) of the Mode register (address: 020Ch) must be enabled during initialization of the

peripheral controller. If DMA is not required by the application, DMACLKON can be permanently disabled to save current. The burst counter, DMA bus width, and the polarity of DC_DREQ and DC_DACK must accordingly be set.

The ISP1761 supports only counter mode DMA transfer. To enable counter mode, ensure that DIS_XFER_CNT in the DcDMAConfiguration register (address: 0238h) is set to zero.

Before starting the DMA transfer, preset the interrupt enable bit IEDMA in the Interrupt Enable register (address: 0214h) and the DMA Interrupt Enable register (address: 0254h). The ISP1761 supports two interrupt trigger modes: level and edge. The pulse width, which in edge mode, is determined by setting the Interrupt Pulse Width register (address: 0280h). The default value is 1Eh, which indicates that the interrupt pulse width is 1 µs. The minimum interrupt pulse width is approximately 30 ns when set to logic 1. Do not write a zero to this register.

The interrupt polarity must also be correctly set.

Remark: DMA can apply to all endpoints on the chip. It, however, can only take place for one endpoint at a time. The selected endpoint is assigned by setting the endpoint number in the DMA Endpoint register (address: 0258h). It will also internally redirect the endpoint buffer of the selected endpoint to the DMA controller bus. In addition, it requires a preceding process to program the endpoint type, the endpoint maximum packet size, and the direction of the endpoint.

When setting the Endpoint Index register (address: 022Ch), the endpoint buffer of the selected endpoint is directed to the internal CPU bus for the PIO access. Therefore, it is required to reconfigure the Endpoint Index register with endpoint number, which is not an endpoint number in use for the DMA transfer to avoid any confusion.

10.1.1.4 Starting DMA

Dynamically assign the DMA Transfer Counter register (address: 0234h) for each DMA transfer.

The transfer will end once transfer counter reaches zero. Bit DMA_XFER_OK in the DMA Interrupt Reason register (address: 0250h) will be asserted to indicate that the DMA transfer has successfully stopped. If the transfer counter is larger than the burst counter, the DC_DREQ signal will drop at the end of each burst transfer. DC_DREQ will reassert at the beginning of each burst. For a 32-bit DMA transfer, the minimum burst length is 4 bytes. This means that the burst length is only one DMA cycle. Therefore, DC_DREQ and DC_DACK will toggle by each DMA cycle. For a 16-bit DMA transfer, the minimum burst length is 2 bytes.

Setting bit GDMA read or GDMA write in the DMA Command register (address: 0230h) will start the DMA transfer.

Remark: DACK and CS_N should not be active at the same time.

10.1.1.5 DMA stop and interrupt handling

The DMA transfer will either successfully be completed or terminated, which can be identified by reading the status in the DcInterrupt register (address: 0218h) and DMA Interrupt Reason register (address: 0250h).

If bit DMA_XFER_OK in the DMA Interrupt Reason register is asserted, it means that the transfer counter has reached zero and the DMA transfer is successfully stopped.

If bit INT_EOT in the DMA Interrupt Reason register is set, it indicates that a short or empty packet is received. This means that DMA transfer terminated. Normally, for an OUT transfer, it means that remote host wishes to terminate the DMA transfer.

If both the bits DMA_XFER_OK and INT_EOT are set, it means that the transfer counter reached zero and the last packet of the transfer is a short packet. Therefore, the DMA transfer is successfully stopped.

Setting bit GDMA Stop in the DMA Command register (address: 0230h) will force the DMA to stop and bit GDMA_STOP in the DMA Interrupt Reason register (address: 0250h) will be set to indicate this event.

Setting bit Reset DMA in the DMA Command register (address: 0230h) will force the DMA to stop and initialize the DMA core to its power-on reset state.

10.2 Endpoint description

Each USB peripheral is logically composed of several independent endpoints. An endpoint acts as a terminus of a communication flow between the USB host and the USB peripheral. At design time, each endpoint is assigned a unique endpoint identifier; see Table 96. The combination of the peripheral address (given by the host during enumeration), the endpoint number, and the transfer direction allows each endpoint to be uniquely referenced.

The peripheral controller has 8 kB of internal FIFO memory, which is shared among the enabled USB endpoints. The two control endpoints are fixed 64 bytes long. Any of the seven IN and seven OUT endpoints can separately be enabled or disabled. The endpoint type (interrupt, isochronous or bulk) and packet size of these endpoints can individually be configured, depending on the requirements of the application. Optional double buffering increases the data throughput of these data endpoints.

Table 96. Endpoint access and programmability

Endpoint identifier	Maximum packet size	Double buffering	Endpoint type	Direction
EP0SETUP	8 bytes (fixed)	no	set-up token	OUT
EP0RX	64 bytes (fixed)	no	control OUT	OUT
EP0TX	64 bytes (fixed)	no	control IN	IN
EP1RX	programmable	yes	programmable	OUT
EP1TX	programmable	yes	programmable	IN
EP2RX	programmable	yes	programmable	OUT
EP2TX	programmable	yes	programmable	IN
EP3RX	programmable	yes	programmable	OUT
EP3TX	programmable	yes	programmable	IN
EP4RX	programmable	yes	programmable	OUT
EP4TX	programmable	yes	programmable	IN
EP5RX	programmable	yes	programmable	OUT
EP5TX	programmable	yes	programmable	IN
EP6RX	programmable	yes	programmable	OUT

Table 96. Endpoint access and programmability *…continued*

Endpoint identifier	Maximum packet size	Double buffering	Endpoint type	Direction
EP6TX	programmable	yes	programmable	IN
EP7RX	programmable	yes	programmable	OUT
EP7TX	programmable	yes	programmable	IN

10.3 Clear buffer

Use clear buffer when data needs to be discarded under the following conditions:

- IN endpoint: If the host aborts a read operation, the residual data in the IN endpoint buffer must be cleared using the Clear Buffer command. See Table 126.
- OUT endpoint: If the host aborts a write operation, the residual data in the OUT endpoint buffer must be cleared using the CLBUF bit. See Table 113.

For example, to clear a double buffer data IN endpoint 1, set the following registers in the firmware as:

1. Assign a value to the Endpoint Index register. It can be any value other than the value assigned to the DMA Endpoint register. In this example, do not assign 3h to the Endpoint Index register. See remark in Section 10.6.1.
2. Assign DMA Endpoint register = 3h
3. Assign DMA Command register = 0Fh
4. Assign DMA Endpoint register = 3h
5. Assign DMA Command register = 0Fh

For example, to clear a double buffer data OUT endpoint 1, set the following registers in the firmware as:

1. Assign a value to the DMA Endpoint register. It can be any value other than the value assigned to the Endpoint Index register. In this example, do not assign 2h to the DMA Endpoint register. See remark in Section 10.6.1.
2. Assign Endpoint Index register = 2h
3. Assign Control Function register = 10Fh
4. Assign Endpoint Index register = 2h
5. Assign Control Function register = 10Fh

10.4 Differences between the ISP1761 and ISP1582 peripheral controllers

This section explains the variations between the ISP1761 and ISP1582 peripheral controllers in terms of register bits and their associated functions.

10.4.1 ISP1761 initialization registers

- The ISP1582 supports 16-bit bus access. The register addresses are 2 bytes aligned. The ISP1761 supports 16-bit and 32-bit bus accesses. To support the 32-bit access, the DATA_BUS_WIDTH bit in the HW Mode Control register must be initialized.

- In 32-bit bus access mode, the register addresses are 4 bytes aligned. Therefore, the DcBufferStatus register can be accessed using the upper-two bytes of the Buffer Length register.
- The SOFTCT bit in the Mode register has been removed. The DP_PULLUP control bit in the OTG Control register is used in the ISP1761 in place of the SOFTCT bit in the ISP1582.
- Added the Interrupt Pulse Width register to define the pulse width of the interrupt signal.

10.4.2 ISP1761 DMA

- DMA mode 1 and DMA mode 2 in the ISP1582 are not supported in the ISP1761.
- In DMA mode 0, counter mode is supported and external-EOT mode has been removed.
- Supports the 16-bit and 32-bit DMA. Does not support the 8-bit DMA.
- The RD_N and WR_N signals are available for the DMA data strobe. These signals are also used as data strobe signals during the PIO access. An internal multiplex will redirect these signals to the DMA controller for the DMA transfer or to registers for the PIO access.

For details on the DMA programming, refer to application note Ref. 7 “ISP1761 Peripheral DMA Initialization (AN10040)”.

10.4.3 ISP1761 peripheral suspend indication

- A HIGH level on the DC_SUSPEND/WAKEUP_N pin indicates that the peripheral has entered suspend mode. The pulse indication mode has been removed.

10.4.4 ISP1761 interrupt and DMA common mode

In common mode, the interrupt and DMA signals of the peripheral controller are redirected to pins that are used by the host controller because the host controller and the peripheral controller share the same pins. Some control bits must be set in the HW Mode Control register, see Section 8.3.1.

10.5 Peripheral controller-specific registers

Table 97. Peripheral controller-specific register overview

Address	Register	Reset value	References
Initialization registers			
0200h	Address	00h	Section 10.5.1 on page 104
020Ch	Mode	0000h	Section 10.5.2 on page 105
0210h	Interrupt Configuration	FCh	Section 10.5.3 on page 106
0212h	Debug	0008h	Section 10.5.4 on page 107
0214h	DcInterruptEnable	0000 0000h	Section 10.5.5 on page 108
0300h	HW Mode Control	0000 0000h	Section 8.3.1 on page 43
0374h	OTG Control	0000 0086h	Section 9.5.2.1 on page 93
Data flow registers			
022Ch	Endpoint Index	20h	Section 10.6.1 on page 109

Table 97. Peripheral controller-specific register overview …continued

Address	Register	Reset value	References
0228h	Control Function	00h	Section 10.6.2 on page 110
0220h	Data Port	0000 0000h	Section 10.6.3 on page 112
021Ch	Buffer Length	0000h	Section 10.6.4 on page 112
021Eh	DcBufferStatus	00h	Section 10.6.5 on page 113
0204h	Endpoint MaxPacketSize	0000h	Section 10.6.6 on page 114
0208h	Endpoint Type	0000h	Section 10.6.7 on page 114
DMA registers			
0230h	DMA Command	FFh	Section 10.7.1 on page 116
0234h	DMA Transfer Counter	0000 0000h	Section 10.7.2 on page 117
0238h	DcDMAConfiguration	0001h	Section 10.7.3 on page 118
023Ch	DMA Hardware	04h	Section 10.7.4 on page 119
0250h	DMA Interrupt Reason	0000h	Section 10.7.5 on page 120
0254h	DMA Interrupt Enable	0000h	Section 10.7.6 on page 121
0258h	DMA Endpoint	00h	Section 10.7.7 on page 121
0264h	DMA Burst Counter	0004h	Section 10.7.8 on page 122
General registers			
0218h	DcInterrupt	0000 0000h	Section 10.8.1 on page 123
0270h	DcChipID	0015 8210h	Section 10.8.2 on page 125
0274h	Frame Number	0000h	Section 10.8.3 on page 125
0278h	DcScratch	0000h	Section 10.8.4 on page 125
027Ch	Unlock Device	0000h	Section 10.8.5 on page 126
0280h	Interrupt Pulse Width	001Eh	Section 10.8.6 on page 126
0284h	Test Mode	00h	Section 10.8.7 on page 127

10.5.1 Address register

This register sets the USB assigned address and enables the USB peripheral. Table 98 shows the bit allocation of the register.

The DEVADDR[6:0] bits will be cleared whenever a bus reset, a power-on reset or a soft reset occurs. The DEVEN bit will be cleared whenever a power-on reset or a soft reset occurs, and will remain unchanged on a bus reset.

In response to standard USB request SET_ADDRESS, firmware must write the (enabled) peripheral address to the Address register, followed by sending an empty packet to the host. The **new** peripheral address is activated when the peripheral receives acknowledgment from the host for the empty packet token.

Table 98. Address register (address 0200h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	DEVEN	DEVADDR[6:0]						
Reset	0	0	0	0	0	0	0	0
Bus reset	unchanged	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 99. Address register (address 0200h) bit description

Bit	Symbol	Description
7	DEVEN	**Device Enable**: Logic 1 enables the device. The device will not respond to the host, unless this bit is set.
6 to 0	DEVADDR[6:0]	**Device Address**: This field specifies the USB device peripheral.

10.5.2 Mode register

This register consists of 2 bytes (bit allocation: see Table 100).

The Mode register controls resume, suspend and wake-up behavior, interrupt activity, soft reset and clock signals.

Table 100. Mode register (address 020Ch) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]						DMACLKON	VBUSSTAT
Reset	0	0	0	0	0	0	0	0
Bus 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	CLKAON	SNDRSU	GOSUSP	SFRESET	GLINTENA	WKUPCS	reserved[1]	
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	unchanged	0	0	unchanged
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 101. Mode register (address 020Ch) bit description

Bit	Symbol	Description
15 to 10	-	reserved
9	DMACLKON	**DMA Clock On**: **1 —** Supply clock to the DMA circuit. **0 —** Power saving mode. The DMA circuit will stop completely to save power.
8	VBUSSTAT	**V_{BUS} Status**: This bit reflects the V_{BUS} pin status. When implementing a pure host or peripheral, the OTG_DISABLE bit in the OTG Control register (374h) must be set to logic 1 so that the VBUSSTAT bit is updated with the correct value.
7	CLKAON	**Clock Always On**: **1 —** Enable the Clock-Always-On feature **0 —** Disable the Clock-Always-On feature When the Clock-Always-On feature is disabled, a GOSUSP event can stop the clock. The clock is stopped after a delay of approximately 2 ms. Therefore, the peripheral controller will consume less power. If the Clock-Always-On feature is enabled, clocks are always running and the GOSUSP event is unable to stop the clock while the peripheral controller enters the suspend state.

Table 101. Mode register (address 020Ch) bit description *...continued*

Bit	Symbol	Description
6	SNDRSU	**Send Resume**: Writing logic 1, followed by logic 0 will generate an upstream resume signal of 10 ms duration, after a 5 ms delay.
5	GOSUSP	**Go Suspend**: Writing logic 1, followed by logic 0 will activate suspend mode.
4	SFRESET	**Soft Reset**: Writing logic 1, followed by logic 0 will enable a software-initiated reset to the ISP1761. A soft reset is similar to a hardware-initiated reset using the RESET_N pin.
3	GLINTENA	**Global Interrupt Enable**: Logic 1 enables all interrupts. Individual interrupts can be masked by clearing the corresponding bits in the DcInterruptEnable register. When this bit is not set, an unmasked interrupt will not generate an interrupt trigger on the interrupt pin. If the global interrupt, however, is enabled while there is any pending unmasked interrupt, an interrupt signal will immediately be generated on the interrupt pin. If the interrupt is set to pulse mode, the interrupt events that were generated before the global interrupt is enabled may be dropped.
2	WKUPCS	**Wake up on Chip Select**: Logic 1 enables wake-up through a valid register read on the ISP1761. A read will invoke the chip clock to restart. A write to the register before the clock is stable may cause malfunctioning.
1 to 0	-	reserved

10.5.3 Interrupt Configuration register

This 1 byte register determines the behavior and polarity of the INT output. The bit allocation is shown in Table 102. When the USB SIE receives or generates an ACK, NAK or NYET, it will generate interrupts depending on three Debug mode fields.

CDBGMOD[1:0] — Interrupts for the control endpoint 0

DDBGMODIN[1:0] — Interrupts for the DATA IN endpoints 1 to 7

DDBGMODOUT[1:0] — Interrupts for the DATA OUT endpoints 1 to 7

The Debug mode settings for CDBGMOD, DDBGMODIN and DDBGMODOUT allow you to individually configure when the ISP1761 sends an interrupt to the external microprocessor. Table 104 lists the available combinations.

Bit INTPOL controls the signal polarity of the INT output: active HIGH or LOW, rising or falling edge. For level-triggering, bit INTLVL must be made logic 0. By setting INTLVL to logic 1, an interrupt will generate a pulse of 60 ns (edge-triggering).

Table 102. Interrupt Configuration register (address 0210h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	CDBGMOD[1:0]		DDBGMODIN[1:0]		DDBGMODOUT[1:0]		INTLVL	INTPOL
Reset	1	1	1	1	1	1	0	0
Bus reset	1	1	1	1	1	1	unchanged	unchanged
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 103. Interrupt Configuration register (address 0210h) bit description

Bit	Symbol	Description
7 to 6	CDBGMOD[1:0]	**Control 0 Debug Mode**: For values, see Table 104
5 to 4	DDBGMODIN[1:0]	**Data Debug Mode IN**: For values, see Table 104
3 to 2	DDBGMODOUT[1:0]	**Data Debug Mode OUT**: For values, see Table 104
1	INTLVL	**Interrupt Level**: Selects signaling mode on output INT: 0 = level; 1 = pulsed. In pulsed mode, an interrupt produces a 60 ns pulse. Bus reset value: unchanged.
0	INTPOL	**Interrupt Polarity**: Selects the signal polarity on output INT: 0 = active LOW; 1 = active HIGH. Bus reset value: unchanged.

Table 104. Debug mode settings

Value	CDBGMOD	DDBGMODIN	DDBGMODOUT
00h	interrupt on all ACK and NAK	interrupt on all ACK and NAK	interrupt on all ACK, NYET and NAK
01h	interrupt on all ACK	interrupt on ACK	interrupt on ACK and NYET
1Xh	interrupt on all ACK and first NAK[1]	interrupt on all ACK and first NAK[1]	interrupt on all ACK, NYET and first NAK[1]

[1] First NAK: The first NAK on an IN or OUT token after a previous ACK response.

10.5.4 Debug register

This register can be accessed using address 0212h in 16-bit bus access mode or using the upper-two bytes of the Interrupt Configuration register in 32-bit bus access mode. For the bit allocation, see Table 105.

Table 105. Debug register (address 0212h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Bus 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[1]							DEBUG
Reset	0	0	0	0	1	0	0	0
Bus reset	0	0	0	0	1	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 106. Debug register (address 0212h) bit allocation

Bit	Symbol	Description
15 to 1	-	reserved
0	DEBUG	Always set this bit to logic 0 in both 16-bit and 32-bit accesses.

10.5.5 DcInterruptEnable register

This register enables or disables individual interrupt sources. The interrupt for each endpoint can individually be controlled through the associated IEPnRX or IEPnTX bits, here n represents the endpoint number. All interrupts can globally be disabled through bit GLINTENA in the Mode register (see Table 100).

An interrupt is generated when the USB SIE receives or generates an ACK or NAK on the USB bus. The interrupt generation depends on Debug mode settings of bit fields CDBGMOD[1:0], DDBGMODIN[1:0] and DDBGMODOUT[1:0].

All data IN transactions use the Transmit buffers (TX) that are handled by DDBGMODIN bits. All data OUT transactions go through the Receive buffers (RX) that are handled by DDBGMODOUT bits. Transactions on control endpoint 0 (IN, OUT and SETUP) are handled by CDBGMOD bits.

Interrupts caused by events on the USB bus (SOF, suspend, resume, bus reset, set up and high-speed status) can also be individually controlled. A bus reset disables all enabled interrupts, except bit IEBRST (bus reset) that remains unchanged.

The DcInterruptEnable register consists of 4 bytes. The bit allocation is given in Table 107.

Table 107. DcInterruptEnable - Device Controller Interrupt Enable register (address 0214h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]						IEP7TX	IEP7RX
Reset	0	0	0	0	0	0	0	0
Bus 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	IEP6TX	IEP6RX	IEP5TX	IEP5RX	IEP4TX	IEP4RX	IEP3TX	IEP3RX
Reset	0	0	0	0	0	0	0	0
Bus 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	IEP2TX	IEP2RX	IEP1TX	IEP1RX	IEP0TX	IEP0RX	reserved[1]	IEP0SETUP
Reset	0	0	0	0	0	0	0	0
Bus 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	IEVBUS	IEDMA	IEHS_STA	IERESM	IESUSP	IEPSOF	IESOF	IEBRST
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	unchanged
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 108. DcInterruptEnable - Device Controller Interrupt Enable register (address 0214h) bit description

Bit	Symbol	Description
31 to 26	-	reserved
25	EP7TX	Logic 1 enables interrupt from the indicated endpoint.
24	EP7RX	Logic 1 enables interrupt from the indicated endpoint.
23	EP6TX	Logic 1 enables interrupt from the indicated endpoint.
22	EP6RX	Logic 1 enables interrupt from the indicated endpoint.
21	EP5TX	Logic 1 enables interrupt from the indicated endpoint.
20	EP5RX	Logic 1 enables interrupt from the indicated endpoint.
19	EP4TX	Logic 1 enables interrupt from the indicated endpoint.
18	EP4RX	Logic 1 enables interrupt from the indicated endpoint.
17	EP3TX	Logic 1 enables interrupt from the indicated endpoint.
16	EP3RX	Logic 1 enables interrupt from the indicated endpoint.
15	EP2TX	Logic 1 enables interrupt from the indicated endpoint.
14	EP2RX	Logic 1 enables interrupt from the indicated endpoint.
13	EP1TX	Logic 1 enables interrupt from the indicated endpoint.
12	IEP1RX	Logic 1 enables interrupt from the indicated endpoint.
11	IEP0TX	Logic 1 enables interrupt from the control IN endpoint 0.
10	IEP0RX	Logic 1 enables interrupt from the control OUT endpoint 0.
9	-	reserved
8	IEP0SETUP	Logic 1 enables interrupt for the set-up data received on endpoint 0.
7	IEVBUS	Logic 1 enables interrupt for V_{BUS} sensing.
6	IEDMA	Logic 1 enables interrupt on detecting a DMA status change.
5	IEHS_STA	Logic 1 enables interrupt on detecting a high-speed status change.
4	IERESM	Logic 1 enables interrupt on detecting a resume state.
3	IESUSP	Logic 1 enables interrupt on detecting a suspend state.
2	IEPSOF	Logic 1 enables interrupt on detecting a pseudo SOF.
1	IESOF	Logic 1 enables interrupt on detecting an SOF.
0	IEBRST	Logic 1 enables interrupt on detecting a bus reset.

10.6 Data flow registers

10.6.1 Endpoint Index register

The Endpoint Index register selects a target endpoint for register access by the microcontroller. The register consists of 1 byte, and the bit allocation is shown in Table 109.

The following registers are indexed:

- Buffer Length
- DcBufferStatus
- Control Function
- Data Port

- Endpoint MaxPacketSize
- Endpoint Type

For example, to access the OUT data buffer of endpoint 1 using the Data Port register, the Endpoint Index register must be written first with 02h.

Remark: The Endpoint Index register and the DMA Endpoint register must not point to the same endpoint, irrespective of IN and OUT.

Table 109. Endpoint Index register (address 022Ch) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved[1]		EP0SETUP	ENDPIDX[3:0]				DIR
Reset	0	0	1	0	0	0	0	0
Bus reset	0	0	1	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 110. Endpoint Index register (address 022Ch) bit description

Bit	Symbol	Description
7 to 6	-	reserved
5	EP0SETUP	**Endpoint 0 Set up**: Selects the SETUP buffer for endpoint 0. **0 —** Data buffer **1 —** SETUP buffer Must be logic 0 for access to endpoints other than set-up token buffer.
4 to 1	ENDPIDX[3:0]	**Endpoint Index**: Selects the target endpoint for register access of buffer length, buffer status, control function, data port, endpoint type and MaxPacketSize.
0	DIR	**Direction bit**: Sets the target endpoint as IN or OUT. **0 —** Target endpoint refers to OUT (RX) FIFO **1 —** Target endpoint refers to IN (TX) FIFO

Table 111. Addressing of endpoint buffers

Buffer name	EP0SETUP	ENDPIDX	DIR
SETUP	1	00h	0
Control OUT	0	00h	0
Control IN	0	00h	1
Data OUT	0	0Xh	0
Data IN	0	0Xh	1

10.6.2 Control Function register

The Control Function register performs the buffer management on endpoints. It consists of 1 byte, and the bit configuration is given in Table 112. The register bits can stall, clear or validate any enabled data endpoint. Before accessing this register, the Endpoint Index register must first be written to specify the target endpoint.

Table 112. Control Function register (address 0228h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved[1]			CLBUF	VENDP	DSEN	STATUS	STALL
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 113. Control Function register (address 0228h) bit description

Bit	Symbol	Description
7 to 5	-	reserved
4	CLBUF	**Clear Buffer**: Logic 1 clears the RX buffer of the indexed endpoint; the TX buffer is not affected. The RX buffer is automatically cleared once the endpoint is completely read. This bit is set only when it is necessary to forcefully clear the buffer. **Remark:** If using double buffer, to clear both the buffers issue the CLBUF command two times, that is, set and clear this bit two times.
3	VENDP	**Validate Endpoint**: Logic 1 validates data in the TX FIFO of an IN endpoint for sending on the next IN token. In general, the endpoint is automatically validated when its FIFO byte count has reached the endpoint MaxPacketSize. This bit is set only when it is necessary to validate the endpoint with the FIFO byte count that is below the Endpoint MaxPacketSize.
2	DSEN	**Data Stage Enable**: This bit controls the response of the ISP1761 to a control transfer. After the completion of the set-up stage, firmware must determine whether a data stage is required. For control OUT, firmware will set this bit and the ISP1761 goes into the data stage. Otherwise, the ISP1761 will NAK the data stage transfer. For control IN, firmware will set this bit before writing data to the TX FIFO and validate the endpoint. If no data stage is required, firmware can immediately set the STATUS bit after the set-up stage. **Remark:** The DSEN bit is cleared once the OUT token is acknowledged by the device and the IN token is acknowledged by the PC host. This bit cannot be read back and reading this bit will return logic 0.
1	STATUS	**Status Acknowledge**: Only applicable for control IN and OUT. This bit controls the generation of ACK or NAK during the status stage of a SETUP transfer. It is automatically cleared when the status stage is completed and a SETUP token is received. No interrupt signal will be generated. **0 —** Sends NAK **1 —** Sends an empty packet following the IN token (peripheral-to-host) or ACK following the OUT token (host-to-peripheral) **Remark:** The STATUS bit is cleared to zero once the zero-length packet is acknowledged by the device or the PC host. **Remark:** Data transfers preceding the status stage must first be fully completed before the STATUS bit can be set.
0	STALL	**Stall Endpoint**: Logic 1 stalls the indexed endpoint. This bit is not applicable for isochronous transfers. **Remark:** Stalling a data endpoint will confuse the Data Toggle bit about the stalled endpoint because the internal logic picks up from where it is stalled. Therefore, the Data Toggle bit must be reset by disabling and re-enabling the corresponding endpoint (by setting bit ENABLE to logic 0, followed by logic 1 in the Endpoint Type register) to reset the PID.

10.6.3 Data Port register

This register provides direct access for a microcontroller to the FIFO of the indexed endpoint.

Peripheral to host (IN endpoint): After each write, an internal counter is automatically incremented, by two in 16-bit mode and four in 32-bit mode, to the next location in the TX FIFO. When all bytes have been written (FIFO byte count = endpoint MaxPacketSize), the buffer is automatically validated. The data packet will then be sent on the next IN token. Whenever required, the Control Function register (bit VENDP) can validate the endpoint whose byte count is less than MaxPacketSize.

Remark: The buffer can automatically be validated using the Buffer Length register.

Host to peripheral (OUT endpoint): After each read, an internal counter is automatically decremented, by two in 16-bit mode and four in 32-bit mode, to the next location in the RX FIFO. When all bytes have been read, the buffer contents are automatically cleared. A new data packet can then be received on the next OUT token. Buffer contents can also be cleared through the Control Function register (bit CLBUF), whenever it is necessary to forcefully clear contents.

The Data Port register description when the ISP1761 is in 32-bit mode is given in Table 114.

Table 114. Data Port register (address 0220h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	DATAPORT [31:0]	R/W	0000 0000h	**Data Port**: A 500 ns delay starting from the reception of the endpoint interrupt may be required for the first read from the data port.

The Data Port register description when the ISP1761 is in 16-bit mode is given in Table 115.

Table 115. Data Port register (address 0220h) bit description

Bit	Symbol	Access	Value	Description
15 to 0	DATAPORT [15:0]	R/W	0000 0000h	**Data Port**: A 500 ns delay starting from the reception of the endpoint interrupt may be required for the first read from the data port.

10.6.4 Buffer Length register

This register determines the current packet size (DATACOUNT) of the indexed endpoint FIFO. The bit description is given in Table 116.

The Buffer Length register is automatically loaded with the FIFO size, when the Endpoint MaxPacketSize register is written (see Table 120). A smaller value can be written when required. After a bus reset, the Buffer Length register is made zero.

IN endpoint: When the data transfer is performed in multiples of MaxPacketSize, the Buffer Length register is not significant. This register is useful only when transferring data that is not a multiple of MaxPacketSize. The following two examples demonstrate the significance of the Buffer Length register.

Example 1: Consider that the transfer size is 512 bytes and the MaxPacketSize is programmed as 64 bytes, the Buffer Length register need not be filled. This is because the transfer size is a multiple of MaxPacketSize, and MaxPacketSize packets will be automatically validated because the last packet is also of MaxPacketSize.

Example 2: Consider that the transfer size is 510 bytes and the MaxPacketSize is programmed as 64 bytes, the Buffer Length register should be filled with 62 bytes just before the microcontroller writes the last packet of 62 bytes. This ensures that the last packet, which is a short packet of 62 bytes, is automatically validated.

Use the VENDP bit in the Control register if you are not using the Buffer Length register.

This is applicable only to PIO mode access.

OUT endpoint: The DATACOUNT value is automatically initialized to the number of data bytes sent by the host on each ACK.

Remark: When using a 16-bit microprocessor bus, the last byte of an odd-sized packet is output as the lower byte (LSByte).

Table 116. Buffer Length register (address 021Ch) bit description

Bit	Symbol	Access	Value	Description
15 to 0	DATACOUNT [15:0]	R/W	0000h	**Data Count**: Determines the current packet size of the indexed endpoint FIFO.

10.6.5 DcBufferStatus register

This register is accessed using an index. The endpoint index must first be set before accessing this register for the corresponding endpoint. It reflects the status of the endpoint FIFO. Table 117 shows the bit allocation of the DcBufferStatus register.

Remark: This register is not applicable to the control endpoint.

Remark: For the endpoint IN data transfer, firmware must ensure a 200 ns delay between writing of the data packet and reading the DcBufferStatus register. For the endpoint OUT data transfer, firmware must also ensure a 200 ns delay between the reception of the endpoint interrupt and reading the DcBufferStatus register. For more information, refer to Ref. 10 “ISP1760/1 Frequently Asked Questions (AN10054)”.

Table 117. DcBufferStatus - Device Controller Buffer Status register (address 021Eh) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved[1]						BUF1	BUF0
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R	R

[1] The reserved bits should always be written with the reset value.

Table 118. DcBufferStatus - Device Controller Buffer Status register (address 021Eh) bit description

Bit	Symbol	Description
7 to 2	-	reserved
1 to 0	BUF[1:0]	**Buffer**: **00 —** The buffers are not filled. **01 —** One of the buffers is filled. **10 —** One of the buffers is filled. **11 —** Both the buffers are filled.

10.6.6 Endpoint MaxPacketSize register

This register determines the maximum packet size for all endpoints, except control 0. The register contains 2 bytes, and the bit allocation is given in Table 119.

Each time the register is written, the Buffer Length register of the corresponding endpoint is re-initialized to the FFOSZ field value. NTRANS bits control the number of transactions allowed in a single microframe for high-speed isochronous and interrupt endpoints only.

Table 119. Endpoint MaxPacketSize register (address 0204h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]			NTRANS[1:0]		FFOSZ[10:8]		
Reset	0	0	0	0	0	0	0	0
Bus 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	FFOSZ[7:0]							
Reset	0	0	0	0	0	0	0	0
Bus 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

[1] The reserved bits should always be written with the reset value.

Table 120. Endpoint MaxPacketSize register (address 0204h) bit description

Bit	Symbol	Description
15 to 13	-	reserved
12 to 11	NTRANS[1:0]	**Number of Transactions**: HS mode only. **00 —** One packet per microframe **01 —** Two packets per microframe **10 —** Three packets per microframe **11 —** reserved These bits are applicable only for isochronous or interrupt transactions.
10 to 0	FFOSZ[10:0]	**FIFO Size**: Sets the FIFO size, in bytes, for the indexed endpoint. Applies to both high-speed and full-speed operations.

The ISP1761 supports all the transfers given in Ref. 1 “Universal Serial Bus Specification Rev. 2.0”.

Each programmable FIFO can be independently configured using its Endpoint MaxPacketSize register (R/W: 04h), but the total physical size of all enabled endpoints (IN plus OUT), including set-up token buffer, control IN and control OUT, must not exceed 8192 bytes.

10.6.7 Endpoint Type register

This register sets the endpoint type of the indexed endpoint: isochronous, bulk or interrupt. It also serves to enable the endpoint and configure it for double buffering. Automatic generation of an empty packet for a zero-length TX buffer can be disabled using bit NOEMPKT. The register contains 2 bytes. See Table 121.

Table 121. Endpoint Type register (address 0208h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Bus 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[1]			NOEMPKT	ENABLE	DBLBUF	ENDPTYP[1:0]	
Reset	0	0	0	0	0	0	0	0
Bus 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

[1] The reserved bits should always be written with the reset value.

Table 122. Endpoint Type register (address 0208h) bit description

Bit	Symbol	Description
15 to 5	-	reserved
4	NOEMPKT	**No Empty Packet**: Logic 0 causes the ISP1761 to return a null length packet for the IN token after the DMA IN transfer is complete. Set to logic 1 to disable the generation of the null length packet.
3	ENABLE	**Endpoint Enable**: Logic 1 enables the FIFO of the indexed endpoint. The memory size is allocated as specified in the Endpoint MaxPacketSize register. Logic 0 disables the FIFO. **Remark:** Stalling a data endpoint will confuse the Data Toggle bit on the stalled endpoint because the internal logic picks up from where it has stalled. Therefore, the Data Toggle bit must be reset by disabling and re-enabling the corresponding endpoint (by setting bit ENABLE to logic 0, followed by logic 1 in the Endpoint Type register) to reset the PID.
2	DBLBUF	**Double Buffering**: Logic 1 enables double buffering for the indexed endpoint. Logic 0 disables double buffering.
1 to 0	ENDPTYP[1:0]	**Endpoint Type**: These bits select the endpoint type as follows. **00 —** Not used **01 —** Isochronous **10 —** Bulk **11 —** Interrupt

10.7 DMA registers

The Generic DMA (GDMA) transfer can be done by writing the proper opcode in the DMA Command register. The control bits are given in Table 123.

GDMA read or write (opcode = 00h/01h) for Generic DMA slave mode

The GDMA (slave) can operate in counter mode. RD_N and WR_N are DMA data strobe signals. These signals are also used as data strobe signals during the PIO access. An internal multiplex will redirect these signals to the DMA Controller for the DMA transfer or to registers for the PIO access.

In counter mode, the DIS_XFER_CNT bit in the DcDMAConfiguration register must be set to logic 0. The DMA Transfer Counter register must be programmed before any DMA command is issued. The DMA transfer counter is set by writing from the LSByte to the MSByte (address: 234h to 237h). The DMA transfer count is internally updated only after the MSByte is written. Once the DMA transfer is started, the transfer counter starts decrementing and on reaching 0, the DMA_XFER_OK bit is set and an interrupt is generated by the ISP1761.

The DMA transfer starts once the DMA command is issued. Any of the following three ways will terminate this DMA transfer:

- Detecting an internal EOT (short packet on an OUT token)
- Resetting the DMA
- GDMA stop command

There are two interrupts that are programmable to differentiate the method of DMA termination: the INT_EOT and DMA_XFER_OK bits in the DMA Interrupt Reason register. For details, see Table 135.

Table 123. Control bits for GDMA read or write (opcode = 00h/01h)

Control bits	Description	Reference
Mode register		
DMACLKON	Set DMACLKON to logic 1	Table 101
DcDMAConfiguration register		
MODE[1:0]	Determines the active read or write data strobe signals	Table 130
WIDTH	Selects the DMA bus width: 16-bit or 32-bit	
DIS_XFER_CNT	Disables the use of the DMA Transfer Counter	
DMA Hardware register		
DACK_POL, DREQ_POL	Select the polarity of the DMA handshake signals	Table 132

Remark: The DMA bus defaults to 3-state, until a DMA command is executed. All the other control signals are not 3-state.

10.7.1 DMA Command register

The DMA Command register is a 1-byte register (for bit allocation, see Table 124) that initiates all DMA transfer activities on the DMA controller. The register is write-only: reading it will return FFh.

Remark: The DMA bus will be in 3-state until a DMA command is executed.

Table 124. DMA Command register (address 0230h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	DMA_CMD[7:0]							
Reset	1	1	1	1	1	1	1	1
Bus reset	1	1	1	1	1	1	1	1
Access	W	W	W	W	W	W	W	W

Table 125. DMA Command register (address 0230h) bit description

Bit	Symbol	Description
7 to 0	DMA_CMD[7:0]	DMA command code; see Table 126.

Table 126. DMA commands

Code	Name	Description
00h	GDMA Read	**Generic DMA IN token transfer**: Data is transferred from the external DMA bus to the internal buffer.
01h	GDMA Write	**Generic DMA OUT token transfer**: Data is transferred from the internal buffer to the external DMA bus.
02h to 0Dh	-	reserved
0Eh	Validate Buffer	**Validate Buffer (for debugging only)**: Request from the microcontroller to validate the endpoint buffer, following a DMA-to-USB data transfer.
0Fh	Clear Buffer	**Clear Buffer**: Request from the microcontroller to clear the endpoint buffer, after a DMA-to-USB data transfer. Logic 1 clears the TX buffer of the indexed endpoint; the RX buffer is not affected. The TX buffer is automatically cleared once data is sent on the USB bus. This bit is set only when it is necessary to forcefully clear the buffer. **Remark:** If using double buffer, to clear both the buffers issue the Clear Buffer command two times, that is, set and clear this bit two times.
10h	-	reserved
11h	Reset DMA	**Reset DMA**: Initializes the DMA core to its power-on reset state. **Remark:** When the DMA core is reset during the Reset DMA command, the DREQ, DACK, RD_N and WR_N handshake pins will temporarily be asserted. This can confuse the external DMA controller. To prevent this, start the external DMA controller **only after** the DMA reset.
12h	-	reserved
13h	GDMA Stop	**GDMA stop**: This command stops the GDMA data transfer. Any data in the OUT endpoint that is not transferred by the DMA will remain in the buffer. The FIFO data for the IN endpoint will be written to the endpoint buffer. An interrupt bit will be set to indicate that the DMA Stop command is complete.
14h to FFh	-	reserved

10.7.2 DMA Transfer Counter register

This 4 bytes register sets up the total byte count for a DMA transfer (DMACR). It indicates the remaining number of bytes left for transfer. The bit allocation is given in Table 127.

For IN endpoint — Because there is a FIFO in the ISP1761 DMA controller, some data may remain in the FIFO during the DMA transfer. The maximum FIFO size is 8 bytes, and the maximum delay time for the data to be shifted to endpoint buffer is 60 ns.

For OUT endpoint — Data will not be cleared for the endpoint buffer, until all the data has been read from the DMA FIFO.

Table 127. DMA Transfer Counter register (address 0234h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	DMACR4 = DMACR[31:24]							
Reset	0	0	0	0	0	0	0	0
Bus 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	DMACR3 = DMACR[23:16]							
Reset	0	0	0	0	0	0	0	0
Bus 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	DMACR2 = DMACR[15:8]							
Reset	0	0	0	0	0	0	0	0
Bus 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	DMACR1 = DMACR[7:0]							
Reset	0	0	0	0	0	0	0	0
Bus 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 128. DMA Transfer Counter register (address 0234h) bit description

Bit	Symbol	Description
31 to 24	DMACR4, DMACR[31:24]	**DMA Counter 4**: DMA transfer counter byte 4
23 to 16	DMACR3, DMACR[23:16]	**DMA Counter 3**: DMA transfer counter byte 3
15 to 8	DMACR2, DMACR[15:8]	**DMA Counter 2**: DMA transfer counter byte 2
7 to 0	DMACR1, DMACR[7:0]	**DMA Counter 1**: DMA transfer counter byte 1

10.7.3 DcDMAConfiguration register

This register defines the DMA configuration for GDMA mode. The DcDMAConfiguration register consists of 2 bytes. The bit allocation is given in Table 129.

Table 129. DcDMAConfiguration - Device Controller Direct Memory Access Configuration register (address 0238h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Bus 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	DIS_ XFER_CNT	reserved[1]			MODE[1:0]		reserved	WIDTH
Reset	0	0	0	0	0	0	0	1
Bus reset	0	0	0	0	0	0	0	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 130. DcDMAConfiguration - Device Controller Direct Memory Access Configuration register (address 0238h) bit description

Bit	Symbol	Description
15 to 8	-	reserved
7	DIS_XFER_CNT	**Disable Transfer Counter**: Write logic 0 to perform DMA operation. Logic 1 disables the DMA transfer counter (see Table 127).
6 to 4	-	reserved
3 to 2	MODE[1:0]	**Mode**: **00 —** WR_N slave strobes data from the DMA bus into the ISP1761; RD_N slave puts data from the ISP1761 on the DMA bus **01, 10, 11 —** reserved
1	-	reserved
0	WIDTH	**Width**: This bit selects the DMA bus width for GDMA. **0 —** 32-bit data bus **1 —** 16-bit data bus

10.7.4 DMA Hardware register

The DMA Hardware register consists of 1 byte. The bit allocation is shown in Table 131.

This register determines the polarity of bus control signals (DACK and DREQ).

Table 131. DMA Hardware register (address 023Ch) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved[1]				DACK_ POL	DREQ_ POL	reserved[1]	
Reset	0	0	0	0	0	1	0	0
Bus reset	0	0	0	0	0	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 132. DMA Hardware register (address 023Ch) bit description

Bit	Symbol	Description
7 to 4	-	reserved
3	DACK_POL	**DACK Polarity**: Selects the DMA acknowledgment polarity. **0 —** DACK is active LOW **1 —** DACK is active HIGH
2	DREQ_POL	**DREQ Polarity**: Selects the DMA request polarity. **0 —** DREQ is active LOW **1 —** DREQ is active HIGH
1 to 0	-	reserved

10.7.5 DMA Interrupt Reason register

This 2-byte register shows the source(s) of DMA interrupt. Each bit is refreshed after a DMA command is executed. An interrupt source is cleared by writing logic 1 to the corresponding bit. On detecting the interrupt, the external microprocessor must read the DMA Interrupt Reason register and mask it with the corresponding bits in the DMA Interrupt Enable register to determine the source of the interrupt.

The bit allocation is given in Table 133.

Table 133. DMA Interrupt Reason register (address 0250h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved			GDMA_STOP	reserved	INT_EOT	reserved[1]	DMA_XFER_OK
Reset	0	0	0	0	0	0	0	0
Bus 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[1]							
Reset	0	0	0	0	0	0	0	0
Bus 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

[1] The reserved bits should always be written with the reset value.

Table 134. DMA Interrupt Reason register (address 0250h) bit description

Bit	Symbol	Description
15 to 13	-	reserved
12	GDMA_STOP	**GDMA Stop**: When the GDMA_STOP command is issued to DMA Command registers, it means that the DMA transfer has successfully terminated.
11	-	reserved
10	INT_EOT	**Internal EOT**: Logic 1 indicates that an internal EOT is detected; see Table 135.

Table 134. DMA Interrupt Reason register (address 0250h) bit description *…continued*

Bit	Symbol	Description
9	-	reserved
8	DMA_XFER_OK	**DMA Transfer OK**: Logic 1 indicates that the DMA transfer has been completed, that is, DMA transfer counter has become zero.
7 to 0	-	reserved

Table 135. Internal EOT-functional relation with the DMA_XFER_OK bit

INT_EOT	DMA_XFER_OK	Description
1	0	During the DMA transfer, there is a premature termination with short packet.
1	1	DMA transfer is completed with a short packet and the DMA transfer counter has reached 0.
0	1	DMA transfer is completed without any short packet and the DMA transfer counter has reached 0.

10.7.6 DMA Interrupt Enable register

This 2 bytes register controls the interrupt generation of the source bits in the DMA Interrupt Reason register. The bit allocation is given in Table 136. The bit description is given in Table 134.

Logic 1 enables the interrupt generation. The values after a (bus) reset are logic 0 (disabled).

Table 136. DMA Interrupt Enable register (address 0254h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]			IE_GDMA_STOP	reserved[1]	IE_INT_EOT	reserved[1]	IE_DMA_XFER_OK
Reset	0	0	0	0	0	0	0	0
Bus 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[1]							
Reset	0	0	0	0	0	0	0	0
Bus 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

[1] The reserved bits should always be written with the reset value.

10.7.7 DMA Endpoint register

This 1 byte register selects a USB endpoint FIFO as the source or destination for DMA transfers. The bit allocation is given in Table 137.

Table 137. DMA Endpoint register (address 0258h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	reserved[1]				EPIDX[2:0]			DMADIR

Table 137. DMA Endpoint register (address 0258h) bit allocation *…continued*

Bit	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0
Bus 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

[1] The reserved bits should always be written with the reset value.

Table 138. DMA Endpoint register (address 0258h) bit description

Bit	Symbol	Description
7 to 4	-	reserved
3 to 1	EPIDX[2:0]	Selects the indicated endpoint for DMA access
0	DMADIR	**DMA Direction**: **0 —** Selects the RX/OUT FIFO for DMA write transfers **1 —** Selects the TX/IN FIFO for DMA read transfers

The DMA Endpoint register must not reference the endpoint that is indexed by the Endpoint Index register (022Ch) at any time. Doing so will result in data corruption. Therefore, if the DMA Endpoint register is unused, point it to an unused endpoint. If the DMA Endpoint register, however, is pointed to an active endpoint, the firmware must not reference the same endpoint on the Endpoint Index register.

10.7.8 DMA Burst Counter register

The bit allocation of the register is given in Table 139.

Table 139. DMA Burst Counter register (address 0264h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]			BURSTCOUNTER[12:8]				
Reset	0	0	0	0	0	0	0	0
Bus 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	BURSTCOUNTER[7:0]							
Reset	0	0	0	0	0	1	0	0
Bus reset	0	0	0	0	0	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 140. DMA Burst Counter register (address 0264h) bit description

Bit	Symbol	Description
15 to 13	-	reserved
12 to 0	BURST COUNTER[12:0]	**Burst Counter**: This register defines the burst length. The counter must be programmed to be a multiple of two in 16-bit mode and four in 32-bit mode. The value of the burst counter must be programmed so that the buffer counter is a factor of the burst counter. In 16-bit mode, DREQ will drop at every DMA read or write cycle when the burst counter equals 2. In 32-bit mode, DREQ will drop at every DMA read or write cycle when the burst counter equals 4.

10.8 General registers

10.8.1 DcInterrupt register

The DcInterrupt register consists of 4 bytes. The bit allocation is given in Table 141.

When a bit is set in the DcInterrupt register, it indicates that the hardware condition for an interrupt has occurred. When the DcInterrupt register content is non-zero, the INT output will be asserted. On detecting the interrupt, the external microprocessor must read the DcInterrupt register to determine the source of the interrupt.

Each endpoint buffer has a dedicated interrupt bit (EPnTX, EPnRX). In addition, various bus states can generate an interrupt: resume, suspend, pseudo SOF, SOF and bus reset. The DMA controller has only one interrupt bit: the source for a DMA interrupt is shown in the DMA Interrupt Reason register.

Each interrupt bit can individually be cleared by writing logic 1. The DMA Interrupt bit can be cleared by writing logic 1 to the related interrupt source bit in the DMA Interrupt Reason register and writing logic 1 to the DMA bit of the DcInterrupt register.

Table 141. DcInterrupt - Device Controller Interrupt register (address 0218h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]						EP7TX	EP7RX
Reset	0	0	0	0	0	0	0	0
Bus 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	EP6TX	EP6RX	EP5TX	EP5RX	EP4TX	EP4RX	EP3TX	EP3RX
Reset	0	0	0	0	0	0	0	0
Bus 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	EP2TX	EP2RX	EP1TX	EP1RX	EP0TX	EP0RX	reserved[1]	EP0SETUP
Reset	0	0	0	0	0	0	0	0
Bus 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	VBUS	DMA	HS_STAT	RESUME	SUSP	PSOF	SOF	BRESET
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	unchanged
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 142. DcInterrupt - Device Controller Interrupt register (address 0218h) bit description

Bit	Symbol	Description
31 to 26	-	reserved
25	EP7TX	Logic 1 indicates the endpoint 7 TX buffer as interrupt source.
24	EP7RX	Logic 1 indicates the endpoint 7 RX buffer as interrupt source.
23	EP6TX	Logic 1 indicates the endpoint 6 TX buffer as interrupt source.
22	EP6RX	Logic 1 indicates the endpoint 6 RX buffer as interrupt source.
21	EP5TX	Logic 1 indicates the endpoint 5 TX buffer as interrupt source.
20	EP5RX	Logic 1 indicates the endpoint 5 RX buffer as interrupt source.
19	EP4TX	Logic 1 indicates the endpoint 4 TX buffer as interrupt source.
18	EP4RX	Logic 1 indicates the endpoint 4 RX buffer as interrupt source.
17	EP3TX	Logic 1 indicates the endpoint 3 TX buffer as interrupt source.
16	EP3RX	Logic 1 indicates the endpoint 3 RX buffer as interrupt source.
15	EP2TX	Logic 1 indicates the endpoint 2 TX buffer as interrupt source.
14	EP2RX	Logic 1 indicates the endpoint 2 RX buffer as interrupt source.
13	EP1TX	Logic 1 indicates the endpoint 1 TX buffer as interrupt source.
12	EP1RX	Logic 1 indicates the endpoint 1 RX buffer as interrupt source.
11	EP0TX	Logic 1 indicates the endpoint 0 data TX buffer as interrupt source.
10	EP0RX	Logic 1 indicates the endpoint 0 data RX buffer as interrupt source.
9	-	reserved
8	EP0SETUP	Logic 1 indicates that a SETUP token was received on endpoint 0.
7	VBUS	Logic 1 indicates a transition from LOW to HIGH on V_{BUS}. When implementing a pure host or peripheral, the OTG_DISABLE bit in the OTG Control register (374h) must be set to logic 1 so that the VBUS bit is updated with the correct value.
6	DMA	**DMA status**: Logic 1 indicates a change in the DMA Interrupt Reason register.
5	HS_STAT	**High-Speed Status**: Logic 1 indicates a change from full-speed to high-speed mode (HS connection). This bit is not set when the system goes into the full-speed suspend.
4	RESUME	**Resume status**: Logic 1 indicates that a status change from suspend to resume (active) was detected.
3	SUSP	**Suspend status**: Logic 1 indicates that a status change from active to suspend was detected on the bus.

Table 142. DcInterrupt - Device Controller Interrupt register (address 0218h) bit description *...continued*

Bit	Symbol	Description
2	PSOF	**Pseudo SOF interrupt**: Logic 1 indicates that a pseudo SOF or µSOF was received. Pseudo SOF is an internally generated clock signal (full-speed: 1 ms period, high-speed: 125 µs period) that is not synchronized to the USB bus SOF or µSOF.
1	SOF	**SOF interrupt**: Logic 1 indicates that a SOF or µSOF was received.
0	BRESET	**Bus Reset**: Logic 1 indicates that a USB bus reset was detected.

10.8.2 DcChipID register

This read-only register contains the chip identification and hardware version numbers. The firmware must check this information to determine functions and features supported. The register contains 3 bytes, and the bit allocation is shown in Table 143.

Table 143. DcChipID - Device Controller Chip Identifier register (address 0270h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	CHIPID[31:0]	R	0015 8210h	**Chip ID**: This registers represents the hardware version number (0015h) and the chip ID (8210h) for the peripheral controller.

10.8.3 Frame Number register

This read-only register contains the frame number of the last successfully received Start-Of-Frame (SOF). The register contains 2 bytes, and the bit allocation is given in Table 144.

Table 144. Frame Number register (address 0274h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved		MICROSOF[2:0]			SOFR[10:8]		
Reset	0	0	0	0	0	0	0	0
Bus 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	SOFR[7:0]							
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

Table 145. Frame Number register (address 0274h) bit description

Bit	Symbol	Description
15 to 14	-	reserved
13 to 11	MICROSOF[2:0]	microframe number
10 to 0	SOFR[10:0]	frame number

10.8.4 DcScratch register

This 16-bit register can be used by the firmware to save and restore information. For example, the device status before it enters the suspend state; see Table 146.

Table 146. DcScratch - Device Controller Scratch register (address 0278h) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	SFIRH[7:0]							
Reset	0	0	0	0	0	0	0	0
Bus reset	unchanged							
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
Bus reset	unchanged							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 147. DcScratch - Device Controller Scratch register (address 0278h) bit description

Bit	Symbol	Description
15 to 8	SFIRH[7:0]	Scratch firmware information register (higher byte)
7 to 0	SFIRL[7:0]	Scratch firmware information register (lower byte)

10.8.5 Unlock Device register

To protect registers from getting corrupted when the ISP1761 goes into suspend, the write operation is disabled. In this case, when the chip resumes, the Unlock Device command must first be issued to this register before attempting to write to the rest of the registers. This is done by writing unlock code (AA37h) to this register. The bit allocation of the Unlock Device register is given in Table 148.

Table 148. Unlock Device register (address 027Ch) bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	ULCODE[15:8] = AAh							
Reset	not applicable							
Bus reset	not applicable							
Access	W	W	W	W	W	W	W	W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Symbol	ULCODE[7:0] = 37h							
Reset	not applicable							
Bus reset	not applicable							
Access	W	W	W	W	W	W	W	W

Table 149. Unlock Device register (address 027Ch) bit description

Bit	Symbol	Description
15 to 0	ULCODE[15:0]	**Unlock Code**: Writing data AA37h unlocks internal registers and FIFOs for writing, following a resume.

10.8.6 Interrupt Pulse Width register

Table 150 shows the bit description of the register.

Table 150. Interrupt Pulse Width register (address 0280h) bit description

Bit	Symbol	Access	Value	Description
15 to 0	INTR_PULSE_WIDTH[15:0]	R/W	001Eh	**Interrupt Pulse Width**: The interrupt signal pulse width is configurable while it is in pulse signaling mode. The minimum pulse width is 3.33 ns when this register is set to logic 1. The power-on reset value of 1Eh allows a pulse of 1 μs to be generated.

10.8.7 Test Mode register

This 1 byte register allows the firmware to set the DP and DM pins to predetermined states for testing purposes. The bit allocation is given in Table 151.

Remark: Only one bit can be set to logic 1 at a time.

Table 151. Test Mode register (address 0284h) bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	FORCEHS	reserved[1]		FORCEFS	PRBS	KSTATE	JSTATE	SE0_NAK
Reset	0	0	0	0	0	0	0	0
Bus reset	unchanged	0	0	unchanged	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[1] The reserved bits should always be written with the reset value.

Table 152. Test Mode register (address 0284h) bit description

Bit	Symbol	Description
7	FORCEHS	**Force High-Speed**: Logic 1[1] forces the hardware to high-speed mode only and disables the chirp detection logic.
6 to 5	-	reserved.
4	FORCEFS	**Force Full-Speed**: Logic 1[1] forces the physical layer to full-speed mode only and disables the chirp detection logic.
3	PRBS	Logic 1[2] sets pins DP and DM to toggle in a predetermined random pattern.
2	KSTATE	**K State**: Writing logic 1[2] sets the DP and DM pins to the K state.
1	JSTATE	**J State**: Writing logic 1[2] sets the DP and DM pins to the J state.
0	SE0_NAK	**SE0 NAK**: Writing logic 1[2] sets pins DP and DM to a high-speed quiescent state. The device only responds to a valid high-speed IN token with a NAK.

[1] Either FORCEHS or FORCEFS must be set at a time.

[2] Of the four bits, PRBS, KSTATE, JSTATE and SE0_NAK, only one bit must be set at a time.

11. Power consumption

Table 153. Power consumption

Number of ports working	I_{CC}
One port working (high-speed)	
V_{CC} = 5.0 V, $V_{CC(I/O)}$ = 3.3 V	90 mA
V_{CC} = 3.3 V, $V_{CC(I/O)}$ = 3.3 V	77 mA
V_{CC} = 5.0 V, $V_{CC(I/O)}$ = 1.8 V	82 mA
V_{CC} = 3.3 V, $V_{CC(I/O)}$ = 1.8 V	77 mA
Two ports working (high-speed)	
V_{CC} = 5.0 V, $V_{CC(I/O)}$ = 3.3 V	110 mA
V_{CC} = 3.3 V, $V_{CC(I/O)}$ = 3.3 V	97 mA
V_{CC} = 5.0 V, $V_{CC(I/O)}$ = 1.8 V	102 mA
V_{CC} = 3.3 V, $V_{CC(I/O)}$ = 1.8 V	97 mA
Three ports working (high-speed)	
V_{CC} = 5.0 V, $V_{CC(I/O)}$ = 3.3 V	130 mA
V_{CC} = 3.3 V, $V_{CC(I/O)}$ = 3.3 V	117 mA
V_{CC} = 5.0 V, $V_{CC(I/O)}$ = 1.8 V	122 mA
V_{CC} = 3.3 V, $V_{CC(I/O)}$ = 1.8 V	117 mA

The idle operating current, I_{CC}, that is, when the ISP1761 is in operational mode, initialized and without any devices connected, is 70 mA. The additional current consumption on I_{CC} is below 1 mA per port in the case of full-speed and low-speed devices.

Deep-sleep suspend mode ensures the lowest power consumption when V_{CC} is always supplied to the ISP1761. In this case, the suspend current, $I_{CC(susp)}$, is typically about 150 μA at ambient temperature of +25 °C. The suspend current may increase if the ambient temperature increases. For details, see Section 7.6.

In hybrid mode, when V_{CC} is disconnected $I_{CC(I/O)}$ will generally be below 100 μA. The average value is 60 μA to 70 μA.

Under the condition of constant read and write accesses occurring on the 32-bit data bus, the maximum $I_{CC(I/O)}$ drawn from $V_{CC(I/O)}$ is measured as 25 mA, when the NXP ISP1761 evaluation board is connected to a BSQUARE PXA255 development platform. This current will vary depending on the platform because of the different access timing, the type of data patterns written on the data bus, and loading on the data bus.

12. Limiting values

Table 154. Limiting values

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

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{CC(I/O)}$	input/output supply voltage		–0.5	+4.6	V
$V_{CC(5V0)}$	supply voltage		–0.5	+5.6	V
$V_{CC(C\_IN)}$	supply voltage		-	4.6	V
I_{lu}	latch-up current	$V_I < 0$ V or $V_I > V_{CC}$	-	100	mA
V_{esd}	electrostatic discharge voltage	$I_{LI} < 1$ µA (all pins)	–4000	+4000	V
T_{stg}	storage temperature		–40	+125	°C

13. Recommended operating conditions

Table 155. Recommended operating conditions

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$V_{CC(I/O)}$	input/output supply voltage	$V_{CC(I/O)}$ = 3.3 V		3.0	3.3	3.6	V
		$V_{CC(I/O)}$ = 1.8 V		1.65	1.8	1.95	V
$V_{CC(5V0)}$	supply voltage			3	-	5.5	V
$V_{CC(C\_IN)}$	supply voltage		[1]	3.15	-	3.6	V
T_{amb}	ambient temperature			–40	-	+85	°C
T_j	junction temperature			–40	-	+125	°C
$I_{CC(susp)}$	suspend supply current	T_{amb} = 25 °C; $V_{CC(5V0)}$ = 3.3 V	[2]	-	150	-	µA

[1] For details, see Figure 17 and Figure 18.

[2] Deep sleep suspend current.

14. Static characteristics

Table 156. Static characteristics: digital pins

All digital pins[1], except pins ID, PSW1_N, PSW2_N, PSW3_N and BAT_ON_N.
T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V						
V_{IH}	HIGH-level input voltage		1.2	-	-	V
V_{IL}	LOW-level input voltage		-	-	0.5	V
V_{hys}	hysteresis voltage		0.4	-	0.7	V
V_{OL}	LOW-level output voltage	I_{OL} = 3 mA	-	-	$0.22V_{CC(I/O)}$	V
V_{OH}	HIGH-level output voltage		$0.8V_{CC(I/O)}$	-	-	V
I_{LI}	input leakage current	V_I = 0 V to $V_{CC(I/O)}$	-	-	1	µA
C_{in}	input capacitance		-	2.75	-	pF
$V_{CC(I/O)}$ = 3.0 V to 3.6 V						
V_{IH}	HIGH-level input voltage		2.0	-	-	V
V_{IL}	LOW-level input voltage		-	-	0.8	V
V_{hys}	hysteresis voltage		0.4	-	0.7	V
V_{OL}	LOW-level output voltage	I_{OL} = 3 mA	-	-	0.4	V
V_{OH}	HIGH-level output voltage		2.4	-	-	V
I_{LI}	input leakage current	V_I = 0 V to $V_{CC(I/O)}$	-	-	1	µA
C_{in}	input capacitance		-	2.75	-	pF

[1] Includes OC1_N/V_{BUS}, OC2_N and OC3_N when used as digital overcurrent pins.

Table 157. Static characteristics: PSW1_N, PSW2_N, PSW3_N

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{OL}	LOW-level output voltage	I_{OL} = 8 mA; pull-up to $V_{CC(5V0)}$	-	-	0.4	V
V_{OH}	HIGH-level output voltage	pull-up to $V_{CC(I/O)}$	-	$V_{CC(I/O)}$	-	V

Table 158. Static characteristics: USB interface block (pins DM1 to DM3 and DP1 to DP3)

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Input levels for high-speed						
V_{HSSQ}	high-speed squelch detection threshold voltage (differential signal amplitude)	squelch detected	-	-	100	mV
		no squelch detected	150	-	-	mV
V_{HSDSC}	high-speed disconnect detection threshold voltage (differential signal amplitude)	disconnect detected	625	-	-	mV
		disconnect not detected	-	-	525	mV
V_{HSCM}	high-speed data signaling common mode voltage range (guideline for receiver)		−50	-	+500	mV

Table 158. Static characteristics: USB interface block (pins DM1 to DM3 and DP1 to DP3) *…continued*

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Output levels for high-speed						
V_{HSOI}	high-speed idle level voltage		−10	-	+10	mV
V_{HSOH}	high-speed data signaling HIGH-level voltage		360	-	440	mV
V_{HSOL}	high-speed data signaling LOW-level voltage		−10	-	+10	mV
V_{CHIRPJ}	chirp J level (differential voltage)		700[1]	-	1100	mV
V_{CHIRPK}	chirp K level (differential voltage)		−900[1]	-	−500	mV
Input levels for full-speed and low-speed						
V_{IH}	HIGH-level input voltage	drive	2.0	-	-	V
V_{IHZ}	HIGH-level input voltage (floating)		2.7	-	3.6	V
V_{IL}	LOW-level input voltage		-	-	0.8	V
V_{DI}	differential input sensitivity	$\lvert V_{DP} - V_{DM} \rvert$	0.2	-	-	V
V_{CM}	differential common mode voltage range		0.8	-	2.5	V
Output levels for full-speed and low-speed						
V_{OH}	HIGH-level output voltage		2.8	-	3.6	V
V_{OL}	LOW-level output voltage		0	-	0.3	V
V_{OSE1}	SE1 output voltage		0.8	-	-	V
V_{CRS}	output signal crossover voltage		1.3	-	2.0	V

[1] The HS termination resistor is disabled, and the pull-up resistor is connected. Only during reset, when both the hub and the device are capable of the high-speed operation.

Table 159. Static characteristics: REF5V

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	HIGH-level input voltage		-	5	-	V

Table 160. Static characteristics: V_{BUS} comparators

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min[1]	Typ	Max[2]	Unit
$V_{A\_VBUS\_VLD}$	A-device V_{BUS} valid voltage		4.4	4.5	4.6	V
$V_{B\_SESS\_VLD}$	B-device session valid voltage	for A-device and B-device	0.8	1.6	2.0	V
$V_{hys(B\_SESS\_VLD)}$	B-device session valid hysteresis voltage		70	150	210	mV
$V_{B\_SESS\_END}$	B-device session end voltage		0.2	0.5	0.8	V

[1] Minimum trigger voltage at extreme low temperature (−40 °C).

[2] Minimum trigger voltage at extreme high temperature (+85 °C).

Table 161. Static characteristics: V_{BUS} resistors

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$R_{UP(VBUS)}$	pull-up resistance on pin V_{BUS}	connect to REG3V3 when VBUS_CHRG = 1	281	680	-	Ω
$R_{DN(VBUS)}$	pull-down resistance on pin V_{BUS}	connect to ground when VBUS_DISCHRG = 1	656	800	-	Ω
$R_{I(idle)(VBUS)(A)}$	idle input resistance on pin V_{BUS} (A-device)	ID pin LOW	40	58.5	100	kΩ
$R_{I(idle)(VBUS)(B)}$	idle input resistance on pin V_{BUS} (B-device)	ID pin HIGH	-	197	280	kΩ

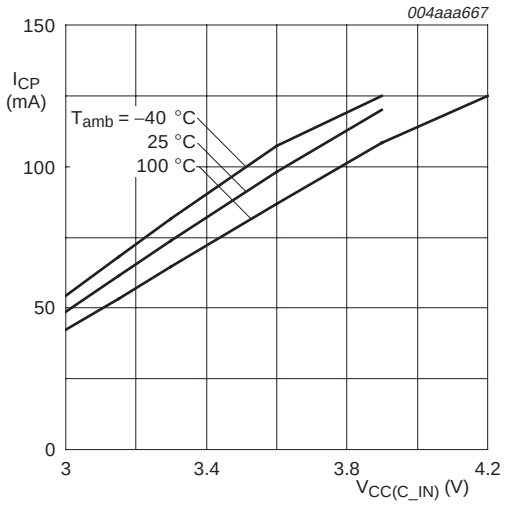


Fig 17. Charge pump current versus voltage at various temperatures (worst case)

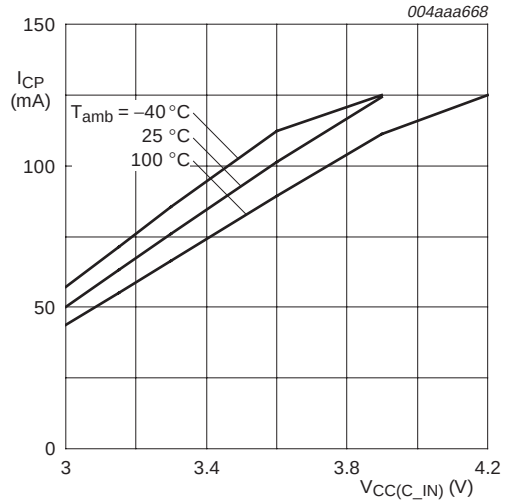


Fig 18. Charge pump current versus voltage at various temperatures (typical case)

15. Dynamic characteristics

Table 162. Dynamic characteristics: system clock timing

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Crystal oscillator						
f_{clk}	clock frequency	crystal[1] [2]	-	12	-	MHz
		oscillator	-	12	-	MHz
External clock input						
J	external clock jitter		-	-	500	ps
δ	clock duty cycle		-	50	-	%
$V_{i(XTAL1)}$	input voltage on pin XTAL1		-	$V_{CC(I/O)}$	-	V
t_r	rise time		-	-	3	ns
t_f	fall time		-	-	3	ns

[1] Recommended values for external capacitors when using a crystal are 22 pF to 27 pF.

[2] Recommended accuracy of the clock frequency is 50 ppm for the crystal and oscillator. The oscillator used depends on $V_{CC(I/O)}$.

Table 163. Dynamic characteristics: CPU interface block

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
SR	slew rate	standard load (rise, fall)	1	-	4	V/ns

Table 164. Dynamic characteristics: high-speed source electrical characteristics

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Driver characteristics						
t_{HSR}	rise time (10 % to 90 %)		500	-	-	ps
t_{HSF}	fall time (10 % to 90 %)		500	-	-	ps
Z_{HSDRV}	driver output impedance (which also serves as high-speed termination)	includes the R_S resistor	40.5	45	49.5	Ω
Clock timing						
t_{HSDRAT}	high-speed data rate		479.76	-	480.24	Mbit/s
t_{HSFRAM}	microframe interval		124.9375	-	125.0625	µs
t_{HSRFI}	consecutive microframe interval difference		1	-	four high-speed bit times	ns

Table 165. Dynamic characteristics: full-speed source electrical characteristics

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Driver characteristics						
t_{FR}	rise time	C_L = 50 pF; 10 % to 90 % of $\|V_{OH} - V_{OL}\|$	4	-	20	ns
t_{FF}	fall time	C_L = 50 pF; 90 % to 10 % of $\|V_{OH} - V_{OL}\|$	4	-	20	ns

Table 165. Dynamic characteristics: full-speed source electrical characteristics *…continued*

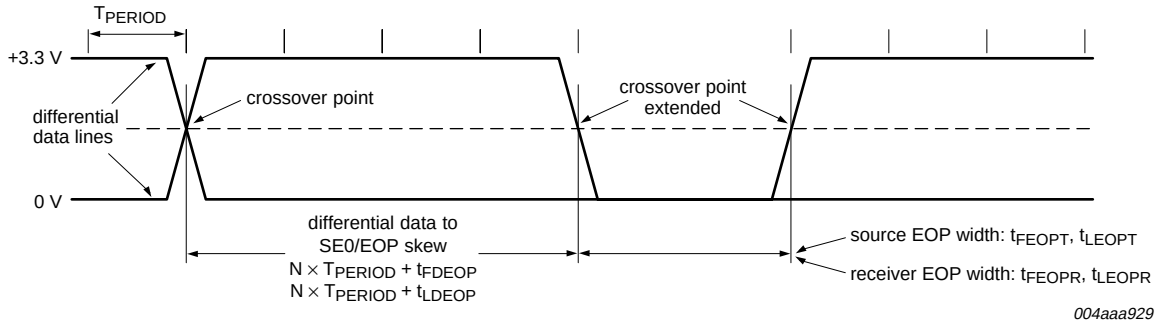
$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{FRFM}	differential rise and fall time matching		90	-	111.1	%
Z_{DRV}	driver output impedance for driver which is not high-speed capable		28	-	44	Ω
Data timing: see Figure 19						
t_{FDEOP}	source jitter for differential transition to SE0 transition	full-speed timing	−2	-	+5	ns
t_{FEOPT}	source SE0 interval of EOP		160	-	175	ns
t_{FEOPR}	receiver SE0 interval of EOP		82	-	-	ns
t_{LDEOP}	upstream facing port source jitter for differential transition to SE0 transition	low-speed timing	−40	-	+100	ns
t_{LEOPT}	source SE0 interval of EOP		1.25	-	1.5	µs
t_{LEOPR}	receiver SE0 interval of EOP		670	-	-	ns
t_{FST}	width of SE0 interval during differential transition		-	-	14	ns

Table 166. Dynamic characteristics: low-speed source electrical characteristics

$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Driver characteristics						
t_{LR}	transition time: rise time		75	-	300	ns
t_{LF}	transition time: fall time		75	-	300	ns
t_{LRFM}	rise and fall time matching		90	-	125	%



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

Fig 19. USB source differential data-to-EOP transition skew and EOP width

15.1 Host timing

15.1.1 PIO timing

15.1.1.1 Register or memory write

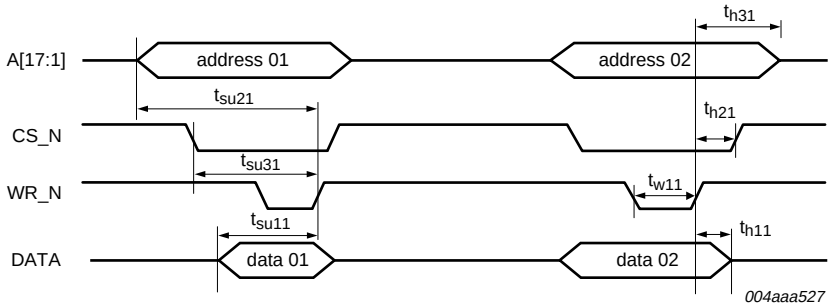


Fig 20. Register or memory write

Table 167. Register or memory write

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{h11}	data hold after WR_N HIGH	2	-	ns
t_{h21}	CS_N hold after WR_N HIGH	1	-	ns
t_{h31}	address hold after WR_N HIGH	2	-	ns
t_{w11}	WR_N pulse width	17	-	ns
t_{su11}	data set-up time before WR_N HIGH	5	-	ns
t_{su21}	address set-up time before WR_N HIGH	5	-	ns
t_{su31}	CS_N set-up time before WR_N HIGH	5	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{h11}	data hold after WR_N HIGH	2	-	ns
t_{h21}	CS_N hold after WR_N HIGH	1	-	ns
t_{h31}	address hold after WR_N HIGH	2	-	ns
t_{w11}	WR_N pulse width	17	-	ns
t_{su11}	data set-up time before WR_N HIGH	5	-	ns
t_{su21}	address set-up time before WR_N HIGH	5	-	ns
t_{su31}	CS_N set-up time before WR_N HIGH	5	-	ns

15.1.1.2 Register read

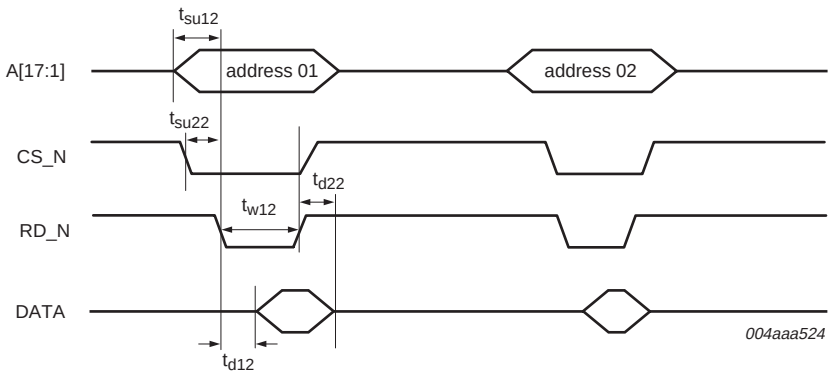


Fig 21. Register read

Table 168. Register read

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{su12}	address set-up time before RD_N LOW	0	-	ns
t_{su22}	CS_N set-up time before RD_N LOW	0	-	ns
t_{w12}	RD_N pulse width	> t_{d12}	-	ns
t_{d12}	data valid time after RD_N LOW	-	35	ns
t_{d22}	data valid time after RD_N HIGH	-	1	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{su12}	address set-up time before RD_N LOW	0	-	ns
t_{su22}	CS_N set-up time before RD_N LOW	0	-	ns
t_{w12}	RD_N pulse width	> t_{d12}	-	ns
t_{d12}	data valid time after RD_N LOW	-	22	ns
t_{d22}	data valid time after RD_N HIGH	-	1	ns

15.1.1.3 Register access

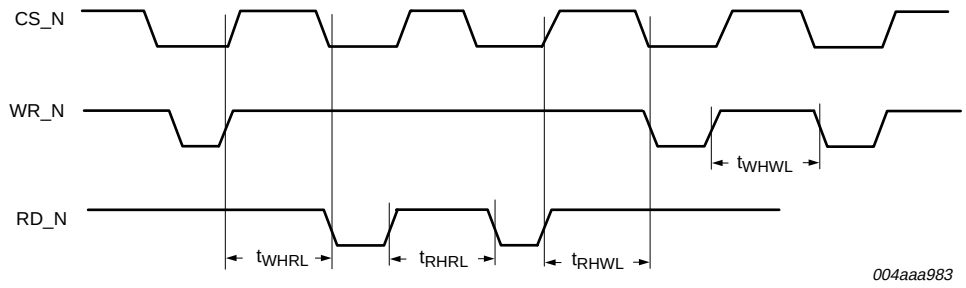


Fig 22. Register access

Table 169. Register access

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

Symbol	Parameter	Min	Max	Unit
t_{WHRL}	WR_N HIGH to RD_N LOW time	25[1]	-	ns
t_{RHRL}	RD_N HIGH to RD_N LOW time	25[1]	-	ns
t_{RHWL}	RD_N HIGH to WR_N LOW time	25	-	ns
t_{WHWL}	WR_N HIGH to WR_N LOW time	25[1]	-	ns

[1] For EHCI operational registers, minimum value is 195 ns.

15.1.1.4 Memory read

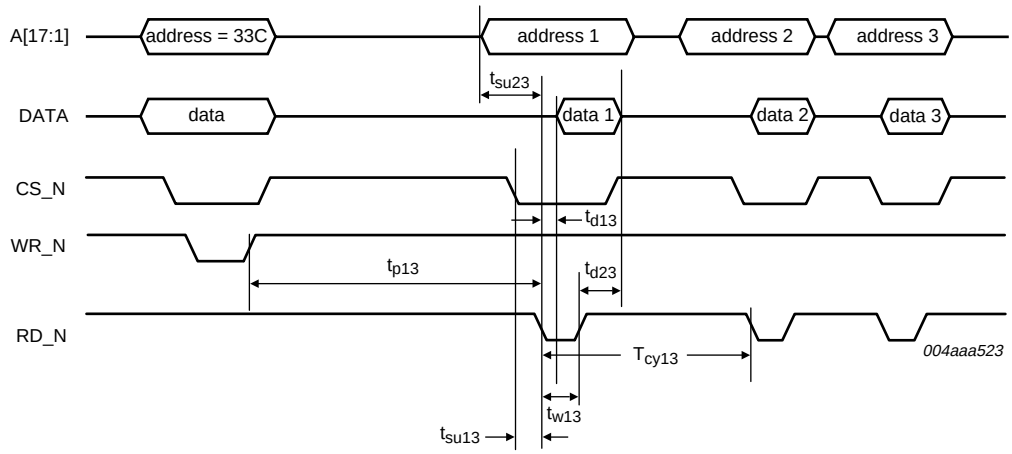


Fig 23. Memory read

Table 170. Memory read

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{p13}	initial pre-fetch time	90	-	ns
T_{cy13}	memory RD_N cycle time	40	-	ns
t_{d13}	data valid time after RD_N LOW	-	31	ns
t_{d23}	data available time after RD_N HIGH	-	1	ns
t_{w13}	RD_N pulse width	32	-	ns
t_{su13}	CS_N set-up time before RD_N LOW	0	-	ns
t_{su23}	address set-up time before RD_N LOW	0	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{p13}	initial pre-fetch time	90	-	ns
T_{cy13}	memory RD_N cycle time	36	-	ns
t_{d13}	data valid time after RD_N LOW	-	20	ns
t_{d23}	data available time after RD_N HIGH	-	1	ns

Table 170. Memory read *…continued*

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

Symbol	Parameter	Min	Max	Unit
t_{w13}	RD_N pulse width	21	-	ns
t_{su13}	CS_N set-up time before RD_N LOW	0	-	ns
t_{su23}	address set-up time before RD_N LOW	0	-	ns

15.1.2 DMA timing

In the following sections:

- Polarity of DACK is active HIGH
- Polarity of DREQ is active HIGH

15.1.2.1 Single cycle: DMA read

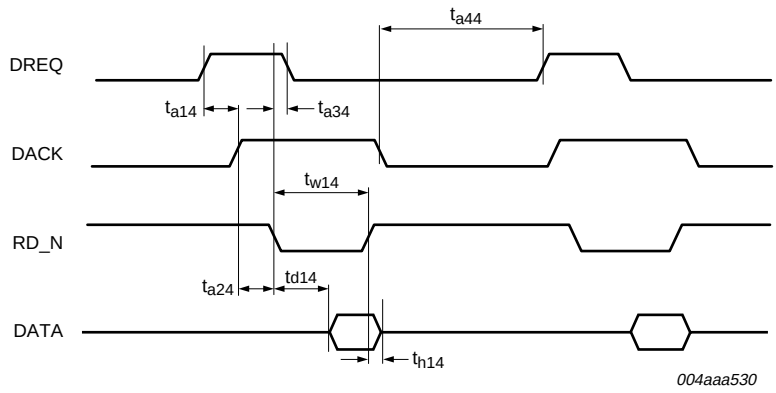


Fig 24. DMA read (single cycle)

Table 171. DMA read (single cycle)

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

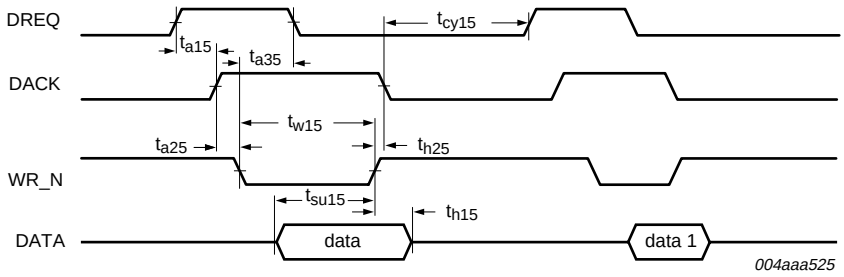
Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{a14}	DACK assertion time after DREQ assertion	0	-	ns
t_{a24}	RD_N assertion time after DACK assertion	0	-	ns
t_{d14}	data valid time after RD_N assertion	-	24	ns
t_{w14}	RD_N pulse width	> t_{d14}	-	ns
t_{a34}	DREQ de-assertion time after RD_N assertion	-	29	ns
t_{a44}	DACK de-assertion to next DREQ assertion time	-	56	ns
t_{h14}	data hold time after RD_N de-asserts	-	5	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{a14}	DACK assertion time after DREQ assertion	0	-	ns
t_{a24}	RD_N assertion time after DACK assertion	0	-	ns
t_{d14}	data valid time after RD_N assertion	-	20	ns
t_{w14}	RD_N pulse width	> t_{d14}	-	ns

Table 171. DMA read (single cycle) *...continued*

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

Symbol	Parameter	Min	Max	Unit
t_{a34}	DREQ de-assertion time after RD_N assertion	-	18	ns
t_{a44}	DACK de-assertion to next DREQ assertion time	-	56	ns
t_{h14}	data hold time after RD_N de-asserts	-	5	ns

15.1.2.2 Single cycle: DMA write



DREQ and DACK are active HIGH.

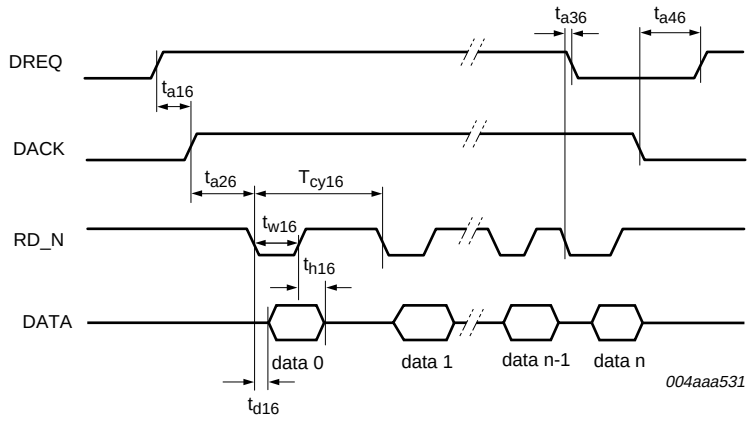
Fig 25. DMA write (single cycle)

Table 172. DMA write (single cycle)

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{a15}	DACK assertion time after DREQ assertion	0	-	ns
t_{a25}	WR_N assertion time after DACK assertion	1	-	ns
t_{h15}	data hold time after WR_N de-assertion	3	-	ns
t_{h25}	DACK hold time after WR_N de-assertion	0	-	ns
t_{su15}	data set-up time before WR_N de-assertion	5.5	-	ns
t_{a35}	DREQ de-assertion time after WR_N assertion	-	28	ns
t_{cy15}	last DACK strobe de-assertion to next DREQ assertion time	-	82	ns
t_{w15}	WR_N pulse width	22	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{a15}	DACK assertion time after DREQ assertion	0	-	ns
t_{a25}	WR_N assertion time after DACK assertion	1	-	ns
t_{h15}	data hold time after WR_N de-assertion	2	-	ns
t_{h25}	DACK hold time after WR_N de-assertion	0	-	ns
t_{su15}	data set-up time before WR_N de-assertion	5.5	-	ns
t_{a35}	DREQ de-assertion time after WR_N assertion	-	16	ns
t_{cy15}	last DACK strobe de-assertion to next DREQ assertion time	-	82	ns
t_{w15}	WR_N pulse width	22	-	ns

15.1.2.3 Multi-cycle: DMA read



DREQ and DACK are active HIGH.

Fig 26. DMA read (multi-cycle burst)

Table 173. DMA read (multi-cycle burst)

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{a16}	DACK assertion after DREQ assertion	0	-	ns
t_{a26}	RD_N assertion after DACK assertion	0	-	ns
t_{d16}	data valid time after RD_N assertion	-	31	ns
t_{w16}	RD_N pulse width	38	-	ns
T_{cy16}	read-to-read cycle time	46	-	ns
t_{a36}	DREQ de-assertion time after last burst RD_N de-assertion	-	30	ns
t_{a46}	DACK de-assertion to next DREQ assertion time	-	82	ns
t_{h16}	data hold time after RD_N de-asserts	-	5	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{a16}	DACK assertion after DREQ assertion	0	-	ns
t_{a26}	RD_N assertion after DACK assertion	0	-	ns
t_{d16}	data valid time after RD_N assertion	-	16	ns
t_{w16}	RD_N pulse width	17	-	ns
T_{cy16}	read-to-read cycle time	38	-	ns
t_{a36}	DREQ de-assertion time after last burst RD_N de-assertion	-	20	ns
t_{a46}	DACK de-assertion to next DREQ assertion time	-	82	ns
t_{h16}	data hold time after RD_N de-asserts	-	5	ns

15.1.2.4 Multi-cycle: DMA write

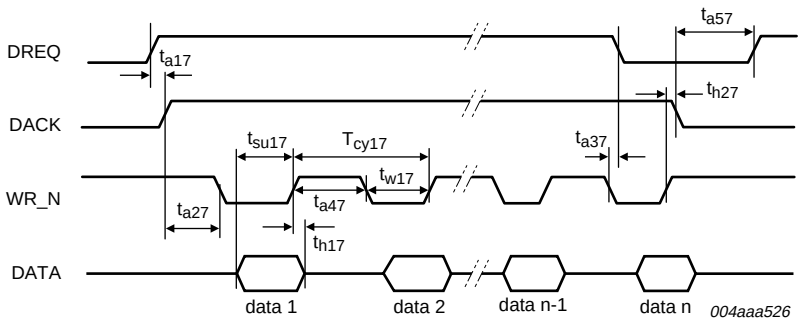


Fig 27. DMA write (multi-cycle burst)

Table 174. DMA write (multi-cycle burst)

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
T_{cy17}	DMA write cycle time	51	-	ns
t_{su17}	data set-up time before WR_N de-assertion	5	-	ns
t_{h17}	data hold time after WR_N de-assertion	2	-	ns
t_{a17}	DACK assertion time after DREQ assertion	0	-	ns
t_{a27}	WR_N assertion time after DACK assertion	2	-	ns
t_{a37}	DREQ de-assertion time at last strobe (WR_N) assertion	-	28	ns
t_{h27}	DACK hold time after WR_N de-assertion	0	-	ns
t_{a47}	strobe de-assertion to next strobe assertion time	34	-	ns
t_{w17}	WR_N pulse width	17	-	ns
t_{a57}	DACK de-assertion to next DREQ assertion time	-	82	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
T_{cy17}	DMA write cycle time	51	-	ns
t_{su17}	data set-up time before WR_N de-assertion	5	-	ns
t_{h17}	data hold time after WR_N de-assertion	2	-	ns
t_{a17}	DACK assertion time after DREQ assertion	0	-	ns
t_{a27}	WR_N assertion time after DACK assertion	1	-	ns
t_{a37}	DREQ de-assertion time at last strobe (WR_N) assertion	-	16	ns
t_{h27}	DACK hold time after WR_N de-assertion	0	-	ns
t_{a47}	strobe de-assertion to next strobe assertion time	34	-	ns
t_{w17}	WR_N pulse width	17	-	ns
t_{a57}	DACK de-assertion to next DREQ assertion time	-	82	ns

15.2 Peripheral timing

15.2.1 PIO timing

15.2.1.1 PIO register read or write

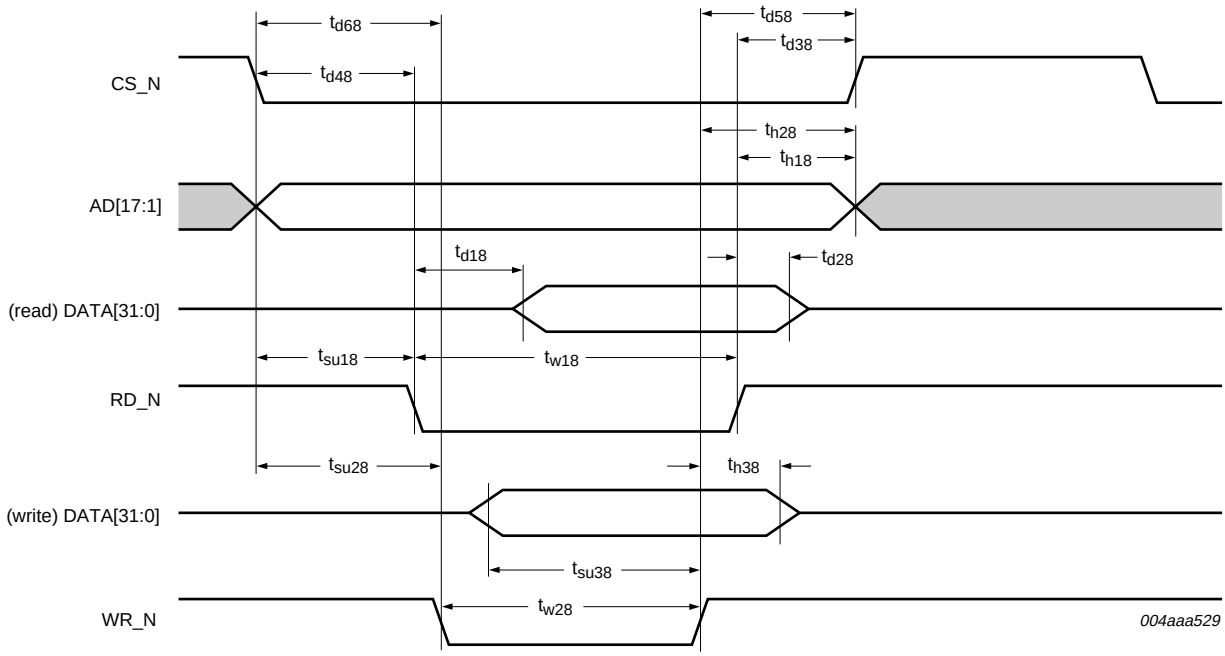


Fig 28. ISP1761 register access timing: separate address and data buses (8051 style)

Table 175. PIO register read or write

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
Reading				
t_{w18}	RD_N LOW pulse width	> t_{d18}	-	ns
t_{su18}	address set-up time before RD_N LOW	0	-	ns
t_{h18}	address hold time after RD_N HIGH	0	-	ns
t_{d18}	RD_N LOW to data valid delay	-	33	ns
t_{d28}	RD_N HIGH to data outputs 3-state delay	-	1	ns
t_{d38}	RD_N HIGH to CS_N HIGH delay	0	-	ns
t_{d48}	CS_N LOW to RD_N LOW delay	0	-	ns
Writing				
t_{w28}	WR_N LOW pulse width	15	-	ns
t_{su28}	address set-up time before WR_N LOW	0	-	ns
t_{h28}	address hold time after WR_N HIGH	0	-	ns
t_{su38}	data set-up time before WR_N HIGH	5	-	ns
t_{h38}	data hold time after WR_N HIGH	2	-	ns
t_{d58}	WR_N HIGH to CS_N HIGH delay	1	-	ns

Table 175. PIO register read or write *...continued*

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

Symbol	Parameter	Min	Max	Unit
t_{d68}	CS_N LOW to WR_N LOW delay	0	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
Reading				
t_{w18}	RD_N LOW pulse width	> t_{d18}	-	ns
t_{su18}	address set-up time before RD_N LOW	0	-	ns
t_{h18}	address hold time after RD_N HIGH	0	-	ns
t_{d18}	RD_N LOW to data valid delay	-	21	ns
t_{d28}	RD_N HIGH to data outputs 3-state delay	0	1	ns
t_{d38}	RD_N HIGH to CS_N HIGH delay	0	-	ns
t_{d48}	CS_N LOW to RD_N LOW delay	0	-	ns
Writing				
t_{w28}	WR_N LOW pulse width	15	-	ns
t_{su28}	address set-up time before WR_N LOW	0	-	ns
t_{h28}	address hold time after WR_N HIGH	1	-	ns
t_{su38}	data set-up time before WR_N HIGH	5	-	ns
t_{h38}	data hold time after WR_N HIGH	2	-	ns
t_{d58}	WR_N HIGH to CS_N HIGH delay	1	-	ns
t_{d68}	CS_N LOW to WR_N LOW delay	0	-	ns

15.2.1.2 PIO register access

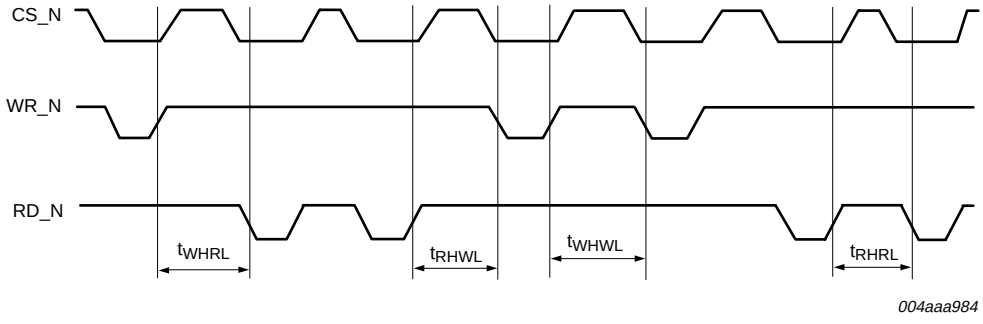


Fig 29. PIO register access

Table 176. Register access

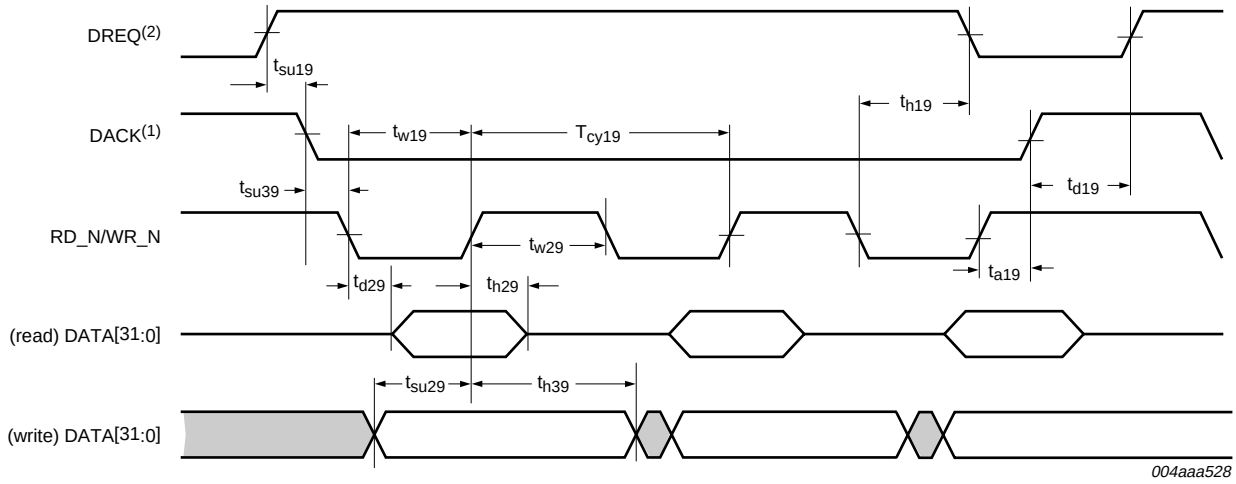
T_{amb} = −40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
t_{WHRL}	WR_N HIGH to RD_N LOW time	86	-	ns
t_{RHWL}	RD_N HIGH to WR_N LOW time	86	-	ns
t_{WHWL}	WR_N HIGH to WR_N LOW time	86[1]	-	ns
t_{RHRL}	RD_N HIGH to RD_N LOW time	86[1]	-	ns

[1] For the Data Port register, the minimum value is 25 ns.

15.2.2 DMA timing

15.2.2.1 DMA read or write



DREQ is continuously asserted until the last transfer is done or the FIFO is full.

Data strobes: RD_N (read) and WR_N (write).

(1) Programmable polarity: shown as active LOW.

(2) Programmable polarity: shown as active HIGH.

Fig 30. DMA read or write

Table 177. DMA read or write

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
T_{cy19}	read or write cycle time	75	-	ns
t_{su19}	DREQ set-up time before first DACK on	10	-	ns
t_{d19}	DREQ on delay after last strobe off	33.33	-	ns
t_{h19}	DREQ hold time after last strobe on	0	53	ns
t_{w19}	RD_N/WR_N pulse width	40	600	ns
t_{w29}	RD_N/WR_N recovery time	36	-	ns
t_{d29}	read data valid delay after strobe on	-	30	ns
t_{h29}	read data hold time after strobe off	-	5	ns
t_{h39}	write data hold time after strobe off	1	-	ns
t_{su29}	write data set-up time before strobe off	10	-	ns
t_{su39}	DACK set-up time before RD_N/WR_N assertion	0	-	ns
t_{a19}	DACK de-assertion after RD_N/WR_N de-assertion	3	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
T_{cy19}	read or write cycle time	75	-	ns
t_{su19}	DREQ set-up time before first DACK on	10	-	ns
t_{d19}	DREQ on delay after last strobe off	33.33	-	ns

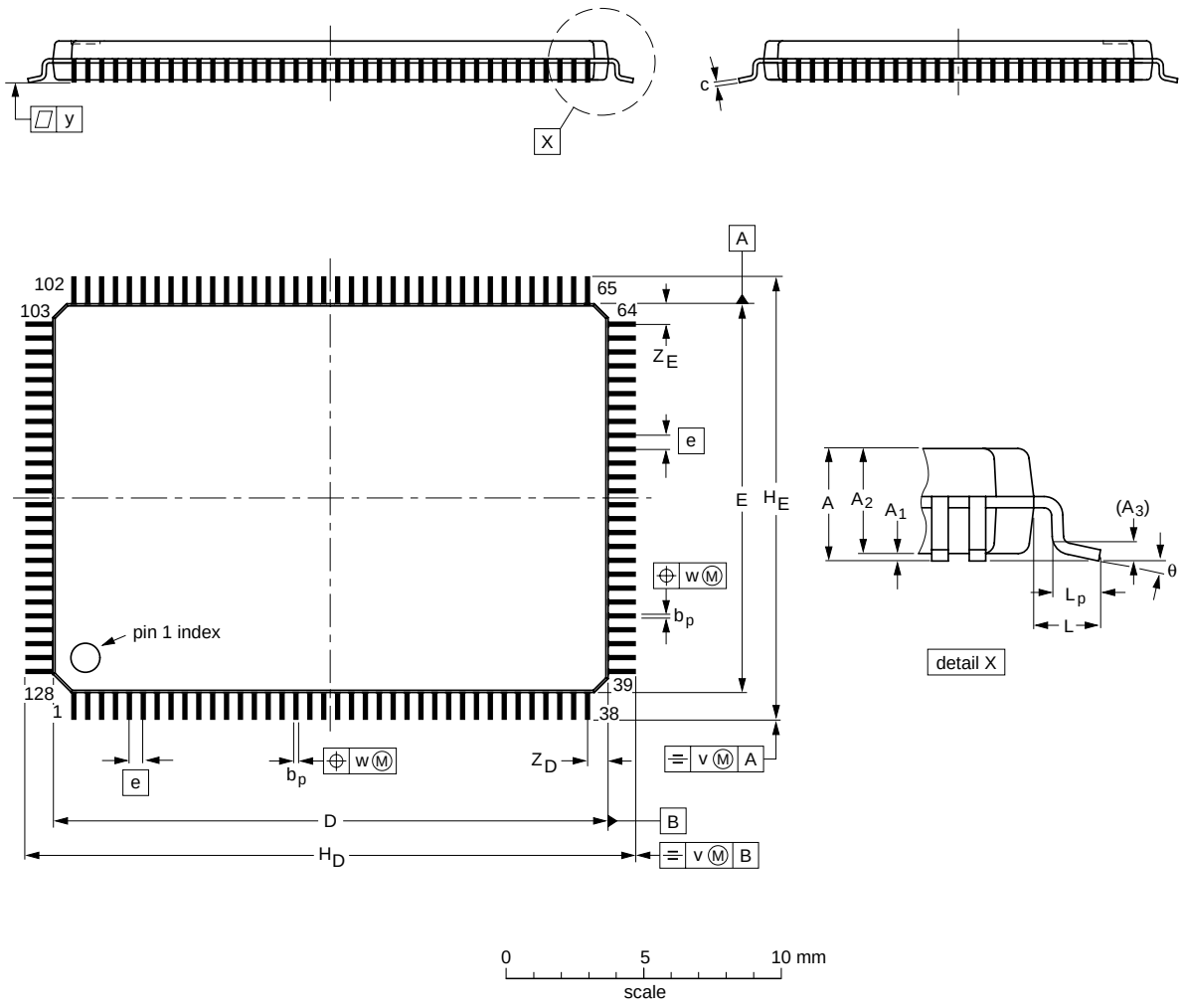
Table 177. DMA read or write *…continued*

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

Symbol	Parameter	Min	Max	Unit
t_{h19}	DREQ hold time after last strobe on	0	53	ns
t_{w19}	RD_N/WR_N pulse width	39	600	ns
t_{w29}	RD_N/WR_N recovery time	36	-	ns
t_{d29}	read data valid delay after strobe on	-	20	ns
t_{h29}	read data hold time after strobe off	-	5	ns
t_{h39}	write data hold time after strobe off	1	-	ns
t_{su29}	write data set-up time before strobe off	10	-	ns
t_{su39}	DACK set-up time before RD_N/WR_N assertion	0	-	ns
t_{a19}	DACK de-assertion after RD_N/WR_N de-assertion	3	-	ns

16. Package outline

LQFP128: plastic low profile quad flat package; 128 leads; body 14 x 20 x 1.4 mm **SOT425-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.15 0.05	1.45 1.35	0.25	0.27 0.17	0.20 0.09	20.1 19.9	14.1 13.9	0.5	22.15 21.85	16.15 15.85	1	0.75 0.45	0.2	0.12	0.1	0.81 0.59	0.81 0.59	7^o 0^o

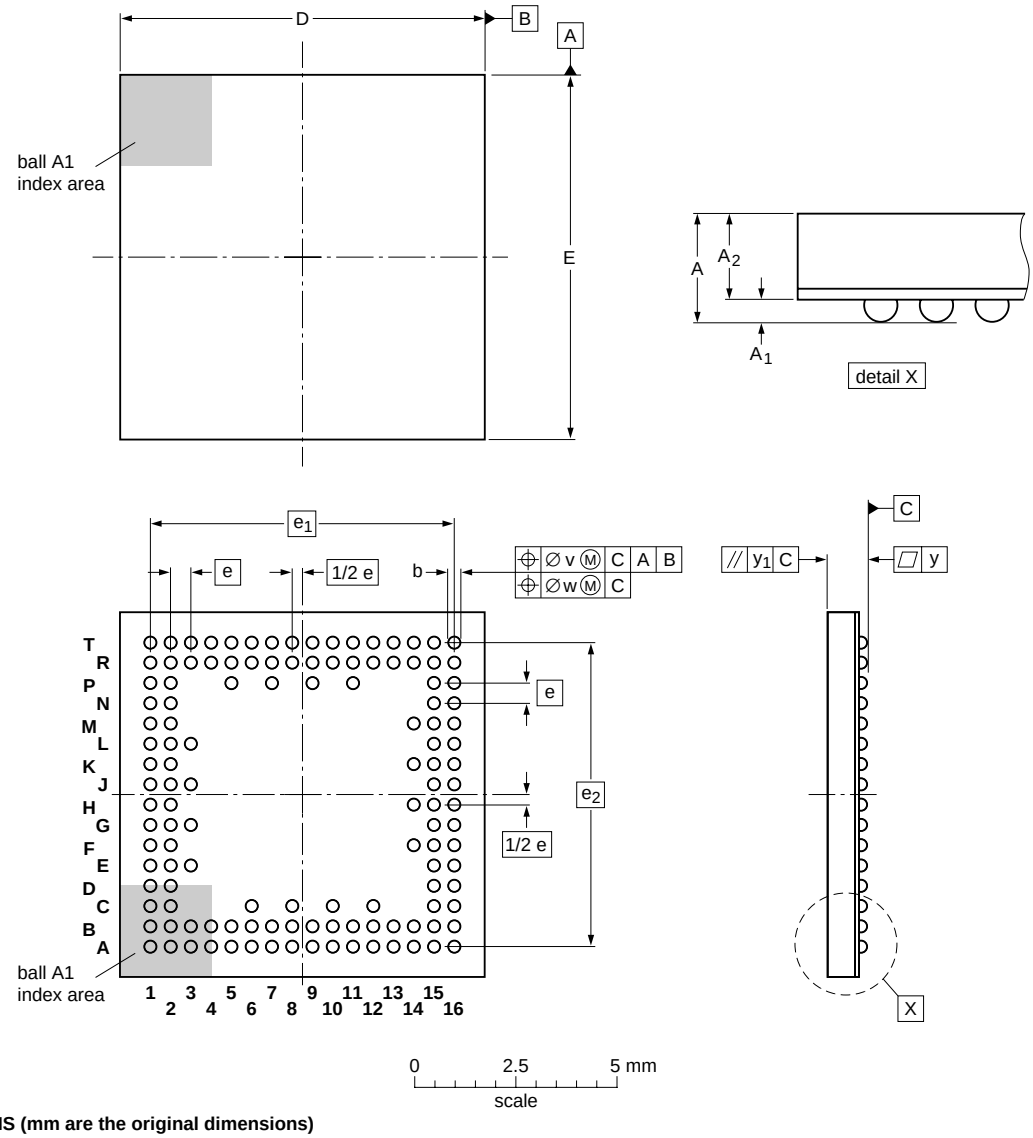
Note

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

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT425-1	136E28	MS-026				~~00-01-19~~ 03-02-20

Fig 31. Package outline SOT425-1 (LQFP128)

TFBGA128: plastic thin fine-pitch ball grid array package; 128 balls; body 9 x 9 x 0.8 mm **SOT857-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max	A_1	A_2	b	D	E	e	e_1	e_2	v	w	y	y_1
mm	1.1	0.25 0.15	0.85 0.75	0.35 0.25	9.1 8.9	9.1 8.9	0.5	7.5	7.5	0.15	0.05	0.08	0.1

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT857-1		MO-195				~~04-05-05~~ 04-06-22

Fig 32. Package outline SOT857-1 (TFBGA128)

17. Soldering of SMD packages

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

17.1 Introduction to soldering

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

17.2 Wave and reflow soldering

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

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

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

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

Key characteristics in both wave and reflow soldering are:

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

17.3 Wave soldering

Key characteristics in wave soldering are:

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

17.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 33) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 178 and 179

Table 178. SnPb eutectic process (from J-STD-020C)

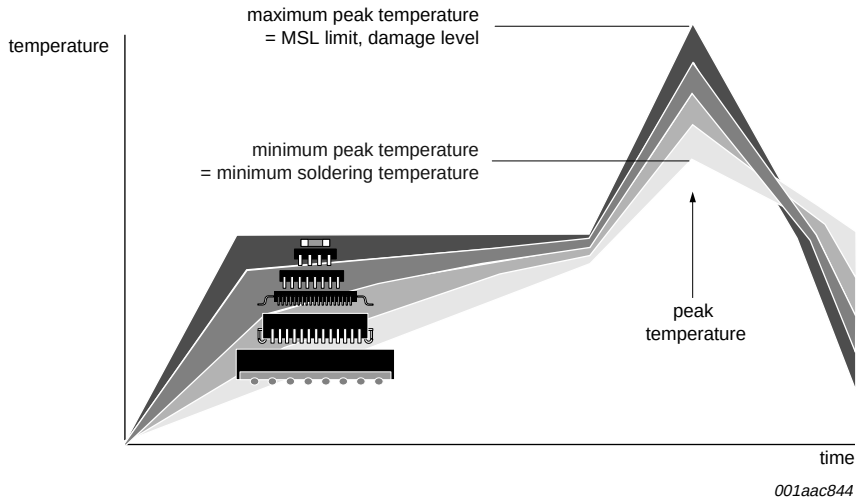
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm^3)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 179. Lead-free process (from J-STD-020C)

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

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

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 33.



MSL: Moisture Sensitivity Level

Fig 33. Temperature profiles for large and small components

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

18. Soldering of through-hole mount packages

18.1 Introduction to soldering through-hole mount packages

This text gives a very brief insight into wave, dip and manual soldering.

Wave soldering is the preferred method for mounting of through-hole mount IC packages on a printed-circuit board.

18.2 Soldering by dipping or by solder wave

Driven by legislation and environmental forces the worldwide use of lead-free solder pastes is increasing. Typical dwell time of the leads in the wave ranges from 3 seconds to 4 seconds at 250 °C or 265 °C, depending on solder material applied, SnPb or Pb-free respectively.

The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg(max)}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

18.3 Manual soldering

Apply the soldering iron (24 V or less) to the lead(s) of the package, either below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 °C and 400 °C, contact may be up to 5 seconds.

18.4 Package related soldering information

Table 180. Suitability of through-hole mount IC packages for dipping and wave soldering

Package	Soldering method	
	Dipping	Wave
CPGA, HCPGA	-	suitable
DBS, DIP, HDIP, RDBS, SDIP, SIL	suitable	suitable[1]
PMFP[2]	-	not suitable

[1] For SDIP packages, the longitudinal axis must be parallel to the transport direction of the printed-circuit board.

[2] For PMFP packages hot bar soldering or manual soldering is suitable.

19. Abbreviations

Table 181. Abbreviations

Acronym	Description
ACK	Acknowledgment
ASIC	Application-Specific Integrated Circuit
ATL	Asynchronous Transfer List
ATX	Analog Transceiver
CS	Complete Split
DMA	Direct Memory Access
DSC	Digital Still Camera
DW	Double Word
EHCI	Enhanced Host Controller Interface
EMI	ElectroMagnetic Interference
EOP	End-Of-Packet
EOT	End-Of-Transfer
ESD	ElectroStatic Discharge
ESR	Effective Series Resistance
FIFO	First In, First Out
FS	Full-Speed
FLS	Frame List Size
GDMA	Generic DMA
GPIO	General-Purpose Input/Output
GPS	Global Positioning System
HC	Host Controller
HNP	Host Negotiation Protocol
HS	High-Speed
iTD	isochronous Transfer Descriptor
INT	Interrupt
ISO	Isochronous
ISR	Interrupt Service Routine
ITL	Isochronous (ISO) Transfer List
LS	Low-Speed
LSByte	Least Significant Byte
MSByte	Most Significant Byte
NAK	Not Acknowledged
NYET	Not Yet
OC	Overcurrent
OHCI	Open Host Controller Interface
OTG	On-the-Go
PCI	Peripheral Component Interconnect
PID	Packet Identifier
PIO	Programmed Input/Output

Table 181. Abbreviations *…continued*

Acronym	Description
PLL	Phase-Locked Loop
PMOS	Positive-channel Metal-Oxide Semiconductor
POR	Power-On Reset
PORP	Power-On Reset Pulse
PTD	Philips Transfer Descriptor
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
SE0	Single Ended 0
SE1	Single Ended 1
SIE	Serial Interface Engine
siTD	split isochronous Transfer Descriptor
SOF	Start-Of-Frame
SRAM	Static Random Access Memory
SRP	Session Request Protocol
SS	Start Split
TT	Transaction Translator
UHCI	Universal Host Controller Interface
USB	Universal Serial Bus

20. References

[1] Universal Serial Bus Specification Rev. 2.0

[2] Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0

[3] On-The-Go Supplement to the USB Specification Rev. 1.3

[4] ISP1582/83 Control Pipe (AN10031)

[5] Interfacing the ISP176x to the Intel PXA25x processor (AN10037)

[6] ISP1582/83 Firmware Programming Guide (AN10039)

[7] ISP1761 Peripheral DMA Initialization (AN10040)

[8] ISP176x Linux Programming Guide (AN10042)

[9] Embedded Systems Design with the ISP176x (AN10043)

[10] ISP1760/1 Frequently Asked Questions (AN10054)

21. Revision history

Table 182. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
ISP1761_5	20080313	Product data sheet	-	ISP1761_4
Modifications:	• Changed mini-A, mini-B and mini-AB to micro-A, micro-B and micro-AB, respectively. • Changed *On-The-Go Supplement to the USB Specification* from Rev. 1.2 to Rev. 1.3. • Table 52 “Power Down Control register (address 0354h) bit description”: updated description for bits 12 and 11. • Section 9.4.3 “HNP implementation and OTG state machine”: updated the third last paragraph. • Section 10.1.1.3 “DMA initialization”: updated the second paragraph. • Table 101 “Mode register (address 020Ch) bit description”: updated description for bit 8. • Section 10.6.2 “Control Function register”: updated bit 2 access and description. • Table 117 “DcBufferStatus - Device Controller Buffer Status register (address 021Eh) bit allocation”: updated access type for bits 1 and 0 from R/W to R. • Table 142 “DcInterrupt - Device Controller Interrupt register (address 0218h) bit description”: updated description for bit 7. • Table 154 “Limiting values”: updated the conditions column of V_{esd}. • Table 160 “Static characteristics: V_{BUS} comparators”: added Table note 1 and Table note 2.			
ISP1761_4	20070305	Product data sheet	-	ISP1761_3
ISP1761_3	20061127	Product data sheet	-	ISP1761_2
ISP1761_2 (9397 750 15191)	20051005	Product data sheet	-	ISP1761_1
ISP1761_1 (9397 750 13258)	20050112	Product data sheet	-	-

22. Legal information

22.1 Data sheet status

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

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

[2] The term 'short data sheet' is explained in section "Definitions".

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

22.2 Definitions

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

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

22.3 Disclaimers

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

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

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

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

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

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

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

22.4 Trademarks

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

SoftConnect — is a trademark of NXP B.V.

23. Contact information

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

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

24. Tables

continued >>

continued >>

continued >>

25. Figures

26. Contents

continued >>

continued >>

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

Date of release: 13 March 2008
Document identifier: ISP1761_5

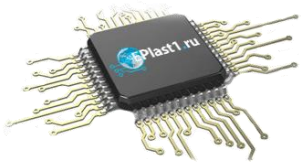
Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.



Как с нами связаться

Телефон: 8 (812) 309 58 32 (многоканальный)
Факс: 8 (812) 320-02-42
Электронная почта: org@eplast1.ru
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.