

# ISP1761

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

Rev. 01 — 12 January 2005

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 the advanced Philips Slave Host Controller and the Philips ISP1582 Peripheral Controller.

The Hi-Speed USB Host Controller and Peripheral Controller comply to *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 *Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0*. The OTG controller is compliant with *On-The-Go Supplement to the USB Specification Rev. 1.0a*.

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 *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, Philips XA, Intel® StrongARM®, NEC® and Toshiba® MIPS, Motorola® 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 for reducing the host system's CPU load

**PHILIPS**

- Separate IRQ, DREQ and DACK lines for the Host Controller and the Peripheral Controller
- Integrated multiconfiguration 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:  $I_{CC(susp)} < 150$   $\mu$ A at the room temperature
- Available in LQFP128 and TFBGA128 packages
- Host Controller-specific features
  - ◆ High performance USB host with integrated high-speed USB transceivers; supports high-speed, full-speed and low-speed
  - ◆ The EHCI core is adapted from *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 X 64 bits)
  - ◆ Additional 2.5 kB separate memory for TT
  - ◆ Individual or global overcurrent protection with built-in sense circuits
  - ◆ Overcurrent circuitry built-in (digital or analog overcurrent protection)
- OTG Controller-specific features
  - ◆ OTG transceiver: fully integrated; compliant with *On-The-Go Supplement to the USB Specification Rev. 1.0a*
  - ◆ 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 *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 x 20 x 1.4 mm	SOT425-1
ISP1761ET <a href="#">[1]</a>	TFBGA128	plastic thin fine-pitch ball grid array package; 128 balls; body 9 x 9 x 0.8 mm	SOT857-1

[1] The ISP1761ET is currently under development.

5. Block diagram

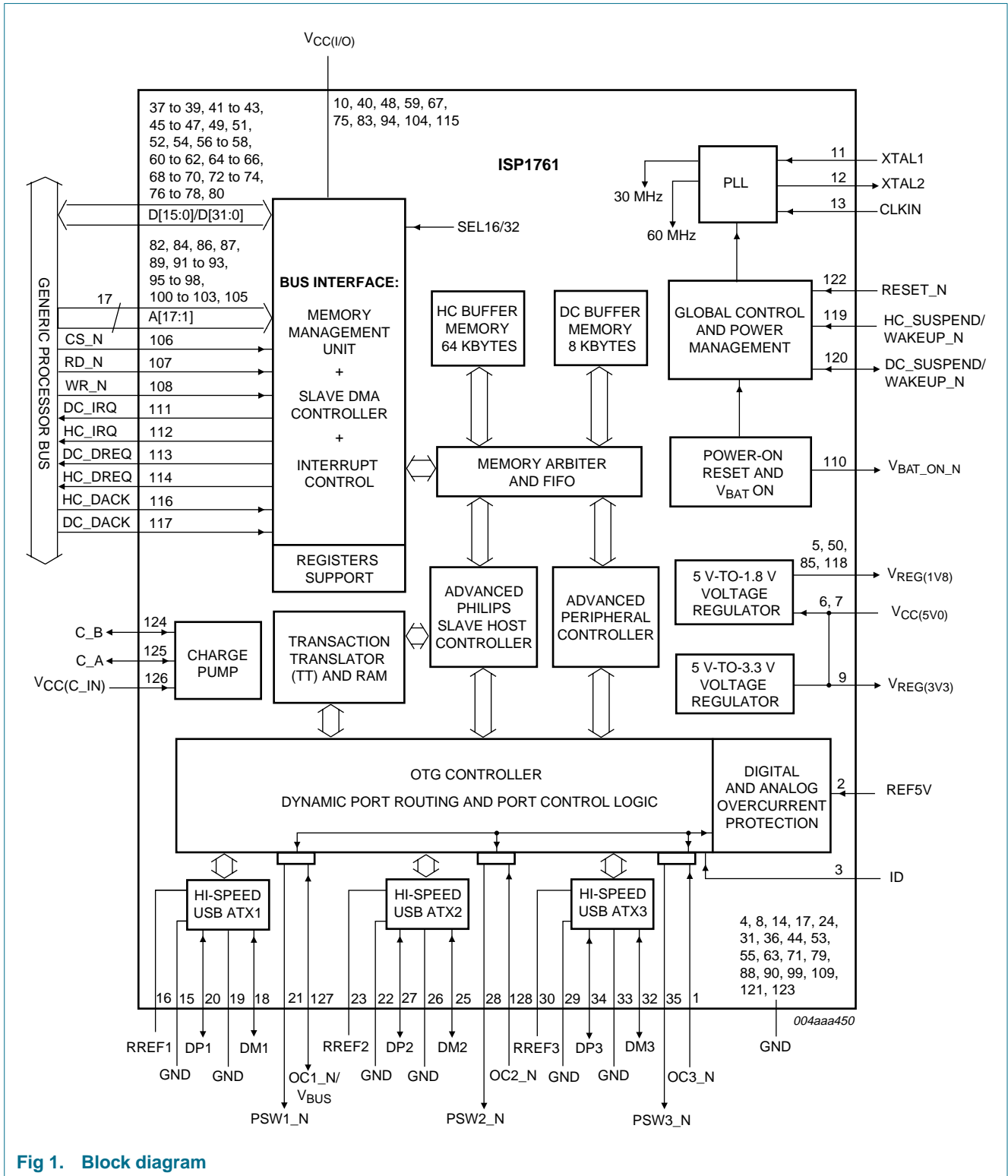


Fig 1. Block diagram

## 6. Pinning information

### 6.1 Pinning

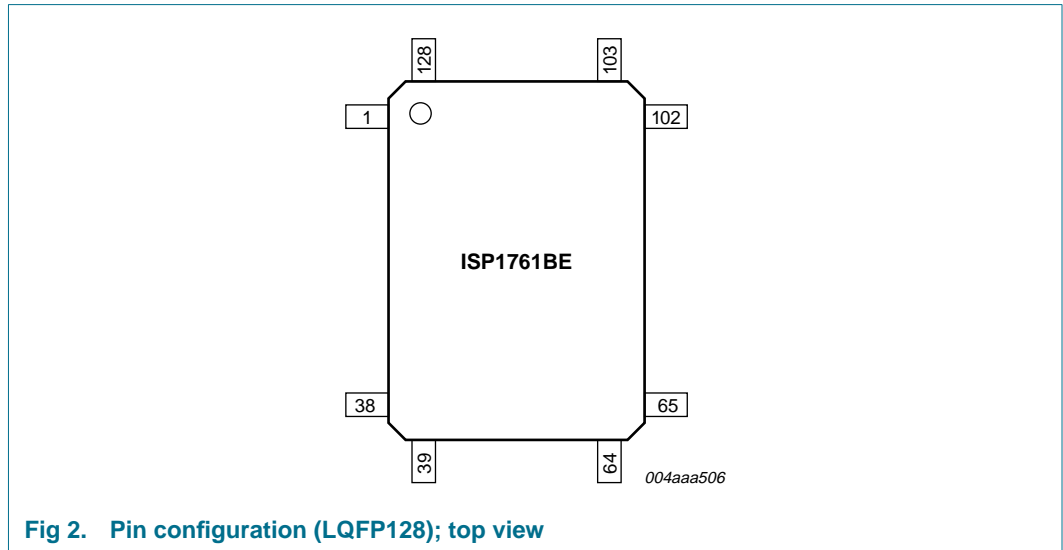


Fig 2. Pin configuration (LQFP128); top view

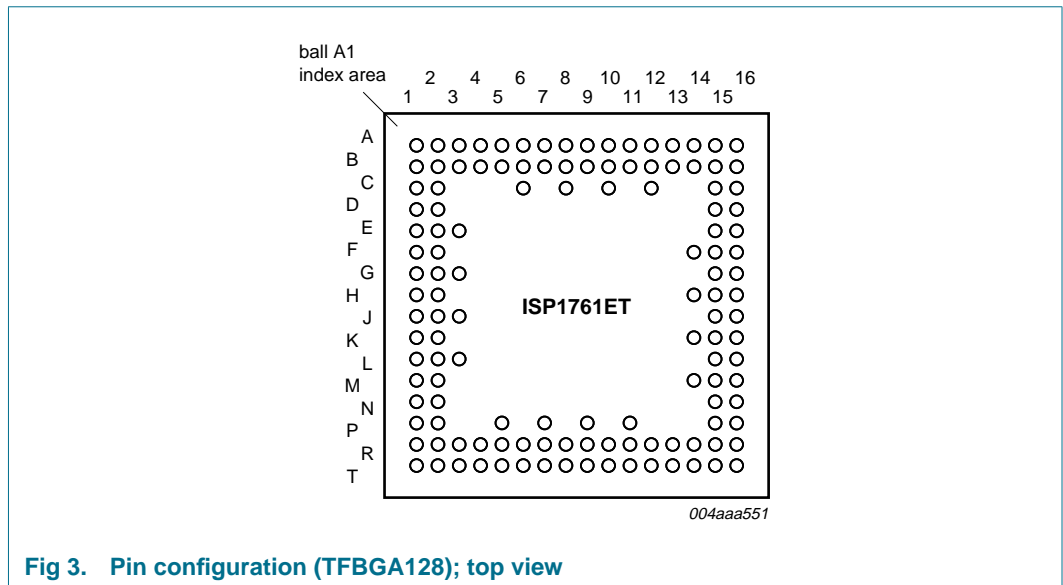


Fig 3. Pin configuration (TFBGA128); top view

## 6.2 Pin description

Table 2: Pin description

Symbol <sup>[1]</sup>	Pin LQFP128	Ball TFBGA128	Type <sup>[2]</sup>	Description
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 k $\Omega$ resistor input, 3.3 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 for detection of the default host or peripheral setting when port 1 is in the OTG mode input, 3.3 V tolerant
GND	4	A1	-	analog ground
$V_{REG(1V8)}$	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 <a href="#">Section 7.7</a>
$V_{CC(5V0)}$	6	C1	P	input to internal regulators (3.0 V to 5.5 V); connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
$V_{CC(5V0)}$	7	D2	P	input to internal regulators (3.0 V to 5.5 V); connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
GND	8	E3	-	oscillator ground
$V_{REG(3V3)}$	9	D1	P	regulator output (3.3 V); for decoupling only; connect a 100 nF capacitor and a 4.7 $\mu$ F to 10 $\mu$ F capacitor; see <a href="#">Section 7.7</a>
$V_{CC(I/O)}$	10	E2	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
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; connect to $V_{CC(I/O)}$ when not in use 3.3 V tolerant
GND	14	G3	-	digital ground
GND	15	G2	-	RREF1 ground
RREF1	16	G1	AI	reference resistor connection; connect a 12 k $\Omega$ $\pm$ 1 % resistor between this pin and the RREF1 ground
GND	17	H2	-	analog ground for port 1
DM1	18	H1	AI/O	downstream data minus port 1
GND	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	22	K2	-	RREF2 ground
RREF2	23	K1	AI	reference resistor connection; connect a 12 k $\Omega$ $\pm$ 1 % resistor between this pin and the RREF2 ground
GND	24	L3	-	analog ground for port 2
DM2	25	L1	AI/O	downstream data minus port 2
GND	26	L2	-	analog ground

Table 2: Pin description...continued

Symbol <sup>[1]</sup>	Pin LQFP128	Ball TFBGA128	Type <sup>[2]</sup>	Description
DP2	27	M2	AI/O	downstream data plus port 2
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	29	N2	-	RREF3 ground
RREF3	30	N1	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF3 ground
GND	31	P2	-	analog ground for port 3
DM3	32	P1	AI/O	downstream data minus port 3
GND	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
GND	36	T2	-	digital ground
DATA0	37	R3	I/O	data bit 0 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
V <sub>CC(I/O)</sub>	40	T4	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
DATA3	41	P5	I/O	data bit 3 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
GND	44	T6	-	digital ground
DATA6	45	R6	I/O	data bit 6 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant



Table 2: Pin description...continued

Symbol <sup>[1]</sup>	Pin LQFP128	Ball TFBGA128	Type <sup>[2]</sup>	Description
V <sub>CC(I/O)</sub>	48	R7	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
DATA9	49	T8	I/O	data bit 9 input and output bidirectional pad, push-pull input, three-state output, 4 mA output drive, 3.3 V tolerant
V <sub>REG(1V8)</sub>	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 <a href="#">Section 7.7</a>
DATA10	51	P9	I/O	data bit 10 input and output bidirectional pad, push-pull input, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
GND	53	R9	-	core ground
DATA12	54	T10	I/O	data bit 12 input and output bidirectional pad, push-pull input, three-state output, 4 mA output drive, 3.3 V tolerant
GND	55	R10	-	digital ground
DATA13	56	P11	I/O	data bit 13 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
V <sub>CC(I/O)</sub>	59	T12	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
DATA16	60	R12	I/O	data bit 16 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
GND	63	R14	-	digital ground
DATA19	64	T14	I/O	data bit 19 input and output bidirectional pad, push-pull input, three-state output, 4 mA output drive, 3.3 V tolerant

Table 2: Pin description...continued

Symbol <sup>[1]</sup>	Pin LQFP128	Ball TFBGA128	Type <sup>[2]</sup>	Description
DATA20	65	T15	I/O	data bit 20 input and output bidirectional pad, push-pull input, three-state output, 4 mA output drive, 3.3 V tolerant
DATA21	66	R15	I/O	data bit 21 input and output bidirectional pad, push-pull input, three-state output, 4 mA output drive, 3.3 V tolerant
V <sub>CC(I/O)</sub>	67	P15	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
DATA22	68	T16	I/O	data bit 22 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
GND	71	N16	-	digital ground
DATA25	72	N15	I/O	data bit 25 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
V <sub>CC(I/O)</sub>	75	M14	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
DATA28	76	L16	I/O	data bit 28 input and output bidirectional pad, push-pull input, three-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, three-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, three-state output, 4 mA output drive, 3.3 V tolerant
GND	79	K15	-	digital ground
DATA31	80	K14	I/O	data bit 31 input and output bidirectional pad, push-pull input, three-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

Table 2: Pin description...continued

Symbol <sup>[1]</sup>	Pin LQFP128	Ball TFBGA128	Type <sup>[2]</sup>	Description
V <sub>CC(I/O)</sub>	83	J15	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
A2	84	H15	I	address pin 2 input, 3.3 V tolerant
V <sub>REG(1V8)</sub>	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 <a href="#">Section 7.7</a>
A3	86	H14	I	address pin 3 input, 3.3 V tolerant
A4	87	F16	I	address pin 4 input, 3.3 V tolerant
GND	88	G15	-	core ground
A5	89	F15	I	address pin 5 input, 3.3 V tolerant
GND	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 <sub>CC(I/O)</sub>	94	D15	P	digital supply; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
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
GND	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 <sub>CC(I/O)</sub>	104	B13	P	digital voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>

Table 2: Pin description...continued

Symbol <sup>[1]</sup>	Pin LQFP128	Ball TFBGA128	Type <sup>[2]</sup>	Description
A17	105	C12	I	address pin 17 input, 3.3 V tolerant
CS_N	106	A12	I	chip select signal that indicates the area 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
GND	109	A11	-	digital ground
V <sub>BAT_ON_N</sub>	110	C10	OD	to indicate the presence of a minimum 3.3 V on pins 6 and 7 (open-drain); connect to V <sub>CC(I/O)</sub> 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	DMAC request for the Peripheral Controller output 4 mA drive, 3.3 V tolerant
HC_DREQ	114	B9	O	DMAC request for Host Controller output 4 mA drive, 3.3 V tolerant
V <sub>CC(I/O)</sub>	115	C8	P	digital voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see <a href="#">Section 7.7</a>
HC_DACK	116	A8	I	Host Controller DMA request acknowledgment; when not in use, connect to V <sub>CC(I/O)</sub> 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 <sub>CC(I/O)</sub> through a 10 kΩ pull-up resistor input, 3.3 V tolerant
V <sub>REG(1V8)</sub>	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 <a href="#">Section 7.7</a>
HC_SUSPEND /WAKEUP_N	119	A7	I/OD	Host Controller suspend and wake-up; three-state suspend output (active LOW) and wake-up input circuits are connected together <ul style="list-style-type: none"> <li>• HIGH = output is three-state; ISP1761 is in suspend mode</li> <li>• LOW = output is LOW; ISP1761 is not in suspend mode.</li> </ul> connect to V <sub>CC(I/O)</sub> through an external 10 kΩ pull-up resistor output pad, open-drain, 4 mA output drive, 3.3 V tolerant

Table 2: Pin description...continued

Symbol [1]	Pin LQFP128	Ball TFBGA128	Type [2]	Description
DC_SUSPEND /WAKEUP_N	120	C6	I/OD	Peripheral Controller suspend and wake-up; three-state suspend output (active LOW) and wake-up input circuits are connected together <ul style="list-style-type: none"> <li>• HIGH = output is three-state; ISP1761 is in suspend mode</li> <li>• LOW = output is LOW; ISP1761 is not in suspend mode.</li> </ul> connect to $V_{CC(I/O)}$ through an external 10 k $\Omega$ pull-up resistor output pad, open-drain, 4 mA output drive, 3.3 V tolerant
GND	121	A6	-	core ground
RESET_N	122	B6	I	external power-up reset; active LOW input, 3.3 V tolerant <b>Remark:</b> During reset, ensure that all the input pins to the ISP1761 are not toggling.
GND	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: <ul style="list-style-type: none"> <li>• Port 1 OC1_N detection when port 1 is configured for host functionality and an external power switch is used; connect to <math>V_{CC(I/O)}</math> through a 10 k<math>\Omega</math> resistor</li> <li>• <math>V_{BUS}</math> out when internal charge pump is used and port 1 is configured for the host functionality; maximum 50 mA current capability; only for port 1</li> <li>• <math>V_{BUS}</math> input detection when port 1 is defined for the peripheral functionality.</li> </ul> input, 3.3 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 k $\Omega$ resistor input, 3.3 V tolerant

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

[2] 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.

## 7. Functional description

### 7.1 ISP1761 internal architecture: Advanced Philips Slave Host Controller and hub

The EHCI block and the Hi-Speed USB hub block are the main components of the Advanced Philips Slave Host Controller.

The EHCI is the latest generation design, with improved data bandwidth. The EHCI in the ISP1761 is adapted from *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 the 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 the 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 *ISP176x Linux Programming Guide (AN10042)*.

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.

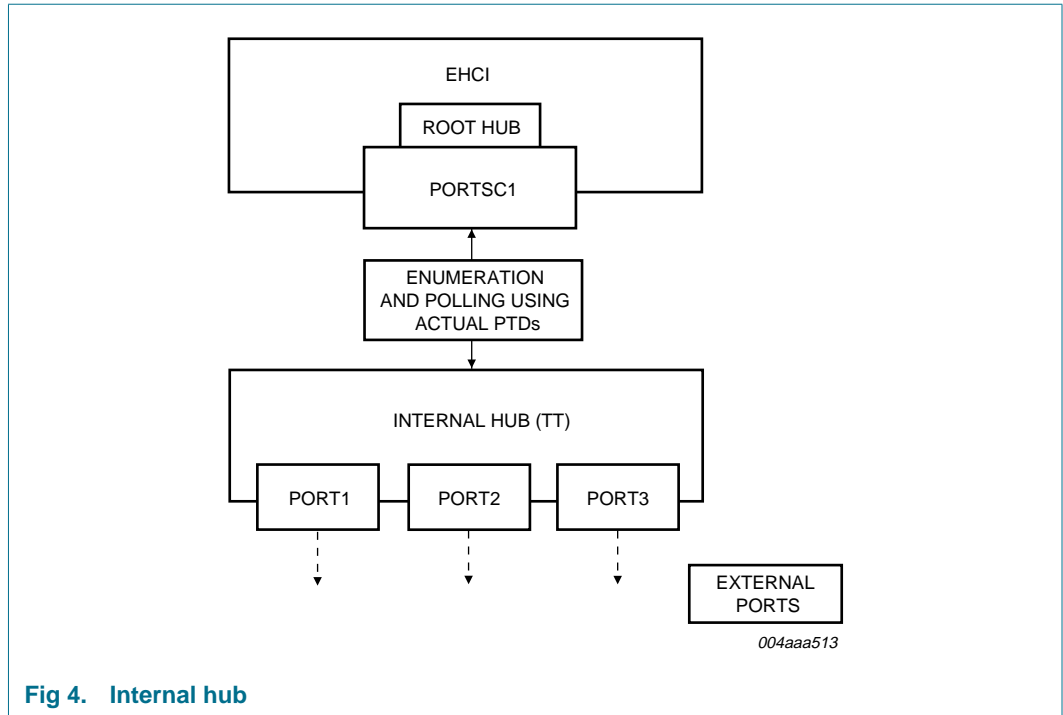


Fig 4. Internal hub

7.1.1 Internal clock scheme

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

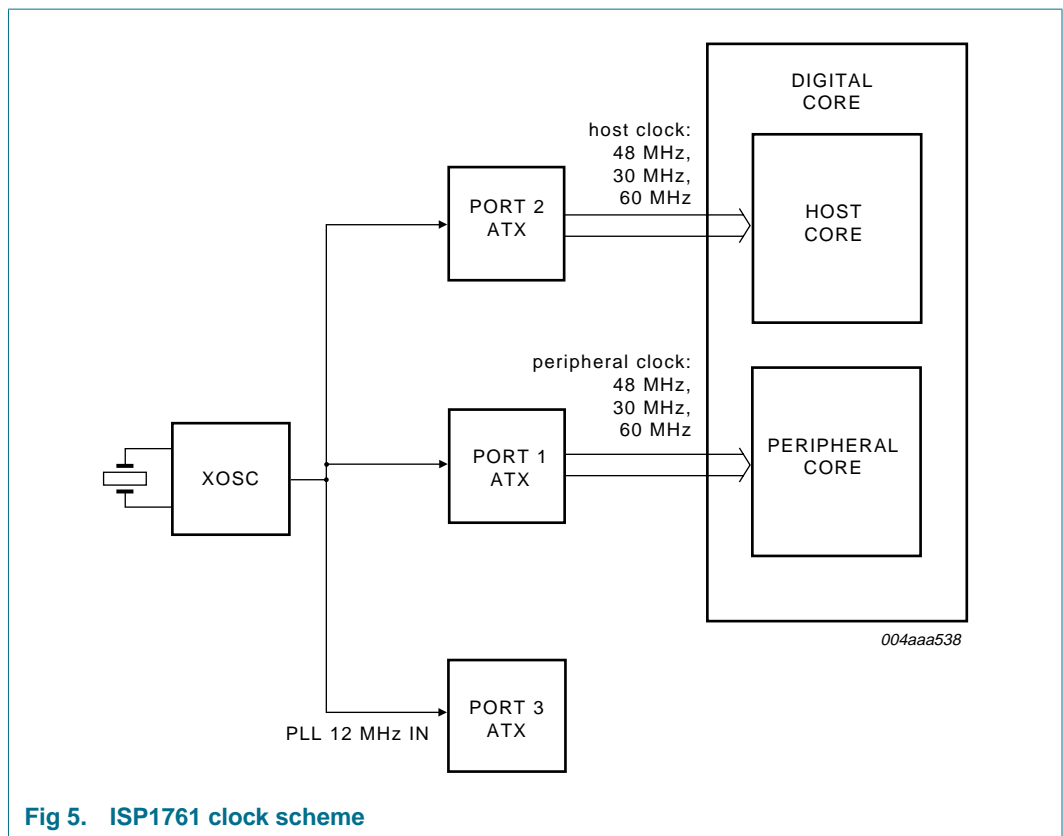


Fig 5. ISP1761 clock scheme

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 registers (1 kB) from the total addressable memory space defined by the ISP1761 (64 kB). This is an optimized value for achieving the highest performance with a 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 memory of the system 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 Parallel I/O (PIO) (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 prefetching 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 bigger 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 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, Philips 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.

Both the PTD and payload memory zones are divided into three dedicated areas for each main type of USB transfer: isochronous (ISO), interrupt (INT) and Acknowledged 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 address space of the registers. 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

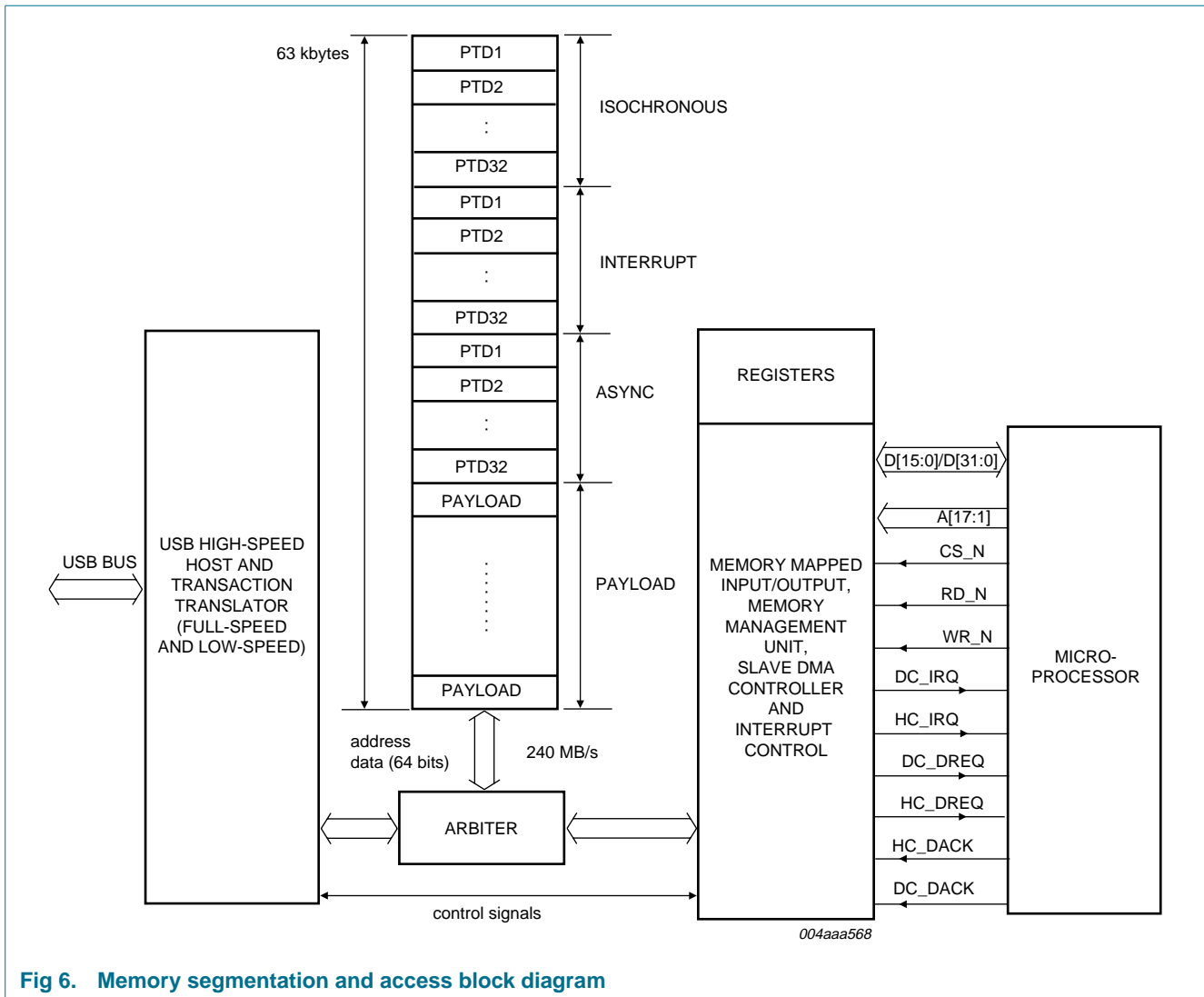


Fig 6. Memory segmentation and access block diagram

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 k $\Omega$  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, as described in the following section.

### 7.3.1 PIO mode access—memory read cycle

The following method has been implemented to reduce the read access timing in the case of 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 prefetch 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 prefetch 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 MUXing of appropriate data to the CPU.

- The address written to the Memory register is incremented and used to successively prefetch 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.

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 the 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 prefetching address before a write cycle to memory.

The ISP1761 internal write address will not be automatically 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 prefetching address before a register read.

The ISP1761 register read address will not be automatically 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 prefetching address before a register write.

The ISP1761 register write address will not be automatically 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—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 deasserted on the last cycle (RD\_N or WR\_N active pulse) of that burst. It will be reasserted shortly after the DACK deassertion, as long as the DMA transfer counter was not reached. DREQ will be deasserted on the last cycle when the DMA transfer counter is reached and will not reasserted until the DMA reprogramming is performed. Both the DREQ and DACK signals are programmable as active LOW or active HIGH, according to the 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 deasserted, 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 the 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 the 16-bit data bus mode and double word aligned in the 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 the 16-bit mode data bus width as compared to the 32-bit mode.
  - The transfer length—expressed in number of bytes—defines the number of bursts. The DREQ will be deasserted 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 deasserted 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 ISP1761 DMA parameters programmed above. 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 the 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 the 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 than 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. Setting some of the bits in these registers to logic 1 will determine the IRQ generation only when the respective AND or OR conditions of completing the respective PTDs is met.



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 timeout 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 Electro-Magnetic 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 the case of a PCI Hi-Speed USB Host Controller with a Hi-Speed USB hub attached.

While the ISP1761 is set in suspend mode, the 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 the following events:

- External USB device connect or disconnect
- Assertion of the CS\_N signal because of any access to the ISP1761
- 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 should 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 three-state and can be pulled down, driving them externally by switching the processor's GPIO lines to the output mode to generate the ISP1761 wake-up.

The HC\_SUSPEND/WAKEUP\_N and DC\_SUSPEND/WAKEUP\_N pins are three-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 the ISP1761 internal blocks to disable for lower power consumption as defined in [Table 8](#).

The lowest suspend current ( $I_{CC(susp)}$ ) that can be achieved is approximately 150  $\mu\text{A}$  at room temperature. The suspend current will increase with the increase in temperature, with approximately 300  $\mu\text{A}$  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  $\mu\text{A}$ . 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 the hybrid mode, which allows only the  $V_{CC(I/O)}$  powered on to avoid loading of the system bus.

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

It is necessary to wait for the CLK\_RDY 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 CLK\_RDY interrupt means that the internal clocks are running and the normal functionality is achieved.

It is estimated that the CLK\_RDY interrupt will be generated less than 100  $\mu\text{s}$  after the wake-up event, if the power to the ISP1761 was on during suspend.

If the ISP1761 is used in the hybrid mode and  $V_{CC(5V0)}$  is off during suspend, a 2 ms reset pulse is required when the power is switched back to on, before starting to program the resume state. 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.

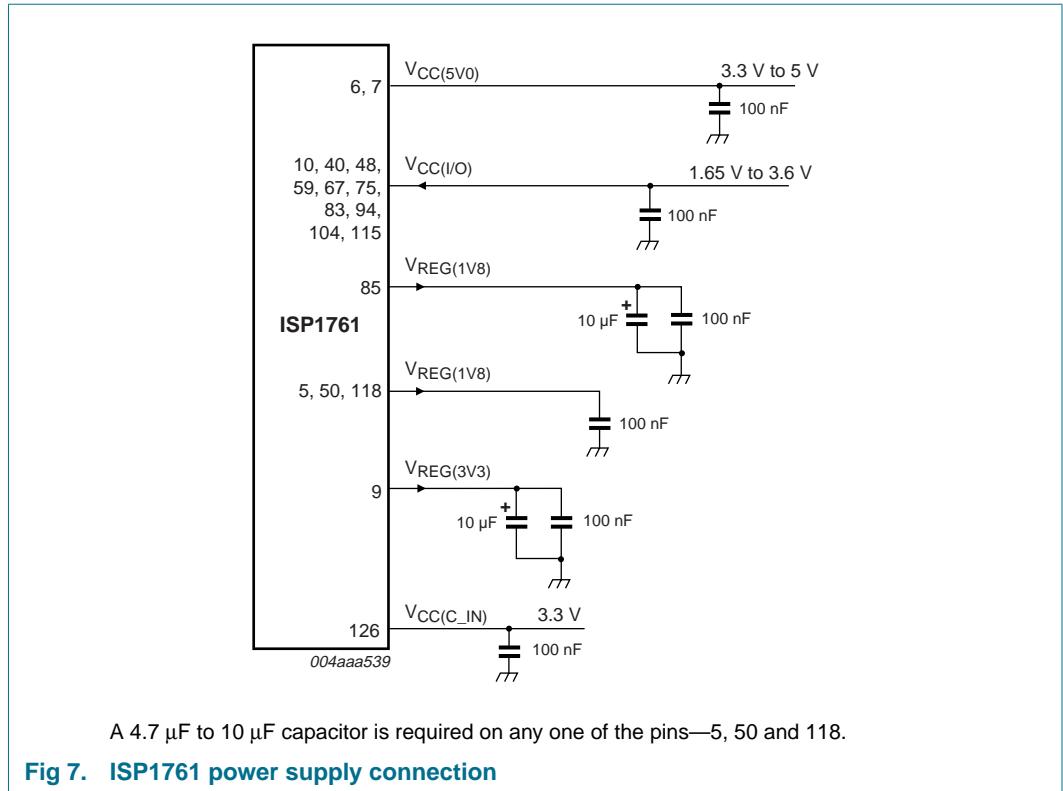
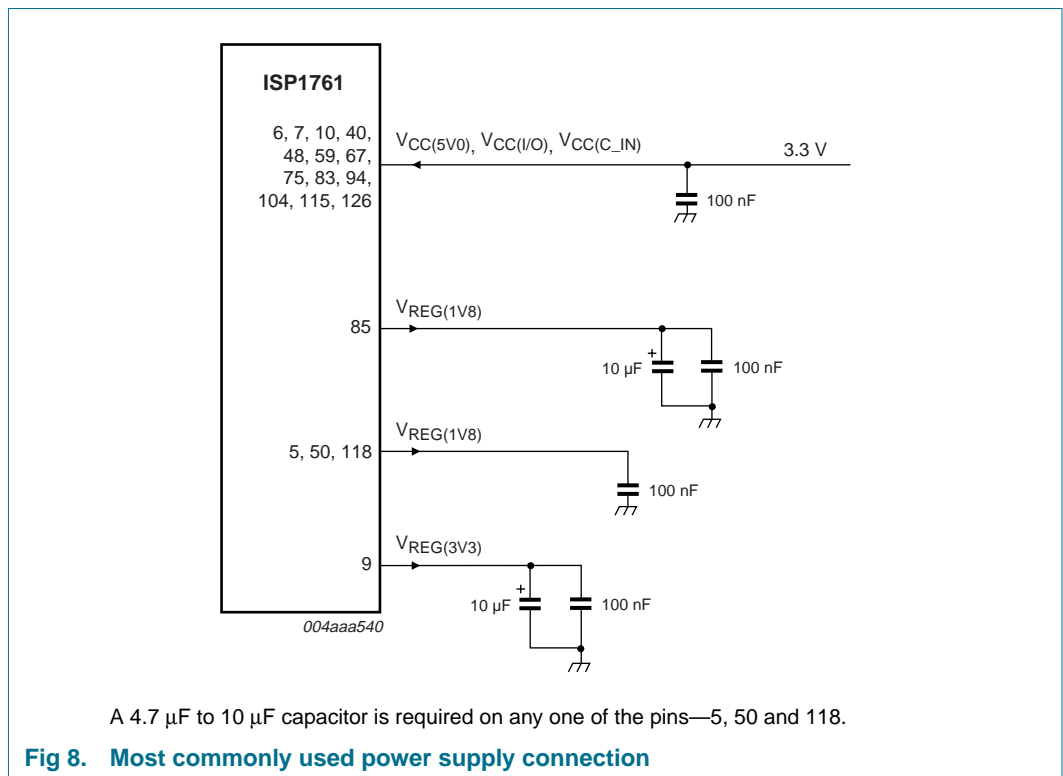


Figure 8 shows the most commonly used power supply connection.



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  $\mu$ A. 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 to on, before starting to program the resume.

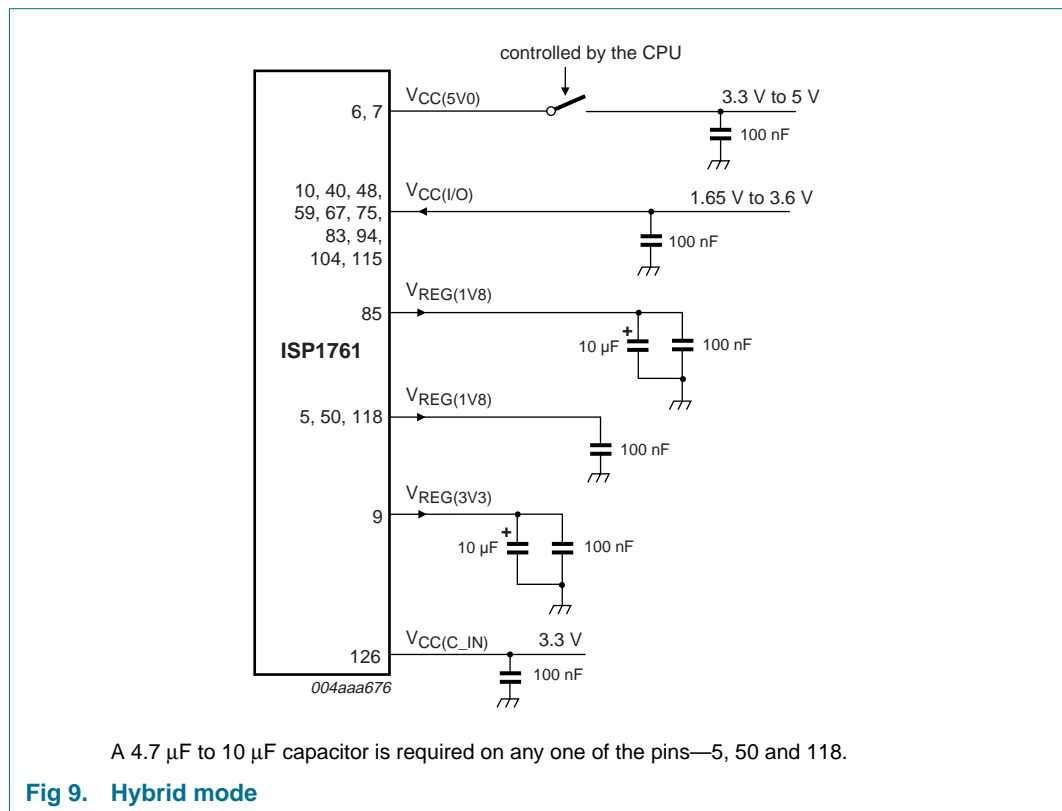


Table 7 shows the status of output pins during hybrid mode.

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,	on	on	normal
HC_DACK, DC_DACK,	on	off	high-Z
HC_SUSPEND/WAKEUP_N, DC_SUSPEND/WAKEUP_N	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 be automatically disabled by the ISP1761 on an overcurrent event occurrence, by deasserting the PSWn\_N signal without any software intervention.

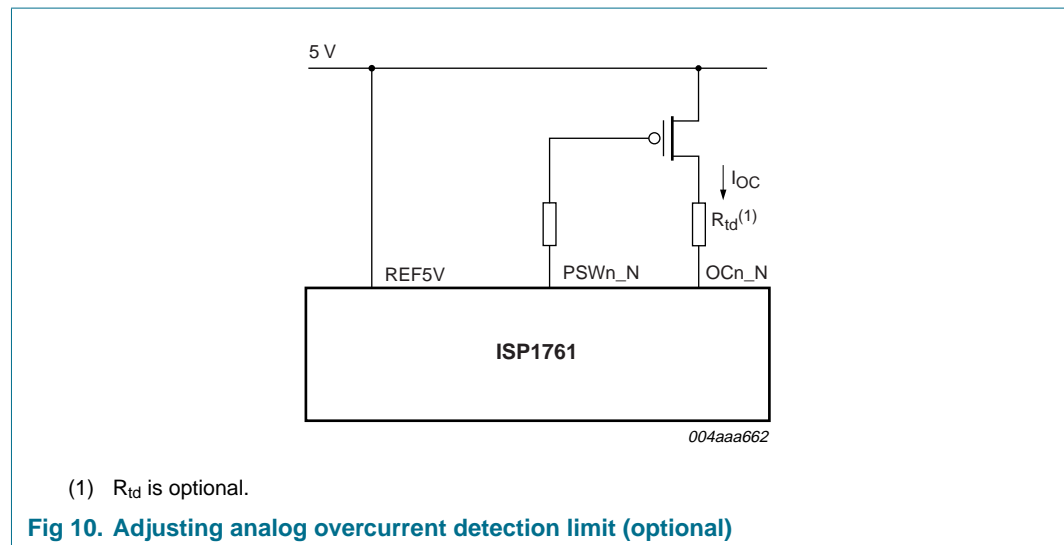
When using the integrated analog overcurrent detection, the range of the overcurrent detection voltage for the ISP1761 is 45 mV to 90 mV. Calculation of the external components should be based on the 45 mV value, with the actual overcurrent detection threshold usually positioned in the middle of the interval.

For an overcurrent limit of 500 mA per port, a PMOS transistor with  $R_{\text{DS(on)}}$  of approximately 100 m $\Omega$  is required. If a PMOS transistor with a lower  $R_{\text{DS(on)}}$  is used, the analog overcurrent detection can be adjusted using a series resistor; see [Figure 10](#).

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

$\Delta V_{\text{PMOS}}$  = voltage drop on PMOS

$I_{\text{OC(nom)}} = 1 \mu\text{A}$ .



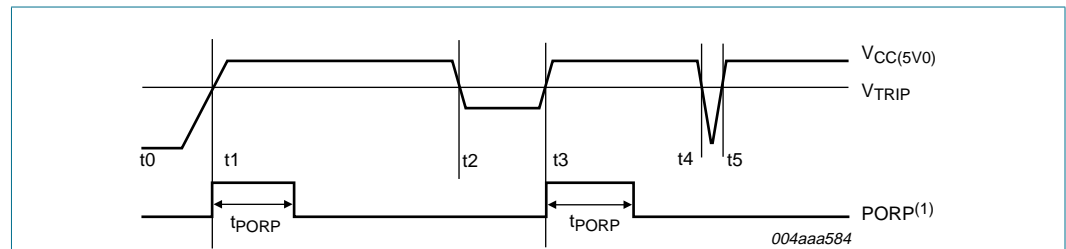
The digital overcurrent scheme requires using an external power switch with integrated overcurrent detection, such as: LM3526, MIC2526 (2 ports) or LM3544 (4 ports). These devices are controlled by PSWn\_N signals corresponding to each port. In the case of overcurrent occurrence, these devices will assert OCn\_N signals. On OCn\_N assertion, the ISP1761 cuts off the port power by deasserting 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 of 1 ms to 3 ms will prevent false overcurrent reporting because of in-rush currents when plugging a USB device.

### 7.9 Power-On Reset (POR)

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

To give a better view of the functionality, Figure 11 shows a possible curve of  $V_{CC(5V0)}$  with dips at  $t_2-t_3$  and  $t_4-t_5$ . If the dip at  $t_4-t_5$  is too short (that is,  $< 11 \mu s$ ), the internal POR pulse will not react and will remain LOW. The internal POR starts with a 1 at  $t_0$ . At  $t_1$ , 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  $11 \mu s$ .

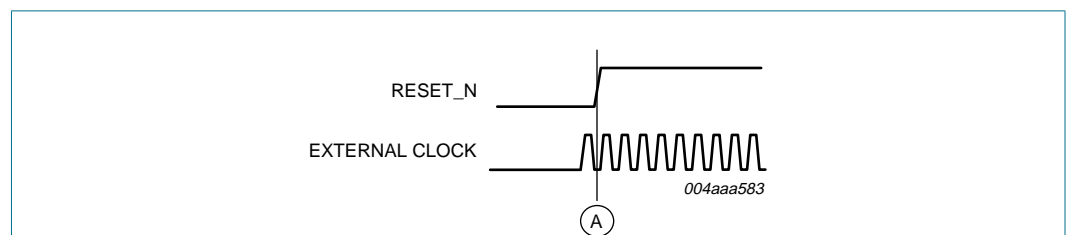


(1) PORP = Power-On Reset Pulse.

**Fig 11. Internal power-on reset timing**

The recommended RESET input pulse length at power-on should 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 for addressing of 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
<b>EHCI capability registers</b>			
0000h	CAPLENGTH	20h	<a href="#">Section 8.1.1 on page 32</a>
0002h	HCVERSION	0100h	<a href="#">Section 8.1.2 on page 32</a>
0004h	HCSPARAMS	0000 0011h	<a href="#">Section 8.1.3 on page 32</a>
0008h	HCCPARAMS	0000 0086h	<a href="#">Section 8.1.4 on page 33</a>
<b>EHCI operational registers</b>			
0020h	USBCMD	0008 0000h	<a href="#">Section 8.2.1 on page 34</a>
0024h	USBSTS	0000 1000h	<a href="#">Section 8.2.2 on page 35</a>
0028h	USBINTR	0000 0000h	<a href="#">Section 8.2.3 on page 36</a>
002Ch	FRINDEX	0000 0000h	<a href="#">Section 8.2.4 on page 36</a>
0030h	CTRLDSSEGMENT	0000 0000h	<a href="#">Section 8.2.5 on page 37</a>
0060h	CONFIGFLAG	0000 0000h	<a href="#">Section 8.2.6 on page 37</a>
0064h	PORTSC1	0000 2000h	<a href="#">Section 8.2.7 on page 38</a>
0130h	ISO PTD Done Map	0000 0000h	<a href="#">Section 8.2.8 on page 40</a>
0134h	ISO PTD Skip Map	FFFF FFFFh	<a href="#">Section 8.2.9 on page 40</a>
0138h	ISO PTD Last PTD	0000 0000h	<a href="#">Section 8.2.10 on page 40</a>
0140h	INT PTD Done Map	0000 0000h	<a href="#">Section 8.2.11 on page 41</a>
0144h	INT PTD Skip Map	FFFF FFFFh	<a href="#">Section 8.2.12 on page 41</a>
0148h	INT PTD Last PTD	0000 0000h	<a href="#">Section 8.2.13 on page 41</a>
0150h	ATL PTD Done Map	0000 0000h	<a href="#">Section 8.2.14 on page 42</a>
0154h	ATL PTD Skip Map	FFFF FFFFh	<a href="#">Section 8.2.15 on page 42</a>
0158h	ATL PTD Last PTD	0000 0000h	<a href="#">Section 8.2.16 on page 42</a>
<b>Configuration registers</b>			
0300h	HW Mode Control	0000 0000h	<a href="#">Section 8.3.1 on page 42</a>
0304h	HcChipID	0001 1761h	<a href="#">Section 8.3.2 on page 44</a>
0308h	HcScratch	0000 0000h	<a href="#">Section 8.3.3 on page 44</a>
030Ch	SW Reset	0000 0000h	<a href="#">Section 8.3.4 on page 44</a>
0330h	HcDMAConfiguration	0000 0000h	<a href="#">Section 8.3.5 on page 45</a>
0334h	HcBufferStatus	0000 0000h	<a href="#">Section 8.3.6 on page 46</a>

**Table 8: Host Controller-specific register overview...continued**

Address	Register	Reset value	References
0338h	ATL Done Timeout	0000 0000h	<a href="#">Section 8.3.7 on page 47</a>
033Ch	Memory	0000 0000h	<a href="#">Section 8.3.8 on page 47</a>
0340h	Edge Interrupt Count	0000 000Fh	<a href="#">Section 8.3.9 on page 48</a>
0344h	DMA Start address	0000 0000h	<a href="#">Section 8.3.10 on page 49</a>
0354h	Power Down Control	03E8 1BA0h	<a href="#">Section 8.3.11 on page 50</a>
<b>Interrupt registers</b>			
0310h	HcInterrupt	0000 0000h	<a href="#">Section 8.4.1 on page 52</a>
0314h	HcInterruptEnable	0000 0000h	<a href="#">Section 8.4.2 on page 54</a>
0318h	ISO IRQ Mask OR	0000 0000h	<a href="#">Section 8.4.3 on page 56</a>
031Ch	INT IRQ Mask OR	0000 0000h	<a href="#">Section 8.4.4 on page 56</a>
0320h	ATL IRQ Mask OR	0000 0000h	<a href="#">Section 8.4.5 on page 56</a>
0324h	ISO IRQ Mask AND	0000 0000h	<a href="#">Section 8.4.6 on page 56</a>
0328h	INT IRQ Mask AND	0000 0000h	<a href="#">Section 8.4.7 on page 57</a>
032Ch	ATL IRQ Mask AND	0000 0000h	<a href="#">Section 8.4.8 on page 57</a>

## 8.1 EHCI capability registers

### 8.1.1 CAPLENGTH register (R: 0000h)

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

**Table 9: CAPLENGTH register: bit description**

Bit	Symbol	Access	Value	Description
7 to 0	CAPLENGTH [7:0]	R	20h	<b>Capability Length:</b> 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 (R: 0002h)

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

**Table 10: HCIVERSION register: bit description**

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

### 8.1.3 HCSPARAMS register (R: 0004h)

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 register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	reserved							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	DPN[3:0]				reserved			P_INDICATOR
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	N_CC[3:0]				N_PCC[3:0]			
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	PRR	reserved		PPC	N_PORTS[3:0]			
<b>Reset</b>	0	0	0	1	0	0	0	1
<b>Access</b>	R	R	R	R	R	R	R	R

Table 12: HCSPARAMS register: bit description

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

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

### 8.1.4 HCCPARAMS register (R: 0008h)

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

Table 13: HCCPARAMS register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	reserved							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	reserved							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	EECP[7:0]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	IST[3:0]			reserved		ASPC	PFLF	64AC
<b>Reset</b>	1	0	0	0	0	1	1	0
<b>Access</b>	R	R	R	R	R	R	R	R

Table 14: HCCPARAMS register: bit description

Bit	Symbol	Description [1]
31 to 16	-	reserved; write logic 0
15 to 8	EECP[7:0]	<b>EHCI Extended Capabilities Pointer:</b> Default = implementation dependent. This optional field indicates the existence of a capabilities list.
7 to 4	IST[3:0]	<b>Isochronous Scheduling Threshold:</b> 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	<b>Asynchronous Scheduling Park Capability:</b> Default = implementation dependent. If this bit is set to logic 1, the Host Controller supports the park feature for high-speed queue heads in the Asynchronous Schedule.
1	PFLF	<b>Programmable Frame List Flag:</b> Default = implementation dependent. If this bit is cleared, the system software must use a frame list length of 1 024 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	64AC	<b>64-bit addressing capability:</b> This field documents the addressing range capability.

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

## 8.2 EHCI operational registers

### 8.2.1 USBCMD register (R/W: 0020h)

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 register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	1	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	LHCR	reserved [1]						RS
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 register: bit description

Bit	Symbol	Description [1]
31 to 8	-	reserved
7	LHCR	<b>Light Host Controller Reset</b> (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 1	-	reserved
0	RS	<b>Run/Stop</b> : 1 = Run, 0 = Stop. When set, the Host Controller executes the schedule.

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

### 8.2.2 USBSTS register (R/W: 0024h)

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 register: 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	1	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 register: bit description

Bit	Symbol	Description [1]
31 to 4	-	reserved; write logic 0
3	FLR	<b>Frame List Rollover:</b> The Host Controller sets this bit to logic 1 when the Frame List Index rolls over from its maximum value to zero.
2	PCD	<b>Port Change Detect:</b> 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0*.

### 8.2.3 USBINTR register (R/W: 0028h)

All the bits in this register are reserved.

### 8.2.4 FRINDEX register (R/W: 002Ch)

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. This register cannot be written unless the Host Controller is in the halted state as indicated by the HCH (HCHalted) bit. 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 register: bit allocation

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

Bit	Symbol	Description [1]
31 to 14	-	reserved
13 to 0	FRINDEX[13:0]	<b>Frame Index:</b> 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0*.

### 8.2.5 CTRLDSSEGMENT register (R/W: 0030h)

The Control Data Structure Segment (CTRLDSSEGMENT) register corresponds to the most significant address bits (bits 63 to 32) for all EHCI data structures. If the 64AC (64-bit Addressing Capability) field in HCCPARAMS is cleared, then this register is not used and software cannot write to it (reading from this register returns zero).

If the 64AC (64-bit Addressing Capability) field in HCCPARAMS is set, this register is used with link pointers to construct 64-bit addresses to EHCI control data structures. This register is concatenated with the link pointer from either the PERIODICLISTBASE, ASYNCLISTADDR, or any control data structure link field to construct a 64-bit address.

This register allows the host software to locate all control data structures within the same 4 GB memory segment.

### 8.2.6 CONFIGFLAG register (R/W: 0060h)

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

Table 21: CONFIGFLAG register: bit allocation

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

Bit	Symbol	Description [1]
31 to 1	-	reserved
0	CF	<b>Configure Flag:</b> 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0*.

### 8.2.7 PORTSC1 register (R, R/W, R/WC (field dependent): 0064h)

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 the 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: PORTSC 1 register: bit allocation

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

Bit	Symbol	Description [1]
31 to 20	-	reserved
19 to 16	PTC[3:0]	<b>Port Test Control:</b> When this field is zero, the port is not operating in a test mode. A non-zero value indicates that it is operating in test mode indicated by the value.
15 to 14	PIC[1:0]	<b>Port Indicator Control:</b> 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 <i>Universal Serial Bus Specification Rev. 2.0</i> . [2]
13	PO	<b>Port Owner:</b> 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	<b>Port Power:</b> 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]	<b>Line Status:</b> This field reflect the current logical levels of the DP (bit 11) and DM (bit 10) signal lines.
9	-	reserved
8	PR	<b>Port Reset:</b> Logic 1 means the port is in the reset state. Logic 0 means the port is not in reset. [2]
7	SUSP	<b>Suspend:</b> Logic 1 means the port is in the suspend state. Logic 0 means the port is not suspended. [2]
6	FPR	<b>Force Port Resume:</b> 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

**Table 24: PORTSC 1 register: bit description...continued**

Bit	Symbol	Description [1]
2	PED	<b>Port Enabled/Disabled:</b> Logic 1 means enable. Logic 0 means disable. [2]
1	ECSC	<b>Connect Status Change:</b> Logic 1 means change in ECCS. Logic 0 means no change. [2]
0	ECCS	<b>Current Connect Status:</b> 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 *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.

### 8.2.8 ISO PTD Done Map register (R: 0130h)

The bit description of the register is given in [Table 25](#).

**Table 25: ISO PTD Done Map register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ISO_PTD_DONE_MAP[31:0]	R	0000 0000h	<b>ISO PTD Done Map:</b> 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.

### 8.2.9 ISO PTD Skip Map register (R/W: 0134h)

[Table 26](#) shows the bit description of the register.

**Table 26: ISO PTD Skip Map register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ISO_PTD_SKIP_MAP[31:0]	R/W	FFFF FFFFh	<b>ISO PTD Skip Map:</b> 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 should not be normally set on the position indicated by NextPTDPointer.

### 8.2.10 ISO PTD Last PTD register (R/W: 0138h)

[Table 27](#) shows the bit description of the ISO PTD Last PTD register.

**Table 27: ISO PTD Last PTD register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ISO_PTD_LAST_PTD[31:0]	R/W	0000 0000h	<b>ISO PTD last PTD:</b> Last PTD of the 32 PTDs. <b>1h</b> — One PTD in ISO <b>2h</b> — Two PTDs in ISO <b>4h</b> — 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 be normally set to a higher position than any other position indicated by the NextPTDPointer from an active PTD.

### 8.2.11 INT PTD Done Map register (R: 0140h)

The bit description of the register is given in [Table 28](#).

**Table 28: INT PTD Done Map register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	INT_PTD_DONE_MAP[31:0]	R	0000 0000h	<b>INT PTD Done Map:</b> 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.

### 8.2.12 INT PTD Skip Map register (R/W: 0144h)

[Table 29](#) shows the bit description of the INT PTD Skip Map register.

**Table 29: INT PTD Skip Map register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	INT_PTD_SKIP_MAP[31:0]	R/W	FFFF FFFFh	<b>INT PTD Skip Map:</b> 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 be normally set on the position indicated by NextPTDPointer.

### 8.2.13 INT PTD Last PTD register (R/W: 0148h)

The bit description of the register is given in [Table 30](#).

**Table 30: INT PTD Last PTD register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	INT_PTD_LAST_PTD[31:0]	R/W	0000 0000h	<b>INT PTD Last PTD:</b> Last PTD of the 32 PTDs. <b>1h</b> — One PTD in INT <b>2h</b> — Two PTDs in INT <b>3h</b> — 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 be normally set to a higher position than any other position indicated by the NextPTDPointer from an active PTD.

### 8.2.14 ATL PTD Done Map register (R: 0150h)

[Table 31](#) shows the bit description of the ATL PTD Done Map register.

**Table 31: ATL PTD Done Map register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ATL_PTD_DONE_MAP[31:0]	R	0000 0000h	<b>ATL PTD Done Map:</b> 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.

### 8.2.15 ATL PTD Skip Map register (R/W: 0154h)

The bit description of the register is given in [Table 32](#).

**Table 32: ATL PTD Skip Map register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ATL_PTD_SKIP_MAP[31:0]	R/W	FFFF FFFFh	<b>ATL PTD Skip Map:</b> 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 be normally set on the position indicated by NextPTDPointer.

### 8.2.16 ATL PTD Last PTD register (R/W: 0158h)

The bit description of the ATL PTD Last PTD register is given in [Table 33](#).

**Table 33: ATL PTD Last PTD register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ATL_PTD_LAST_PTD[31:0]	R/W	0000 0000h	<b>ATL PTD Last PTD:</b> Last PTD of the 32 PTDs. <b>1h</b> — One PTD in ATL <b>2h</b> — Two PTDs in ATL <b>4h</b> — 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 be normally 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 (R/W: 0300h)

[Table 34](#) shows the bit allocation of the register.

Table 34: HW Mode Control register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	ALL_ATX_RESET	reserved [1]						
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	ANA_DIGI_OC	reserved [1]			DEV_DMA	COMN_IRQ	COMN_DMA	DATA_BUS_WIDTH
<b>Reset</b>	0	0	0	0	0	0	0	1
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	reserved	DACK_POL	DREQ_POL	reserved [1]		INTR_POL	INTR_LEVEL	GLOBAL_INTR_EN
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 register: bit description

Bit	Symbol	Description
31	ALL_ATX_RESET	<b>All ATX Reset:</b> 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	<b>Analog Digital Overcurrent:</b> This bit selects analog or digital overcurrent detection on pins OC1_N/V <sub>BUS</sub> , OC2_N and OC3_N. 0 — Digital overcurrent 1 — Analog overcurrent.
14 to 12	-	reserved; write logic 0
11	DEV_DMA	<b>Device DMA:</b> 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	<b>Common IRQ:</b> When this bit is set, DC_IRQ will be generated on the HC_IRQ pin.
9	COMN_DMA	<b>Common DMA:</b> 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	<b>Data Bus Width:</b> 0 — defines a 16-bit data bus width 1 — sets a 32-bit data bus width. <b>Remark:</b> Setting this bit will affect all the controllers on the chip (Host Controller, Peripheral Controller and OTG Controller).

Table 35: HW Mode Control register: bit description...continued

Bit	Symbol	Description
7	-	reserved; write logic 0
6	DACK_POL	<b>DACK Polarity:</b> 1 — indicates that the DACK input is active HIGH 0 — indicates active LOW.
5	DREQ_POL	<b>DREQ Polarity:</b> 1 — indicates that the DREQ output is active HIGH 0 — indicates active LOW.
4 to 3	-	reserved; write logic 0
2	INTR_POL	<b>Interrupt Polarity:</b> 0 — active LOW 1 — active HIGH.
1	INTR_LEVEL	<b>Interrupt Level:</b> 0 — INT level triggered 1 — INT is edge triggered. A pulse of certain width is generated.
0	GLOBAL_INTR_EN	<b>Global Interrupt Enable:</b> 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 (R: 0304h)

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 register: bit description

Bit	Symbol	Access	Value	Description
31 to 0	CHIPID [31:0]	R	0001 1761h	<b>Chip ID:</b> This register represents the hardware version number (0001h) and the chip ID (1761h) for the Host Controller.

### 8.3.3 HcScratch register (R/W: 0308h)

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 register: bit description

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

### 8.3.4 SW Reset register (R/W: 030Ch)

[Table 38](#) shows the bit allocation of the register.

Table 38: SW Reset register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	reserved [1]						RESET_ HC	RESET_ ALL
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 register: bit description

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

### 8.3.5 HcDMAConfiguration register (R/W: 0330h)

The bit allocation of the HcDMAConfiguration register is given in [Table 40](#).

Table 40: HcDMAConfiguration register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	DMA_COUNTER[23:16]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	DMA_COUNTER[15:8]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 <sup>[1]</sup>				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 register: bit description**

Bit	Symbol	Description
31 to 8	DMA_COUNTER [23:0]	<b>DMA Counter:</b> The number of bytes to be transferred (read or write). <b>Remark:</b> 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]	<b>DMA Burst Length:</b> <b>00</b> — Single DMA burst <b>01</b> — 4-cycle DMA burst <b>10</b> — 8-cycle DMA burst <b>11</b> — 16-cycle DMA burst.
1	ENABLE_DMA	<b>Enable DMA:</b> <b>0</b> — Terminate DMA <b>1</b> — Enable DMA.
0	DMA_READ_WRITE_SEL	<b>DMA Read or Write Select:</b> Indicates if the DMA operation is a write or read (to or from the ISP1761). <b>0</b> — DMA write to the ISP1761 internal RAM is set <b>1</b> — DMA read from the ISP1761 internal RAM.

### 8.3.6 HcBufferStatus register (R/W: 0334h)

Table 42 shows the bit allocation of the HcBufferStatus register.

**Table 42: HcBufferStatus register: bit allocation**

Bit	31	30	29	28	27	26	25	24
Symbol	reserved <sup>[1]</sup>							
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 <sup>[1]</sup>							
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

<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	reserved [1]					ISO_BUF_ FILL	INT_BUF_ FILL	ATL_BUF_ FILL
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 register: bit description**

Bit	Symbol	Description
31 to 3	-	reserved
2	ISO_BUF_ FILL	<b>ISO Buffer Filled:</b> <b>1</b> — Indicates one of the ISO PTDs is filled, and the ISO PTD area will be processed <b>0</b> — Indicates there is no PTD in this area. Therefore, processing of the ISO PTDs will be completely skipped.
1	INT_BUF_ FILL	<b>INT Buffer Filled:</b> <b>1</b> — Indicates one of the INT PTDs is filled, and the INT PTD area will be processed <b>0</b> — Indicates there is no PTD in this area. Therefore, processing of the INT PTDs will be completely skipped.
0	ATL_BUF_ FILL	<b>ATL Buffer Filled:</b> <b>1</b> — Indicates one of the ATL PTDs is filled, and the ATL PTD area will be processed <b>0</b> — Indicates there is no PTD in this area. Therefore, processing of the ATL PTDs will be completely skipped.

### 8.3.7 ATL Done Timeout register (R/W: 0338h)

The bit description of the ATL Done Timeout register is given in [Table 44](#).

**Table 44: ATL Done Timeout register: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ATL_DONE_ TIMEOUT [31:0]	R/W	0000 0000h	<b>ATL Done Timeout:</b> This register determines the ATL done timeout interrupt. This register defines the timeout in ms after which the ISP1761 asserts the INT line, if enabled. It is applicable to the ATL done PTDs only.

### 8.3.8 Memory register (R/W: 033Ch)

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: bit allocation

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

Bit	Symbol	Description
31 to 18	-	reserved
17 to 16	MEM_BANK_SEL[1:0]	<b>Memory Bank Select:</b> Up to four memory banks can be selected. For details on internal memory read description, see <a href="#">Section 7.3.1</a> . Applicable to PIO mode memory read or write data transfers only.
15 to 0	START_ADDR_MEM_READ[15:0]	<b>Start Address for Memory Read Cycles:</b> 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 (R/W: 0340h)

[Table 47](#) shows the bit allocation of the register.

Table 47: Edge Interrupt Count register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	MIN_WIDTH[7:0]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	reserved [1]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W



<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	NO_OF_CLK[15:8]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	NO_OF_CLK[7:0]							
<b>Reset</b>	0	0	0	0	1	1	1	1
<b>Access</b>	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: bit description**

Bit	Symbol	Description
31 to 24	MIN_WIDTH[7:0]	<b>Minimum Width:</b> Indicates the minimum width between two edge interrupts in $\mu$ SOFs (1 $\mu$ SOF = 125 $\mu$ s). 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]	<b>Number of Clocks:</b> 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 (W: 0344h)

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: bit allocation**

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

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

**8.3.11 Power Down Control register (R/W: 0354h)**

This register is used to turn off power to the 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: 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: bit description

Bit <sup>[1]</sup>	Symbol	Description
31 to 16	CLK_OFF_COUNTER [15:0]	<p><b>Clock Off Counter:</b> Determines the wake-up status duration after any wake-up event before the ISP1761 goes back into suspend mode. This timeout is applicable only if, during the given interval, the Host Controller is not programmed back to normal functionality.</p> <p><b>03E8h</b> — 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.</p> <p><b>FFFFh</b> — The maximum value. It determines a maximum wake-up time of 500 ms.</p> <p>The setting of this register is based on the 100 kHz <math>\pm</math> 40 % LazyClock frequency. It is a multiple of 10 <math>\mu</math>s period. In 16-bit mode, a write operation to these bits with any value will determine a fixed wake-up time of 50 ms.</p>
15 to 13	-	reserved
12	PORT3_PD	<p><b>Port 3 Pull-Down:</b> Controls port 3 pull-down resistors.</p> <p><b>0</b> — Port 3 pull-down resistors are connected in suspend</p> <p><b>1</b> — Port 3 pull-down resistors are not connected in suspend.</p>
11	PORT2_PD	<p><b>Port 2 Pull-Down:</b> Controls port 2 pull-down resistors.</p> <p><b>0</b> — Port 2 internal pull-down resistors are connected in suspend</p> <p><b>1</b> — Port 2 internal pull-down resistors are not connected in suspend.</p>
10	VBATDET_PWR	<p><b>V<sub>BAT</sub> Detector Powered:</b> Controls the power to the V<sub>BAT</sub> detector.</p> <p><b>0</b> — V<sub>BAT</sub> detector is powered or enabled in suspend</p> <p><b>1</b> — V<sub>BAT</sub> detector is not powered or disabled in suspend.</p>
9 to 6	-	reserved; write logic 0
5	BIASEN	<p><b>BIAS Circuits Powered:</b> Controls the power to internal BIAS circuits.</p> <p><b>0</b> — Internal BIAS circuits are not powered in suspend</p> <p><b>1</b> — Internal BIAS circuits are powered in suspend.</p>
4	VREG_ON	<p><b>V<sub>REG</sub> Powered:</b> Enables or disables the internal 3.3 V and 1.8 V regulators when the ISP1761 is in suspend.</p> <p><b>0</b> — Internal regulators are powered in suspend</p> <p><b>1</b> — Internal regulators are not powered in suspend.</p>
3	OC3_PWR	<p><b>OC3_N Powered:</b> Controls the powering of the overcurrent detection circuitry for port 3.</p> <p><b>0</b> — Overcurrent detection is powered on or enabled during suspend</p> <p><b>1</b> — Overcurrent detection is powered off or disabled during suspend.</p> <p>This may be useful when connecting a faulty device while the system is in standby.</p>

**Table 52: Power Down Control register: bit description...continued**

Bit <sup>[1]</sup>	Symbol	Description
2	OC2_PWR	<b>OC2_N Powered:</b> 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	<b>OC1_N Powered:</b> 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	<b>Host Controller Clock Enabled:</b> 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 (R/W: 0310h)

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 register: bit allocation**

Bit	31	30	29	28	27	26	25	24
Symbol	reserved <sup>[1]</sup>							
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 <sup>[1]</sup>							
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 <sup>[1]</sup>					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 <sup>[1]</sup>	DMA EOTINT	reserved <sup>[1]</sup>		SOFITLINT
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

**Table 54: HcInterrupt register: bit description**

Bit	Symbol	Description
31 to 11	-	reserved; write logic 0
10	OTG_IRQ	<b>OTG_IRQ:</b> Indicates that IRQ was asserted because of events present in the OTG Interrupt Latch register. <b>0</b> — No IRQ was asserted <b>1</b> — IRQ was asserted. For details, see <a href="#">Section 7.4</a> .
9	ISO_IRQ	<b>ISO_IRQ:</b> Indicates that IRQ was asserted because 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. <b>0</b> — No IRQ assertion determined by the completion of ISO PTDs <b>1</b> — IRQ asserted because of completing ISO PTDs. For details, see <a href="#">Section 7.4</a> .
8	ATL_IRQ	<b>ATL_IRQ:</b> Indicates that an IRQ was asserted because 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. <b>0</b> — No IRQ assertion determined by the completion of ATL PTDs <b>1</b> — IRQ asserted because of completing ATL PTD. For details, see <a href="#">Section 7.4</a> .
7	INT_IRQ	<b>INT_IRQ:</b> Indicates that an IRQ was asserted because 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. <b>0</b> — No IRQ assertion determined by the completion of INT PTDs <b>1</b> — IRQ asserted because of completing INT PTD. For details, see <a href="#">Section 7.4</a> .
6	CLKREADY	<b>Clock Ready:</b> Indicates that IRQ was asserted as the internal clock signals are running stable. Useful after a power-on or wake-up cycle. <b>0</b> — No CLKREADY event has occurred <b>1</b> — IRQ generated because of a CLKREADY event.

**Table 54: HcInterrupt register: bit description...continued**

Bit	Symbol	Description
5	HCSUSP	<b>Host Controller Suspend:</b> Indicates that the Host Controller has entered suspend mode. <b>0</b> — No IRQ generated because of the Host Controller entering suspend mode <b>1</b> — IRQ generated because of the Host Controller entering 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 logic 0
3	DMAEOT INT	<b>DMA EOT Interrupt:</b> Indicates DMA transfer completion. <b>0</b> — DMA transfer is not complete <b>1</b> — IRQ asserted because the DMA transfer is complete.
2 to 1	-	reserved; write logic 0
0	SOFITLINT	<b>SOT ITL Interrupt:</b> <b>0</b> — No SOF event has occurred <b>1</b> — An SOF event has occurred.

**8.4.2 HcInterruptEnable register (R/W: 0314h)**

This register allows enabling or disabling of the IRQ generation because of various events as described in [Table 55](#).

**Table 55: HcInterruptEnable register: 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	CLK READY_ E	HCSUSP_ E	reserved [1]	DMAEOT INT_ E	reserved [1]		SOFITLINT_ 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	

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

Table 56: HcInterruptEnable register: bit description

Bit	Symbol	Description
31 to 11	-	reserved; write logic 0
10	OTG_IRQ_E	<b>OTG_IRQ Enable:</b> Controls the IRQ assertion because of events present in the OTG Interrupt Latch register. <b>0</b> — No IRQ will be asserted <b>1</b> — IRQ will be asserted. For details, see <a href="#">Section 7.4</a> .
9	ISO_IRQ_E	<b>ISO IRQ Enable:</b> Controls the IRQ assertion because of completing one or more ISO PTDs matching the ISO IRQ Mask AND or ISO IRQ Mask OR register bits combination. <b>0</b> — No IRQ will be asserted because of completing ISO PTDs <b>1</b> — IRQ will be asserted. For details, see <a href="#">Section 7.4</a> .
8	ATL_IRQ_E	<b>ATL IRQ Enable:</b> Controls the IRQ assertion because of completing one or more ATL PTDs matching the ATL IRQ Mask AND or ATL IRQ Mask OR register bits combination. <b>0</b> — No IRQ will be asserted because of completing ATL PTDs <b>1</b> — IRQ will be asserted. For details, see <a href="#">Section 7.4</a> .
7	INT_IRQ_E	<b>INT IRQ Enable:</b> Controls the IRQ assertion because of completing one or more INT PTDs matching the INT IRQ Mask AND or INT IRQ Mask OR register bits combination. <b>0</b> — No IRQ will be asserted because of completing INT PTDs <b>1</b> — IRQ will be asserted. For details, see <a href="#">Section 7.4</a> .
6	CLKREADY_E	<b>Clock Ready Enable:</b> Enables the IRQ assertion when internal clock signals are running stable. Useful after power-on or wake-up. <b>0</b> — No IRQ will be generated after a CLKREADY_E event has occurred <b>1</b> — IRQ will be generated after a CLKREADY_E event.
5	HCSUSP_E	<b>Host Controller Suspend Enable:</b> Enables the IRQ generation when the Host Controller enters suspend mode. <b>0</b> — No IRQ will be generated because of the Host Controller entering suspend mode <b>1</b> — IRQ will be generated at the Host Controller entering suspend mode.
4	-	reserved; write logic 0
3	DMAEOT INT_E	<b>DMA EOT Interrupt Enable:</b> Controls assertion of IRQ on the DMA transfer completion. <b>0</b> — No IRQ will be generated after the DMA transfer is completed <b>1</b> — IRQ will be asserted because of the DMA transfer completion.
2 to 1	-	reserved; write logic 0
0	SOFITLINT_E	<b>SOT ITL Interrupt Enable:</b> Controls the IRQ generation at every SOF occurrence. <b>0</b> — No IRQ will be generated on an SOF occurrence <b>1</b> — IRQ will be asserted at every SOF.

### 8.4.3 ISO IRQ MASK OR register (R/W: 0318h)

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: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ISO_IRQ_MASK_OR [31:0]	R/W	0000 0000h	<b>ISO IRQ Mask OR:</b> Represents a direct map for ISO PTDs 31 to 0. <b>0</b> — No OR condition defined between ISO PTDs <b>1</b> — 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 (R/W: 031Ch)

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: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	INT_IRQ_MASK_OR [31:0]	R/W	0000 0000h	<b>INT IRQ Mask OR:</b> Represents a direct map for INT PTDs 31 to 0. <b>0</b> — No OR condition defined between INT PTDs 31 to 0 <b>1</b> — 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 (R/W: 0320h)

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: bit description**

Bit	Symbol	Access	Value	Description
31 to 0	ATL_IRQ_MASK_OR [31:0]	R/W	0000 0000h	<b>ATL IRQ Mask OR:</b> Represents a direct map for ATL PTDs 31 to 0. <b>0</b> — No OR condition defined between the ATL PTDs <b>1</b> — 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 (R/W: 0324h)

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: bit description**

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

#### 8.4.7 INT IRQ MASK AND register (R/W: 0328h)

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: bit description**

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

#### 8.4.8 ATL IRQ MASK AND register (R/W: 032Ch)

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 SAND register: bit description**

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

### 8.5 Philips Transfer Descriptor

The standard EHCI data structures as described in *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 the EHCI data structures that are optimized for the ISP1761, while keeping the architecture of the EHCI data structures the same. This is because the ISP1761 is a slave Host Controller and has no bus master capability.

EHCI manages schedules in two lists: periodic and asynchronous. The data structures are designed to provide the maximum flexibility required by USB, minimize memory traffic, and 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 root of the periodic schedule—the PERIODICLISTBASE register—is the physical memory base address of the periodic frame list. The periodic frame list is an array memory pointer. The objects referenced from the frame list must be valid schedule data structures. The asynchronous list base is also a common list of queue heads (endpoints) that are served in a schedule. These endpoint data structures are further linked to the EHCI transfer descriptor that is the valid schedule (queue PTD).

The Periodic Schedule Enable (ISO\_BUF\_FULL and INT\_BUF\_FULL) or Asynchronous Schedule Enable (ATL\_BUF\_FULL) bits can enable traversal to these lists. Enabling a list indicates the presence of valid schedule in the list. The system software starts at these points, schedules the first transfer inside the shared memory of the ISP1761, and sets up the ATL, INTL or ITL bit corresponding to the type of transfer scheduled in the shared memory.

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 deallocated 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 the EHCI data structure, until it reaches the terminate bit in a microframe. If a schedule is enabled, the Host Controller starts executing from the ISO schedule, before it goes to the INTL schedule, and then to the ATL schedule.

The EHCI periodic and asynchronous lists are traversed by the software according to the EHCI traversal rule, and executed only from the asynchronous schedule after it encounters the end of the periodic schedule. The Host Controller traverses the ISO, INTL and ATL schedules. It fetches the element and begins traversing the graph of linked schedule data structures.

The last bit identifies the end of the schedule for each type of transfer, indicating the rest of the channels are empty. Once a transition is completed, the Host Controller executes from the next transfer descriptor in the schedule until the end of the microframe.

The completion of a transfer is indicated to the software by the interrupt that can be grouped over the 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 group and individual PTDs that can interrupt the CPU for a schedule, when the logical conditions of the done bit is true in the shared memory that completes the interrupt.

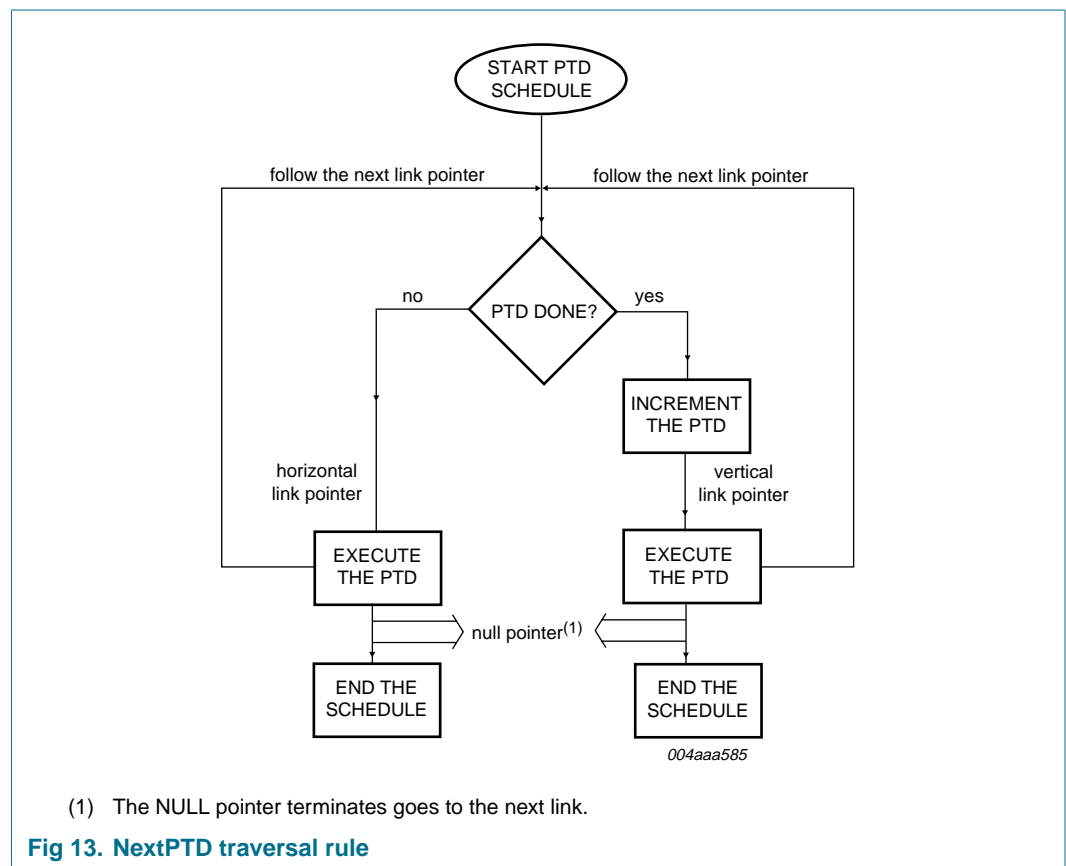
Interrupts are of four types and the latency can be programmed in multiples of  $\mu$ SOF (125  $\mu$ s):

- ISO interrupt
- INTL interrupt
- ATL Interrupt
- SOF—start of frame interrupt for the data transfer.

A static PTD that schedules inside the ISP1761 shared memory allows using the NextPTD mechanism that will enable the Host Controller driver to schedule the multiple PTDs that are of single endpoint and reduce the interrupt to the CPU.

The NextPTD traversal rules defined by the ISP1761 hardware are:

1. Start the ATL header traversal.
2. If the current PTD is active and not done, perform the transaction.
3. Follow the next link pointer.
4. If PTD is not active and done, jump to the next PTD.
5. If the next link pointer is NULL, it means the end of the traversal.



### 8.5.1 High-speed bulk IN and OUT, QHA

Table 63 shows the bit allocation of the high-speed bulk IN and OUT, Queue Head Asynchronous (QHA)<sup>1</sup>.

**Table 63: High-speed bulk IN and OUT, QHA: bit allocation**

<b>Bit</b>	<b>63</b>	<b>62</b>	<b>61</b>	<b>60</b>	<b>59</b>	<b>58</b>	<b>57</b>	<b>56</b>	<b>55</b>	<b>54</b>	<b>53</b>	<b>52</b>	<b>51</b>	<b>50</b>	<b>49</b>	<b>48</b>	<b>47</b>	<b>46</b>	<b>45</b>	<b>44</b>	<b>43</b>	<b>42</b>	<b>41</b>	<b>40</b>	<b>39</b>	<b>38</b>	<b>37</b>	<b>36</b>	<b>35</b>	<b>34</b>	<b>33</b>	<b>32</b>
<b>DW7</b>	reserved																															
<b>DW5</b>	reserved																															
<b>DW3</b>	A	H	B	X	[1]	P	D T	Cerr [1:0]	NakCnt[3:0]			reserved			NrBytesTransferred[14:0] (32 kB for high-speed)																	
<b>DW1</b>	reserved													S	EP Type [1:0]	Token [1:0]	DeviceAddress[6:0]				EndPt[3:0] 31 to 34											
<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>DW6</b>	reserved																															
<b>DW4</b>	reserved														J	NextPTDPointer[4:0]																
<b>DW2</b>	reserved		RL[3:0]			[1]	DataStartAddress[15:0]										reserved															
<b>DW0</b>	[2]	Mult [1:0]		MaxPacketLength[10:0]							NrBytesToTransfer[14:0] (32 kB for high-speed)										[1]	V										

[1] Reserved.

[2] EndPt[0].

1. Patent-pending: High-speed bulk IN and OUT, QHA.

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

Bit	Symbol	Access	Description
<b>DW7</b>			
63 to 32	reserved	-	-
<b>DW6</b>			
31 to 0	reserved	-	-
<b>DW5</b>			
63 to 32	reserved	-	-
<b>DW4</b>			
31 to 6	reserved	-	0; not applicable for QHA.
5	J	<b>SW</b> — writes	<b>Jump:</b> <b>0</b> — To increment the PTD pointer <b>1</b> — To enable the next PTD branching.
4 to 0	NextPTDPointer [4:0]	<b>SW</b> — writes	<b>Next PTD Counter:</b> Next PTD branching assigned by the PTD pointer.
<b>DW3</b>			
63	A	<b>SW</b> — sets <b>HW</b> — resets	<b>Active:</b> Write the same value as that in V.
62	H	<b>HW</b> — writes	<b>Halt:</b> This bit correspond to the Halt bit of the Status field of QH.
61	B	<b>HW</b> — writes	<b>Babble:</b> This bit correspond to the Babble Detected bit in the Status field of the iTD, SiTD or QH. <b>1</b> — When babbling is detected, A and V are set to 0.
60	X	<b>HW</b> — writes	<b>Error:</b> This bit corresponds to the Transaction Error bit in the Status field of iTD, SiTD or QH (Exec_Trans, the signal name is xacterr). <b>0</b> — No PID error. <b>1</b> — 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.
59	reserved	-	-
58	P	<b>HW</b> — writes	<b>Ping:</b> For high-speed transactions, this bit corresponds to the Ping state bit in the Status field of a QH. <b>0</b> — Ping is not set. <b>1</b> — Ping is set. Software sets this bit to 0.
57	DT	<b>HW</b> — updates <b>SW</b> — writes	<b>Data Toggle:</b> 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.
56 to 55	Cerr[1:0]	<b>HW</b> — writes <b>SW</b> — writes	<b>Error Counter.</b> This field corresponds to the Cerr[1:0] field in QH. The default value of this field is zero for isochronous transactions. <b>00</b> — The transaction will not retry. <b>11</b> — The transaction will retry three times. Hardware will decrement these values. When the transaction has tried three times, X error will be updated.
54 to 51	NakCnt[3:0]	<b>HW</b> — writes <b>SW</b> — writes	<b>NAK Counter.</b> This field corresponds to the NakCnt field in QH. 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 ACKed.
50 to 47	reserved	-	-

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

Bit	Symbol	Access	Description
46 to 32	NrBytesTransferred [14:0]	<b>HW</b> — writes <b>SW</b> — writes 0000	<b>Number of Bytes Transferred:</b> 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.
<b>DW2</b>			
31 to 29	reserved	-	Set to 0 for QHA.
28 to 25	RL[3:0]	<b>SW</b> — writes	<b>Reload:</b> 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 QHA.
23 to 8	DataStartAddress [15:0]	<b>SW</b> — writes	<b>Data Start Address:</b> This is the start address for the 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	-	-
<b>DW1</b>			
63 to 47	reserved	-	Always 0 for QHA.
46	S	<b>SW</b> — writes	This bit indicates whether a split transaction has to be executed: <b>0</b> — High-speed transaction <b>1</b> — Split transaction.
45 to 44	EPTYPE[1:0]	<b>SW</b> — writes	<b>Transaction type:</b> <b>00</b> — Control <b>10</b> — Bulk.
43 to 42	Token[1:0]	<b>SW</b> — writes	<b>Token:</b> Identifies the token Packet Identifier (PID) for this transaction: <b>00</b> — OUT <b>01</b> — IN <b>10</b> — SETUP <b>11</b> — PING (written by hardware only).
41 to 35	DeviceAddress[6:0]	<b>SW</b> — writes	<b>Device Address:</b> This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	<b>SW</b> — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
<b>DW0</b>			
31	EndPt[0]	<b>SW</b> — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
30 to 29	Mult[1:0]	<b>SW</b> — writes	<b>Multiplier:</b> 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.  For QHA, this is a copy of the Async Schedule Park mode count, if the Async Schedule Park mode is enabled. These EHCI registers need to be set to reflect multiple cycles. Applicable for high-speed only.  Set this field to 01b. You can also set it to 11b and 10b depending on your application. 00b is undefined.

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

Bit	Symbol	Access	Description
28 to 18	MaxPacketLength [10:0]	<b>SW</b> — writes	<b>Maximum Packet Length:</b> 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 B. The maximum packet size for the isochronous transfer is also variable at any whole number.
17 to 3	NrBytesToTransfer [14:0]	<b>SW</b> — writes	<b>Number of Bytes to Transfer:</b> 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).
2 to 1	reserved	-	-
0	V	<b>SW</b> — sets <b>HW</b> — resets	<b>Valid:</b> <b>0</b> — This bit is deactivated when the entire PTD is executed—across $\mu$ SOF and SOF—or when a fatal error is encountered. <b>1</b> — Software updates to one when there is payload to be sent or received even across ms boundary. The current PTD is active.

### 8.5.2 High-speed isochronous IN and OUT, iTD

Table 65 shows the bit allocation of the high-speed isochronous IN and OUT, isochronous Transfer Descriptor (iTID)<sup>2</sup>.

**Table 65: High-speed isochronous IN and OUT, iTD: 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[7:0]									
DW5	ISOIN_2[7:0]							ISOIN_1[11:0]														ISOIN_0[11:0]										
DW3	A	H	B	reserved											NrBytesTransferred[14:0] (32 kB for high-speed)																	
DW1	reserved													S	EP Type [1:0]	Token [1:0]	DeviceAddress[6:0]						EndPt[3:0] 34 to 31									
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[3:0]							
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]									
DW0	[2]	Mult [1:0]		MaxPacketLength[10:0]										NrBytesToTransfer[14:0] (32 kB for high-speed)														[1]	V			

[1] Reserved.

[2] EndPt[0].

2. Patent-pending: High-speed isochronous IN and OUT, iTD.



**Table 66: High-speed isochronous IN and OUT, iTD: bit description**

Bit	Symbol	Access	Description
<b>DW7</b>			
63 to 52	ISOIN_7[11:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF7, if $\mu$ SA[7] is set to 1 and frame number is correct.
51 to 40	ISOIN_6[11:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF6, if $\mu$ SA[6] is set to 1 and frame number is correct.
39 to 32	ISOIN_5[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF5 (bits 11 to 4), if $\mu$ SA[5] is set to 1 and frame number is correct.
<b>DW6</b>			
31 to 28	ISOIN_5[3:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF5 (bits 3 to 0), if $\mu$ SA[5] is set to 1 and frame number is correct.
27 to 16	ISOIN_4[11:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF4, if $\mu$ SA[4] is set to 1 and frame number is correct.
15 to 4	ISOIN_3[11:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF3, if $\mu$ SA[3] is set to 1 and frame number is correct.
3 to 0	ISOIN_2[3:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF2 (bits 11 to 8), if $\mu$ SA[2] is set to 1 and frame number is correct.
<b>DW5</b>			
63 to 56	ISOIN_2[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF2 (bits 7 to 0), if $\mu$ SA[2] is set to 1 and frame number is correct.
55 to 44	ISOIN_1[11:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF1, if $\mu$ SA[1] is set to 1 and frame number is correct.
43 to 32	ISOIN_0[11:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF0, if $\mu$ SA[0] is set to 1 and frame number is correct.
<b>DW4</b>			
31 to 29	Status7[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF7
28 to 26	Status6[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF6
25 to 23	Status5[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF5
22 to 20	Status4[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF4
19 to 17	Status3[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF3
16 to 14	Status2[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF2
13 to 11	Status1[2:0]	<b>HW</b> — writes	ISO IN or OUT status at $\mu$ SOF1
10 to 8	Status0[2:0]	<b>HW</b> — writes	Status of the payload on the USB bus for this $\mu$ SOF after ISO has been delivered. <b>Bit 0</b> — Transaction Error (IN and OUT) <b>Bit 1</b> — Babble (IN token only) <b>Bit 2</b> — underrun (OUT token only).
7 to 0	$\mu$ SA[7:0]	<b>SW</b> — writes (0 => 1) <b>HW</b> — writes (1 => 0) After processing	<b><math>\mu</math>SOF Active:</b> When the frame number of bits DW1[7:3] match the frame number of USB bus, these bits are checked for 1 before they are sent for $\mu$ SOF. For example: If $\mu$ SA[7:0] = 1, 1, 1, 1, 1, 1, 1, 1: send ISO every $\mu$ SOF of the entire ms. If $\mu$ SA[7:0] = 0, 1, 0, 1, 0, 1, 0, 1: send ISO only on $\mu$ SOF0, $\mu$ SOF2, $\mu$ SOF4 and $\mu$ SOF6.
<b>DW3</b>			
63	A	<b>SW</b> — sets	<b>Active:</b> This bit is the same as the Valid bit.
62	H	<b>HW</b> — writes	<b>Halt:</b> Only one bit for the entire ms. When this bit is set, the Valid bit is reset. The device decides to stall an endpoint.

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

Bit	Symbol	Access	Description
61	B	HW — writes	<b>Babble:</b> Not applicable here.
60 to 47	reserved	-	Set to 0 for isochronous.
46 to 32	NrBytesTransferred [14:0]	HW — writes	<b>Number of Bytes Transferred:</b> 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 per PTD.
<b>DW2</b>			
31 to 24	reserved	-	Set to 0 for isochronous.
23 to 8	DataStartAddress [15:0]	SW — writes	<b>Data Start Address:</b> This is the start address for the 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	<b>Bits 2 to 0</b> — Don't care <b>Bits 7 to 3</b> — Frame number that this PTD will be sent for ISO OUT or IN.
<b>DW1</b>			
63 to 47	reserved	-	-
46	S	SW — writes	This bit indicates whether a split transaction has to be executed. <b>0</b> — High-speed transaction <b>1</b> — Split transaction.
45 to 44	EPType[1:0]	SW — writes	<b>Endpoint type:</b> <b>01</b> — Isochronous.
43 to 42	Token[1:0]	SW — writes	<b>Token:</b> This field indicates the token PID for this transaction: <b>00</b> — OUT <b>01</b> — IN.
41 to 35	DeviceAddress[6:0]	SW — writes	<b>Device Address:</b> 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	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
<b>DW0</b>			
31	EndPt[0]	SW — writes	<b>Endpoint:</b> 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 isochronous OUT and IN: <b>If Mult[1:0] is 01</b> — Data Toggle is Data0 <b>If Mult[1:0] is 10</b> — Data Toggle is Data1 <b>If Mult[1:0] is 11</b> — Data Toggle is Data2, and so on. For details, refer to <i>Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0</i> , Appendix D.
28 to 18	MaxPacketLength [10:0]	SW — writes	<b>Maximum Packet Length:</b> 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 B. The maximum packet size for the isochronous transfer is also variable at any whole number.

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

Bit	Symbol	Access	Description
17 to 3	NrBytesToTransfer [14:0]	<b>SW</b> — writes	<b>Number of Bytes Transferred:</b> 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).
2 to 1	reserved	-	-
0	V	<b>HW</b> — resets <b>SW</b> — sets	<b>0</b> — This bit is deactivated when the entire PTD is executed—across $\mu$ SOF and SOF—or when a fatal error is encountered. <b>1</b> — Software updates to one when there is payload to be sent or received even across ms boundary. The current PTD is active.

### 8.5.3 High-speed interrupt IN and OUT, QHP

Table 67 shows the bit allocation of the high-speed interrupt IN and OUT, Queue Head Periodic (QHP)<sup>3</sup>.

**Table 67: High-speed interrupt IN and OUT, QHP: 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[7:0]									
DW5	INT_IN_2[7:0]							INT_IN_1[11:0]														INT_IN_0[11:0]										
DW3	A	H	reserved				D T	Cerr [1:0]	reserved							NrBytesTransferred[14:0] (32 kB for high-speed)																
DW1	reserved													S	EP Type [1:0]	Token [1:0]	DeviceAddress[6:0]					EndPt[3:0] 31 to 34										
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[3:0]							
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]										
DW0	[2]	Mult [1:0]		MaxPacketLength[10:0]										NrBytesToTransfer[14:0] (32 kB for high-speed)														[1]	V			

[1] Reserved.

[2] EndPt[0].

3. Patent-pending: High-speed interrupt IN and OUT, QHP.

**Table 68: High-speed interrupt IN and OUT, QHP: bit description**

Bit	Symbol	Access	Description
<b>DW7</b>			
63 to 52	INT_IN_7[[11:0]	HW — writes	Bytes received during $\mu$ SOF7, if $\mu$ SA[7] is set to 1 and frame number is correct.
51 to 40	INT_IN_6[11:0]	HW — writes	Bytes received during $\mu$ SOF6, if $\mu$ SA[6] is set to 1 and frame number is correct.
39 to 32	INT_IN_5[7:0]	HW — writes	Bytes received during $\mu$ SOF5 (bits 7 to 0), if $\mu$ SA[5] is set to 1 and frame number is correct.
<b>DW6</b>			
31 to 28	INT_IN_5[3:0]	HW — writes	Bytes received during $\mu$ SOF5 (bits 3 to 0), if $\mu$ SA[5] is set to 1 and frame number is correct.
27 to 16	INT_IN_4[11:0]	HW — writes	Bytes received during $\mu$ SOF4, if $\mu$ SA[4] is set to 1 and frame number is correct.
15 to 4	INT_IN_3[11:0]	HW — writes	Bytes received during $\mu$ SOF3, if $\mu$ SA[3] is set to 1 and frame number is correct.
3 to 0	INT_IN_2[3:0]	HW — writes	Bytes received during $\mu$ SOF2 (bits 11 to 8), if $\mu$ SA[2] is set to 1 and frame number is correct.
<b>DW5</b>			
63 to 56	INT_IN_2[7:0]	HW — writes	Bytes received during $\mu$ SOF2 (bits 7 to 0), if $\mu$ SA[2] is set to 1 and frame number is correct.
55 to 44	INT_IN_1[11:0]	HW — writes	Bytes received during $\mu$ SOF1, if $\mu$ SA[1] is set to 1 and frame number is correct.
43 to 32	INT_IN_0[11:0]	HW — writes	Bytes received during $\mu$ SOF0, if $\mu$ SA[0] is set to 1 and frame number is correct.
<b>DW4</b>		<b>INT OUT or IN</b>	
31 to 29	Status7[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF7
28 to 26	Status6[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF6
25 to 23	Status5[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF5
22 to 20	Status4[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF4
19 to 17	Status3[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF3
16 to 14	Status2[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF2
13 to 11	Status1[2:0]	HW — writes	INT IN or OUT status of $\mu$ SOF1
10 to 8	Status0[2:0]	HW — writes	Status of the payload on the USB bus for this $\mu$ SOF after INT has been delivered. <b>Bit 0</b> — Transaction Error (IN and OUT) <b>Bit 1</b> — Babble (IN token only) <b>Bit 2</b> — underrun (OUT token only).
7 to 0	$\mu$ SA[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 $\mu$ SOF. For example: When $\mu$ SA[7:0] = 1, 1, 1, 1, 1, 1, 1, 1: send INT for every $\mu$ SOF of the entire ms. When $\mu$ SA[7:0] = 0, 1, 0, 1, 0, 1, 0, 1: send INT for $\mu$ SOF0, $\mu$ SOF2, $\mu$ SOF4 and $\mu$ SOF6. When $\mu$ SA[7:0] = 1, 0, 0, 0, 1, 0, 0, 0 = send INT for every fourth $\mu$ SOF.
<b>DW3</b>			
63	A	HW — writes SW — writes	<b>Active:</b> Write the same value as that in V.

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

Bit	Symbol	Access	Description																																								
62	H	HW — writes	<b>Halt:</b> Transaction is halted.																																								
61 to 58	reserved	-	-																																								
57	DT	HW — writes SW — writes	<b>Data Toggle:</b> 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	<b>Error Counter:</b> This field corresponds to the Cerr[1:0] field in the QH. The default value of this field is zero for isochronous transactions.																																								
54 to 47	reserved	-	-																																								
46 to 32	NrBytes Transferred [14:0]	HW — writes	<b>Number of Bytes Transferred:</b> 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.																																								
<b>DW2</b>																																											
31 to 24	reserved	-	-																																								
23 to 8	DataStart Address [15:0]	SW — writes	<b>Data Start Address:</b> This is the start address for the 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 for ms-based polling. The INT polling rate is defined as $2^{(b-1)} \mu\text{SOF}$ , 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.																																								
			<table border="1"> <thead> <tr> <th>b</th> <th>Rate</th> <th>μFrame[7:3]</th> <th>μSA[7:0]</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1 μSOF</td> <td>0 0000</td> <td>1111 1111</td> </tr> <tr> <td>2</td> <td>2 μSOF</td> <td>0 0000</td> <td>1010 1010 or 0101 0101</td> </tr> <tr> <td>3</td> <td>4 μSOF</td> <td>0 0000</td> <td>any 2 bits set</td> </tr> <tr> <td>4</td> <td>1 ms</td> <td>0 0000</td> <td>any 1 bit set</td> </tr> <tr> <td>5</td> <td>2 ms</td> <td>0 0001</td> <td>any 1 bit set</td> </tr> <tr> <td>6</td> <td>4 ms</td> <td>0 0010 to 0 0011</td> <td>any 1 bit set</td> </tr> <tr> <td>7</td> <td>8 ms</td> <td>0 0100 to 0 0111</td> <td>any 1 bit set</td> </tr> <tr> <td>8</td> <td>16 ms</td> <td>0 1000 to 0 1111</td> <td>any 1 bit set</td> </tr> <tr> <td>9</td> <td>32 ms</td> <td>1 0000 to 1 1111</td> <td>any 1 bit set</td> </tr> </tbody> </table>	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
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																																								
<b>DW1</b>																																											
63 to 47	reserved	-	-																																								
46	S	SW — writes	This bit indicates if a split transaction has to be executed: <b>0</b> — High-speed transaction <b>1</b> — Split transaction.																																								
45 to 44	EPTtype[1:0]	SW — writes	<b>Endpoint type:</b> <b>11</b> — Interrupt.																																								
43 to 42	Token[1:0]	SW — writes	<b>Token:</b> This field indicates the token PID for this transaction: <b>00</b> — OUT <b>01</b> — IN.																																								

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

Bit	Symbol	Access	Description
41 to 35	DeviceAddress [6:0]	SW — writes	<b>Device Address:</b> 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	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
<b>DW0</b>			
31	EndPt[0]	SW — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
30 to 29	Mult[1:0]	SW — writes	<b>Multiplier:</b> 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	<b>Maximum Packet Length:</b> 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	<b>Number of Bytes to Transfer:</b> This field indicates the number of bytes can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB).
2 to 1	reserved	-	-
0	V	SW — sets HW — resets	<b>Valid:</b> <b>0</b> — This bit is deactivated when the entire PTD is executed—across $\mu$ SOF and SOF—or when a fatal error is encountered. <b>1</b> — Software updates to one when there is payload to be sent or received even across ms boundary. The current PTD is active.

### 8.5.4 Start and complete split for bulk, QHA-SS/CS

Table 69 shows the bit allocation of start split and complete split for bulk, Queue Head Asynchronous Start Split and Complete Split (QHA-SS/CS)<sup>4</sup>.

**Table 69: Start and complete split for bulk, QHASS/CS: 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	S C	[1]	D T	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[3:0] (31 to 34)								
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] (32 kB for high-speed)														[1]	V				

[1] Reserved.

[2] EndPt[0].

4. Patent-pending: Start and complete split for bulk, QHA-SS/CS.



Table 70: Start and complete split for bulk, QHASS/CS: bit description

Bit	Symbol	Access	Description
<b>DW7</b>			
63 to 32	reserved	-	-
<b>DW6</b>			
31 to 0	reserved	-	-
<b>DW5</b>			
63 to 32	reserved	-	-
<b>DW4</b>			
31 to 6	reserved	-	-
5	J	<b>SW</b> — writes	<b>0</b> — To increment the PTD pointer <b>1</b> — To enable the next PTD branching.
4 to 0	NextPTDPointer [1:0]	<b>SW</b> — writes	<b>Next PTD Pointer:</b> Next PTD branching assigned by the PTD pointer.
<b>DW3</b>			
63	A	<b>SW</b> — sets <b>HW</b> — resets	<b>Active:</b> Write the same value as that in V.
62	H	<b>HW</b> — writes	<b>Halt:</b> This bit correspond to the Halt bit of the Status field of QH.
61	B	<b>HW</b> — writes	<b>Babble:</b> This bit correspond to the Babble Detected bit in the Status field of the iTD, SiTD or QH. <b>1</b> — when babbling is detected, A and V are set to 0.
60	X		<b>Transaction Error:</b> This bit corresponds to the Transaction Error bit in the status field.
59	SC	<b>SW</b> — writes 0 <b>HW</b> — updates	<b>Start/Complete:</b> <b>0</b> — Start split <b>1</b> — Complete split.
58	reserved	-	-
57	DT	<b>HW</b> — writes <b>SW</b> — writes	<b>Data Toggle:</b> Set the Data Toggle bit to start for the PTD.
56 to 55	Cerr[1:0]	<b>HW</b> — updates <b>SW</b> — writes	<b>Error Counter:</b> This field contains the error count for start and complete split (QHASS). 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 <i>Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0</i> , Section 4.12.1.2.  If retry has insufficient time at the beginning of a new SOF, the first PTD must be this retry. This can be accomplished by if aperiodic PTD is not advanced.
54 to 51	NakCnt[3:0]	<b>HW</b> — writes <b>SW</b> — writes	<b>NAK Counter:</b> 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	-	-
46 to 32	NrBytesTransferred [14:0]	<b>HW</b> — writes	<b>Number of Bytes Transferred:</b> This field indicates the number of bytes sent or received for this transaction.
<b>DW2</b>			
31 to 29	reserved	-	-

Table 70: Start and complete split for bulk, QHASS/CS: bit description...continued

Bit	Symbol	Access	Description															
28 to 25	RL[3:0]	SW — writes	<b>Reload.</b> 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	<b>Data Start Address:</b> This is the start address for the 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	-	-															
<b>DW1</b>																		
63 to 57	HubAddress[6:0]	SW — writes	<b>Hub Address:</b> This indicates the hub address. Zero for the internal or embedded hub.															
56 to 50	PortNumber[6:0]	SW — writes	<b>Port Number:</b> 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. The following applies to start split and complete split only.															
			<table border="1"> <thead> <tr> <th>Bulk</th> <th>Control</th> <th>S</th> <th>E</th> <th>Remarks</th> </tr> </thead> <tbody> <tr> <td>I/O</td> <td>I/O</td> <td>1</td> <td>0</td> <td>low-speed</td> </tr> <tr> <td>I/O</td> <td>I/O</td> <td>0</td> <td>0</td> <td>full-speed</td> </tr> </tbody> </table>	Bulk	Control	S	E	Remarks	I/O	I/O	1	0	low-speed	I/O	I/O	0	0	full-speed
Bulk	Control	S	E	Remarks														
I/O	I/O	1	0	low-speed														
I/O	I/O	0	0	full-speed														
47	reserved	-	-															
46	S	SW — writes	This bit indicates whether a split transaction has to be executed: <b>0</b> — High-speed transaction <b>1</b> — Split transaction.															
45 to 44	EPTYPE[1:0]	SW — writes	<b>Endpoint Type:</b> <b>00</b> — Control <b>10</b> — Bulk.															
43 to 42	Token[1:0]	SW — writes	<b>Token:</b> This field indicates the PID for this transaction. <b>00</b> — OUT <b>01</b> — IN <b>10</b> — SETUP.															
41 to 35	DeviceAddress [6:0]	SW — writes	<b>Device Address:</b> 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	<b>Endpoint:</b> This is the USB address of the endpoint within the function.															
<b>DW0</b>																		
31	EndPt[0]	SW — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.															
30 to 29	reserved	-	-															
28 to 18	MaximumPacket Length[10:0]	SW — writes	<b>Maximum Packet Length:</b> 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 B as defined in the <i>Universal Serial Bus Specification Rev. 2.0</i> .															

Table 70: Start and complete split for bulk, QHASS/CS: bit description...continued

Bit	Symbol	Access	Description
17 to 3	NrBytesToTransfer [14:0]	SW — writes	<b>Number of Bytes to Transfer:</b> 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	<b>Valid:</b> <b>0</b> — This bit is deactivated when the entire PTD is executed—across $\mu$ SOF and SOF—or when a fatal error is encountered. <b>1</b> — Software updates to one when there is payload to be sent or received even across ms boundary. The current PTD is active.

### 8.5.5 Start and complete split for isochronous, SiTD

Table 71 shows the bit allocation for start and complete split for isochronous, Split isochronous Transfer Descriptor (SiTD)<sup>5</sup>.

**Table 71: Start and complete split for isochronous, SiTD: 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] [2]													
DW3	A	H	B	X	S	[1]	D	reserved														NrBytesTransferred[11:0]										
DW1	HubAddress[6:0]						PortNumber[6:0]						reserved		S	EP Type [1:0]	Token [1:0]	DeviceAddress[6:0]				EndPt[3:0] (31 to 34)										
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	[3]	[1]	TT_MPS_Len[10:0]						NrBytesToTransfer[14:0] (1 kB for full-speed)										[1]	V												

[1] Reserved.

[2] Note the change in the position of USCS[7:0] and NrBytesReceived\_CS\_IN.

[3] EndPt[0].

5. Patent-pending: Start and complete split for isochronous, SiTD.

**Table 72: Start and complete split for isochronous, SiTD: bit description**

Bit	Symbol	Access	Description
<b>DW7</b>			
63 to 40	reserved	-	-
39 to 32	ISO_IN_7[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF7, if $\mu$ SA[7] is set to 1 and frame number is correct.
<b>DW6</b>			
31 to 24	ISO_IN_6[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF6, if $\mu$ SA[6] is set to 1 and frame number is correct.
23 to 16	ISO_IN_5[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF5, if $\mu$ SA[5] is set to 1 and frame number is correct.
15 to 8	ISO_IN_4[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF4, if $\mu$ SA[4] is set to 1 and frame number is correct.
7 to 0	ISO_IN_3[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF3, if $\mu$ SA[3] is set to 1 and frame number is correct.
<b>DW5</b>			
63 to 56	ISO_IN_2[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF2 (bits 7 to 0), if $\mu$ SA[2] is set to 1 and frame number is correct.
55 to 48	ISO_IN_1[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF1, if $\mu$ SA[1] is set to 1 and frame number is correct.
47 to 40	ISO_IN_0[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF0 if $\mu$ SA[0] is set to 1 and frame number is correct.
39 to 32	$\mu$ SCS[7:0]	<b>SW</b> — writes (0 => 1) <b>HW</b> — writes (1 => 0) After processing	All bits can be set to one for every transfer. It specifies which $\mu$ SOF the complete split needs to be sent. Valid only for IN. Start split (SS) and complete split (CS) active bits— $\mu$ SA = 0000 0001, $\mu$ S CS = 0000 0100—will cause SS to execute in $\mu$ Frame0 and CS in $\mu$ Frame2.
<b>DW4</b>			
31 to 29	Status7[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF7
28 to 26	Status6[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF6
25 to 23	Status5[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF5
22 to 20	Status4[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF4
19 to 17	Status3[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF3
16 to 14	Status2[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF2
13 to 11	Status1[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF1
10 to 8	Status0[2:0]	<b>HW</b> — writes	Isochronous IN or OUT status of $\mu$ SOF0 <b>Bit 0</b> — Transaction Error (IN and OUT) <b>Bit 1</b> — Babble (IN token only) <b>Bit 2</b> — underrun (OUT token only).
7 to 0	$\mu$ SA[7:0]	<b>SW</b> — writes (0 => 1) <b>HW</b> — writes (1 => 0) After processing	Specifies which $\mu$ SOF the start split needs to be placed. <b>For OUT token:</b> 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 $\mu$ SOF. <b>For IN token:</b> Only $\mu$ SOF0, $\mu$ SOF1, $\mu$ SOF2 or $\mu$ SOF3 can be set to 1. Nothing can be set for $\mu$ SOF4 and above.

Table 72: Start and complete split for isochronous, SiTD: bit description...continued

Bit	Symbol	Access	Description
<b>DW3</b>			
63	A	<b>SW</b> — sets <b>HW</b> — resets	<b>Active:</b> Write the same value as that in V.
62	H	<b>HW</b> — writes	<b>Halt:</b> The Halt bit is set when any microframe transfer status has a stalled or halted condition.
61	B	<b>HW</b> — writes	<b>Babble:</b> This bit corresponds to bit 1 of Status0 to Status7 for every microframe transfer status.
60	X	<b>HW</b> — writes	<b>Transaction Error:</b> This bit corresponds to bit 0 of Status0 to Status7 for every microframe transfer status.
59	SC	<b>SW</b> — writes 0 <b>HW</b> — updates	<b>Start/Complete:</b> <b>0</b> — Start split <b>1</b> — Complete split.
58	reserved	-	-
57	DT	<b>HW</b> — writes <b>SW</b> — writes	<b>Data Toggle:</b> Set the Data Toggle bit to start for the PTD.
56 to 44	reserved	-	-
43 to 32	NrBytesTransferred [11:0]	<b>HW</b> — writes	<b>Number of Bytes Transferred:</b> This field indicates the number of bytes sent or received for this transaction.
<b>DW2</b>			
31 to 24	reserved	-	-
23 to 8	DataStartAddress [15:0]	<b>SW</b> — writes	<b>Data Start Address:</b> This is the start address for the 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]	<b>SW</b> — writes	Bits 7 to 3 determine which frame to execute.
<b>DW1</b>			
63 to 57	HubAddress [6:0]	<b>SW</b> — writes	<b>Hub Address:</b> This indicates the hub address. Zero for the internal or embedded hub.
56 to 50	PortNumber [6:0]	<b>SW</b> — writes	<b>Port Number:</b> This indicates the port number of the hub or embedded TT.
49 to 47	reserved	-	-
46	S	<b>SW</b> — writes	<b>Split:</b> This bit indicates whether a split transaction has to be executed: <b>0</b> — High-speed transaction <b>1</b> — Split transaction.
45 to 44	EPTtype[1:0]	<b>SW</b> — writes	<b>Transaction type:</b> <b>01</b> — Isochronous.
43 to 42	Token[1:0]	<b>SW</b> — writes	<b>Token:</b> Token PID for this transaction: <b>00</b> — OUT <b>01</b> — IN.
41 to 35	Device Address[6:0]	<b>SW</b> — writes	<b>Device Address:</b> This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	<b>SW</b> — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
<b>DW0</b>			
31	EndPt[0]	<b>SW</b> — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.
30 to 29	reserved	-	-

Table 72: Start and complete split for isochronous, SiTD: bit description...continued

Bit	Symbol	Access	Description
28 to 18	TT_MPS_Len [10:0]	<b>SW</b> — writes	<b>Transaction Translator Maximum Packet Size Length:</b> 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 ms is greater than 188 B, this field should be set to 188 B for an OUT token and 192 B for an IN token. Otherwise, this field should be equal to the total bytes sent.
17 to 3	NrBytesTo Transfer[14:0]	<b>SW</b> — writes	<b>Number of Bytes to Transfer:</b> 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 B because in SiTD the maximum allowable payload for a full-speed device is 1023 B. This field indirectly becomes the maximum packet size of the downstream device.
2 to 1	reserved	-	-
0	V	<b>SW</b> — sets <b>HW</b> — resets	<b>0</b> — This bit is deactivated when the entire PTD is executed—across $\mu$ SOF and SOF—or when a fatal error is encountered. <b>1</b> — Software updates to one when there is payload to be sent or received even across ms boundary. The current PTD is active.

### 8.5.6 Start and complete split for interrupt

Table 73 shows the bit allocation of start and complete split for interrupt<sup>6</sup>.

**Table 73: 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	S	<sup>[1]</sup>	D	Cerr	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 [3:0]										
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	<sup>[2]</sup>	<sup>[1]</sup>	MaxPacketLength[10:0]						NrBytesToTransfer[14:0] (4 kB for full-speed and low-speed)											<sup>[1]</sup>	V											

[1] Reserved.

[2] EndPt[0].

6. Patent-pending: Start and complete split for interrupt.



**Table 74: Start and complete split for interrupt: bit description**

Bit	Symbol	Access	Description
<b>DW7</b>			
63 to 40	reserved	-	-
39 to 32	INT_IN_7[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF7, if $\mu$ SA[7] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
<b>DW6</b>			
31 to 24	INT_IN_6[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF6, if $\mu$ SA[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]	<b>HW</b> — writes	Bytes received during $\mu$ SOF5, if $\mu$ SA[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]	<b>HW</b> — writes	Bytes received during $\mu$ SOF4, if $\mu$ SA[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]	<b>HW</b> — writes	Bytes received during $\mu$ SOF3, if $\mu$ SA[3] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
<b>DW5</b>			
63 to 56	INT_IN_2[7:0]	<b>HW</b> — writes	Bytes received during $\mu$ SOF2 (bits 7 to 0), if $\mu$ SA[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]	<b>HW</b> — writes	Bytes received during $\mu$ SOF1, if $\mu$ SA[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]	<b>HW</b> — writes	Bytes received during $\mu$ SOF0 if $\mu$ SA[0] is set to 1 and frame number is correct. The new value continuously overwrites the old value.
39 to 32	$\mu$ SCS[7:0]	<b>SW</b> — writes (0 => 1) <b>HW</b> — writes (1 => 0) After processing	All bits can be set to one for every transfer. It specifies which $\mu$ SOF the complete split needs to be sent. Valid only for IN. Start split (SS) and complete split (CS) active bits— $\mu$ SA = 0000 0001, $\mu$ S CS = 0000 0100—will cause SS to execute in $\mu$ Frame0 and CS in $\mu$ Frame2.
<b>DW4</b>			
31 to 29	Status7[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF7
28 to 26	Status6[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF6
25 to 23	Status5[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF5
22 to 20	Status4[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF4
19 to 17	Status3[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF3
16 to 14	Status2[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF2
13 to 11	Status1[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF1
10 to 8	Status0[2:0]	<b>HW</b> — writes	Interrupt IN or OUT status of $\mu$ SOF0 <b>Bit 0</b> — Transaction Error (IN and OUT) <b>Bit 1</b> — Babble (IN token only) <b>Bit 2</b> — underrun (OUT token only).
7 to 0	$\mu$ SA[7:0]	<b>SW</b> — writes (0 => 1) <b>HW</b> — writes (1 => 0) After processing	Specifies which $\mu$ SOF the start split needs to be placed. <b>For OUT token:</b> 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 $\mu$ SOF. <b>For IN token:</b> Only $\mu$ SOF0, $\mu$ SOF1, $\mu$ SOF2 or $\mu$ SOF3 can be set to 1. Nothing can be set for $\mu$ SOF4 and above.

Table 74: Start and complete split for interrupt: bit description...continued

Bit	Symbol	Access	Description																		
<b>DW3</b>																					
63	A	<b>SW</b> — sets <b>HW</b> — resets	<b>Active:</b> Write the same value as that in V.																		
62	H	<b>HW</b> — writes	<b>Halt:</b> The Halt bit is set when any microframe transfer status has a stalled or halted condition.																		
61	B	<b>HW</b> — writes	<b>Babble:</b> This bit corresponds to bit 1 of Status0 to Status7 for every microframe transfer status.																		
60	X	<b>HW</b> — writes	<b>Transaction Error:</b> This bit corresponds to bit 0 of Status0 to Status7 for every microframe transfer status.																		
59	SC	<b>SW</b> — writes 0 <b>HW</b> — updates	<b>Start/Complete:</b> 0 — Start split 1 — Complete split.																		
58	reserved	-	-																		
57	DT	<b>HW</b> — writes <b>SW</b> — writes	<b>Data Toggle:</b> For an interrupt transfer, set correct bit to start the PTD.																		
56 to 55	Cerr[1:0]	<b>HW</b> — writes <b>SW</b> — writes	<b>Error Counter:</b> This field corresponds to the Cerr[1:0] field in QH. 00 — The transaction will not retry. 11 — The transaction will retry three times. Hardware will decrement these values. When the transaction has tried three times, X error will be updated.																		
54 to 44	reserved	-	-																		
43 to 32	NrBytes Transferred [11:0]	<b>HW</b> — writes	<b>Number of Bytes Transferred:</b> This field indicates the number of bytes sent or received for this transaction.																		
<b>DW2</b>																					
31 to 24	reserved	-	-																		
23 to 8	DataStart Address[15:0]	<b>SW</b> — writes	<b>Data Start Address:</b> This is the start address for the 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	$\mu$ Frame[7:0]	<b>SW</b> — writes	Bits 7 to 3 is the ms polling rate. Polling rate is defined as $2^{(b-1)} \mu$ SOF; where b = 4 to 16. When b is 4, every ms is executed.																		
			<table border="1"> <thead> <tr> <th>b</th> <th>Rate</th> <th><math>\mu</math>Frame[7:3]</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>2 ms</td> <td>0 0001</td> </tr> <tr> <td>6</td> <td>4 ms</td> <td>0 0010 or 0 0011</td> </tr> <tr> <td>7</td> <td>8 ms</td> <td>0 0100 or 0 0111</td> </tr> <tr> <td>8</td> <td>16 ms</td> <td>0 1000 or 0 1111</td> </tr> <tr> <td>9</td> <td>32 ms</td> <td>1 0000 or 1 1111</td> </tr> </tbody> </table>	b	Rate	$\mu$ Frame[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
b	Rate	$\mu$ Frame[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																			
<b>DW1</b>																					
63 to 57	HubAddress [6:0]	<b>SW</b> — writes	<b>Hub Address:</b> This indicates the hub address. Zero for the internal or embedded hub.																		
56 to 50	PortNumber [6:0]	<b>SW</b> — writes	<b>Port Number:</b> This indicates the port number of the hub or embedded TT.																		

Table 74: Start and complete split for interrupt: bit description...continued

Bit	Symbol	Access	Description												
49 to 48	SE[1:0]	SW — writes	This depends on the endpoint type and direction. It is valid only for split transactions. The following applies to start split and complete split only. <table border="1"> <thead> <tr> <th>Interrupt</th> <th>S</th> <th>E</th> <th>Remarks</th> </tr> </thead> <tbody> <tr> <td>I/O</td> <td>1</td> <td>0</td> <td>low-speed</td> </tr> <tr> <td>I/O</td> <td>0</td> <td>0</td> <td>full-speed</td> </tr> </tbody> </table>	Interrupt	S	E	Remarks	I/O	1	0	low-speed	I/O	0	0	full-speed
Interrupt	S	E	Remarks												
I/O	1	0	low-speed												
I/O	0	0	full-speed												
47	reserved	-	-												
46	S	SW — writes	This bit indicates whether a split transaction has to be executed: <b>0</b> — High-speed transaction <b>1</b> — Split transaction.												
45 to 44	EPTYPE[1:0]	SW — writes	Transaction type: <b>11</b> — Interrupt.												
43 to 42	Token[1:0]	SW — writes	Token PID for this transaction: <b>00</b> — OUT <b>01</b> — IN.												
41 to 35	DeviceAddress [6:0]	SW — writes	<b>Device Address:</b> 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	<b>Endpoint:</b> This is the USB address of the endpoint within the function.												
<b>DW0</b>															
31	EndPt[0]	SW — writes	<b>Endpoint:</b> This is the USB address of the endpoint within the function.												
30 to 29	reserved	-	-												
28 to 18	MaxPacket Length[10:0]	SW — writes	<b>Maximum Packet Length:</b> 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 B as defined in the <i>Universal Serial Bus Specification Rev. 2.0</i> .												
17 to 3	NrBytesTo Transfer[14:0]	SW — writes	<b>Number of Bytes to Transfer:</b> 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	<b>0</b> — This bit is deactivated when the entire PTD is executed—across $\mu$ SOF and SOF—or when a fatal error is encountered. <b>1</b> — Software updates to one when there is payload to be sent or received even across ms boundary. The current PTD is active.												

## 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 the 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 mini-AB receptacle. If ID = LOW (mini-A plug connected), it becomes an A-device, which is a host by default. If ID = HIGH (mini-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 deasserting 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.

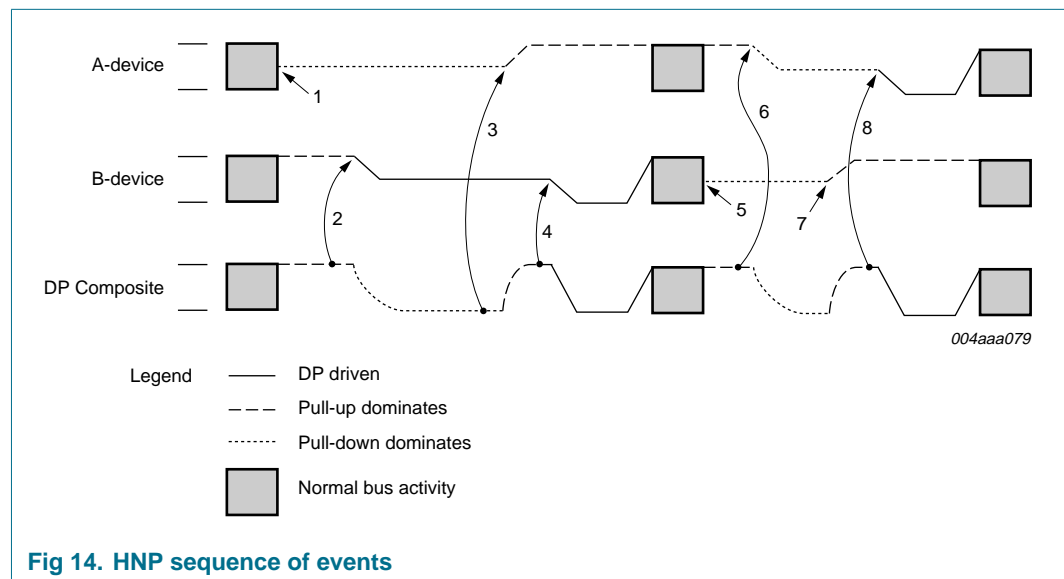
To enable the SRP detection through the DP pulsing, set DP\_SRP (bit 2) in the OTG Interrupt Enable Rise register.

## 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  $\mu$ s 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.
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  $\mu$ s 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 the state diagrams for the dual-role A-device and the dual-role B-device, respectively. For a detailed explanation, refer to *On-The-Go Supplement to the USB 2.0 Specification Rev. 1.0a*.

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.





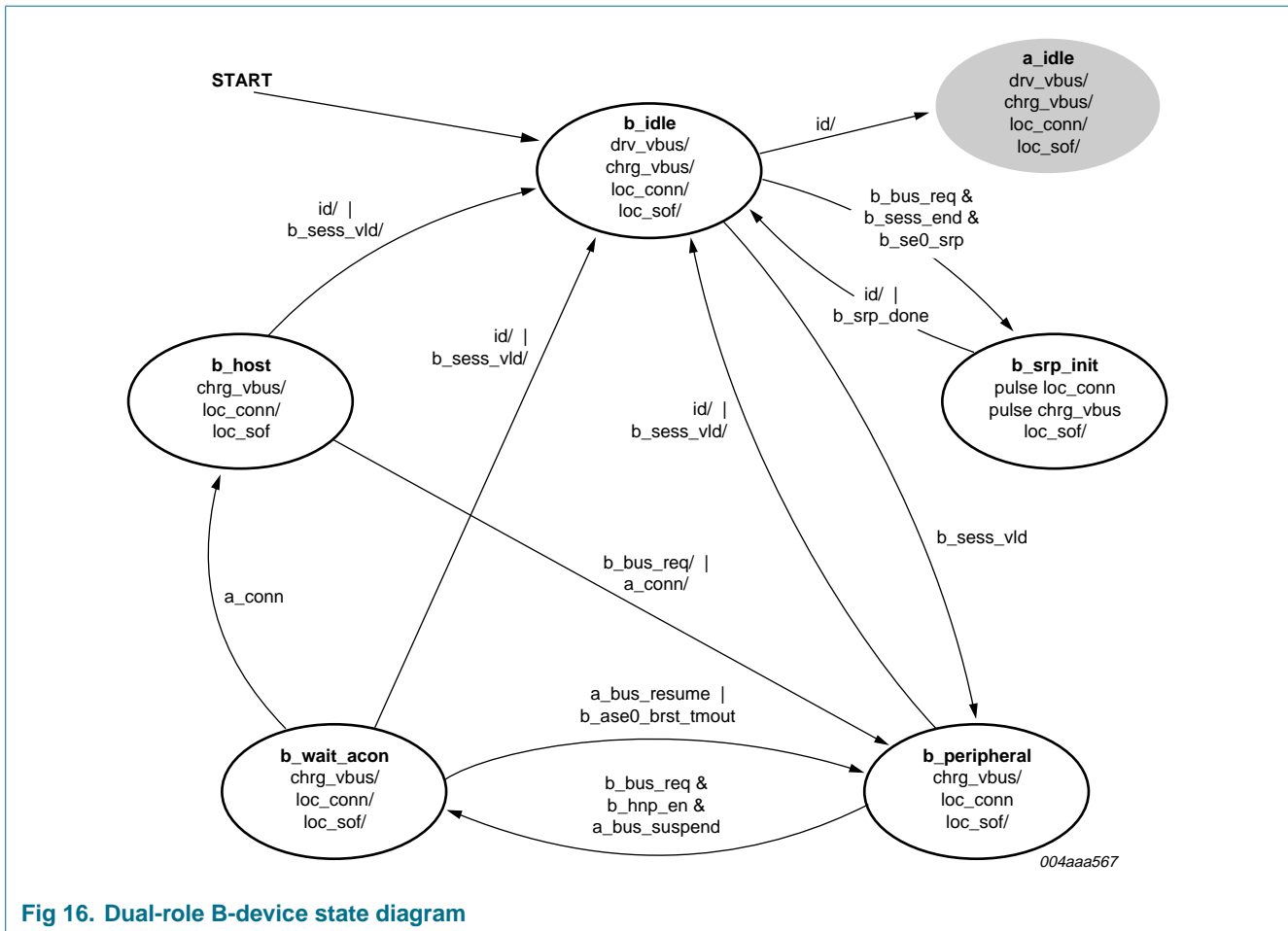


Fig 16. Dual-role B-device state diagram

### 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 the 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 setting 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 the 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 75: OTG Controller-specific register overview**

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

**Table 76: Address mapping of registers: 32-bit data bus mode**

Address	Byte 3	Byte 2	Byte 1	Byte 0
<b>Device ID registers</b>				
0370h	Product ID (read only)		Vendor ID (read only)	
<b>OTG Control register</b>				
0374h	OTG Control (clear)		OTG Control (set)	
<b>OTG Interrupt registers</b>				
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 76: Address mapping of registers: 32-bit data bus mode...continued**

Address	Byte 3	Byte 2	Byte 1	Byte 0
<b>OTG Timer register</b>				
0388h	OTG Timer (Lower word—clear)		OTG Timer (Lower word—set)	
038Ch	OTG Timer (Higher word—clear)		OTG Timer (Higher word—set)	

**Table 77: Address mapping of registers: 16-bit data bus mode**

Address	Byte 1	Byte 0	Reference
<b>Device ID registers</b>			
0370h	Vendor ID (read only)		<a href="#">Section 9.5.1.1 on page 91</a>
0372h	Product ID (read only)		<a href="#">Section 9.5.1.2 on page 91</a>
<b>OTG Control register</b>			
0374h	OTG Control (set)		<a href="#">Section 9.5.2.1 on page 92</a>
0376h	OTG Control (clear)		
<b>OTG Interrupt registers</b>			
0378h	OTG Status (read only)		<a href="#">Section 9.5.3.1 on page 93</a>
037Ah	reserved		-
037Ch	OTG Interrupt Latch (set)		<a href="#">Section 9.5.3.2 on page 94</a>
037Eh	OTG Interrupt Latch (clear)		
0380h	OTG Interrupt Enable Fall (set)		<a href="#">Section 9.5.3.3 on page 95</a>
0382h	OTG Interrupt Enable Fall (clear)		
0384h	OTG Interrupt Enable Rise (set)		<a href="#">Section 9.5.3.4 on page 95</a>
0386h	OTG Interrupt Enable Rise (clear)		
<b>OTG Timer register</b>			
0388h	OTG Timer (Lower word—set)		<a href="#">Section 9.5.4.1 on page 96</a>
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 (R: 0370h)

[Table 78](#) shows the bit description of the register.

**Table 78: Vendor ID register: bit description**

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

### 9.5.1.2 Product ID register (R: 0372h)

The bit description of the register is given in [Table 79](#).

**Table 79: Product ID register: 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 (S/C: 0374h/0376h)

Table 80 shows the bit allocation of the register.

Table 80: OTG Control register: 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_PULL_DOWN	DP_PULL_DOWN	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 81: OTG Control register: 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. <b>Remark:</b> 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 the 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 <sub>BUS</sub> to V <sub>CC(I/O)</sub> through a resistor
5	VBUS_DISCHRG	Discharge V <sub>BUS</sub> to ground through a resistor
4	VBUS_DRV	Drive V <sub>BUS</sub> to 5 V using the charge pump
3	SEL_CP_EXT	0 — internal charge pump selected 1 — external charge pump selected.

Table 81: OTG Control register: bit description...continued

Bit [1]	Symbol	Description
2	DM_PULLDOWN	<b>DM pull down:</b> 0 — Disable 1 — Enable.
1	DP_PULLDOWN	<b>DP pull down:</b> 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. <b>Remark:</b> When port 1 is in the 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 <sub>BUS</sub> sensing signal to connect the DP line to HIGH through a pull-up resistor. (V <sub>BUS</sub> is an internal signal. When 5 V is present on the V <sub>BUS</sub> pin, V <sub>BUS</sub> = 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 (R: 0378h)

This register indicates the current state of the signals that can generate an interrupt. The bit allocation of the register is given in Table 82.

Table 82: OTG Status register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved					OTG_SUSPEND	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 83.

Table 83: OTG Status register: bit description

Bit	Symbol	Description
15 to 11	-	reserved for future use
10	OTG_SUSPEND	Indicates that the bus is idle for > 3 ms
9	-	reserved
8	B_SE0_SRP	2 ms of SE0 detected in the B-idle state

**Table 83: OTG Status register: bit description...continued**

Bit	Symbol	Description
7	B_SESS_END	$V_{BUS} < 0.8\text{ V}$
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 $V_{BUS} > 4.4\text{ V}$

**9.5.3.2 OTG Interrupt Latch register (S/C: 037Ch/037Eh)**

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 84](#).

**Table 84: OTG Interrupt Latch register: bit allocation**

Bit	15	14	13	12	11	10	9	8
Symbol	reserved [1]					OTG_SUSPEND	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 85: OTG Interrupt Latch register: bit description**

Bit	Symbol	Description
15 to 11	-	reserved for future use
10	OTG_SUSPEND	Indicates that the bus is idle for > 3 ms
9	OTG_TMR_TIMEOUT	OTG timer timeout
8	B_SE0_SRP	2 ms of SE0 detected in the B-idle state
7	B_SESS_END	$V_{BUS} < 0.8\text{ V}$
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 (S/C: 0380h/0382h)

Table 86 shows the bit allocation of this register that enables interrupts on transition from HIGH-to-LOW.

Table 86: OTG Interrupt Enable Fall register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved [1]					OTG_SUSPEND	reserved	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	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 87: OTG Interrupt Enable Fall register: bit description

Bit	Symbol	Description
15 to 11	-	reserved for future use
10	OTG_SUSPEND	IRQ asserted when the bus exits from the idle state
9	-	reserved
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 (S/C: 0384h/0386h)

This register (see Table 88 for bit allocation) enables interrupts on transition from LOW-to-HIGH.

**Table 88: OTG Interrupt Enable Rise register: bit allocation**

Bit	15	14	13	12	11	10	9	8
Symbol	reserved [1]					OTG_SUSPEND	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 Enable Rise register: bit description**

Bit	Symbol	Description
15 to 11	-	reserved
10	OTG_SUSPEND	IRQ asserted when the bus is idle for more than 3 ms
9	OTG_TMR_TIMEOUT	IRQ asserted on OTG timer timeout
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 (Low word S/C: 0388h/038Ah; high word S/C: 038Ch/038Eh)

This is a 32-bit register organized as two 16-bit fields. These two fields have separate set and clear addresses. [Table 90](#) shows the bit allocation of the register.



Table 90: OTG Timer register: bit allocation

<b>Bit</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Symbol</b>	START_TMR	reserved [1]						
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
<b>Bit</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Symbol</b>	TIMER_INIT_VALUE[23:16]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	TIMER_INIT_VALUE[15:8]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C	R/S/C
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	TIMER_INIT_VALUE[7:0]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 Timer register: 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. <b>0</b> — stop the timer <b>1</b> — 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 Philips *ISP1582 Hi-Speed Universal Serial Bus peripheral controller* IC. The functionality of the Peripheral Controller in the ISP1761 is similar to the ISP1582 in the 16-bit bus mode. In addition, the register sets are also similar, with only a few variations.

The USB Chapter 9 protocol handling 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 the PIO mode or the 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 *ISP1581 Programming Guide (AN10004)* and *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 deasserted. As a result, the DMA controller deasserts 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 deasserted 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 deasserted. As a result, the DMA controller deasserts 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 deasserted 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 the 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 be accordingly set.

The ISP1761 supports only the counter mode DMA transfer. To enable the counter mode, ensure that DIS\_XFER\_CNT in the DcDMAConfiguration register (address: 0238h) is set to zero. Set bit EOT\_POL in the DMA Hardware register (address: 023Ch) to logic 1, to make the EOT function invalid because the ISP1761 does not support the external EOT mode.

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 an 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  $\mu$ s. 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 also must 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 B. 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 B.

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

#### 10.1.1.5 DMA stop and interrupt handling

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

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 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 92](#). 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 B long. Any of the 7 IN and 7 OUT endpoints can be separately enabled or disabled. The endpoint type (interrupt, isochronous or bulk) and packet size of these endpoints can be individually configured, depending on the requirements of the application. Optional double buffering increases the data throughput of these data endpoints.

**Table 92: Endpoint access and programmability**

Endpoint identifier	Maximum packet size	Double buffering	Endpoint type	Direction
EP0RX	64 B (fixed)	No	Control IN	IN
EP0TX	64 B (fixed)	No	Control OUT	OUT
EP1RX	Programmable	Yes	Programmable	IN
EP1TX	Programmable	Yes	Programmable	OUT
EP2RX	Programmable	Yes	Programmable	IN
EP2TX	Programmable	Yes	Programmable	OUT
EP3RX	Programmable	Yes	Programmable	IN
EP3TX	Programmable	Yes	Programmable	OUT
EP4RX	Programmable	Yes	Programmable	IN
EP4TX	Programmable	Yes	Programmable	OUT
EP5RX	Programmable	Yes	Programmable	IN
EP5TX	Programmable	Yes	Programmable	OUT
EP6RX	Programmable	Yes	Programmable	IN

Table 92: Endpoint access and programmability...continued

Endpoint identifier	Maximum packet size	Double buffering	Endpoint type	Direction
EP6TX	Programmable	Yes	Programmable	OUT
EP7RX	Programmable	Yes	Programmable	IN
EP7TX	Programmable	Yes	Programmable	OUT

### 10.3 Differences between the ISP1761 and ISP1582 Peripheral Controller

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

#### 10.3.1 ISP1761 initialization registers

- The ISP1582 supports the 16-bit bus access. The register addresses are 2 B aligned. The ISP1761 supports the 16-bit and 32-bits 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 B 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.

#### 10.3.2 ISP1761 DMA

- The DACK-only mode has been removed. It only supports the counter mode.
- The external-EOT mode has been removed. There is no EOT pin on the chip.
- 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 *ISP1761 Peripheral DMA Initialization (AN10040)*.

#### 10.3.3 ISP1761 peripheral suspend indication

- A HIGH level indicates that the peripheral has entered suspend mode. The pulse indication mode has been removed.

#### 10.3.4 ISP1761 interrupt and DMA common mode

In the 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](#).

## 10.4 Peripheral Controller-specific registers

**Table 93: Peripheral Controller-specific register overview**

Address	Register	Reset value	References
<b>Initialization registers</b>			
0200h	Address	00h	<a href="#">Section 10.4.1 on page 102</a>
020Ch	Mode	0000h	<a href="#">Section 10.4.2 on page 103</a>
0210h	Interrupt Configuration	FCh	<a href="#">Section 10.4.3 on page 104</a>
0212h	Debug	0000h	<a href="#">Section 10.4.4 on page 105</a>
0214h	DcInterruptEnable	0000 0000h	<a href="#">Section 10.4.5 on page 106</a>
0300h	HW Mode Control	0000 0000h	<a href="#">Section 8.3.1 on page 42</a>
0374h	OTG Control	0000 0086h	<a href="#">Section 9.5.2.1 on page 92</a>
<b>Data flow registers</b>			
022Ch	Endpoint Index	00h	<a href="#">Section 10.5.1 on page 107</a>
0228h	Control Function	00h	<a href="#">Section 10.5.2 on page 109</a>
0220h	Data Port	0000h	<a href="#">Section 10.5.3 on page 109</a>
021Ch	Buffer Length	0000h	<a href="#">Section 10.5.4 on page 110</a>
021Eh	DcBufferStatus	00h	<a href="#">Section 10.5.5 on page 111</a>
0204h	Endpoint MaxPacketSize	0000h	<a href="#">Section 10.5.6 on page 111</a>
0208h	Endpoint Type	0000h	<a href="#">Section 10.5.7 on page 113</a>
<b>DMA registers</b>			
0230h	DMA Command	FFh	<a href="#">Section 10.6.1 on page 114</a>
0234h	DMA Transfer Counter	0000 0000h	<a href="#">Section 10.6.2 on page 115</a>
0238h	DcDMAConfiguration	0001h	<a href="#">Section 10.6.3 on page 116</a>
023Ch	DMA Hardware	04h	<a href="#">Section 10.6.4 on page 117</a>
0250h	DMA Interrupt Reason	0000h	<a href="#">Section 10.6.5 on page 118</a>
0254h	DMA Interrupt Enable	0000h	<a href="#">Section 10.6.6 on page 119</a>
0258h	DMA Endpoint	00h	<a href="#">Section 10.6.7 on page 120</a>
0264h	DMA Burst Counter	0002h	<a href="#">Section 10.6.8 on page 120</a>
<b>General registers</b>			
0218h	DcInterrupt	0000 0000h	<a href="#">Section 10.7.1 on page 121</a>
0270h	DcChipID	0015 8210h	<a href="#">Section 10.7.2 on page 123</a>
0274h	Frame Number	0000h	<a href="#">Section 10.7.3 on page 123</a>
0278h	DcScratch	0000h	<a href="#">Section 10.7.4 on page 124</a>
027Ch	Unlock Device	0000h	<a href="#">Section 10.7.5 on page 124</a>
0280h	Interrupt Pulse Width	001Eh	<a href="#">Section 10.7.6 on page 125</a>
0284h	Test Mode	00h	<a href="#">Section 10.7.7 on page 125</a>

### 10.4.1 Address register (R/W: 0200h)

This register sets the USB assigned address and enables the USB peripheral. [Table 94](#) shows the bit allocation of the register.

The DEVADDR 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 be set after a bus reset.

In response to the standard USB request SET\_ADDRESS, the 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.

**Table 94: Address register: 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	1	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 95: Address register: bit description**

Bit	Symbol	Description
7	DEVEN	<b>Device Enable:</b> Logic 1 enables the peripheral.
6 to 0	DEVADDR[6:0]	<b>Device Address:</b> This field specifies the USB device peripheral.

### 10.4.2 Mode register (R/W: 020Ch)

This register consists of 2 B (bit allocation: see [Table 96](#)).

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

**Table 96: Mode register: bit allocation**

Bit	15	14	13	12	11	10	9	8
Symbol	reserved <sup>[1]</sup>						DMACLK ON	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 <sup>[1]</sup>	
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 97: Mode register: bit description**

Bit	Symbol	Description
15 to 10	-	reserved
9	DMACLKON	<b>DMA Clock On:</b> 1 — Supply clock to the DMA circuit 0 — Power saving mode. The DMA circuit will stop completely to save power.
8	VBUSSTAT	<b>V<sub>BUS</sub> Status:</b> This bit reflects the V <sub>BUS</sub> pin status.



Table 97: Mode register: bit description...continued

Bit	Symbol	Description
7	CLKAON	<p><b>Clock Always On:</b></p> <p><b>1</b> — Enable the Clock-Always-On feature</p> <p><b>0</b> — Disable the Clock-Always-On feature.</p> <p>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.</p> <p>If the Clock-Always-On feature is enabled, the clocks are always running and the GOSUSP event is unable to stop the clock while the Peripheral Controller enters the suspend state.</p>
6	SNDRSU	<p><b>Send Resume:</b> Writing logic 1, followed by logic 0 will generate an upstream resume signal of 10 ms duration, after a 5 ms delay.</p>
5	GOSUSP	<p><b>Go Suspend:</b> Writing logic 1, followed by logic 0 will activate suspend mode.</p>
4	SFRESET	<p><b>Soft Reset:</b> 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).</p>
3	GLINTENA	<p><b>Global Interrupt Enable:</b> Logic 1 enables all interrupts. Individual interrupts can be masked by clearing the corresponding bits in the DcInterruptEnable register.</p> <p>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 be immediately generated on the interrupt pin. (If the interrupt is set to the pulse mode, the interrupt events that were generated before the global interrupt is enabled may be dropped.)</p>
2	WKUPCS	<p><b>Wake up on Chip Select:</b> 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.)</p>
1 to 0	-	reserved

### 10.4.3 Interrupt Configuration register (R/W: 0210h)

This 1 B register determines the behavior and polarity of the INT output. The bit allocation is shown in [Table 98](#). When the USB SIE receives or generates an ACK, NAK or STALL, 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 100](#) 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 98: Interrupt Configuration register: 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 99: Interrupt Configuration register: bit description**

Bit	Symbol	Description
7 to 6	CDBGMOD[1:0]	<b>Control 0 Debug Mode:</b> For values, see <a href="#">Table 100</a>
5 to 4	DDBGMODIN[1:0]	<b>Data Debug Mode IN:</b> For values, see <a href="#">Table 100</a>
3 to 2	DDBGMODOUT[1:0]	<b>Data Debug Mode OUT:</b> For values, see <a href="#">Table 100</a>
1	INTLVL	<b>Interrupt Level:</b> Selects the signaling mode on output INT (0 = level; 1 = pulsed). In the pulsed mode, an interrupt produces a 60 ns pulse. Bus reset value: unchanged.
0	INTPOL	<b>Interrupt Polarity:</b> Selects signal polarity on output INT (0 = active LOW; 1 = active HIGH). Bus reset value: unchanged.

**Table 100: 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 <sup>[1]</sup>	interrupt on all ACK and first NAK <sup>[1]</sup>	interrupt on all ACK, NYET and first NAK <sup>[1]</sup>

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

#### 10.4.4 Debug register (R/W: 0212h)

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 101](#).

**Table 101: Debug register: bit allocation**

Bit	15	14	13	12	11	10	9	8
Symbol	reserved <sup>[1]</sup>							
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 <sup>[1]</sup>							DEBUG
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 102: Debug register: bit allocation**

Bit	Symbol	Description
15 to 1	-	reserved
0	DEBUG	Always set this bit to logic 0 when the ISP1761 is in 16-bit bus access mode, or set bit 16 of the Interrupt Configuration register to logic 0 when the ISP1761 is in 32-bit bus access mode.

**10.4.5 DcInterruptEnable register (R/W: 0214h)**

This register enables or disables individual interrupt sources. The interrupt for each endpoint can be individually controlled through the associated IEPnRX or IEPnTX bits, here n represents the endpoint number. All interrupts can be globally disabled through bit GLINTENA in the Mode register (see [Table 96](#)).

An interrupt is generated when the USB SIE receives or generates an ACK or NAK on the USB bus. The interrupt generation depends on the 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 the DDBGMODIN bits. All data OUT transactions go through the Receive buffers (RX) that are handled by the DDBGMODOUT bits. Transactions on control endpoint 0—IN, OUT and SETUP—are handled by the CDBGMOD bits.

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

The DcInterruptEnable register consists of 4 B. The bit allocation is given in [Table 103](#).

**Table 103: DcInterruptEnable register: 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 104: DcInterruptEnable register: 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 setup data received on endpoint 0.
7	IEVBUS	Logic 1 enables interrupt for V <sub>BUS</sub> sensing.
6	IEDMA	Logic 1 enables interrupt on detection of a DMA status change.
5	IEHS_STA	Logic 1 enables interrupt on detection of a high-speed status change.
4	IERESM	Logic 1 enables interrupt on detection of a resume state.
3	IESUSP	Logic 1 enables interrupt on detection of a suspend state.
2	IEPSOF	Logic 1 enables interrupt on detection of a Pseudo SOF.
1	IESOF	Logic 1 enables interrupt on detection of an SOF.
0	IEBRST	Logic 1 enables interrupt on detection of a bus reset.

## 10.5 Data flow registers

### 10.5.1 Endpoint Index register (R/W: 022Ch)

The Endpoint Index register selects a target endpoint for register access by the microcontroller. The register consists of 1 B, and the bit allocation is shown in [Table 105](#).

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 Index register must not point to the same endpoint.

**Table 105: Endpoint Index register: bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved [1]		EPOSETUP	ENDPIDX[3:0]				DIR
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 106: Endpoint Index register: bit description**

Bit	Symbol	Description
7 to 6	-	reserved
5	EPOSETUP	<b>Endpoint 0 Setup:</b> Selects the SETUP buffer for endpoint 0. <b>0</b> — EP0 data buffer <b>1</b> — SETUP buffer. Must be logic 0 for access to endpoints other than endpoint 0.
4 to 1	ENDPIDX[3:0]	<b>Endpoint Index:</b> Selects the target endpoint for register access of Buffer Length, Control Function, Data Port, Endpoint Type and MaxPacketSize.
0	DIR	<b>Direction bit:</b> Sets the target endpoint as IN or OUT. <b>0</b> — Target endpoint refers to OUT (RX) FIFO <b>1</b> — Target endpoint refers to IN (TX) FIFO.

**Table 107: Addressing of endpoint 0 buffers**

Buffer name	EPOSETUP	ENDPIDX	DIR
SETUP	1	00h	0
Data OUT	0	00h	0
Data IN	0	00h	1

### 10.5.2 Control Function register (R/W: 0228h)

The Control Function register performs the buffer management on endpoints. It consists of 1 B, and the bit configuration is given in [Table 108](#). The register bits can stall, clear or validate any enabled data endpoint. Before accessing this register, the Endpoint Index register must be written first to specify the target endpoint.

**Table 108: Control Function register: bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved <sup>[1]</sup>			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	R/W	R/W	R/W

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

**Table 109: Control Function register: bit description**

Bit	Symbol	Description
7 to 5	-	reserved
4	CLBUF	<b>Clear Buffer:</b> 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.
3	VENDP	<b>Validate Endpoint:</b> Logic 1 validates the 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	<b>Data Stage Enable:</b> This bit controls the response of the ISP1761 to a control transfer. When this bit is set, the ISP1761 goes to the data stage; otherwise, the ISP1761 will NAK the data stage transfer until the firmware explicitly responds to the setup command.
1	STATUS	<b>Status Acknowledge:</b> 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. <b>0</b> — Sends NAK <b>1</b> — Sends an empty packet following the IN token (host-to-peripheral) or ACK following the OUT token (peripheral-to-host).
0	STALL	<b>Stall Endpoint:</b> Logic 1 stalls the indexed endpoint. This bit is not applicable for isochronous transfers. <b>Remark:</b> ‘Stall’ing a data endpoint will confuse the Data Toggle bit regarding the stalled endpoint because the internal logic starts 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 or logic 1 in the Endpoint Type register) to reset the PID.

### 10.5.3 Data Port register (R/W: 0220h)

This 2 B register provides direct access for a microcontroller to the FIFO of the indexed endpoint. The bit description is shown in [Table 110](#).

**Peripheral to host (IN endpoint):** After each write, an internal counter is automatically incremented by two 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.

**Host to peripheral (OUT endpoint):** After each read, an internal counter is automatically decremented by two 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. The buffer contents can also be cleared through the Control Function register (bit CLBUF), whenever it is necessary to forcefully clear the contents.

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

**Table 110: Data Port register: bit description**

Bit	Symbol	Access	Value	Description
15 to 0	DATAPORT[15:0]	R/W	0000h	<b>Data Port:</b> A 500 ns delay may be required for the first read from the Data Port.

#### 10.5.4 Buffer Length register (R/W: 021Ch)

This register determines the current packet size (DATACOUNT) of the indexed endpoint FIFO. The bit description is given in [Table 111](#).

The Buffer Length register is automatically loaded with the FIFO size, when the Endpoint MaxPacketSize register is written (see [Table 115](#)). 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 B and the MaxPacketSize is programmed as 64 B, the Buffer Length register need not be filled. This is because the transfer size is a multiple of MaxPacketSize, and the MaxPacketSize packets will be automatically validated because the last packet is also of MaxPacketSize.

Example 2: Consider that the transfer size is 510 B and the MaxPacketSize is programmed as 64 B, the Buffer Length register should be filled with 62 B just before the microcontroller writes the last packet of 62 B. This ensures that the last packet, which is a short packet of 62 B, 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 the 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 111: Buffer Length register: bit description**

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

### 10.5.5 DcBufferStatus register (R/W: 021Eh)

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 112](#) shows the bit allocation of the DcBufferStatus register.

**Table 112: DcBufferStatus register: bit allocation**

Bit	7	6	5	4	3	2	1	0
<b>Symbol</b>	reserved <sup>[1]</sup>						BUF1	BUF0
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 113: DcBufferStatus register: bit description**

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

### 10.5.6 Endpoint MaxPacketSize register (R/W: 0204h)

This register determines the maximum packet size for all endpoints, except control 0. The register contains 2 B, and the bit allocation is given in [Table 114](#).

Each time the register is written, the Buffer Length registers of all endpoints are reinitialized to the FFOSZ field value. The NTRANS bits control the number of transactions allowed in a single microframe (for high-speed isochronous and interrupt endpoints only).

**Table 114: Endpoint MaxPacketSize register: bit allocation**

Bit	15	14	13	12	11	10	9	8
Symbol	reserved <sup>[1]</sup>			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 115: Endpoint MaxPacketSize register: bit description**

Bit	Symbol	Description
15 to 13	-	reserved
12 to 11	NTRANS[1:0]	<b>Number of Transactions.</b> HS mode only. <b>00</b> — 1 packet per microframe <b>01</b> — 2 packets per microframe <b>10</b> — 3 packets per microframe <b>11</b> — reserved. These bits are applicable only for isochronous or interrupt transactions.
10 to 0	FFOSZ[10:0]	<b>FIFO Size:</b> Sets the FIFO size, in bytes, for the indexed endpoint. Applies to both high-speed and full-speed operations (see <a href="#">Table 116</a> ).

**Table 116: Programmable FIFO size**

NTRANS[1:0]	FFOSZ[10:0]	Non-isochronous	Isochronous
0h	08h	8 B	-
0h	10h	16 B	-
0h	20h	32 B	-
0h	40h	64 B	-
0h	80h	128 B	-
0h	100h	256 B	-
0h	200h	512 B	-
2h	400h	-	3072 B

Each programmable FIFO can be independently configured through its Endpoint MaxPacketSize register (R/W: 04h), but the total physical size of all enabled endpoints (IN plus OUT) must not exceed 8192 B.



### 10.5.7 Endpoint Type register (R/W: 0208h)

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 B, and the bit allocation is shown in [Table 117](#).

Table 117: Endpoint Type register: bit allocation

Bit	15	14	13	12	11	10	9	8
Symbol	reserved <sup>[1]</sup>							
Reset	0	0	0	0	0	0	0	0
Bus reset	0	0	0	0	0	0	0	0
Access								
Bit	7	6	5	4	3	2	1	0
Symbol	reserved <sup>[1]</sup>			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 118: Endpoint Type register: bit description

Bit	Symbol	Description
15 to 5	-	reserved
4	NOEMPKT	<b>No Empty Packet:</b> Logic 0 causes an empty packet to be appended to the next IN token of the USB data, if the Buffer Length register or the Endpoint MaxPacketSize register is zero. Logic 1 disables this function. This bit is applicable only in the DMA mode.
3	ENABLE	<b>Endpoint Enable:</b> 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. <b>Remark:</b> ‘Stall’ing a data endpoint will confuse the Data Toggle bit on the stalled endpoint because the internal logic starts 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 or logic 1 in the Endpoint Type register) to reset the PID.
2	DBLBUF	<b>Double Buffering:</b> Logic 1 enables double buffering for the indexed endpoint. Logic 0 disables double buffering.
1 to 0	ENDPTYP[1:0]	<b>Endpoint Type:</b> These bits select the endpoint type as follows. <b>00</b> — not used <b>01</b> — Isochronous <b>10</b> — Bulk <b>11</b> — Interrupt.

### 10.6 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 119](#).

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

The GDMA (slave) can operate in the 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 the 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 131](#).

**Table 119: Control bits for GDMA read or write (opcode = 00h/01h)**

Control bits	Description	Reference
<b>Mode register</b>		
DMACKON	Set DMACKON to logic 1	<a href="#">Table 97</a>
<b>DcDMAConfiguration register</b>		
MODE[1:0]	Determines the active read or write data strobe signals	<a href="#">Table 126</a>
WIDTH	Selects the DMA bus width: 16 or 32 bits	
DIS_XFER_CNT	Disables the use of the DMA Transfer Counter	
<b>DMA Hardware register</b>		
ENDIAN[1:0]	Determines whether the data is to be byte swapped or normal	<a href="#">Table 128</a>
DACK_POL, DREQ_POL	Select the polarity of the DMA handshake signals	

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

### 10.6.1 DMA Command register (W: 0230h)

The DMA Command register is a 1 B register (for bit allocation, see [Table 120](#)) 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 three-state until a DMA command is executed.

Table 120: DMA Command register: 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 121: DMA Command register: bit description

Bit	Symbol	Description
7 to 0	DMA_CMD[7:0]	DMA command code; see <a href="#">Table 122</a> .

Table 122: DMA commands

Code	Name	Description
00h	GDMA Read	<b>Generic DMA IN token transfer (slave mode only):</b> Data is transferred from the external DMA bus to the internal buffer.
01h	GDMA Write	<b>Generic DMA OUT token transfer (slave mode only):</b> Data is transferred from the internal buffer to the external DMA bus.
02h to 0Dh	-	reserved
0Eh	Validate Buffer	<b>Validate Buffer (for debugging only):</b> Request from the microcontroller to validate the endpoint buffer, following a DMA to USB data transfer.
0Fh	Clear Buffer	<b>Clear Buffer:</b> Request from the microcontroller to clear the endpoint buffer after a USB to DMA data transfer.
10h	-	reserved
11h	Reset DMA	<b>Reset DMA:</b> Initializes the DMA core to its power-on reset state. <b>Remark:</b> When the DMA core is reset during the Reset DMA command, the DREQ, DACK, RD_N and WR_N handshake pins will be temporarily asserted. This can confuse the external DMA controller. To prevent this, start the external DMA controller <b>only after</b> the DMA reset.
12h	-	reserved
13h	GDMA Stop	<b>GDMA stop:</b> 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

### 10.6.2 DMA Transfer Counter register (R/W: 0234h)

This 4 B 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 123](#).

**For IN endpoint** — As 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 B, 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.

If the DMA counter is disabled in the DMA transfer, it will still decrement and rollover when it reaches zero.

**Table 123: DMA Transfer Counter register: 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 124: DMA Transfer Counter register: bit description**

Bit	Symbol	Description
31 to 24	DMACR4, DMACR[31:24]	<b>DMA Counter 4:</b> DMA transfer counter byte 4 (MSB)
23 to 16	DMACR3, DMACR[23:16]	<b>DMA Counter 3:</b> DMA transfer counter byte 3
15 to 8	DMACR2, DMACR[15:8]	<b>DMA Counter 2:</b> DMA transfer counter byte 2
7 to 0	DMACR1, DMACR[7:0]	<b>DMA Counter 1:</b> DMA transfer counter byte 1 (LSB)

### 10.6.3 DcDMAConfiguration register (R/W: 0238h)

This register defines the DMA configuration for the GDMA mode. The DcDMAConfiguration register consists of 2 B. The bit allocation is given in [Table 125](#).

Table 125: DcDMAConfiguration register: 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 126: DcDMAConfiguration register: bit description

Bit	Symbol	Description
15 to 8	-	reserved
7	DIS_XFER_CNT	<b>Disable Transfer Counter:</b> Logic 1 disables the DMA Transfer Counter (see <a href="#">Table 123</a> ).
6 to 4	-	reserved
3 to 2	MODE[1:0]	<b>Mode:</b> These bits only affect the GDMA slave handshake signals. <b>00</b> — WR_N slave strobes data from the DMA bus into the ISP1761; RD_N slave puts data from the ISP1761 on the DMA bus <b>01, 10, 11</b> — reserved
1	-	reserved
0	WIDTH	<b>Width:</b> This bit selects the DMA bus width for the GDMA slave. <b>0</b> — 32-bit data bus <b>1</b> — 16-bit data bus.

### 10.6.4 DMA Hardware register (R/W: 023Ch)

The DMA Hardware register consists of 1 B. The bit allocation is shown in [Table 127](#).

This register determines the polarity of the bus control signals (DACK and DREQ). It also controls whether the upper and lower parts of the data bus are swapped (bits ENDIAN[1:0]) for the GDMA (slave) mode.

Table 127: DMA Hardware register: bit allocation

Bit	7	6	5	4	3	2	1	0
Symbol	ENDIAN[1:0]		EOT_POL	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 128: DMA Hardware register: bit description

Bit	Symbol	Description
7 to 6	ENDIAN[1:0]	<b>Endian:</b> These bits determine whether the data bus is swapped between the internal RAM and the DMA bus. <b>00</b> — Normal data representation 16-bit bus: MSB on DATA[15:8], LSB on DATA[7:0] <b>01</b> — Swapped data representation 16-bit bus: MSB on DATA[7:0], LSB on DATA[15:8] <b>10, 11</b> — reserved.
5	EOT_POL	Selects the polarity of the End-Of-Transfer input; used in GDMA (slave) mode only. <b>0</b> — EOT is active LOW <b>1</b> — EOT is active HIGH.
4	-	reserved
3	DACK_POL	<b>DACK Polarity:</b> Selects the DMA acknowledgment polarity. <b>0</b> — DACK is active LOW <b>1</b> — DACK is active HIGH.
2	DREQ_POL	<b>DREQ Polarity:</b> Selects the DMA request polarity. <b>0</b> — DREQ is active LOW <b>1</b> — DREQ is active HIGH.
1 to 0	-	reserved

### 10.6.5 DMA Interrupt Reason register (R/W: 0250h)

This 2 B 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. When the register is read, perform a logical AND with the corresponding bits of the DMA Interrupt Enable register.

The bit allocation is given in [Table 129](#).

Table 129: DMA Interrupt Reason register: bit allocation

Bit	15	14	13	12	11	10	9	8
<b>Symbol</b>	reserved			GDMA_STOP	reserved	INT_EOT	reserved <sup>[1]</sup>	DMA_XFER_OK
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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
<b>Symbol</b>	reserved <sup>[1]</sup>							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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: DMA Interrupt Reason register: bit description**

Bit	Symbol	Description
15 to 13	-	reserved
12	GDMA_STOP	<b>GDMA Stop:</b> When the GDMA_STOP command is issued to the DMA Command registers, it means that the DMA transfer has successfully terminated.
11	-	reserved
10	INT_EOT	<b>Internal EOT:</b> Logic 1 indicates that an internal EOT is detected; see <a href="#">Table 131</a> .
9	-	reserved
8	DMA_XFER_OK	<b>DMA Transfer OK:</b> Logic 1 indicates that the DMA transfer has been completed (DMA Transfer Counter has become zero). This bit is only used in the GDMA (slave) mode.
7 to 0	-	reserved

**Table 131: 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 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.

### 10.6.6 DMA Interrupt Enable register (R/W: 0254h)

This 2 B register controls the interrupt generation of the source bits in the DMA Interrupt Reason register. The bit allocation is given in [Table 132](#). The bit description is given in [Table 130](#).

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

**Table 132: DMA Interrupt Enable register: bit allocation**

Bit	15	14	13	12	11	10	9	8
<b>Symbol</b>	reserved <sup>[1]</sup>			IE_GDMA_STOP	reserved <sup>[1]</sup>	IE_INT_EOT	reserved <sup>[1]</sup>	IE_DMA_XFER_OK
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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
<b>Symbol</b>	reserved <sup>[1]</sup>							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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.

### 10.6.7 DMA Endpoint register (R/W: 0258h)

This 1 B register selects a USB endpoint FIFO as the source or destination for DMA transfers. The bit allocation is given in [Table 133](#).

**Table 133: DMA Endpoint register: bit allocation**

Bit	7	6	5	4	3	2	1	0
Symbol	reserved <sup>[1]</sup>				EPIDX[2:0]			DMADIR
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 Endpoint register: bit description**

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

The DMA Endpoint register must not reference the endpoint that is indexed by the Endpoint Index register (022Ch) at any time. Doing so would 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.

### 10.6.8 DMA Burst Counter register (R/W: 0264h)

The bit allocation of the register is given in [Table 135](#).

**Table 135: DMA Burst Counter register: bit allocation**

Bit	15	14	13	12	11	10	9	8
Symbol	reserved <sup>[1]</sup>				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	0	1	0
Bus reset	0	0	0	0	0	0	1	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 136: DMA Burst Counter register: bit description**

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

## 10.7 General registers

### 10.7.1 DcInterrupt register (R/W: 0218h)

The DcInterrupt register consists of 4 B. The bit allocation is given in [Table 137](#).

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 nonzero, 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 be individually 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 137: DcInterrupt register: bit allocation**

Bit	31	30	29	28	27	26	25	24
Symbol	reserved <sup>[1]</sup>						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 <sup>[1]</sup>	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 138: DcInterrupt register: 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 V <sub>BUS</sub> is turned on.
6	DMA	<b>DMA status:</b> Logic 1 indicates a change in the DMA Status register.
5	HS_STAT	<b>High Speed Status:</b> Logic 1 indicates a change from the full-speed to high-speed mode (HS connection). This bit is not set, when the system goes into the full-speed suspend.
4	RESUME	<b>Resume status:</b> Logic 1 indicates that a status change from suspend to resume (active) was detected.
3	SUSP	<b>Suspend status:</b> Logic 1 indicates that a status change from active to suspend was detected on the bus.

Table 138: DcInterrupt register: bit description...continued

Bit	Symbol	Description
2	PSOF	<b>Pseudo SOF interrupt:</b> Logic 1 indicates that a pseudo SOF or $\mu$ SOF was received. Pseudo SOF is an internally generated clock signal (full-speed: 1 ms period, high-speed: 125 $\mu$ s period) synchronized to the USB bus SOF or $\mu$ SOF.
1	SOF	<b>SOF interrupt:</b> Logic 1 indicates that a SOF or $\mu$ SOF was received.
0	BRESET	<b>Bus Reset:</b> Logic 1 indicates that a USB bus reset was detected. When the SW_SEL_HC_DC bit in the OTG Control register is set, BRESET will not be set, instead, this interrupt bit will report SE0 on DP and DM for 2 ms.

### 10.7.2 DcChipID register (R: 0270h)

This read-only register contains the chip identification and the hardware version numbers. The firmware should check this information to determine the functions and features supported. The register contains 3 B, and the bit allocation is shown in [Table 139](#).

Table 139: DcChipID register: bit description

Bit	Symbol	Access	Value	Description
31 to 0	CHIPID [31:0]	R	0015 8210h	<b>Chip ID:</b> This registers represents the hardware version number (0015h) and the chip ID (8210h) for the Peripheral Controller.

### 10.7.3 Frame Number register (R: 0274h)

This read-only register contains the frame number of the last successfully received Start-Of-Frame (SOF). The register contains 2 B, and the bit allocation is given in [Table 140](#). In the case of 8-bit access, the register content is returned lower byte first.

Table 140: Frame Number register: bit allocation

Bit	15	14	13	12	11	10	9	8
<b>Symbol</b>	reserved		MICROSOF[2:0]			SOFR[10:8]		
<b>Power Reset</b>	0	0	0	0	0	0	0	0
<b>Bus Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
<b>Symbol</b>	SOFR[7:0]							
<b>Power Reset</b>	0	0	0	0	0	0	0	0
<b>Bus Reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	R	R	R	R	R	R	R	R

Table 141: Frame Number register: 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

**10.7.4 DcScratch register (R/W: 0278h)**

This 16-bit register can be used by the firmware to save and restore information. For example, the peripheral status before it enters the suspend state. The content of this register will not be altered by a bus reset. The bit allocation is given in [Table 142](#).

**Table 142: DcScratch register: bit allocation**

<b>Bit</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Symbol</b>	SFIRH[7:0]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	unchanged							
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<b>Bit</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Symbol</b>	SFIRL[7:0]							
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	unchanged							
<b>Access</b>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**Table 143: DcScratch register: 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)

**10.7.5 Unlock Device register (W: 027Ch)**

To protect the registers from getting corrupted when the ISP1761 goes into suspend, the write operation is disabled if the PWRON bit in the Mode register is set to logic 0. In this case, when the chip resumes, the Unlock Device command must be first 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 144](#).

**Table 144: Unlock Device register: bit allocation**

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

**Table 145: Unlock Device register: bit description**

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

When the PWRON bit in the Mode register is logic 1, the chip is powered. In such a case, you do not need to issue the Unlock command because the microprocessor is powered and therefore, the RD\_N, WR\_N and CS\_N signals maintain their states.

When the PWRON bit is logic 0, the RD\_N, WR\_N and CS\_N signals are floating because the microprocessor is not powered. To protect the ISP1761 registers from being corrupted during suspend, register write is locked when the chip goes into suspend. Therefore, you need to issue the Unlock command to unlock the ISP1761 registers.

### 10.7.6 Interrupt Pulse Width register (R/W: 0280h)

[Table 146](#) shows the bit description of the register.

**Table 146: Interrupt Pulse Width register: bit description**

Bit	Symbol	Access	Value	Description
15 to 0	INTR_PULSE_WIDTH[15:0]	R/W	001Eh	<b>Interrupt Pulse Width:</b> The interrupt signal pulse width is configurable while it is in the 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.

### 10.7.7 Test Mode register (R/W: 0284h)

This 1 B register allows the firmware to set the DP and DM pins to predetermined states for testing purposes. The bit allocation is given in [Table 147](#).

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

**Table 147: Test Mode register: bit allocation**

Bit	7	6	5	4	3	2	1	0
<b>Symbol</b>	FORCEHS	reserved <a href="#">[1]</a>		FORCEFS	PRBS	KSTATE	JSTATE	SE0_NAK
<b>Reset</b>	0	0	0	0	0	0	0	0
<b>Bus reset</b>	0	0	0	0	0	0	0	0
<b>Access</b>	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 148: Test Mode register: bit description**

Bit	Symbol	Description
7	FORCEHS	<b>Force High-Speed:</b> Logic 1 <a href="#">[1]</a> forces the hardware to the high-speed mode only and disables the chirp detection logic.
6 to 5	-	reserved.
4	FORCEFS	<b>Force Full-Speed:</b> Logic 1 <a href="#">[1]</a> forces the physical layer to the full-speed mode only and disables the chirp detection logic.
3	PRBS	Logic 1 <a href="#">[2]</a> sets the DP and DM pins to toggle in a predetermined random pattern.

Table 148: Test Mode register: bit description...continued

Bit	Symbol	Description
2	KSTATE	<b>K State:</b> Writing logic 1 [2] sets the DP and DM pins to the K state.
1	JSTATE	<b>J State:</b> Writing logic 1 [2] sets the DP and DM pins to the J state.
0	SE0_NAK	<b>SE0 NAK:</b> Writing logic 1 [2] sets the DP and DM pins 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 should be set at a time.

[2] Of the four bits (PRBS, KSTATE, JSTATE and SE0\_NAK), only one bit should be set at a time.

## 11. Power consumption

Table 149: Power consumption

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

**Remark:** 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.

**Remark:** 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  $\mu\text{A}$  at room temperature. The suspend current may increase if the ambient temperature increases. For details, see [Section 7.6](#).

**Remark:** In hybrid mode, when  $V_{CC}$  is disconnected  $I_{CC(I/O)}$  will be generally below 100  $\mu\text{A}$ . The average value is 60  $\mu\text{A}$  to 70  $\mu\text{A}$ .

## 12. Limiting values

**Table 150: Absolute maximum ratings**

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

Symbol	Parameter	Conditions	Min	Max	Unit
$V_{CC(I/O)}$	supply voltage		-0.5	+4.6	V
$V_{CC(5V0)}$	supply voltage		-0.5	+5.6	V
$V_{CC(C\_IN)}$	supply voltage		-	4.5	V
$I_{lu}$	latch-up current	$V_I < 0$ or $V_I > V_{CC}$	-	100	mA
$V_{esd}$	electrostatic discharge voltage	$I_{LI} < 1 \mu A$	-4000	+4000	V
$T_{stg}$	storage temperature		-40	+125	°C

## 13. Recommended operating conditions

**Table 151: Recommended operating conditions**

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{CC(I/O)}$	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	-	4.2	V
$T_{amb}$	ambient temperature		-40	-	+85	°C
$I_{CC(susp)}$	deep sleep suspend current	$T_{amb} = 25$ °C, $V_{CC(5V0)} = 3.3$ V	-	150	-	μA

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



## 14. Static characteristics

**Table 152: Static characteristics: digital pins**

All digital pins [1], except pins ID, PSW1\_N, PSW2\_N, PSW3\_N and V<sub>BAT\_ON\_N</sub>.  
V<sub>CC(I/O)</sub> = 3.0 V to 3.6 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>IH</sub>	HIGH-level input voltage		2.0	-	-	V
V <sub>IL</sub>	LOW-level input voltage		-	-	0.8	V
V <sub>hys</sub>	hysteresis voltage		0.4	-	0.7	V
V <sub>OL</sub>	LOW-level output voltage	I <sub>OL</sub> = 3 mA	-	-	0.4	V
V <sub>OH</sub>	HIGH-level output voltage		2.4	-	-	V
I <sub>LI</sub>	input leakage current	0 < V <sub>IN</sub> < V <sub>CC(I/O)</sub>	-	-	1	μA
C <sub>IN</sub>	input pin capacitance		-	2.75	-	pF

[1] Includes OC1\_N/V<sub>BUS</sub>, OC2\_N and OC3\_N when used as digital overcurrent pins.

**Table 153: Static characteristics: digital pins**

All digital pins [1], except pins ID, PSW1\_N, PSW2\_N, PSW3\_N and V<sub>BAT\_ON\_N</sub>.  
V<sub>CC(I/O)</sub> = 1.65 V to 1.95 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>IH</sub>	HIGH-level input voltage		1.2	-	-	V
V <sub>IL</sub>	LOW-level input voltage		-	-	0.5	V
V <sub>hys</sub>	hysteresis voltage		0.4	-	0.7	V
V <sub>OL</sub>	LOW-level output voltage	I <sub>OL</sub> = 3 mA	-	-	0.22V <sub>CC(I/O)</sub>	V
V <sub>OH</sub>	HIGH-level output voltage		0.8V <sub>CC(I/O)</sub>	-	-	V
I <sub>LI</sub>	input leakage current	0 < V <sub>IN</sub> < V <sub>CC(I/O)</sub>	-	-	1	μA
C <sub>IN</sub>	input pin capacitance		-	2.75	-	pF

[1] Includes OC1\_N/V<sub>BUS</sub>, OC2\_N and OC3\_N when used as digital overcurrent pins.

**Table 154: Static characteristics: PSW1\_N, PSW2\_N, PSW3\_N**

V<sub>CC(I/O)</sub> = 1.65 V to 3.6 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>OL</sub>	LOW-level output voltage	I <sub>OL</sub> = 8 mA, pull-up to V <sub>CC(5V0)</sub>	-	-	0.4	V
V <sub>OH</sub>	HIGH-level output voltage	pull-up to V <sub>CC(I/O)</sub>	-	V <sub>CC(I/O)</sub>	-	V

**Table 155: Static characteristics: USB interface block (pins DM1 to DM3 and DP1 to DP3)**

$V_{CC(I/O)} = 1.65\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Input levels for high-speed</b>						
V <sub>HSSQ</sub>	squelch detection threshold (differential signal amplitude)	squelch detected	-	-	100	mV
		no squelch detected	150	-	-	mV
V <sub>HSDSC</sub>	disconnect detection threshold (differential signal amplitude)	disconnect detected	625	-	-	mV
		disconnect not detected	-	-	525	mV
V <sub>HSCM</sub>	data signaling common mode voltage range		-50	-	+500	mV
<b>Output levels for high-speed</b>						
V <sub>HSOI</sub>	idle state		-10	-	+10	mV
V <sub>HSOH</sub>	data signaling HIGH		360	-	440	mV
V <sub>HSOL</sub>	data signaling LOW		-10	-	+10	mV
V <sub>CHIRPJ</sub>	Chirp J level (differential voltage)		700 [1]	-	1100	mV
V <sub>CHIRPK</sub>	Chirp K level (differential voltage)		-900 [1]	-	-500	mV
<b>Input levels for full-speed and low-speed</b>						
V <sub>IH</sub>	HIGH-level input voltage (drive)		2.0	-	-	V
V <sub>IHZ</sub>	HIGH-level input voltage (floating)		2.7	-	3.6	V
V <sub>IL</sub>	LOW-level input voltage		-	-	0.8	V
V <sub>DI</sub>	differential input sensitivity	$ V_{DP} - V_{DM} $	0.2	-	-	V
V <sub>CM</sub>	differential common mode range		0.8	-	2.5	V
<b>Output levels for full-speed and low-speed</b>						
V <sub>OH</sub>	HIGH-level output voltage		2.8	-	3.6	V
V <sub>OL</sub>	LOW-level output voltage		0	-	0.3	V
V <sub>OSEI</sub>	SEI		0.8	-	-	V
V <sub>CRS</sub>	output signal crossover point 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 156: Static characteristics: REF5V**

$V_{CC(I/O)} = 1.65\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V <sub>IH</sub>	HIGH-level input voltage		-	5	-	V

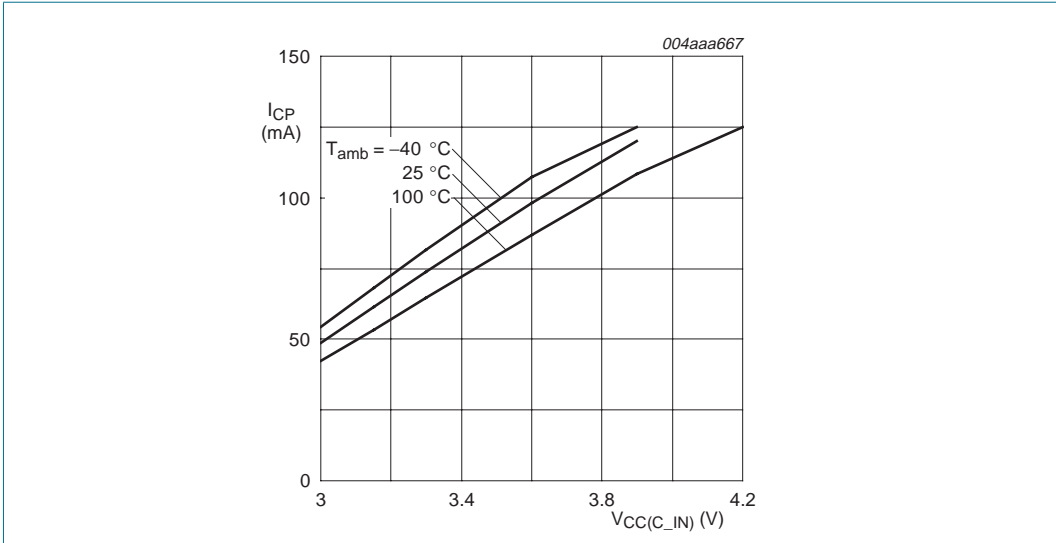


Fig 17. Charge pump current versus voltage at various temperatures (worst case)

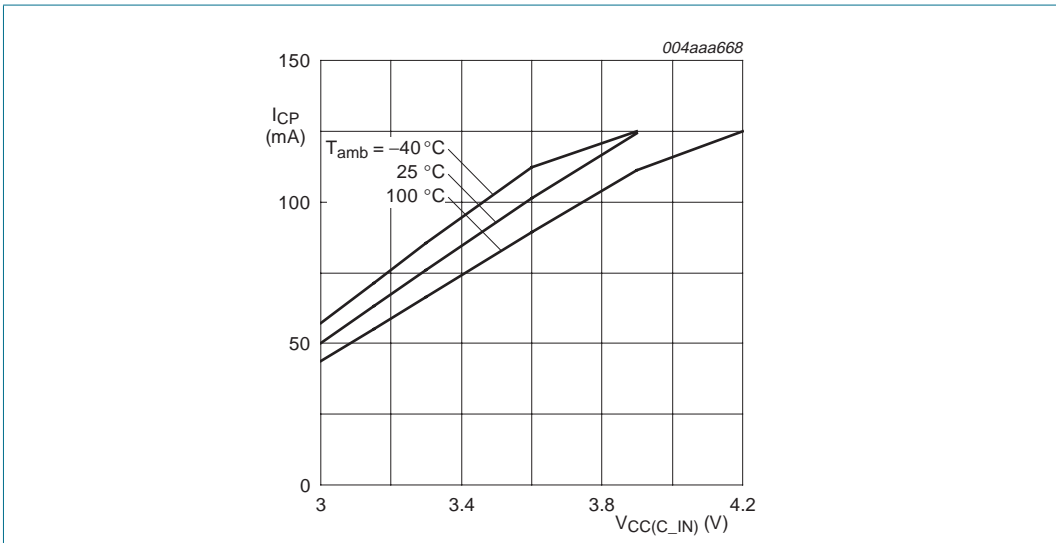


Fig 18. Charge pump current versus voltage at various temperatures (typical case)

## 15. Dynamic characteristics

**Table 157: Dynamic characteristics: system clock timing**

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Crystal oscillator</b>						
f <sub>clk</sub>	clock frequency [1]	crystal [2]	-	12	-	MHz
		oscillator	-	12	-	MHz
<b>External clock input</b>						
J	external clock jitter		-	-	500	ps
δ	clock duty cycle		-	50	-	%
V <sub>clk</sub>	amplitude		-	V <sub>CC(I/O)</sub>	-	V
t <sub>CR</sub> , t <sub>CF</sub>	rise time and fall time		-	-	3	ns

[1] Recommended accuracy of the clock frequency is 50 ppm for the crystal and oscillator. The oscillator used depends on V<sub>CC(I/O)</sub>.

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

**Table 158: Dynamic characteristics: CPU interface block**

V<sub>CC(I/O)</sub> = 1.65 V to 3.6 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.

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

**Table 159: Dynamic characteristics: high-speed source electrical characteristics**

V<sub>CC(I/O)</sub> = 1.65 V to 3.6 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Driver characteristics</b>						
t <sub>HSR</sub>	high-speed differential rise time	10 % to 90 %	500	-	-	ps
t <sub>HSF</sub>	high-speed differential fall time	90 % to 10 %	500	-	-	ps
Z <sub>HSDRV</sub>	drive output resistance (this also serves as a high-speed termination)	includes the R <sub>S</sub> resistor	40.5	45	49.5	Ω
<b>Clock timing</b>						
t <sub>HSDRAT</sub>	data rate		479.76	-	480.24	Mbit/s
t <sub>HSFRAM</sub>	microframe interval		124.9375	-	125.0625	μs
t <sub>HSRFI</sub>	consecutive microframe interval difference		1	-	four high-speed bit times	ns

**Table 160: Dynamic characteristics: full-speed source electrical characteristics**

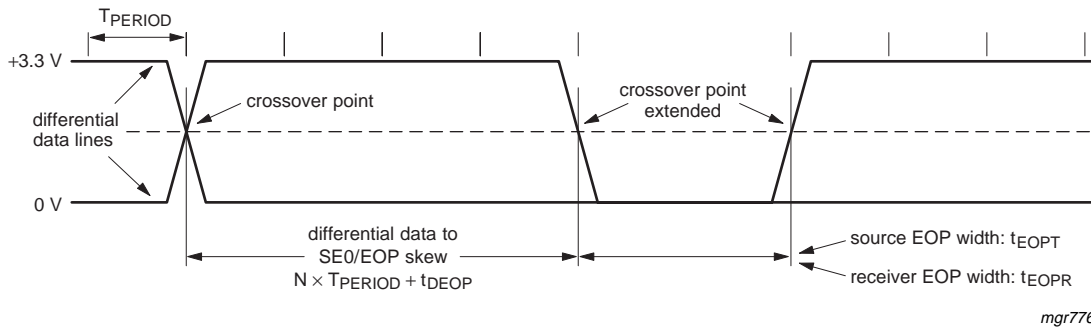
$V_{CC(I/O)} = 1.65\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Driver characteristics</b>						
$t_{FR}$	rise time	$C_L = 50\text{ pF}$ ; 10 % to 90 % of $ V_{OH} - V_{OL} $	4	-	20	ns
$t_{FF}$	fall time	$C_L = 50\text{ pF}$ ; 90 % to 10 % of $ V_{OH} - V_{OL} $	4	-	20	ns
$t_{FRFM}$	differential rise and fall time matching		90	-	111.1	%
$Z_{DRV}$	driver output resistance for the driver that is not high-speed capable		28	-	44	$\Omega$
<b>Data timing: see Figure 19</b>						
$t_{FDEOP}$	source jitter for differential transition to SEO 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
$t_{LDEOP}$	source jitter for differential transition to SEO transition	low-speed timing	-40	-	+100	ns
$t_{LEOPT}$	source SE0 interval of EOP		1.25	-	1.5	$\mu\text{s}$
$t_{LEOPR}$	receiver SE0 interval of EOP		670	-	-	ns
$t_{FST}$	width of SE0 interval during differential transaction		-	-	14	ns

**Table 161: Dynamic characteristics: low-speed source electrical characteristics**

$V_{CC(I/O)} = 1.65\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Driver characteristics</b>						
$t_{LR}$	rise time		75	-	300	ns
$t_{LF}$	fall time		75	-	300	ns
$t_{LRFM}$	differential rise and fall time matching		90	-	125	%



$T_{PERIOD}$  is the bit duration corresponding with the USB data rate.  
 $t_{DEOP}$  is the source jitter for differential transition to SE0 transition.  
 Full-speed timing symbols have a subscript prefix 'F', low-speed timing symbols have a prefix 'L'.

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

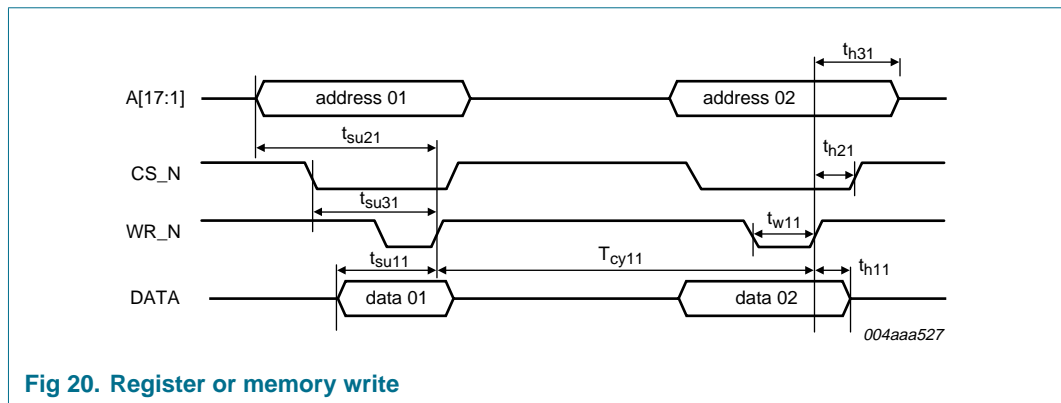


Fig 20. Register or memory write

Table 162: Register or memory write

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^\circ\text{C to }+85\text{ }^\circ\text{C}$ ; unless otherwise specified.

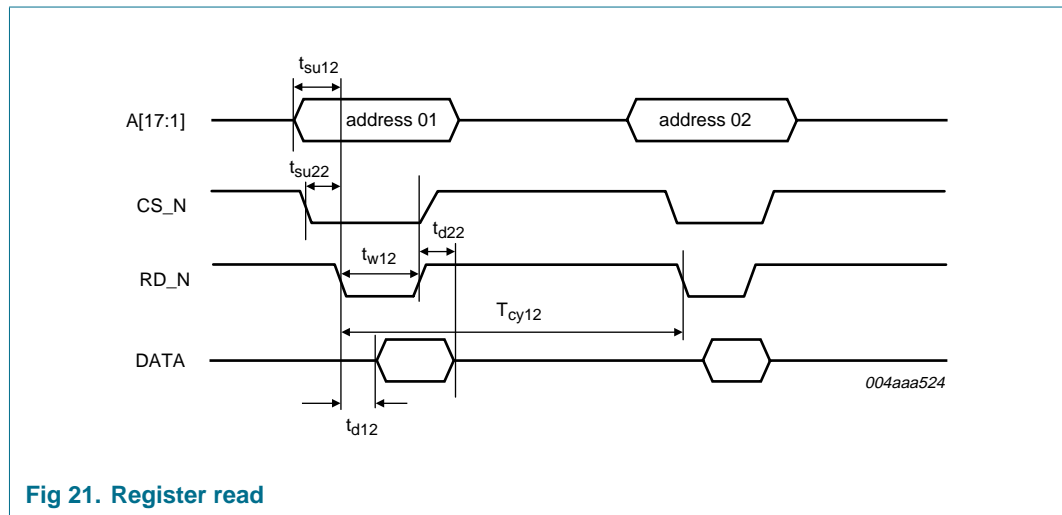
Symbol	Parameter	Min	Max	Unit
$t_{h11}$	data hold after WR_N HIGH	2	-	ns
$t_{h21}$	CS_N hold after WR_N HIGH	1	-	ns
$t_{h31}$	address hold after WR_N HIGH	2	-	ns
$t_{w11}$	WR_N pulse width	17	-	ns
$T_{cy11}$	WR_N to WR_N cycle time	36	-	ns
$t_{su11}$	data setup time before WR_N HIGH	5	-	ns
$t_{su21}$	address setup time before WR_N HIGH	5	-	ns
$t_{su31}$	CS_N setup time before WR_N HIGH	5	-	ns

**Table 163: Register or memory write**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{h11}$	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
$T_{cy11}$	WR_N to WR_N cycle time	36	-	ns
$t_{su11}$	data setup time before WR_N HIGH	5	-	ns
$t_{su21}$	address setup time before WR_N HIGH	5	-	ns
$t_{su31}$	CS_N setup time before WR_N HIGH	5	-	ns

15.1.1.2 Register read



**Fig 21. Register read**

**Table 164: Register read**

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{su12}$	address setup time before RD_N LOW	0	-	ns
$t_{su22}$	CS_N setup 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
$T_{cy12}$	read-to-read cycle time	40	-	ns

**Table 165: Register read**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{su12}$	address setup time before RD_N LOW	0	-	ns
$t_{su22}$	CS_N setup time before RD_N LOW	0	-	ns
$t_{w12}$	RD_N pulse width	$t_{d12}$	-	ns

**Table 165: Register read...continued**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{d12}$	data valid time after RD_N LOW	-	22	ns
$t_{d22}$	data valid time after RD_N HIGH	-	1	ns
$T_{cy12}$	read-to-read cycle time	36	-	ns

15.1.1.3 Memory read

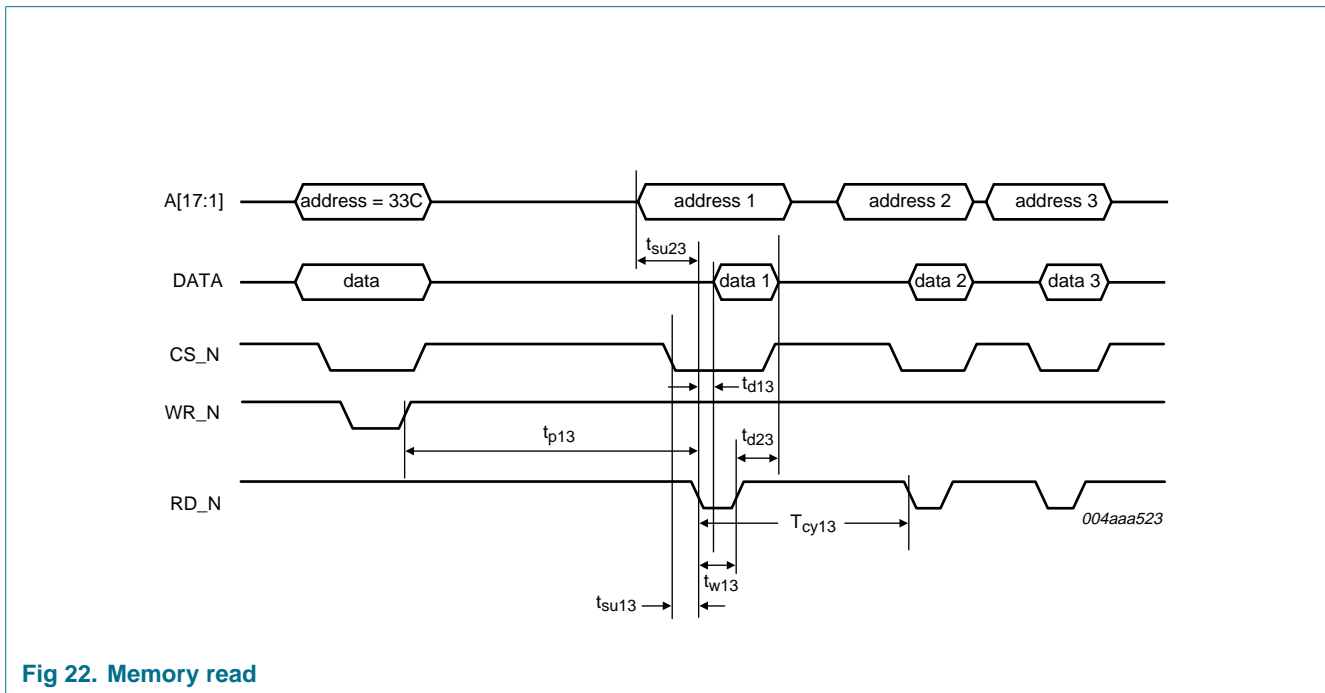


Fig 22. Memory read

**Table 166: Memory read**

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{p13}$	initial prefetch 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	$t_{d13}$	-	ns
$t_{su13}$	CS_N setup time before RD_N LOW	0	-	ns
$t_{su23}$	address setup time before RD_N LOW	0	-	ns

**Table 167: Memory read**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{p13}$	initial prefetch 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 167: Memory read...continued**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

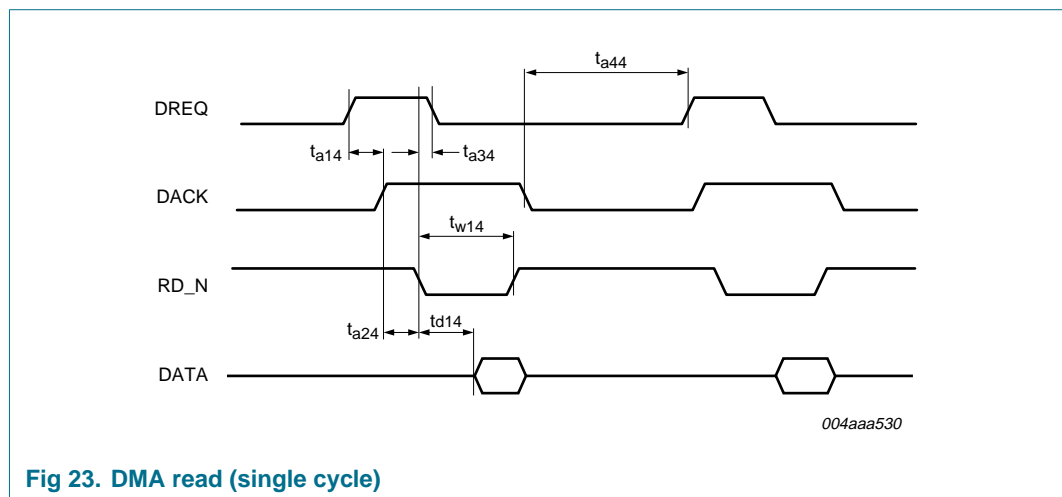
Symbol	Parameter	Min	Max	Unit
$t_{w13}$	RD_N pulse width	$t_{d13}$	-	ns
$t_{su13}$	CS_N setup time before RD_N LOW	0	-	ns
$t_{su23}$	address setup time before RD_N LOW	0	-	ns

**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 23. DMA read (single cycle)**

**Table 168: DMA read (single cycle)**

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{a14}$	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 deassertion time after RD_N assertion	23	-	ns
$t_{a44}$	DACK deassertion to next DREQ assertion time	-	56	ns

**Table 169: DMA read (single cycle)**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

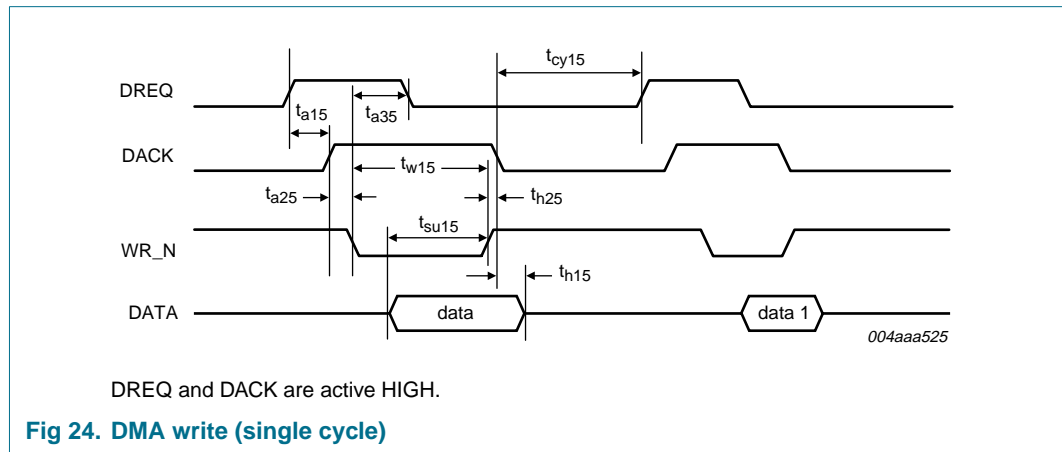
Symbol	Parameter	Min	Max	Unit
$t_{a14}$	DACK assertion time after DREQ assertion	0	-	ns
$t_{a24}$	RD_N assertion time after DACK assertion	0	-	ns
$t_{d14}$	data valid time after RD_N assertion	-	20	ns

**Table 169: DMA read (single cycle)...***continued*

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{w14}$	RD_N pulse width	$t_{d14}$	-	ns
$t_{a34}$	DREQ deassertion time after RD_N assertion	11	-	ns
$t_{a44}$	DACK deassertion to next DREQ assertion time	-	56	ns

**15.1.2.2 Single cycle: DMA write**



**Table 170: DMA write (single cycle)**

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{a15}$	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 deassertion	3	-	ns
$t_{h25}$	DACK hold time after WR_N deassertion	0	-	ns
$t_{su15}$	data setup time before WR_N deassertion	5.5	-	ns
$t_{a35}$	DREQ deassertion time after WR_N assertion	22	-	ns
$t_{cy15}$	last DACK strobe deassertion to next DREQ assertion time	82	-	ns
$t_{w15}$	WR_N pulse width	22	-	ns

**Table 171: DMA write (single cycle)**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

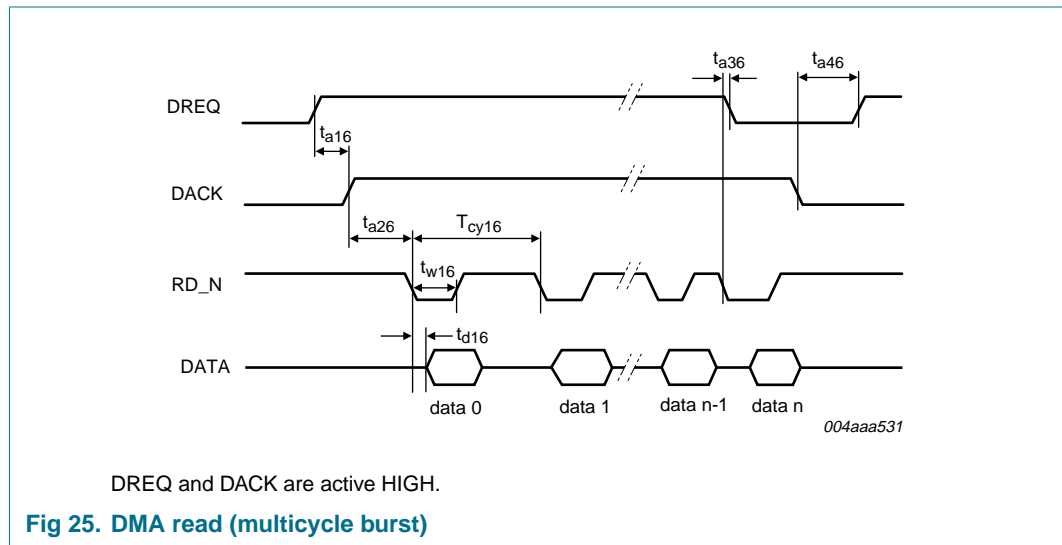
Symbol	Parameter	Min	Max	Unit
$t_{a15}$	DACK assertion time after DREQ assertion	0	-	ns
$t_{a25}$	WR_N assertion time after DACK assertion	1	-	ns
$t_{h15}$	data hold time after WR_N deassertion	2	-	ns
$t_{h25}$	DACK hold time after WR_N deassertion	0	-	ns
$t_{su15}$	data setup time before WR_N deassertion	5.5	-	ns

**Table 171: DMA write (single cycle)...***continued*

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{a35}$	DREQ deassertion time after WR_N assertion	8.9	-	ns
$t_{cy15}$	last DACK strobe deassertion to next DREQ assertion time	82	-	ns
$t_{w15}$	WR_N pulse width	22	-	ns

**15.1.2.3 Multicycle: DMA read**



**Table 172: DMA read (multicycle burst)**

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{a16}$	DACK assertion time after DREQ assertion	0	-	ns
$t_{a26}$	RD_N assertion time after DACK assertion	0	-	ns
$t_{d16}$	data valid time after RD_N assertion	-	31	ns
$t_{w16}$	RD_N pulse width	$t_{d16}$	-	ns
$T_{cy16}$	read-to-read cycle time	40	-	ns
$t_{a36}$	DREQ deassertion time after last burst RD_N deassertion	20	-	ns
$t_{a46}$	DACK deassertion to next DREQ assertion time	-	82	ns

**Table 173: DMA read (multicycle burst)**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{a16}$	DACK assertion time after DREQ assertion	0	-	ns
$t_{a26}$	RD_N assertion time after DACK assertion	0	-	ns
$t_{d16}$	data valid time after RD_N assertion	-	16	ns
$t_{w16}$	RD_N pulse width	$t_{d16}$	-	ns
$T_{cy16}$	read-to-read cycle time	36	-	ns
$t_{a36}$	DREQ deassertion time after last burst RD_N deassertion	11	-	ns
$t_{a46}$	DACK deassertion to next DREQ assertion time	-	82	ns

15.1.2.4 Multicycle: DMA write

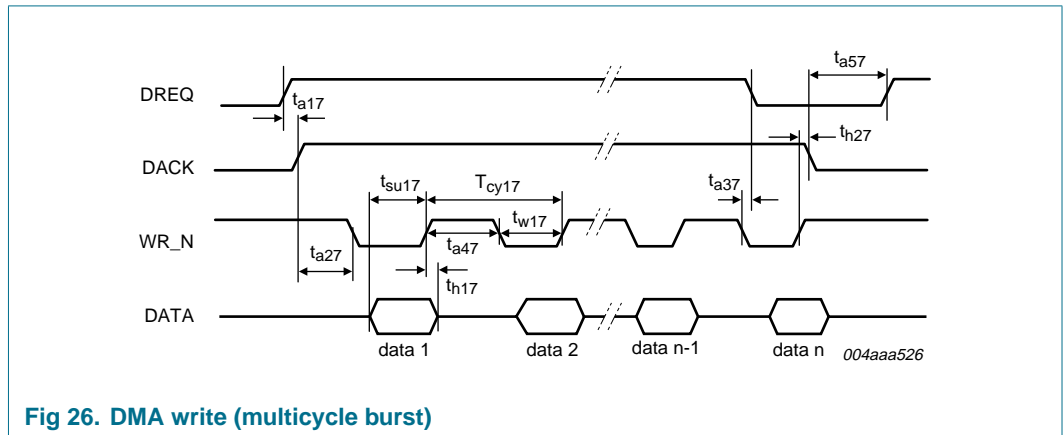


Fig 26. DMA write (multicycle burst)

Table 174: DMA write (multicycle burst)

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^\circ\text{C to }+85\text{ }^\circ\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$T_{cy17}$	DMA write cycle time	51	-	ns
$t_{su17}$	data setup time before WR_N deassertion	5	-	ns
$t_{h17}$	data hold time after WR_N deassertion	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 deassertion time at last strobe (WR_N) assertion	20	-	ns
$t_{h27}$	DACK hold time after WR_N deassertion	0	-	ns
$t_{a47}$	strobe deassertion to next strobe assertion time	34	-	ns
$t_{w17}$	WR_N pulse width	17	-	ns
$t_{a57}$	DACK deassertion to next DREQ assertion time	-	82	ns

Table 175: DMA write (multicycle burst)

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^\circ\text{C to }+85\text{ }^\circ\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$T_{cy17}$	DMA write cycle time	51	-	ns
$t_{su17}$	data setup time before WR_N deassertion	5	-	ns
$t_{h17}$	data hold time after WR_N deassertion	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 deassertion time at last strobe (WR_N) assertion	0	-	ns
$t_{h27}$	DACK hold time after WR_N deassertion	0	-	ns
$t_{a47}$	strobe deassertion to next strobe assertion time	34	-	ns
$t_{w17}$	WR_N pulse width	17	-	ns
$t_{a57}$	DACK deassertion 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

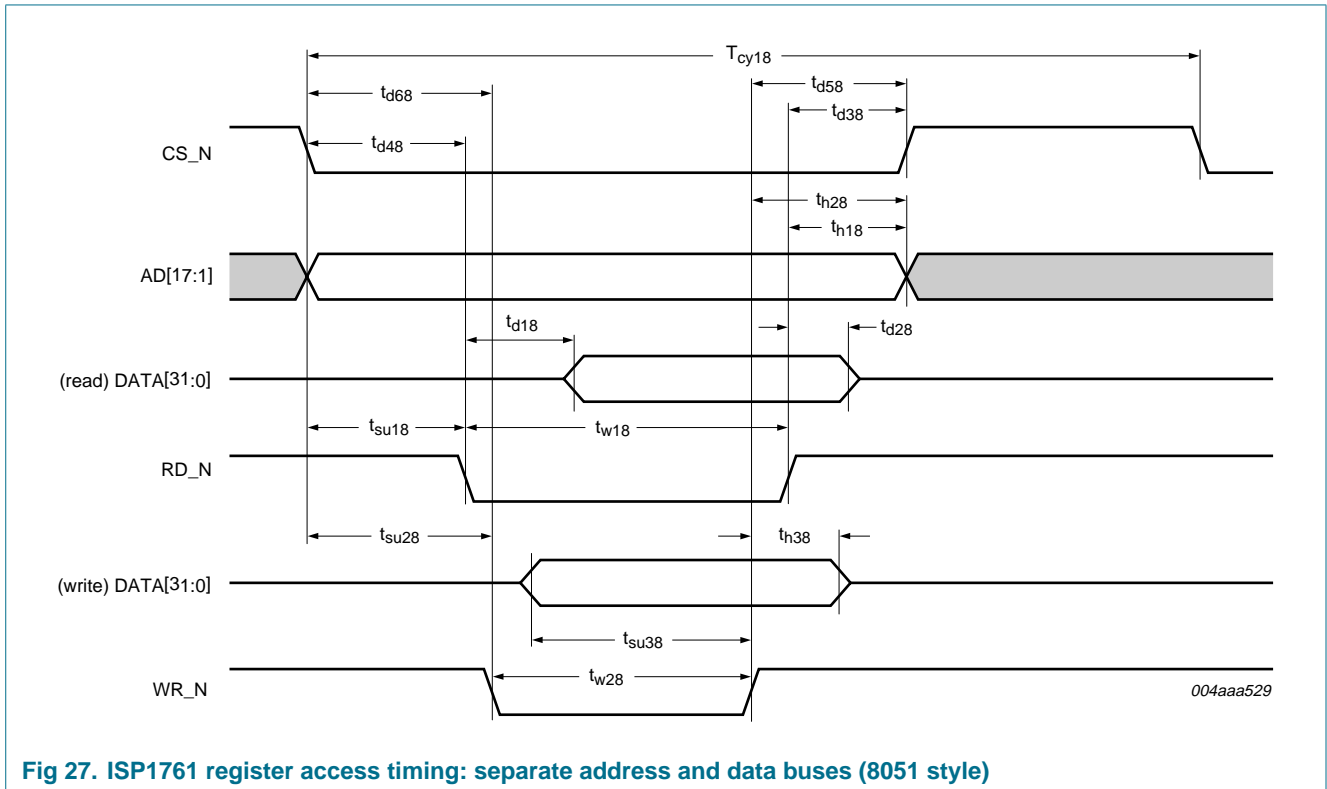


Fig 27. ISP1761 register access timing: separate address and data buses (8051 style)

Table 176: PIO register read or write

$V_{CC(I/O)} = 1.65 V$  to  $1.95 V$ ;  $T_{amb} = -40^{\circ}C$  to  $+85^{\circ}C$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
<b>Reading</b>				
$t_{w18}$	RD_N LOW pulse width	$t_{d1}$	-	ns
$t_{su18}$	address setup 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 three-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
<b>Writing</b>				
$t_{w28}$	WR_N LOW pulse width	15	-	ns
$t_{su28}$	address setup time before WR_N LOW	0	-	ns
$t_{h28}$	address hold time after WR_N HIGH	0	-	ns
$t_{su38}$	data setup 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 176: PIO register read or write...continued**

$V_{CC(I/O)} = 1.65\text{ V to }1.95\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$t_{d68}$	CS_N LOW to WR_N LOW delay	0	-	ns
<b>General</b>				
$T_{cy18}$	read or write cycle time	40	-	ns

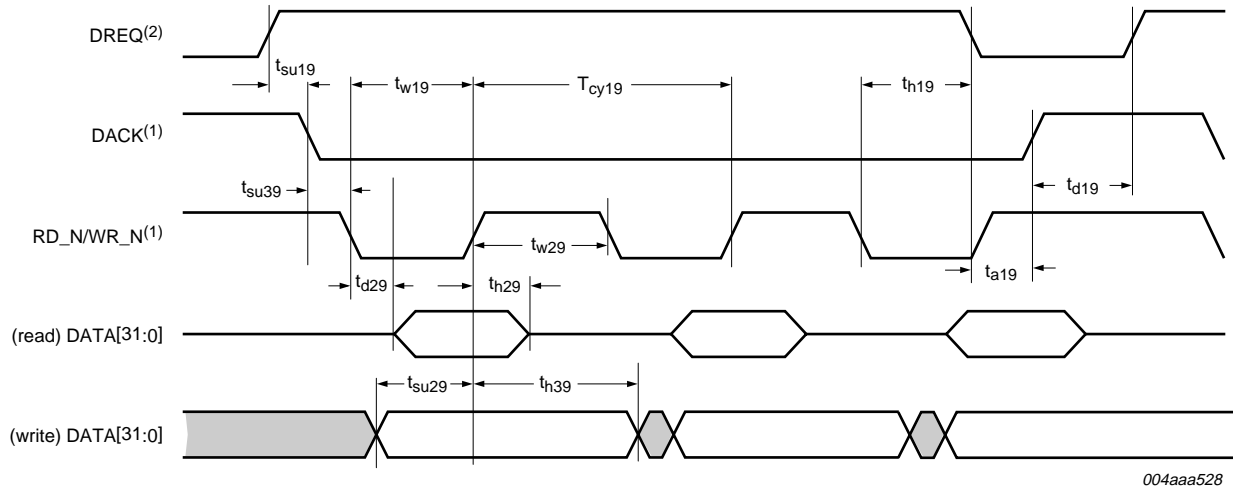
**Table 177: PIO register read or write**

$V_{CC(I/O)} = 3.3\text{ V to }3.6\text{ V}$ ;  $T_{amb} = -40\text{ }^{\circ}\text{C to }+85\text{ }^{\circ}\text{C}$ ; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
<b>Reading</b>				
$t_{w18}$	RD_N LOW pulse width	$t_{d1}$	-	ns
$t_{su18}$	address setup 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 three-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
<b>Writing</b>				
$t_{w28}$	WR_N LOW pulse width	15	-	ns
$t_{su28}$	address setup time before WR_N LOW	0	-	ns
$t_{h28}$	address hold time after WR_N HIGH	1	-	ns
$t_{su38}$	data setup 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
<b>General</b>				
$T_{cy18}$	read or write cycle time	40	-	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 28. DMA read or write

Table 178: DMA read or write

V<sub>CC(I/O)</sub> = 1.65 V to 1.95 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
T <sub>cy19</sub>	read or write cycle time	75	-	ns
t <sub>su19</sub>	DREQ setup time before first DACK on	10	-	ns
t <sub>d19</sub>	DREQ on delay after last strobe off	33.33	-	ns
t <sub>h19</sub>	DREQ hold time after last strobe on	0	53	ns
t <sub>w19</sub>	RD_N/WR_N pulse width	40	600	ns
t <sub>w29</sub>	RD_N/WR_N recovery time	36	-	ns
t <sub>d29</sub>	read data valid delay after strobe on	-	30	ns
t <sub>h29</sub>	read data hold time after strobe off	-	5	ns
t <sub>h39</sub>	write data hold time after strobe off	1	-	ns
t <sub>su29</sub>	write data setup time before strobe off	10	-	ns
t <sub>su39</sub>	DACK setup time before RD_N/WR_N assertion	0	-	ns
t <sub>a19</sub>	DACK deassertion after RD_N/WR_N deassertion	0	30	ns

**Table 179: DMA read or write***V<sub>CC(I/O)</sub> = 3.3 V to 3.6 V; T<sub>amb</sub> = -40 °C to +85 °C; unless otherwise specified.*

Symbol	Parameter	Min	Max	Unit
T <sub>cy19</sub>	read or write cycle time	75	-	ns
t <sub>su19</sub>	DREQ setup time before first DACK on	10	-	ns
t <sub>d19</sub>	DREQ on delay after last strobe off	33.33	-	ns
t <sub>h19</sub>	DREQ hold time after last strobe on	0	53	ns
t <sub>w19</sub>	RD_N/WR_N pulse width	39	600	ns
t <sub>w29</sub>	RD_N/WR_N recovery time	36	-	ns
t <sub>d29</sub>	read data valid delay after strobe on	-	20	ns
t <sub>h29</sub>	read data hold time after strobe off	-	5	ns
t <sub>h39</sub>	write data hold time after strobe off	1	-	ns
t <sub>su29</sub>	write data setup time before strobe off	10	-	ns
t <sub>su39</sub>	DACK setup time before RD_N/WR_N assertion	0	-	ns
t <sub>a19</sub>	DACK deassertion after RD_N/WR_N deassertion	0	30	ns



16. Package outline

LQFP128: plastic low profile quad flat package; 128 leads; body 14 x 20 x 1.4 mm

SOT425-1

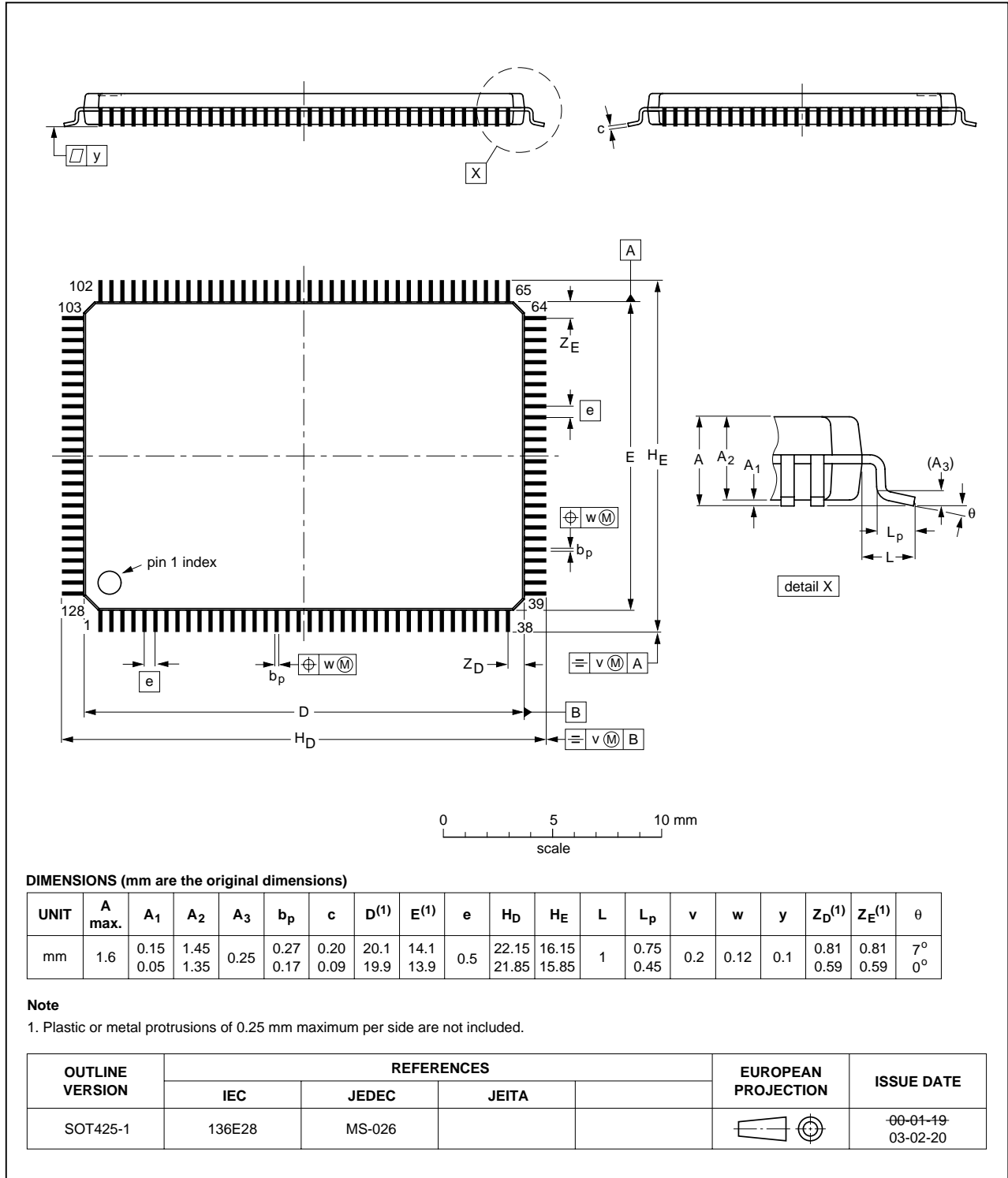


Fig 29. Package outline (LQFP128)

TFBGA128: plastic thin fine-pitch ball grid array package; 128 balls; body 9 x 9 x 0.8 mm

SOT857-1

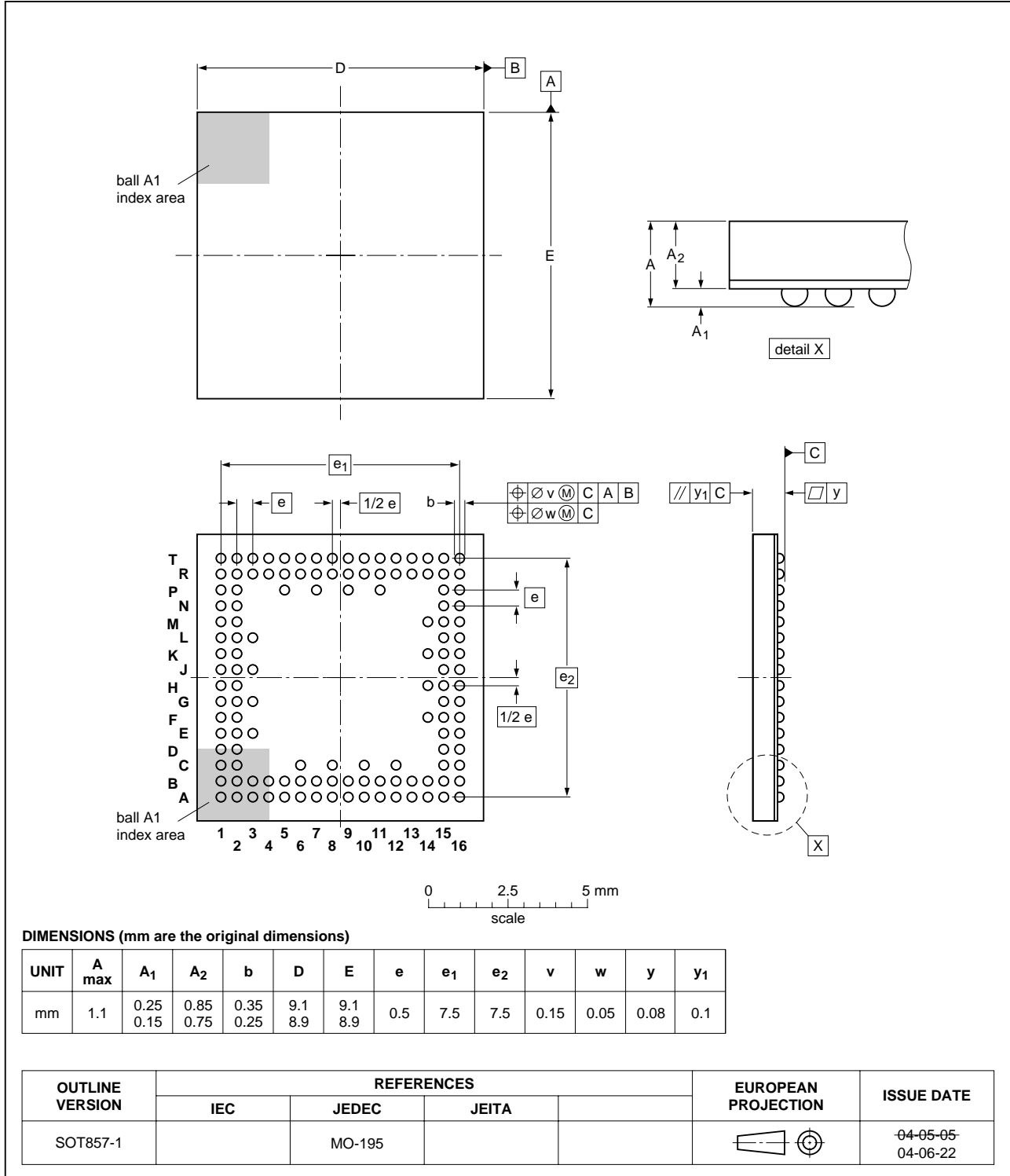


Fig 30. Package outline (TFBGA128)

## 17. Soldering

### 17.1 Introduction to soldering surface mount packages

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

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

### 17.2 Reflow soldering

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

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

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

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

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

### 17.3 Wave soldering

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

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

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

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

- smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

The footprint must incorporate solder thieves at the downstream end.

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

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

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

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

### 17.4 Manual soldering

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

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

### 17.5 Package related soldering information

**Table 180: Suitability of surface mount IC packages for wave and reflow soldering methods**

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

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

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

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

- [4] These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.
- [5] If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
- [6] Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.
- [7] Wave soldering is suitable for SSOP, TSSOP, VSO and VSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.
- [8] Image sensor packages in principle should not be soldered. They are mounted in sockets or delivered pre-mounted on flex foil. However, the image sensor package can be mounted by the client on a flex foil by using a hot bar soldering process. The appropriate soldering profile can be provided on request.
- [9] Hot bar soldering or manual soldering is suitable for PMFP packages.

## 18. Abbreviations

Table 181: Abbreviations

Acronym	Description
ATL	Acknowledged Transfer List
DMA	Direct Memory Access
DSC	Digital Still Camera
EHCI	Enhanced Host Controller Interface
EMI	Electro-Magnetic Interference
FS	full-speed
FIFO	First In, First Out
GPS	Global Positioning System
HC	Host Controller
HNP	Host Negotiation Protocol
HS	high-speed
iTD	isochronous Transfer Descriptor
ISR	Interrupt Service Routine
INT	INTerrupt
ISO	ISOchronous
ITL	Isochronous (ISO) Transfer List
LS	low-speed
OHCI	Open Host Controller Interface
OTG	On-the-Go
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistant
PLL	Phase-Locked Loop
PIO	Programmed Input/Output
PMOS	Positive-channel Metal-Oxide Semiconductor
POR	Power-On Reset
PTD	Philips Transfer Descriptor
QHA	Queue Head Asynchronous
QHA-SS/CS	Queue Head Asynchronous Start Split and Complete Split

Table 181: Abbreviations...continued

Acronym	Description
QHP	Queue Head Periodic
RISC	Reduced Instruction Set Computer
SiTD	Split isochronous Transfer Descriptor
SIE	Serial Interface Engine
SRP	Session Request Protocol
TT	Transaction Translator
UHCI	Universal Host Controller Interface
USB	Universal Serial Bus

## 19. 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. 1.0a*
- [4] *Interfacing the ISP76x to the Intel® PXA250 Processor (AN10037)*
- [5] *ISP1761 Peripheral DMA Initialization (AN10040)*
- [6] *ISP176x Linux Programming Guide (AN10042)*
- [7] *Embedded Systems Design with the ISP176x (AN10043)*
- [8] *ISP1581 Programming Guide (AN10004)*
- [9] *ISP1582/83 Control Pipe (AN10031).*

## 20. Revision history

Table 182: Revision history

Document ID	Release date	Data sheet status	Change notice	Doc. number	Supersedes
ISP1761_1	20050112	Product data sheet	-	9397 750 13258	-

## 21. Data sheet status

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

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

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

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

## 22. Definitions

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

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

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

## 23. Disclaimers

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

## 25. Contact information

For additional information, please visit: <http://www.semiconductors.philips.com>

For sales office addresses, send an email to: [sales.addresses@www.semiconductors.philips.com](mailto:sales.addresses@www.semiconductors.philips.com)

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

## 24. Trademarks

**DragonBall** — is a trademark of Motorola, Inc.

**Hitachi** — is a registered trademark of Hitachi, Ltd.

**Intel** — is a registered trademark of Intel Corporation.

**Motorola** — is a registered trademark of Motorola, Inc.

**NEC** — is a registered trademark of NEC Corporation.

**PowerPC** — is a registered trademark of IBM Corporation.

**SoftConnect** — is a trademark of Koninklijke Philips Electronics N.V.

**StrongARM** — is a registered trademark of ARM Limited.

**Toshiba** — is a registered trademark of Toshiba Corporation.

26. Tables

Table 1: Ordering information . . . . .	4	Table 48: Edge Interrupt Count register: bit description . . .	49
Table 2: Pin description . . . . .	7	Table 49: DMA Start Address register: bit allocation . . .	49
Table 3: Port connection scenarios . . . . .	16	Table 50: DMA Start Address register: bit description . . .	50
Table 4: Memory address . . . . .	18	Table 51: Power Down Control register: bit allocation . . .	50
Table 5: Using the IRQ Mask AND or IRQ Mask OR registers . . . . .	24	Table 52: Power Down Control register: bit description . . .	51
Table 6: Hybrid mode . . . . .	28	Table 53: HcInterrupt register: bit allocation . . . . .	52
Table 7: Pin status during hybrid mode . . . . .	28	Table 54: HcInterrupt register: bit description . . . . .	53
Table 8: Host Controller-specific register overview . . . . .	31	Table 55: HcInterruptEnable register: bit allocation . . . . .	54
Table 9: CAPLENGTH register: bit description . . . . .	32	Table 56: HcInterruptEnable register: bit description . . . . .	55
Table 10: HCIVERSION register: bit description . . . . .	32	Table 57: ISO IRQ MASK OR register: bit description . . . . .	56
Table 11: HCSPARAMS register: bit allocation . . . . .	33	Table 58: INT IRQ MASK OR register: bit description . . . . .	56
Table 12: HCSPARAMS register: bit description . . . . .	33	Table 59: ATL IRQ MASK OR register: bit description . . . . .	56
Table 13: HCCPARAMS register: bit allocation . . . . .	34	Table 60: ISO IRQ MASK AND register: bit description . . . . .	57
Table 14: HCCPARAMS register: bit description . . . . .	34	Table 61: INT IRQ MASK AND register: bit description . . . . .	57
Table 15: USBCMD register: bit allocation . . . . .	35	Table 62: ATL IRQ MASK SAND register: bit description . . . . .	57
Table 16: USBCMD register: bit description . . . . .	35	Table 63: High-speed bulk IN and OUT, QHA: bit allocation . . . . .	60
Table 17: USBSTS register: bit allocation . . . . .	36	Table 64: High-speed bulk IN and OUT, QHA: bit description . . . . .	61
Table 18: USBSTS register: bit description . . . . .	36	Table 65: High-speed isochronous IN and OUT, iTD: bit allocation . . . . .	64
Table 19: FRINDEX register: bit allocation . . . . .	37	Table 66: High-speed isochronous IN and OUT, iTD: bit description . . . . .	65
Table 20: FRINDEX register: bit description . . . . .	37	Table 67: High-speed interrupt IN and OUT, QHP: bit allocation . . . . .	68
Table 21: CONFIGFLAG register: bit allocation . . . . .	38	Table 68: High-speed interrupt IN and OUT, QHP: bit description . . . . .	69
Table 22: CONFIGFLAG register: bit description . . . . .	38	Table 69: Start and complete split for bulk, QHASS/CS: bit allocation . . . . .	72
Table 23: PORTSC 1 register: bit allocation . . . . .	39	Table 70: Start and complete split for bulk, QHASS/CS: bit description . . . . .	73
Table 24: PORTSC 1 register: bit description . . . . .	39	Table 71: Start and complete split for isochronous, SiTD: bit allocation . . . . .	76
Table 25: ISO PTD Done Map register: bit description . . . . .	40	Table 72: Start and complete split for isochronous, SiTD: bit description . . . . .	77
Table 26: ISO PTD Skip Map register: bit description . . . . .	40	Table 73: Start and complete split for interrupt: bit allocation . . . . .	80
Table 27: ISO PTD Last PTD register: bit description . . . . .	40	Table 74: Start and complete split for interrupt: bit description . . . . .	81
Table 28: INT PTD Done Map register: bit description . . . . .	41	Table 75: OTG Controller-specific register overview . . . . .	90
Table 29: INT PTD Skip Map register: bit description . . . . .	41	Table 76: Address mapping of registers: 32-bit data bus mode . . . . .	90
Table 30: INT PTD Last PTD register: bit description . . . . .	41	Table 77: Address mapping of registers: 16-bit data bus mode . . . . .	91
Table 31: ATL PTD Done Map register: bit description . . . . .	42	Table 78: Vendor ID register: bit description . . . . .	91
Table 32: ATL PTD Skip Map register: bit description . . . . .	42	Table 79: Product ID register: bit description . . . . .	91
Table 33: ATL PTD Last PTD register: bit description . . . . .	42	Table 80: OTG Control register: bit allocation . . . . .	92
Table 34: HW Mode Control register: bit allocation . . . . .	43	Table 81: OTG Control register: bit description . . . . .	92
Table 35: HW Mode Control register: bit description . . . . .	43		
Table 36: HcChipID register: bit description . . . . .	44		
Table 37: HcScratch register: bit description . . . . .	44		
Table 38: SW Reset register: bit allocation . . . . .	45		
Table 39: SW Reset register: bit description . . . . .	45		
Table 40: HcDMAConfiguration register: bit allocation . . . . .	45		
Table 41: HcDMAConfiguration register: bit description . . . . .	46		
Table 42: HcBufferStatus register: bit allocation . . . . .	46		
Table 43: HcBufferStatus register: bit description . . . . .	47		
Table 44: ATL Done Timeout register: bit description . . . . .	47		
Table 45: Memory register: bit allocation . . . . .	48		
Table 46: Memory register: bit description . . . . .	48		
Table 47: Edge Interrupt Count register: bit allocation . . . . .	48		

continued >>



Table 82: OTG Status register: bit allocation . . . . .	93	Table 124: DMA Transfer Counter register: bit	
Table 83: OTG Status register: bit description . . . . .	93	description . . . . .	116
Table 84: OTG Interrupt Latch register: bit allocation . . .	94	Table 125: DcDMAConfiguration register: bit allocation . .	117
Table 85: OTG Interrupt Latch register: bit description . .	94	Table 126: DcDMAConfiguration register: bit description . .	117
Table 86: OTG Interrupt Enable Fall register: bit		Table 127: DMA Hardware register: bit allocation . . . . .	117
allocation . . . . .	95	Table 128: DMA Hardware register: bit description . . . . .	118
Table 87: OTG Interrupt Enable Fall register: bit		Table 129: DMA Interrupt Reason register: bit allocation . .	118
description . . . . .	95	Table 130: DMA Interrupt Reason register: bit	
Table 88: OTG Interrupt Enable Rise register: bit		description . . . . .	119
allocation . . . . .	96	Table 131: Internal EOT-functional relation with the	
Table 89: OTG Interrupt Enable Rise register: bit		DMA_XFER_OK bit . . . . .	119
description . . . . .	96	Table 132: DMA Interrupt Enable register: bit allocation . .	119
Table 90: OTG Timer register: bit allocation . . . . .	97	Table 133: DMA Endpoint register: bit allocation . . . . .	120
Table 91: OTG Timer register: bit description . . . . .	97	Table 134: DMA Endpoint register: bit description . . . . .	120
Table 92: Endpoint access and programmability . . . . .	100	Table 135: DMA Burst Counter register: bit allocation . . .	120
Table 93: Peripheral Controller-specific register		Table 136: DMA Burst Counter register: bit description . .	121
overview . . . . .	102	Table 137: DcInterrupt register: bit allocation . . . . .	121
Table 94: Address register: bit allocation . . . . .	103	Table 138: DcInterrupt register: bit description . . . . .	122
Table 95: Address register: bit description . . . . .	103	Table 139: DcChipID register: bit description . . . . .	123
Table 96: Mode register: bit allocation . . . . .	103	Table 140: Frame Number register: bit allocation . . . . .	123
Table 97: Mode register: bit description . . . . .	103	Table 141: Frame Number register: bit description . . . . .	123
Table 98: Interrupt Configuration register: bit allocation .	105	Table 142: DcScratch register: bit allocation . . . . .	124
Table 99: Interrupt Configuration register: bit		Table 143: DcScratch register: bit description . . . . .	124
description . . . . .	105	Table 144: Unlock Device register: bit allocation . . . . .	124
Table 100: Debug mode settings . . . . .	105	Table 145: Unlock Device register: bit description . . . . .	125
Table 101: Debug register: bit allocation . . . . .	105	Table 146: Interrupt Pulse Width register: bit description .	125
Table 102: Debug register: bit allocation . . . . .	106	Table 147: Test Mode register: bit allocation . . . . .	125
Table 103: DcInterruptEnable register: bit allocation . . .	106	Table 148: Test Mode register: bit description . . . . .	125
Table 104: DcInterruptEnable register: bit description . .	107	Table 149: Power consumption . . . . .	127
Table 105: Endpoint Index register: bit allocation . . . . .	108	Table 150: Absolute maximum ratings . . . . .	128
Table 106: Endpoint Index register: bit description . . . .	108	Table 151: Recommended operating conditions . . . . .	128
Table 107: Addressing of endpoint 0 buffers . . . . .	108	Table 152: Static characteristics: digital pins . . . . .	129
Table 108: Control Function register: bit allocation . . . .	109	Table 153: Static characteristics: digital pins . . . . .	129
Table 109: Control Function register: bit description . . .	109	Table 154: Static characteristics: PSW1_N, PSW2_N,	
Table 110: Data Port register: bit description . . . . .	110	PSW3_N . . . . .	129
Table 111: Buffer Length register: bit description . . . . .	111	Table 155: Static characteristics: USB interface block	
Table 112: DcBufferStatus register: bit allocation . . . . .	111	(pins DM1 to DM3 and DP1 to DP3) . . . . .	130
Table 113: DcBufferStatus register: bit description . . . .	111	Table 156: Static characteristics: REF5V . . . . .	130
Table 114: Endpoint MaxPacketSize register: bit		Table 157: Dynamic characteristics: system clock timing .	132
allocation . . . . .	112	Table 158: Dynamic characteristics: CPU interface block .	132
Table 115: Endpoint MaxPacketSize register: bit		Table 159: Dynamic characteristics: high-speed source	
description . . . . .	112	electrical characteristics . . . . .	132
Table 116: Programmable FIFO size . . . . .	112	Table 160: Dynamic characteristics: full-speed source	
Table 117: Endpoint Type register: bit allocation . . . . .	113	electrical characteristics . . . . .	133
Table 118: Endpoint Type register: bit description . . . . .	113	Table 161: Dynamic characteristics: low-speed source	
Table 119: Control bits for GDMA read or write		electrical characteristics . . . . .	133
(opcode = 00h/01h) . . . . .	114	Table 162: Register or memory write . . . . .	134
Table 120: DMA Command register: bit allocation . . . . .	115	Table 163: Register or memory write . . . . .	135
Table 121: DMA Command register: bit description . . . . .	115	Table 164: Register read . . . . .	135
Table 122: DMA commands . . . . .	115	Table 165: Register read . . . . .	135
Table 123: DMA Transfer Counter register: bit allocation .	116	Table 166: Memory read . . . . .	136

continued >>



Table 167: Memory read .....136  
 Table 168: DMA read (single cycle) .....137  
 Table 169: DMA read (single cycle) .....137  
 Table 170: DMA write (single cycle) .....138  
 Table 171: DMA write (single cycle) .....138  
 Table 172: DMA read (multicycle burst) .....139  
 Table 173: DMA read (multicycle burst) .....139  
 Table 174: DMA write (multicycle burst) .....140  
 Table 175: DMA write (multicycle burst) .....140  
 Table 176: PIO register read or write .....141  
 Table 177: PIO register read or write .....142  
 Table 178: DMA read or write .....143  
 Table 179: DMA read or write .....144  
 Table 180: Suitability of surface mount IC packages for wave  
 and reflow soldering methods .....148  
 Table 181: Abbreviations .....149  
 Table 182: Revision history .....150

continued >>

## 27. Figures

Fig 1.	Block diagram. . . . .	5
Fig 2.	Pin configuration (LQFP128); top view. . . . .	6
Fig 3.	Pin configuration (TFBGA128); top view. . . . .	6
Fig 4.	Internal hub. . . . .	15
Fig 5.	ISP1761 clock scheme. . . . .	15
Fig 6.	Memory segmentation and access block diagram. . . . .	19
Fig 7.	ISP1761 power supply connection. . . . .	27
Fig 8.	Most commonly used power supply connection. . . . .	27
Fig 9.	Hybrid mode. . . . .	28
Fig 10.	Adjusting analog overcurrent detection limit (optional). . . . .	29
Fig 11.	Internal power-on reset timing. . . . .	30
Fig 12.	Clock with respect to the external power-on reset. . . . .	30
Fig 13.	NextPTD traversal rule. . . . .	59
Fig 14.	HNP sequence of events. . . . .	86
Fig 15.	Dual-role A-device state diagram. . . . .	88
Fig 16.	Dual-role B-device state diagram. . . . .	89
Fig 17.	Charge pump current versus voltage at various temperatures (worst case). . . . .	131
Fig 18.	Charge pump current versus voltage at various temperatures (typical case). . . . .	131
Fig 19.	USB source differential data-to-EOP transition skew and EOP width. . . . .	134
Fig 20.	Register or memory write. . . . .	134
Fig 21.	Register read. . . . .	135
Fig 22.	Memory read. . . . .	136
Fig 23.	DMA read (single cycle). . . . .	137
Fig 24.	DMA write (single cycle). . . . .	138
Fig 25.	DMA read (multicycle burst). . . . .	139
Fig 26.	DMA write (multicycle burst). . . . .	140
Fig 27.	ISP1761 register access timing: separate address and data buses (8051 style). . . . .	141
Fig 28.	DMA read or write. . . . .	143
Fig 29.	Package outline (LQFP128). . . . .	145
Fig 30.	Package outline (TFBGA128). . . . .	146

continued >>

28. Contents

<b>1</b>	<b>General description</b> . . . . .	<b>1</b>	8.2.8	ISO PTD Done Map register (R: 0130h) . . . .	40
<b>2</b>	<b>Features</b> . . . . .	<b>1</b>	8.2.9	ISO PTD Skip Map register (R/W: 0134h) . . .	40
<b>3</b>	<b>Applications</b> . . . . .	<b>3</b>	8.2.10	ISO PTD Last PTD register (R/W: 0138h) . . .	40
3.1	Host/peripheral roles . . . . .	3	8.2.11	INT PTD Done Map register (R: 0140h) . . . .	41
<b>4</b>	<b>Ordering information</b> . . . . .	<b>4</b>	8.2.12	INT PTD Skip Map register (R/W: 0144h) . . .	41
<b>5</b>	<b>Block diagram</b> . . . . .	<b>5</b>	8.2.13	INT PTD Last PTD register (R/W: 0148h) . . .	41
<b>6</b>	<b>Pinning information</b> . . . . .	<b>6</b>	8.2.14	ATL PTD Done Map register (R: 0150h) . . . .	42
6.1	Pinning . . . . .	6	8.2.15	ATL PTD Skip Map register (R/W: 0154h) . . .	42
6.2	Pin description . . . . .	7	8.2.16	ATL PTD Last PTD register (R/W: 0158h) . . .	42
<b>7</b>	<b>Functional description</b> . . . . .	<b>14</b>	8.3	Configuration registers . . . . .	42
7.1	ISP1761 internal architecture: Advanced Philips Slave Host Controller and hub . . . . .	14	8.3.1	HW Mode Control register (R/W: 0300h) . . . .	42
7.1.1	Internal clock scheme . . . . .	15	8.3.2	HcChipID register (R: 0304h) . . . . .	44
7.2	Host Controller buffer memory block . . . . .	16	8.3.3	HcScratch register (R/W: 0308h) . . . . .	44
7.2.1	General considerations . . . . .	16	8.3.4	SW Reset register (R/W: 030Ch) . . . . .	44
7.2.2	Structure of the ISP1761 Host Controller memory . . . . .	17	8.3.5	HcDMAConfiguration register (R/W: 0330h) . .	45
7.3	Accessing the ISP1761 Host Controller memory: PIO and DMA . . . . .	19	8.3.6	HcBufferStatus register (R/W: 0334h) . . . . .	46
7.3.1	PIO mode access—memory read cycle . . . . .	20	8.3.7	ATL Done Timeout register (R/W: 0338h) . . .	47
7.3.2	PIO mode access—memory write cycle . . . . .	20	8.3.8	Memory register (R/W: 033Ch) . . . . .	47
7.3.3	PIO mode access—register read cycle . . . . .	21	8.3.9	Edge Interrupt Count register (R/W: 0340h) . .	48
7.3.4	PIO mode access—register write cycle . . . . .	21	8.3.10	DMA Start Address register (W: 0344h) . . . .	49
7.3.5	DMA—read and write operations . . . . .	21	8.3.11	Power Down Control register (R/W: 0354h) . .	50
7.4	Interrupts . . . . .	22	8.4	Interrupt registers . . . . .	52
7.5	Phase-Locked Loop (PLL) clock multiplier . . .	24	8.4.1	HcInterrupt register (R/W: 0310h) . . . . .	52
7.6	Power management . . . . .	25	8.4.2	HcInterruptEnable register (R/W: 0314h) . . .	54
7.7	Power supply . . . . .	26	8.4.3	ISO IRQ MASK OR register (R/W: 0318h) . . .	56
7.7.1	Hybrid mode . . . . .	28	8.4.4	INT IRQ MASK OR register (R/W: 031Ch) . . .	56
7.8	Overcurrent detection . . . . .	29	8.4.5	ATL IRQ MASK OR register (R/W: 0320h) . . .	56
7.9	Power-On Reset (POR) . . . . .	30	8.4.6	ISO IRQ MASK AND register (R/W: 0324h) . .	56
<b>8</b>	<b>Host Controller</b> . . . . .	<b>31</b>	8.4.7	INT IRQ MASK AND register (R/W: 0328h) . .	57
8.1	EHCI capability registers . . . . .	32	8.4.8	ATL IRQ MASK AND register (R/W: 032Ch) . .	57
8.1.1	CAPLENGTH register (R: 0000h) . . . . .	32	8.5	Philips Transfer Descriptor . . . . .	57
8.1.2	HCIVERSION register (R: 0002h) . . . . .	32	8.5.1	High-speed bulk IN and OUT, QHA . . . . .	60
8.1.3	HCSPARAMS register (R: 0004h) . . . . .	32	8.5.2	High-speed isochronous IN and OUT, iTD . . .	64
8.1.4	HCCPARAMS register (R: 0008h) . . . . .	33	8.5.3	High-speed interrupt IN and OUT, QHP . . . .	68
8.2	EHCI operational registers . . . . .	34	8.5.4	Start and complete split for bulk, QHA-SS/CS	72
8.2.1	USBCMD register (R/W: 0020h) . . . . .	34	8.5.5	Start and complete split for isochronous, SiTD	76
8.2.2	USBSTS register (R/W: 0024h) . . . . .	35	8.5.6	Start and complete split for interrupt . . . . .	80
8.2.3	USBINTR register (R/W: 0028h) . . . . .	36	<b>9</b>	<b>OTG Controller</b> . . . . .	<b>84</b>
8.2.4	FRINDEX register (R/W: 002Ch) . . . . .	36	9.1	Introduction . . . . .	84
8.2.5	CTRLDSSEGMENT register (R/W: 0030h) . . .	37	9.2	Dual-role device . . . . .	84
8.2.6	CONFIGFLAG register (R/W: 0060h) . . . . .	37	9.3	Session Request Protocol (SRP) . . . . .	85
8.2.7	PORTSC1 register (R, R/W, R/WC (field dependent): 0064h) . . . . .	38	9.3.1	B-device initiating SRP . . . . .	85
			9.3.2	A-device responding to SRP . . . . .	85
			9.4	Host Negotiation Protocol (HNP) . . . . .	86
			9.4.1	Sequence of HNP events . . . . .	86
			9.4.2	OTG state diagrams . . . . .	87
			9.4.3	HNP implementation and OTG state machine	89
			9.5	OTG Controller registers . . . . .	90

continued >>

9.5.1	Device Identification registers . . . . .	91	10.6.4	DMA Hardware register (R/W: 023Ch) . . . . .	117
9.5.1.1	Vendor ID register (R: 0370h) . . . . .	91	10.6.5	DMA Interrupt Reason register (R/W: 0250h) . . . . .	118
9.5.1.2	Product ID register (R: 0372h) . . . . .	91	10.6.6	DMA Interrupt Enable register (R/W: 0254h) . . . . .	119
9.5.2	OTG Control register . . . . .	92	10.6.7	DMA Endpoint register (R/W: 0258h) . . . . .	120
9.5.2.1	OTG Control register (S/C: 0374h/0376h) . . . . .	92	10.6.8	DMA Burst Counter register (R/W: 0264h) . . . . .	120
9.5.3	OTG Interrupt registers . . . . .	93	10.7	General registers . . . . .	121
9.5.3.1	OTG Status register (R: 0378h) . . . . .	93	10.7.1	DcInterrupt register (R/W: 0218h) . . . . .	121
9.5.3.2	OTG Interrupt Latch register (S/C: 037Ch/037Eh) . . . . .	94	10.7.2	DcChipID register (R: 0270h) . . . . .	123
9.5.3.3	OTG Interrupt Enable Fall register (S/C: 0380h/0382h) . . . . .	95	10.7.3	Frame Number register (R: 0274h) . . . . .	123
9.5.3.4	OTG Interrupt Enable Rise register (S/C: 0384h/0386h) . . . . .	95	10.7.4	DcScratch register (R/W: 0278h) . . . . .	124
9.5.4	OTG Timer register . . . . .	96	10.7.5	Unlock Device register (W: 027Ch) . . . . .	124
9.5.4.1	OTG Timer register (Low word S/C: 0388h/038Ah; high word S/C: 038Ch/038Eh) . . . . .	96	10.7.6	Interrupt Pulse Width register (R/W: 0280h) . . . . .	125
<b>10</b>	<b>Peripheral Controller . . . . .</b>	<b>98</b>	10.7.7	Test Mode register (R/W: 0284h) . . . . .	125
10.1	Introduction . . . . .	98	<b>11</b>	<b>Power consumption . . . . .</b>	<b>127</b>
10.1.1	Direct Memory Access (DMA) . . . . .	98	<b>12</b>	<b>Limiting values . . . . .</b>	<b>128</b>
10.1.1.1	DMA for the IN endpoint . . . . .	98	<b>13</b>	<b>Recommended operating conditions . . . . .</b>	<b>128</b>
10.1.1.2	DMA for the OUT endpoint . . . . .	98	<b>14</b>	<b>Static characteristics . . . . .</b>	<b>129</b>
10.1.1.3	DMA initialization . . . . .	98	<b>15</b>	<b>Dynamic characteristics . . . . .</b>	<b>132</b>
10.1.1.4	Starting DMA . . . . .	99	15.1	Host timing . . . . .	134
10.1.1.5	DMA stop and interrupt handling . . . . .	99	15.1.1	PIO timing . . . . .	134
10.2	Endpoint description . . . . .	100	15.1.1.1	Register or memory write . . . . .	134
10.3	Differences between the ISP1761 and ISP1582 Peripheral Controller . . . . .	101	15.1.1.2	Register read . . . . .	135
10.3.1	ISP1761 initialization registers . . . . .	101	15.1.1.3	Memory read . . . . .	136
10.3.2	ISP1761 DMA . . . . .	101	15.1.2	DMA timing . . . . .	137
10.3.3	ISP1761 peripheral suspend indication . . . . .	101	15.1.2.1	Single cycle: DMA read . . . . .	137
10.3.4	ISP1761 interrupt and DMA common mode . . . . .	101	15.1.2.2	Single cycle: DMA write . . . . .	138
10.4	Peripheral Controller-specific registers . . . . .	102	15.1.2.3	Multicycle: DMA read . . . . .	139
10.4.1	Address register (R/W: 0200h) . . . . .	102	15.1.2.4	Multicycle: DMA write . . . . .	140
10.4.2	Mode register (R/W: 020Ch) . . . . .	103	15.2	Peripheral timing . . . . .	141
10.4.3	Interrupt Configuration register (R/W: 0210h) . . . . .	104	15.2.1	PIO timing . . . . .	141
10.4.4	Debug register (R/W: 0212h) . . . . .	105	15.2.1.1	PIO register read or write . . . . .	141
10.4.5	DcInterruptEnable register (R/W: 0214h) . . . . .	106	15.2.2	DMA timing . . . . .	143
10.5	Data flow registers . . . . .	107	15.2.2.1	DMA read or write . . . . .	143
10.5.1	Endpoint Index register (R/W: 022Ch) . . . . .	107	<b>16</b>	<b>Package outline . . . . .</b>	<b>145</b>
10.5.2	Control Function register (R/W: 0228h) . . . . .	109	<b>17</b>	<b>Soldering . . . . .</b>	<b>147</b>
10.5.3	Data Port register (R/W: 0220h) . . . . .	109	17.1	Introduction to soldering surface mount packages . . . . .	147
10.5.4	Buffer Length register (R/W: 021Ch) . . . . .	110	17.2	Reflow soldering . . . . .	147
10.5.5	DcBufferStatus register (R/W: 021Eh) . . . . .	111	17.3	Wave soldering . . . . .	147
10.5.6	Endpoint MaxPacketSize register (R/W: 0204h) . . . . .	111	17.4	Manual soldering . . . . .	148
10.5.7	Endpoint Type register (R/W: 0208h) . . . . .	113	17.5	Package related soldering information . . . . .	148
10.6	DMA registers . . . . .	113	<b>18</b>	<b>Abbreviations . . . . .</b>	<b>149</b>
10.6.1	DMA Command register (W: 0230h) . . . . .	114	<b>19</b>	<b>References . . . . .</b>	<b>150</b>
10.6.2	DMA Transfer Counter register (R/W: 0234h) . . . . .	115	<b>20</b>	<b>Revision history . . . . .</b>	<b>150</b>
10.6.3	DcDMAConfiguration register (R/W: 0238h) . . . . .	116	<b>21</b>	<b>Data sheet status . . . . .</b>	<b>151</b>
			<b>22</b>	<b>Definitions . . . . .</b>	<b>151</b>
			<b>23</b>	<b>Disclaimers . . . . .</b>	<b>151</b>
			<b>24</b>	<b>Trademarks . . . . .</b>	<b>151</b>

continued &gt;&gt;

25 Contact information ..... 151



© Koninklijke Philips Electronics N.V. 2005

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Date of release: 12 January 2005  
Document number: 9397 750 13258

Published in The Netherlands