SAF1760

Hi-Speed Universal Serial Bus host controller for embedded applications

Rev. 2 — 19 June 2012 Product data sheet

1. General description

The SAF1760 is a Hi-Speed Universal Serial Bus (USB) host controller with a generic processor interface. It integrates one Enhanced Host Controller Interface (EHCI), one Transaction Translator (TT) and three transceivers. The host controller portion of the SAF1760 and the three transceivers comply to Ref. 1 “Universal Serial Bus Specification Rev. 2.0”. The EHCI portion of the SAF1760 is adapted from Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

The integrated high-performance Hi-Speed USB transceivers enable the SAF1760 to handle all Hi-Speed USB transfer speed modes: high-speed (480 Mbit/s), full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s). The three downstream ports allow simultaneous connection of three devices at different speeds (high-speed, full-speed and low-speed).

The generic processor interface allows the SAF1760 to be connected to various processors as a memory-mapped resource. The SAF1760 is a slave host: it does not require **bus-mastering** capabilities of the host system bus. The interface can be configured, ensuring compatibility with a variety of processors. Data transfer can be performed on 16 bits or 32 bits, using Programmed Input/Output (PIO) or Direct Memory Access (DMA) with major control signals configurable as active LOW or active HIGH.

Integration of the TT allows connection to full-speed and low-speed devices, without the need of integrating Open Host Controller Interface (OHCI) or Universal Host Controller Interface (UHCI). Instead of dealing with two sets of software drivers, EHCI and OHCI or UHCI, you need to deal with only one set, EHCI, that dramatically reduces software complexity and IC cost.

2. Features and benefits

- Automotive qualified in accordance with AEC-Q100
- The host controller portion of the SAF1760 complies with Ref. 1 “Universal Serial Bus Specification Rev. 2.0”
- The EHCI portion of the SAF1760 is adapted from Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”
- Contains three integrated Hi-Speed USB transceivers that support high-speed, full-speed and low-speed modes
- Integrates a TT for original USB (full-speed and low-speed) device support
- Up to 64 kB internal memory (8 k $\times$ 64 bit) accessible through a generic processor interface; operation in multitasking environments is made possible by the implementation of virtual segmentation mechanism with bank switching on task request

- Generic processor interface, non-multiplexed and variable latency, with a configurable 32-bit or 16-bit external data bus; the processor interface can be defined as variable-latency or SRAM type (memory mapping)
- Slave DMA support to reduce the load of the host system CPU during the data transfer to or from the memory
- Integrated Phase-Locked Loop (PLL) with a 12 MHz crystal or an external clock input
- Integrated multi-configuration FIFO
- Optimized **msec-based** or **multi-msec-based** Proprietary Transfer Descriptor (PTD) interrupt
- Tolerant I/O for low voltage CPU interface (1.65 V to 3.6 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 and low-voltage reset
- Supports suspend and remote wake-up
- Target current consumption:
  - Normal operation; one port in high-speed active: I_{CC} < 100 mA
  - Suspend mode: $I_{CC(susp)}$ < 150 µA at room temperature
- Built-in configurable overcurrent circuitry (digital or analog overcurrent protection)

3. Applications

The SAF1760 can be used to implement a Hi-Speed USB compliant host controller connected to most of the CPUs present in the market today, having a generic processor interface with de-multiplexed address and data bus. This is because of the efficient slave-type interface of the SAF1760.

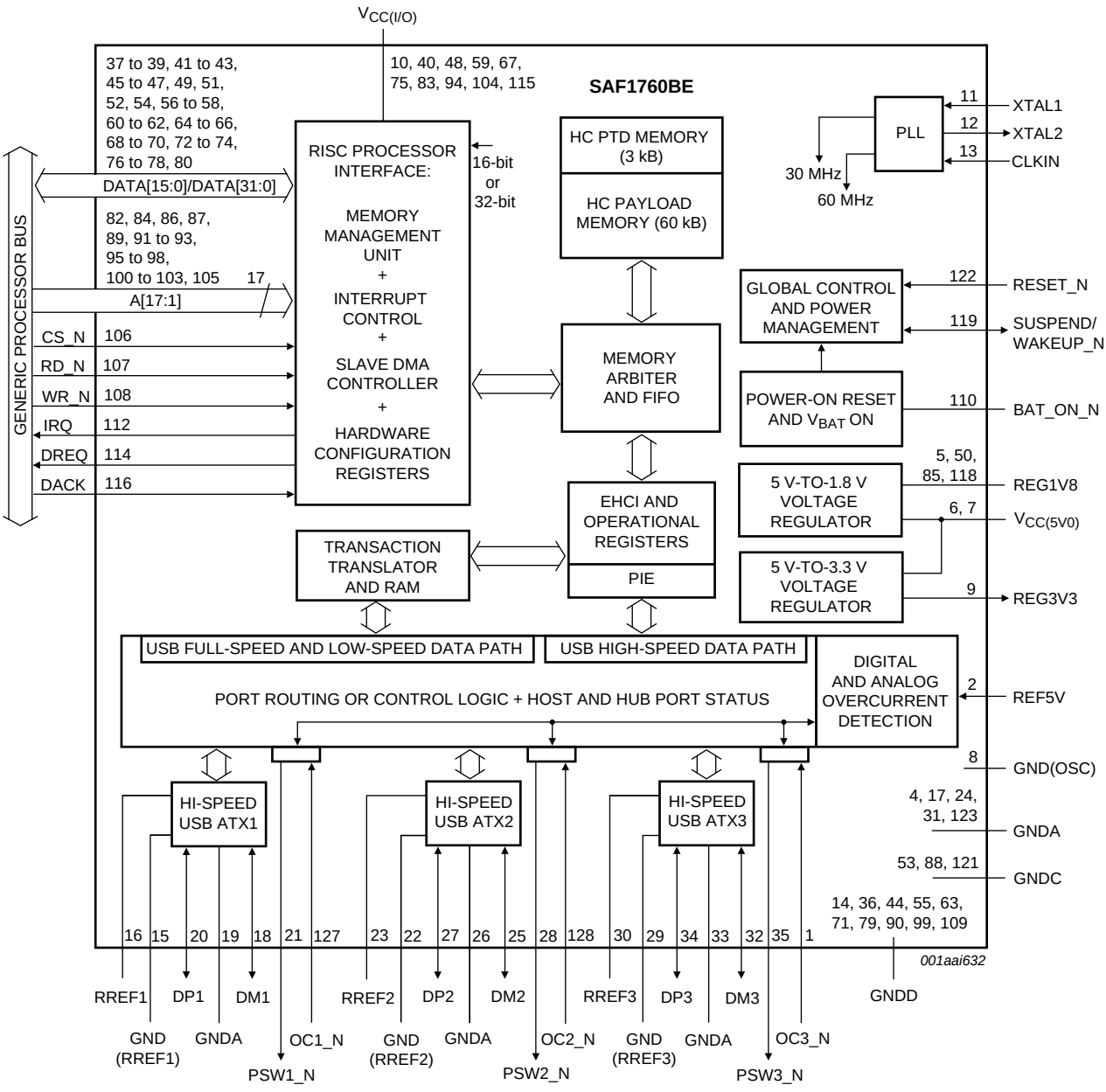
This NXP USB product can only be used in automotive applications. Inclusion or use of the NXP USB products in other than automotive applications is not permitted and for your company's own risk. Your company agrees to full indemnify NXP for any damages resulting from such inclusion or use.

4. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
SAF1760BE	LQFP128	plastic low profile quad flat package; 128 leads; body 14 × 20 × 1.4 mm	SOT425-1

5. Block diagram



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

Fig 1. Block diagram

6. Pinning information

6.1 Pinning

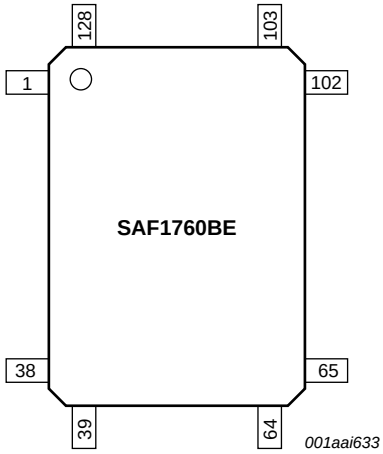


Fig 2. Pin configuration (LQFP128); top view

6.2 Pin description

Table 2. Pin description

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
OC3_N	1	AI	port 3 analog (5 V input) and digital overcurrent input; if not used, connect to $V_{CC(I/O)}$ through a 10 kΩ resistor input, 5 V tolerant
REF5V	2	AI	5 V reference input for analog OC detector; connect a 100 nF decoupling capacitor
TEST1	3	I	connect to ground
GNDA	4	G	analog ground
REG1V8	5	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor; for details on additional capacitor placement, see Section 7.8
$V_{CC(5V0)}$	6	P	input to internal regulators (3.0 V to 5.5 V); connect a 100 nF decoupling capacitor; see Section 7.8
$V_{CC(5V0)}$	7	P	input to internal regulators (3.0 V to 5.5 V); connect a 100 nF decoupling capacitor; see Section 7.8
GND(OSC)	8	G	oscillator ground
REG3V3	9	P	regulator output (3.3 V); for decoupling only; connect a 100 nF capacitor and a 4.7 µF-to-10 µF capacitor; see Section 7.8
$V_{CC(I/O)}$	10	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
XTAL1	11	AI	12 MHz crystal connection input; connect to ground if an external clock is used; see Table 89
XTAL2	12	AO	12 MHz crystal connection output

Table 2. Pin description *...continued*

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
CLKIN	13	I	12 MHz oscillator or clock input; when not in use, connect to $V_{CC(I/O)}$
GNDD	14	G	digital ground
GND(RREF1)	15	G	RREF1 ground
RREF1	16	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF1 ground
GNDA[4]	17	G	analog ground
DM1	18	AI/O	downstream data minus port 1
GNDA	19	G	analog ground
DP1	20	AI/O	downstream data plus port 1
PSW1_N	21	OD	power switch port 1, active LOW output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
GND(RREF2)	22	G	RREF2 ground
RREF2	23	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF2 ground
GNDA[5]	24	G	analog ground
DM2	25	AI/O	downstream data minus port 2
GNDA	26	G	analog ground
DP2	27	AI/O	downstream data plus port 2
PSW2_N	28	OD	power switch port 2, active LOW output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
GND(RREF3)	29	G	RREF3 ground
RREF3	30	AI	reference resistor connection; connect a 12 kΩ ± 1 % resistor between this pin and the RREF3 ground
GNDA[6]	31	G	analog ground
DM3	32	AI/O	downstream data minus port 3
GNDA	33	G	analog ground
DP3	34	AI/O	downstream data plus port 3
PSW3_N	35	OD	power switch port 3, active LOW output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
GNDD	36	G	digital ground
DATA0	37	I/O	data bit 0 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA1	38	I/O	data bit 1 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA2	39	I/O	data bit 2 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
$V_{CC(I/O)}$	40	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
DATA3	41	I/O	data bit 3 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA4	42	I/O	data bit 4 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA5	43	I/O	data bit 5 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	44	G	digital ground
DATA6	45	I/O	data bit 6 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA7	46	I/O	data bit 7 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA8	47	I/O	data bit 8 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	48	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
DATA9	49	I/O	data bit 9 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
REG1V8	50	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor; for details on additional capacitor placement, see Section 7.8
DATA10	51	I/O	data bit 10 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA11	52	I/O	data bit 11 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDC	53	G	core ground
DATA12	54	I/O	data bit 12 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	55	G	digital ground
DATA13	56	I/O	data bit 13 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
DATA14	57	I/O	data bit 14 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA15	58	I/O	data bit 15 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	59	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
DATA16	60	I/O	data bit 16 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA17	61	I/O	data bit 17 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA18	62	I/O	data bit 18 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	63	G	digital ground
DATA19	64	I/O	data bit 19 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA20	65	I/O	data bit 20 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA21	66	I/O	data bit 21 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	67	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
DATA22	68	I/O	data bit 22 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA23	69	I/O	data bit 23 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA24	70	I/O	data bit 24 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	71	G	digital ground
DATA25	72	I/O	data bit 25 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA26	73	I/O	data bit 26 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
DATA27	74	I/O	data bit 27 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
$V_{CC(I/O)}$	75	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
DATA28	76	I/O	data bit 28 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA29	77	I/O	data bit 29 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
DATA30	78	I/O	data bit 30 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
GNDD	79	G	digital ground
DATA31	80	I/O	data bit 31 input and output bidirectional pad, push-pull input, 3-state output, 4 mA output drive, 3.3 V tolerant
TEST2	81	G	connect to ground
A1	82	I	address pin 1 input, 3.3 V tolerant
$V_{CC(I/O)}$	83	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
A2	84	I	address pin 2 input, 3.3 V tolerant
REG1V8	85	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor and a 4.7 µF-to-10 µF capacitor; see Section 7.8
A3	86	I	address pin 3 input, 3.3 V tolerant
A4	87	I	address pin 4 input, 3.3 V tolerant
GNDC	88	G	core ground
A5	89	I	address pin 5 input, 3.3 V tolerant
GNDD	90	G	digital ground
A6	91	I	address pin 6 input, 3.3 V tolerant
A7	92	I	address pin 7 input, 3.3 V tolerant
A8	93	I	address pin 8 input, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
$V_{CC(I/O)}$	94	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
A9	95	I	address pin 9 input, 3.3 V tolerant
A10	96	I	address pin 10 input, 3.3 V tolerant
A11	97	I	address pin 11 input, 3.3 V tolerant
A12	98	I	address pin 12 input, 3.3 V tolerant
GNDD	99	G	digital ground
A13	100	I	address pin 13 input, 3.3 V tolerant
A14	101	I	address pin 14 input, 3.3 V tolerant
A15	102	I	address pin 15 input, 3.3 V tolerant
A16	103	I	address pin 16 input, 3.3 V tolerant
$V_{CC(I/O)}$	104	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
A17	105	I	address pin 17 input, 3.3 V tolerant
CS_N	106	I	chip select signal assertion indicates the SAF1760 being accessed; active LOW input, 3.3 V tolerant
RD_N	107	I	read enable; active LOW input, 3.3 V tolerant
WR_N	108	I	write enable; active LOW input, 3.3 V tolerant
GNDD	109	G	digital ground
BAT_ON_N	110	OD	to indicate the presence of a minimum 3.3 V on pins 6 and 7 (open-drain); connect to $V_{CC(I/O)}$ through a 10 kΩ pull-up resistor output pad, push-pull open-drain, 8 mA output drive, 5 V tolerant
n.c.	111	NC	not connected
IRQ	112	O	host controller interrupt signal output pad, 4 mA drive, 3.3 V tolerant
n.c.	113	NC	not connected
DREQ	114	O	DMA controller request for the host controller output pad, 4 mA drive, 3.3 V tolerant

Table 2. Pin description *...continued*

Symbol[1][2]	Pin	Type[3]	Description
	LQFP128		
$V_{CC(I/O)}$	115	P	digital supply voltage; 1.65 V to 3.6 V; connect a 100 nF decoupling capacitor; see Section 7.8
DACK	116	I	host controller DMA request acknowledgment; when not in use, connect to $V_{CC(I/O)}$ through a 10 kΩ pull-up resistor input, 3.3 V tolerant
TEST3	117	I	connect to $V_{CC(I/O)}$ through a 10 kΩ pull-up resistor
REG1V8	118	P	core power output (1.8 V); internal 1.8 V for the digital core; used for decoupling; connect a 100 nF capacitor; for details on additional capacitor placement, see Section 7.8
SUSPEND/ WAKEUP_N	119	I/OD	host controller suspend and wake-up; 3-state suspend output (active LOW) and wake-up input circuits are connected together • HIGH = output is 3-state; SAF1760 is in suspend mode • LOW = output is LOW; SAF1760 is not in suspend mode connect to $V_{CC(I/O)}$ through an external 10 kΩ pull-up resistor output pad, open-drain, 4 mA output drive, 3.3 V tolerant
TEST4	120	I	pull up to $V_{CC(I/O)}$
GNDC	121	G	core ground
RESET_N	122	I	external power-up reset; active LOW; when reset is asserted, it is expected that bus signals are idle, that is, not toggling input, 3.3 V tolerant **Remark:** During reset, ensure that all the input pins to the SAF1760 are not toggling and are in their inactive states.
GNDA	123	G	analog ground
TEST5	124	AI/O	connect a 220 nF capacitor between this pin and pin 125
TEST6	125	AI/O	connect a 220 nF capacitor between this pin and pin 124
TEST7	126	I	connect to 3.3 V
OC1_N	127	AI	port 1 analog (5 V input) and digital overcurrent input; if not used, connect to $V_{CC(I/O)}$ through a 10 kΩ resistor input, 5 V tolerant
OC2_N	128	AI	port 2 analog (5 V input) and digital overcurrent input; if not used, connect to $V_{CC(I/O)}$ through a 10 kΩ resistor input, 5 V tolerant

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

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

[3] I = input only; O = output only; I/O = digital input/output; OD = open-drain output; AI/O = analog input/output; AI = analog input; P = power; G = ground supply; NC = not connected.

[4] For port 1.

[5] For port 2.

[6] For port 3.

7. Functional description

7.1 SAF1760 internal architecture: advanced NXP slave host controller and hub

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

The EHCI is the latest generation design, with improved data bandwidth. The EHCI in the SAF1760 is adapted from Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.

The internal Hi-Speed USB hub block replaces the companion host controller block used in the original architecture of a PCI Hi-Speed USB host controllers to handle full-speed and low-speed modes. The hardware architecture in the SAF1760 is simplified to help reduce cost and development time, by eliminating the additional work involved in implementing the OHCI software required to support full-speed and low-speed modes.

Figure 3 shows the internal architecture of the SAF1760. The SAF1760 implements the 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 (USB 2.0) 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.

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.

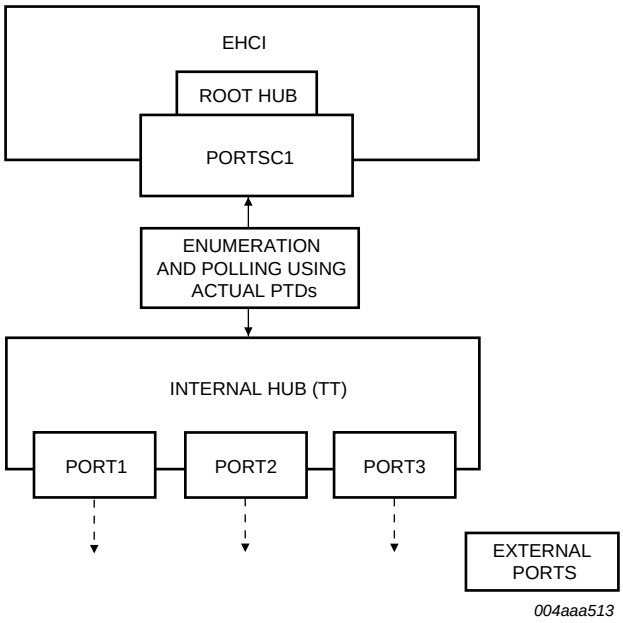


Fig 3. Internal hub

7.1.1 Internal clock scheme and port selection

The SAF1760 has three ports. Figure 4 shows the internal clock scheme of the SAF1760.

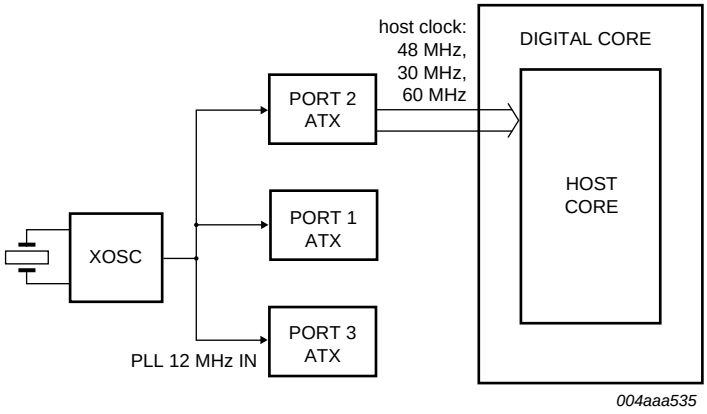


Fig 4. SAF1760 clock scheme

Figure 4 shows that the host clock is derived from port 2. Port 2 does not need to be enabled by software, if only port 1 or port 3 is used. No port needs to be disabled by external pull-up resistors, if not used. The DP and DM of the unused ports need not be externally pulled HIGH because there are internal pull-down resistors on each port that are enabled by default.

Table 3 lists the various port connection scenarios.

Table 3. Port connection scenarios

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

7.2 Host controller buffer memory block

7.2.1 General considerations

The internal addressable host controller buffer memory is 63 kB. The 63 kB effective memory size is the result of subtracting the size of the registers (1 kB) from the total addressable memory space defined in the SAF1760 (64 kB). This is the optimized value to achieve the highest performance with minimal cost.

The SAF1760 is a slave host controller. This means that it does not need access to the local bus of the system to transfer data from the system memory to the SAF1760 internal memory, unlike the case of the original PCI Hi-Speed USB host controllers. Therefore, correct data must be transferred to both the PTD area and the payload area by PIO (using CPU access) or programmed DMA.

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

The considerations mentioned are also the main reason for implementing the pre-fetching technique, instead of using a READY signal. The resulting architecture avoids **freezing** of the local bus, by asserting READY, enhancing the SAF1760 memory access time, and avoiding introduction of programmed additional wait states. For details, see Section 7.3 and Section 8.3.8.

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

A larger buffer also implies a larger amount of data can be transferred. The 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 for up to a few milliseconds before requiring further CPU intervention for data movement.

The internal architecture of the SAF1760 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 various data transfer schemes, depending on the number and type of USB devices present. For example: push-pull; data can be written to half of the memory while data in the other half is being accessed by the host controller and sent on the USB bus. This is useful especially when a high-bandwidth **continuous or periodic** data flow is required.

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

7.2.2 Structure of the SAF1760 host controller memory

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

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

A maximum of 32 PTD areas and their allocated payload areas can be defined for each type of transfer. The structure of a PTD is similar for every transfer type and consists of eight Double Words (DWs) that must be correctly programmed for a correct USB data transfer. The reserved bits of a PTD must be set to logic 0. A detailed description of the PTD structure can be found in Section 9.

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 SAF1760 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

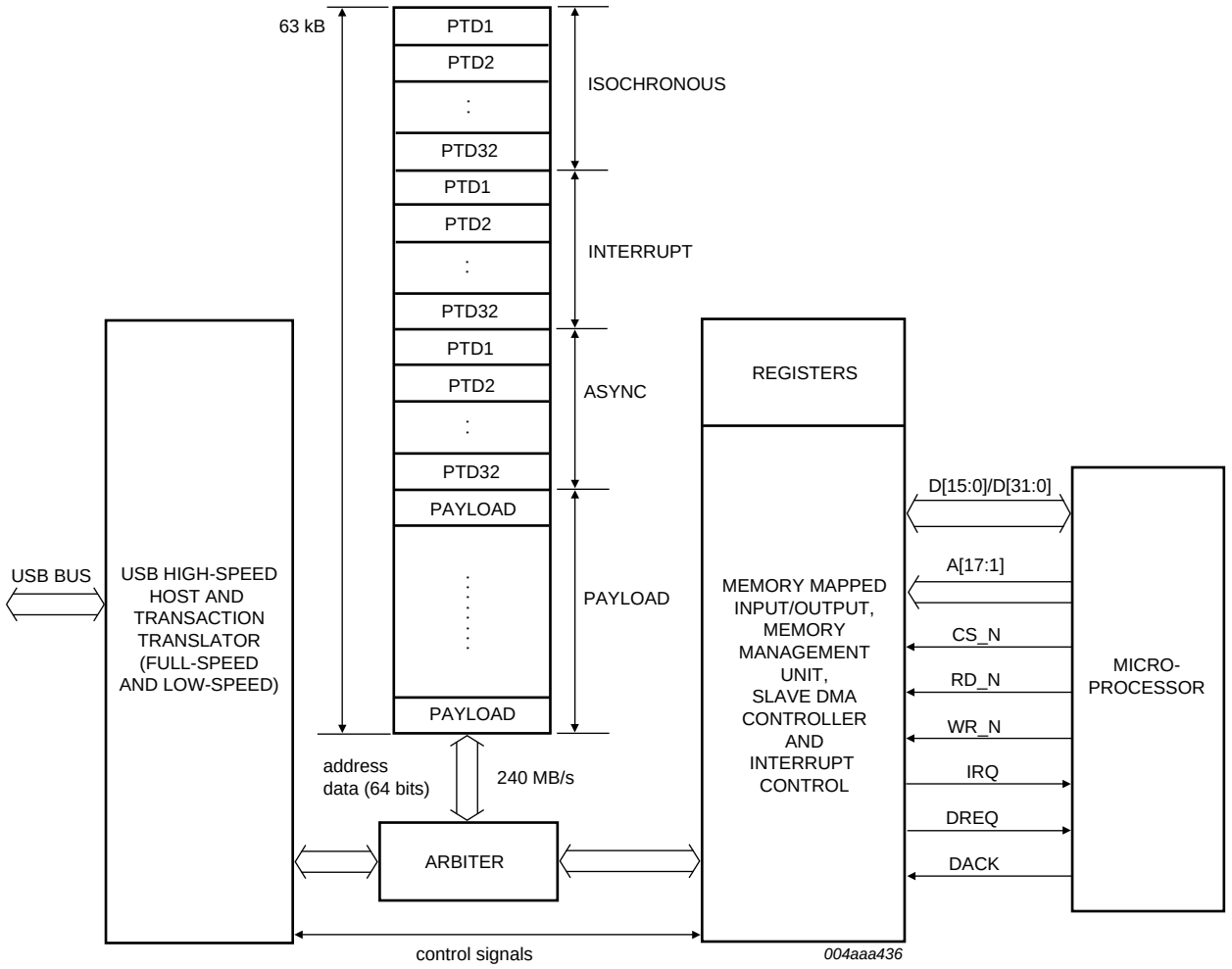


Fig 5. Memory segmentation and access block diagram

Both the CPU interface logic and the USB host controller require access to the internal SAF1760 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 SAF1760 host controller memory: PIO and DMA

The CPU interface of the SAF1760 can be configured for a 16-bit or 32-bit data bus width.

When the SAF1760 is configured for a 16-bit data bus width, the upper unused 16 data lines must be pulled up to $V_{CC(I/O)}$. This can be achieved by connecting DATA[31:16] lines together to a single 10 kΩ pull-up resistor. The 16-bit or 32-bit data bus width configuration is done by programming bit 8 of the HW Mode Control register. This will determine the register and memory access types in both PIO and DMA modes. 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 SAF1760 DMA is enabled.

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 pre-fetch data for the memory read of that bank.

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

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

  For example, consider the following sequence of operations:

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

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

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

  Consequently, the RD_N assertion with A[17:16] = 01b will return data from 400Ch because the bank1 read stopped there in the previous cycle. Also, RD_N assertions with A[17:16] = 10b 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 writes access is similar to a normal memory access. It is not necessary to set the pre-fetching address before a write cycle to the memory.

The SAF1760 internal write address will not be automatically incremented during consecutive write accesses; unlike in a series of SAF1760 memory read cycles. The memory write address must be incremented before every access.

7.3.3 PIO mode access, register read cycle

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

The SAF1760 register read address will not be automatically incremented during consecutive read accesses; unlike in a series of SAF1760 memory read cycles. The SAF1760 register read address must be correctly specified before every access.

7.3.4 PIO mode access, register write cycle

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

The SAF1760 register write address will not be automatically incremented during consecutive write accesses; unlike in a series of SAF1760 memory read cycles. The SAF1760 register write address must be correctly specified before every access.

7.3.5 DMA mode, read and write operations

The internal SAF1760 host controller DMA is a slave DMA. The host system processor or DMA must ensure the data transfer to or from the SAF1760 memory.

The SAF1760 DMA supports a DMA burst length of 1, 4, 8 and 16 cycles for both the 16-bit and 32-bit data bus width. DREQ will be asserted at the beginning of the first burst of a DMA transfer and will be de-asserted on the last cycle, RD_N or WR_N active pulse, of that burst. It will be reasserted shortly after the DACK de-assertion, as long as the DMA transfer counter was not reached. DREQ will be de-asserted on the last cycle when the DMA transfer counter is reached and will not be reasserted until the DMA reprogramming is performed. Both 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 SAF1760 memory address. A register or memory access or access to other system memory can occur in between DMA bursts, whenever the bus is released because DACK is de-asserted, without affecting the DMA transfer counter or the current address.

Any memory area can be accessed by the systems 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 SAF1760 does not implement EOT to signal the end of a DMA transfer. If programmed, an interrupt may be generated by the SAF1760 at the end of the DMA transfer.

The slave DMA of the SAF1760 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 SAF1760 memory is available for the data transfer and the SAF1760 DMA programming is completed.

It is also possible that the systems DMA will perform a memory-to-memory type of transfer between the system memory and the SAF1760 memory. The SAF1760 will be accessed in PIO mode. Consequently, memory read operations must be preceded by initializing the Memory register (address 033Ch), as described in Section 7.3.1. No IRQ will be generated by the SAF1760 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 SAF1760 DMA programming involves:

- Set the active levels of signals DREQ and DACK in the HW Mode Control register.
- The DMA Start Address register contains the first memory address at which the data transfer will start. It must be word-aligned in 16-bit data bus mode and double word aligned in 32-bit data bus mode.
- The programming of the DMA Configuration register specifies:
  - The type of transfer that will be performed: read or write.
  - The burst size, expressed in bytes, is specified, regardless of the data bus width. For the same burst size, a double number of cycles will be generated in 16-bit mode data bus width as compared to 32-bit mode.
  - The transfer length, expressed in number of bytes, defines the number of bursts. The DREQ will be de-asserted and asserted to generate the next burst, as long as there are bytes to be transferred. At the end of a transfer, the DREQ will be de-asserted and an IRQ can be generated if DMAEOTINT (bit 3 in the Interrupt 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 systems 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 SAF1760 memory will be affected.

– Enable ENABLE_DMA (bit 1) of the DMA Configuration register to determine the assertion of DREQ immediately after setting the bit.

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

The programming of the systems DMA must match the programming of the SAF1760 DMA parameters. Only one DMA transfer may take place at a time. PIO mode data transfer may occur simultaneously with a DMA data transfer, in the same or a different memory area.

7.4 Interrupts

The SAF1760 will assert an IRQ according to the source or event in the Interrupt 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 Interrupt Enable register. The software will need to clear the interrupt status bits in the Interrupt register before enabling individual interrupt enable bits.

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

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

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

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

The asserted bits in the Interrupt register can be cleared by writing back the same value to the Interrupt register. This means that writing logic 1 to each of the set bits will reset the 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 Interrupt 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 Interrupt register by writing logic 1 to clear the Interrupt register bits that are set.
- If an interrupt is made edge-sensitive and is asserted, writing to clear the Interrupt 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 suspend and resume cycles, when an interrupt is generated to signal a wake-up event.

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

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

The registers definition shows that the AND or OR conditions are applicable to the same category of PTDs: ISO, INT, 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 the done map is set and the corresponding bit n of the OR Mask register is set.
- If the OR mask for any done bit is not set, then the AND mask comes into picture. An IRQ is generated if all the corresponding done bits of the AND Mask register are set. For example: If bits 2, 4 and 10 are set in the AND Mask register, an IRQ is generated only if bits 2, 4, 10 of the done map are set.
- If using the IRQ interval setting for the bulk PTD, an interrupt will only occur at the regular time interval as programmed in the ATL Done Timeout register. Even if an interrupt event occurs before the time-out of the register, no IRQ will be generated until the time is up.

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

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

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

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

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

7.5 Phase-Locked Loop (PLL) clock multiplier

The internal PLL requires a 12 MHz input, which can be a 12 MHz crystal or a 12 MHz clock already existing in the system with a precision better than 50×10^{-6}. This allows the use of a low-cost 12 MHz crystal that also minimizes ElectroMagnetic Interference (EMI). When an external crystal is used, make sure the CLKIN pin is connected to $V_{CC(I/O)}$.

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

No external components are required for the PLL operation.

7.6 Power management

The SAF1760 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.

When the SAF1760 is in suspend mode, the main internal clocks will be stopped to ensure minimum power consumption. An internal LazyClock of 100 kHz ± 40 % will continue running. This allows initiating a resume on one of these events:

- External USB device connect or disconnect
- CS_N signal asserted when the SAF1760 is accessed
- Driving the SUSPEND/WAKEUP_N pin to a LOW level

The SUSPEND/WAKEUP_N pin is a bidirectional pin. This pin must be connected to the GPIO pins of a processor.

The wake up state can be verified by reading the LOW level of this pin. If the level is HIGH, it means that the SAF1760 is in the suspend state.

The SUSPEND/WAKEUP_N pin requires a pull-up because in the SAF1760 suspended state the pin becomes 3-state and can be pulled down, driving it externally by switching the processors GPIO line to output mode to generate the SAF1760 wake-up.

The SUSPEND/WAKEUP_N pin is a 3-state output. It is also an input to the internal wake-up logic.

When in suspend mode, the SAF1760 internal wake-up circuitry will sense the status of the SUSPEND/WAKEUP_N pin:

- If it remains pulled-up, no wake-up is generated because a HIGH is sensed by the internal wake-up circuit.
- If the pin is externally pulled LOW, for example, by the GPIO line or just as a test by jumper, 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 SAF1760 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.

The Power-Down Control register allows additionally the SAF1760 internal blocks to be disabled for lower power consumption as defined in Table 51.

A very low suspend current can be achieved by completely switching off the $V_{CC(5V0)}$ using an external PMOS transistor, controlled by one of the GPIO pins of the processor.

When the SAF1760 power is always on, the time from wake-up to suspend will be approximately 100 ms.

It is necessary to wait for the CLKREADY interrupt assertion before programming the SAF1760 because internal clocks are stopped during deep-sleep suspend and restarted after the first wake-up event. The occurrence of the CLKREADY interrupt means that internal clocks are running and the normal functionality is achieved.

It is estimated that the CLKREADY interrupt will be generated less than 100 µs after the wake-up event, if the power to the SAF1760 was on during suspend.

If the SAF1760 is used in hybrid mode and $V_{CC(5V0)}$ is off during suspend, a 3 ms reset pulse is required when the power is switched back on, before the resume programming sequence starts. This will ensure that internal clocks are running and all logics reach a stable initial state.

7.7 Overcurrent detection

The SAF1760 can implement a digital or analog overcurrent detection scheme. Bit 15 of the HW Mode Control register can be programmed to select the analog or digital overcurrent detection. An analog overcurrent detection circuit is integrated on-chip. The main features of this circuit are self reporting, automatic resetting, low-trip time and low cost. This circuit offers an easy solution at no extra hardware cost on the board. The port power will automatically be disabled by the SAF1760 on an overcurrent event occurrence, by de-asserting the PSWn_N signal without any software intervention.

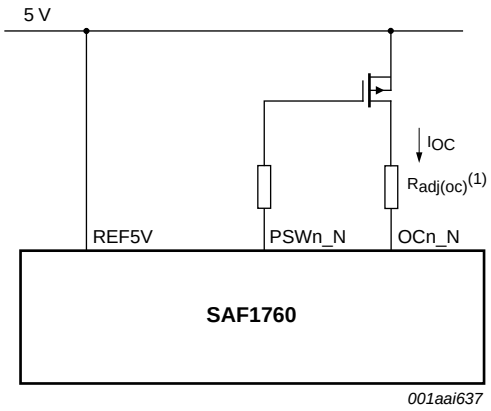
When using the integrated analog overcurrent detection, the range of the overcurrent detection voltage for the SAF1760 is 45 mV to 120 mV. Calculation of external components should be based on the 45 mV value, with the actual overcurrent detection threshold usually positioned in the middle of the interval.

For an overcurrent limit of 500 mA per port, a PMOS transistor with R_{DSon} of approximately 100 mΩ is required. If a PMOS transistor with a lower R_{DSon} is used, the analog overcurrent detection can be adjusted using a series resistor; see Figure 6.

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

ΔV_{PMOS} = voltage drop on PMOS

$I_{OC(nom)}$ = 1 µA



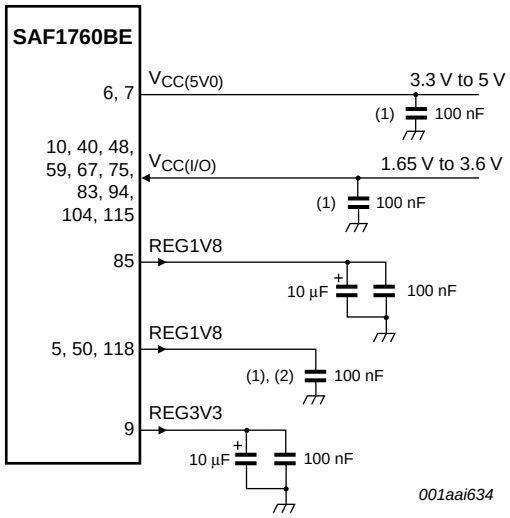
(1) $R_{adj(oc)}$ is optional.

Fig 6. Adjusting analog overcurrent detection limit (optional)

The digital overcurrent scheme requires using an external power switch with integrated overcurrent detection, such as LM3526, MIC2526 (2 ports) or LM3544 (4 ports). These devices are controlled by PSWn_N signals corresponding to each port. In the case of overcurrent occurrence, these devices will assert OCn_N signals. On OCn_N assertion, the SAF1760 cuts off the port power by de-asserting PSWn_N. The external integrated power switch will also automatically cut off the port power in the case of an overcurrent event, by implementing a thermal shutdown. An internal delay filter will prevent false overcurrent reporting because of in-rush currents when plugging a USB device. Because of this internal delay, as soon as OCn_N is asserted, PSWn_N will switch off the external PMOS in less than 15 ms.

7.8 Power supply

Figure 7 shows the SAF1760 power supply connection.

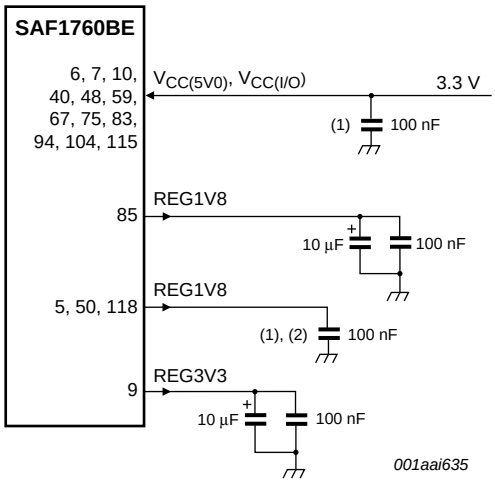


(1) Each supply voltage pin must be connected to a 100 nF decoupling capacitor

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

Fig 7. SAF1760 power supply connection

Figure 8 shows the most commonly used power supply connection.



(1) Each supply voltage pin must be connected to a 100 nF decoupling capacitor

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

Fig 8. Most commonly used power supply connection

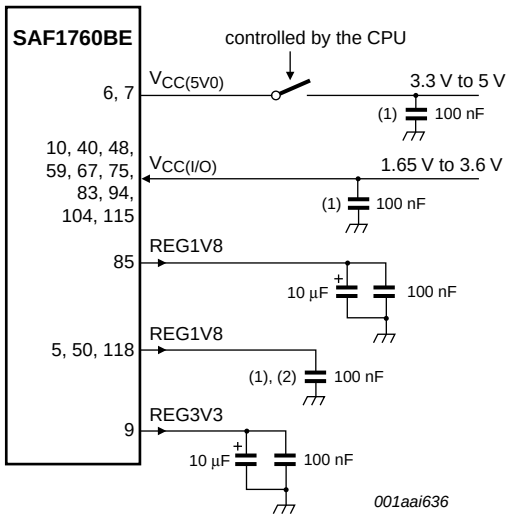
7.8.1 Hybrid mode

Table 6 shows the description of hybrid mode.

Table 6. Hybrid mode

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

In hybrid mode (see Figure 9), $V_{CC(5V0)}$ can be switched off using an external PMOS transistor, controlled using one of the GPIO pins of the processor. This helps to reduce the suspend current, $I_{CC(I/O)}$, below 100 µA. If the SAF1760 is used in hybrid mode and $V_{CC(5V0)}$ is off during suspend, a 3 ms reset pulse is required when power is switched back on, before the resume programming sequence starts.



(1) Each supply voltage pin must be connected to a 100 nF decoupling capacitor

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

Fig 9. Hybrid mode

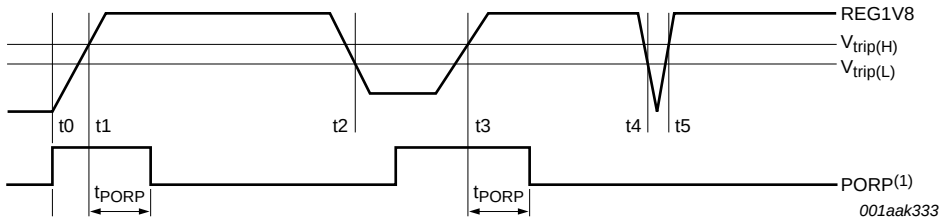
Table 7 shows the status of output pins in hybrid mode.

Table 7. Pin status in hybrid mode

Pins	$V_{CC(I/O)}$	$V_{CC(5V0)}$	Status
DATA[31:0], A[17:1], TEST1, TEST2, TEST3, TEST4, TEST5, TEST6, TEST7, DREQ, DACK, IRQ, SUSPEND/WAKEUP_N	on	on	normal
	on	off	high-Z
	off	X	undefined
CS_N, RESET_N, RD_N, WR_N	on	X	input
	off	X	undefined

7.9 Power-On Reset (POR)

Figure 10 shows a possible curve of REG1V8 with dips at t2 to t3 and t4 to t5. The PORP starts with a HIGH at t0. At t1, the detector will see the passing of the trip level $V_{trip(H)}$ and a delay element will add another t_{PORP} before the PORP drops to 0. If the dip at t4 to t5 is too short, less than 11 µs, the PORP will not react and will remain LOW. A HIGH on PORP will be generated whenever REG1V8 drops below $V_{trip(L)}$ for more than 11 µs.

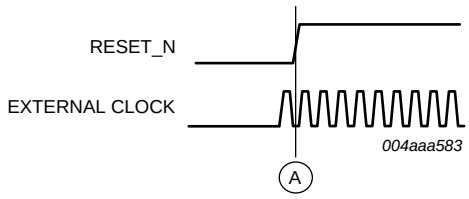


(1) PORP = Power-On Reset Pulse.

Fig 10. Internal power-on reset timing

The recommended RESET input pulse length at power-on must be at least 3 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 11 shows the availability of the clock with respect to the external POR.



Stable external clock is available at A.

Fig 11. Clock with respect to the external power-on reset

8. Registers

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, which is 32-bit data. In case of a 16-bit data bus width, two subsequent accesses are necessary to complete the register read or write cycle.
- Operational registers range from 0000h to 01FFh. Configuration registers range from 0300h to 03FFh.

Table 8. Register overview

Address	Register	Reset value	References
EHCI capability registers			
0000h	CAPLENGTH	20h	Section 8.1.1
0002h	HCIVERSION	0100h	Section 8.1.2
0004h	HCSPARAMS	0000 0011h	Section 8.1.3
0008h	HCCPARAMS	0000 0086h	Section 8.1.4
EHCI operational registers			
0020h	USBCMD	0008 0B00h	Section 8.2.1
0024h	USBSTS	0000 0000h	Section 8.2.2
0028h	USBINTR	0000 0000h	Section 8.2.3
002Ch	FRINDEX	0000 0000h	Section 8.2.4
0060h	CONFIGFLAG	0000 0000h	Section 8.2.5
0064h	PORTSC1	0000 2000h	Section 8.2.6
0130h	ISO PTD Done Map	0000 0000h	Section 8.2.7
0134h	ISO PTD Skip Map	FFFF FFFFh	Section 8.2.8
0138h	ISO PTD Last PTD	0000 0000h	Section 8.2.9
0140h	INT PTD Done Map	0000 0000h	Section 8.2.10
0144h	INT PTD Skip Map	FFFF FFFFh	Section 8.2.11
0148h	INT PTD Last PTD	0000 0000h	Section 8.2.12
0150h	ATL PTD Done Map	0000 0000h	Section 8.2.13
0154h	ATL PTD Skip Map	FFFF FFFFh	Section 8.2.14
0158h	ATL PTD Last PTD	0000 0000h	Section 8.2.15
0200h to 02FFh	reserved	-	-
Configuration registers			
0300h	HW Mode Control	0000 0100h	Section 8.3.1
0304h	Chip ID	0001 1761h	Section 8.3.2
0308h	Scratch	0000 0000h	Section 8.3.3
030Ch	SW Reset	0000 0000h	Section 8.3.4
0330h	DMA Configuration	0000 0000h	Section 8.3.5
0334h	Buffer Status	0000 0000h	Section 8.3.6
0338h	ATL Done Timeout	0000 0000h	Section 8.3.7
033Ch	Memory	0000 0000h	Section 8.3.8
0340h	Edge Interrupt Count	0000 000Fh	Section 8.3.9

Table 8. Register overview …continued

Address	Register	Reset value	References
0344h	DMA Start Address	0000 0000h	Section 8.3.10
0354h	Power-Down Control	03E8 1BA0h	Section 8.3.11
0374h	Port 1 Control	0086 0086h	Section 8.3.12
Interrupt registers			
0310h	Interrupt	0000 0000h	Section 8.4.1
0314h	Interrupt Enable	0000 0000h	Section 8.4.2
0318h	ISO IRQ Mask OR	0000 0000h	Section 8.4.3
031Ch	INT IRQ Mask OR	0000 0000h	Section 8.4.4
0320h	ATL IRQ Mask OR	0000 0000h	Section 8.4.5
0324h	ISO IRQ Mask AND	0000 0000h	Section 8.4.6
0328h	INT IRQ Mask AND	0000 0000h	Section 8.4.7
032Ch	ATL IRQ Mask AND	0000 0000h	Section 8.4.8

8.1 EHCI capability registers

8.1.1 CAPLENGTH register

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

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

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

8.1.2 HCIVERSION register

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

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

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

8.1.3 HCSPARAMS register

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

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

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

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

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

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

8.1.4 HCCPARAMS register

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

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

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

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

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

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

8.2 EHCI operational registers

8.2.1 USBCMD register

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

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

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

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

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

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

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

8.2.2 USBSTS register

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

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

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

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

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

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

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

8.2.3 USBINTR register

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

8.2.4 FRINDEX register

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

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

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

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

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

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

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

8.2.5 CONFIGFLAG register

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

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

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

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

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

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

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

8.2.6 PORTSC1 register

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

- No peripheral connected
- Port disabled

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

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

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

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

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

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

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

Bit	Symbol	Description[1]
2	PED	**Port Enabled/Disabled**: Logic 1 means enable. Logic 0 means disable.[2]
1	ECSC	**Connect Status Change**: Logic 1 means change in ECCS. Logic 0 means no change.[2]
0	ECCS	**Current Connect Status**: Logic 1 indicates a device is present on the port. Logic 0 indicates no device is present.[2]

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

[2] These fields read logic 0, if the PP (Port Power) bit in register PORTSC1 is logic 0.

8.2.7 ISO PTD Done Map register

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

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

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

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

8.2.8 ISO PTD Skip Map register

Table 26 shows the bit description of the register.

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

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

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

8.2.9 ISO PTD Last PTD register

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

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

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

Once the LastPTD bit corresponding to a PTD is set, this will be the last PTD processed (checking V = logic 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.

8.2.10 INT PTD Done Map register

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

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

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

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

8.2.11 INT PTD Skip Map register

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

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

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

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

8.2.12 INT PTD Last PTD register

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

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

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

Once the LastPTD bit corresponding to a PTD is set, this will be the last PTD processed (checking V = logic 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.

8.2.13 ATL PTD Done Map register

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

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

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

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

8.2.14 ATL PTD Skip Map register

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

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

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

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

8.2.15 ATL PTD Last PTD register

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

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

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

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

8.3 Configuration registers

8.3.1 HW Mode Control register

Table 34 shows the bit allocation of the register.

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

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

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

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

Bit	Symbol	Description
31	ALL_ATX_RESET	**All ATX Reset**: For debugging purposes (not used normally). **1 —** Enable reset, then write back logic 0 **0 —** No reset
30 to 16	-	reserved; write logic 0
15	ANA_DIGI_OC	**Analog Digital Overcurrent**: This bit selects analog or digital overcurrent detection on pins OC1_N, OC2_N and OC3_N. **0 —** Digital overcurrent **1 —** Analog overcurrent
14 to 9	-	reserved; write logic 0
8	DATA_BUS_WIDTH	**Data Bus Width**: **0 —** Defines a 16-bit data bus width **1 —** Sets a 32-bit data bus width
7	-	reserved; write logic 0
6	DACK_POL	**DACK Polarity**: **1 —** Indicates that the DACK input is active HIGH **0 —** Indicates active LOW

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

Bit	Symbol	Description
5	DREQ_POL	**DREQ Polarity**: **1 —** Indicates that the DREQ output is active HIGH **0 —** Indicates active LOW
4 to 3	-	reserved; write logic 0
2	INTR_POL	**Interrupt Polarity**: **0 —** Active LOW **1 —** Active HIGH
1	INTR_LEVEL	**Interrupt Level**: **0 —** INT is level triggered. **1 —** INT is edge triggered. A pulse of certain width is generated.
0	GLOBAL_INTR_EN	**Global Interrupt Enable**: This bit must be set to logic 1 to enable the IRQ signal assertion. **0 —** IRQ assertion is disabled. IRQ will never be asserted, regardless of other settings or IRQ events. **1 —** IRQ assertion is enabled. IRQ will be asserted according to the Interrupt Enable register, and events setting and occurrence.

8.3.2 Chip ID register

Read this register to get the ID of the SAF1760. The 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. Chip ID - Chip Identifier register (address 0304h) bit description

Bit	Symbol	Access	Value	Description
31 to 0	CHIPID[31:0]	R	0001 1761h	**Chip ID**: This register represents the hardware version number (0001h) and the chip ID (1761h). **Remark:** The chip ID is for internal use to identify the SAF176x product family.

8.3.3 Scratch register

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

Table 37. Scratch register (address 0308h) bit description

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

8.3.4 SW Reset register

Table 38 shows the bit allocation of the register.

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

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

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

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

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

8.3.5 DMA Configuration register

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

Table 40. DMA Configuration register (address 0330h) bit allocation

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

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

Table 41. DMA Configuration register (address 0330h) bit description

Bit	Symbol	Description
31 to 8	DMA_COUNTER[23:0]	**DMA Counter**: The number of bytes to be transferred (read or write). **Remark:** Different number of bursts will be generated for the same transfer length programmed in 16-bit and 32-bit modes because DMA_COUNTER is in number of bytes.
7 to 4	-	reserved
3 to 2	BURST_LEN[1:0]	**DMA Burst Length**: **00 —** Single DMA burst **01 —** 4-cycle DMA burst **10 —** 8-cycle DMA burst **11 —** 16-cycle DMA burst
1	ENABLE_DMA	**Enable DMA**: **0 —** Terminate DMA **1 —** Enable DMA
0	DMA_READ_WRITE_SEL	**DMA Read/Write Select**: Indicates if the DMA operation is a write or read to or from the SAF1760. **0 —** DMA write to the SAF1760 internal RAM is set **1 —** DMA read from the SAF1760 internal RAM

8.3.6 Buffer Status register

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

Remark: Software can set these bits during the initialization.

Table 42 shows the bit allocation of the Buffer Status register.

Table 42. Buffer Status register (address 0334h) bit allocation

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

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

Table 43. Buffer Status register (address 0334h) bit description

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

8.3.7 ATL Done Timeout register

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

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

Bit	Symbol	Access	Value	Description
31 to 0	ATL_DONE_ TIMEOUT[31:0]	R/W	0000 0000h	**ATL Done Timeout**: This register determines the ATL done time-out interrupt. This register defines the time-out in milliseconds after which the SAF1760 asserts the INT line, if enabled. It is applicable to ATL done PTDs only.

8.3.8 Memory register

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

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

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

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

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

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

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

8.3.9 Edge Interrupt Count register

Table 47 shows the bit allocation of the register.

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

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

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

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

Bit	Symbol	Description
31 to 24	MIN_WIDTH [7:0]	**Minimum Width**: Indicates the minimum width between two edge interrupts in µSOFs (1 µSOF = 125 µs). This is not valid for level interrupts. A count of zero means that interrupts occur as and when an event occurs.
23 to 16	-	reserved
15 to 0	NO_OF_CLK [15:0]	**Number of Clocks**: Count in number of clocks that the edge interrupt must be kept asserted on the interface. The default value is 000Fh. Thus, 15 cycles of 30 MHz clock will make the default IRQ pulse width approximately 500 ns.

8.3.10 DMA Start Address register

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

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

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

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

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

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

8.3.11 Power-Down Control register

This register is used to turn off power to the internal blocks of the SAF1760 to obtain maximum power savings. Table 51 shows the bit allocation of the register.

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

Bit	31	30	29	28	27	26	25	24
Symbol	CLK_OFF_COUNTER[15:8]							
Reset	0	0	0	0	0	0	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Symbol	CLK_OFF_COUNTER[7:0]							
Reset	1	1	1	0	1	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]			PORT3_PD	PORT2_PD	VBATDET_PWR	reserved[1]	
Reset	0	0	0	1	1	0	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Symbol	reserved[1]		BIASEN	VREG_ON	OC3_PWR	OC2_PWR	OC1_PWR	HC_CLK_EN
Reset	1	0	1	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

Bit[1]	Symbol	Description
31 to 16	CLK_OFF_COUNTER [15:0]	**Clock Off Counter**: Determines the wake-up status duration after any wake-up event before the SAF1760 goes back into suspend mode. This time-out is applicable only if, during the given interval, the host controller is not programmed back to the normal functionality. **03E8h —** The default value. It determines the default wake-up interval of 10 ms. A value of zero implies that the host controller never wakes up on any of the events. This may be useful when using the SAF1760 as a peripheral to save power by permanently programming the host controller in suspend. **FFFFh —** The maximum value. It determines a maximum wake-up time of 500 ms. The setting of this register is based on the 100 kHz ± 40 % LazyClock frequency. It is a multiple of 10 µs period. **Remark:** In 16-bit mode, the default value is 17E8h. A write operation to these bits with any value fixes the clock off counter at 1400h. This value is equivalent to a fixed wake-up time of 50 ms.
15 to 13	-	reserved
12	PORT3_PD	**Port 3 Pull-Down**: Controls port 3 pull-down resistors. **0 —** Port 3 internal pull-down resistors are not connected. **1 —** Port 3 internal pull-down resistors are connected.
11	PORT2_PD	**Port 2 Pull-Down**: Controls port 2 pull-down resistors. **0 —** Port 2 internal pull-down resistors are not connected. **1 —** Port 2 internal pull-down resistors are connected.

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

Bit[1]	Symbol	Description
10	VBATDET_PWR	**VBAT Detector Powered**: Controls the power to the V_{BAT} detector. **0 —** V_{BAT} detector is powered or enabled in suspend. **1 —** V_{BAT} detector is not powered or disabled in suspend.
9 to 6	-	reserved; write reset value
5	BIASEN	**Bias Circuits Powered**: Controls the power to internal bias circuits. **0 —** Internal bias circuits are not powered in suspend. **1 —** Internal bias circuits are powered in suspend.
4	VREG_ON	**VREG Powered**: Enables or disables the internal 3.3 V and 1.8 V regulators when the SAF1760 is in suspend. **0 —** Internal regulators are normally powered in suspend. **1 —** Internal regulators switch to low power mode (in suspend mode).
3	OC3_PWR	**OC3_N Powered**: Controls the powering of the overcurrent detection circuitry for port 3. **0 —** Overcurrent detection is powered-on or enabled during suspend. **1 —** Overcurrent detection is powered-off or disabled during suspend. This may be useful when connecting a faulty device while the system is in standby.
2	OC2_PWR	**OC2_N Powered**: Controls the powering of the overcurrent detection circuitry for port 2. **0 —** Overcurrent detection is powered-on or enabled during suspend. **1 —** Overcurrent detection is powered-off or disabled during suspend. This may be useful when connecting a faulty device while the system is in standby.
1	OC1_PWR	**OC1_N Powered**: Controls the powering of the overcurrent detection circuitry for port 1. **0 —** Overcurrent detection is powered-on or enabled during suspend. **1 —** Overcurrent detection is powered-off or disabled during suspend. This may be useful when connecting a faulty device while the system is in standby.
0	HC_CLK_EN	**Host Controller Clock Enabled**: Controls internal clocks during suspend. **0 —** Clocks are disabled during suspend. This is the default value. Only the LazyClock of 100 kHz ± 40 % will be left running in suspend if this bit is logic 0. If clocks are stopped during suspend, CLKREADY IRQ will be generated when all clocks are running stable. **1 —** All clocks are enabled even in suspend.

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

8.3.12 Port 1 Control register

The values read from the lower 16 bits and the upper 16 bits of this register are always the same. Table 53 shows the bit allocation of the register.

Table 53. Port 1 Control register (address 0374h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	PORT1_INIT2	reserved						
Reset	1	0	0	0	0	1	1	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							
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	PORT1_INIT1	reserved		PORT1_POWER[1:0]		reserved		
Reset	0	0	0	1	0	1	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 54. Port 1 Control register (address 0374h) bit description

Bit[1]	Symbol	Description
31 to 24	-	reserved; write reset value
23	PORT1_INIT2	**Port 1 Initialization 2**: Write logic 1 at the SAF1760 initialization. It will clear both this bit and bit 7. Affects only port 1.
22 to 8	-	reserved
7	PORT1_INIT1	**Port 1 Initialization 1**: Must be reset to logic 0 at power-up initialization for correct operation of port 1. Correct host controller functionality is not ensured if set to logic 1 (affects only port 1). To clear this bit, logic 1 must be written to bit 23 during the SAF1760 initialization. This is not required for the normal functionality of port 2 and port 3.
6 to 5	-	reserved
4 to 3	PORT1_POWER[1:0]	**Port 1 Power**: Set these bits to 11b. These bits must be set to enable port 1 power.
2 to 0	-	reserved; write reset value

[1] For correct port 1 initialization, write 0080 0018h to this register after power-on.

8.4 Interrupt registers

8.4.1 Interrupt register

The bits of this register indicate the interrupt source, defining the events that determined the INT generation. Clearing the bits that were set because of the events listed is done by writing back logic 1 to the respective position. All bits must be reset before enabling new interrupt events. These bits will be set, regardless of the setting of bit GLOBAL_INTR_EN in the HW Mode Control register. Table 55 shows the bit allocation of the Interrupt register.

Table 55. Interrupt register (address 0310h) bit allocation

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

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

Table 56. Interrupt register (address 0310h) bit description

Bit	Symbol	Description
31 to 10	-	reserved; write reset value
9	ISO_IRQ	**ISO IRQ**: Indicates that an ISO PTD was completed, or the PTDs corresponding to the bits set in the ISO IRQ Mask AND or ISO IRQ Mask OR register bits combination were completed. The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** No ISO PTD event occurred. **1 —** ISO PTD event occurred. For details, see Section 7.4.
8	ATL_IRQ	**ATL IRQ**: Indicates that an ATL PTD was completed, or the PTDs corresponding to the bits set in the ATL IRQ Mask AND or ATL IRQ Mask OR register bits combination were completed. The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** No ATL PTD event occurred. **1 —** ATL PTD event occurred. For details, see Section 7.4.
7	INT_IRQ	**INT IRQ**: Indicates that an INT PTD was completed, or the PTDs corresponding to the bits set in the INT IRQ Mask AND or INT IRQ Mask OR register bits combination were completed. The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** No INT PTD event occurred. **1 —** INT PTD event occurred. For details, see Section 7.4.
6	CLKREADY	**Clock Ready**: Indicates that internal clock signals are running stable. The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** No CLKREADY event has occurred. **1 —** CLKREADY event occurred.
5	HC_SUSP	**Host Controller Suspend**: Indicates that the host controller has entered suspend mode. The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** The host controller did not enter suspend mode. **1 —** The host controller entered suspend mode. If the Interrupt Service Routine (ISR) accesses the SAF1760, it will wake up for the time specified in bits 31 to 16 of the Power-Down Control register.
4	-	reserved; write reset value
3	DMAEOT INT	**DMA EOT Interrupt**: Indicates the DMA transfer completion. The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** No DMA transfer is completed. **1 —** DMA transfer is complete.
2	-	reserved; write reset value; value is zero just after reset and changes to one after a short while
1	SOFITLINT	**SOT ITL Interrupt**: The IRQ line will be asserted if the respective enable bit in the HCInterruptEnable register is set. **0 —** No SOF event has occurred. **1 —** An SOF event has occurred.
0	-	reserved; write reset value; value is zero just after reset and changes to one after a short while

8.4.2 Interrupt Enable register

This register allows enabling or disabling of the IRQ generation because of various events as described in Table 57.

Table 57. Interrupt Enable register (address 0314h) bit allocation

Bit	31	30	29	28	27	26	25	24
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Symbol	reserved[1]							
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Symbol	reserved[1]						ISO_IRQ_E	ATL_IRQ_E
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Symbol	INT_IRQ_E	CLKREADY_E	HCSUSP_E	reserved[1]	DMAEOTINT_E	reserved[1]	SOFITLINT_E	reserved[1]
Reset	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Table 58. Interrupt Enable register (address 0314h) bit description

Bit	Symbol	Description
31 to 10	-	reserved; write logic 0
9	ISO_IRQ_E	**ISO IRQ Enable**: Controls the IRQ assertion when one or more ISO PTDs matching the ISO IRQ Mask AND or ISO IRQ Mask OR register bits combination are completed. **0 —** No IRQ will be asserted when ISO PTDs are completed. **1 —** IRQ will be asserted. For details, see Section 7.4.
8	ATL_IRQ_E	**ATL IRQ Enable**: Controls the IRQ assertion when one or more ATL PTDs matching the ATL IRQ Mask AND or ATL IRQ Mask OR register bits combination are completed. **0 —** No IRQ will be asserted when ATL PTDs are completed. **1 —** IRQ will be asserted. For details, see Section 7.4.
7	INT_IRQ_E	**INT IRQ Enable**: Controls the IRQ assertion when one or more INT PTDs matching the INT IRQ Mask AND or INT IRQ Mask OR register bits combination are completed. **0 —** No IRQ will be asserted when INT PTDs are completed. **1 —** IRQ will be asserted. For details, see Section 7.4.

Table 58. Interrupt Enable register (address 0314h) bit description *...continued*

Bit	Symbol	Description
6	CLKREADY_E	**Clock Ready Enable**: Enables the IRQ assertion when internal clock signals are running stable. Useful after wake-up. **0 —** No IRQ will be generated after a CLKREADY_E event. **1 —** IRQ will be generated after a CLKREADY_E event.
5	HCSUSP_E	**Host Controller Suspend Enable**: Enables the IRQ generation when the host controller enters suspend mode. **0 —** No IRQ will be generated when the host controller enters suspend mode. **1 —** IRQ will be generated when the host controller enters suspend mode.
4	-	reserved; write logic 0
3	DMAEOTINT_E	**DMA EOT Interrupt Enable**: Controls assertion of IRQ on the DMA transfer completion. **0 —** No IRQ will be generated when a DMA transfer is completed. **1 —** IRQ will be asserted when a DMA transfer is completed.
2	-	reserved; must be written with logic 0
1	SOFITLINT_E	**SOT ITL Interrupt Enable**: Controls the IRQ generation at every SOF occurrence. **0 —** No IRQ will be generated on an SOF occurrence. **1 —** IRQ will be asserted at every SOF.
0	-	reserved; must be written with logic 0

8.4.3 ISO IRQ Mask OR register

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

Table 59. ISO IRQ Mask OR register (address 0318h) bit description

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

8.4.4 INT IRQ Mask OR register

Each bit of this register (see Table 60) 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 60. INT IRQ Mask OR register (address 031Ch) bit description

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

8.4.5 ATL IRQ Mask OR register

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

Table 61. ATL IRQ Mask OR register (address 0320h) bit description

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

8.4.6 ISO IRQ Mask AND register

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

Table 62 provides the bit description of the register.

Table 62. ISO IRQ Mask AND register (address 0324h) bit description

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

8.4.7 INT IRQ Mask AND register

Each bit of this register (see Table 63) 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 63. INT IRQ Mask AND register (address 0328h) bit description

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

8.4.8 ATL IRQ Mask AND register

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

Table 64 shows the bit description of the register.

Table 64. ATL IRQ Mask AND register (address 032Ch) bit description

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

9. Proprietary Transfer Descriptor (PTD)

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

The PTD structures of the SAF1760 are translations of the EHCI data structures that are optimized for the SAF1760. It, however, still follows the basic EHCI architecture. This optimized form of EHCI data structures is necessary because the SAF1760 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 reduce hardware and software complexity. The SAF1760 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 SAF1760 has a maximum of 32 ISO, 32 INTL and 32 ATL PTDs. These PTDs are used as channels to transfer data from the shared memory to the USB bus. These channels are allocated and de-allocated on receiving the transfer from the core USB driver.

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

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

There are four types of interrupts in the SAF1760: ISO, INTL, ATL and SOF. The latency can be programmed in multiples of µSOF (125 µs).

The NextPTD pointer is a feature that allows the SAF1760 to jump unused and skip PTDs. This will improve the PTD transversal latency time. The NextPTD pointer is not meant for same or single endpoint. The NextPTD works only in forward direction.

The NextPTD traversal rules defined by the SAF1760 hardware are:

1. Start the PTD memory vertical traversal, considering the skip and LastPTD information, as follows.
2. If the current PTD is active and not done, perform the transaction.
3. Follow the NextPTD pointer as specified in bits 4 to 0 of DW4.
4. If combined with LastPTD, the LastPTD setting must be at a higher address than the NextPTD specified. Both have to be set in a logical manner.
5. If combined with skip, the skip must not be set (logically) on the same position corresponding to NextPTD, pointed by the NextPTD pointer.
6. If PTD is set for skip, it will be neglected and the next vertical PTD will be considered.
7. If the skipped PTD already has a setting including a NextPTD pointer that will not be taken into consideration, the behavior will be just as described in the preceding step.

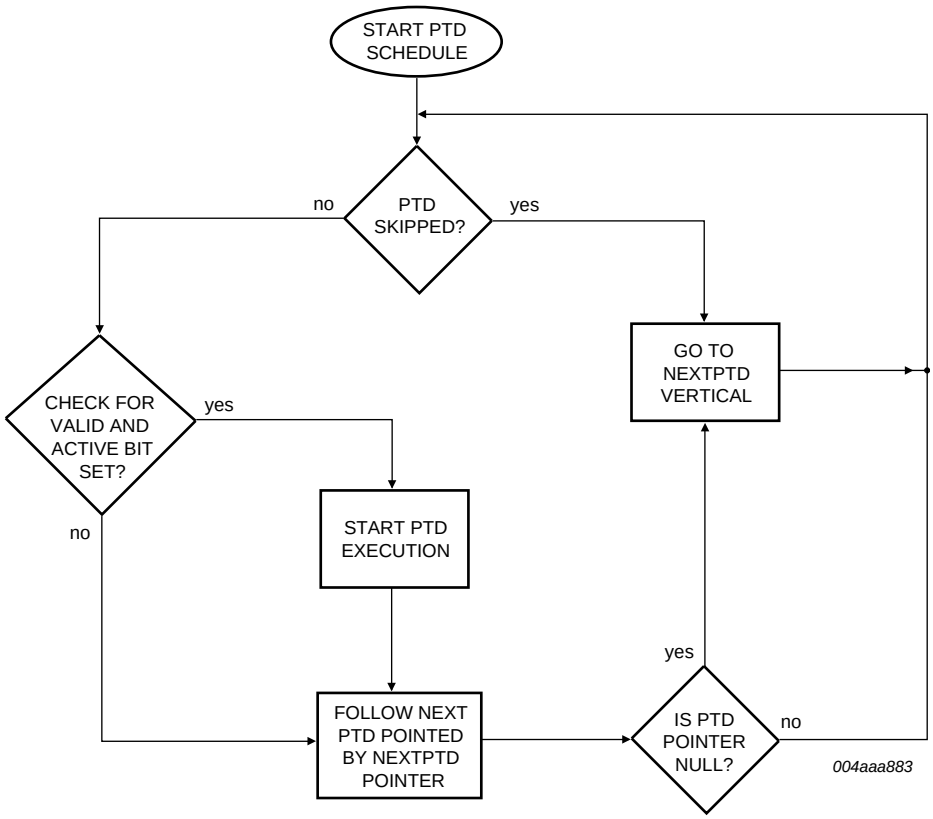


Fig 12. NextPTD traversal rule

9.1 High-speed bulk IN and OUT

Table 65 shows the bit allocation of the high-speed bulk IN and OUT, asynchronous Transfer Descriptor (TD).

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

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

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

[1] Reserved.

[2] EndPt[0].

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

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

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

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

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

Bit	Symbol	Access	Value	Description
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	Mult[1:0]	**SW —** writes	-	**Multiplier**: This field is a multiplier used by the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. Set this field to 01b. You can also set it to 11b and 10b depending on your application. 00b is undefined.
28 to 18	MaxPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from an endpoint in a single data packet. The maximum packet size for a bulk transfer is 512 bytes. The maximum packet size for the isochronous transfer is also variable at any whole number.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB – 1 B).
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**Valid**: **0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

9.2 High-speed isochronous IN and OUT

Table 67 shows the bit allocation of the high-speed isochronous IN and OUT, isochronous Transfer Descriptor (iTD).

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

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

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

[1] Reserved.

[2] EndPt[0].

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

Bit	Symbol	Access	Value	Description
DW7				
63 to 52	ISOIN_7[11:0]	**HW —** writes	-	Bytes received during µSOF7, if µSA[7] is set to 1 and frame number is correct.
51 to 40	ISOIN_6[11:0]	**HW —** writes	-	Bytes received during µSOF6, if µSA[6] is set to 1 and frame number is correct.
39 to 32	ISOIN_5[11:4]	**HW —** writes	-	Bytes received during µSOF5 (bits 11 to 4), if µSA[5] is set to 1 and frame number is correct.
DW6				
31 to 28	ISOIN_5[3:0]	**HW —** writes	-	Bytes received during µSOF5 (bits 3 to 0), if µSA[5] is set to 1 and frame number is correct.
27 to 16	ISOIN_4[11:0]	**HW —** writes	-	Bytes received during µSOF4, if µSA[4] is set to 1 and frame number is correct.
15 to 4	ISOIN_3[11:0]	**HW —** writes	-	Bytes received during µSOF3, if µSA[3] is set to 1 and frame number is correct.
3 to 0	ISOIN_2[11:8]	**HW —** writes	-	Bytes received during µSOF2 (bits 11 to 8), if µSA[2] is set to 1 and frame number is correct.
DW5				
63 to 56	ISOIN_2[7:0]	**HW —** writes	-	Bytes received during µSOF2 (bits 7 to 0), if µSA[2] is set to 1 and frame number is correct.
55 to 44	ISOIN_1[11:0]	**HW —** writes	-	Bytes received during µSOF1, if µSA[1] is set to 1 and frame number is correct.
43 to 32	ISOIN_0[11:0]	**HW —** writes	-	Bytes received during µSOF0, if µSA[0] is set to 1 and frame number is correct.
DW4				
31 to 29	Status7[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF7
28 to 26	Status6[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF6
25 to 23	Status5[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF5
22 to 20	Status4[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF4
19 to 17	Status3[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF3
16 to 14	Status2[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF2
13 to 11	Status1[2:0]	**HW —** writes	-	ISO IN or OUT status at µSOF1
10 to 8	Status0[2:0]	**HW —** writes	-	Status of the payload on the USB bus for this µSOF after ISO has been delivered. **Bit 0 —** Transaction error (IN and OUT) **Bit 1 —** Babble (IN token only) **Bit 2 —** Underrun (OUT token only)
7 to 0	µSA[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	**mSOF Active**: When the frame number of bits DW1[7:3] match the frame number of the USB bus, these bits are checked for 1 before they are sent for µSOF. For example: If µSA[7:0] = 1111 1111b: send ISO every µSOF of the entire millisecond. If µSA[7:0] = 0101 0101b: send ISO only on µSOF0, µSOF2, µSOF4 and µSOF6.

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

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

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

Bit	Symbol	Access	Value	Description
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	Mult[1:0]	**SW —** writes	-	This field is a multiplier counter used by the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. For details, refer to Appendix D of Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”.
28 to 18	MaxPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from the endpoint in a single data packet. The maximum packet size for an isochronous transfer is 1024 bytes. The maximum packet size for the isochronous transfer is also variable at any whole number.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB – 1 B).
2 to 1	reserved	-	-	-
0	V	**HW —** resets **SW —** sets	-	**0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

9.3 High-speed interrupt IN and OUT

Table 69 shows the bit allocation of the high-speed interrupt IN and OUT, periodic Transfer Descriptor (pTD).

Table 69. High-speed interrupt IN and OUT: bit allocation

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

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

[1] Reserved.

[2] EndPt[0].

Table 70. High-speed interrupt IN and OUT: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 52	INT_IN_7[11:0]	**HW —** writes	-	Bytes received during µSOF7, if µSA[7] is set to 1 and frame number is correct.
51 to 40	INT_IN_6[11:0]	**HW —** writes	-	Bytes received during µSOF6, if µSA[6] is set to 1 and frame number is correct.
39 to 32	INT_IN_5[11:4]	**HW —** writes	-	Bytes received during µSOF5 (bits 11 to 4), if µSA[5] is set to 1 and frame number is correct.
DW6				
31 to 28	INT_IN_5[3:0]	**HW —** writes	-	Bytes received during µSOF5 (bits 3 to 0), if µSA[5] is set to 1 and frame number is correct.
27 to 16	INT_IN_4[11:0]	**HW —** writes	-	Bytes received during µSOF4, if µSA[4] is set to 1 and frame number is correct.
15 to 4	INT_IN_3[11:0]	**HW —** writes	-	Bytes received during µSOF3, if µSA[3] is set to 1 and frame number is correct.
3 to 0	INT_IN_2[11:8]	**HW —** writes	-	Bytes received during µSOF2 (bits 11 to 8), if µSA[2] is set to 1 and frame number is correct.
DW5				
63 to 56	INT_IN_2[7:0]	**HW —** writes	-	Bytes received during µSOF2 (bits 7 to 0), if µSA[2] is set to 1 and frame number is correct.
55 to 44	INT_IN_1[11:0]	**HW —** writes	-	Bytes received during µSOF1, if µSA[1] is set to 1 and frame number is correct.
43 to 32	INT_IN_0[11:0]	**HW —** writes	-	Bytes received during µSOF0, if µSA[0] is set to 1 and frame number is correct.
DW4				
31 to 29	Status7[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF7
28 to 26	Status6[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF6
25 to 23	Status5[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF5
22 to 20	Status4[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF4
19 to 17	Status3[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF3
16 to 14	Status2[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF2
13 to 11	Status1[2:0]	**HW —** writes	-	INT IN or OUT status of µSOF1
10 to 8	Status0[2:0]	**HW —** writes	-	Status of the payload on the USB bus for this µSOF after INT has been delivered. **Bit 0 —** Transaction error (IN and OUT) **Bit 1 —** Babble (IN token only) **Bit 2 —** Underrun (OUT token only)
7 to 0	µSA[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	When the frame number of bits DW2[7:3] match the frame number of the USB bus, these bits are checked for 1 before they are sent for µSOF. For example: When µSA[7:0] = 1111 1111b: send INT for every µSOF of the entire millisecond. When µSA[7:0] = 0101 0101b: send INT for µSOF0, µSOF2, µSOF4 and µSOF6. When µSA[7:0] = 1000 1000b: send INT for every fourth µSOF.

Table 70. High-speed interrupt IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
DW3				
63	A	**HW —** writes **SW —** writes	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: Transaction is halted.
61 to 58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: Set the Data Toggle bit to start the PTD. Software writes the current transaction toggle value. Hardware writes the next transaction toggle value.
56 to 55	Cerr[1:0]	**HW —** writes **SW —** writes	-	**Error Counter**. This field corresponds to the Cerr[1:0] field in the TD. The default value of this field is zero for isochronous transactions.
54 to 47	reserved	-	-	-
46 to 32	NrBytes Transferred [14:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction. If Mult[1:0] is greater than one, it is possible to store intermediate results in this field.
DW2				
31 to 24	reserved	-	-	-
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the direct CPU address. RAM address = (CPU address – 400h) / 8
7 to 0	µFrame[7:0]	**SW —** writes	-	Bits 7 to 3 represent the polling rate in milliseconds. The INT polling rate is defined as $2^{(b-1)}$ µSOF, where b is 1 to 9. When b is 1, 2, 3 or 4, use µSA to define polling because the rate is equal to or less than 1 ms. Bits 7 to 3 are set to 0. Polling checks µSA bits for µSOF rates. See Table 71.
DW1				
63 to 47	reserved	-	-	-
46	S	**SW —** writes	-	This bit indicates if a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Endpoint type**: **11 —** Interrupt
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: This field indicates the token PID for this transaction: **00 —** OUT **01 —** IN
41 to 35	DeviceAddress [6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by the buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.

Table 70. High-speed interrupt IN and OUT: bit description *…continued*

Bit	Symbol	Access	Value	Description
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	Mult[1:0]	**SW —** writes	-	**Multiplier**: This field is a multiplier counter used by the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. Set this field to 01b. You can also set it to 11b and 10b, depending on your application. 00b is undefined.
28 to 18	MaxPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from the endpoint in a single data packet.
17 to 3	NrBytesTo Transfer[14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field (32 kB – 1 B).
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**Valid**: **0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

Table 71. Microframe description

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

9.4 Start and complete split for bulk

Table 72 shows the bit allocation of Start Split (SS) and Complete Split (CS) for bulk, asynchronous Start Split and Complete Split (SS/CS) Transfer Descriptor (TD).

Table 72. Start and complete split for bulk: bit allocation

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

[1] Reserved.

[2] EndPt[0].

Table 73. Start and complete split for bulk: bit description

Bit	Symbol	Access	Value	Description
DW7				
63 to 32	reserved	-	-	-
DW6				
31 to 0	reserved	-	-	-
DW5				
63 to 32	reserved	-	-	-
DW4				
31 to 6	reserved	-	-	-
5	J	**SW —** writes	-	**0 —** To increment the PTD pointer. **1 —** To enable the next PTD branching.
4 to 0	NextPTDPointer[4:0]	**SW —** writes	-	Next PTD branching assigned by the PTD pointer.
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: This bit corresponds to the Halt bit of the Status field of TD.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to the Babble Detected bit in the Status field of iTD, siTD or TD. **1 —** When babbling is detected, A and V are set to 0.
60	X	**HW —** writes	-	**Transaction Error**: This bit corresponds to the Transaction Error bit in the status field.
		SW — writes	-	**0 —** Before scheduling
59	SC	**SW —** writes 0 **HW —** updates	-	**Start/Complete**: **0 —** Start split **1 —** Complete split
58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: Set the Data Toggle bit to start for the PTD.
56 to 55	Cerr[1:0]	**HW —** updates **SW —** writes	-	**Error Counter**: This field contains the error count for asynchronous start and complete split (SS/CS) TD. When an error has no response or bad response, Cerr[1:0] will be decremented to zero and then Valid will be set to zero. A NAK or NYET will reset Cerr[1:0]. For details, refer to Section 4.12.1.2 of Ref. 2 “Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0”. If retry has insufficient time at the beginning of a new SOF, the first PTD must be this retry. This can be accomplished if aperiodic PTD is not advanced.
54 to 51	NakCnt[3:0]	**HW —** writes **SW —** writes	-	**NAK Counter**: The V bit is reset if NakCnt decrements to zero and RL is a nonzero value. Not applicable to isochronous split transactions.
50 to 47	reserved	-	-	-
46 to 32	NrBytes Transferred[14:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction.

Table 73. Start and complete split for bulk: bit description *...continued*

Bit	Symbol	Access	Value	Description
DW2				
31 to 29	reserved	-	-	-
28 to 25	RL[3:0]	**SW —** writes	-	**Reload**: If RL is set to 0h, hardware ignores the NakCnt value. Set RL and NakCnt to the same value before a transaction. For full-speed and low-speed transactions, set this field to 0000b. Not applicable to isochronous start split and complete split.
24	reserved	-	-	-
23 to 8	DataStartAddress [15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the direct CPU address. RAM address = (CPU address – 400h) / 8
7 to 0	reserved	-	-	-
DW1				
63 to 57	HubAddress[6:0]	**SW —** writes	-	**Hub Address**: This indicates the hub address.
56 to 50	PortNumber[6:0]	**SW —** writes	-	**Port Number**: This indicates the port number of the hub or embedded TT.
49 to 48	SE[1:0]	**SW —** writes	-	This depends on the endpoint type and direction. It is valid only for split transactions. Table 74 applies to start split and complete split only.
47	reserved	-	-	-
46	S	**SW —** writes	-	This bit indicates whether a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	**Endpoint Type**: **00 —** Control **10 —** Bulk
43 to 42	Token[1:0]	**SW —** writes	-	**Token**: This field indicates the PID for this transaction. **00 —** OUT **01 —** IN **10 —** SETUP
41 to 35	DeviceAddress[6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.

Table 73. Start and complete split for bulk: bit description *...continued*

Bit	Symbol	Access	Value	Description
DW0				
31	EndPt[0]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.
30 to 29	reserved	-	-	-
28 to 18	MaximumPacket Length[10:0]	**SW —** writes	-	**Maximum Packet Length**: This field indicates the maximum number of bytes that can be sent to or received from an endpoint in a single data packet. The maximum packet size for full-speed is 64 bytes as defined in the Ref. 1 “Universal Serial Bus Specification Rev. 2.0”.
17 to 3	NrBytesToTransfer [14:0]	**SW —** writes	-	**Number of Bytes to Transfer**: This field indicates the number of bytes that can be transferred by this data structure. It is used to indicate the depth of the DATA field.
2 to 1	reserved	-	-	-
0	V	**SW —** sets **HW —** resets	-	**Valid**: **0 —** This bit is deactivated when the entire PTD is executed, or when a fatal error is encountered. **1 —** Software updates to one when there is payload to be sent or received. The current PTD is active.

Table 74. SE description

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

9.5 Start and complete split for isochronous

Table 75 shows the bit allocation for start and complete split for isochronous, split isochronous Transfer Descriptor (siTD).

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

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

[1] Reserved.

[2] EndPt[0].

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

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

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

Bit	Symbol	Access	Value	Description
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: The Halt bit is set when any microframe transfer status has a stalled or halted condition.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to bit 1 of Status0 to Status7 for every microframe transfer status.
60	X	**HW —** writes	-	**Transaction Error**: This bit corresponds to bit 0 of Status0 to Status7 for every microframe transfer status.
59	SC	**SW —** writes 0 **HW —** updates	-	**Start/Complete**: **0 —** Start split **1 —** Complete split
58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: Set the Data Toggle bit to start for the PTD.
56 to 44	reserved	-	-	-
43 to 32	NrBytes Transferred[11:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction.
DW2				
31 to 24	reserved	-	-	-
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the CPU address.
7 to 0	µFrame[7:0]	**SW —** writes	-	Bits 7 to 3 determine which frame to execute.
DW1				
63 to 57	HubAddress [6:0]	**SW —** writes	-	**Hub Address**: This indicates the hub address.
56 to 50	PortNumber [6:0]	**SW —** writes	-	**Port Number**: This indicates the port number of the hub or embedded TT.
49 to 47	reserved	-	-	-
46	S	**SW —** writes	-	This bit indicates whether a split transaction has to be executed: **0 —** High-speed transaction **1 —** Split transaction
45 to 44	EPType[1:0]	**SW —** writes	-	Transaction type: **01 —** Isochronous
43 to 42	Token[1:0]	**SW —** writes	-	Token PID for this transaction: **00 —** OUT **01 —** IN
41 to 35	DeviceAddress [6:0]	**SW —** writes	-	**Device Address**: This is the USB address of the function containing the endpoint that is referred to by this buffer.
34 to 32	EndPt[3:1]	**SW —** writes	-	**Endpoint**: This is the USB address of the endpoint within the function.

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

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

9.6 Start and complete split for interrupt

Table 77 shows the bit allocation of start and complete split for interrupt.

Table 77. Start and complete split for interrupt: bit allocation

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

[1] Reserved.

[2] EndPt[0].

Table 78. Start and complete split for interrupt: bit description

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

Table 78. Start and complete split for interrupt: bit description *...continued*

Bit	Symbol	Access	Value	Description
7 to 0	µSA[7:0]	**SW —** writes (0 → 1) **HW —** writes (1 → 0) After processing	-	Specifies which µSOF the start split needs to be placed. **For OUT token**: When the frame number of bits DW1[7:3] matches the frame number of the USB bus, these bits are checked for one before they are sent for the µSOF. **For IN token**: Only µSOF0, µSOF1, µSOF2 or µSOF3 can be set to 1. Nothing can be set for µSOF4 and above.
DW3				
63	A	**SW —** sets **HW —** resets	-	**Active**: Write the same value as that in V.
62	H	**HW —** writes	-	**Halt**: The Halt bit is set when any microframe transfer status has a stalled or halted condition.
61	B	**HW —** writes	-	**Babble**: This bit corresponds to bit 1 of Status0 to Status7 for every microframe transfer status.
60	X	**HW —** writes	-	**Transaction Error**: This bit corresponds to bit 0 of Status0 to Status7 for every microframe transfer status.
59	SC	**SW —** writes 0 **HW —** updates	-	**Start/Complete**: **0 —** Start split **1 —** Complete split
58	reserved	-	-	-
57	DT	**HW —** writes **SW —** writes	-	**Data Toggle**: For an interrupt transfer, set correct bit to start the PTD.
56 to 55	Cerr[1:0]	**HW —** writes **SW —** writes	-	**Error Counter**: This field corresponds to the Cerr[1:0] field in TD. **00 —** The transaction will not retry. **11 —** The transaction will retry three times. Hardware will decrement these values.
54 to 44	reserved	-	-	-
43 to 32	NrBytes Transferred[11:0]	**HW —** writes	-	**Number of Bytes Transferred**: This field indicates the number of bytes sent or received for this transaction.
DW2				
31 to 24	reserved	-	-	-
23 to 8	DataStart Address[15:0]	**SW —** writes	-	**Data Start Address**: This is the start address for data that will be sent or received on or from the USB bus. This is the internal memory address and not the CPU address.
7 to 0	µFrame[7:0]	**SW —** writes	-	Bits 7 to 3 is the polling rate in milliseconds. Polling rate is defined as $2^{(b-1)}$ µSOF; where b = 4 to 16. When b is 4, executed every millisecond. See Table 79.
DW1				
63 to 57	HubAddress [6:0]	**SW —** writes	-	**Hub Address**: This indicates the hub address.
56 to 50	PortNumber[6:0]	**SW —** writes	-	**Port Number**: This indicates the port number of the hub or embedded TT.
49 to 48	SE[1:0]	**SW —** writes	-	This depends on the endpoint type and direction. It is valid only for split transactions. Table 80 applies to start split and complete split only.
47	reserved	-	-	-

Table 78. Start and complete split for interrupt: bit description *...continued*

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

Table 79. Microframe description

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

Table 80. SE description

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

10. Power consumption

Table 81. Power consumption

Number of ports working	I_{CC}
One port working (high-speed)	
$V_{CC(5V0)}$ = 5.0 V, $V_{CC(I/O)}$ = 3.3 V	90 mA
$V_{CC(5V0)}$ = 3.3 V, $V_{CC(I/O)}$ = 3.3 V	77 mA
$V_{CC(5V0)}$ = 5.0 V, $V_{CC(I/O)}$ = 1.8 V	82 mA
$V_{CC(5V0)}$ = 3.3 V, $V_{CC(I/O)}$ = 1.8 V	77 mA
Two ports working (high-speed)	
$V_{CC(5V0)}$ = 5.0 V, $V_{CC(I/O)}$ = 3.3 V	110 mA
$V_{CC(5V0)}$ = 3.3 V, $V_{CC(I/O)}$ = 3.3 V	97 mA
$V_{CC(5V0)}$ = 5.0 V, $V_{CC(I/O)}$ = 1.8 V	102 mA
$V_{CC(5V0)}$ = 3.3 V, $V_{CC(I/O)}$ = 1.8 V	97 mA
Three ports working (high-speed)	
$V_{CC(5V0)}$ = 5.0 V, $V_{CC(I/O)}$ = 3.3 V	130 mA
$V_{CC(5V0)}$ = 3.3 V, $V_{CC(I/O)}$ = 3.3 V	117 mA
$V_{CC(5V0)}$ = 5.0 V, $V_{CC(I/O)}$ = 1.8 V	122 mA
$V_{CC(5V0)}$ = 3.3 V, $V_{CC(I/O)}$ = 1.8 V	117 mA

The idle operating current, I_{CC}, that is, when the SAF1760 is in operational mode, initialized and without any devices connected, is 70 mA. The additional current consumption on I_{CC} is below 1 mA per port in the case of full-speed and low-speed devices.

Deep-sleep suspend mode ensures the lowest power consumption when $V_{CC(5V0)}$ is always supplied to the SAF1760. The suspend current $I_{CC(susp)}$ is typically about 150 µA at room temperature. The suspend current may increase if the ambient temperature increases. For details, see Section 7.6.

In hybrid mode, when $V_{CC(5V0)}$ is disconnected, $I_{CC(I/O)}$ will generally be below 100 µA. The average value is 60 µA to 70 µA.

Under the condition of constant read and write accesses occurring on the 32-bit data bus, the maximum $I_{CC(I/O)}$ drawn from $V_{CC(I/O)}$ is measured as 25 mA when the NXP SAF1760 evaluation board is connected to a BSQUARE PXA255 development platform. This current will vary depending on the platform because of the different access timing, the type of data patterns written on the data bus, and loading on the data bus.

11. Limiting values

Table 82. Limiting values

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

Symbol	Parameter	Conditions		Min	Max	Unit
$V_{CC(I/O)}$	input/output supply voltage			–0.5	+4.6	V
$V_{CC(5V0)}$	supply voltage (5.0 V)			–0.5	+5.6	V
V_{ESD}	electrostatic discharge voltage	human body model	[1]			
		pin 123, 124, 125, 126		–1750	+1750	V
		all other pins		–2000	+2000	V
		charge device model	[2]			
		corner pins		–750	+750	V
		all other pins		–500	+500	V
T_{stg}	storage temperature			–40	+125	°C

[1] Class 2 according to *JEDEC JESD22-A114*.

[2] Class III following *JEDEC JESD22-C101*.

12. Recommended operating conditions

Table 83. Recommended operating conditions

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
$V_{CC(I/O)}$	input/output supply voltage	$V_{CC(I/O)}$ = 3.3 V		3.0	3.3	3.6	V
		$V_{CC(I/O)}$ = 1.8 V		1.65	1.8	1.95	V
$V_{CC(5V0)}$	supply voltage (5.0 V)			3.0	-	5.5	V
T_{amb}	ambient temperature			–40	-	+85	°C
$I_{CC(susp)}$	suspend supply current	$V_{CC(5V0)}$ = 3.3 V	[1]				
		T_{amb} = 25 °C		-	150	-	µA
		T_{amb} = 40°C		-	300	-	µA
		T_{amb} = 85°C		-	1	-	mA

[1] Deep-sleep suspend mode.

13. Static characteristics

Table 84. Static characteristics: digital pins

Digital pins: A[17:1], DATA[31:0], CS_N, RD_N, WR_N, DACK, DREQ, IRQ, RESET_N, SUSPEND/WAKEUP_N, CLKIN, OC1_N, OC2_N, OC3_N.

OC1_N, OC2_N and OC3_N are used as digital overcurrent pins; T_{amb} = –40 °C to +85 °C; unless otherwise specified.

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

Table 85. Static characteristics: PSW1_N, PSW2_N, PSW3_N

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

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

Table 86. Static characteristics: POR

T_{amb} = –40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{trip(H)}$	HIGH-level trip voltage		1.0	1.2	1.4	V
$V_{trip(L)}$	LOW-level trip voltage		0.95	1.1	1.3	V
t_{PORP}	internal POR pulse width	after REG1V8 > $V_{trip(H)}$	200	-	-	ns

Table 87. Static characteristics: REF5V

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

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

Table 88. Static characteristics: USB interface block (pins DM1 to DM3 and DP1 to DP3)

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

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Input levels for high-speed							
V_{HSSQ}	high-speed squelch detection threshold voltage (differential signal amplitude)	squelch detected		-	-	100	mV
		no squelch detected		150	-	-	mV
V_{HSDSC}	high-speed disconnect detection threshold voltage (differential signal amplitude)	disconnect detected		625	-	-	mV
		disconnect not detected		-	-	525	mV
V_{HSCM}	high-speed data signaling common mode voltage range (guideline for receiver)			–50	-	+500	mV
Output levels for high-speed							
V_{HSOI}	high-speed idle level voltage			–10	-	+10	mV
V_{HSOH}	high-speed data signaling HIGH-level voltage			360	-	440	mV
V_{HSOL}	high-speed data signaling LOW-level voltage			–10	-	+10	mV
V_{CHIRPJ}	Chirp J level (differential voltage)		[1]	700	-	1100	mV
V_{CHIRPK}	Chirp K level (differential voltage)		[1]	–900	-	–500	mV
Input levels for full-speed and low-speed							
V_{IH}	HIGH-level input voltage			2.0	-	-	V
V_{IHZ}	HIGH-level input voltage (floating) for low-/full-speed			2.7	-	3.6	V
V_{IL}	LOW-level input voltage			-	-	0.8	V
V_{DI}	differential input sensitivity voltage	$\lvert V_{DP} - V_{DM} \rvert$		0.2	-	-	V
V_{CM}	differential common mode voltage range			0.8	-	2.5	V
Output levels for full-speed and low-speed							
V_{OH}	HIGH-level output voltage			2.8	-	3.6	V
V_{OL}	LOW-level output voltage			0	-	0.3	V
V_{OSE1}	single-ended 1 (SE1) output voltage			0.8	-	-	V
V_{CRS}	output signal crossover voltage			1.3	-	2.0	V

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

14. Dynamic characteristics

Table 89. Dynamic characteristics: system clock timing

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

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

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

[2] Recommended accuracy of the clock frequency is 50×10^{-6} for the crystal and oscillator. The oscillator used depends on $V_{CC(I/O)}$.

Table 90. Dynamic characteristics: CPU interface block

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
SR	slew rate	standard load	1	-	4	V/ns

Table 91. Dynamic characteristics: high-speed source electrical characteristics

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

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Driver characteristics							
t_{HSR}	rise time (10 % to 90 %)			500	-	-	ps
t_{HSF}	fall time (10 % to 90 %)			500	-	-	ps
Z_{HSDRV}	driver output impedance (which also serves as high-speed termination)	includes the R_S resistor	[1]	40.5	45	49.5	Ω
Clock timing							
t_{HSDRAT}	high-speed data rate			479.76	-	480.24	Mbit/s
t_{HSFRAM}	microframe interval			124.9375	-	125.0625	µs
t_{HSRFI}	consecutive microframe interval difference			1	-	8.33	ns

[1] This also serves as a high-speed termination.

Table 92. Dynamic characteristics: full-speed source electrical characteristics

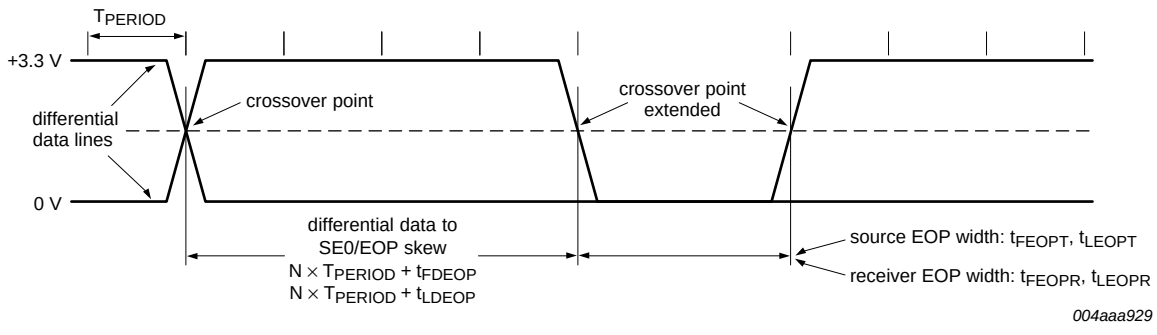
$V_{CC(I/O)}$ = 1.65 V to 3.6 V; T_{amb} = –40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Driver characteristics						
t_{FR}	rise time	C_L = 50 pF; 10 % to 90 % of $\|V_{OH} - V_{OL}\|$	4	-	20	ns
t_{FF}	fall time	C_L = 50 pF; 90 % to 10 % of $\|V_{OH} - V_{OL}\|$	4	-	20	ns
t_{FRFM}	differential rise and fall time matching		90	-	111.1	%
Data timing: see Figure 13						
t_{FDEOP}	source jitter for differential transition to SE0 transition	full-speed timing	–2	-	+5	ns
t_{FEOPT}	source SE0 interval of EOP		160	-	175	ns
t_{FEOPR}	receiver SE0 interval of EOP		82	-	-	ns
t_{LDEOP}	upstream facing port source jitter for differential transition to SE0 transition	low-speed timing	–40	-	+100	ns
t_{LEOPT}	source SE0 interval of EOP		1.25	-	1.5	µs
t_{LEOPR}	receiver SE0 interval of EOP		670	-	-	ns
t_{FST}	width of SE0 interval during differential transition		-	-	14	ns

Table 93. Dynamic characteristics: low-speed source electrical characteristics

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Driver characteristics						
t_{LR}	transition time: rise time		75	-	300	ns
t_{LF}	transition time: fall time		75	-	300	ns
t_{LRFM}	rise and fall time matching		90	-	125	%



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

Fig 13. USB source differential data-to-EOP transition skew and EOP width

14.1 PIO timing

14.1.1 Register or memory write

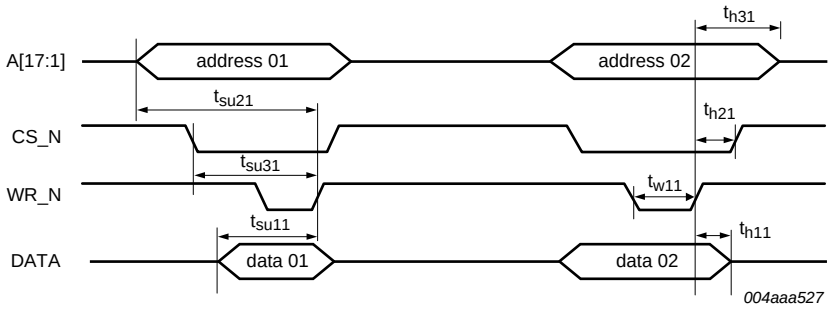


Fig 14. Register or memory write

Table 94. Register or memory write

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{h11}	data hold after WR_N HIGH	2	-	ns
t_{h21}	CS_N hold after WR_N HIGH	1	-	ns
t_{h31}	address hold after WR_N HIGH	2	-	ns
t_{w11}	WR_N pulse width	17	-	ns
t_{su11}	data set-up time before WR_N HIGH	5	-	ns
t_{su21}	address set-up time before WR_N HIGH	5	-	ns
t_{su31}	CS_N set-up time before WR_N HIGH	5	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{h11}	data hold after WR_N HIGH	2	-	ns
t_{h21}	CS_N hold after WR_N HIGH	1	-	ns
t_{h31}	address hold after WR_N HIGH	2	-	ns
t_{w11}	WR_N pulse width	17	-	ns
t_{su11}	data set-up time before WR_N HIGH	5	-	ns
t_{su21}	address set-up time before WR_N HIGH	5	-	ns
t_{su31}	CS_N set-up time before WR_N HIGH	5	-	ns

14.1.2 Register read

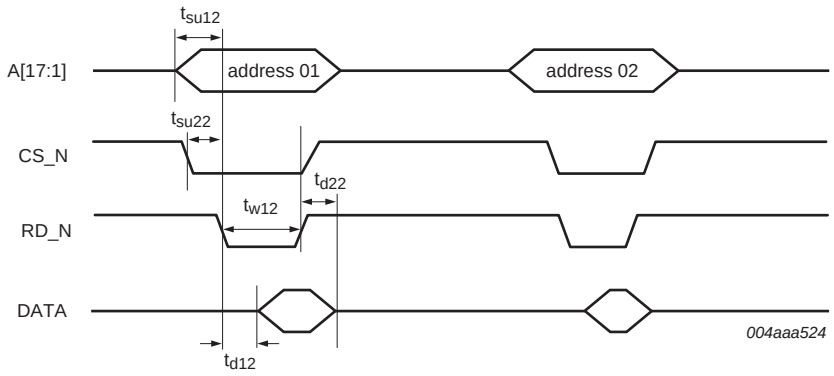


Fig 15. Register read

Table 95. Register read

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{su12}	address set-up time before RD_N LOW	0	-	ns
t_{su22}	CS_N set-up time before RD_N LOW	0	-	ns
t_{w12}	RD_N pulse width	> t_{d12}	-	ns
t_{d12}	data valid time after RD_N LOW	-	35	ns
t_{d22}	data valid time after RD_N HIGH	-	1	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{su12}	address set-up time before RD_N LOW	0	-	ns
t_{su22}	CS_N set-up time before RD_N LOW	0	-	ns
t_{w12}	RD_N pulse width	> t_{d12}	-	ns
t_{d12}	data valid time after RD_N LOW	-	22	ns
t_{d22}	data valid time after RD_N HIGH	-	1	ns

14.1.3 Register access

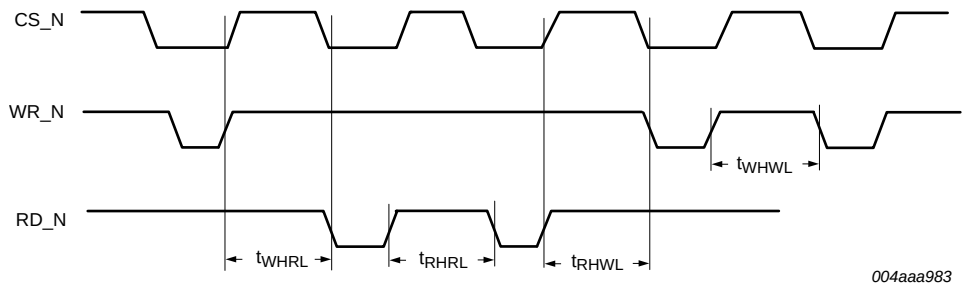


Fig 16. Register access

Table 96. Register access

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

Symbol	Parameter	Min	Max	Unit
t_{WHRL}	WR_N HIGH to RD_N LOW time	25[1]	-	ns
t_{RHRL}	RD_N HIGH to RD_N LOW time	25[1]	-	ns
t_{RHWL}	RD_N HIGH to WR_N LOW time	25	-	ns
t_{WHWL}	WR_N HIGH to WR_N LOW time	25[1]	-	ns

[1] For EHCI operational registers, minimum value is 195 ns.

14.1.4 Memory read

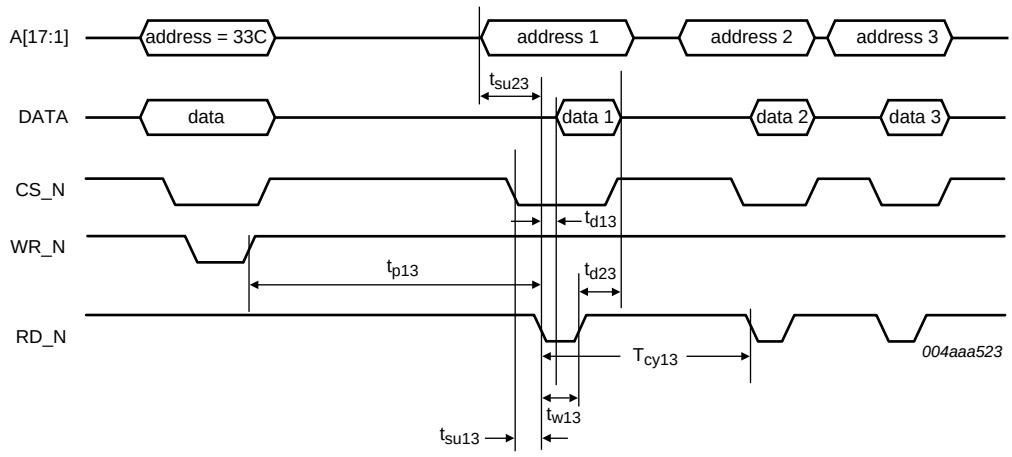


Fig 17. Memory read

Table 97. Memory read

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{p13}	initial pre-fetch time	90	-	ns
T_{cy13}	memory RD_N cycle time	40	-	ns
t_{d13}	data valid time after RD_N LOW	-	31	ns
t_{d23}	data available time after RD_N HIGH	-	1	ns
t_{w13}	RD_N pulse width	32	-	ns
t_{su13}	CS_N set-up time before RD_N LOW	0	-	ns
t_{su23}	address set-up time before RD_N LOW	0	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{p13}	initial pre-fetch time	90	-	ns
T_{cy13}	memory RD_N cycle time	36	-	ns
t_{d13}	data valid time after RD_N LOW	-	20	ns

Table 97. Memory read *...continued*

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

Symbol	Parameter	Min	Max	Unit
t_{d23}	data available time after RD_N HIGH	-	1	ns
t_{w13}	RD_N pulse width	21	-	ns
t_{su13}	CS_N set-up time before RD_N LOW	0	-	ns
t_{su23}	address set-up time before RD_N LOW	0	-	ns

14.2 DMA timing

In the following sections:

- Polarity of DACK is active HIGH
- Polarity of DREQ is active HIGH.

14.2.1 Single cycle: DMA read

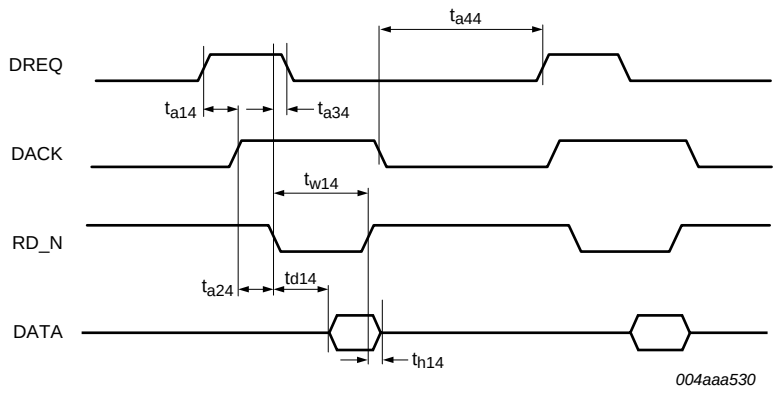


Fig 18. DMA read (single cycle)

Table 98. DMA read (single cycle)

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

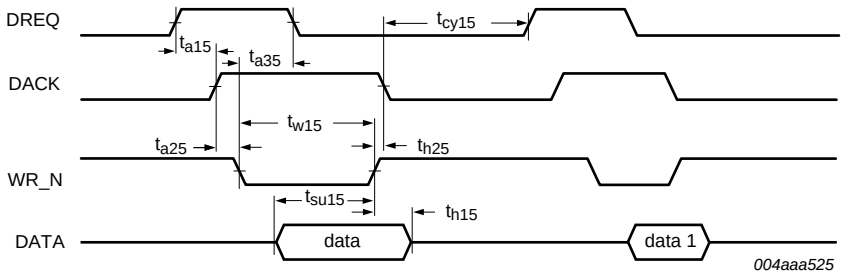
Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{a14}	DACK assertion time after DREQ assertion	0	-	ns
t_{a24}	RD_N assertion time after DACK assertion	0	-	ns
t_{d14}	data valid time after RD_N assertion	-	24	ns
t_{w14}	RD_N pulse width	> t_{d14}	-	ns
t_{a34}	DREQ de-assertion time after RD_N assertion	-	29	ns
t_{a44}	DACK de-assertion to next DREQ assertion time	-	56	ns
t_{h14}	data hold time after RD_N de-asserts	-	5	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{a14}	DACK assertion time after DREQ assertion	0	-	ns
t_{a24}	RD_N assertion time after DACK assertion	0	-	ns
t_{d14}	data valid time after RD_N assertion	-	20	ns

Table 98. DMA read (single cycle) *…continued*

T_{amb} = –40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
t_{w14}	RD_N pulse width	> t_{d14}	-	ns
t_{a34}	DREQ de-assertion time after RD_N assertion	-	18	ns
t_{a44}	DACK de-assertion to next DREQ assertion time	-	56	ns
t_{h14}	data hold time after RD_N de-asserts	-	5	ns

14.2.2 Single cycle: DMA write



DREQ and DACK are active HIGH.

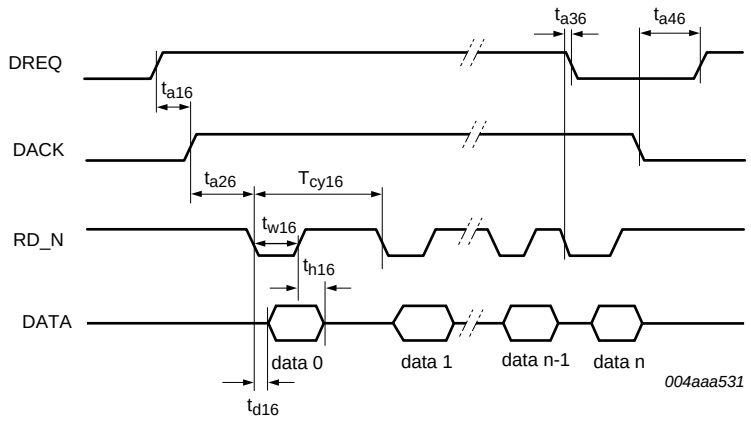
Fig 19. DMA write (single cycle)

Table 99. DMA write (single cycle)

T_{amb} = –40 °C to +85 °C; unless otherwise specified.

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{a15}	DACK assertion time after DREQ assertion	0	-	ns
t_{a25}	WR_N assertion time after DACK assertion	1	-	ns
t_{h15}	data hold time after WR_N de-assertion	3	-	ns
t_{h25}	DACK hold time after WR_N de-assertion	0	-	ns
t_{su15}	data set-up time before WR_N de-assertion	5.5	-	ns
t_{a35}	DREQ de-assertion time after WR_N assertion	-	28	ns
t_{cy15}	last DACK strobe de-assertion to next DREQ assertion time	-	82	ns
t_{w15}	WR_N pulse width	22	-	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{a15}	DACK assertion time after DREQ assertion	0	-	ns
t_{a25}	WR_N assertion time after DACK assertion	1	-	ns
t_{h15}	data hold time after WR_N de-assertion	2	-	ns
t_{h25}	DACK hold time after WR_N de-assertion	0	-	ns
t_{su15}	data set-up time before WR_N de-assertion	5.5	-	ns
t_{a35}	DREQ de-assertion time after WR_N assertion	-	16	ns
t_{cy15}	last DACK strobe de-assertion to next DREQ assertion time	-	82	ns
t_{w15}	WR_N pulse width	22	-	ns

14.2.3 Multi-cycle: DMA read



DREQ and DACK are active HIGH.

Fig 20. DMA read (multi-cycle burst)

Table 100. DMA read (multi-cycle burst)

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
t_{a16}	DACK assertion after DREQ assertion time	0	-	ns
t_{a26}	RD_N assertion after DACK assertion time	0	-	ns
t_{d16}	data valid time after RD_N assertion	-	31	ns
t_{w16}	RD_N pulse width	38	-	ns
T_{cy16}	read-to-read cycle time	46	-	ns
t_{a36}	DREQ de-assertion time after last burst RD_N de-assertion	-	30	ns
t_{a46}	DACK de-assertion to next DREQ assertion time	-	82	ns
t_{h16}	data hold time after RD_N de-asserts	-	5	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
t_{a16}	DACK assertion after DREQ assertion time	0	-	ns
t_{a26}	RD_N assertion after DACK assertion time	0	-	ns
t_{d16}	data valid time after RD_N assertion	-	16	ns
t_{w16}	RD_N pulse width	17	-	ns
T_{cy16}	read-to-read cycle time	38	-	ns
t_{a36}	DREQ de-assertion time after last burst RD_N de-assertion	-	20	ns
t_{a46}	DACK de-assertion to next DREQ assertion time	-	82	ns
t_{h16}	data hold time after RD_N de-asserts	-	5	ns

14.2.4 Multi-cycle: DMA write

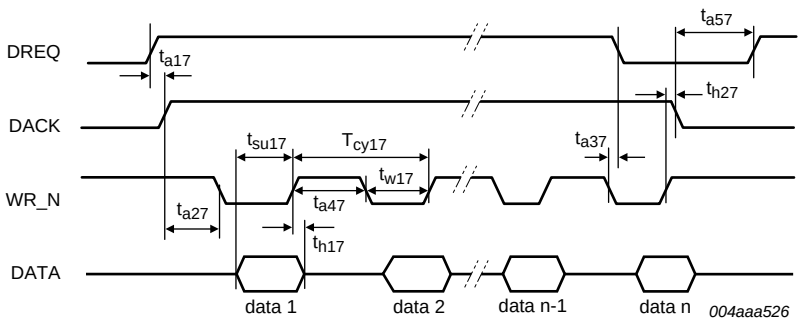


Fig 21. DMA write (multi-cycle burst)

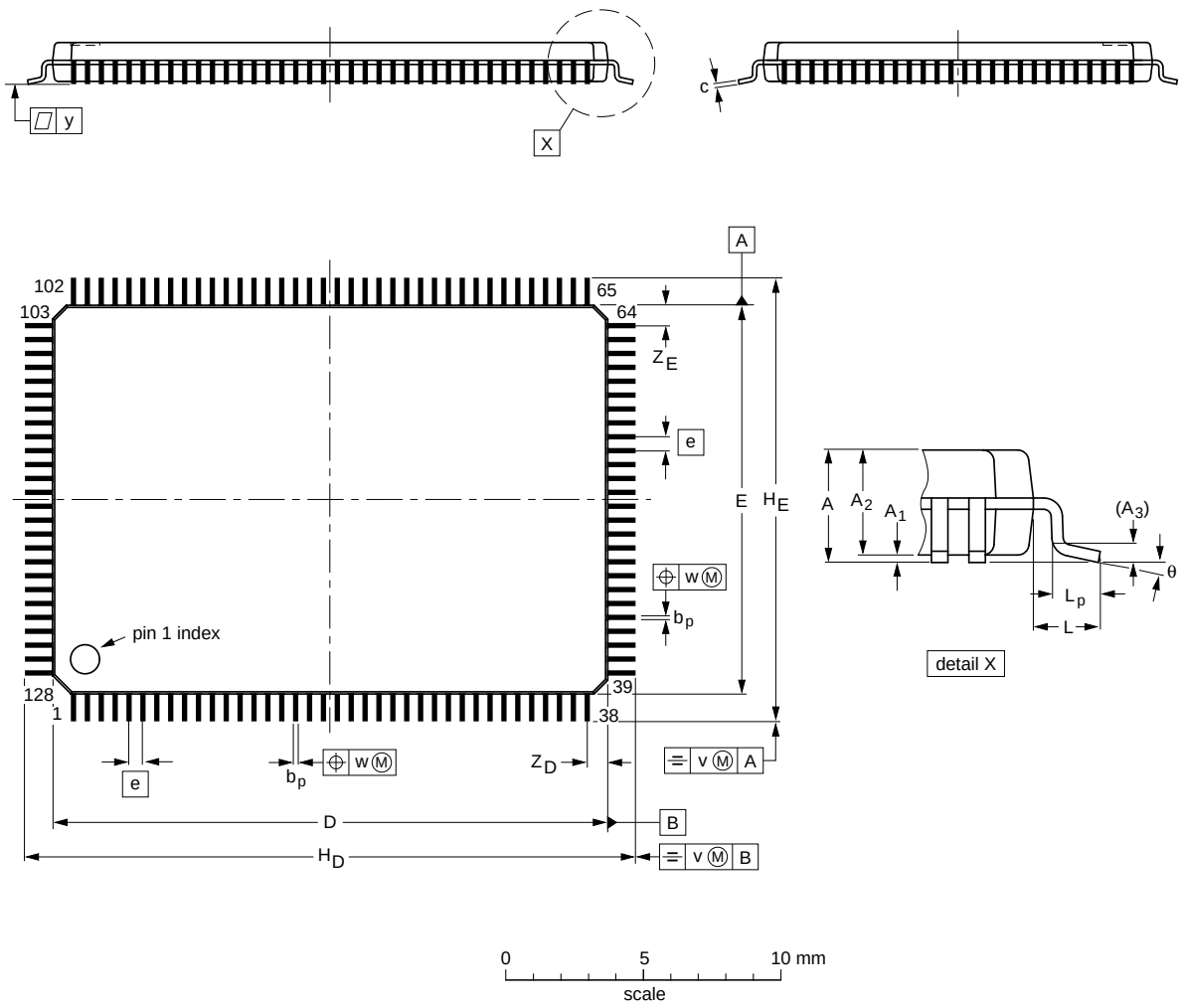
Table 101. DMA write (multi-cycle burst)

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

Symbol	Parameter	Min	Max	Unit
$V_{CC(I/O)}$ = 1.65 V to 1.95 V				
T_{cy17}	DMA write cycle time	51	-	ns
t_{su17}	data set-up time before WR_N de-assertion	5	-	ns
t_{h17}	data hold time after WR_N de-assertion	2	-	ns
t_{a17}	DACK assertion time after DREQ assertion	0	-	ns
t_{a27}	WR_N assertion time after DACK assertion	2	-	ns
t_{a37}	DREQ de-assertion time at last strobe (WR_N) assertion	-	28	ns
t_{h27}	DACK hold time after WR_N de-assertion	0	-	ns
t_{a47}	strobe de-assertion to next strobe assertion time	34	-	ns
t_{w17}	WR_N pulse width	17	-	ns
t_{a57}	DACK de-assertion to next DREQ assertion time	-	82	ns
$V_{CC(I/O)}$ = 3.3 V to 3.6 V				
T_{cy17}	DMA write cycle time	51	-	ns
t_{su17}	data set-up time before WR_N de-assertion	5	-	ns
t_{h17}	data hold time after WR_N de-assertion	2	-	ns
t_{a17}	DACK assertion time after DREQ assertion	0	-	ns
t_{a27}	WR_N assertion time after DACK assertion	1	-	ns
t_{a37}	DREQ de-assertion time at last strobe (WR_N) assertion	-	16	ns
t_{h27}	DACK hold time after WR_N de-assertion	0	-	ns
t_{a47}	strobe de-assertion to next strobe assertion time	34	-	ns
t_{w17}	WR_N pulse width	17	-	ns
t_{a57}	DACK de-assertion to next DREQ assertion time	-	82	ns

15. Package outline

LQFP128: plastic low profile quad flat package; 128 leads; body 14 x 20 x 1.4 mm **SOT425-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.15 0.05	1.45 1.35	0.25	0.27 0.17	0.20 0.09	20.1 19.9	14.1 13.9	0.5	22.15 21.85	16.15 15.85	1	0.75 0.45	0.2	0.12	0.1	0.81 0.59	0.81 0.59	7^o 0^o

Note

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

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT425-1	136E28	MS-026				~~00-01-19~~ 03-02-20

Fig 22. Package outline SOT425-1 (LQFP128)

16. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 "Surface mount reflow soldering description"*.

16.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

16.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

16.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

16.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 23) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 102 and 103

Table 102. SnPb eutectic process (from J-STD-020C)

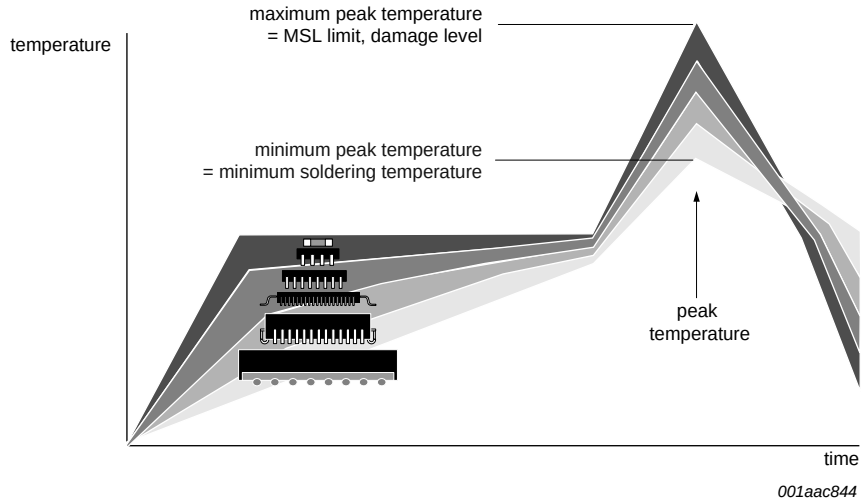
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm^3)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 103. Lead-free process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm^3)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

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

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 23.



MSL: Moisture Sensitivity Level

Fig 23. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

17. Appendix

17.1 Errata added on 2009-04-23

17.1.1 Problem description

When the SAF1760 is programmed to perform infinite retries on **Not Acknowledged (NAK) IN tokens**, the SAF1760 does not generate the retry IN tokens on its own. According to the SAF1760 data sheet, if register RL is programmed as zero, NakCnt is ignored. This means that irrespective of the value of NakCnt, the SAF1760 must retry indefinitely because there is no NakCnt to limit the number of retries for a NAK IN token. The SAF1760 hardware, however, does not retry the NAK IN token.

17.1.2 Implication

After an IN token is seen on the USB bus and is NAK-ed by the downstream device, the SAF1760 will not perform any automatic retry. Instead an interrupt will be generated in the ATL_IRQ bit of the HcInterrupt register (310h), and the software must refresh the IN token Proprietary Transfer Descriptor (PTD) for the retry to occur on the USB bus. This causes the software to constantly service this retry, unless the application instructs it to stop retrying.

17.1.3 Workaround

17.1.3.1 Software retry mechanism

Set program register RL = 1111b and NakCnt = 1111b. In this case, interrupt will be generated when NakCnt reaches zero and the software has to reload the transfer descriptor.

17.1.3.2 Hardware retry mechanism

Set program register RL = 0000b, NakCnt = 0000b and Cerr = 10b. In this case, interrupt will not be generated for NAKs and hardware will retry indefinitely, until the device responds with a data or an ACK.

17.2 Errata added on 2009-04-23

17.2.1 Problem description

When at least two USB devices are simultaneously running, it is observed that sometimes the INT corresponding to one of the USB devices stops occurring. This may be observed sometimes with USB-to-serial or USB-to-network devices.

The problem is not noticed when only USB mass storage devices are running.

17.2.2 Implication

This issue is because of the clearing of the respective **Done Map bit** on reading the ATL PTD Done Map register when an INT is generated by another PTD completion, but is not found set on that read access. In this situation, the respective Done Map bit will remain reset and no further INT will be asserted, so the data transfer corresponding to that USB device will stop.

17.2.3 Workaround

An **SOF INT** can be used instead of an **ATL INT** with polling on Done bits. A time-out can be implemented and if a certain Done bit is never set, verification of the PTD completion can be done by reading PTD contents (valid bit).

17.3 Errata added on 2009-04-23

17.3.1 Problem description

When a low-speed or full-speed device is attached, after some time, the low-speed or full-speed device suddenly gets disconnected.

The following sequence is observed when the problem occurs:

- The hub class driver detects a change of port status on the problematic port (through the interrupt endpoint of the hub).
- When a Get Port Status command is sent to the problematic port, the **Port Enable** bit of the Port Status is cleared. This indicates that a port error has occurred. However, the current connection status still indicates that a device is present on the port.
- The hub driver sends a Port Reset command because of the clearing of the port enable bit. This causes the disconnection of the attached device and its renumeration.
- Before the hub driver detects the port status change, all active transfers on the problematic port are halted.
- Low-speed or full-speed devices connected to other ports are not affected when the problem occurs.
- The problem occurs with low-speed or full-speed devices.
- When low-speed or full-speed devices are connected through a high-speed hub, the problem will not occur.

- During testing, it is observed that the problem always occurs on the port on which the device was last attached.

17.3.2 Implication

The implication will be serious if the device is getting disconnected during the data transfer.

17.3.3 Workaround

The software workaround will check if a port has suddenly been disabled (**Port Enable** bit cleared) when a device is still connected to the port. Once it detects this condition, the software workaround will perform the necessary steps to re-enable the port and reschedule any halted transfer because of the error condition.

The following actions are taken by the software once the error condition is detected:

- Increment the count of the variable that keeps the number of times the ports have been force enabled.
- Determine which active PTDs (ATL and INTL) are scheduled to the affected port and suspend them.
- Determine the speed of the device connected to the affected port.
- Put the internal hub in **Force Configure** mode.
- Force enable the affected port (this will set the Port Enable bit of the affected port to 1 again).
- Remove the force enable on the affected port.

  Remark: If the force enable is set, the particular port will always be enabled even if the device connected has been removed.
- Put the internal hub back to normal mode (exit from the force configure mode).
- Re-active all the suspended PTDs.

There are two conditions when the software workaround will be invoked and appropriate actions will be taken to determine if it truly is a problematic behavior (see Section 17.3.3.1, Section 17.3.3.2 and Section 17.3.3.3).

17.3.3.1 Condition 1

Condition 1 refers to the condition when **Port Enable/Disable Change** event is detected in the Hub Class driver. To determine and resolve the problematic condition, the following steps are taken:

- Determine if the hub event has occurred on one of the internal hubs three ports and if the connected device is either full-speed or low-speed.
- Determine if the port enable bit is cleared when a device is still connected on the port.
- Ensure that the port has not been force enabled three consecutive times by checking the variable that keeps track of the number of times the port has been force enabled.
- Otherwise, reset the port if it has been force enabled three consecutive times.
- Invoke the software workaround.

Check the Port Status again to see if the port has recovered. See also Section 17.3.3.3.

17.3.3.2 Condition 2

Condition 2 refers to the condition when a HALT occurs during PTD processing without other bus errors (that is, babble, transaction error).

To determine and resolve the problematic condition, the following steps are taken:

- Check the completion status once a PTD scheduled towards a full-speed or low-speed device and connected through the internal hub is completed.
- If the PTD has been completed successfully, clear the variable that keeps track of the number of the times the port has been force enabled.
- If the PTD has been completed with a HALT condition, get the port status of the port on which it is connected to the internal hub.
- Determine if the Port Enable bit is cleared when a device is still connected to the port.
- Ensure that the port has not been force enabled three consecutive times by checking the variable that keeps track of the number of times the port has been force enabled. Otherwise, reset the port if it has been force enabled three consecutive times.
- Invoke the software workaround.

See also Section 17.3.3.3.

17.3.3.3 Remarks

Because of this erratum, **Keep-Alive EOP** will not appear on the USB bus for more than 3 ms and the device will enter the suspend state. After implementing the above mentioned workaround, **Keep-Alive EOP** will start and the device will wake up to continue the function. In rare cases, because of the device implementation, there is a possibility that the device may not accept further requests from the host after the workaround. In such cases, application must perform the following **partial enumeration** steps to make the device work:

1. Port power off
2. Port power on
3. Get descriptor
4. Set interface

As SAF1760 has individual port power control mechanism, above sequence will be effective if the full-speed or low-speed device is directly connected to SAF1760 ports.

18. Abbreviations

Table 104. Abbreviations

Acronym	Description
ACK	ACKnowledgment
ASIC	Application-Specific Integrated Circuit
ASYNC	ASYNChronous
ATL	Asynchronous Transfer List
ATX	Analog Transceiver
BCD	Binary Coded Decimal
CPU	Central Processing Unit
CS	Complete Split
DSC	Digital Still Camera
DMA	Direct Memory Access
DW	Double Word
EHCI	Enhanced Host Controller Interface
EMI	ElectroMagnetic Interference
EOP	End-Of-Packet
EOT	End-Of-Transfer
ESR	Effective Series Resistance
FIFO	First In, First Out
FLS	Frame List Size
GPIO	General-Purpose I/O
HC	Host Controller
HS	High-Speed
HW	HardWare
IC	Integrated Circuit
ID	IDentification
IEC	International Electrotechnical Commission
INT	INTerrupt
I/O	Input and Output
IRQ	Interrupt ReQuest
ISO	ISOchronous
ISR	Interrupt Service Routine
iTD	isochronous Transfer Descriptor
ITL	Isochronous (ISO) Transfer List
LS	Line Status
NAK	Not AcKnowledged
NYET	Not YET
OC	OverCurrent
OHCI	Open Host Controller Interface
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistant

Table 104. Abbreviations *...continued*

Acronym	Description
PID	Packet ID
PIE	Parallel Interface Engine
PIO	Programmed I/O
PLL	Phase-Locked Loop
PMOS	Positive-channel Metal-Oxide Semiconductor
POR	Power-On Reset
PORP	Power-On Reset Pulse
PP	Port Power
PPC	Port Power Control
pTD	periodic Transfer Descriptor
PTD	Proprietary Transfer Descriptor
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
R/S	Run/Stop
R/W	Read/Write
SE0	Single Ended 0
SE1	Single Ended 1
siTD	split isochronous Transfer Descriptor
SOF	Start-Of-Frame
SRAM	Static RAM
SS	Start Split
SW	SoftWare
TD	Transfer Descriptor
TT	Transaction Translator
UHCI	Universal Host Controller Interface
USB	Universal Serial Bus
XOSC	crystal OSCillator[1]

[1] Letter X became a synonym for “crystal”.

19. Glossary

Bulk transfer — One of the four USB transfer types. It is an aperiodic, large burst communication, typically used for a transfer, which works with any available bandwidth. A bulk transfer can also be delayed until more bandwidth becomes available.

Endpoint — A uniquely addressable portion of an USB device that is the source or sink of information in a communication flow between the host and the device.

LazyClock — A slow clock frequency that is kept running while the chip is in suspend mode.

Microframe — A 125 µs time base established on high-speed buses.

20. References

[1] Universal Serial Bus Specification Rev. 2.0

[2] Enhanced Host Controller Interface Specification for Universal Serial Bus Rev. 1.0

21. Revision history

Table 105. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
SAF1760 v.2	20120619	Product data sheet	-	SAF1760 v.1
Modifications:	• Limit application to automotive use			
SAF1760_1	20091109	Product data sheet	-	-

22. Legal information

22.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

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

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

22.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

22.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use in automotive applications — This NXP Semiconductors product has been qualified for use in automotive applications. Unless otherwise agreed in writing, the product is not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

22.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

23. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

24. Tables

Table 1. Ordering information 2
Table 2. Pin description 4
Table 3. Port connection scenarios 13
Table 4. Memory address 15
Table 5. Using the IRQ Mask AND or IRQ Mask OR registers 21
Table 6. Hybrid mode 25
Table 7. Pin status in hybrid mode 25
Table 8. Register overview 27
Table 9. CAPLENGTH - Capability Length register (address 0000h) bit description 28
Table 10. HCIVERSION - Host Controller Interface Version Number register (address 0002h) bit description 28
Table 11. HCSPARAMS - Host Controller Structural Parameters register (address 0004h) bit allocation 29
Table 12. HCSPARAMS - Host Controller Structural Parameters register (address 0004h) bit description 29
Table 13. HCCPARAMS - Host Controller Capability Parameters register (address 0008h) bit allocation 30
Table 14. HCCPARAMS - Host Controller Capability Parameters register (address 0008h) bit description 30
Table 15. USBCMD - USB Command register (address 0020h) bit allocation 31
Table 16. USBCMD - USB Command register (address 0020h) bit description 31
Table 17. USBSTS - USB Status register (address 0024h) bit allocation 32
Table 18. USBSTS - USB Status register (address 0024h) bit description 32
Table 19. FRINDEX - Frame Index register (address: 002Ch) bit allocation 33
Table 20. FRINDEX - Frame Index register (address: 002Ch) bit description 33
Table 21. CONFIGFLAG - Configure Flag register (address 0060h) bit allocation 34
Table 22. CONFIGFLAG - Configure Flag register (address 0060h) bit description 34
Table 23. PORTSC1 - Port Status and Control 1 register (address 0064h) bit allocation 35
Table 24. PORTSC1 - Port Status and Control 1 register (address 0064h) bit description 35
Table 25. ISO PTD Done Map register (address 0130h) bit description 36
Table 26. ISO PTD Skip Map register (address 0134h) bit description 36
Table 27. ISO PTD Last PTD register (address 0138h) bit description 36
Table 28. INT PTD Done Map register (address 0140h) bit description 37
Table 29. INT PTD Skip Map register (address 0144h) bit description 37
Table 30. INT PTD Last PTD register (address 0148h) bit description 37
Table 31. ATL PTD Done Map register (address 0150h) bit description 38
Table 32. ATL PTD Skip Map register (address 0154h) bit description 38
Table 33. ATL PTD Last PTD register (address 0158h) bit description 38
Table 34. HW Mode Control - Hardware Mode Control register (address 0300h) bit allocation 39
Table 35. HW Mode Control - Hardware Mode Control register (address 0300h) bit description 39
Table 36. Chip ID - Chip Identifier register (address 0304h) bit description 40
Table 37. Scratch register (address 0308h) bit description 40
Table 38. SW Reset - Software Reset register (address 030Ch) bit allocation 41
Table 39. SW Reset - Software Reset register (address 030Ch) bit description 41
Table 40. DMA Configuration register (address 0330h) bit allocation 42
Table 41. DMA Configuration register (address 0330h) bit description 42
Table 42. Buffer Status register (address 0334h) bit allocation 43
Table 43. Buffer Status register (address 0334h) bit description 43
Table 44. ATL Done Timeout register (address 0338h) bit description 44
Table 45. Memory register (address 033Ch) bit allocation 44
Table 46. Memory register (address 033Ch) bit description 44
Table 47. Edge Interrupt Count register (address 0340h) bit allocation 45
Table 48. Edge Interrupt Count register (address 0340h) bit description 45
Table 49. DMA Start Address register (address 0344h) bit allocation 46

continued >>

Table 50. DMA Start Address register (address 0344h) bit description 46
Table 51. Power-Down Control register (address 0354h) bit allocation 47
Table 52. Power-Down Control register (address 0354h) bit description 47
Table 53. Port 1 Control register (address 0374h) bit allocation 49
Table 54. Port 1 Control register (address 0374h) bit description 49
Table 55. Interrupt register (address 0310h) bit allocation 50
Table 56. Interrupt register (address 0310h) bit description 51
Table 57. Interrupt Enable register (address 0314h) bit allocation 52
Table 58. Interrupt Enable register (address 0314h) bit description 52
Table 59. ISO IRQ Mask OR register (address 0318h) bit description 53
Table 60. INT IRQ Mask OR register (address 031Ch) bit description 53
Table 61. ATL IRQ Mask OR register (address 0320h) bit description 54
Table 62. ISO IRQ Mask AND register (address 0324h) bit description 54
Table 63. INT IRQ Mask AND register (address 0328h) bit description 54
Table 64. ATL IRQ Mask AND register (address 032Ch) bit description 54
Table 65. High-speed bulk IN and OUT: bit allocation . . . 57
Table 66. High-speed bulk IN and OUT: bit description . . 58
Table 67. High-speed isochronous IN and OUT: bit allocation 61
Table 68. High-speed isochronous IN and OUT: bit description 62
Table 69. High-speed interrupt IN and OUT: bit allocation 65
Table 70. High-speed interrupt IN and OUT: bit description 66
Table 71. Microframe description 68
Table 72. Start and complete split for bulk: bit allocation 69
Table 73. Start and complete split for bulk: bit description 70
Table 74. SE description 72
Table 75. Start and complete split for isochronous: bit allocation 73
Table 76. Start and complete split for isochronous: bit description 74
Table 77. Start and complete split for interrupt: bit allocation 77
Table 78. Start and complete split for interrupt: bit description 78
Table 79. Microframe description 80
Table 80. SE description 80
Table 81. Power consumption 81
Table 82. Limiting values 82
Table 83. Recommended operating conditions 82
Table 84. Static characteristics: digital pins 83
Table 85. Static characteristics: PSW1_N, PSW2_N, PSW3_N 83
Table 86. Static characteristics: POR 83
Table 87. Static characteristics: REF5V 83
Table 88. Static characteristics: USB interface block (pins DM1 to DM3 and DP1 to DP3) 84
Table 89. Dynamic characteristics: system clock timing . 85
Table 90. Dynamic characteristics: CPU interface block 85
Table 91. Dynamic characteristics: high-speed source electrical characteristics 85
Table 92. Dynamic characteristics: full-speed source electrical characteristics 86
Table 93. Dynamic characteristics: low-speed source electrical characteristics 86
Table 94. Register or memory write 87
Table 95. Register read 88
Table 96. Register access 89
Table 97. Memory read 89
Table 98. DMA read (single cycle) 90
Table 99. DMA write (single cycle) 91
Table 100. DMA read (multi-cycle burst) 92
Table 101. DMA write (multi-cycle burst) 93
Table 102. SnPb eutectic process (from J-STD-020C) . . 96
Table 103. Lead-free process (from J-STD-020C) 96
Table 104. Abbreviations 101
Table 105. Revision history 103

25. Figures

Fig 1. Block diagram 3
Fig 2. Pin configuration (LQFP128); top view 4
Fig 3. Internal hub 12
Fig 4. SAF1760 clock scheme 12
Fig 5. Memory segmentation and access block diagram 15
Fig 6. Adjusting analog overcurrent detection limit (optional) 23
Fig 7. SAF1760 power supply connection 24
Fig 8. Most commonly used power supply connection . 24
Fig 9. Hybrid mode 25
Fig 10. Internal power-on reset timing 26
Fig 11. Clock with respect to the external power-on reset 26
Fig 12. NextPTD traversal rule 56
Fig 13. USB source differential data-to-EOP transition skew and EOP width 86
Fig 14. Register or memory write 87
Fig 15. Register read 88
Fig 16. Register access 88
Fig 17. Memory read 89
Fig 18. DMA read (single cycle) 90
Fig 19. DMA write (single cycle) 91
Fig 20. DMA read (multi-cycle burst) 92
Fig 21. DMA write (multi-cycle burst) 93
Fig 22. Package outline SOT425-1 (LQFP128) 94
Fig 23. Temperature profiles for large and small components 97

26. Contents

1 General description 1
2 Features and benefits 1
3 Applications 2
4 Ordering information 2
5 Block diagram 3
6 Pinning information 4
6.1 Pinning 4
6.2 Pin description 4
7 Functional description 11
7.1 SAF1760 internal architecture: advanced NXP slave host controller and hub 11
7.1.1 Internal clock scheme and port selection 12
7.2 Host controller buffer memory block 13
7.2.1 General considerations 13
7.2.2 Structure of the SAF1760 host controller memory 14
7.3 Accessing the SAF1760 host controller memory: PIO and DMA 16
7.3.1 PIO mode access, memory read cycle 16
7.3.2 PIO mode access, memory write cycle 17
7.3.3 PIO mode access, register read cycle 17
7.3.4 PIO mode access, register write cycle 17
7.3.5 DMA mode, read and write operations 17
7.4 Interrupts 19
7.5 Phase-Locked Loop (PLL) clock multiplier . . . 21
7.6 Power management 21
7.7 Overcurrent detection 22
7.8 Power supply 24
7.8.1 Hybrid mode 25
7.9 Power-On Reset (POR) 26
8 Registers 27
8.1 EHCI capability registers 28
8.1.1 CAPLENGTH register 28
8.1.2 HCIVERSION register 28
8.1.3 HCSPARAMS register 29
8.1.4 HCCPARAMS register 30
8.2 EHCI operational registers 31
8.2.1 USBCMD register 31
8.2.2 USBSTS register 32
8.2.3 USBINTR register 33
8.2.4 FRINDEX register 33
8.2.5 CONFIGFLAG register 34
8.2.6 PORTSC1 register 34
8.2.7 ISO PTD Done Map register 36
8.2.8 ISO PTD Skip Map register 36
8.2.9 ISO PTD Last PTD register 36
8.2.10 INT PTD Done Map register 37
8.2.11 INT PTD Skip Map register 37
8.2.12 INT PTD Last PTD register 37
8.2.13 ATL PTD Done Map register 38
8.2.14 ATL PTD Skip Map register 38
8.2.15 ATL PTD Last PTD register 38
8.3 Configuration registers 39
8.3.1 HW Mode Control register 39
8.3.2 Chip ID register 40
8.3.3 Scratch register 40
8.3.4 SW Reset register 41
8.3.5 DMA Configuration register 42
8.3.6 Buffer Status register 43
8.3.7 ATL Done Timeout register 44
8.3.8 Memory register 44
8.3.9 Edge Interrupt Count register 45
8.3.10 DMA Start Address register 46
8.3.11 Power-Down Control register 47
8.3.12 Port 1 Control register 49
8.4 Interrupt registers 50
8.4.1 Interrupt register 50
8.4.2 Interrupt Enable register 52
8.4.3 ISO IRQ Mask OR register 53
8.4.4 INT IRQ Mask OR register 53
8.4.5 ATL IRQ Mask OR register 53
8.4.6 ISO IRQ Mask AND register 54
8.4.7 INT IRQ Mask AND register 54
8.4.8 ATL IRQ Mask AND register 54
9 Proprietary Transfer Descriptor (PTD) 55
9.1 High-speed bulk IN and OUT 56
9.2 High-speed isochronous IN and OUT 60
9.3 High-speed interrupt IN and OUT 64
9.4 Start and complete split for bulk 68
9.5 Start and complete split for isochronous 72
9.6 Start and complete split for interrupt 76
10 Power consumption 81
11 Limiting values 82
12 Recommended operating conditions 82
13 Static characteristics 83
14 Dynamic characteristics 85
14.1 PIO timing 87
14.1.1 Register or memory write 87
14.1.2 Register read 88
14.1.3 Register access 88
14.1.4 Memory read 89
14.2 DMA timing 90
14.2.1 Single cycle: DMA read 90

continued >>

14.2.2 Single cycle: DMA write 91
14.2.3 Multi-cycle: DMA read. 92
14.2.4 Multi-cycle: DMA write 93
15 Package outline 94
16 Soldering of SMD packages 95
16.1 Introduction to soldering 95
16.2 Wave and reflow soldering 95
16.3 Wave soldering 95
16.4 Reflow soldering 96
17 Appendix 97
17.1 Errata added on 2009-04-23. 97
17.1.1 Problem description 97
17.1.2 Implication 97
17.1.3 Workaround 97
17.1.3.1 Software retry mechanism 97
17.1.3.2 Hardware retry mechanism. 98
17.2 Errata added on 2009-04-23. 98
17.2.1 Problem description 98
17.2.2 Implication 98
17.2.3 Workaround 98
17.3 Errata added on 2009-04-23. 98
17.3.1 Problem description 98
17.3.2 Implication 99
17.3.3 Workaround 99
17.3.3.1 Condition 1 99
17.3.3.2 Condition 2 100
17.3.3.3 Remarks 100
18 Abbreviations. 101
19 Glossary 102
20 References 103
21 Revision history. 103
22 Legal information. 104
22.1 Data sheet status 104
22.2 Definitions. 104
22.3 Disclaimers 104
22.4 Trademarks. 105
23 Contact information. 105
24 Tables 106
25 Figures 108
26 Contents 109

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 19 June 2012
Document identifier: SAF1760

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

NXP:

SAF1760BE/V1,518 SAF1760BE/V1,557