

TAMPERE UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTING AND ELECTRICAL ENGINEERING

DEPARTMENT OF COMPUTER SYSTEMS

HIBI_PE_DMA
SW Reference Manual

Author:
Lasse Lehtonen

Updated:
March 2, 2012

Contents

1	Modules	1
2	Data Structures	1
3	File List	1
4	General Functions	2
4.1	Function Documentation	2
4.1.1	hpd_initialize	2
5	TX Functions	2
5.1	Function Documentation	2
5.1.1	hpd_tx_base_conf	2
5.1.2	hpd_tx_base_conf_gen	3
5.1.3	hpd_tx_send	3
5.1.4	hpd_tx_send_copy	3
5.1.5	hpd_tx_send_copy_gen	3
5.1.6	hpd_tx_send_gen	4
6	RX Packet Functions	4
6.1	Function Documentation	4
6.1.1	hpd_rx_packet_get_conf	4
6.1.2	hpd_rx_packet_get_conf_gen	5
6.1.3	hpd_rx_packet_init	5
6.1.4	hpd_rx_packet_init_gen	5
6.1.5	hpd_rx_packet_poll	6
6.1.6	hpd_rx_packet_poll_gen	6
6.1.7	hpd_rx_packet_read	6
6.1.8	hpd_rx_packet_read_gen	6
6.1.9	hpd_rx_packet_reinit	7
6.1.10	hpd_rx_packet_reinit_gen	7
7	RX Stream Functions	7
7.1	Function Documentation	8
7.1.1	hpd_rx_stream_ack	8
7.1.2	hpd_rx_stream_ack_gen	8
7.1.3	hpd_rx_stream_get_conf	8

7.1.4	hpd_rx_stream_get_conf_gen	9
7.1.5	hpd_rx_stream_init	9
7.1.6	hpd_rx_stream_init_gen	9
7.1.7	hpd_rx_stream_poll	10
7.1.8	hpd_rx_stream_poll_gen	10
7.1.9	hpd_rx_stream_read	10
7.1.10	hpd_rx_stream_read_gen	11
7.1.11	hpd_rx_stream_reinit	11
7.1.12	hpd_rx_stream_reinit_gen	11
8	IRQ Functions	11
8.1	Function Documentation	12
8.1.1	hpd_irq_clear_vector	12
8.1.2	hpd_irq_clear_vector_gen	12
8.1.3	hpd_irq_disable	12
8.1.4	hpd_irq_disable_gen	13
8.1.5	hpd_irq_enable	13
8.1.6	hpd_irq_enable_gen	13
8.1.7	hpd_irq_get_vector	13
8.1.8	hpd_irq_get_vector_gen	13
8.1.9	hpd_irq_packet_ack	13
8.1.10	hpd_irq_packet_ack_gen	14
8.1.11	hpd_irq_stream_ack	14
8.1.12	hpd_irq_stream_ack_gen	14
9	HPD_iface Struct Reference	14
9.1	Detailed Description	14
9.2	Field Documentation	15
9.2.1	base_address	15
9.2.2	n_packet_channels	15
9.2.3	n_stream_channels	15
9.2.4	rx_packets	15
9.2.5	rx_streams	15
9.2.6	tx_base_address	15
9.2.7	tx_buffer_bytes	15
9.2.8	tx_hibi_address	15
9.2.9	tx_hibi_command	15

10 HPD_rx_packet Struct Reference	15
10.1 Detailed Description	16
10.2 Field Documentation	16
10.2.1 rx_base_address	16
10.2.2 rx_buffer_bytes	16
10.2.3 rx_hibi_address	16
11 HPD_rx_stream Struct Reference	16
11.1 Detailed Description	16
11.2 Field Documentation	17
11.2.1 rx_base_address	17
11.2.2 rx_buffer_bytes	17
11.2.3 rx_hibi_address	17
11.2.4 rx_read_words	17
12 hpd_config.h File Reference	17
12.1 Detailed Description	18
12.2 Variable Documentation	18
12.2.1 hpd_ifaces	18
13 hpd_functions.h File Reference	18
13.1 Detailed Description	20
14 hpd_macros.h File Reference	21
14.1 Detailed Description	22
14.2 Define Documentation	22
14.2.1 HPD_CLEAR_IRQ_CHAN	22
14.2.2 HPD_CLEAR_IRQ_REG	22
14.2.3 HPD_GET_CONF_REG	22
14.2.4 HPD_GET_IRQ_REG	22
14.2.5 HPD_IRQ_DIS	23
14.2.6 HPD_IRQ_ENA	23
14.2.7 HPD_RX_GET_WORDS	23
14.2.8 HPD_RX_HIBI_ADDR	23
14.2.9 HPD_RX_HIBI_DATA	23
14.2.10 HPD_RX_INIT	23
14.2.11 HPD_RX_MEM_ADDR	23
14.2.12 HPD_RX_WORDS	23

14.2.13 HPD_TX_CMD	23
14.2.14 HPD_TX_CMD_READ	23
14.2.15 HPD_TX_CMD_WRITE	23
14.2.16 HPD_TX_CMD_WRITE_MSG	24
14.2.17 HPD_TX_GET_DONE	24
14.2.18 HPD_TX_HIBI_ADDR	24
14.2.19 HPD_TX_MEM_ADDR	24
14.2.20 HPD_TX_START	24
14.2.21 HPD_TX_WORDS	24

Module Index

1 Modules

Here is a list of all modules:

General Functions	2
TX Functions	2
RX Packet Functions	4
RX Stream Functions	7
IRQ Functions	11

Data Structure Index

2 Data Structures

Here are the data structures with brief descriptions:

HPD_iface (Struct holding information about HIBI_PE_DMA components)	14
HPD_rx_packet (Holds configuration information of one RX packet channel)	15
HPD_rx_stream (Holds configuration information of one RX stream channel)	16

File Index

3 File List

Here is a list of all documented files with brief descriptions:

hpd_config.h (HIBI_PE_DMA configuration file)	17
hpd_functions.h (Platform independed C functions)	18



Module Documentation

4 General Functions

Functions

- void [hpd_initialize](#) ()
Initializes HIBI_PE_DMA component according to hpd_config.c.

4.1 Function Documentation

4.1.1 void [hpd_initialize](#) ()

Initializes HIBI_PE_DMA component according to hpd_config.c.

Reads settings from the data structure defined in hpd_config.c and configures memory address, buffer size and hibi address registers on all HIBI_PE_DMA interfaces but doesn't start RX channel receiving.

5 TX Functions

Functions

- void [hpd_tx_base_conf_gen](#) (int base, int words, int iface)
Configures HIBI_PE_DMA's TX base settings for interface iface.
- void [hpd_tx_base_conf](#) (int base, int words)
Configures HIBI_PE_DMA's TX base settings for default interface.
- void [hpd_tx_send_gen](#) (int daddr, int words, int haddr, int iface)
Send packet through HIBI_PE_DMA interface iface. Data must be in a memory accessible by HIBI_PE_DMA.
- void [hpd_tx_send](#) (int daddr, int words, int haddr)
Send packet through HIBI_PE_DMA default interface. Data must be in a memory accessible by HIBI_PE_DMA.
- void [hpd_tx_send_copy_gen](#) (int daddr, int words, int haddr, int iface)
Send packet through HIBI_PE_DMA interface iface. Copies the data first to the memory accessible by HIBI_PE_DMA.
- void [hpd_tx_send_copy](#) (int daddr, int words, int haddr)
Send packet through HIBI_PE_DMA default interface. Copies the data first to the memory accessible by HIBI_PE_DMA.

5.1 Function Documentation

5.1.1 void [hpd_tx_base_conf](#) (int base, int words)

Configures HIBI_PE_DMA's TX base settings for default interface.

Parameters

in	base	TX buffer's base address.
in	words	Size of the TX buffer in words.

5.1.2 void hpd_tx_base_conf_gen (int base, int words, int iface)

Configures HIBI_PE_DMA's TX base settings for interface *iface*.

Parameters

in	<i>base</i>	TX buffer's base address.
in	<i>words</i>	Size of the TX buffer in words.

5.1.3 void hpd_tx_send (int daddr, int words, int haddr)

Send packet through HIBI_PE_DMA default interface. Data must be in a memory accessible by HIBI_PE_DMA.

Parameters

in	<i>daddr</i>	Beginning address for the data to be sent.
in	<i>words</i>	How many words to sent.
in	<i>haddr</i>	Where to send the packet on HIBI.

This function configures registers on HIBI_PE_DMA and starts the transfer. It sends *words* amount of words starting from *daddr*. Address *daddr* must be accessible by HIBI_PE_DMA.

5.1.4 void hpd_tx_send_copy (int daddr, int words, int haddr)

Send packet through HIBI_PE_DMA default interface. Copies the data first to the memory accessible by HIBI_PE_DMA.

Parameters

in	<i>daddr</i>	Beginning address for the data to be sent.
in	<i>words</i>	How many words to sent.
in	<i>haddr</i>	Where to send the packet on HIBI.

This function configures registers on HIBI_PE_DMA and starts the transfer. It sends *words* amount of words starting from *daddr*. This function copies the data from *daddr* to the shared memory accessible by HIBI_PE_DMA before sending. If packet is larger than TX buffer it's sent in pieces.

5.1.5 void hpd_tx_send_copy_gen (int daddr, int words, int haddr, int iface)

Send packet through HIBI_PE_DMA interface *iface*. Copies the data first to the memory accessible by HIBI_PE_DMA.

Parameters

in	<i>daddr</i>	Beginning address for the data to be sent.
in	<i>words</i>	How many words to sent.
in	<i>haddr</i>	Where to send the packet on HIBI.
in	<i>iface</i>	Which interface to use.

This function configures registers on HIBI_PE_DMA and starts the transfer. It sends *words* amount of words starting from *daddr*. This function copies the data from *daddr* to the shared memory accessible by HIBI_PE_DMA.

DMA before sending. If packet is larger than TX buffer it's sent in pieces.

5.1.6 void `hpd_tx_send_gen` (int *daddr*, int *words*, int *haddr*, int *iface*)

Send packet through HIBI_PE_DMA interface *iface*. Data must be in a memory accessible by HIBI_PE_DMA.

Parameters

in	<i>daddr</i>	Beginning address for the data to be sent.
in	<i>words</i>	How many words to sent.
in	<i>haddr</i>	Where to send the packet on HIBI.
in	<i>iface</i>	Which interface to use.

This function configures registers on HIBI_PE_DMA and starts the transfer. It sends *words* amount of words starting from *daddr*. Address *daddr* must be accessible by HIBI_PE_DMA.

6 RX Packet Functions

Functions

- void `hpd_rx_packet_init_gen` (int *chan*, int *daddr*, int *words*, int *haddr*, int *iface*)
Initialize packet channel for reception on interface iface.
- void `hpd_rx_packet_init` (int *chan*, int *daddr*, int *words*, int *haddr*)
Initialize packet channel for reception on default interface.
- void `hpd_rx_packet_reinit_gen` (int *chan*, int *iface*)
Reinitializes packet channel with previous settings on interface iface.
- void `hpd_rx_packet_reinit` (int *chan*)
Reinitializes packet channel with previous settings on default interface.
- void `hpd_rx_packet_read_gen` (int *chan*, void **buffer*, int *iface*)
Read RX packet from channel chan to buffer without checking the channel's status from interface iface.
- void `hpd_rx_packet_read` (int *chan*, void **buffer*)
Read RX packet from channel chan to buffer without checking the channel's status from default interface.
- void `hpd_rx_packet_get_conf_gen` (int *chan*, int **rx_base*, int **rx_bytes*, int **rx_haddr*, int *iface*)
Gets the configuration information of packet channel chan from interface iface.
- void `hpd_rx_packet_get_conf` (int *chan*, int **rx_base*, int **rx_bytes*, int **rx_haddr*)
Gets the configuration information of packet channel chan from default interface.
- int `hpd_rx_packet_poll_gen` (int *chan*, int *iface*)
Poll RX packet channel chan from interface iface to check if it's received all data.
- int `hpd_rx_packet_poll` (int *chan*)
Poll RX packet channel chan from default interface to check if it's received all data.

6.1 Function Documentation

6.1.1 void `hpd_rx_packet_get_conf` (int *chan*, int * *rx_base*, int * *rx_bytes*, int * *rx_haddr*)

Gets the configuration information of packet channel *chan* from default interface.

Parameters

in	<i>chan</i>	Which channel to read.
out	<i>rx_base</i>	Base address where channel stores the data.
out	<i>rx_bytes</i>	Size of the RX buffer in bytes.
out	<i>rx_haddr</i>	HIBI address that this channel is listening to.

Gets configuration information from one RX packet channel. Use NULL to disable returning particular parameter.

6.1.2 void hpd_rx_packet_get_conf_gen (int *chan*, int * *rx_base*, int * *rx_bytes*, int * *rx_haddr*, int *iface*)

Gets the configuration information of packet channel *chan* from interface *iface*.

Parameters

in	<i>chan</i>	Which channel to read.
out	<i>rx_base</i>	Base address where channel stores the data.
out	<i>rx_bytes</i>	Size of the RX buffer in bytes.
out	<i>rx_haddr</i>	HIBI address that this channel is listening to.
in	<i>iface</i>	Which interface to read from.

Gets configuration information from one RX packet channel. Use NULL to disable returning particular parameter.

6.1.3 void hpd_rx_packet_init (int *chan*, int *daddr*, int *words*, int *haddr*)

Initialize packet channel for reception on default interface.

Parameters

in	<i>chan</i>	Which channel.
in	<i>daddr</i>	Where to store the packet.
in	<i>words</i>	How many to expect.
in	<i>haddr</i>	Hibi address to listen to.

6.1.4 void hpd_rx_packet_init_gen (int *chan*, int *daddr*, int *words*, int *haddr*, int *iface*)

Initialize packet channel for reception on interface *iface*.

Parameters

in	<i>chan</i>	Which channel.
in	<i>daddr</i>	Where to store the packet.
in	<i>words</i>	How many to expect.
in	<i>haddr</i>	Hibi address to listen to.
in	<i>iface</i>	Which interface to use.

6.1.5 int hpd_rx_packet_poll (int *chan*)

Poll RX packet channel *chan* from default interface to check if it's received all data.

Parameters

in	<i>chan</i>	Which channel to check.
----	-------------	-------------------------

Returns

One if packet channel has received a full packet, zero otherwise.

Warning

Undefined behavior if used for stream or uninitialized channels.

6.1.6 int hpd_rx_packet_poll_gen (int *chan*, int *iface*)

Poll RX packet channel *chan* from interface *iface* to check if it's received all data.

Parameters

in	<i>chan</i>	Which channel to check.
in	<i>iface</i>	From which interface.

Returns

One if packet channel has received a packet, zero otherwise.

Warning

Undefined behaviour if used for stream or uninitialized channels.

6.1.7 void hpd_rx_packet_read (int *chan*, void * *buffer*)

Read RX packet from channel *chan* to *buffer* without checking the channel's status from default interface.

Parameters

in	<i>chan</i>	Which channel to read.
out	<i>buffer</i>	Place to store the data.

This function copies channels RX buffer without checking whether or not it has received anything. Used when it's already known that there is whole packet available e.g. through interrupts or polling.

6.1.8 void hpd_rx_packet_read_gen (int *chan*, void * *buffer*, int *iface*)

Read RX packet from channel *chan* to *buffer* without checking the channel's status from interface *iface*.

Parameters

in	<i>chan</i>	Which channel to read.
out	<i>buffer</i>	Place to store the data.

in	iface	Interface to use.
----	-------	-------------------

This function copies channels RX buffer without checking whether or not it has received anything. Used when it's already known that there is whole packet available e.g. through interrupts or polling.

6.1.9 void hpd_rx_packet_reinit (int chan)

Reinitializes packet channel with previous settings on default interface.

Warning

Undefined behaviour if used for stream channels.

Parameters

in	chan	Which channel.
----	------	----------------

6.1.10 void hpd_rx_packet_reinit_gen (int chan, int iface)

Reinitializes packet channel with previous settings on interface *iface*.

Parameters

in	chan	Which channel.
in	iface	Which interface to use.

7 RX Stream Functions

Functions

- void [hpd_rx_stream_init_gen](#) (int chan, int daddr, int words, int haddr, int iface)
Initialize stream channel for reception on interface iface.
- void [hpd_rx_stream_init](#) (int chan, int daddr, int words, int haddr)
Initialize stream channel for reception on default interface.
- void [hpd_rx_stream_reinit_gen](#) (int chan, int iface)
Reinitializes stream channel with previous settings on interface iface.
- void [hpd_rx_stream_reinit](#) (int chan)
Reinitializes stream channel with previous settings on default interface.
- void [hpd_rx_stream_get_conf_gen](#) (int chan, int *rx_base, int *rx_bytes, int *rx_haddr, int iface)
Gets the configuration information of stream channel chan from interface iface.
- void [hpd_rx_stream_get_conf](#) (int chan, int *rx_base, int *rx_bytes, int *rx_haddr)
Gets the configuration information of stream channel chan from default interface.
- int [hpd_rx_stream_poll_gen](#) (int chan, int iface)
Poll if RX stream channel chan on interface iface has received any words.
- int [hpd_rx_stream_poll](#) (int chan)
Poll if RX stream channel chan on default interface has received any words.
- void [hpd_rx_stream_read_gen](#) (int chan, int words, int *buffer, int iface)

Read words amount of words from channel chan on interface iface and store them to buffer.

- void `hpd_rx_stream_read` (int chan, int words, int *buffer)

Read words amount of words from channel chan on default interface store them to buffer.

- void `hpd_rx_stream_ack_gen` (int chan, int words, int iface)

Acknowledgege to stream channel on interface iface that words amount of words have been read from it.

- void `hpd_rx_stream_ack` (int chan, int words)

Acknowledgege to stream channel on default interface that words amount of words have been read from it.

7.1 Function Documentation

7.1.1 void `hpd_rx_stream_ack` (int chan, int words)

Acknowledgege to stream channel on default interface that *words* amount of words have been read from it.

Parameters

in	<i>chan</i>	Which channel to check.
in	<i>words</i>	How many words was read.
in	<i>iface</i>	From which interface.

This functions is used to tell HIBI_PE_DMA that some words have been read from its RX stream channel so it can use that buffer again for receiving data.

Warning

Undefined behaviour if used for packet channels.

7.1.2 void `hpd_rx_stream_ack_gen` (int chan, int words, int iface)

Acknowledgege to stream channel on interface *iface* that *words* amount of words have been read from it.

Parameters

in	<i>chan</i>	Which channel to check.
in	<i>words</i>	How many words was read.
in	<i>iface</i>	From which interface.

This functions is used to tell HIBI_PE_DMA that some words have been read from its RX stream channel so it can use that buffer again for receiving data.

Warning

Undefined behaviour if used for packet channels.

7.1.3 void `hpd_rx_stream_get_conf` (int chan, int * rx_base, int * rx_bytes, int * rx_haddr)

Gets the configuration information of stream channel *chan* from default interface.

Parameters

in	<i>chan</i>	Which channel to read.
out	<i>rx_base</i>	Base address where channel stores the data.
out	<i>rx_bytes</i>	Size of the RX buffer in bytes.
out	<i>rx_haddr</i>	HIBI address that this channel is listening to.

Gets configuration information from one RX stream channel. Use NULL to disable returning particular parameter.

7.1.4 void hpd_rx_stream_get_conf_gen (int *chan*, int * *rx_base*, int * *rx_bytes*, int * *rx_haddr*, int *iface*)

Gets the configuration information of stream channel *chan* from interface *iface*.

Parameters

in	<i>chan</i>	Which channel to read.
out	<i>rx_base</i>	Base address where channel stores the data.
out	<i>rx_bytes</i>	Size of the RX buffer in bytes.
out	<i>rx_haddr</i>	HIBI address that this channel is listening to.
in	<i>iface</i>	Which interface to read from.

Gets configuration information from one RX stream channel. Use NULL to disable returning particular parameter.

7.1.5 void hpd_rx_stream_init (int *chan*, int *daddr*, int *words*, int *haddr*)

Initialize stream channel for reception on default interface.

Parameters

in	<i>chan</i>	Which channel.
in	<i>daddr</i>	Starting address of the buffer.
in	<i>words</i>	Size of the receive buffer.
in	<i>haddr</i>	Hibi address to listen to.

7.1.6 void hpd_rx_stream_init_gen (int *chan*, int *daddr*, int *words*, int *haddr*, int *iface*)

Initialize stream channel for reception on interface *iface*.

Parameters

in	<i>chan</i>	Which channel.
in	<i>daddr</i>	Starting address of the buffer.
in	<i>words</i>	Size of the receive buffer.
in	<i>haddr</i>	Hibi address to listen to.
in	<i>iface</i>	Which interface to use.

7.1.7 int hpd_rx_stream_poll (int chan)

Poll if RX stream channel *chan* on default interface has received any words.

Parameters

in	<i>chan</i>	Which channel to check.
----	-------------	-------------------------

Returns

Number of received words.

Warning

Undefined behavior if used on packet or uninitialized channel.

7.1.8 int hpd_rx_stream_poll_gen (int chan, int iface)

Poll if RX stream channel *chan* on interface *iface* has received any words.

Parameters

in	<i>chan</i>	Which channel to check.
in	<i>iface</i>	From which interface.

Returns

Number of received words.

Warning

Undefined behavior if used on packet or uninitialized channel.

7.1.9 void hpd_rx_stream_read (int chan, int words, int * buffer)

Read *words* amount of words from channel *chan* on default interface store them to *buffer*.

Parameters

in	<i>chan</i>	Which channel to check.
in	<i>words</i>	How many words to read.
in	<i>buffer</i>	Where to store read words.

Reads from a stream channel and acknowledges the channel about the amount read. This function should be used only when it's known that the channel has received at least *words* words of data.

Warning

No need to use ack function after this function because acknowledging is handled in this function. Undefined behaviour if used for packet channels.

7.1.10 void hpd_rx_stream_read_gen (int chan, int words, int * buffer, int iface)

Read *words* amount of words from channel *chan* on interface *iface* and store them to *buffer*.

Parameters

in	<i>chan</i>	Which channel to check.
in	<i>words</i>	How many words to read.
in	<i>buffer</i>	Where to store read words.
in	<i>iface</i>	From which interface.

Reads from a stream channel and acknowledges the channel about the amount read. This function should be used only when it's known that the channel has received at least *words* words of data.

Warning

Don't use ack function after this function because acknowledging is handled in this function. Undefined behaviour if used for packet channels.

7.1.11 void hpd_rx_stream_reinit (int chan)

Reinitializes stream channel with previous settings on default interface.

Warning

Undefined behaviour if used for packet channels.

Parameters

in	<i>chan</i>	Which channel.
----	-------------	----------------

7.1.12 void hpd_rx_stream_reinit_gen (int chan, int iface)

Reinitializes stream channel with previous settings on interface *iface*.

Parameters

in	<i>chan</i>	Which channel.
in	<i>iface</i>	Which interface to use.

8 IRQ Functions

Functions

- void [hpd_irq_enable_gen](#) (int iface)
Enable HIBI_PE_DMA interrupts on interface iface.
- void [hpd_irq_enable](#) ()
Enable HIBI_PE_DMA interrupts on default interface.
- void [hpd_irq_disable_gen](#) (int iface)

- Disable HIBI_PE_DMA interrupts on interface iface.*

 - void `hpd_irq_disable` ()
- Disable HIBI_PE_DMA interrupts on default interface.*

 - void `hpd_irq_packet_ack_gen` (int chan, int iface)

Acknowledge IRQ for packet channel chan on interface iface.

 - void `hpd_irq_packet_ack` (int chan)

Acknowledge IRQ for packet channel chan on default interface.

 - void `hpd_irq_stream_ack_gen` (int chan, int iface)

Acknowledge IRQ for stream channel chan on interface iface.

 - void `hpd_irq_stream_ack` (int chan)

Acknowledge IRQ for stream channel chan on default interface.

 - int `hpd_irq_get_vector_gen` (int iface)

Returns IRQ vector from HIBI_PE_DMA interface iface.

 - int `hpd_irq_get_vector` ()

Returns IRQ vector from HIBI_PE_DMA default interface.

 - void `hpd_irq_clear_vector_gen` (int mask, int iface)

Clear interrupts by mask on interface iface.

 - void `hpd_irq_clear_vector` (int mask)

Clear interrupts by mask on default interface.

8.1 Function Documentation

8.1.1 void `hpd_irq_clear_vector` (int *mask*)

Clear interrupts by mask on default interface.

Every bit high in the mask will clear the corresponding interrupt source.

Parameters

in	<i>mask</i>	Clear interrupt mask.
in	<i>iface</i>	Which interface.

8.1.2 void `hpd_irq_clear_vector_gen` (int *mask*, int *iface*)

Clear interrupts by mask on interface *iface*.

Every bit high in the mask will clear the corresponding interrupt source.

Parameters

in	<i>mask</i>	Clear interrupt mask.
in	<i>iface</i>	Which interface.

8.1.3 void `hpd_irq_disable` ()

Disable HIBI_PE_DMA interrupts on default interface.

Disables interrupts.

8.1.4 void hpd_irq_disable_gen (int *iface*)

Disable HIBI_PE_DMA interrupts on interface *iface*.

Parameters

in	<i>iface</i>	Which interface.
----	--------------	------------------

8.1.5 void hpd_irq_enable ()

Enable HIBI_PE_DMA interrupts on default interface.

Enables interrupts.

8.1.6 void hpd_irq_enable_gen (int *iface*)

Enable HIBI_PE_DMA interrupts on interface *iface*.

Parameters

in	<i>iface</i>	Which interface.
----	--------------	------------------

8.1.7 int hpd_irq_get_vector ()

Returns IRQ vector from HIBI_PE_DMA default interface.

Returns

IRQ vector.

8.1.8 int hpd_irq_get_vector_gen (int *iface*)

Returns IRQ vector from HIBI_PE_DMA interface *iface*.

Returns a vector in which a high bit indicates an active interrupt source.

Parameters

in	<i>iface</i>	Which interface.
----	--------------	------------------

Returns

IRQ vector.

8.1.9 void hpd_irq_packet_ack (int *chan*)

Acknowledge IRQ for packet channel *chan* on default interface.

Parameters

in	<i>chan</i>	Channel to acknowledge.
----	-------------	-------------------------

8.1.10 void hpd_irq_packet_ack_gen (int chan, int iface)

Acknowledge IRQ for packet channel *chan* on interface *iface*.

Parameters

in	<i>chan</i>	Channel to acknowledge.
in	<i>iface</i>	Which interface.

8.1.11 void hpd_irq_stream_ack (int chan)

Acknowledge IRQ for stream channel *chan* on default interface.

Parameters

in	<i>chan</i>	Channel to acknowledge.
----	-------------	-------------------------

8.1.12 void hpd_irq_stream_ack_gen (int chan, int iface)

Acknowledge IRQ for stream channel *chan* on interface *iface*.

Parameters

in	<i>chan</i>	Channel to acknowledge.
in	<i>iface</i>	Which interface.

Data Structure Documentation

9 HPD_iface Struct Reference

Struct holding information about HIBI_PE_DMA components.

```
#include <hpd_config.h>
```

Data Fields

- int [base_address](#)
- int [tx_base_address](#)
- int [tx_buffer_bytes](#)
- int [tx_hibi_address](#)
- int [tx_hibi_command](#)
- int [n_stream_channels](#)
- int [n_packet_channels](#)
- [HPD_rx_stream](#) * [rx_streams](#)
- [HPD_rx_packet](#) * [rx_packets](#)

9.1 Detailed Description

Struct holding information about HIBI_PE_DMA components.

9.2 Field Documentation

9.2.1 `int HPD_iface::base_address`

Base address for this HIBI_PE_DMA.

9.2.2 `int HPD_iface::n_packet_channels`

Number of packet channels available.

9.2.3 `int HPD_iface::n_stream_channels`

Number of stream channels available.

9.2.4 `HPD_rx_packet* HPD_iface::rx_packets`

Table of rx packet channels.

9.2.5 `HPD_rx_stream* HPD_iface::rx_streams`

Table of rx stream channels.

9.2.6 `int HPD_iface::tx_base_address`

TX buffer's start on shared mem.

9.2.7 `int HPD_iface::tx_buffer_bytes`

Size of the TX buffer in bytes.

9.2.8 `int HPD_iface::tx_hibi_address`

Target HIBI address for sending.

9.2.9 `int HPD_iface::tx_hibi_command`

HIBI command for sending.

The documentation for this struct was generated from the following file:

- [hpd_config.h](#)

10 HPD_rx_packet Struct Reference

Holds configuration information of one RX packet channel.

```
#include <hpd_config.h>
```

Data Fields

- int [rx_base_address](#)
- int [rx_buffer_bytes](#)
- int [rx_hibi_address](#)

10.1 Detailed Description

Holds configuration information of one RX packet channel.

10.2 Field Documentation

10.2.1 int HPD_rx_packet::rx_base_address

Address of this channel's RX buffer.

10.2.2 int HPD_rx_packet::rx_buffer_bytes

Size of the RX buffer in bytes.

10.2.3 int HPD_rx_packet::rx_hibi_address

HIBI address for receiving data.

The documentation for this struct was generated from the following file:

- [hpd_config.h](#)

11 HPD_rx_stream Struct Reference

Holds configuration information of one RX stream channel.

```
#include <hpd_config.h>
```

Data Fields

- int [rx_base_address](#)
- int [rx_buffer_bytes](#)
- int [rx_hibi_address](#)
- int [rx_read_words](#)

11.1 Detailed Description

Holds configuration information of one RX stream channel.

11.2 Field Documentation

11.2.1 `int HPD_rx_stream::rx_base_address`

Address of this channel's RX buffer.

11.2.2 `int HPD_rx_stream::rx_buffer_bytes`

Size of the RX buffer in bytes.

11.2.3 `int HPD_rx_stream::rx_hibi_address`

HIBI address for receiving data.

11.2.4 `int HPD_rx_stream::rx_read_words`

Current read pointer.

The documentation for this struct was generated from the following file:

- [hpd_config.h](#)

File Documentation

12 `hpd_config.h` File Reference

HIBI_PE_DMA configuration file.

Data Structures

- struct [HPD_rx_stream](#)
Holds configuration information of one RX stream channel.
- struct [HPD_rx_packet](#)
Holds configuration information of one RX packet channel.
- struct [HPD_iface](#)
Struct holding information about HIBI_PE_DMA components.

Defines

- `#define` [NUM_OF_HIBI_PE_DMAS](#)
Total number HIBI_PE_DMA components this processor uses and sees on its memory mapped bus.

Variables

- [HPD_iface](#) `hpd_ifaces` [`NUM_OF_HIBI_PE_DMAS`]

12.1 Detailed Description

HIBI_PE_DMA configuration file.

Author

Lasse Lehtonen

Date

2012-02-16

This file along with its .c file is used to configure HIBI_PE_DMA components for the functions to know where they are. This file should be unique for each processor using HIBI_PE_DMA components.

Copyright

Funbase IP library Copyright (C) 2012 TUT Department of Computer Systems

This file is part of HIBI_PE_DMA

This source file may be used and distributed without restriction provided that this copyright statement is not removed from the file and that any derivative work contains the original copyright notice and the associated disclaimer. This source file is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This source is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this source; if not, download it from <http://www.opencores.org/lgpl.shtml>

12.2 Variable Documentation

12.2.1 HPD_iface hpd_ifaces[NUM_OF_HIBI_PE_DMAS]

Actual interface information is located in the c file.

13 hpd_functions.h File Reference

Platform independent C functions.

Functions

- void [hpd_initialize](#) ()
Initializes HIBI_PE_DMA component according to hpd_config.c.
- void [hpd_tx_base_conf_gen](#) (int base, int words, int iface)
Configures HIBI_PE_DMA's TX base settings for interface iface.
- void [hpd_tx_base_conf](#) (int base, int words)

- Configures HIBI_PE_DMA's TX base settings for default interface.*
- void `hpd_tx_send_gen` (int daddr, int words, int haddr, int iface)
Send packet through HIBI_PE_DMA interface iface. Data must be in a memory accessible by HIBI_PE_DMA.
- void `hpd_tx_send` (int daddr, int words, int haddr)
Send packet through HIBI_PE_DMA default interface. Data must be in a memory accessible by HIBI_PE_DMA.
- void `hpd_tx_send_copy_gen` (int daddr, int words, int haddr, int iface)
Send packet through HIBI_PE_DMA interface iface. Copies the data first to the memory accessible by HIBI_PE_DMA.
- void `hpd_tx_send_copy` (int daddr, int words, int haddr)
Send packet through HIBI_PE_DMA default interface. Copies the data first to the memory accessible by HIBI_PE_DMA.
- void `hpd_rx_packet_init_gen` (int chan, int daddr, int words, int haddr, int iface)
Initialize packet channel for reception on interface iface.
- void `hpd_rx_packet_init` (int chan, int daddr, int words, int haddr)
Initialize packet channel for reception on default interface.
- void `hpd_rx_packet_reinit_gen` (int chan, int iface)
Reinitializes packet channel with previous settings on interface iface.
- void `hpd_rx_packet_reinit` (int chan)
Reinitializes packet channel with previous settings on default interface.
- void `hpd_rx_packet_read_gen` (int chan, void *buffer, int iface)
Read RX packet from channel chan to buffer without checking the channel's status from interface iface.
- void `hpd_rx_packet_read` (int chan, void *buffer)
Read RX packet from channel chan to buffer without checking the channel's status from default interface.
- void `hpd_rx_packet_get_conf_gen` (int chan, int *rx_base, int *rx_bytes, int *rx_haddr, int iface)
Gets the configuration information of packet channel chan from interface iface.
- void `hpd_rx_packet_get_conf` (int chan, int *rx_base, int *rx_bytes, int *rx_haddr)
Gets the configuration information of packet channel chan from default interface.
- int `hpd_rx_packet_poll_gen` (int chan, int iface)
Poll RX packet channel chan from interface iface to check if it's received all data.
- int `hpd_rx_packet_poll` (int chan)
Poll RX packet channel chan from default interface to check if it's received all data.
- void `hpd_rx_stream_init_gen` (int chan, int daddr, int words, int haddr, int iface)
Initialize stream channel for reception on interface iface.
- void `hpd_rx_stream_init` (int chan, int daddr, int words, int haddr)
Initialize stream channel for reception on default interface.
- void `hpd_rx_stream_reinit_gen` (int chan, int iface)
Reinitializes stream channel with previous settings on interface iface.
- void `hpd_rx_stream_reinit` (int chan)
Reinitializes stream channel with previous settings on default interface.
- void `hpd_rx_stream_get_conf_gen` (int chan, int *rx_base, int *rx_bytes, int *rx_haddr, int iface)
Gets the configuration information of stream channel chan from interface iface.
- void `hpd_rx_stream_get_conf` (int chan, int *rx_base, int *rx_bytes, int *rx_haddr)
Gets the configuration information of stream channel chan from default interface.
- int `hpd_rx_stream_poll_gen` (int chan, int iface)
Poll if RX stream channel chan on interface iface has received any words.
- int `hpd_rx_stream_poll` (int chan)
Poll if RX stream channel chan on default interface has received any words.

- void `hpd_rx_stream_read_gen` (int chan, int words, int *buffer, int iface)
Read words amount of words from channel chan on interface iface and store them to buffer.
- void `hpd_rx_stream_read` (int chan, int words, int *buffer)
Read words amount of words from channel chan on default interface store them to buffer.
- void `hpd_rx_stream_ack_gen` (int chan, int words, int iface)
Acknowledge to stream channel on interface iface that words amount of words have been read from it.
- void `hpd_rx_stream_ack` (int chan, int words)
Acknowledge to stream channel on default interface that words amount of words have been read from it.
- void `hpd_irq_enable_gen` (int iface)
Enable HIBI_PE_DMA interrupts on interface iface.
- void `hpd_irq_enable` ()
Enable HIBI_PE_DMA interrupts on default interface.
- void `hpd_irq_disable_gen` (int iface)
Disable HIBI_PE_DMA interrupts on interface iface.
- void `hpd_irq_disable` ()
Disable HIBI_PE_DMA interrupts on default interface.
- void `hpd_irq_packet_ack_gen` (int chan, int iface)
Acknowledge IRQ for packet channel chan on interface iface.
- void `hpd_irq_packet_ack` (int chan)
Acknowledge IRQ for packet channel chan on default interface.
- void `hpd_irq_stream_ack_gen` (int chan, int iface)
Acknowledge IRQ for stream channel chan on interface iface.
- void `hpd_irq_stream_ack` (int chan)
Acknowledge IRQ for stream channel chan on default interface.
- int `hpd_irq_get_vector_gen` (int iface)
Returns IRQ vector from HIBI_PE_DMA interface iface.
- int `hpd_irq_get_vector` ()
Returns IRQ vector from HIBI_PE_DMA default interface.
- void `hpd_irq_clear_vector_gen` (int mask, int iface)
Clear interrupts by mask on interface iface.
- void `hpd_irq_clear_vector` (int mask)
Clear interrupts by mask on default interface.

13.1 Detailed Description

Platform independent C functions.

Author

Lasse Lehtonen

Date

2012-02-14

This file introduces a set of functions for handling HIBI_PE_DMA more easily. Functions are written in platform independent C using macros defined in `hdp_macros.h`.

Warning

These functions store the HIBI_PE_DMA configuration information in an internal structure to avoid unnecessary configuration. This implies that all communication with HIBI_PE_DMA must be done with these functions or their behaviour may become undefined.

Copyright

Funbase IP library Copyright (C) 2012 TUT Department of Computer Systems

This file is part of HIBI_PE_DMA

This source file may be used and distributed without restriction provided that this copyright statement is not removed from the file and that any derivative work contains the original copyright notice and the associated disclaimer. This source file is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This source is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this source; if not, download it from <http://www.opencores.org/lgpl.shtml>

14 hpd_macros.h File Reference

Platform independent C macros.

Defines

- #define `HPD_RX_INIT`(chan, base)
- #define `HPD_GET_CONF_REG`(var, base)
- #define `HPD_IRQ_ENA`(base)
- #define `HPD_IRQ_DIS`(base)
- #define `HPD_TX_START`(base)
- #define `HPD_TX_GET_DONE`(var, base)
- #define `HPD_GET_IRQ_REG`(var, base)
- #define `HPD_CLEAR_IRQ_CHAN`(chan, base)
- #define `HPD_CLEAR_IRQ_REG`(mask, base)
- #define `HPD_TX_MEM_ADDR`(addr, base)
- #define `HPD_TX_WORDS`(words, base)
- #define `HPD_TX_CMD`(comm, base)
- #define `HPD_TX_CMD_WRITE`(base)
- #define `HPD_TX_CMD_READ`(base)
- #define `HPD_TX_CMD_WRITE_MSG`(base)
- #define `HPD_TX_HIBI_ADDR`(haddr, base)
- #define `HPD_RX_HIBI_DATA`(var, base)
- #define `HPD_RX_MEM_ADDR`(chan, addr, base)
- #define `HPD_RX_GET_WORDS`(var, chan, base)
- #define `HPD_RX_WORDS`(chan, words, base)
- #define `HPD_RX_HIBI_ADDR`(chan, haddr, base)

14.1 Detailed Description

Platform independent C macros.

Author

Lasse Lehtonen

Date

2012-02-28

This file introduces necessary platform independent macros for configuring HIBI_PE_DMA.

Copyright

Funbase IP library Copyright (C) 2012 TUT Department of Computer Systems

This file is part of HIBI_PE_DMA

This source file may be used and distributed without restriction provided that this copyright statement is not removed from the file and that any derivative work contains the original copyright notice and the associated disclaimer. This source file is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This source is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this source; if not, download it from <http://www.opencores.org/lgpl.shtml>

14.2 Define Documentation

14.2.1 #define HPD_CLEAR_IRQ_CHAN(*chan*, *base*)

Clear interrupts for channel *chan*.

14.2.2 #define HPD_CLEAR_IRQ_REG(*mask*, *base*)

Clear interrupts for high bits in mask.

14.2.3 #define HPD_GET_CONF_REG(*var*, *base*)

Return configuration register in *var*.

14.2.4 #define HPD_GET_IRQ_REG(*var*, *base*)

Return interrupt register in *var*.

14.2.5 `#define HPD_IRQ_DIS(base)`

Disable interrupts.

14.2.6 `#define HPD_IRQ_ENA(base)`

Enable interrupts.

14.2.7 `#define HPD_RX_GET_WORDS(var, chan, base)`

Return the amount of words stream channel has received since last acknowledge in *var*.

14.2.8 `#define HPD_RX_HIBI_ADDR(chan, haddr, base)`

Set channel *chan* to receive from HIBI address *haddr*.

14.2.9 `#define HPD_RX_HIBI_DATA(var, base)`

Returns current value on hibi bus in *var*.

14.2.10 `#define HPD_RX_INIT(chan, base)`

Set channel *chan* to receive data with previously set configuration

14.2.11 `#define HPD_RX_MEM_ADDR(chan, addr, base)`

Set channel *chan* to write received data to memory starting from *addr*.

14.2.12 `#define HPD_RX_WORDS(chan, words, base)`

Set channel *chan* to receive words amount of words.

14.2.13 `#define HPD_TX_CMD(comm, base)`

Set HIBI command to use for the transaction

14.2.14 `#define HPD_TX_CMD_READ(base)`

Use normal READ HIBI command for sending.

14.2.15 `#define HPD_TX_CMD_WRITE(base)`

Use normal WRITE HIBI command for sending.

14.2.16 #define HPD_TX_CMD_WRITE_MSG(*base*)

Use message priority WRITE HIBI command for sending.

14.2.17 #define HPD_TX_GET_DONE(*var, base*)

Poll if previous transfer has completed, var is 1 if true.

14.2.18 #define HPD_TX_HIBI_ADDR(*haddr, base*)

Set target HIBI address.

14.2.19 #define HPD_TX_MEM_ADDR(*addr, base*)

Set memory address where the packet to be sent begins.

14.2.20 #define HPD_TX_START(*base*)

Start the transfer with previously set configuration.

14.2.21 #define HPD_TX_WORDS(*words, base*)

Set amount of words to be sent.

Index

- base_address
 - HPD_iface, 14
- General Functions, 1
 - hpd_initialize, 2
- HPD_CLEAR_IRQ_CHAN
 - hpd_macros.h, 22
- HPD_CLEAR_IRQ_REG
 - hpd_macros.h, 22
- hpd_config.h, 17
 - hpd_ifaces, 18
- hpd_functions.h, 18
- HPD_GET_CONF_REG
 - hpd_macros.h, 22
- HPD_GET_IRQ_REG
 - hpd_macros.h, 22
- HPD_iface, 14
 - base_address, 14
 - n_packet_channels, 14
 - n_stream_channels, 15
 - rx_packets, 15
 - rx_streams, 15
 - tx_base_address, 15
 - tx_buffer_bytes, 15
 - tx_hibi_address, 15
 - tx_hibi_command, 15
- hpd_ifaces
 - hpd_config.h, 18
- hpd_initialize
 - General Functions, 2
- hpd_irq_clear_vector
 - IRQ Functions, 12
- hpd_irq_clear_vector_gen
 - IRQ Functions, 12
- HPD_IRQ_DIS
 - hpd_macros.h, 22
- hpd_irq_disable
 - IRQ Functions, 12
- hpd_irq_disable_gen
 - IRQ Functions, 12
- HPD_IRQ_ENA
 - hpd_macros.h, 23
- hpd_irq_enable
 - IRQ Functions, 12
- hpd_irq_enable_gen
 - IRQ Functions, 12
- hpd_irq_get_vector
 - IRQ Functions, 13
- hpd_irq_get_vector_gen
 - IRQ Functions, 13
- hpd_irq_packet_ack
 - IRQ Functions, 13
- hpd_irq_packet_ack_gen
 - IRQ Functions, 13
- hpd_irq_stream_ack
 - IRQ Functions, 13
- hpd_irq_stream_ack_gen
 - IRQ Functions, 14
- hpd_macros.h, 21
 - HPD_CLEAR_IRQ_CHAN, 22
 - HPD_CLEAR_IRQ_REG, 22
 - HPD_GET_CONF_REG, 22
 - HPD_GET_IRQ_REG, 22
 - HPD_IRQ_DIS, 22
 - HPD_IRQ_ENA, 23
 - HPD_RX_GET_WORDS, 23
 - HPD_RX_HIBI_ADDR, 23
 - HPD_RX_HIBI_DATA, 23
 - HPD_RX_INIT, 23
 - HPD_RX_MEM_ADDR, 23
 - HPD_RX_WORDS, 23
 - HPD_TX_CMD, 23
 - HPD_TX_CMD_READ, 23
 - HPD_TX_CMD_WRITE, 23
 - HPD_TX_CMD_WRITE_MSG, 23
 - HPD_TX_GET_DONE, 24
 - HPD_TX_HIBI_ADDR, 24
 - HPD_TX_MEM_ADDR, 24
 - HPD_TX_START, 24
 - HPD_TX_WORDS, 24
- HPD_RX_GET_WORDS
 - hpd_macros.h, 23
- HPD_RX_HIBI_ADDR
 - hpd_macros.h, 23
- HPD_RX_HIBI_DATA
 - hpd_macros.h, 23
- HPD_RX_INIT
 - hpd_macros.h, 23
- HPD_RX_MEM_ADDR
 - hpd_macros.h, 23
- HPD_rx_packet, 15
 - rx_base_address, 16
 - rx_buffer_bytes, 16
 - rx_hibi_address, 16
- hpd_rx_packet_get_conf
 - RX Packet Functions, 4
- hpd_rx_packet_get_conf_gen
 - RX Packet Functions, 4
- hpd_rx_packet_init
 - RX Packet Functions, 5

- hpd_rx_packet_init_gen
 - RX Packet Functions, 5
- hpd_rx_packet_poll
 - RX Packet Functions, 5
- hpd_rx_packet_poll_gen
 - RX Packet Functions, 6
- hpd_rx_packet_read
 - RX Packet Functions, 6
- hpd_rx_packet_read_gen
 - RX Packet Functions, 6
- hpd_rx_packet_reinit
 - RX Packet Functions, 6
- hpd_rx_packet_reinit_gen
 - RX Packet Functions, 7
- HPD_rx_stream, 16
 - rx_base_address, 16
 - rx_buffer_bytes, 16
 - rx_hibi_address, 17
 - rx_read_words, 17
- hpd_rx_stream_ack
 - RX Stream Functions, 8
- hpd_rx_stream_ack_gen
 - RX Stream Functions, 8
- hpd_rx_stream_get_conf
 - RX Stream Functions, 8
- hpd_rx_stream_get_conf_gen
 - RX Stream Functions, 8
- hpd_rx_stream_init
 - RX Stream Functions, 9
- hpd_rx_stream_init_gen
 - RX Stream Functions, 9
- hpd_rx_stream_poll
 - RX Stream Functions, 9
- hpd_rx_stream_poll_gen
 - RX Stream Functions, 10
- hpd_rx_stream_read
 - RX Stream Functions, 10
- hpd_rx_stream_read_gen
 - RX Stream Functions, 10
- hpd_rx_stream_reinit
 - RX Stream Functions, 11
- hpd_rx_stream_reinit_gen
 - RX Stream Functions, 11
- HPD_RX_WORDS
 - hpd_macros.h, 23
- hpd_tx_base_conf
 - TX Functions, 2
- hpd_tx_base_conf_gen
 - TX Functions, 2
- HPD_TX_CMD
 - hpd_macros.h, 23
- HPD_TX_CMD_READ
 - hpd_macros.h, 23
- HPD_TX_CMD_WRITE
 - hpd_macros.h, 23
- HPD_TX_CMD_WRITE_MSG
 - hpd_macros.h, 23
- HPD_TX_GET_DONE
 - hpd_macros.h, 24
- HPD_TX_HIBI_ADDR
 - hpd_macros.h, 24
- HPD_TX_MEM_ADDR
 - hpd_macros.h, 24
- hpd_tx_send
 - TX Functions, 2
- hpd_tx_send_copy
 - TX Functions, 3
- hpd_tx_send_copy_gen
 - TX Functions, 3
- hpd_tx_send_gen
 - TX Functions, 3
- HPD_TX_START
 - hpd_macros.h, 24
- HPD_TX_WORDS
 - hpd_macros.h, 24
- IRQ Functions, 11
 - hpd_irq_clear_vector, 12
 - hpd_irq_clear_vector_gen, 12
 - hpd_irq_disable, 12
 - hpd_irq_disable_gen, 12
 - hpd_irq_enable, 12
 - hpd_irq_enable_gen, 12
 - hpd_irq_get_vector, 13
 - hpd_irq_get_vector_gen, 13
 - hpd_irq_packet_ack, 13
 - hpd_irq_packet_ack_gen, 13
 - hpd_irq_stream_ack, 13
 - hpd_irq_stream_ack_gen, 14
- n_packet_channels
 - HPD_iface, 14
- n_stream_channels
 - HPD_iface, 15
- RX Packet Functions, 4
 - hpd_rx_packet_get_conf, 4
 - hpd_rx_packet_get_conf_gen, 4
 - hpd_rx_packet_init, 5
 - hpd_rx_packet_init_gen, 5
 - hpd_rx_packet_poll, 5
 - hpd_rx_packet_poll_gen, 6
 - hpd_rx_packet_read, 6
 - hpd_rx_packet_read_gen, 6
 - hpd_rx_packet_reinit, 6
 - hpd_rx_packet_reinit_gen, 7
- RX Stream Functions, 7

- hpd_rx_stream_ack, 8
- hpd_rx_stream_ack_gen, 8
- hpd_rx_stream_get_conf, 8
- hpd_rx_stream_get_conf_gen, 8
- hpd_rx_stream_init, 9
- hpd_rx_stream_init_gen, 9
- hpd_rx_stream_poll, 9
- hpd_rx_stream_poll_gen, 10
- hpd_rx_stream_read, 10
- hpd_rx_stream_read_gen, 10
- hpd_rx_stream_reinit, 11
- hpd_rx_stream_reinit_gen, 11
- rx_base_address
 - HPD_rx_packet, 16
 - HPD_rx_stream, 16
- rx_buffer_bytes
 - HPD_rx_packet, 16
 - HPD_rx_stream, 16
- rx_hibi_address
 - HPD_rx_packet, 16
 - HPD_rx_stream, 17
- rx_packets
 - HPD_iface, 15
- rx_read_words
 - HPD_rx_stream, 17
- rx_streams
 - HPD_iface, 15
- TX Functions, 2
 - hpd_tx_base_conf, 2
 - hpd_tx_base_conf_gen, 2
 - hpd_tx_send, 2
 - hpd_tx_send_copy, 3
 - hpd_tx_send_copy_gen, 3
 - hpd_tx_send_gen, 3
- tx_base_address
 - HPD_iface, 15
- tx_buffer_bytes
 - HPD_iface, 15
- tx_hibi_address
 - HPD_iface, 15
- tx_hibi_command
 - HPD_iface, 15