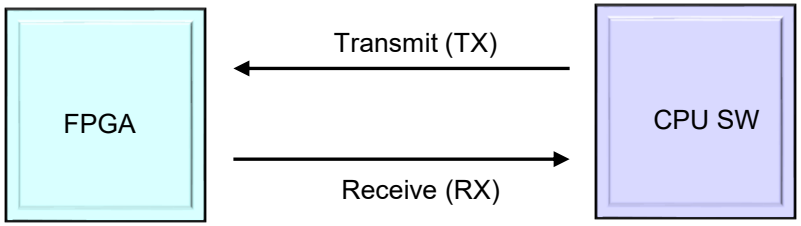
Basic design principles to design with the Smartlogic DMA IP Cores for PCIe

Thomas Zerrer

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Data flow definition RX and TX



Receive (RX)

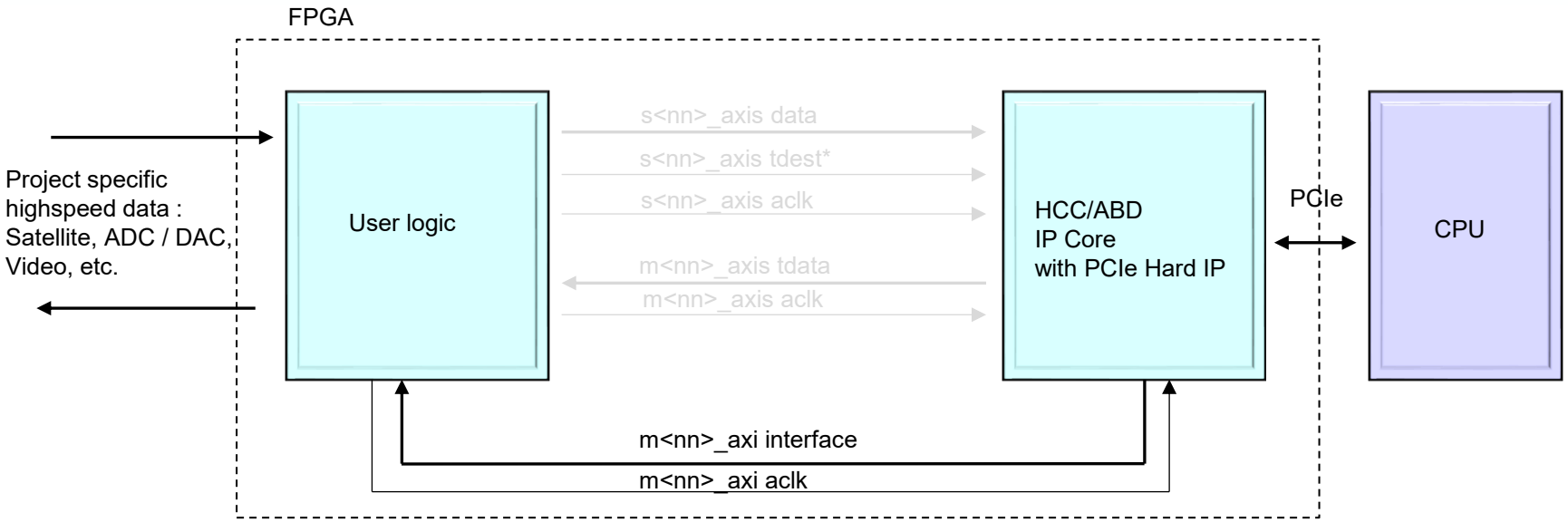
- FPGA transmits data to CPU via DMA
- AXI Stream Slaves receive data from the user logic
- Memory Write TLPs are issued by the FPGA
- User Guide defines this as „DMA Write“

Transmit (TX)

- FPGA reads data from CPU via DMA
- AXI Stream Masters present data to the user logic
- Memory Read TLPs are issued by the FPGA
- User guide defines this as „DMA Read“

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Smartlogic Multichannel DMA IP core – interfacing user I/O



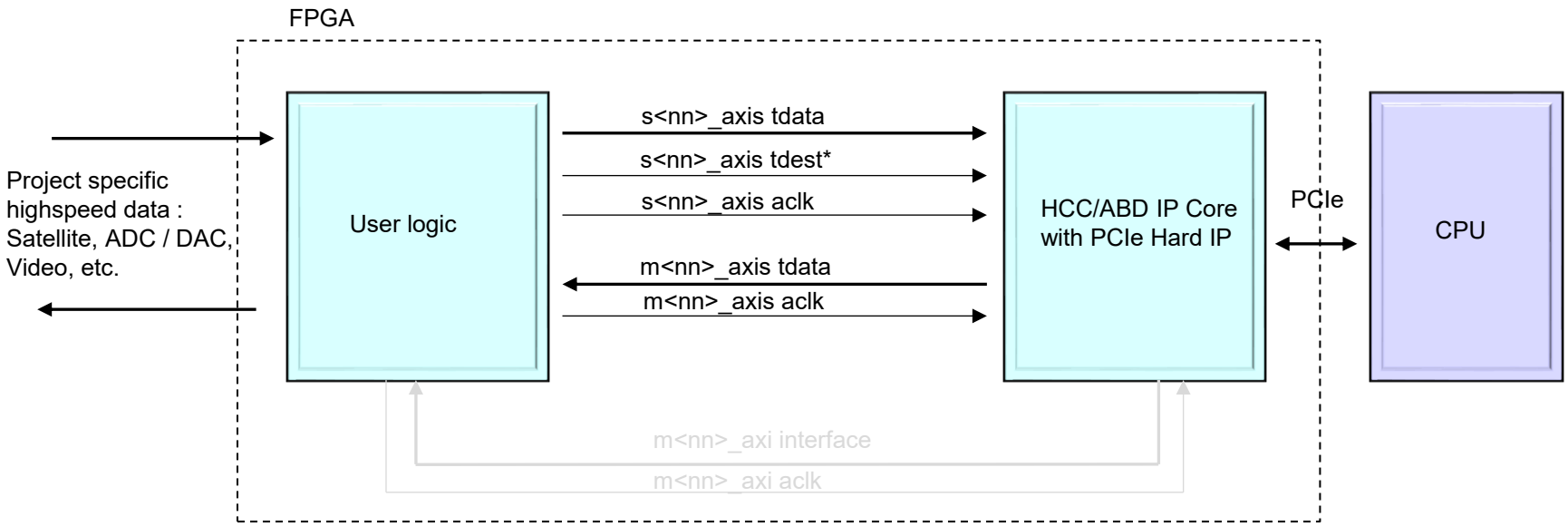
AXI (memorymapped) Master interfaces m<nn>_axi:

- used to configure user logic registers
- each interface is clocked with a dedicated (asynchronous) user clock
- datawidth configurable in powers of 2
- up to 8 AXI Masters available

* Only available for HCC/ABD Core

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Smartlogic multichannel DMA IP core streaming interfaces



AXI Stream slave interfaces (s<nn>_axis) for RX:

- user logic transmits data to the RX data buffers via one of the 16 available s<nn>_axis interfaces
- each interface is clocked with a dedicated (asynchronous) clock input
- each interface contains the TDEST sideband signal to identify the destination of the transfer*
- databitwidth adjustable in powers of 2

AXI Stream master interfaces (m<nn>_axis) for TX:

- IP core reads TX data to the user logic via one of the 16 available m<nn>_axis interfaces
- each interface is clocked with a dedicated (asynchronous) clock input
- databitwidth adjustable in powers of 2

* Only available for HCC/ABD Core

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Use case 1 : Fixed size data transmission

SMARTLOGIC

The Smartlogic PCIe DMA IP cores support fixed size and variable sized datatransmission.

In this section the fixed size data transmission is explained in detail.

Fixed size transmission transfers blocks of data with a fixed size

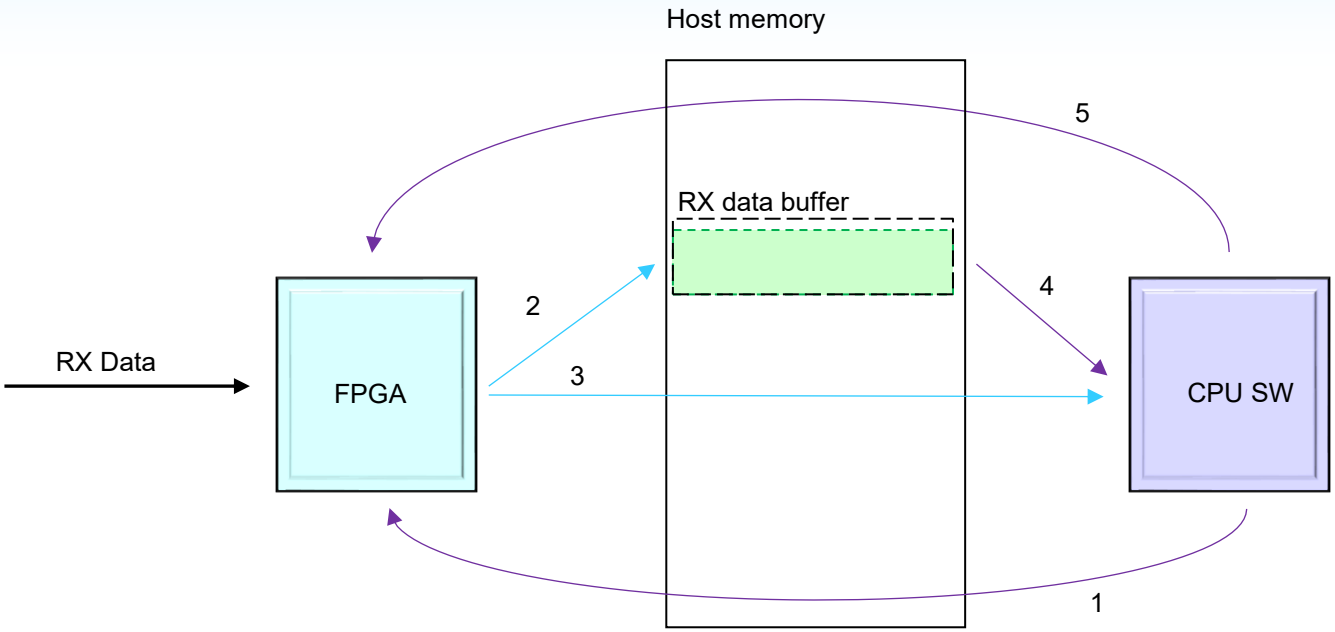
Examples:

- Video Frames (downstream and or upstream)
- Supplying data for DACs (typically downstream)

The following slides show the details for this use case. The variable sized use case is discussed in the section following the fixed size section.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Fixed size : RX – FPGA writes data to the host memory



FPGA

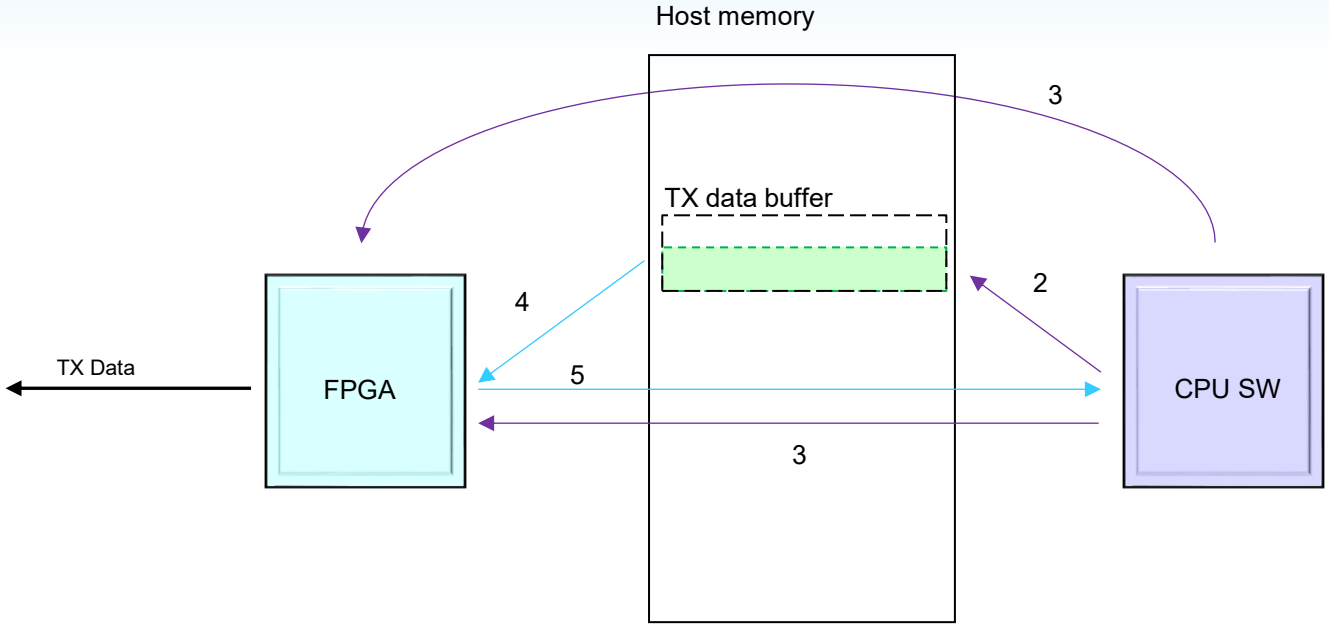
- Step 2:
FPGA transmits valid data to the next empty RX data buffer
- Step 3:
FPGA flags that the RX buffer transmission is complete and issues an interrupt

Smartlogic User Mode Library (together with kernel mode driver)

- Step 1 : User initializes channel and UML will transmit the start addresses of the configured data buffers
- Step 4 : Driver reacts to the interrupt and the data buffer is offered to the user application (pointer to data buffer)
(Either user callback function called or user polls asynchronously)
- Step 5 : Driver transfers the start address of the now empty RX buffer back to the FPGA, when user returns the data buffer

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Fixed size : TX – FPGA reads data from the host memory



FPGA

- Step 4:
FPGA reads the data buffer
- Step 5:
FPGA issues an interrupt

User Application / Smartlogic Driver

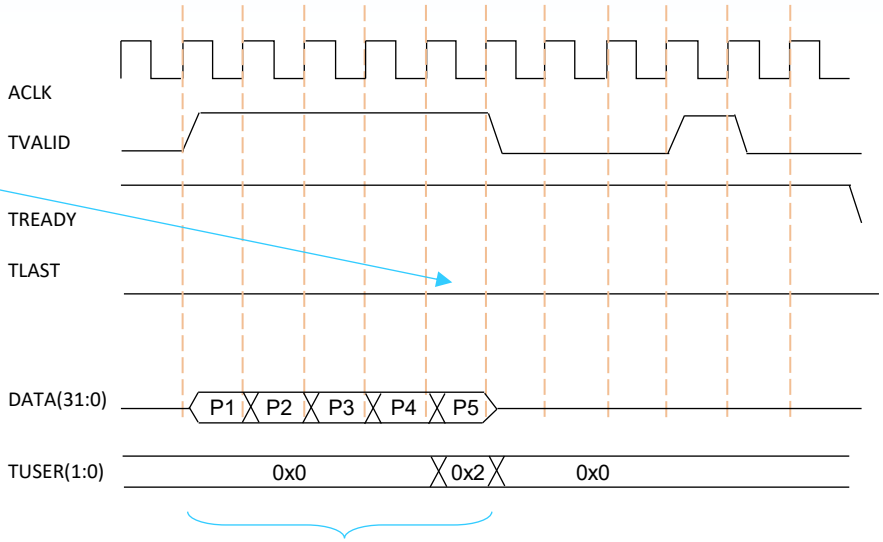
- Step 1:
User app checks if there is an empty TX data buffer
- Step 2:
User app fills the data buffer
- Step 3: Driver informs the FPGA to read the data buffer by writing the startaddress. Length is fixed

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Fixed size : How RX data is passed to the FLEX IP core

Timing diagram for s<nn>_axis:

Data for DB is fed in first without TLAST.

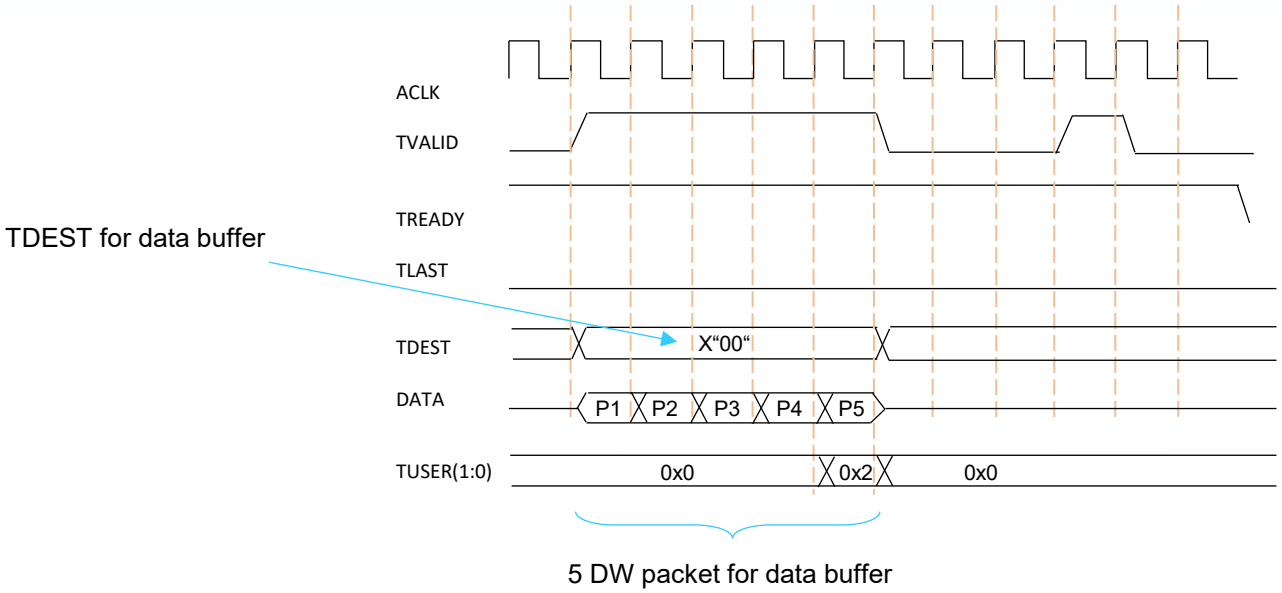


- Back to back transmission is possible.
- Tuser must be driven with 0x2 in the last dataphase of data
- In case of an uneven byte count for DB, the remaining bytes of the last data phase are automatically padded by the core with 0x0
- Example shows 32-Bit s_axis data bitwidth. Other data bit widths are allowed.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Fixed size : How RX data is passed to the HCC/ABD IP core

Timing diagram for s<nn>_axis:



- Back to back transmission is possible
- Tuser must be driven with 0x2 in the last dataphase to switch to the next DB. In fixed mode, tuser must be issued always after the same fixed size (typically the last data phase of the DB)
- TDEST must not change during a packet
- Example shows 32-Bit s_axis data bitwidth. Other data bit widths (powers of 2) are allowed. See UG for details

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Fixed size : DB channel mapping in sldma_test.cpp

The reference design DMA_Demo3 features up to 8 upstream channels that transmit data.

The following table shows the default mapping of the TDEST values for data :

s_axis if #	TDEST for DB #	
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	

If this mapping is kept, the sldma_test.cpp adaptions can be held at a minimum.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Use case 2 : Variable size data transmission

The Smartlogic IP cores also support variable size data transmission

This type of data transmission transfers blocks of data with a variable block length

Examples:

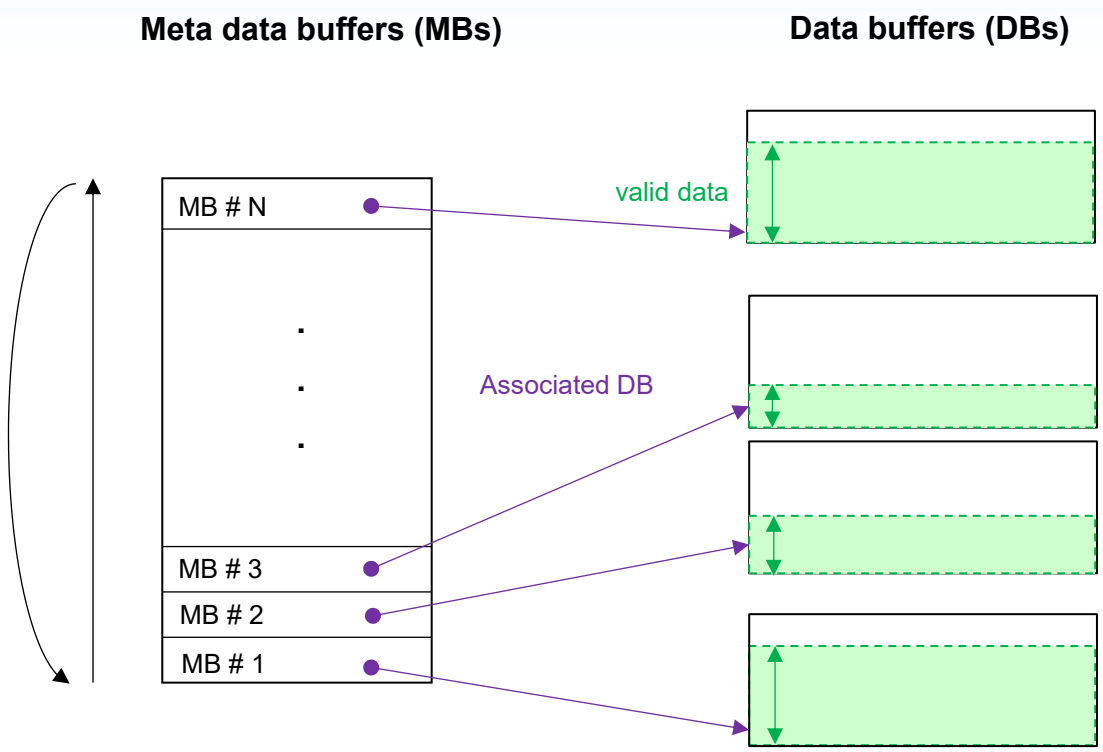
- Ethernet frames (downstream and or upstream)

The following slides show the details for this use case.

Note : The Flex IP Core supports variable sized transfers only upstream

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : Definition of terms „MB“ and „DB“

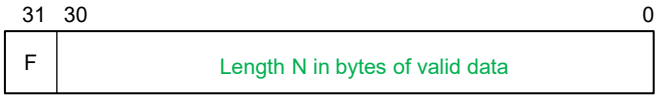


The meta data buffers (MB) reside in host memory and must be contiguous. They provide the length in the first DW and user defined side data.

The data buffers (DB) are allocated in host memory.They have to be contiguous. Their startaddresses can be anywhere in host memory.

For RX & TX startaddresses certain alignments rules apply.

First Dword in each MB must have the following format:



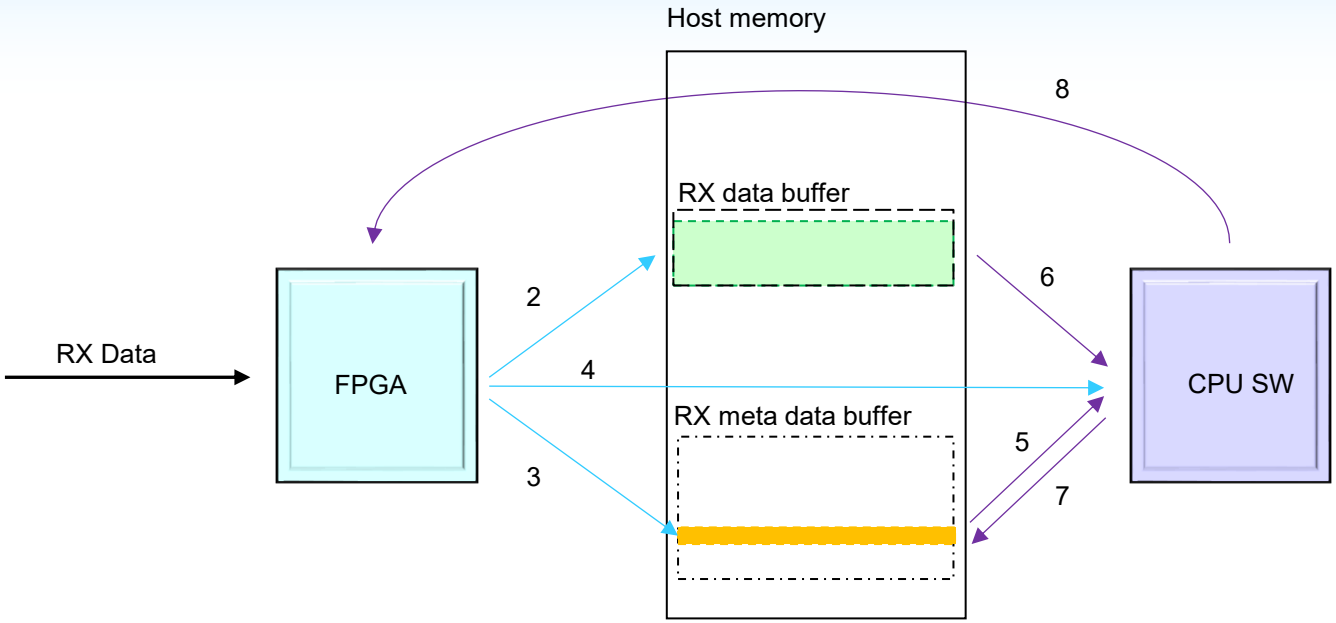
F = Full = ‚1‘, identifies that the associated DB contains valid data with length N
‚0‘ identifies an empty data buffer

The user is free to define the following:

- Number of MBs and associated DBs per channel
- Meta data buffer content is user defined (timestamp, tags, identifier) except for the first DW
- 64-Bit Core : MB length can be any DW number
- 256-Bit Core : MB length can be either 1 DW or multiples of 8 DW

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : RX – FPGA writes data to the host memory



FPGA

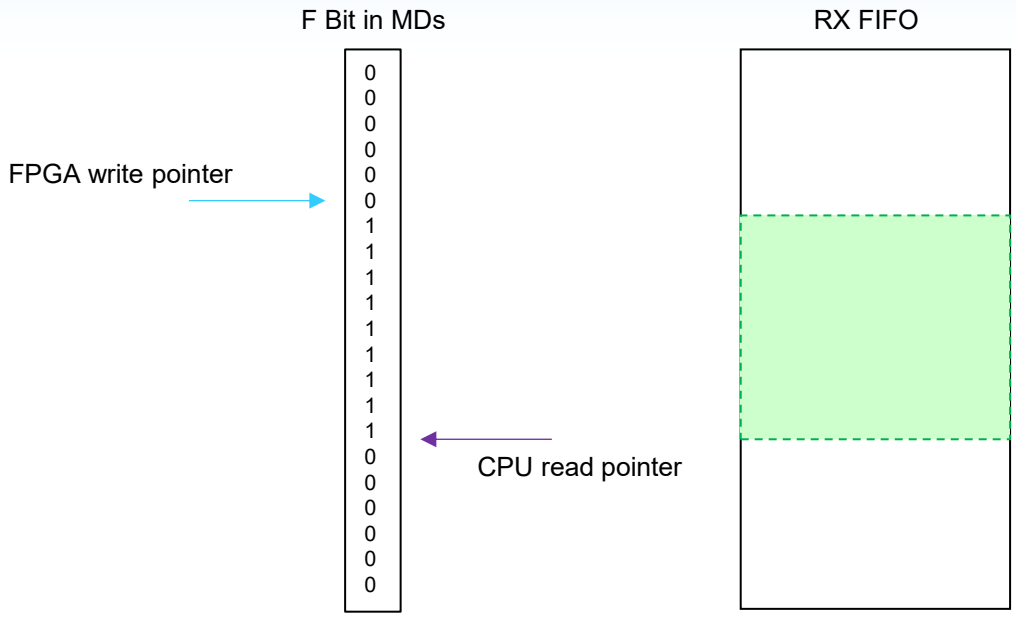
- Step 2:
FPGA transmits valid data to the next empty data buffer
- Step 3:
FPGA transmits the meta data and qualifies the associated data buffer as „full“
- Step 4:
FPGA issues an interrupt

Smartlogic User Mode Library (together with kernel mode driver)

- Step 1 : User initializes channel and UML will transmit the start addresses of the configured data buffers
- Step 5: Driver reacts to the interrupt and examines the first DW of meta data
- Step 6: If the first DW has the F-Bit set to ‚1‘, the data buffer with length N is offered to the user application
- Step 7: After the data is processed, the driver clears the F-Bit in the first meta data DW in order to indicate, that the databuffer is empty
- Step 8 : Driver transfers the start address of the now empty buffer back to the FPGA

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : RX can be considered as a FIFO



RX overrun protection

- The FPGA fills only databuffers for which it has permission
- Permission to fill a specific DB is granted by the CPU in the following way:

In the beginning, all startaddresses are transmitted to the FPGA, so that the FPGA is allowed to fill each buffer one time. After one data buffer is filled, the corresponding address is cleared out of the address FIFO.

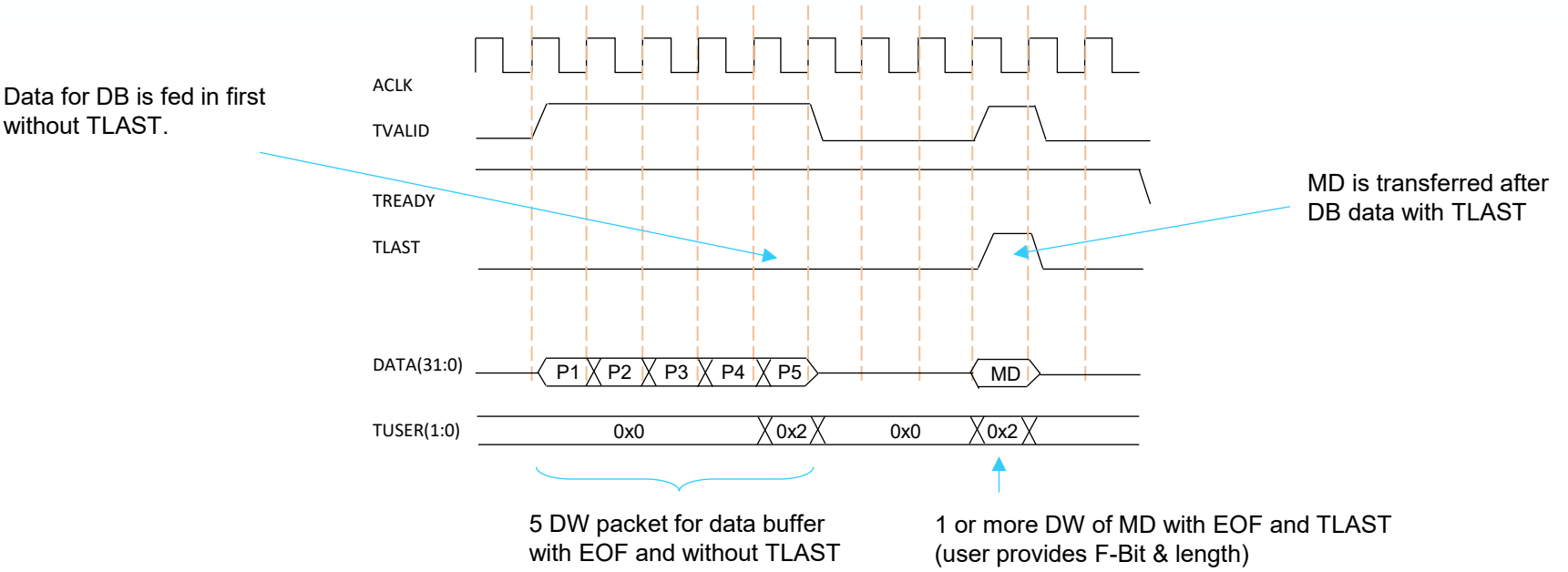
After the CPU has processed the data in the receive data buffer the cpu gives write permission again to this buffer by writing the address of this buffer back in the FPGA address FIFO of the core.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : How RX data and meta data are passed (FLEX IP)

SMARTLOGIC

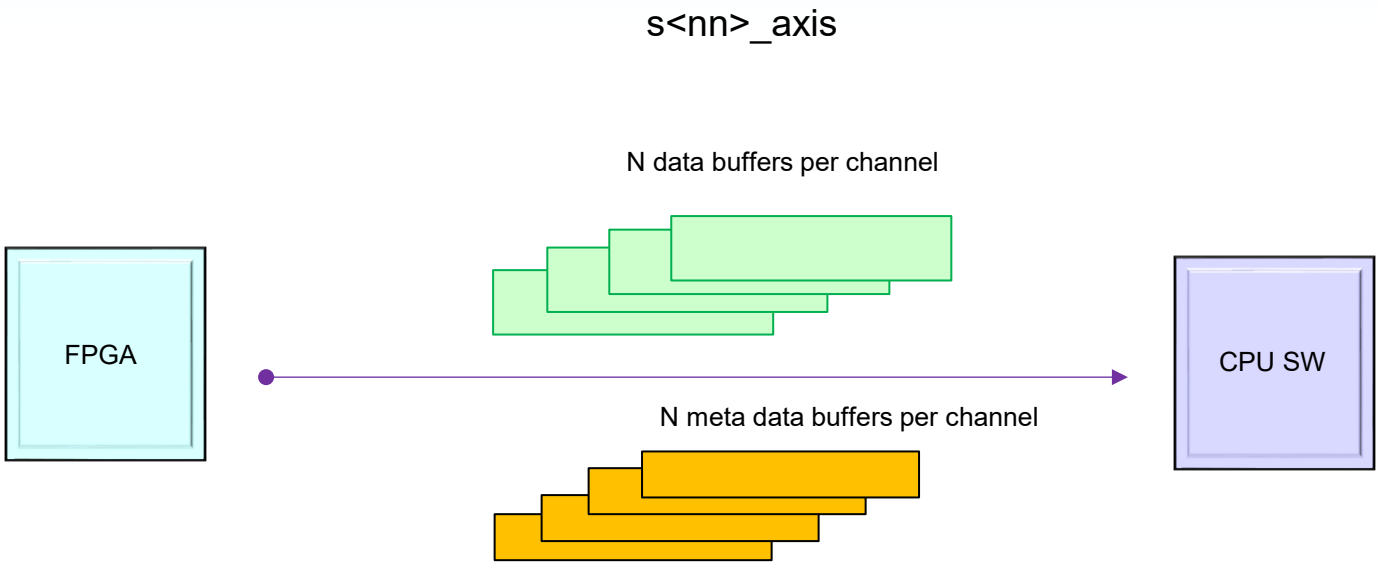
Timing diagram for s<nn>_axis:



- Metadata is transmitted on the same s_axis interface after the data buffer data.
- Back to back transmission is possible.
- The end of a MD must be signaled with tlast.
- Tuser must be driven with 0x2 in the last data phase of data and the last dataphase of MD. In variable size tuser = 0x2 can be issued at any location within the DB (i.e. the DB can be filled partially). However the user has to take care to issue a tuser =0x2 at least with the final data phase of the DB.
- In case of an uneven byte count for DB, pad the last dataphase with dummy data. This is uncritical, since the true byte length is presented to the CPU via the MD length field.
- Example shows 32-Bit s_axis data bitwidth. Other data bit widths are allowed, but each MD must contain at least 1 DW of data.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

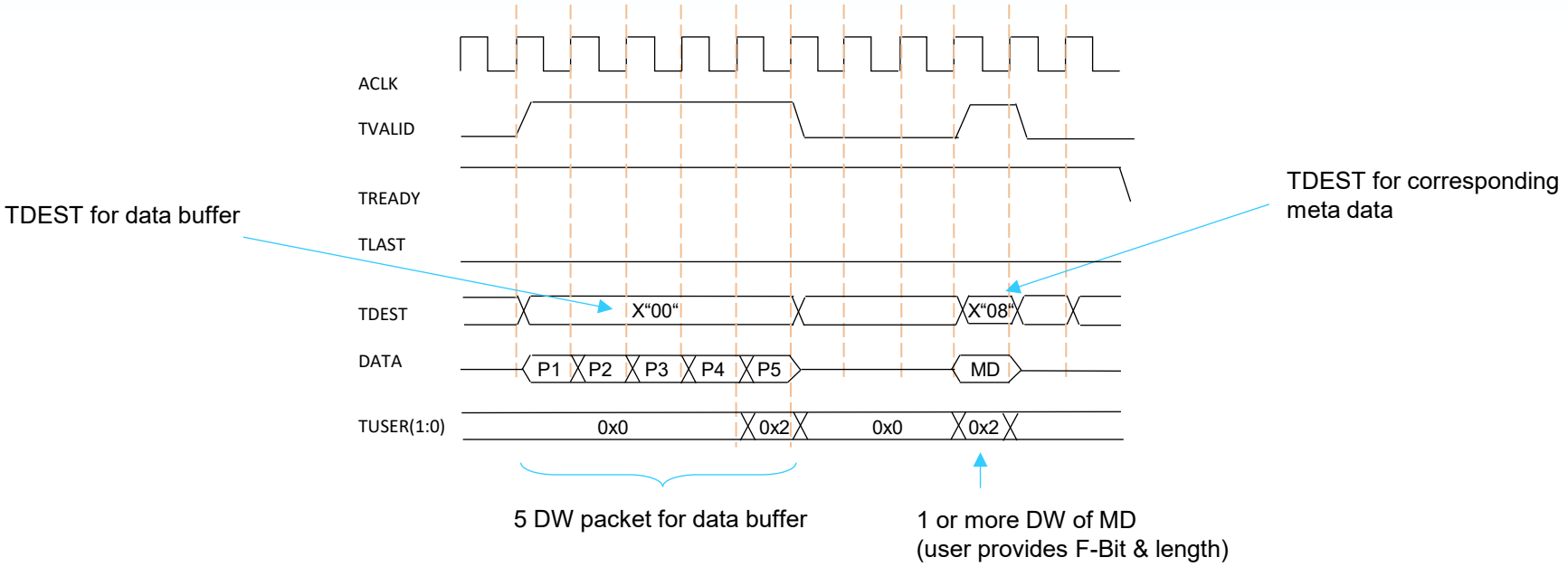
Variable size : Block Diagram



Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : How RX data and meta data are passed (HCC/ABD IP)

Timing diagram for s<nn>_axis:



- MDs are transmitted on the same s_axis interface after the data buffer data
- Back to back transmission is possible
- The end of a packet of a MD must be signaled with tlast and tuser = 0x2.
- Tuser must be driven with 0x2 in the last data phase of data and the last dataphase of MD. In variable size tuser = 0x2 can be issued at any location within the DB (i.e. the DB can be filled partially). However the user has to take care to issue a tuser =0x2 at least with the final data phase of the DB.
- TDEST must not change during a packet
- Example shows 32-Bit s_axis data bitwidth. Other data bit widths are allowed, but each MD must contain at least 1 DW of data.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

TLAST behavior:

TLAST forces the current stream packet to be transferred to the destination buffer. The current data buffer is not marked as „filled“ and streaming will continue at the end of the current write pointer of the same data buffer.

TUSER = 0b10 also forces the current stream packet to be transferred, but it will mark the current data buffer as „filled“ and the next AXI stream packet will be transferred to the next data buffer.

Most application do not require that s<nn>_axis_tlast is used and this signal can be safely tied to ‚0‘. This will have the positive advantage, that all packets are transferred as maximum sized packets and bandwidth will be utilized efficiently.

There might be cases, where the usage of TLAST makes sense. The following examples show, where:

- Transmission of faulty video frames in security applications, where some pixels might get lost, and where the frame synchronization is still important. See „Line Mode“ in the User Guide for details
- Latency critical applications, where it is known, that the current data should be transferred immediately to its destination. Important is, that in this case the s_axis data width has to be 256-Bits to avoid side effects.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : DB and MB channel mapping in sldma_test.cpp

SMARTLOGIC

The reference design DMA_Demo3 features 8 upstream channels that transmit data and meta data.

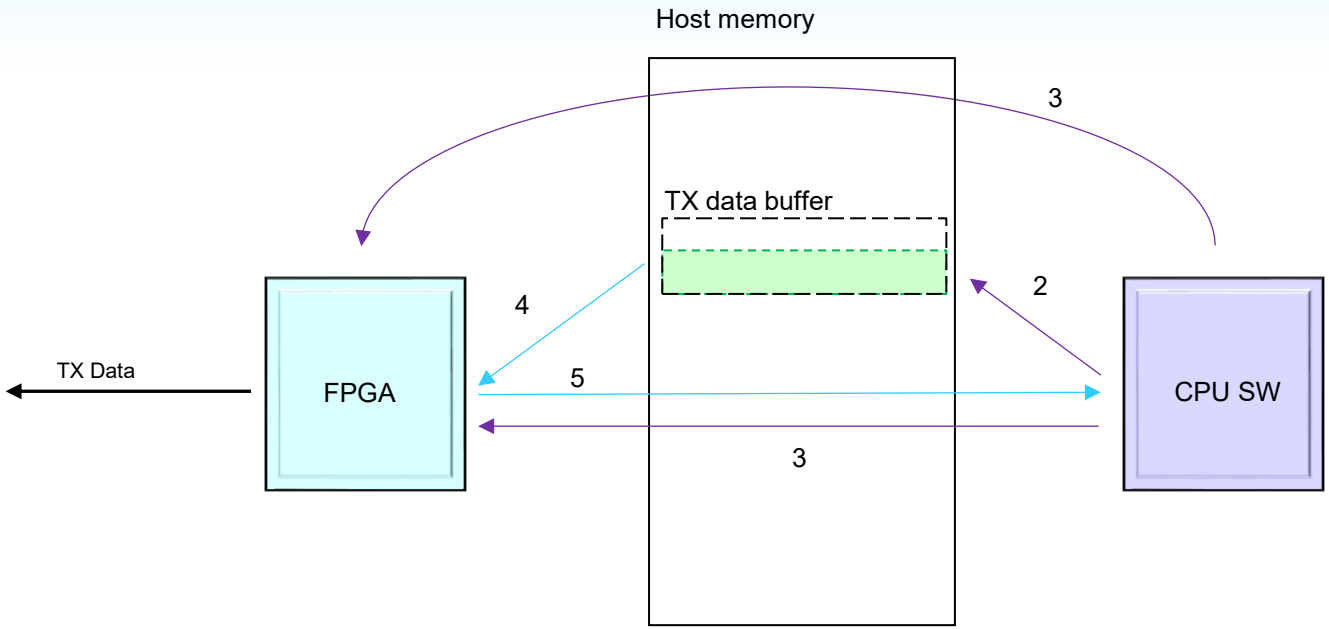
The following table shows the default mapping of the TDEST values for data and meta data :

s_axis if #	TDEST for DB #	TDEST for MB #
0	0	8
1	1	9
2	2	10
3	3	11
4	4	12
5	5	13
6	6	14
7	7	15

Please make sure to set interrupt_scheme_g to 1 and set write_tdest_width_g to 4
If this mapping is kept, the sldma_test.cpp adaptions can be held at a minimum.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

Variable size : TX – FPGA reads data from the host memory



FPGA

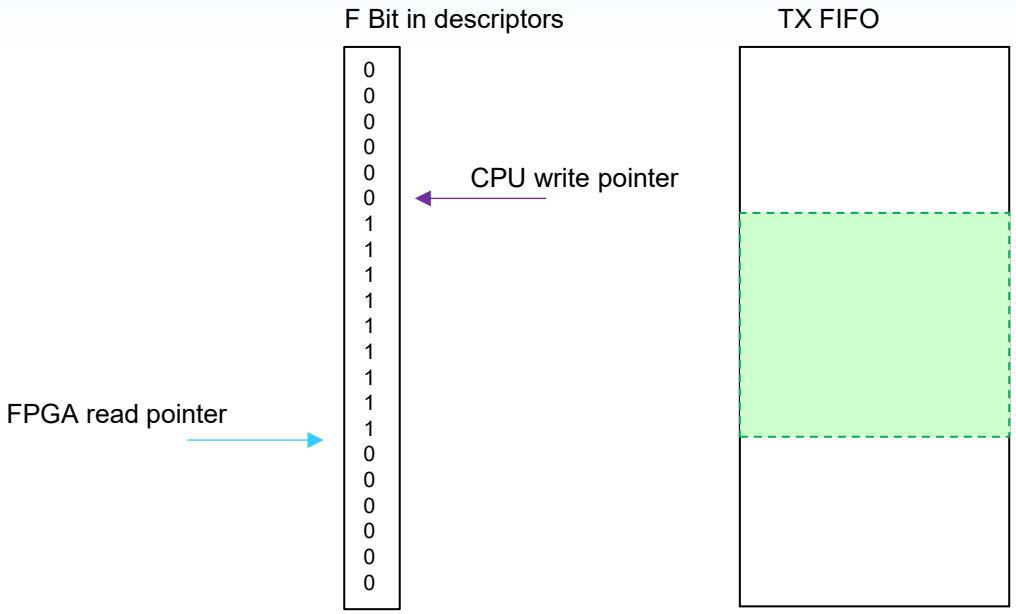
- Step 4:
FPGA reads the data buffer
- Step 5:
FPGA issues an interrupt

User Application / Smartlogic Driver

- Step 1:
User app checks if there is an empty TX data buffer
- Step 2:
User app fills the data buffer
- Step 3: Driver informs the FPGA to read the data buffer by writing startaddress and length of TX DB

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

TX can be considered as a FIFO



TX overrun protection

- The user app has to issue a request for a TX data buffer pointer
- Permission to reuse a TX data buffer is granted by the FPGA in the following way:

In the beginning, user app has the permission to fill all TX data buffers one time

After the FPGA has read the data in a transmit data buffer, the FPGA gives write permission again to the CPU for this buffer by issueing an interrupt to the driver. The interrupt indicates that the TX data buffer has been successfully transmitted and can be re-used again.

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential

16 channels : DB and MB channel mapping

SMARTLOGIC

In case more than 8 channels are required, the following TDEST mapping is required for data and meta data:

s_axis if #	TDEST for DB #	TDEST for MB #
0	0	16
1	1	17
2	2	18
3	3	19
4	4	20
5	5	21
…	…	…
15	15	31

Please make sure to set interrupt_scheme_g to 4 and set write_tdest_width_g to 5

Copyright Smartlogic 2005-2024, All Rights reserved. Confidential