

PCI DMA Core
FPGA Design Specification

Example Only

Company:	
Contact:	

Author:	
Date:	
Revision:	1.0

Revision History

Revision	Date	Author	Description
1.0			First Draft

Applicable Documents

Table of Contents

1 INTRODUCTION.....6

2 GENERAL DESCRIPTION6

3 CLOCKS.....7

4 I/O VOLTAGE.....7

5 RESETS8

5.1 POWER-ON RESET.....8

5.2 PCI RESET8

5.3 SOFTWARE RESET8

6 PCI INTERFACE9

6.1 TARGET INTERFACE.....9

6.1.1 Base Address Registers9

6.1.2 Memory Map9

6.1.3 Registers.....10

6.1.4 Error/Exception Handling15

6.2 INITIATOR INTERFACE.....15

6.2.1 General Description.....16

6.2.2 TX DMA16

6.2.3 RX DMA18

7 BACKEND APPLICATION INTERFACE21

7.1 DATA FIFOS21

7.2 LOOPBACK LOGIC.....21

8. PINOUTS.....22

APPENDIX A – ABBREVIATIONS.....23

List of Figures

Figure 1 Top Level Functional Diagram..... 7
Figure 2 TX DMA Linear Flow 17
Figure 3 Data format in Tx Data FIFO (linear DMA mode) 17
Figure 5 RX Linear DMA Flow 19
Figure 6 Data format of Rx Data FIFO (Linear mode) 19

List of Tables

Table 1 BAR0 Memory Map 9
Table 2 TX DMA Length Register..... 10
Table 3 TX DMA Address Register..... 10
Table 4 TX DMA Control Register 11
Table 5 TX DMA Status Register..... 11
Table 6 RX DMA Length Register 12
Table 7 RX DMA Address Register 12
Table 8 RX DMA Control Register..... 12
Table 9 RX DMA Status Register 13
Table 10 Interrupt Mask Register 13
Table 11 Interrupt Cause Register..... 14
Table 12 Current Interrupt Source Register..... 15

1 Introduction

This document describes a PCI 64/66 target and initiator design with DMA. A detailed description of hardware to be implemented in a FPGA will be presented.

In this document, TX represents data to the FPGA via the PCI Bus and RX represents data from the FPGA via the PCI bus.

2 General Description

This design consists of the following functional blocks:

1. FPGA PCI Core 64/66
2. Target
 - a. Target Interface logic to FPGA PCI Core
 - b. Registers
 - i. General purpose Registers
 - ii. DMA Registers
 - iii. Interrupt Registers
 - c. Reset Logic
3. Initiator
 - a. Initiator Interface logic to FPGA PCI Core
 - b. TX DMA
 - c. RX DMA
 - d. DMA Arbiter
 - e. TX Data FIFO
 - f. RX Data FIFO
4. Backend Application Interface
 - a. Data FIFOs
 - b. Loopback Logic

[Figure 1](#) shows block diagram of the design.

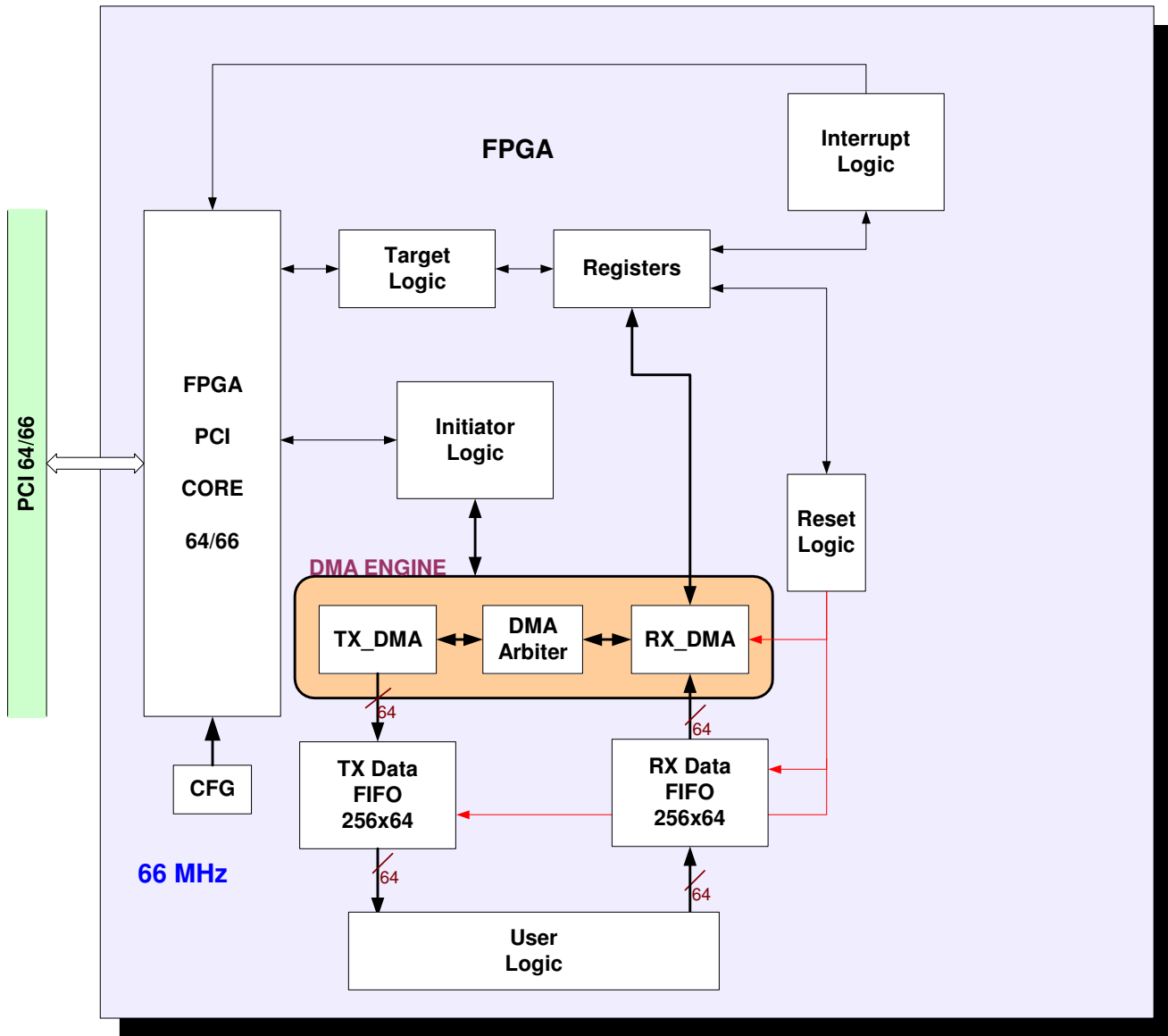


Figure 1 Top Level Functional Diagram

3 Clocks

This design will support one clock.

1. PCI clock of 15 ns (approx. 66 MHz)

4 I/O Voltage

This design will support only PCI 3.3 voltage standard.

5 Resets

The FPGA will support the following resets:

1. Power-on Reset
2. PCI Reset
3. Software Reset

5.1 Power-On Reset

Typically on power up, the FPGA will be loaded from serial prom and the PCI bus must be reset. The PCI spec v2.2 states that reset must remain asserted until the following conditions are met:

1. RST# is asserted a minimum of 1 ms
2. CLK has been oscillating for a minimum of 100 ms
3. Voltage levels have remained above minimum levels for about 100 ms

Additionally, initialization time begins when RST# is deasserted and completes 2^{25} PCI clocks later. At 33 MHz this equates to 1.0066 secs, at 66 MHz this equates to 0.5033 secs

If a target is accessed during this initialization time, it is allowed to do any of the following:

1. Ignore the request
2. Claim the access and hold in wait states until it can complete the request
3. Claim the access and terminate with retry

End customer must ensure the FPGA FPGA has ample time to download its bit stream and meet initialization requirements for PCI.

5.2 PCI Reset

When PCI Reset is asserted the whole FPGA will be asynchronously reset to a known state. However it will not initiate a reloading of the FPGA from the serial PROM.

5.3 Software Reset

The FPGA will support reset of the following functional blocks:

1. TX DMA engine
2. TX Data FIFO
3. RX DMA engine
4. RX Data FIFO

See Tables [5](#) & [9](#) for detailed description of register bits.

6 PCI Interface

The PCI interface is broken down into two major blocks, Target and Initiator interfaces.

6.1 Target Interface

PCI target control logic will handle all read and write accesses across the PCI bus. This logic will permit only the reading/writing of registers. Target logic will support one Base Address Register, BAR, as defined below.

6.1.1 Base Address Registers

Only BAR0 will be supported, with the following features:

1. 256 Words of 32 bits MEMORY address space
2. All addresses are 32 bit aligned
3. Register accesses:
 - a. Single data phase transfers
 - b. prefetchable
 - c. Non-posted
 - d. Byte Enables not supported

6.1.2 Memory Map

Table 1 BAR0 Memory Map

Offset	Name	Description
0x00	Reserved	
0x04	TX_DMA_LEN	TX DMA Length. Length of block of data
0x08	TX_DMA_ADDR	TX DMA Address. Address in Host memory where Tx Data resides
0x0C	TX_DMA_CTRL	TX DMA control register
0x10	TX_DMA_STAT	TX DMA status register
0x14 - 0x20	Reserved	
0x24	RX_DMA_LEN	RX DMA Length. Length of block of data
0x28	RX_DMA_ADDR	RX DMA Address. Address in Host memory where Rx Data will reside
0x2C	RX_DMA_CTRL	RX DMA control register
0x30	RX_DMA_STAT	RX DMA status register
0x34	RX_DMA_COUNT	
0x38 - 0x80	Reserved	
0x84	INT_MASK	Interrupt Mask. A one on any bit enables the interrupt. Zero will mask out the interrupt. All interrupts are masked with a zero after reset.
0x88	INT_CAUSE	Interrupt Cause. Indicates which interrupt source was activated. Always active high. To clear interrupt, write a one to desired bit position. Writing a zero has no effect.
0x8C	INT_CURRENT	Current Interrupt Source. Holds the non-modified values of all interrupt sources.
0x90 - 0xFC	Reserved	

Note only address bits[7:2] will be decoded for register accesses. As such, address aliasing will be inherent to the design.

6.1.3 Registers

Note Bit31 is the most significant bit. Bit0 is the least significant bit.

Table 2 TX DMA Length Register

	MSB			LSB	
Bit	31	26	25	2	0
Type	R/O		R/W		R/O
Power-on State	N/A		All zeroes		N/A
Name	N/A		TX_DMA_LEN		N/A
Description	Unused. Upon a PCI read, these bits will be driven to zero.		TX DMA Length. Length of block of data		All lengths will be aligned to 8 byte boundaries. As such these bits are unused. Upon a PCI read, these bits will be driven to zero.

Table 3 TX DMA Address Register

	MSB			LSB	
Bit	31	3	2	2	0
Type	R/W			R/O	
Power-on State	All zeroes			N/A	
Name	TX_DMA_ADDR			N/A	
Description	TX DMA Address. Address in Host memory where Tx Data resides			All addresses will be aligned to 64 bit boundaries. As such these bits are unused. Upon a PCI read, these bits will be driven to zero.	

Table 4 TX DMA Control Register

Bit	Type	Power-on State	Name	Description
0	R/W	0	START	On Write Access: 0: Do nothing 1: Start DMA Engine It will be cleared on any of the following conditions a. completion of DMA b. STOP = 1 c. ABORT = 1 A read access of this bit will reflect the current contents of this bit
1	R/W	0	STOP	On Write Access: 0: Do nothing 1: Stop DMA Engine after current access A read access of this bit will reflect the current contents of this bit
2	R/W	0	ABORT	On Write Access: 0: Do nothing 1: Abort DMA. Reset DMA engine and FIFO A read access of this bit will reflect the current contents of this bit
3 - 31	N/A	-	-	Unused

Table 5 TX DMA Status Register

Bit	Type	Power-on State	Name	Description
0	R/O	0	DMA_ACTIVE	On Read Access: 0: DMA inactive 1: DMA active
1	R/O	0	DMA_PENDING	On Read Access: 0: No stop pending 1: Stop pending
2	R/O	0	FIFO_EMPTY	On Read Access: 0: Not empty 1: Empty
3	R/O	0	FIFO_HALF_EMPTY	On Read Access: 0: > 1024 Bytes 1: ≤ 1024 Bytes
4	R/O	0	FIFO_FULL	On Read Access: 0: Not full 1: Full
5	R/O	0	FIFO_ALMOST_FULL	On Read Access: 0: < 1984 Bytes 1: ≥ 1984 Bytes
6	R/O	0	FIFO_ALMOST_EMPTY	On Read Access: 0: > 64 Bytes 1: ≤ 64 Bytes
7	R/O	0	DMA_ERROR	On Read Access: 0: No Error 1: DMA Error
8-31	N/A	-	-	Unused

Table 6 RX DMA Length Register

	MSB			LSB	
Bit	31	26	25	2	0
Type	R/O		R/W		R/O
Power-on State	N/A		All zeroes		N/A
Name	N/A		RX_DMA_LEN		N/A
Description	Unused. Upon a PCI read, these bits will be driven to zero.		RX DMA Length. Length of block of data		All lengths will be aligned to 8 byte boundaries. As such these bits are unused. Upon a PCI read, these bits will be driven to zero.

Table 7 RX DMA Address Register

	MSB			LSB	
Bit	31	3	2	0	
Type	R/W			R/O	
Power-on State	All zeroes			N/A	
Name	RX_DMA_ADDR			N/A	
Description	RX DMA Address. Address in Host memory where Rx Data will reside			All addresses will be aligned to 64 bit boundaries. As such these bits are unused. Upon a PCI read, these bits will be driven to zero.	

Table 8 RX DMA Control Register

Bit	Type	Power-on State	Name	Description
0	R/W	0	START	On Write Access: 0: Do nothing 1: Start DMA Engine It will be cleared on any of the following conditions a. completion of DMA b. STOP = 1 c. ABORT = 1 A read access of this bit will reflect the current contents of this bit
1	R/W	0	STOP	On Write Access: 0: Do nothing 1: Stop DMA Engine after current access A read access of this bit will reflect the current contents of this bit
2	R/W	0	ABORT	On Write Access: 0: Do nothing 1: Abort DMA. Reset DMA engine and FIFO A read access of this bit will reflect the current contents of this bit
3 - 31	N/A	-	-	Unused

Table 9 RX DMA Status Register

Bit	Type	Power-on State	Name	Description
0	R/O	0	DMA_ACTIVE	On Read Access: 0: DMA inactive 1: DMA active
1	R/O	0	DMA_PENDING	On Read Access: 0: No stop pending 1: Stop pending
2	R/O	0	FIFO_EMPTY	On Read Access: 0: Not empty 1: Empty
3	R/O	0	FIFO_HALF_EMPTY	On Read Access: 0: > 1024 Bytes 1: ≤ 1024 Bytes
4	R/O	0	FIFO_FULL	On Read Access: 0: Not full 1: Full
5	R/O	0	FIFO_ALMOST_FULL	On Read Access: 0: < 1984 Bytes 1: ≥ 1984 Bytes
6	R/O	0	FIFO_ALMOST_EMPTY	On Read Access: 0: > 64 Bytes 1: ≤ 64 Bytes
7	R/O	0	DMA_ERROR	On Read Access: 0: No Error 1: DMA Error
8-31	N/A	-	-	Unused

Table 10 Interrupt Mask Register

Bit	Type	Power-on State	Name	Description
0	R/W	0	TABORT_ERR	Target Abort Interrupt. On Write Access: 0: Mask Interrupt 1: Enable Interrupt A read access of this bit will reflect the current contents of this bit
1	R/W	0	MABORT_ERR	Master Abort Interrupt. On Write Access: 0: Mask Interrupt 1: Enable Interrupt A read access of this bit will reflect the current contents of this bit
2	R/W	0	FAIL64_INT	FAIL64 Interrupt. On Write Access: 0: Mask Interrupt 1: Enable Interrupt A read access of this bit will reflect the current contents of this bit
3	R/W	0	TX_DMA_INT	TX DMA interrupt. On Write Access: 0: Mask Interrupt 1: Enable Interrupt A read access of this bit will reflect the current contents of this bit
4	R/W	0	RX_DMA_INT	RX DMA interrupt. On Write Access: 0: Mask Interrupt

Bit	Type	Power-on State	Name	Description
				1: Enable Interrupt A read access of this bit will reflect the current contents of this bit
5-31	N/A	-	-	Unused

Table 11 Interrupt Cause Register

Bit	Type	Power-on State	Name	Description
0	R/W	0	TABORT_ERR	Target Abort Interrupt. On Write Access: 0: Do nothing 1: Clear interrupt On Read Access: 0: interrupt inactive 1: interrupt active
1	R/W	0	MABORT_ERR	Master Abort Interrupt. On Write Access: 0: Do nothing 1: Clear interrupt On Read Access: 0: interrupt inactive 1: interrupt active
2	R/W	0	FAIL64_INT	FAIL64 Interrupt. On Write Access: 0: Do nothing 1: Clear interrupt On Read Access: 0: interrupt inactive 1: interrupt active
3	R/W	0	TX_DMA_INT	TX DMA interrupt. On Write Access: 0: Do nothing 1: Clear interrupt On Read Access: 0: interrupt inactive 1: interrupt active
4	R/W	0	RX_DMA_INT	RX DMA interrupt. On Write Access: 0: Do nothing 1: Clear interrupt On Read Access: 0: interrupt inactive 1: interrupt active
5-31	N/A	-	-	Unused

Table 12 Current Interrupt Source Register

Bit	Type	Power-on State	Name	Description
0	R/O	0	TABORT_ERR	Target Abort Interrupt. On Read Access: 0: interrupt inactive 1: interrupt active
1	R/O	0	MABORT_ERR	Master Abort Interrupt. On Read Access: 0: interrupt inactive 1: interrupt active
2	R/O	0	FAIL64_INT	FAIL64 Interrupt. On Read Access: 0: interrupt inactive 1: interrupt active
3	R/O	0	TX_DMA_INT	TX DMA interrupt. On Read Access: 0: interrupt inactive 1: interrupt active
4	R/O	0	RX_DMA_INT	RX DMA interrupt. On Read Access: 0: interrupt inactive 1: interrupt active
5-31	N/A	-	-	Unused

6.1.4 Error/Exception Handling

The following conditions will be handled by the target logic:

1. Write accesses to read only registers will be ignored
2. Read accesses to write only registers will result in undetermined behavior.
3. No target aborts will be generated.

All other error or exception conditions will be ignored or result in undefined behavior.

6.2 Initiator Interface

The initiator interface is comprised of the following blocks:

1. Initiator Interface Logic
2. TX design
 - a. Linear DMA Engine
 - b. Data FIFO
3. RX design
 - a. Linear DMA Engine
 - b. Data FIFO

6.2.1 General Description

The DMA Controller is designed to transfer data between the PCI bus and a generic FIFO Interface. The design also has a registers interface, which allows the CPU to configure the Controller registers and initiate the transfers.

Features of the DMA Controller are:

- Design to work with a single block in memory.
- Needs to be “kick started” via a write to a register to initiate transfers.
- Buffer start and end can be at any valid 64 bit aligned address.
- Amount of data to be transferred can be any multiple of 8 bytes.

6.2.2 TX DMA

Types of TX DMA are supported:

- a. Linear DMA

6.2.2.1 Linear DMA

6.2.2.1.1 General Operation

The linear DMA is a traditional block DMA with a DMA length (26 bits) and a buffer address register.

For the TX linear DMA, the following registers are used:

- TX_DMA_CTRL
- TX_DMA_STAT
- TX_DMA_ADDR
- TX_DMA_LEN

The DMA engine initiates transfer when START bit is asserted and the data FIFO is less than half full. The DMA engine transfers TX_DMA_LEN amount of data from the address TX_DMA_ADDR. Figure 2 depicts the TX DMA linear flow. The ABORT signal will asynchronously reset the flow. START bit will be cleared on STOP or ABORT, or completion of DMA (with or without errors)

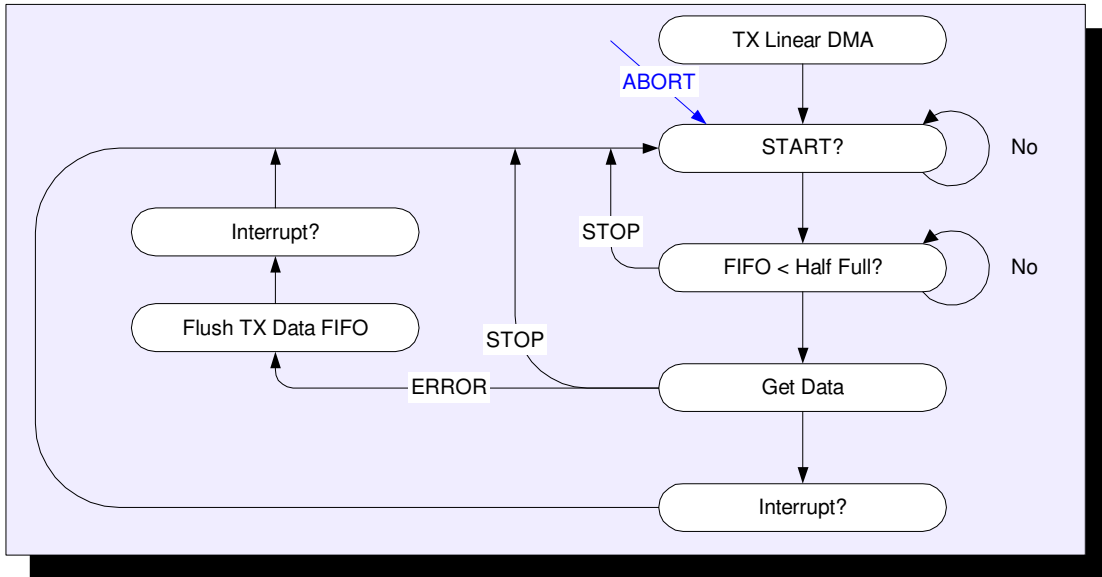


Figure 2 TX DMA Linear Flow

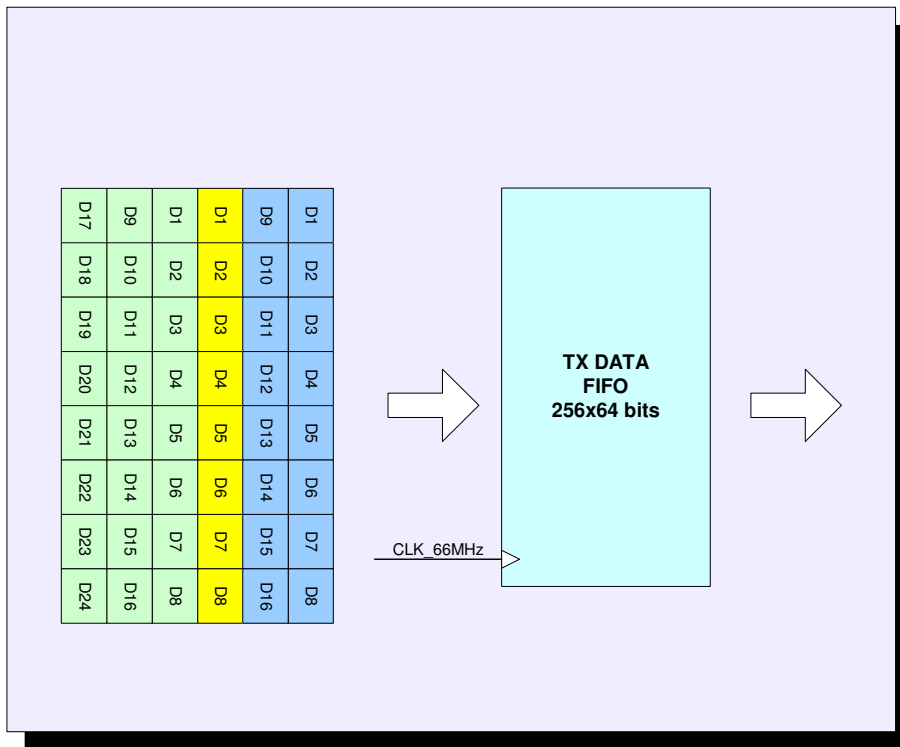


Figure 3 Data format in Tx Data FIFO (linear DMA mode)

6.2.2.1.2 Error/Exception Handling

The following errors and/or exceptions are possible:

1. Fail64 (Failed access : Try 64bit access to 32bit target)
2. Target Abort
3. Master Abort
4. Parity Error (PERR#)
5. System Error (SERR#)
6. TX FIFO never half empty
7. Infinite PCI Retry loop

Of the above errors, only Fail64, Target abort and Master abort will be supported. On error, the FIFO will be flushed and if enabled, an interrupt will be generated.

6.2.3 RX DMA

Types of RX DMA are supported:

- a. Linear DMA

6.2.3.1 Linear DMA

6.2.3.1.1 General Operation

The linear DMA is a traditional block DMA with a DMA length (26 bits) and a buffer address register.

For the RX linear DMA, the following registers are used:

- RX_DMA_CTRL
- RX_DMA_STAT
- RX_DMA_ADDR
- RX_DMA_LEN

The DMA engine initiates transfer when START bit is asserted. The DMA engine transfers RX_DMA_LEN amount of data from the address RX_DMA_ADDR. Figure 5 depicts the RX DMA linear flow. The ABORT signal will asynchronously reset the flow. START will be cleared on STOP or ABORT, or completion of DMA

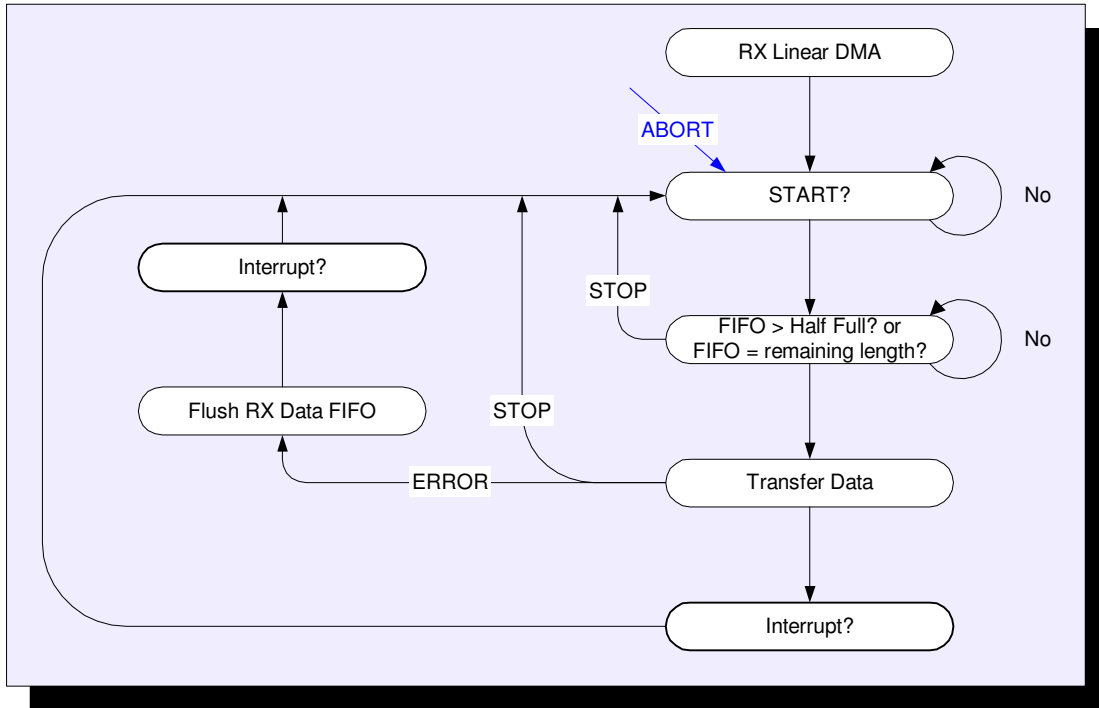


Figure 5 RX Linear DMA Flow

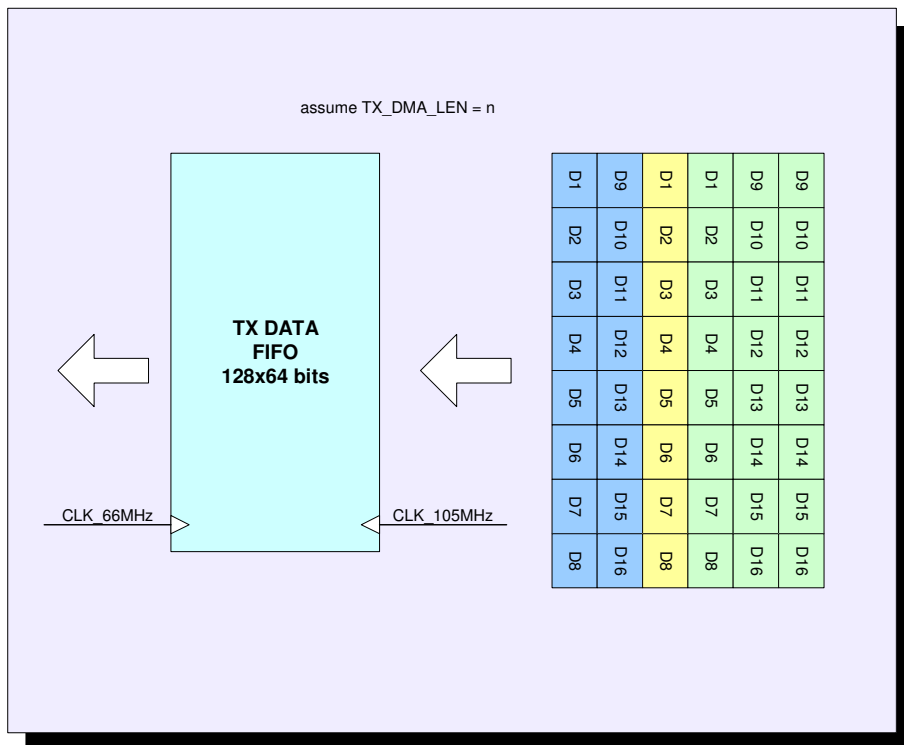


Figure 6 Data format of Rx Data FIFO (Linear mode)

6.2.3.1.2 Error/Exception Handling

The following errors and/or exceptions are possible:

1. Fail64
2. Target Abort
3. Master Abort
4. Parity Error (PERR#)
5. System Error (SERR#)
6. RX FIFO never half full
7. RX FIFO never receive required amount of data
8. Infinite PCI Retry loop

Of the above errors, only Fail64, Target abort and Master abort will be supported. On error, the FIFO will be flushed and if enabled, an interrupt will be generated.

7 Backend Application Interface

7.1 Data FIFOs

The DMA data FIFOs will be synchronous. Data will be formatted as shown in each DMA section.

7.2 LoopBack Logic

As long as the TX FIFO is not empty and the RX FIFO is not full, the loop back logic will read data out of the TX FIFO and write it into the RX FIFO.

8. Pinouts

Signal	Signal Type	FPGA Pin	Description
PCLK	I	C11	PCI Clock
RST_I	I	A3	PCI RESET_N
REQ_O	O	B3	PCI REQ_N
GNT_I	I/O	C4	PCI GNT_N
CBE_IO<3>	I/O	D5	PCI Command / Byte Enable
CBE_IO<2>	I/O	E2	PCI Command / Byte Enable
CBE_IO<1>	I/O	E3	PCI Command / Byte Enable
CBE_IO<0>	I/O	E1	PCI Command / Byte Enable
AD_IO<31>	I/O	F5	PCI AD (Address / Data)
AD_IO<30>	I/O	F4	PCI AD (Address / Data)
AD_IO<29>	I/O	F3	PCI AD (Address / Data)
AD_IO<28>	I/O	F2	PCI AD (Address / Data)
AD_IO<27>	I/O	F1	PCI AD (Address / Data)
AD_IO<26>	I/O	G5	PCI AD (Address / Data)
AD_IO<25>	I/O	G4	PCI AD (Address / Data)
AD_IO<24>	I/O	G3	PCI AD (Address / Data)
AD_IO<23>	I/O	G2	PCI AD (Address / Data)
AD_IO<22>	I/O	G1	PCI AD (Address / Data)
AD_IO<21>	I/O	H5	PCI AD (Address / Data)
AD_IO<20>	I/O	H3	PCI AD (Address / Data)
AD_IO<19>	I/O	H4	PCI AD (Address / Data)
AD_IO<18>	I/O	H2	PCI AD (Address / Data)
AD_IO<17>	I/O	H1	PCI AD (Address / Data)
AD_IO<16>	I/O	J6	PCI AD (Address / Data)
AD_IO<15>	I/O	J4	PCI AD (Address / Data)
AD_IO<14>	I/O	J5	PCI AD (Address / Data)
AD_IO<13>	I/O	J3	PCI AD (Address / Data)
AD_IO<12>	I/O	J2	PCI AD (Address / Data)
AD_IO<11>	I/O	J1	PCI AD (Address / Data)
AD_IO<10>	I/O	K5	PCI AD (Address / Data)
AD_IO<9>	I/O	K6	PCI AD (Address / Data)
AD_IO<8>	I/O	K3	PCI AD (Address / Data)
AD_IO<7>	I/O	K4	PCI AD (Address / Data)
AD_IO<6>	I/O	K2	PCI AD (Address / Data)
AD_IO<5>	I/O	K1	PCI AD (Address / Data)
AD_IO<4>	I/O	L1	PCI AD (Address / Data)
AD_IO<3>	I/O	L3	PCI AD (Address / Data)
AD_IO<2>	I/O	L2	PCI AD (Address / Data)
AD_IO<1>	I/O	L4	PCI AD (Address / Data)
AD_IO<0>	I/O	L5	PCI AD (Address / Data)
FRAME_IO	I/O	L6	PCI FRAME_N
IRDY_IO	I	M1	PCI Initiator ready
TRDY_IO	O	M2	PCI Target ready
STOP_IO	I/O	M3	PCI STOP_N
PAR_IO	I/O	M4	PCI PAR
PERR_IO	I/O	M5	PCI_PERR_N
SERR_IO	I/O	M6	PCI_SERR_N
IDSEL_I	I	N1	PCI IDSEL
DEVSEL_IO	I/O	N2	PCI DEVSEL_N
INTA_O	O	N3	PCI INTR_A

Appendix A – Abbreviations

FPGA	Field Programmable Gate Array
R/W	Read and/or Write
R/O	Read Only
W/O	Write Only
N/A	Not Applicable
tbd	to be defined