PCI LVDS 8T

8 Channel LVDS Serial Interface

Dynamic Engineering
435 Park Drive, Ben Lomond, CA 95005
831-336-8891
www.dyneng.com

This document contains information of proprietary interest to Dynamic Engineering. It has been supplied in confidence and the recipient, by accepting this material, agrees that the subject matter will not be copied or reproduced, in whole or in part, nor its contents revealed in any manner or to any person except to meet the purpose for which it was delivered.

Dynamic Engineering has made every effort to ensure that this manual is accurate and complete. Still, the company reserves the right to make improvements or changes in the product described in this document at any time and without notice. Furthermore, Dynamic Engineering assumes no liability arising out of the application or use of the device described herein.

The electronic equipment described herein generates, uses, and can radiate radio frequency energy. Operation of this equipment in a residential area is likely to cause radio interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Dynamic Engineering's products are not authorized for use as critical components in life support devices or systems without the express written approval of the president of Dynamic Engineering.

This product has been designed to operate with PCI and compatible user-provided equipment. Connection of incompatible hardware is likely to cause serious damage.

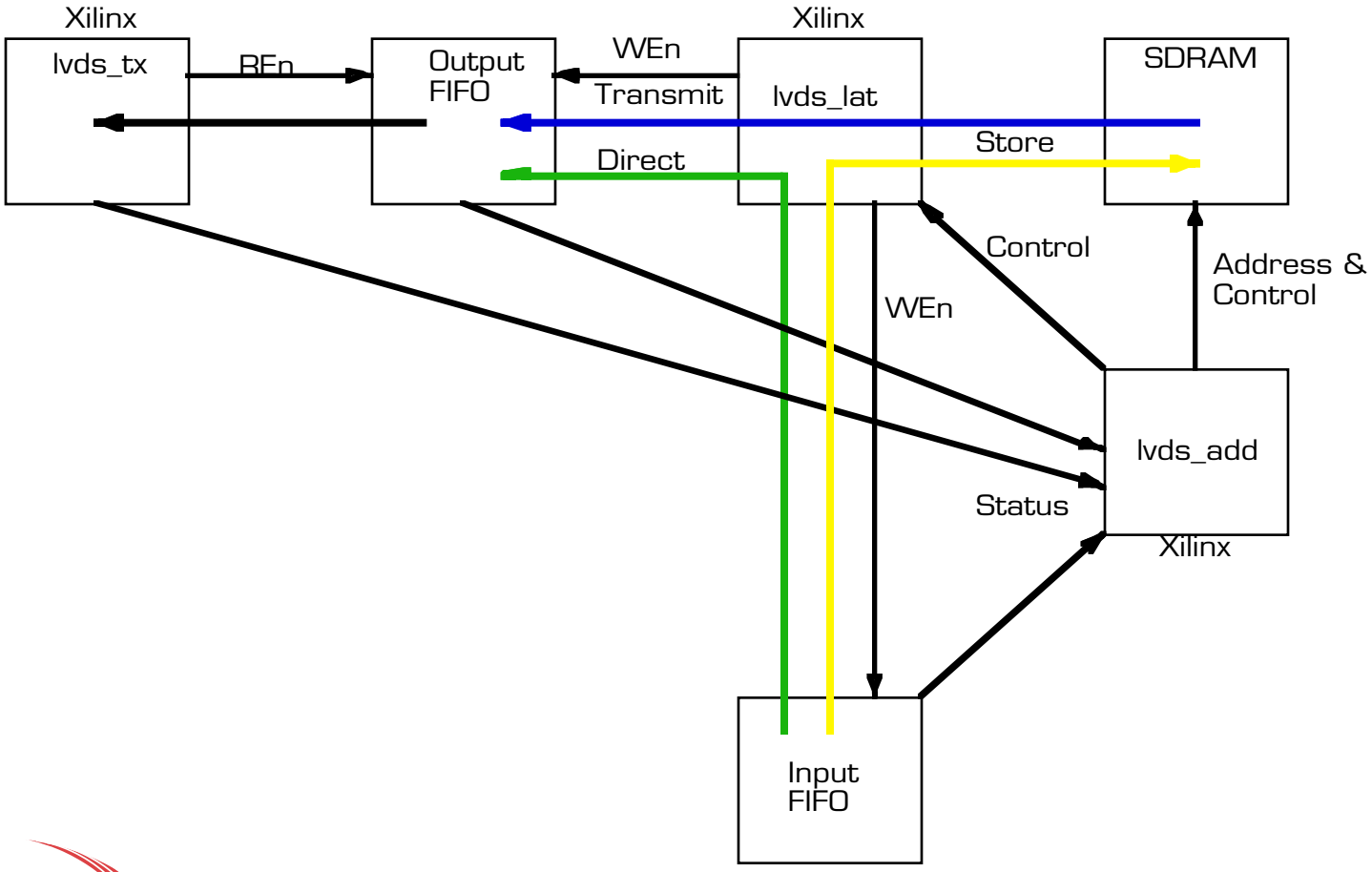
©2000,2001,2002 by Dynamic Engineering
Manual Revision D. Revised 4/10/02
10-2001-0903

Table of Contents

List of Figures

Introduction

FIGURE 1 PCI_LVDS_8T DATA FLOW DIAGRAM



The PCI_LVDS_8T has three basic modes of operation. Data can be stored [store] into the bulk memory [SDRAM], Data can be retrieved from the SDRAM and transmitted [transmit] or data can be directly sent from the PCI interface to the output [direct]. The Data flow is supported by FIFOs and embedded state-machines.

There are 8 channels of LVDS supported. Each pair of LVDS channels is controlled by a "TX" Xilinx. There are 4 TX Xilinx installed per PCI_LVDS_8T. The TX Xilinx will read the data from the Output FIFO, convert from 32 bit to 16 bit data, do the final data processing and stay in synchronization with the LVDS reference clock. Each of the LVDS channels has a separate Output FIFO.

The Output FIFO is sourced from the Input FIFO or the SDRAM via the Latch Xilinx. One Latch Xilinx supports 4 LVDS channels. There are two Latch Xilinx installed per PCI_LVDS_8T. The Latch Xilinx provides the data bus width increase and reduction to match the 32 bit FIFOs with the 64 Bit SDRAM. The Data path between the FIFOs operates at 66 MHz and the path between the Latch and the SDRAM operates at 33 MHz. The Latch design is supported by the Address Generator Xilinx.

The Address Generator Xilinx creates the control for the SDRAM and Latch Xilinx devices. The Address Generator creates the start-up sequence, refresh, burst read and burst write controls for the SDRAM, plus the control signals for the Latch Xilinx.

The Address Generator works with a round-robin arbitration scheme looking at the Output FIFO's half full flag and start bit to determine which channel to transfer to next. Data is moved into the Output FIFO at 66 MHz as 32 bit data. Data is removed as 16 bit data at 25 Mhz [Max] giving the Address Generator more than 5:1 ratio on any one channel providing the necessary margin to support the 4 channels per Address Generator plus the SDRAM overhead [refresh and arbitration]. There are two Address Generator Xilinx's per PCI_LVDS_8T.

There are 9 Xilinx devices total including the DMA Xilinx, which handles the basic decoding for the board and the DMA interface into the Input FIFO.

The modes of operation are supported by clocking and starting options. Internal and external clock and start pulses can be used to control the frequency and synchronization.

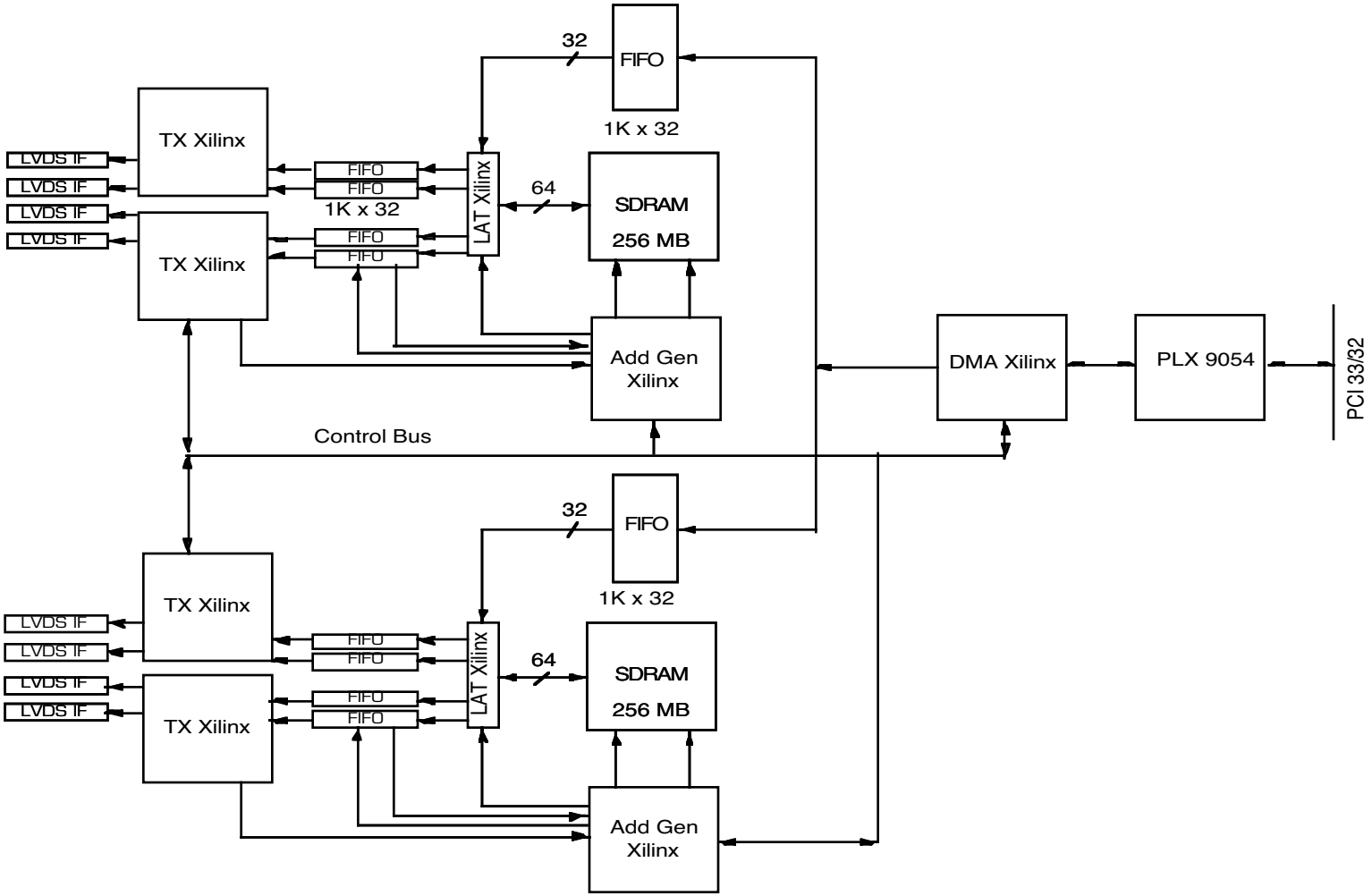
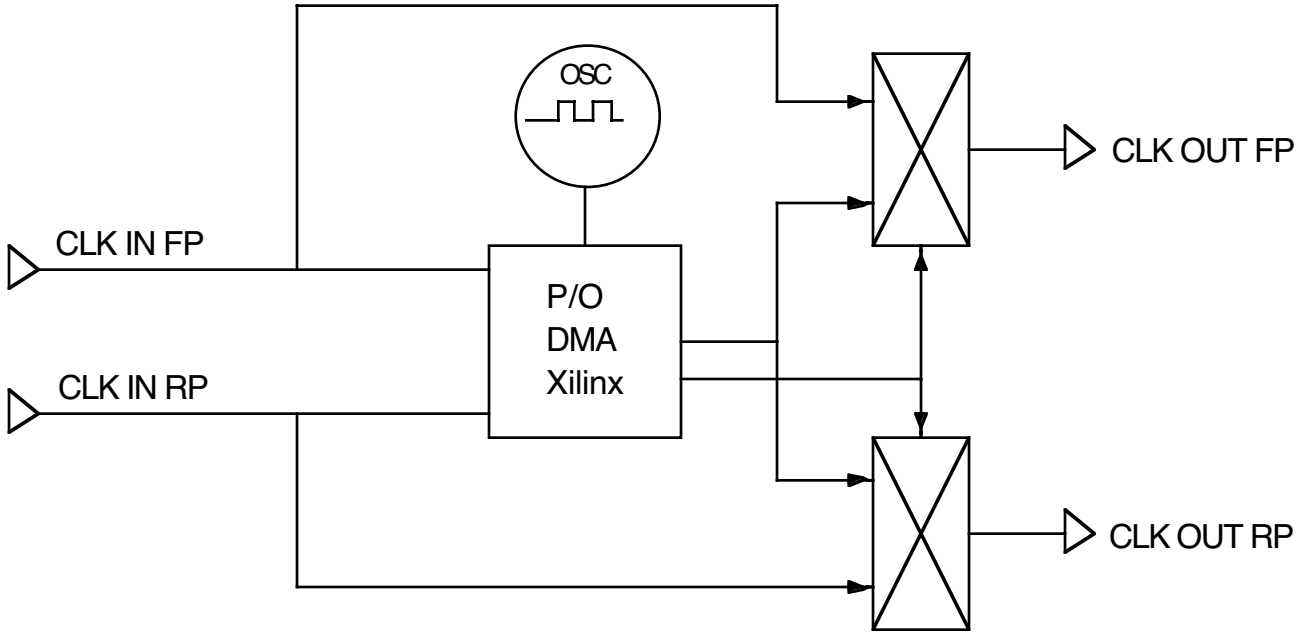


FIGURE 2 PCI_LVDS_8T BLOCK DIAGRAM

FIGURE 3 PCI_LVDS_8T REFERENCE CLOCK DISTRIBUTION



The '8T can select the FP [front panel], RP [rear panel] or on-board clock references. A typical configuration would be to use the FP input to one card within a chassis and route that to the RP. The rest of the cards within the chassis can use the RP connectors for clock distribution. The FP connectors can be used between chassis and the RP within the chassis for clean wiring. The clock and Start pulse distribution is accomplished with LVDS drivers and receivers plus high speed muxes for a low skew distribution architecture.

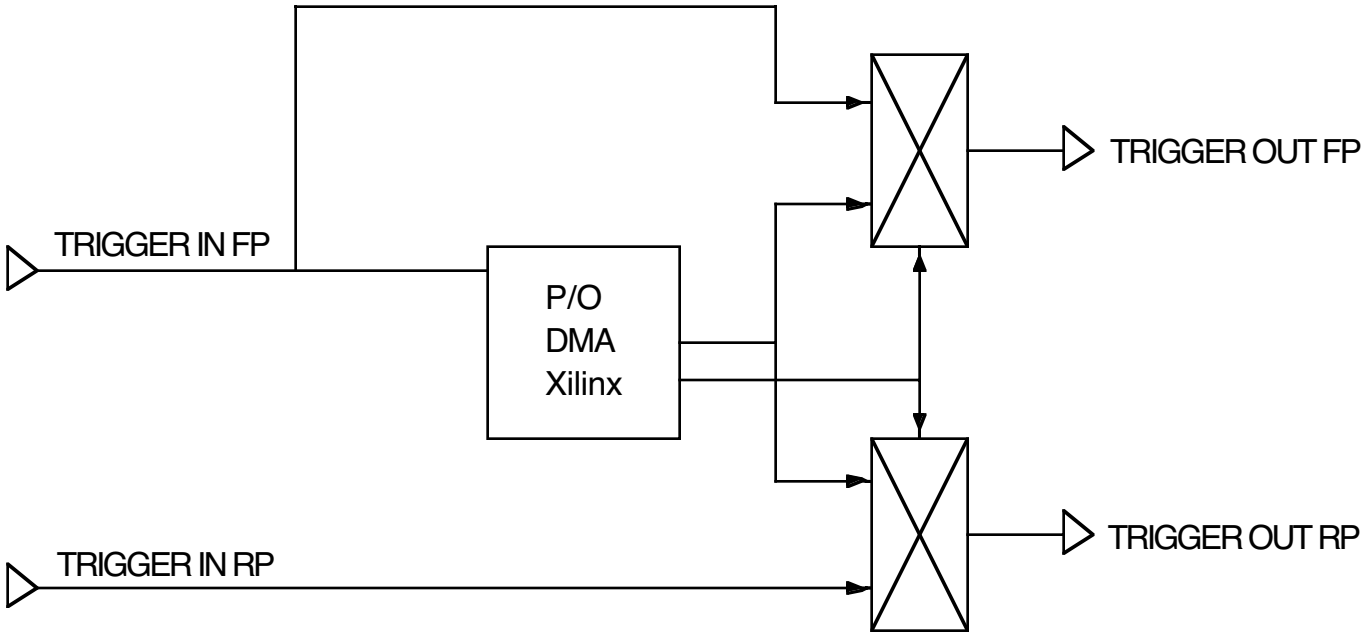


FIGURE 4 PCI_LVDS_8T TRIGGER DISTRIBUTION

The Start pulse has a similar mechanism for distribution. The start pulse is synchronized to the reference clock and distributed directly to the TX Xilinx devices. The local trigger can be generated via software command or external triggers can be utilized to allow for master slave chaining. The FP external trigger is routed to the DMA Xilinx to allow an external trigger to be routed through to the RP connector.

Memory Map

(Addresses shown as byte)

TX Channels 0,1

Decode number	Address offset	Chip	Definition
0	0000	TX01	CNTL_STAT_0
	0004	TX01	DATA_RDBACK_0
	0008	TX01	W_COUNT_0
	000C	TX01	X_COUNT_0
	0010	TX01	Y_COUNT_0
	0014	TX01	Z_COUNT_0
	0018	TX01	IDLE_0_0
	001C	TX01	IDLE_1_0
	0020	TX01	CNTL_STAT_1
	0024	TX01	DATA_RDBACK_1
	0028	TX01	W_COUNT_1
	002C	TX01	X_COUNT_1
	0030	TX01	Y_COUNT_1
	0034	TX01	Z_COUNT_1
	0038	TX01	IDLE_0_1
	003C	TX01	IDLE_1_1

TX Channels 2,3

Decode number	Address offset	Chip	Definition
1	0040	TX23	CNTL_STAT_2
	0044	TX23	DATA_RDBACK_2
	0048	TX23	W_COUNT_2
	004C	TX23	X_COUNT_2
	0050	TX23	Y_COUNT_2
	0054	TX23	Z_COUNT_2
	0058	TX23	IDLE_0_2
	005C	TX23	IDLE_1_2
	0060	TX23	CNTL_STAT_3
	0064	TX23	DATA_RDBACK_3
	0068	TX23	W_COUNT_3
	006C	TX23	X_COUNT_3
	0070	TX23	Y_COUNT_3
	0074	TX23	Z_COUNT_3
	0078	TX23	IDLE_0_3
	007C	TX23	IDLE_1_3

Address Generator Channels 0-3

Decode number	Address offset	Chip	Definition
2	0080	ADDO_3	ADD CH 0 A
	0084	ADDO_3	ADD CH 1 A
	0088	ADDO_3	ADD CH 2 A
	008C	ADDO_3	ADD CH 3 A
	0090	ADDO_3	ADD CH 0 B
	0094	ADDO_3	ADD CH 1 B
	0098	ADDO_3	ADD CH 2 B
	009C	ADDO_3	ADD CH 3 B
	00A0	ADDO_3	ADD CH 0 C
	00A4	ADDO_3	ADD CH 1 C
	00A8	ADDO_3	ADD CH 2 C
	00AC	ADDO_3	ADD CH 3 C
	00B0	ADDO_3	ADD CH 0 D
	00B4	ADDO_3	ADD CH 1 D
	00B8	ADDO_3	ADD CH 2 D
	00BC	ADDO_3	ADD CH 3 D
3	spare		

TX Channels 4,5

Decode number	Address offset	Chip	Definition
4	0100	TX45	CNTL_STAT_4
	0104	TX45	DATA_RDBACK_4
	0108	TX45	W_COUNT_4
	010C	TX45	X_COUNT_4
	0110	TX45	Y_COUNT_4
	0114	TX45	Z_COUNT_4
	0118	TX45	IDLE_0_4
	011C	TX45	IDLE_1_4
	0120	TX45	CNTL_STAT_5
	0124	TX45	DATA_RDBACK_5
	0128	TX45	W_COUNT_5
	012C	TX45	X_COUNT_5
	0130	TX45	Y_COUNT_5
	0134	TX45	Z_COUNT_5
	0138	TX45	IDLE_0_5
	013C	TX45	IDLE_1_5

TX Channels 6,7

Decode number	Address offset	Chip	Definition
5	0140	TX67	CNTL_STAT_6
	0144	TX67	DATA_RDBACK_6
	0148	TX67	W_COUNT_6
	014C	TX67	X_COUNT_6
	0150	TX67	Y_COUNT_6
	0154	TX67	Z_COUNT_6
	0158	TX67	IDLE_0_6
	015C	TX67	IDLE_1_6
	0160	TX67	CNTL_STAT_7

0164	TX67	DATA_RDBACK_7
0168	TX67	W_COUNT_7
016C	TX67	X_COUNT_7
0170	TX67	Y_COUNT_7
0174	TX67	Z_COUNT_7
0178	TX67	IDLE_0_7
017C	TX67	IDLE_1_7

Address Generator Channels 4-7

Decode number	**Address offset**	**Chip**	**Definition**
6	0180	ADD4_7	ADD CH 4 A
	0184	ADD4_7	ADD CH 5 A
	0188	ADD4_7	ADD CH 6 A
	018C	ADD4_7	ADD CH 7 A
	0190	ADD4_7	ADD CH 4 B
	0194	ADD4_7	ADD CH 5 B
	0198	ADD4_7	ADD CH 6 B
	019C	ADD4_7	ADD CH 7 B
	01A0	ADD4_7	ADD CH 4 C
	01A4	ADD4_7	ADD CH 5 C
	01A8	ADD4_7	ADD CH 6 C
	01AC	ADD4_7	ADD CH 7 C
	01B0	ADD4_7	ADD CH 4 D
	01B4	ADD4_7	ADD CH 5 D
	01B8	ADD4_7	ADD CH 6 D
	01BC	ADD4_7	ADD CH 7 D

7,8,9 spare

PLX Interface, Decode and Control

Decode number	Address offset	Chip	Definition
10	0400		DMA Base Control r-w
	0404		DMA Timing Control r-w
	0408		
	040C		
	0410		
	0414		
	0418		
	041C		
	0420		
	0424		
	0428		
	042C		
	0430		
	0434		DMA Xilinx programmed status read
	0438		DMA FIFO slave write
	043C		DMA functional Status read / DMA Status Clear write
	2XXX		DMA Data Read
11..7F		spare	

DMA Definitions

DMA Base Control

$0400
Read – Write

Bit#	Definition
0	Reset_0
1	Reset_1
2	Reset_2
3	Reset_3
4	Reset_4
6-5	spare
7	LED control
8	Write_EN_STD
9	Write_EN_DMA
10	Channel
15-11	spare
16	Int En 0
17	Int En 1
18	Int En 2
19	Int En 3
20	Int En 4
21	Int En 5
22	Int En 6
23	Int En 7
24	Interrupt master en
25	Force Int

Reset_0 when 0 resets the TX Xilinx devices (4). When '1' enables the TX Xilinx devices.

Reset_1 when 0 resets the Output FIFOs (16). When '1' enables the Output FIFOs. The FIFOs must be enabled then reset then re-enabled as part of initialization. The clock selection should be to PCI clock for this operation. Please refer to the Clock Control description for more details.

Reset_2 when 0 resets the Latch Xilinx devices (2). When '1' enables the Latch Xilinx devices.

Reset_3 when 0 resets the Address Generator Xilinx devices (2). When '1' enables the Address Generator Xilinx devices.

Reset_4 when 0 resets the Input FIFO devices (4). When '1' enables the Intput FIFO devices. The FIFOs must be enabled then reset then re-enabled as part of initialization.

Int En X. When set '1' and the corresponding Done bit is received active the interrupt to the host is asserted via the PLX device. The PLX will also have to be enabled to cause an interrupt. The master interrupt within the DMA Xilinx will need to be enabled. Clear the interrupt by clearing the done bit or masking off with the enable. Until the done bit is cleared do not re-enable the interrupt source.

Interrupt master en when 1 enables the DMA Xilinx to assert an interrupt request to the PLX chip and in turn to the PCI bus. The PLX chip has a bi-directional interrupt request line which must be programmed to be an input before setting the Interrupt master en. A logic conflict will exist if the PLX device is not properly programmed. Default is '0'.

Force Int when '1' will cause an interrupt to be set. Also requires Interrupt Master enable and PLX interrupt enable. Useful for software debugging and test purposes. Clear by setting low.

Write_EN_STD when '1' enables the write state-machine in standard mode. The State Machine will move data from the internal register to the FIFO when a slave write is detected. The Channel and enable must be set prior to the writing.

Write_EN_DMA when '1' enables the write state-machine in the DMA mode. When the PLX device accesses to the DMA write address space, the state-machine starts up and asserts READYn. Data is continuously supplied by the PLX device until BLASTn is asserted. [1 data word per PCI clock to support a burst transfer]. When the burst is completed and BLASTn is asserted the PLX has completed the burst write. There are still two data words within the internal pipeline. The state-machine will transfer the last two words then return to wait for the next burst. The state-machine assumes that the address generator has been started prior to the DMA – there will always be room within the Input FIFO. The FIFO error [see status register] indicates that the FIFO went full at some point. When additional reads from the DMA read address take place the process restarts. Due to PCI requirements the burst length is limited to approximately 250 clocks per burst. The fill rate is the PCi rate = 33 Mhz and the transfer rate to SDRAM is 66 MHz.

Only one of the two writes should be selected at a time.

Channel when 'O' select Input FIFO 0 which corresponds to channels 0-3. When '1' selects Input FIFO 1 corresponding to channels 4-7. Channel should be selected before selecting Write Std or Write DMA. Unexpected results will occur otherwise.

LED Control when '1' will turn on the LED and when 'O' will turn off the LED.

DMA Timing Control

$0404

Read – Write

Bit#	Definition
0	pulse out enable [pls_out_en]
1	master enable 1 = enabled, 0 = disabled [master_en]
2	master_slave [mas_slv]
3	spare
4	pulse out select 1 = use FP external trigger 0 = use local pulse out en
15-5	spare
27-16	divisor
28	RS – clock post selector
29	spare
31-30	PS – clock pre-selector

Clock Pre-Selector

00	oscillator
01	front panel [FP] external clock source
10	rear panel [RP] external clock source
11	pci clock

The clock pre-selector is used to select which reference clock to use with the divisor hardware. [The clock source]

Divisor [11-0] are the clock divisor select bits. The clock source is divided by a counter. The reference clock for the counter is selected with the CLK Pre-Selector. The output frequency is {reference / [2(n+1)]}. N≥1. The reference oscillator is 16.56 MHz.. The counter divides by N+1 due to counting from 0 ->n before rolling over. The output is then divided by 2 to produce a square wave output.

Post Selector when '1' sets clock out to clock divided, when '0' sets clock out to pre-selector reference value. When operating with the oscillator or external sources the undivided option is usually correct. When selecting the

PCI source the divider should be used.

MAS_SLV when ‘0’ sets the hardware to slave mode. In slave mode the Start pulse is received, buffered, and retransmitted out along with the reference clock at each of the external connectors. The clocks are also routed to the DMA Xilinx to be used as references locally. In Master mode the clock and start pulse inputs are ignored and the local version is driven off board. Please note that the TX parts will need to be properly programmed for clock source and enable. The default value is slave. It is intended that all boards except the designated master are slaves. The source of the clock is selected separately. To receive a clock in the FP and distribute to the RP connector; choose the master designation and use the clock select to use the FP reference clock.

Master_En when ‘1’ provides a local master enable to the TX FPGAs. The Master enable for each board should be set after the Add_Gen and TX devices are programmed. If the TX parts are programmed to wait for an external trigger, then the master_en will enable the channels to react to the external trigger. When the external trigger is received, multiple cards can be synchronized to transmit. If the channels are programmed to operate without an external trigger then the local channels will be enabled together with a common clock allowing for coherent start-up across 8 channels

The source for the driven external triggers is selectable. When in master mode the external trigger is driven from a local source. The **Pulse Out Select** bit is used to select the external FP trigger [in] or the local Pulse Out En source. The purpose for the selection is to allow the FP trigger to be used as the source and the RP and FP outputs driven from that source. An external master trigger can then be used to control a chassis without having to wire to all of the FP connectors. The RP headers can be used for a low skew, “neat” wiring implementation. If the Pulse Out Select bit is cleared then the local Pulse Out En generated pulse can be used to cause a synchronous start-up across multiple cards.

When Pulse Out Select = ‘1’ the Master_En is held off until the FP external trigger is received. When the trigger is received the Master_En is asserted along with the PLS_OUT. The added Flip-Flop in the path on the slave cards is accounted for with a 1 clock skew between the Master_En and PLS_OUT. When the channels have completed the transmission, the Master_En should be set to ‘0’, then ‘1’ to rearm for the next external trigger pulse.

When Pulse Out Select = ‘O’ the internal controls govern. Master_en is driven when it is set [no hold-off] and Pulse_Out_En controls the Pulse Out signal.

PLS_OUT_EN when set causes a pulse to be generated. If in Master mode this pulse is driven off card to provide a start pulse for external hardware. The master enable for the local channels should be set at the same time on the card designated as master. The local TX parts will use the master enable. The slave cards will use the start pulse. The pulse enable is self-clearing. Set to ‘1’ to cause the pulse.

To use an external clock and trigger from the front panel and with rear panel distribution make the following selections:

1. Select one card for the point of entry and designate that as the master
2. Set the remaining cards to slaves
3. Select the FP clock reference on the master card
4. Select the RP clock reference on the slave cards [or FP if desired to use that connector]
5. Set Pulse Out Select = ‘1’ on the Master card and = ‘O’ on Slave cards
6. Set the TX chips to wait for the external trigger on slave cards [select RP or FP depending on your wiring]
7. Set the TX chips on the master card to start with the local master enable.

The PCI_LVDS_8T is designed to offset the master enable from the trigger out signal to account for the input FF [synchronization stage] that the slave cards have. All cards can start on the same clock. The clock itself will have skew due to the transmission chain. The specification for the design - 4 clock periods across cards is met under all conditions.

The LVDS receivers and Drivers are fast, and do have an inherent delay. Using the FP or RP distribution will create a delay between the cards. The individual cards re-synchronize the trigger pulse in parallel with the buffering. The LVDS driver has 1.5 nS max delay, the LVDS receiver has 2.5 nS max delay, and the MUX adds .25 nS plus trace length of less than 1 nS => plus cable delay ~6nS delay. At 25 MHz reference rate = 40 nS period 6-7 cards would be the cards per clock period. The targeted chassis has 10 cards in parallel for 9 delays and approximately 54 nS between the master and last slave [worst case]. 54 nS is 1-2 clock periods. The clock and start pulse follow the same path and have the same worst case delays. The clock and trigger utilize two halves of dual driver and receiver pairs with low skew figures. The trigger is 2 periods wide to prevent the skew

from causing a missed start.

The channels can be started with less skew between them by using an external clock and trigger distribution instead of the buffer / repeater network that is built into the cards.

DMA Status

Bit#	Definition
0	Done channel 0
1	Done channel 1
2	Done channel 2
3	Done channel 3
4	Done channel 4
5	Done channel 5
6	Done channel 6
7	Done channel 7
8	FIFO_0_Err
9	FIFO_1_Err
10	gnd
11	gnd
12	gnd
13	gnd
14	gnd
15	Interrupt RQST
23-16	SW7-0
24	gnd
25	gnd
26	mt_outOn
27	hf_outOn
28	ff_outOn
29	mt_out1n
30	hf_out1n
31	ff_out1n

mt_xn is active low. '0' – FIFO is empty

hf_xn is active low ‘0’ – FIFO is half full or more
ff_xn is active low ‘0’ – FIFO is full.

0 corresponds to channels 0-3, and 1 to channels 4-7.

Done channel X when 1 is done meaning that the requested samples have been transferred. The Data has been captured and stored into SDRAM, or read from SDRAM to the output FIFO as programmed. The done bits are cleared by writing a ‘1’ to the corresponding position and then writing with a ‘0’ to enable. The done bits are used to create an interrupt to the host if the corresponding interrupt enable bit is set. The bits should be cleared after the Xilinx’s are enabled from reset to clear any transition induced status changes.

FIFO_X_Err when ‘1’ indicates that the Output FIFO for that channel has become full at some point. In Retrieve mode this is not a problem. In Direct mode this is an overflow error. Clear by writing with the corresponding bit set to ‘1’.

SW7-0 reflect the settings of the user defined dip-switch on the board. It is envisioned that the switch is used as a board level “address” to identify a specific slot and cable with a particular device number.

DMA FIFO Holding Register

$0438

Bit#	Definition
31-0	Input FIFO data

Write data stored to the Input FIFO. Select the Input FIFO to write with the Channel definition. Enable the process with Write_EN_STD. The data is automatically written when a write to the holding register is detected. The hardware overlaps the write to the register and update of the FIFO to reduce the access time from the PCI bus.

DMA Xilinx Status

$0434

Bit#	Definition
0	DN01
1	DN23
2	DN45
3	DN67
4	DNL0
5	DNL1
6	DNA0
7	DNA1

DNx are the done bits from the Xilinx devices. After initialization the Done signal should be '1'.

DMA FIFO Holding Register

$2xxx

Bit#	**Definition**
31-0	Output FIFO data

Write the data stored to the Input FIFO. Select the Input FIFO to write to with the Channel definition. Enable the process with Write_EN_DMA. When enabled and a write to this address occurs the DMA engine within the Xilinx is started. The data is written into the holding register, and then moved to the FIFO. An additional latch is used to create a three deep pipeline. The initial READY signal is held off until the pipeline is ready for continuous data. The data within the register is updated on each clock until the BLASTn signal is asserted by the PLX device. BLASTn will be asserted when the Burst transfer from the PCI bus is halted due to length of transfer or software intervention. The PLX device requires about 6 clocks to arbitrate for the bus and start the transfer. The max

time permitted for a transfer is 256 clocks leaving about 250 data transfers per burst. The transfer is interrupted with the BLASTn signal. The 8T design completes the writes of the data left in the pipeline when BLASTn is detected. The pipeline can handle 1,2, 3+ length bursts. During the Process the software should not attempt to access the PCI_LVDS_8T.

Address Generator Definitions

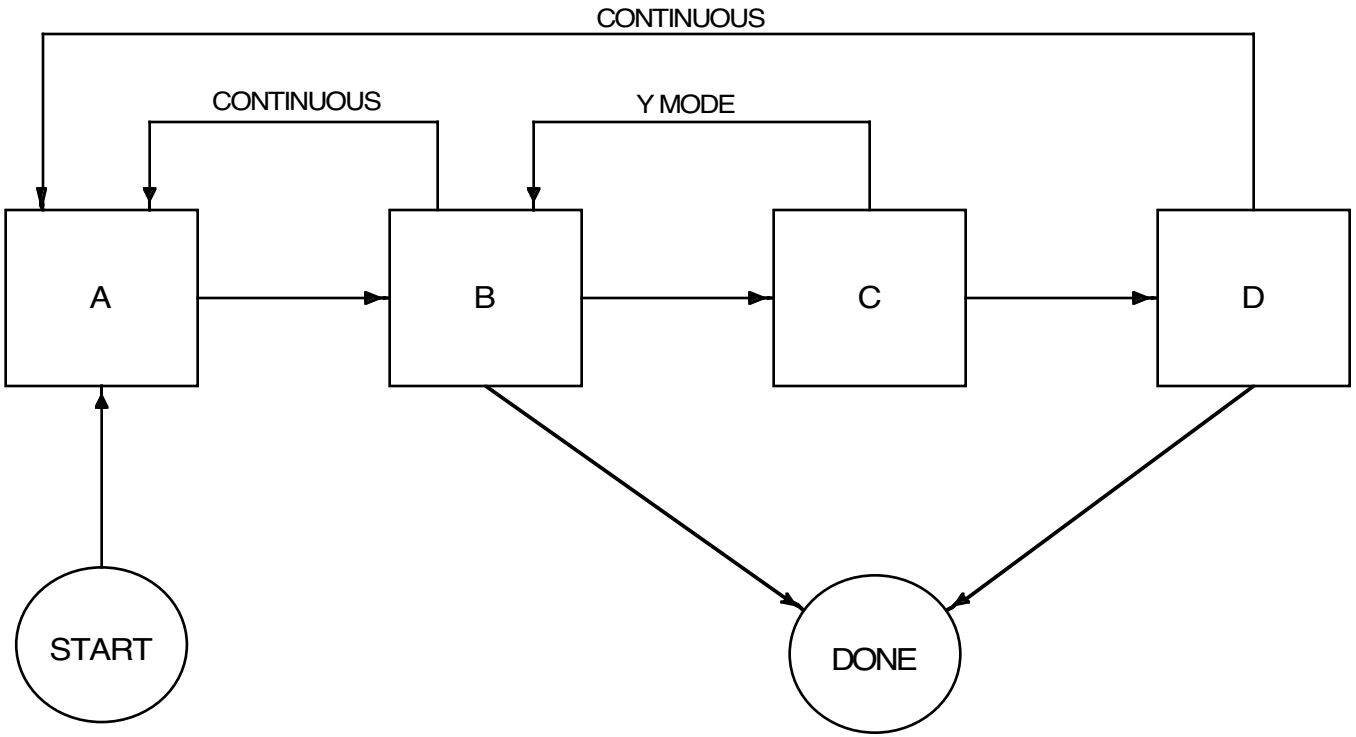


FIGURE 5 PCI_LVDS_8T ADDRESS GENERATOR ADDRESS LOOP OPTIONS

A,B,C,D represent the Address Generator registers A-D. Each register is available for each channel. The Address generator can be programmed to sequence: 1) from A to B and stop 2) from A to B and then loop

starting at A again 3) from A to B the loop from B to C "Y" times, then progress to D 4) same as 3 but loop from D to A. The following register definitions provide the programming details.

Address Generator SDRAM Address A Registers

0080	ADD0_3	ADD CH 0 A
0084	ADD0_3	ADD CH 1 A
0088	ADD0_3	ADD CH 2 A
008C	ADD0_3	ADD CH 3 A
0180	ADD4_7	ADD CH 4 A
0184	ADD4_7	ADD CH 5 A
0188	ADD4_7	ADD CH 6 A
018C	ADD4_7	ADD CH 7 A

Read – Write

Bit#	**Definition**
31	Start
30	Continuous
29	Y Count En
28	Store_transmit
27	Direct
26 - 25	unused / undefined
24-0	Initial address to access SDRAM for Channel x

Eight registers at different offsets with the same bit definitions provided for the 0..7 channels.

The Address A register is the address to start with for accessing SDRAM. The SDRAM is organized as 64 bit words. The addresses increment with groups of 8 bytes. To select offset 8M bytes the address would be 1M long words. With a range of 24-0 the entire 256 Mb address space is selectable by any channel. All transfers will start and stop on 64 bit boundaries [by hardware definition]. In no case are smaller than 64 bit data words written to or read from the SDRAM.

Start when '1' causes that channel to start-up. Up to 4 channels can operate in Transmit mode. Only channel 0 or 4 can be used for Store, and any one channel can be used for Direct mode.

Continuous when set for a channel causes the channel to re-start at the end of the programmed sequence. When '0' the done bit will be set and the programmed operation stop at the end of the sequence. If started with the continuous bit set and later the bit is cleared then the process will terminate at the end of the current sequence.

Y Count En when '1' causes the hardware to use the Y count for that channel and to loop from address B to C – Y times.

Y times
<-
A -> B -> C -> D With Y Count En = '1' the address will sequence from A to B then to C. At C if the Y count is not at the terminal count then the next address will be B. When the terminal count is reached the process will continue on to D. At D if the continuous bit is set the next address will be A. If the continuous bit is not set then the process is complete and the start bit will be cleared and the done bit set.

With Y Count En = '0' the start address is A and the stop address is B. With the continuous bit set the sequence will repeat.

There are no restrictions for page boundaries of the SDRAM. If the initial starting address is placed just before an SDRAM page boundary, then the hardware will process the first part of the transfer as a sequence of individual transfers then go to burst mode at the page boundary. If the length causes the address to cross a page boundary then the pipeline will detect this and stop prior to the page boundary, insert the proper control cycles and restart the burst on the next page. The hardware can handle the different combinations of starting, finishing, and crossing addresses automatically and at full rate as defined in the specification. If attempting to operate outside of the specification, then pay attention to the boundary conditions to increase performance. Pages are 0x00-0x3ff.

Any combination of channels can be activated for transmit mode. Only Channel 0 or 4 can be used for Store and any one channel can be used for Direct mode. The channels should be activated before the corresponding

channels on the Transmit Xilinx in transmit mode to make sure that there is data stored in the output FIFO. The Store command must be started before the DMA sequence is initialized for loading data into SDRAM.

Address Generator SDRAM Address B Registers

0090	ADD0_3	ADD CH 0 B
0094	ADD0_3	ADD CH 1 B
0098	ADD0_3	ADD CH 2 B
009C	ADD0_3	ADD CH 3 B
0190	ADD4_7	ADD CH 4 B
0194	ADD4_7	ADD CH 5 B
0198	ADD4_7	ADD CH 6 B
019C	ADD4_7	ADD CH 7 B

Read – Write

Bit#	**Definition**
31	Initialization – Channel 0,4 B only
30	Load Complete – Channel 0,4 B only
29-28	unused / undefined
27	FF – read only FIFO Full Status
26	HF – read only FIFO Half Full Status
25	MT – read only FIFO Empty Status
24-0	Address Channel X register B definition

Eight registers at different offsets with the same bit definitions provided for the 0..7 channels.

The FIFO status bits are active low and read only.

The B register address is only for the termination or loop restart address depending on the Y Count En control bit. The Address is included in the transmission when used as a terminal address. [Up to and including] Addresses are in 64 bit words.

The SDRAM requires initialization. The initialization involves a pre-charge cycle followed by a register write and then two refresh cycles. The hardware will stay in the low power undefined state until the initialization bit is set high. The SDRAM has an internal register which is controlled by writing to it with the "data" placed on the address lines. Channel 0 Register A has been selected as the initialization source for the "data". **Prior to setting the initialization bit the data 0x27 must be written into the CH 0 A register**. During the initialization cycle the 0x27 will be transferred to the SDRAM to set the CAS delay to 2 cycles and the burst size to page size. The hardware automatically will do the pre-charge, register write and refresh cycles. Once the initialization bit is set the software should delay for 2 uS before setting the channel 0 address to correspond to the channel 0 definition. The other channels can be set earlier because they are not involved with initialization. A further delay of 100 uS is required before using the SDRAM to allow for proper initialization.

After initialization the hardware enters a series of states called init_refresh where the SDRAM is ready to go but a command has not been detected. A counter delays for 256 clocks then a refresh occurs in an endless loop. When a command [start] has been set then the hardware will begin processing that command.

Load Complete [LC] when set tells the Store statemachine that the last of the data to transfer from the Input FIFO to the SDRAM is available. The State-Machine will ignore the state of the HF flag and read the remaining data to complete the data Store operation. This bit is used once the DMA has completed the transfer from host memory into the PCI_LVDS_8T. The data at the end of the transfer will be stuck in the FIFO because the state machine will not transfer data until there is at least 1/2 FIFO to transfer. The last 1/2 FIFO or less will be transferred as a final burst when the LC bit is set. This bit is not self cleared.

Address Generator SDRAM Address C Registers

00A0	ADD0_3	ADD CH 0 C
00A4	ADD0_3	ADD CH 1 C
00A8	ADD0_3	ADD CH 2 C
00AC	ADD0_3	ADD CH 3 C
01A0	ADD4_7	ADD CH 4 C
01A4	ADD4_7	ADD CH 5 C
01A8	ADD4_7	ADD CH 6 C
01AC	ADD4_7	ADD CH 7 C

Read – Write

Bit#	**Definition**
31-25	Y Count
24-0	Address C definition channel X

Eight registers at different offsets 0..7 channels.

The Address C definition is used when the Y Count En = '1'. The address represents the terminal address for the loop B-> C. The loop operates up to and including address C. In loop mode the next address after C will be B or C+1 if the loop is completed and the sequence is progressing toward D.

The Y count is the upper 7 bits of the C Address Register. The Count is tested at Address C when the Y Count En = '1'. If Y Count En = '0' the process stops at B with the loop count, and registers C and D ignored. The count range is 1-7F. The loop counts at B and tests at C. A '0' value would be 128 loops. If the continuous bit is set then the Y loop is repeated Y count times each time the main loop is repeated; indefinitely, stopped when the Start bit is cleared or the continuous bit is cleared at the end of the process.

Address Generator SDRAM Address D Registers

00B0	ADD0_3	ADD CH 0 D
00B4	ADD0_3	ADD CH 1 D
00B8	ADD0_3	ADD CH 2 D
00BC	ADD0_3	ADD CH 3 D
01B0	ADD4_7	ADD CH 4 D
01B4	ADD4_7	ADD CH 5 D
01B8	ADD4_7	ADD CH 6 D
01BC	ADD4_7	ADD CH 7 D

Read – Write

Bit#	**Definition**
24-0	Address D definition channel X

Eight registers at different offsets 0..7 channels.

The Address D definition is used when the Y Count En = '1'. The address represents the terminal address for the process. The loop operates up to and including address D. In Continuous mode the next address after D is A. In standard mode the start bit is reset and the done bit is set once D is reached on a per channel basis.

TX Definitions

TX CNTL_STAT Definition Registers

0000	TX01	CNTL_STAT_0
0030	TX01	CNTL_STAT_1
0040	TX23	CNTL_STAT_2
0070	TX23	CNTL_STAT_3
0100	TX45	CNTL_STAT_4
0130	TX45	CNTL_STAT_5
0140	TX67	CNTL_STAT_6
0170	TX67	CNTL_STAT_7

Read – Write

Bit#	Definition
0	Start
1	Restart
2	Repeat Mode En
3	Serializer Enable
4	E1 Enable
5	E2 Enable
6	Read-Back mode
7	clock select
8	error – read only status bit
9	done – read only status bit

Serializer enable : When ‘0’ the serializer is in power down mode. When ‘1’ the serializer is enabled. The

serializer should be enabled with the idle pattern for 10 mS before starting a data transfer to allow the PLL to properly synchronize with the reference clock selected.

Start when '1' will start the transmitter sending data. The process assumes that the Address generator has been started before and that data is available in the Output FIFO. The process will continue until the done bit is received from the Address Generator and the FIFO becomes empty. If the FIFO is empty before the done bit is received the error flag is set. The Idle Pattern 0 is transmitted until the Start bit is set. Once set the pattern will become the data pattern until the end of the transmission. At the end of the transmission the second Idle Pattern will be transmitted.

Repeat Mode when '1' enables the TX chip to interpret the upper data bits and to insert extra data based on the value found. The data is repeated for a programmed number of times. W,X,Y,Z registers are contain the counts.

Bits15,14	
00	W
01	X
10	Y
11	Z

The data is repeated for the programmed count on a per word basis. When Repeat Mode En = '0' the upper bits are ignored.

The local Start and Master start [from DMA Xilinx] are always used to transmit. In addition the E1 or E2 external enables can be required. When E1 Enable = '1' then the transmitter for that channel will not transmit until the later of Start, Master and E1 are all set. Similarly if E2 enable is set then the transmission is held off until the E2 external enable is received. If both bits are '0' then the master or Start will control when data is transmitted.

Read-Back Mode is selected to be able to read data from the FIFO to the internal control bus for loop-back purposes.

The Clock select is used to select the reference clock for the TX and FIFO to be the PCI clock or the muxed external or local reference clocks. See DMA Xilinx for more details on clock selection for operational modes. The PCI clock should be selected for the data loop-back mode.

The Error bit will be set and held when various error conditions occur.

The Done bit is a transitory bit and is only useful for factory test.

TX W Registers

0008	W_COUNT_0
0028	W_COUNT_1
0048	W_COUNT_2
0068	W_COUNT_3
0108	W_COUNT_4
0128	W_COUNT_5
0148	W_COUNT_6
0168	W_COUNT_7

Read – write
7 – 0 set the number of times to repeat data marked W.

TX X Registers

000C	X_COUNT_0
002C	X_COUNT_1
004C	X_COUNT_2
006C	X_COUNT_3
010C	X_COUNT_4
012C	X_COUNT_5
014C	X_COUNT_6
016C	X_COUNT_7

Read – write
7 – 0 set the number of times to repeat data marked X.

TX Y Registers

0010	Y_COUNT_0
0030	Y_COUNT_1
0050	Y_COUNT_2
0070	Y_COUNT_3
0110	Y_COUNT_4
0130	Y_COUNT_5
0150	Y_COUNT_6
0170	Y_COUNT_7

Read – write
7 – 0 set the number of times to repeat data marked Y.

TX Z Registers

0014	Z_COUNT_0
0034	Z_COUNT_1
0054	Z_COUNT_2
0074	Z_COUNT_3
0114	Z_COUNT_4
0134	Z_COUNT_5
0154	Z_COUNT_6
0174	Z_COUNT_7

Read – write
7 – 0 set the number of times to repeat data marked Z.

TX Idle 0 Registers

0018	IDLE_0_0
0038	IDLE_0_1
0058	IDLE_0_2
0078	IDLE_0_3
0118	IDLE_0_4
0138	IDLE_0_5
0158	IDLE_0_6
0178	IDLE_0_7

Read – write

13 – 0 Initial pattern to send while waiting for Start.

TX Idle 1 Registers

0018	IDLE_1_0
0038	IDLE_1_1
0058	IDLE_1_2
0078	IDLE_1_3
0118	IDLE_1_4
0138	IDLE_1_5
0158	IDLE_1_6
0178	IDLE_1_7

Read – write

13 – 0 Pattern to send upon completion of data transmission.

TX Data Read-back

0004	DATA_RDBK_0
0024	DATA_RDBK_1
0044	DATA_RDBK_2
0064	DATA_RDBK_3
0104	DATA_RDBK_4
0124	DATA_RDBK_5
0144	DATA_RDBK_6
0164	DATA_RDBK_7

Read only
31 – 0 Data from Output FIFO read for loop-back testing purposes. Please note that once read the data is removed from the FIFO and needs to be reloaded before transmission can occur.

Operational Brief:

Data can be loaded into the local memory [2 x 256 MB SDRAM] via DMA or single write transfers [Store]. Once the data is transferred the hardware is configured for transmission [Transmit]. The appropriate channels are enabled and configured for synchronization options. The designated trigger event occurs and the data is transferred from SDRAM to the output serial channels.

In **Store** mode the data is input from the PCI bus and written to the SDRAM. The data can be input by single word writes from the host or by DMA programmed by the host and executed by the on-board DMA engine. The 9054 provides the PCI interface and DMA engine. The DMA Xilinx supports the DMA transfers by managing the input FIFO. The DMA transfers data from the PCI bus based memory to the Input FIFO. The Address Generator monitors the FIFO flags and when sufficient data is present bursts the data into memory. The data transfer from FIFO to SDRAM operates at 66 MHz and the PCI bus bursts at 33 MHz. The PCI_LVDS_8T can handle continuous input data bursts. The Half full flag is used to manage the bursts to the SDRAM. The Address Generator in conjunction with the DMA Xilinx switches from burst to single word transfers when there is less then 1/2 FIFO to read and the data has been completely transferred from the PCI memory into the Input FIFO. Any LW bounded length can be transferred into the Input FIFO. Any quad-word bounded length can be transferred into the SDRAM. In the case of a mis-match the data is repeated to fill the last location. Channel 0 and channel 4 are used to control the SDRAM data filling.

In **Transmit** mode the Address Generator works with the TX Xilinx to move data from SDRAM to the serializers. The software configures the Address Generator to enable the appropriate channels and initialize the initial pointer(s) and modes. The TX is also initialized for mode. The Address Generator when started will immediately fill the output FIFO for each enabled channel, then go into a round-robin arbitration mode to determine when to move additional data. The TX channels will wait until the programmed trigger event occurs then move data to the serializers. As soon as the TX channels have read enough data to cause the half-full flags to indicate not half-full the Address Generator will move another burst of data to that channel. The Address Generator will continue to operate until all enabled channels have completed. Normally all of the channels are in the same mode, but they do not have to be. All channels that are enabled have to be in Transmit mode. Within Transmit mode the channels

are independent for the looping options.

In **Direct** mode the data is moved from the Input FIFO to the Output FIFO bypassing the SDRAM. The half-full flags are used to determine when there is data in the input side and room in the output side. The output clock rate and expansion factor determine the rate data is read from the Output FIFO. The PCI bus activity will determine if the input data can keep pace with the output. With no expansion the PCI side has a 2X advantage over the output side at max transmission rate.

The Hardware should be initialized Address generator first, TX second, DMA third, PLX last. The PLX chip can be initialized all but the last step of enabling the DMA function before doing the Xilinx definitions. Once the PLX is started the data can begin to flow through the board. In Direct mode The PLX will start to fill the Input FIFO. When the FIFO is half full the Address Generator will start to transfer from the Input FIFO to the Output FIFO for the channel that is enabled. The TX will sense that there is data in the Output FIFO and begin the transmission [should already be enabled in this mode]. The Address Generator will be bursting 512 – 32 bit words => 1024 16 bit data for transmitting. The DMA Xilinx will sense that there is room in the Input FIFO as soon as the Address Generator has read below the half-full point and start another DMA from host memory. The Address Generator will attempt to keep the Output FIFO close to Full by loading 1/2 FIFO each time the FIFO has 1/2 FIFO available. The DMA Xilinx will attempt to keep the Input FIFO full by loading via DMA 1/2 FIFO each time the Input FIFO is less than 1/2 full. The PCI bus has more than 2x the base bandwidth of the output data stream. As long as the PCI bus is available, without delay, the Direct mode can function without gaps in the output data stream. If any of the TX expansion factors are used the leverage of the PCI bus over the transmission rate increases making the "job" easier.

The PLX device should be programmed to start the DMA transfer. The PLX will then start-up when commanded, wait for the DMA Xilinx to respond [FIFO is less than half full on programmed channel and state-machine has started up] then request the PCI bus. The Bus arbitration and start-up take 6 clocks. Assuming that the PCI bus is not busy delaying the start of the burst, the DMA Xilinx and the PCI bus will be synchronized and data will flow from the PCI Bus to the Input FIFO. If the PCI bus has an issue or the max time has expired the PLX device will issue a BLASTn signal to the DMA Xilinx to halt the transfer. When there is room within the PLX FIFO the process will restart. The pipeline will retain the previous state and begin again where it left off. If the PCI bus is

not used by other devices the only down time will be the 6 clocks for arbitration plus an additional 3 within the DMA Xilinx to restart. 250 data samples per 250+9 clocks for an effective bandwidth of 127.4 Mbytes/Sec. The SDRAM can support the data rate in Store mode. The Direct mode will be restricted by the samples gathered and the clock rate of the Output data stream.