PEX 8311/C5409/21 AN

June 16, 2006

Version 1.0

Texas Instruments TMS320VC5409/21 DSP to PEX 8311 Application Note

Features

- 2.5 GHz, x1 PCI-Express Interface
- 66 MHz x 32 bit non-multiplexed Local Bus Interface.
- Two direct-mapped channels with FIFO for PCI express-initiated transfers to/from Local bus (Direct Slave)
- One direct-mapped channel with FIFO for Local Bus-initiated transactions to/from PCI Express space (Direct Master)
- Two independent DMA channels, with 256 byte FIFOs, multiple operating modes, and predictive read-ahead,
- On-chip shared memory for storage of DMA descriptor lists, and application-specific shared data

General Description

This application describes how to interface an array of Digital Signal Processors (DSPs), in this case the Texas Instruments TMS320VC5409/21 family processors, to PCI Express interface using the PEX 8311 PCI Express-to-Local Bus Bridge.

The PEX 8311 is highly optimized for I/O processing. The PEX 8311 in a DSP system is a data-moving engine while the DSPs are co-processors. The PEX 8311 can also allow a PCI-Express master to perform housekeeping duties in DSP-based systems, allowing the DSPs to achieve higher signal processing rates.

The PEX 8311 with the TMS320VC5409/5421 can be used in Voice over IP (VoIP), high-speed image processing, and DSP resource card applications.

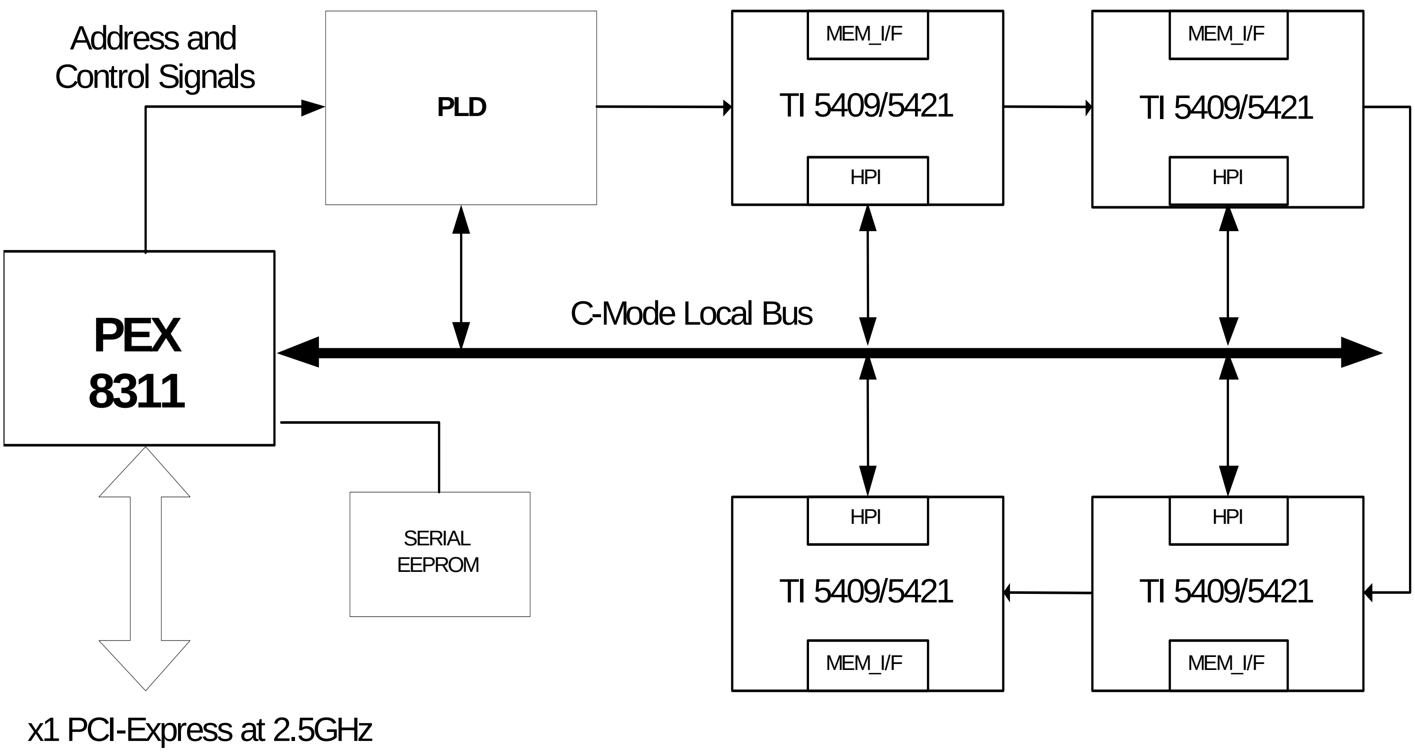

Figure 1. PLX PEX 8311 and TI TMS320VC5409/21 PCI Subsystem

Table of Contents

List of Figures

1. INTRODUCTION

This application note describes the connection between the PLX PEX 8311 bridge chip and the Texas Instruments (TI) TMS320VC5409/5421 DSP. This application note can be used as a basis to build a PCI-Express adapter that plugs into a PCI-Express backplane. Figure 1 shows a system level block diagram, which includes connections between the PEX 8311, TI 5409/5421 DSPs and PLD and serial EEPROM.

The 16-bit HPI Bus provided by the 5409/5421 simplifies the interconnection between the PEX 8311 and the TI 5409/5421. The connection between the PEX 8311 Local bus and the 5409/5421 16-bit HPI Bus optimizes the data rate between the two devices for high-speed burst transfers.

Note: The TI 5409/21 are target only devices and therefore do not have bus mastering capability on their HPI.

1.1 Advantages of PEX 8311

One of the advantages of using the PEX 8311 over other PCI-Express bridge devices are that the PEX 8311 has two DMA channels for better throughput. The PEX 8311 can run up to 66MHz on the Local bus and x1, 2.5GHz on the PCI-Express bus.

Another advantage is that the cost of the PEX 8311 is lower than other solutions. If a RISC CPU is not required in an application, then the PEX 8311 is an excellent choice.

1.2 PEX 8311 PCI-Express Interface

The PEX 8311 PCI-Express interface is an x1 interface. It runs in 2.5 Gbps data rate per direction.

1.3 PEX 8311 DMA Controllers

The PEX 8311 has 2 DMA channels. Both channels have the full features of, Block Transfer Scatter/Gather (descriptor chain), and Demand (DREQDACK) modes. The scatter-gather DMA controllers allow for extremely efficient management of multiple channels of data with minimal software overhead for transfer between PCI-Express and Local buses. For efficient transfers between PCI-Express and local memory, DMA should be used as software overhead is minimal and high burst throughput is easily achieved.

For Block mode or Demand mode DMA, a CPU on the PCI-Express bus can be used to set up the PEX 8311 DMA registers. In this case, external memory and external logic for the memory controller are not used or required as data is transferred directly from/to the DSPs.

When using scatter/gather DMA with the PEX 8311, the 8311’s shared memory can be used to store DMA descriptors, as well as application specific data. This shared memory can be accessed from Local bus or PCI express. DMA performance considerations and programming tips are given in the PEX 8311 design note entitled “Optimizing PEX 8311 PCI Express-to-Local Bus DMA Performance”

2. Initialization

2.1 Serial EEPROM

The initialization of the PEX 8311's internal configuration registers is accomplished by two serial EEPROM. After reset is removed, the PEX 8311 reads data from the two serial EEPROM and uses them to program its internal configuration registers. It is also possible to configure from the PCI-Express host. Refer to the PEX 8311 Data Book, as listed in the Reference section, for more details.

3. ARCHITECTURE

For the connections of the PEX 8311 to the TI 5409/21, we are assuming the EMIF memory bus of the DSP is disabled. The reason is that Texas Instruments has multiplexed the 16-bit HPI bus with the EMIF pins. Since we are using the 16-bit HPI bus, we are disabling the EMIF bus. Code and data for DSPs reside in the internal DSP RAM.

3.1 PEX 8311 Connections to TI DSPs

Some of the functions of the PLDs, in Figures 2 and 3, are to invert the LW/R# signal, to generate the HDS[2:1]#, HCS#, and READY# signals.

To connect to the TI 5409 and TI 5421, the C-Bus mode (non-multiplexed) of the PEX 8311 was chosen. This is the most efficient way to communicate between the PEX 8311 and the TI DSPs.

3.1.1 Pin Connections for PLX PEX 8311 to TI 5409

Below is a list of the PEX 8311 signals, their corresponding TI 5409 DSP signals, and a brief description of what they do.

LA[16:2] - A[15:1]: Local bus address signals.
LBE[1]# - A[0]: Least Significant Bit of Local bus address signal.
LAD[15:0] - D[15:0]: Local bus data signals.
LA[31:30] - (PLD) - HCS# : Generate HCS# chip select signal for TI 5409.
LINTI - TOUT: Local interrupt.
LW/R# - (PLD) - HR/W#: Local Write/ Read signal inverted by external PLD.
RD_STRB# (PLD) - HDS[1]#: Local Read strobe.
WR_STRB# (PLD) - HDS[2]#: Local Write strobe.
HRDY: The PLD polls the HRDY signal to know when the TI DSP is ready for the next access.

Figure 2 shows the signal connections between the PEX 8311 and TI 5409.

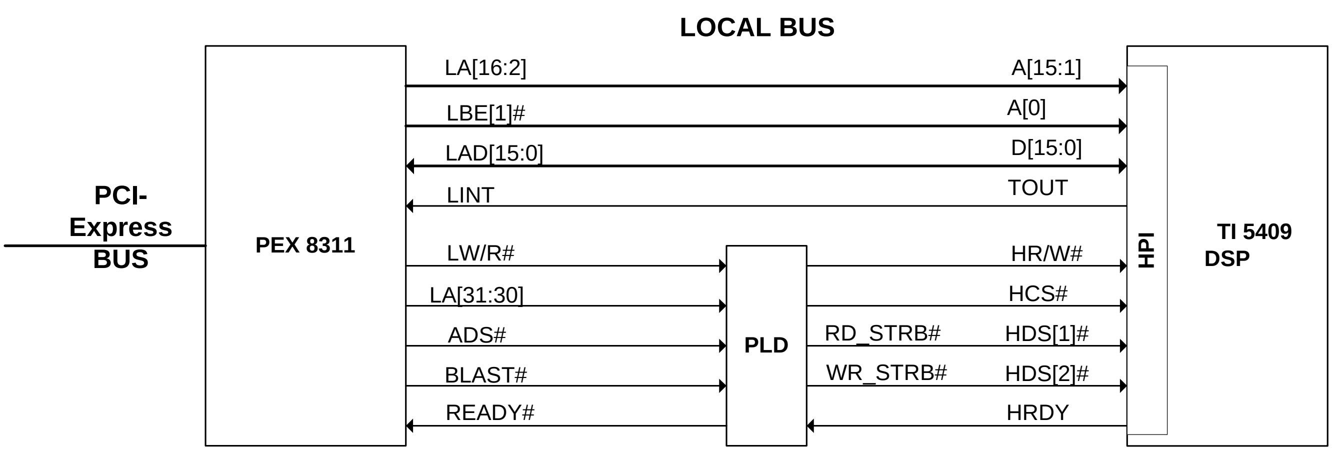

Figure 2. PLX PEX 8311 and TI 5409 Signal Connections

3.1.2 Pin Connections for PLX PEX 8311 to TI 5421

Below is a list of the PEX 8311 signals, their corresponding TI 5421 DSP signals, and a brief description of what they do.

LA[18:2] - HA[17:1]: Local bus address signals.
LBE[1]# - HA[0]: Least Significant Bit of Local bus address signal.
LAD[15:0] - HD[15:0]: Local bus data signals.
LA[31:30] - (PLD) - HCS#: Generate the HCS# chip select signal for TI 5421.
LINTI - A_GPIO3: Local interrupt.
LW/R# - (PLD) - HR/W#: Local Write/ Read signal inverted by external PLD.
RD_STRB# (PLD) - HDS[1]#: Local Read strobe.
WR_STRB# (PLD)- HDS[2]#: Local Write strobe.
HRDY: The PLD polls the HRDY signal to know when the TI DSP is ready for the next access.

Figure 3 shows the signal connections between the PEX 8311 and TI 5421.

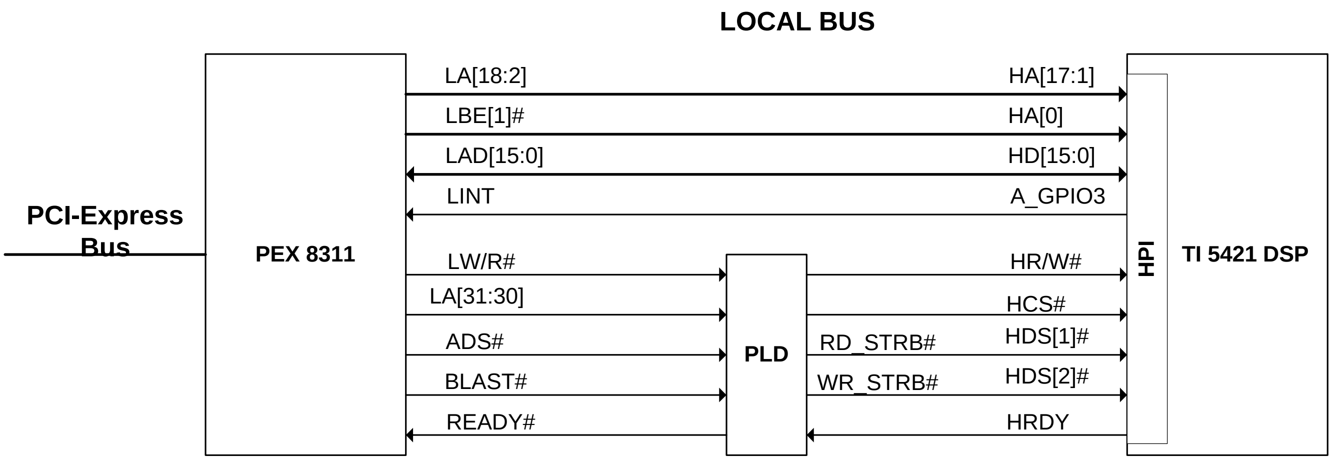

Figure 3. PLX PEX 8311 and TI 5421 Signal Connections

3.2 Local Bus Interface Voltage

The TI DSP 5409/5421 are a 1.8 Volt core, 3.3 Volt I/O device. The PEX 8311 is a 3.3 Volt device with 5 Volt tolerant I/O. Therefore, no level translators are necessary between the two devices.

3.3 Performance

The Local bus clock of the PEX 8311 can run up to 66MHz while the PCI-Express bus clock can run up to 2.5GHz. However, the HPI Bus performance limits the overall transfer rate of the Local bus.

For the PCI-Express bus, the transfer rate can up to 250 MB/sec (8bit/10bit encoder, 2G / 8). For the Local bus, the transfer rate is approximately 8 MB/sec for the ‘5409 and ‘5421 because of the required wait states of the DSP. There are 16 wait states inserted for the read cycle.

3.4 Programmable Logic

Although both TI 5409 and TI 5421 buses are compatible with the C-Bus mode of the PEX 8311, there are some small differences, which require some programmable logic. The PLD accomplishes several tasks:

1. Generate the chip select HCS# signal for the TI 5409/21.
2. Invert the LW/R# signal.
3. Generate the RD_STRB# and WR_STRB# for the TI DSPs
4. Generate the READY# signal to the PEX 8311.

The PLD generates the HCS# chip select from the LA[31:30] signals. The HCS# is asserted when LA[31:30] equals to 2'b01. The PLD then generates the HDS[1]# for a nonmultiplexed read cycle and HDS[2]# for a nonmultiplexed write cycle. It then generates the READY# signal back to the PEX 8311. The PLD checks the status of the BLAST# signal to determine if it is a single cycle or burst cycle access. Finally, the PLD polls the HRDY signal from the DSP to see if the DSP is ready for the next access.

Note: This Verilog code is a simplified version. It has not been tested on a real board. Designers should modify this code as necessary to suit their specific application.

```
// This file contains the Verilog code to generate the Read and Write strobes to the TI 54x DSPs module
//
PEX8311_Decoder3 (lwr_l,clk,reset_l,ads_l,blast_l, hds1_read_l, hds2_write_l, hcs_l,
                          addr,ready_l, hrdy, hrw_l);
input lwr_l;
input clk;
input reset_l;
input ads_l;
input blast_l;
output hds1_read_l;
output hds2_write_l;
output hcs_l;
input addr;
output ready_l;
input hrdy;
output hrw_l;

reg [3:0] strobe_state;
wire hcs_l;
wire [1:0] addr;
reg hds1_read_l;
reg hds2_write_l;
reg ready_l;
wire hrdy;
wire hrw_l;
reg [3:0]counter;
reg counter_ena;
reg counter_clear;
wire counter_ovf;

parameter IDLE          = 4'b0000;
parameter RD_WR         = 4'b0001;
parameter READ1         = 4'b0010;
parameter READ2         = 4'b0011;
parameter READ3         = 4'b0100;
```

```
parameter READ4          = 4'b0101;
parameter WRITE1         = 4'b0110;
parameter WRITE2         = 4'b0111;
parameter WRITE3         = 4'b1000;
parameter WRITE4         = 4'b1001;

assign hcs_l = !(addr[1:0] == 2'b01); // Chip Select for TI DSP 54x
assign hrw_l = !lwr_l;   // Inversion of the LWR_L signal

// State Machine to generate the Read and Write strobes

always @(posedge clk or negedge reset_l)
  begin
        if (reset_l == 0)
        begin
                hds1_read_l = 1'b1;
                hds2_write_l = 1'b1;
                ready_l = 1'b1;
                strobe_state <= IDLE;
        end
        else
                case (strobe_state)
                IDLE : begin
                                if (hcs_l == 1'b0 && ads_l == 1'b0 && hrdy ==1'b1)
                                strobe_state <= RD_WR;
                          else

                                strobe_state <= IDLE;
                          end

                RD_WR : begin
                                if (lwr_l == 0)
                                strobe_state <= READ1;
                                else
                                strobe_state <= WRITE1;
                                end

                READ1 :begin
                                hds1_read_l = 1'b0;
                                counter_ena = 1'b1;
                                if (counter_ovf == 1'b1)
                                strobe_state <= READ2;
                                else
                                strobe_state <= READ1;
                                end


                READ2: begin
                                hds1_read_l = 1'b0;
                                ready_l = 1'b0;
                                counter_clear = 1'b1;
                                counter_ena = 1'b0;
                                strobe_state <= READ3;
                                end

                READ3: begin
                                hds1_read_l = 1'b1;
                                ready_l = 1'b1;
                                if (blast_l ==0 )
```

```
                                        strobe_state <= IDLE;
                                        else
                                        strobe_state <= READ4;
                                        end

                    READ4: begin
                                        if (hrdy ==1'b1)
                                        strobe_state <= READ1;
                                        else
                                        strobe_state <= READ4;
                                        end

                    WRITE1 : begin
                                        hds2_write_l = 1'b0;
                                         strobe_state <= WRITE2;
                                        end

                    WRITE2 : begin
                                        hds2_write_l = 1'b0;
                                        ready_l = 1'b0;
                                        strobe_state <= WRITE3;
                                        end

                    WRITE3: begin
                                        hds2_write_l = 1'b1;
                                        ready_l =1;
                                        if (blast_l ==0)
                                        strobe_state <= IDLE;
                                        else
                                        strobe_state <= WRITE4;
                                        end

                    WRITE4: begin
                                        if (hrdy == 1'b1)
                                        strobe_state <= WRITE1;
                                        else
                                        strobe_state <= WRITE4;
                                        end
                    endcase
  end

// Counter to insert the wait states for the READY_L signal and HDS1_READ_L for read cycle
always @(posedge clk or negedge reset_l)
  begin
          if (reset_l == 0)
          counter[3:0] = 4'b0000;
          else if (counter_ena ==1)
          counter[3:0] = counter +1;
          else if (counter_clear ==1)
          counter[3:0] = 4'b0000;
          end

assign counter_ovf = (counter[3:0] == 4'b1000);

 endmodule
```

4. REFERENCES

The following is a list of additional documentation to provide the reader with further information about the TI 5409/5421 and PEX 8311.

- *PEX 8311 Data Book*, PLX Technology, Inc., www.plxtech.com
- *PEX 8311 Design Note: DMA Performance Considerations*, PLX Technology, Inc., www.plxtech.com
- *PEX 8311 Application Note: Texas Instruments TMS320C6202 DSP to PEX 8311,* PLX Technology, Inc,. www.plxtech.com
- *Texas Instruments TMS320VC5409 Data Sheet,* Texas Instruments, Inc., http://www.ti.com
- *Texas Instruments TMS320VC5421 Data Sheet,* Texas Instruments, Inc.. http://www.ti.com