

Spartan-3 for PCI Express

Starter Kit Board User Guide v1.3

UG256 May 23, 2007





Xilinx is disclosing this Document and Intellectual Property (hereinafter “the Design”) to you for use in the development of designs to operate on, or interface with Xilinx FPGAs. Except as stated herein, none of the Design may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Any unauthorized use of the Design may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

Xilinx does not assume any liability arising out of the application or use of the Design; nor does Xilinx convey any license under its patents, copyrights, or any rights of others. You are responsible for obtaining any rights you may require for your use or implementation of the Design. Xilinx reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of Xilinx. Xilinx assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

THE DESIGN IS PROVIDED “AS IS” WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XILINX, OR ITS AGENTS OR EMPLOYEES. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DESIGN, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE DESIGN, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XILINX IN CONNECTION WITH YOUR USE OF THE DESIGN, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XILINX HEREUNDER FOR USE OF THE DESIGN. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XILINX WOULD NOT MAKE AVAILABLE THE DESIGN TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Design is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems (“High-Risk Applications”). Xilinx specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Design in such High-Risk Applications is fully at your risk.

© 2007 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
5/08/06	1.1	Initial Xilinx release
7/21/06	1.2	Minor Editorial Updates
5/23/07	1.3	Updated file names to conform to PCI-SIG guidelines for trademarks.

Table of Contents

Preface: About This Guide

Acknowledgements	5
Contents	5
Additional Resources	6
Conventions	7
Typographical	7
Online Document	8

Chapter 1: Introduction

Key Components and Features	9
The Evaluation Core for PCI Express	11
Additional Core Resources	11
Technical Support	11
Document	11

Chapter 2: PCI Express Interface

Overview	13
Operation	14
Connections	14
Related Resources	15

Chapter 3: EXP Expansion Connectors

Overview	17
Operation	17
Connections	18
UCF Location Constraints for the EXP Connectors JX1	18

Chapter 4: Clock Sources

Overview	23
Operation	23
Connections	23
UCF Example Constraints for Global Clock Inputs	23

Chapter 5: Switches and Buttons

Overview	25
Operation	25
Connections	25
UCF Example Constraints for Board Switches	26

Chapter 6: LEDs

Overview	27
Operation	27
Connections	27

Chapter 7: VGA Display Port

Overview	29
Operation	29
Connections	30
Related Resources	30

Chapter 8: RS-232 PORT

Overview	31
Operation	31
Connections	31

Chapter 9: DDR SDRAM

Overview	33
Operation	33
UCF	37
Address	37
Control	37
Data	37
Related Resources	38

Chapter 10: SPI Serial Flash

Overview	39
Operation	39
Connections	40
Related Resources	40

Chapter 11: FPGA Configuration

Overview	41
Operation	42
Configuration Mode jumpers	42
PROG Switch	42
DONE LED	42
Platform Flash Design Revisioning	42
Using the Platform Flash for User Storage	42
Reprogramming Platform Flash from the FPGA	43
Connections	43
Related Resources	43

Chapter 12: Power Supplies

Overview	45
Input Power Regulations	45
Voltage Supervision	46
Operation	47
PCI Express Edge Connector as Source	47
Disk Drive Connector as Source	47
Barrel Jack as Source	47
Related Resources	47

Appendix A: Example User Constraints File (UCF)

Appendix B: Schematics

About This Guide

This Spartan™-3 Starter Board Kit for PCI Express User Guide provides basic information about the capabilities, functions, and design of the Xilinx Spartan-3 PCI Express Starter Kit Board. It includes general information for using the various peripheral functions included on the board. For detailed reference designs, including VHDL or Verilog source code, please visit the [Spartan-3 Starter Board for PCI Express product page](#).

Acknowledgements

Xilinx wishes to thank the following companies for their support of the Spartan-3 Starter Kit board for PCI Express:

- Avnet Electronics
- Philips Semiconductors for the PCI Express x1 lane PHY
- Micron Technology, Inc. for the 32M x 16 DDR SDRAM
- STMicroelectronics for the 4M x 1 SPI serial EEPROM
- Texas Instruments Incorporated for the various power supply solutions

Contents

This guide contains the following chapters:

- [Preface, “About this Guide”](#) introduces the organization and purpose of this guide, a list of additional resources, and the conventions used in this document.
- [Chapter 1, “Introduction”](#) describes the Starter Kit Board, and provides information about its key components and features. This chapter also provides information about additional resources, recommended design experience, providing feedback, and securing technical support.
- [Chapter 2, “PCI Express Interface”](#) provides detailed information about the interface for PCI-Express (PCIe) two-chip solution consisting of the Xilinx Spartan-3 FPGA and the Philips PX1011A-EL1 PCI Express PHY (PX1011A).
- [Chapter 3, “EXP Expansion Connectors”](#) describes expansion capabilities for customized user application daughter cards and interfaces over two EXP expansion connectors.
- [Chapter 4, “Clock Sources”](#) details the Spartan-3 FPGA global clock input pins that can be used for high-performance, low-skew clocking.
- [Chapter 5, “Switches and Buttons”](#) describes the operation, connection, and example constraints for use-defined switches and buttons.

- [Chapter 6, “LEDs”](#) details the overview, operation, and connections of the eight user LEDs.
- [Chapter 7, “VGA Display Port”](#) describes the video display port that is available through an on-board Philips TDA8777 Triple DAC.
- [Chapter 8, “RS-232 PORT”](#) describes the DCE compatible RS-232 serial port for connection to most computer serial ports.
- [Chapter 9, “DDR SDRAM”](#) provides information about the two 512 MB (32M x 16) Micron Technology DDR SDRAM (MT46V32M16) that provide a 32-bit data interface to the Spartan-3 FPGA.
- [Chapter 10, “SPI Serial Flash”](#) details the STMicroelectronics M25P40 4Mb SPI serial flash and its possible uses.
- [Chapter 11, “FPGA Configuration”](#) describes the two supported FPGA configuration options.
- [Chapter 12, “Power Supplies”](#) describes the power supplies for all components on the board, as well as the EXP expansion connectors.
- [Appendix A, “Example User Constraints File \(UCF\)”](#) contains the contents of the MASTER Constraints File for the starter board.
- [Appendix B, “Schematics”](#) contains comprehensive and detailed schematic maps.

Additional Resources

For additional information and resources, see www.xilinx.com/support. To go directly to a specific area of the support site, click a link in the table below.

Resource	Description/URL
Tutorials	Tutorials covering Xilinx design flows, from design entry to verification and debugging www.xilinx.com/support/techsup/tutorials/index.htm
Answer Browser	Database of Xilinx solution records www.xilinx.com/xlnx/xil_ans_browser.jsp
Application Notes	Descriptions of device-specific design techniques and approaches www.xilinx.com/xlnx/xweb/xil_publications_index.jsp?category=Application+Notes
Data Sheets	Device-specific information on Xilinx device characteristics, including readback, boundary scan, configuration, length count, and debugging www.xilinx.com/xlnx/xweb/xil_publications_index.jsp
Problem Solvers	Interactive tools that allow you to troubleshoot your design issues www.xilinx.com/support/troubleshoot/psolvers.htm
Tech Tips	Latest news, design tips, and patch information for the Xilinx design environment www.xilinx.com/xlnx/xil_tt_home.jsp

Conventions

This document uses the following conventions.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
Courier font	Messages, prompts, signal names, and program files	speed grade: - 100
Courier bold	Literal commands you enter in a syntactical statement	ngdbuild design_name
<i>Italics</i>	Variables in a syntax statement for which you must supply values	See the <i>Development System Reference Guide</i> for more information.
	References to other manuals	See the <i>User Guide</i> for details.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are <i>not</i> connected.
Shading	Unsupported or reserved items	This feature is not supported
Square brackets []	Optional entry or parameter, with the exception of bus specifications. For bus specifications, brackets are required, for example bus [7:0] .	ngdbuild [option_name] design_name
Braces { }	A list of items from which you must choose one or more	lowpwr = {on off}
Vertical bar	Separates items in a list of choices	lowpwr = {on off}
Vertical ellipsis .	Omitted repetitive material	IOB #1: Name = QOUT' IOB #2: Name = CLKIN' . . .
Horizontal ellipsis ...	Omitted repetitive material	allow block block_name loc1 loc2 ... locn;
Notations	The prefix '0x' or the suffix 'h' indicate hexadecimal notation	A read of address 0x00112975 returns 45524943h
	An '_n' means the signal is active low	usr_teof_n is active low

Online Document

The following conventions are used in this document for cross-references and links to URLs.

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See “ Additional Resources ” for more information. See “ Title Formats ” in Chapter 1 for detailed information.
Blue, underlined text	Hyperlink to a website (URL)	Go to www.xilinx.com for the latest speed files.

Introduction

The Xilinx Spartan-3 Starter Kit Board for PCI-Express provides everything you need to build a low-cost Spartan-3-based PCI Express application. To help you understand the full functionality and features of this board kit, the LogiCORE™ evaluation core for PCI-Express can be downloaded and installed at no extra cost. This IP core has been tested and defined to work with the board kit, and is an integral part of the comprehensive solution.

Key Components and Features

The key features of the Spartan-3 PCI Express Kit board are listed below:

- Xilinx XC3S1000 Spartan-3 FPGA
 - ♦ Up to 391 user I/O pins
 - ♦ 676-pin FBGA package
 - ♦ Over 17,000 logic cells
- Xilinx 8 Mbit Platform Flash configuration PROM
- Philips 2.5 Gbps, PCI Express single lane PHY
- EXP compatible expansion connectors
 - ♦ 168 User I/O
 - ♦ Full and half card support
 - ♦ 2.5V and 3.3V
 - ♦ 2 global clock inputs
- 128 Mbyte (512 Mbit x 2) of DDR SDRAM, x32 data interface, 100+ MHz
- VGA display port
- 9-pin RS-232 serial port
- 4 Mbits of SPI serial flash memory with data and code storage
- 25.175 MHz clock oscillator
- 50 MHz clock oscillator
- 8-pin DIP socket for auxiliary clock input
- Eight discrete LEDs
- Three push button switches
- Four position DIP switch

Figure 1-1 illustrates the block diagram of the board functions.

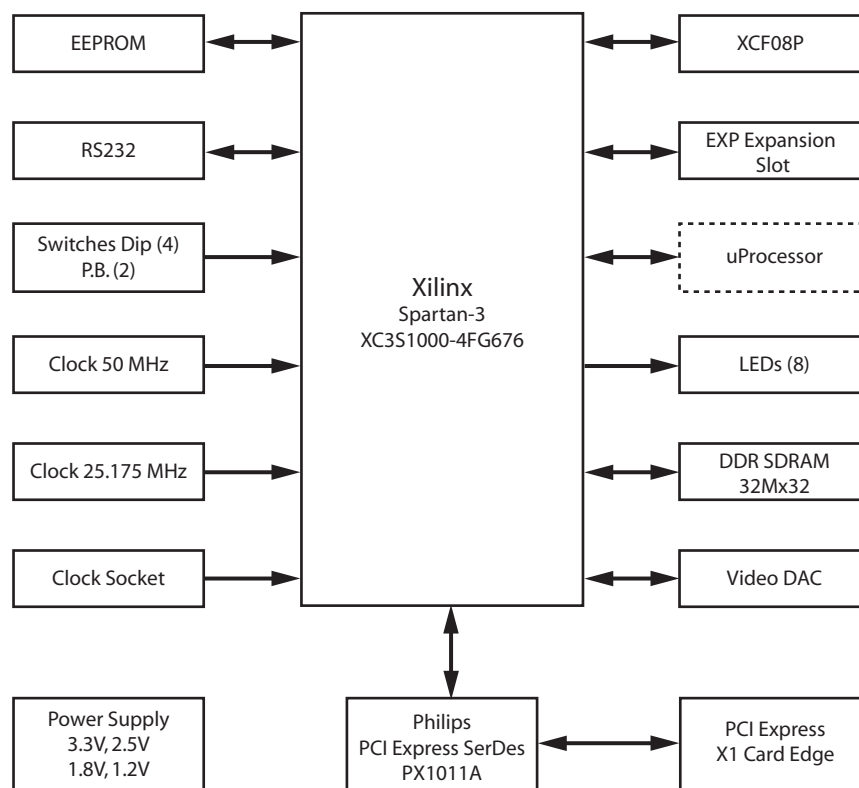


Figure 1-1: Spartan-3 Starter Kit Board for PCI Express Block Diagram

Figure 1-2 shows a detailed diagram of the board connectors, jumpers, and main functional blocks.

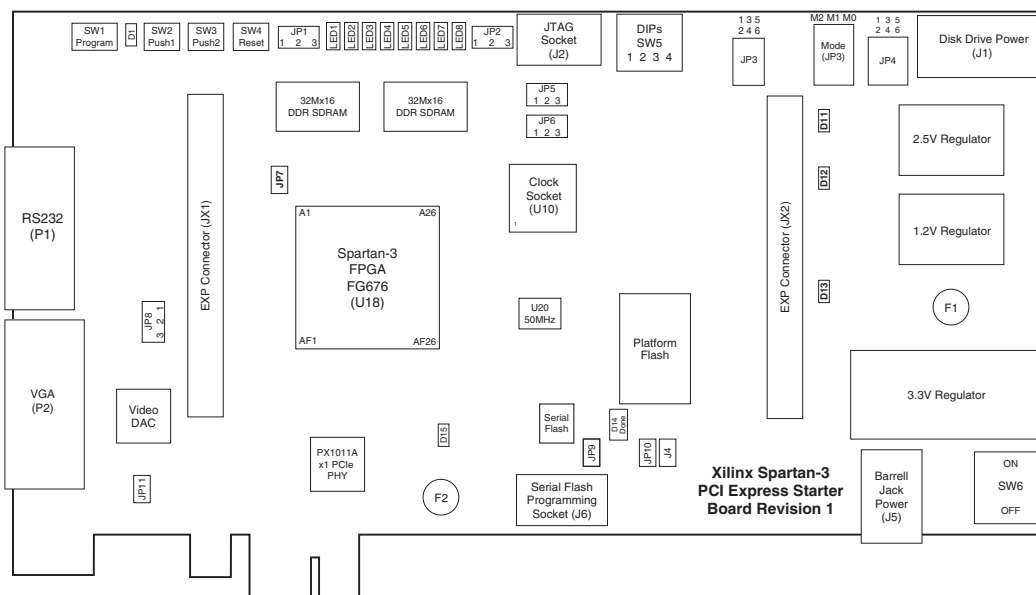


Figure 1-2: Spartan-3 Starter Kit Board for PCI Express Functional Diagram

The Evaluation Core for PCI Express

The instructions for downloading the evaluation core are included in the Spartan-3 Starter Kit Board for PCI-Express hardware packaging.

Additional Core Resources

For additional information and documentation for the LogiCORE Endpoint for PCI Express core, go to the product page at www.xilinx.com/pciexpress.

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team with expertise using the LogiCORE Endpoint for PCI Express core.

Xilinx will provide technical support for use of this product as described in the LogiCORE Endpoint for PCI Express User Guide and the LogiCORE Endpoint for PCI Express Getting Started Guide. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.

Document

For comments or suggestions about the documentation, please submit a WebCase from www.xilinx.com/support/clearexpress/websupport.htm. Be sure to include the following information:

- Document title
- Document number
- Page number(s) to which your comments refer
- Explanation of your comments

PCI Express Interface

Overview

The Spartan-3 Starter Kit Board for PCI-Express provides an PCI-Express (PCIe) interface using a two-chip solution consisting of the Xilinx Spartan-3 FPGA and the Philips PX1011A-EL1 PCI Express PHY (PX1011A). The PX1011A is a high-performance, low-power PCI Express electrical Physical layer (PHY) that handles the low level PCI Express protocol and signaling. The Spartan-3 FPGA is configured to include a PCIe PIPE (PHY Interface for PCI Express) endpoint core that interfaces to the external PX1011A, and provides the MAC layer function for the PCIe interface.

The PX1011A PHY includes features such as data serialization and de-serialization, 8b/10b encoding, analog buffers, elastic buffer and receiver detection, which provides superior performance to the media access control (MAC) layer in the FPGA. The PX1011A is a 2.5 Gb/sec. PCI Express PHY with 8-bit data PXPIPE interface. Its PXPIPE interface is a superset of the PHY Interface for the PCI Express (PIPE) specification, enhanced and adapted for off-chip applications. It contains a source synchronous clock for transmit and receive data. The 8-bit data interface operates at 250 MHz with SSTL_2 signaling. The SSTL_2 signaling is compatible with the I/O interfaces available in the Xilinx Spartan-3 FPGA.

Figure 2-1 shows a simple block diagram of the FPGA and PHY implementation.

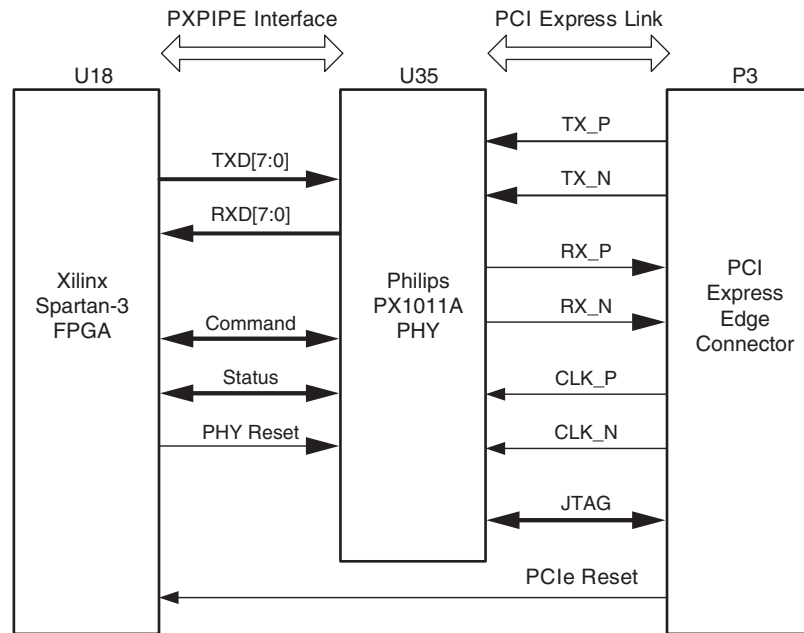


Figure 2-1: Spartan-3 FPGA and PX1011A PHY Connections

Operation

The PX1011A-EL1 PCI Express PHY consists of the physical coding sub-layer (PCS), a Serializer and De-serializer (SerDes) and a set of I/Os (pads). The PCI Express PHY handles the low level PCI Express protocol and signaling. The PX1011A interface between the MAC and PHY is a superset of the PIPE specification, with the addition of source synchronous clocks for RX and TX data to simplify timing closure. The digital interface between the FPGA-based MAC and the PHY is also referred to as PXPIPE interface. It consists of 8-bit input and output words, each with control signals and source synchronous clocks. The data rate across the PXPIPE is 250 MB per second in both directions.

The PCI Express link consists of a differential input and differential output pair. The data rate of these signals is 2.5 Gb per second.

The PIPE receive (RXD) and transmit (TXD) signals must be properly terminated SSTL2-I signals at both the driving device and the receiving device. PCB signal routing for these signals are length matched to minimize skew.

Connections

The following shows the user constraints file (UCF) location constraints for the PX1011A PCI Express PHY interface.

```

# System reset signal from the PCIe interface to the FPGA
NET "sys_reset_n"          LOC = "AE4" | IOSTANDARD = LVCMOS25 | IOBDELAY = NONE ;

# Receive Input Signals
NET "rxclk"                LOC = "AE13" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "phystatus"            LOC = "AF12" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxvalid"              LOC = "AD12" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxstatus<2>"          LOC = "AC11" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxstatus<1>"          LOC = "AD10" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxstatus<0>"          LOC = "AC10" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<0>"            LOC = "AF8" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<1>"            LOC = "AE8" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<2>"            LOC = "AC7" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<3>"            LOC = "AF6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<4>"            LOC = "AE6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<5>"            LOC = "AD6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<6>"            LOC = "AC6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<7>"            LOC = "AE5" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<7>"            LOC = "AD5" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxlecidle"            LOC = "AF4" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;

# Transmit Output Signals
NET "resetn"               LOC = "AF24" | IOSTANDARD = SSTL2_I ;
NET "rxpolarity"           LOC = "AE24" | IOSTANDARD = SSTL2_I ;
NET "txlecidle"            LOC = "AF23" | IOSTANDARD = SSTL2_I ;
NET "txcompliance"         LOC = "AE23" | IOSTANDARD = SSTL2_I ;
NET "powerdown<1>"         LOC = "AD23" | IOSTANDARD = SSTL2_I ;
NET "powerdown<0>"         LOC = "AF22" | IOSTANDARD = SSTL2_I ;
NET "txdata<0>"            LOC = "AE22" | IOSTANDARD = SSTL2_I ;
NET "txdetectrx_loopback" LOC = "AF21" | IOSTANDARD = SSTL2_I ;
NET "txclk"                LOC = "AE21" | IOSTANDARD = SSTL2_I ;
NET "txdata<7>"            LOC = "AD21" | IOSTANDARD = SSTL2_I ;
NET "txdata<6>"            LOC = "AF20" | IOSTANDARD = SSTL2_I ;
NET "txdata<5>"            LOC = "AE20" | IOSTANDARD = SSTL2_I ;
NET "txdata<4>"            LOC = "AF19" | IOSTANDARD = SSTL2_I ;
NET "txdata<3>"            LOC = "AE19" | IOSTANDARD = SSTL2_I ;
NET "txdata<2>"            LOC = "AF15" | IOSTANDARD = SSTL2_I ;
NET "txdata<1>"            LOC = "AE15" | IOSTANDARD = SSTL2_I ;
NET "txdata<0>"            LOC = "AD15" | IOSTANDARD = SSTL2_I ;

```

Related Resources

For more information about the Xilinx LogiCORE Endpoint for PCI Express PIPE, see the *Xilinx LogiCORE Endpoint 1-Lane for PCI Express PIPE User Guide*.

For information about the Philips PCI Express PHY see the [Philips PX1011A-EL1 PCI Express PHY data sheet](#).

EXP Expansion Connectors

Overview

The Spartan-3 Starter Kit Board for PCI-Express provides expansion capabilities for customized user application daughter cards and interfaces over two EXP expansion connectors. The EXP expansion connectors on the PCI Express board can support two half-card EXP modules, or a single dual slot EXP module. Both off-the-shelf EXP modules and user-developed modules can easily be plugged onto the Spartan-3 Starter Kit Board for PCI-Express to add features and functions to the backend application of the main board.

Operation

The EXP specification defines a 132-pin connector, with 24 power, 24 grounds, and 84 user I/Os. The standard EXP configuration implemented on the Spartan-3 Board for PCI Express uses two connectors (Samtec part number QTE-060-09-F-D-A) in a dual slot EXP configuration, for a total of 168 user I/Os. Using a jumper, you can set the voltage levels for the EXP user I/O to either 2.5V or 3.3V. JP5 sets the I/O voltage for both the JX1 and JX2 EXP connectors by setting the VCCO voltage for the 4 banks of the FPGA that connect to the EXP I/O. [Figure 3-1](#) shows the JP5 settings.

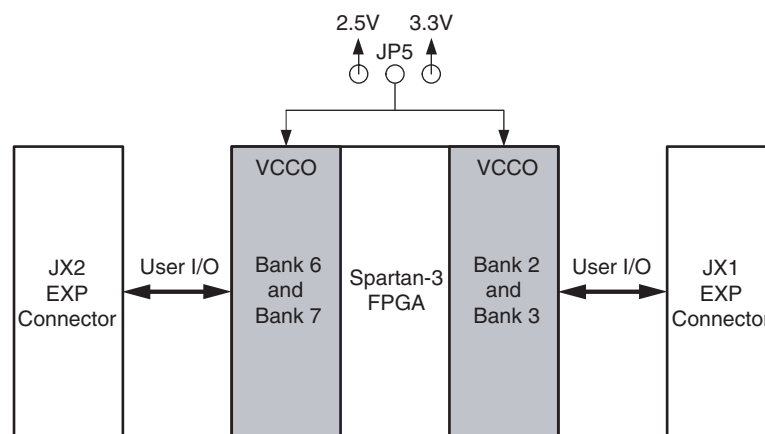


Figure 3-1: EXP User I/O Voltage Settings

The EXP specification defines four user signal types: Single Ended I/O, Differential I/O, Differential and Single Ended Clock Inputs, and Differential and Single Ended Clock

Outputs. Table 3-1 shows a summary of the signal categories for the standard, dual format EXP slot.

Table 3-1: EXP User I/O Types

Signal Category	Pins per Connector	Pins per Dual EXP Slot
Single-ended I/O	34	68
Single-ended clocks	2	4
Differential I/O pairs	22	44
Differential clock input pair	1	2
Differential clock output pair	1	2
2.5V pins (333 mA per pin)	12	24
3.3V pins (333 mA per pin)	12	24
Grounds	24	48
Total	108	216

Because the FPGA I/Os can be configured for either single-ended or differential use, the differential I/Os defined in the EXP specification can serve a dual role. All the differential I/O signals can be configured as either differential pairs or single-ended signals, as required by the end application. In providing differential signaling, higher performance LVDS interfaces can be implemented between the baseboard and EXP module. Connection to high speed A/Ds, D/As, and flat panel displays are possible with this signaling configuration. Applications that require single-ended signals only can use each differential pair as two single-ended signals, for a total of 78 single-ended I/O per connector (156 total in the dual slot configuration).

Connections

The Spartan-3 FPGA user I/O pins connect to the two EXP connectors, JX1 and JX2, as shown in the example UCF figures below. The JX1 and JX2 connectors are Samtec QTE-060-09-F-D-A high performance plugs that mate to Samtec QSE-060-01-F-D-A high-performance receptacles, located on the daughter card. Samtec also provides several high-performance ribbon cables that will mate to the JX1 and JX2 connectors.

UCF Location Constraints for the EXP Connectors JX1

```

Net JX1_SE_IO_0      LOC = M3      ; # B6GIO_0
Net JX1_SE_IO_1      LOC = J7      ; # B7GIO_0
Net JX1_SE_IO_2      LOC = M7      ; # B6GIO_1
Net JX1_SE_IO_3      LOC = J6      ; # B7GIO_1
Net JX1_SE_IO_4      LOC = N7      ; # B6GIO_2
Net JX1_SE_IO_5      LOC = H5      ; # B7GIO_2
Net JX1_SE_IO_6      LOC = M8      ; # B6GIO_3
Net JX1_SE_IO_7      LOC = H2      ; # B7GIO_3
Net JX1_SE_IO_8      LOC = N8      ; # B6GIO_4
Net JX1_SE_IO_9      LOC = J5      ; # B7GIO_4
Net JX1_SE_IO_10     LOC = P8      ; # B6GIO_5
Net JX1_SE_IO_11     LOC = J4      ; # B7GIO_5
Net JX1_SE_IO_12     LOC = P2      ; # B6GIO_6

```

```

Net JX1_SE_IO_13      LOC = K7      ; # B7GIO_6
Net JX1_SE_IO_14      LOC = P7      ; # B6GIO_7
Net JX1_SE_IO_15      LOC = K5      ; # B7GIO_7
Net JX1_SE_IO_16      LOC = R1      ; # B6GIO_8
Net JX1_SE_IO_17      LOC = L8      ; # B7GIO_8
Net JX1_SE_IO_18      LOC = P1      ; # B6GIO_9
Net JX1_SE_IO_19      LOC = L7      ; # B7GIO_9
Net JX1_SE_IO_20      LOC = R2      ; # B6GIO_10
Net JX1_SE_IO_21      LOC = H1      ; # B7GIO_10
Net JX1_SE_IO_22      LOC = R3      ; # B6GIO_11
Net JX1_SE_IO_23      LOC = L1      ; # B7GIO_11
Net JX1_SE_IO_24      LOC = T1      ; # B6GIO_12
Net JX1_SE_IO_25      LOC = L2      ; # B7GIO_12
Net JX1_SE_IO_26      LOC = T2      ; # B6GIO_13
Net JX1_SE_IO_27      LOC = L4      ; # B7GIO_13
Net JX1_SE_IO_28      LOC = V6      ; # B7GIO_14
Net JX1_SE_IO_29      LOC = U7      ; # B7GIO_15
Net JX1_SE_IO_30      LOC = W5      ; # B6GIO_14
Net JX1_SE_IO_31      LOC = V7      ; # B6GIO_15
Net JX1_SE_IO_32      LOC = R8      ; # B7GIO_17
Net JX1_SE_IO_33      LOC = R7      ; # B7GIO_18

Net JX1_SE_CLK_IN     LOC = B13     ; # B0_GCLK7
Net JX1_SE_CLK_OUT    LOC = T4      ; # B7GIO_16

Net JX1_DIFF_N_0      LOC = AB4     ; # B6GPIOn_0
Net JX1_DIFF_N_1      LOC = W7      ; # B7GPIOn_0
Net JX1_DIFF_N_2      LOC = AC2     ; # B6GPIOn_1
Net JX1_DIFF_N_3      LOC = V5      ; # B7GPIOn_1
Net JX1_DIFF_N_4      LOC = W4      ; # B6GPIOn_2
Net JX1_DIFF_N_5      LOC = T8      ; # B7GPIOn_2
Net JX1_DIFF_N_6      LOC = W2      ; # B6GPIOn_3
Net JX1_DIFF_N_7      LOC = R6      ; # B7GPIOn_3
Net JX1_DIFF_N_8      LOC = U4      ; # B6GPIOn_4
Net JX1_DIFF_N_9      LOC = P6      ; # B7GPIOn_4
Net JX1_DIFF_N_10     LOC = U2      ; # B6GPIOn_5
Net JX1_DIFF_N_11     LOC = M6      ; # B7GPIOn_5
Net JX1_DIFF_N_12     LOC = P4      ; # B6GPIOn_6
Net JX1_DIFF_N_13     LOC = N3      ; # B7GPIOn_6
Net JX1_DIFF_N_14     LOC = N1      ; # B6GPIOn_7
Net JX1_DIFF_N_15     LOC = L5      ; # B7GPIOn_7
Net JX1_DIFF_N_16     LOC = K1      ; # B6GPIOn_9
Net JX1_DIFF_N_17     LOC = J2      ; # B7GPIOn_8
Net JX1_DIFF_N_18     LOC = K3      ; # B6GPIOn_10
Net JX1_DIFF_N_19     LOC = H3      ; # B7GPIOn_9
Net JX1_DIFF_N_20     LOC = G1      ; # B6GPIOn_11
Net JX1_DIFF_N_21     LOC = E3      ; # B7GPIOn_10

Net JX1_DIFF_P_0      LOC = AB3     ; # B6GPIOp_0
Net JX1_DIFF_P_1      LOC = W6      ; # B7GPIOp_0
Net JX1_DIFF_P_2      LOC = AC1     ; # B6GPIOp_1
Net JX1_DIFF_P_3      LOC = V4      ; # B7GPIOp_1
Net JX1_DIFF_P_4      LOC = W3      ; # B6GPIOp_2
Net JX1_DIFF_P_5      LOC = T7      ; # B7GPIOp_2
Net JX1_DIFF_P_6      LOC = W1      ; # B6GPIOp_3
Net JX1_DIFF_P_7      LOC = R5      ; # B7GPIOp_3
Net JX1_DIFF_P_8      LOC = U3      ; # B6GPIOp_4
Net JX1_DIFF_P_9      LOC = P5      ; # B7GPIOp_4
Net JX1_DIFF_P_10     LOC = U1      ; # B6GPIOp_5

```

```

Net JX1_DIFF_P_11      LOC = M5 ; # B7GPIOp_5
Net JX1_DIFF_P_12      LOC = P3 ; # B6GPIOp_6
Net JX1_DIFF_P_13      LOC = N4 ; # B7GPIOp_6
Net JX1_DIFF_P_14      LOC = N2 ; # B6GPIOp_7
Net JX1_DIFF_P_15      LOC = L6 ; # B7GPIOp_7
Net JX1_DIFF_P_16      LOC = K2 ; # B6GPIOp_9
Net JX1_DIFF_P_17      LOC = J3 ; # B7GPIOp_8
Net JX1_DIFF_P_18      LOC = K4 ; # B6GPIOp_10
Net JX1_DIFF_P_19      LOC = H4 ; # B7GPIOp_9
Net JX1_DIFF_P_20      LOC = G2 ; # B6GPIOp_11
Net JX1_DIFF_P_21      LOC = E4 ; # B7GPIOp_10

Net JX1_DIFF_CLK_P_IN  LOC = D2 ; # B6GPIOp_12
Net JX1_DIFF_CLK_N_IN  LOC = D1 ; # B6GPIOn_12

Net JX1_DIFF_CLK_P_OUT LOC = M2 ; # B6GPIOp_8
Net JX1_DIFF_CLK_N_OUT LOC = M1 ; # B6GPIOn_8

```

UCF Location Constraints for the EXP Connectors JX2

```

Net JX2_SE_IO_0      LOC = M19 ; # B2GIO_0
Net JX2_SE_IO_1      LOC = P20 ; # B3GIO_0
Net JX2_SE_IO_2      LOC = M20 ; # B2GIO_1
Net JX2_SE_IO_3      LOC = T20 ; # B3GIO_1
Net JX2_SE_IO_4      LOC = K20 ; # B2GIO_2
Net JX2_SE_IO_5      LOC = P21 ; # B3GIO_2
Net JX2_SE_IO_6      LOC = J20 ; # B2GIO_3
Net JX2_SE_IO_7      LOC = R21 ; # B3GIO_3
Net JX2_SE_IO_8      LOC = H20 ; # B2GIO_4
Net JX2_SE_IO_9      LOC = P24 ; # B3GIO_4
Net JX2_SE_IO_10     LOC = J21 ; # B2GIO_5
Net JX2_SE_IO_11     LOC = P22 ; # B3GIO_5
Net JX2_SE_IO_12     LOC = H21 ; # B2GIO_6
Net JX2_SE_IO_13     LOC = R24 ; # B3GIO_6
Net JX2_SE_IO_14     LOC = H22 ; # B2GIO_7
Net JX2_SE_IO_15     LOC = R22 ; # B3GIO_7
Net JX2_SE_IO_16     LOC = J22 ; # B2GIO_8
Net JX2_SE_IO_17     LOC = T23 ; # B3GIO_8
Net JX2_SE_IO_18     LOC = J23 ; # B2GIO_9
Net JX2_SE_IO_19     LOC = T22 ; # B3GIO_9
Net JX2_SE_IO_20     LOC = L23 ; # B2GIO_10
Net JX2_SE_IO_21     LOC = U22 ; # B3GIO_10
Net JX2_SE_IO_22     LOC = M24 ; # B2GIO_11
Net JX2_SE_IO_23     LOC = T21 ; # B3GIO_11
Net JX2_SE_IO_24     LOC = T19 ; # B2GIO_12
Net JX2_SE_IO_25     LOC = V23 ; # B3GIO_12
Net JX2_SE_IO_26     LOC = N20 ; # B2GIO_13
Net JX2_SE_IO_27     LOC = V22 ; # B3GIO_13
Net JX2_SE_IO_28     LOC = W22 ; # B3GIO_14
Net JX2_SE_IO_29     LOC = U20 ; # B3GIO_16
Net JX2_SE_IO_30     LOC = AC26 ; # B2GIO_14
Net JX2_SE_IO_31     LOC = K21 ; # B2GIO_15
Net JX2_SE_IO_32     LOC = U21 ; # B3GIO_17
Net JX2_SE_IO_33     LOC = V20 ; # B3GIO_18

Net JX2_SE_CLK_IN     LOC = AE14 ; # B4_GCLK1
Net JX2_SE_CLK_OUT    LOC = V21 ; # B3GIO_15

Net JX2_DIFF_N_0      LOC = AB24 ; # B2GPIOn_0

```

```

Net JX2_DIFF_N_1      LOC = W26 ; # B3GPIOn_0
Net JX2_DIFF_N_2      LOC = Y26 ; # B2GPIOn_1
Net JX2_DIFF_N_3      LOC = U24 ; # B3GPIOn_1
Net JX2_DIFF_N_4      LOC = W24 ; # B2GPIOn_2
Net JX2_DIFF_N_5      LOC = N23 ; # B3GPIOn_2
Net JX2_DIFF_N_6      LOC = V25 ; # B2GPIOn_3
Net JX2_DIFF_N_7      LOC = N25 ; # B3GPIOn_3
Net JX2_DIFF_N_8      LOC = U26 ; # B2GPIOn_4
Net JX2_DIFF_N_9      LOC = M25 ; # B3GPIOn_4
Net JX2_DIFF_N_10     LOC = T26 ; # B2GPIOn_5
Net JX2_DIFF_N_11     LOC = L25 ; # B3GPIOn_5
Net JX2_DIFF_N_12     LOC = R26 ; # B2GPIOn_6
Net JX2_DIFF_N_13     LOC = K25 ; # B3GPIOn_6
Net JX2_DIFF_N_14     LOC = P26 ; # B2GPIOn_7
Net JX2_DIFF_N_15     LOC = J24 ; # B3GPIOn_7
Net JX2_DIFF_N_16     LOC = L19 ; # B2GPIOn_9
Net JX2_DIFF_N_17     LOC = H25 ; # B3GPIOn_8
Net JX2_DIFF_N_18     LOC = L21 ; # B2GPIOn_10
Net JX2_DIFF_N_19     LOC = E23 ; # B3GPIOn_9
Net JX2_DIFF_N_20     LOC = K23 ; # B2GPIOn_11
Net JX2_DIFF_N_21     LOC = D25 ; # B3GPIOn_10

Net JX2_DIFF_P_0      LOC = AB23 ; # B2GPIOp_0
Net JX2_DIFF_P_1      LOC = W25 ; # B3GPIOp_0
Net JX2_DIFF_P_2      LOC = Y25 ; # B2GPIOp_1
Net JX2_DIFF_P_3      LOC = U23 ; # B3GPIOp_1
Net JX2_DIFF_P_4      LOC = W23 ; # B2GPIOp_2
Net JX2_DIFF_P_5      LOC = N24 ; # B3GPIOp_2
Net JX2_DIFF_P_6      LOC = V24 ; # B2GPIOp_3
Net JX2_DIFF_P_7      LOC = N26 ; # B3GPIOp_3
Net JX2_DIFF_P_8      LOC = U25 ; # B2GPIOp_4
Net JX2_DIFF_P_9      LOC = M26 ; # B3GPIOp_4
Net JX2_DIFF_P_10     LOC = T25 ; # B2GPIOp_5
Net JX2_DIFF_P_11     LOC = L26 ; # B3GPIOp_5
Net JX2_DIFF_P_12     LOC = R25 ; # B2GPIOp_6
Net JX2_DIFF_P_13     LOC = K26 ; # B3GPIOp_6
Net JX2_DIFF_P_14     LOC = P25 ; # B2GPIOp_7
Net JX2_DIFF_P_15     LOC = J25 ; # B3GPIOp_7
Net JX2_DIFF_P_16     LOC = L20 ; # B2GPIOp_9
Net JX2_DIFF_P_17     LOC = H26 ; # B3GPIOp_8
Net JX2_DIFF_P_18     LOC = L22 ; # B2GPIOp_10
Net JX2_DIFF_P_19     LOC = E24 ; # B3GPIOp_9
Net JX2_DIFF_P_20     LOC = K24 ; # B2GPIOp_11
Net JX2_DIFF_P_21     LOC = D26 ; # B3GPIOp_10

Net JX2_DIFF_CLK_P_IN LOC = H24 ; # B2GPIOp_12
Net JX2_DIFF_CLK_N_IN LOC = H23 ; # B2GPIOn_12

Net JX2_DIFF_CLK_P_OUT LOC = N22 ; # B2GPIOp_8
Net JX2_DIFF_CLK_N_OUT LOC = N21 ; # B2GPIOn_8

```


Clock Sources

Overview

The Spartan-3 FPGA has eight global clock input pins that can be used for high-performance, low-skew clocking. All of these clock pins are used on the Spartan-3 Board for PCI Express and are defined in this chapter.

Operation

Table 4-1 shows the eight clock inputs to the Spartan-3 FPGA. Six of the eight clocks are buffered to avoid voltage compatibility issues that could result from the adjustable voltage bank settings.

Table 4-1: Global Clock Inputs

Function	Clock Input	FPGA Pin	Global Clock Buffer
25.175 MHz Video Clock	VIDEO_CLK	AF14	GCLK0
JX2 Clock Input	JX2_SE_CLK_IN	AE14	GCLK1
DDR Clock Feedback	DDR_CLK_FB_IN	AD13	GCLK2
PCIe RX Clock	RXCLK	AE13	GCLK3
User Clock - SMT	CLK_SMT	C14	GCLK4
User Clock - Socket	CLK_SOCKET	B14	GCLK5
50 MHz Clock	CLK_50MHZ	A13	GCLK6
JX1 Clock Input	JX1_SE_CLK_IN	B13	GCLK7

Two user clock options are available through the user clock socket (U10) and the unpopulated SMT layout at U26. Both of these locations can accept a user supplied, 3.3V compatible clock oscillator device.

Connections

UCF Example Constraints for Global Clock Inputs

```
Net CLK_50MHZ          LOC = A13  | IOSTANDARD = LVCMOS25; # U20 - 50MHz OSC
```

```

Net CLK_SOCKET      LOC = B14 | IOSTANDARD = LVCMOS25; # U10 - 3.3V OSC
Socket
Net CLK_SMT         LOC = C14 | IOSTANDARD = LVCMOS25; # U26 - 3.3V OSC SMT

NET VIDEO_CLK       LOC = AF14 | IOSTANDARD = LVCMOS25;

Net JX1_SE_CLK_IN   LOC = B13 ; # B0_GCLK7
Net JX2_SE_CLK_IN   LOC = AE14 ; # B4_GCLK1

NET "rxclk"         LOC = "AE13" | IOSTANDARD = SSTL2_I | IOBDELAY =
NONE ;

Net DDR_CLK_FB_IN   LOC = AD13 | IOSTANDARD = SSTL2_I;

```

Switches and Buttons

Overview

The Spartan-3 Board for PCI Express has three user-defined push-button switches (USR1, USR2, and RST) and four user defined DIP switches (SW5-1 through SW5-4). Although the RESET push button is labeled “RST” its actual function is still user defined.

Operation

Figure 5-1 shows the basic circuit for the four DIP switches on the Spartan-3 Board for PCI Express. All four DIP switches are packaged in the SW5 four position DIP switch. When the DIP switch is in the ON position, the signal to the FPGA pin is high, and when the switch is in the OFF position, the signal is low.

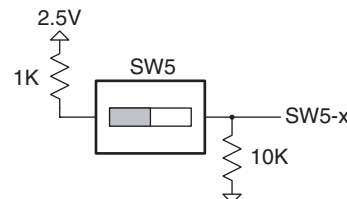


Figure 5-1: DIP Switch

Figure 5-2 shows the basic circuit for the three push button switches on the Spartan-3 Board for PCI Express. When the push button switch is pressed, the signal to the FPGA is low, and when the switch is not pressed, the signal is high.

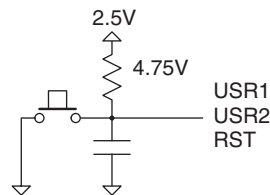


Figure 5-2: Push Button Switch

Connections

UCF examples of the various switch signals are provided in this section.

UCF Example Constraints for Board Switches

```
Net FPGA_RESETEn      LOC = F21 | IOSTANDARD = LVCMOS25; # Push button
SW4

Net PUSH_BUTTON<0>    LOC = N5   | IOSTANDARD = LVCMOS33; # SW3 - User PB1
Net PUSH_BUTTON<1>    LOC = K22  | IOSTANDARD = LVCMOS33; # SW2 - User PB2

Net DIP_SW<0>          LOC = F20  | IOSTANDARD = LVCMOS25; # SW5:1
Net DIP_SW<1>          LOC = G19  | IOSTANDARD = LVCMOS25; # SW5:2
Net DIP_SW<2>          LOC = B23  | IOSTANDARD = LVCMOS25; # SW5:3
Net DIP_SW<3>          LOC = F11  | IOSTANDARD = LVCMOS25; # SW5:4
```

LEDs

Overview

The Spartan-3 Board for PCI Express has eight user LEDs that are located along the top of the board, just to the right of the push-button switches.

Operation

Figure 6-1 shows the basic circuit diagram for the LEDs on the Spartan-3 Board for PCI Express. A high logic level on the FPGA pin will turn the corresponding LED ON and a low logic level will turn the LED OFF.

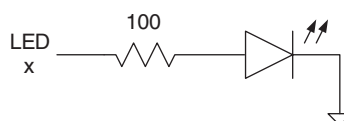


Figure 6-1: Basic Circuit LED Diagram

Connections

A UCF example of the LED signals is listed below.

NET LED<0>	LOC = H11	IOSTANDARD = LVCMOS25; # LED1
NET LED<1>	LOC = C22	IOSTANDARD = LVCMOS25;
NET LED<2>	LOC = C23	IOSTANDARD = LVCMOS25;
NET LED<3>	LOC = N19	IOSTANDARD = LVCMOS33;
NET LED<4>	LOC = A22	IOSTANDARD = LVCMOS25; # LED5
NET LED<5>	LOC = A23	IOSTANDARD = LVCMOS25;
NET LED<6>	LOC = D16	IOSTANDARD = LVCMOS25;
NET LED<7>	LOC = E18	IOSTANDARD = LVCMOS25; # LED8

VGA Display Port

Overview

The Spartan-3 Board for PCI Express includes a video display port through an on-board Philips TDA8777 Triple DAC. Although the DAC is capable of 10-bit resolution, only 8-bits per color (RGB) are connected to the FPGA. A DB15 connector (P2) is provided for easy connection to standard monitors.

Operation

Figure 7-1 shows a functional diagram of the Video Port implemented on the Spartan-3 Board for PCI Express.

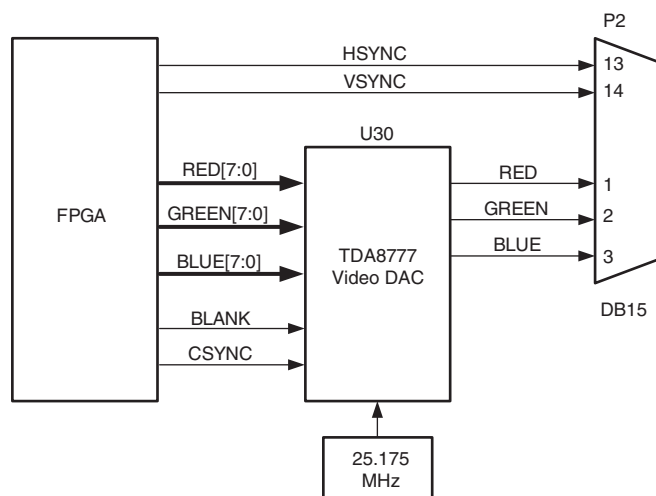


Figure 7-1: Video Port Functional Diagram

The FPGA is responsible for driving the video DAC with 8-bits of red, green, and blue. It must also provide the timing and control signals to both the DAC and monitor itself. The provided 25.175 MHz clock oscillator enables 640 x 480 VGA resolution.

Two jumpers, JP8 and JP11, are also part of this circuit. JP8 enables and disables the 25.175 MHz oscillator. When JP8 is set to pins 1–2, the oscillator is disabled. When it is set to pins 2–3, it is enabled. Jumper JP11 controls the power mode of the Video DAC. JP11 is open for normal DAC operation and jumpered closed to place the video DAC in power-save mode.

Connections

A UCF example of the Video Port is shown below.

NET DAC_B<0>	LOC = W15	IOSTANDARD = LVCMOS25;
NET DAC_B<1>	LOC = AA13	IOSTANDARD = LVCMOS25;
NET DAC_B<2>	LOC = Y11	IOSTANDARD = LVCMOS25;
NET DAC_B<3>	LOC = AB9	IOSTANDARD = LVCMOS25;
NET DAC_B<4>	LOC = AA9	IOSTANDARD = LVCMOS25;
NET DAC_B<5>	LOC = Y9	IOSTANDARD = LVCMOS25;
NET DAC_B<6>	LOC = Y8	IOSTANDARD = LVCMOS25;
NET DAC_B<7>	LOC = AB7	IOSTANDARD = LVCMOS25;
NET DAC_G<0>	LOC = AA17	IOSTANDARD = LVCMOS25;
NET DAC_G<1>	LOC = AA16	IOSTANDARD = LVCMOS25;
NET DAC_G<2>	LOC = AC16	IOSTANDARD = LVCMOS25;
NET DAC_G<3>	LOC = AB16	IOSTANDARD = LVCMOS25;
NET DAC_G<4>	LOC = AB15	IOSTANDARD = LVCMOS25;
NET DAC_G<5>	LOC = AA15	IOSTANDARD = LVCMOS25;
NET DAC_G<6>	LOC = Y16	IOSTANDARD = LVCMOS25;
NET DAC_G<7>	LOC = W16	IOSTANDARD = LVCMOS25;
NET DAC_R<0>	LOC = AC21	IOSTANDARD = LVCMOS25;
NET DAC_R<1>	LOC = AB21	IOSTANDARD = LVCMOS25;
NET DAC_R<2>	LOC = AA20	IOSTANDARD = LVCMOS25;
NET DAC_R<3>	LOC = AB20	IOSTANDARD = LVCMOS25;
NET DAC_R<4>	LOC = Y18	IOSTANDARD = LVCMOS25;
NET DAC_R<5>	LOC = AA18	IOSTANDARD = LVCMOS25;
NET DAC_R<6>	LOC = AB17	IOSTANDARD = LVCMOS25;
NET DAC_R<7>	LOC = AC17	IOSTANDARD = LVCMOS25;
NET DAC_HSYNC	LOC = AB6	IOSTANDARD = LVCMOS25;
NET DAC_VSYNC	LOC = AA7	IOSTANDARD = LVCMOS25;
NET DAC_CS	LOC = AA6	IOSTANDARD = LVCMOS25;
NET DAC_BLANK	LOC = AD4	IOSTANDARD = LVCMOS25;
NET VIDEO_CLK	LOC = AF14	IOSTANDARD = LVCMOS25;

Related Resources

For more information about the Philips TDA8777 Video DAC, go to the Philips website, www.semiconductors.philips.com.

RS-232 PORT

Overview

The Spartan-3 Board for PCI Express provides a DCE compatible RS-232 serial port for connection to most computer serial ports.

Operation

Figure 8-1 shows the connection between the FPGA and the DB9 connector. The FPGA supplies serial output data using 2.5V LVC MOS levels to the TI device, which in turn converts the logic value to the appropriate RS-232 voltage level. Similarly, the TI device converts the RS-232 serial input data to 2.5V LVC MOS levels for the FPGA.

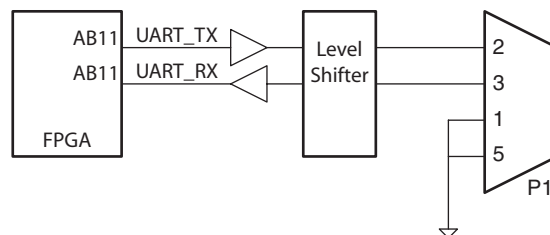


Figure 8-1: RS 232 Serial Port Diagram

Connections

A UCF example of the RS-232 serial port signals is listed below.

```
Net UART_RX      LOC = AA11 | IOSTANDARD = LVCMOS25;
Net UART_TX      LOC = AB11 | IOSTANDARD = LVCMOS25;
```


DDR SDRAM

Overview

The Spartan-3 Board for PCI Express includes two 512 Mb (32M x 16) Micron Technology DDR SDRAM (MT46V32M16). This provides a 32-bit data interface to the Spartan-3 FPGA. All DDR SDRAM interface pins connect to the FPGA I/O Banks 0 and 1—with the exception of the DDR clock feedback pin, which is in Bank 5. These three FPGA banks and the DDR SDRAM chips are powered by 2.5V, generated by a PTH03000W regulator from the 3.3V supply rail of the board. The 1.25V reference voltage, common to the FPGA and DDR SDRAM, is generated using TPS51100.

All DDR SDRAM interface signals are terminated. An SSTL_2, Class II I/O standard is used. The system supply for signal termination resistors uses the TPS51100, which sets the termination voltage equal to VREF and tracks variations in the DC level of VREF.

The DDR feedback clock pin DDR_CLK_FB (output Pin B20) is fed back into FPGA pin AD13 in I/O Bank 5 to have best access to one of the FPGA Digital Clock Managers (DCMs). This path is required when using the MicroBlaze OPB DDR controller. The MicroBlaze OPB DDR SDRAM controller IP core documentation is also available from within the EDK 8.1i development software.

Operation

The 512Mb DDR SDRAM is a high-speed CMOS, dynamic random-access memory containing 536,870,912 bits. It is internally configured as a quad-bank DRAM.

The 512Mb DDR SDRAM uses a double-data-rate architecture to achieve high-speed operation. The double-data-rate architecture is essentially a 2n-prefetch architecture with an interface designed to transfer two data words per clock cycle at the I/O pins. A single read or write access for the 512Mb DDR SDRAM effectively consists of a single 2n-bit wide, one-clock-cycle data transfer at the internal DRAM core and two corresponding n-bit wide, one-half-clock-cycle data transfers at the I/O pins.

A bidirectional data strobe (DQS) is transmitted externally, along with data, for use in data capture at the receiver. DQS is a strobe transmitted by the DDR SDRAM during READs and by the memory controller during WRITEs. DQS is edge-aligned with data for READs and center-aligned with data for WRITEs. The x16 offering has two data strobes, one for the lower byte and one for the upper byte.

The 512Mb DDR SDRAM operates from a differential clock (CK and CK#); the crossing of CK going HIGH and CK# going LOW will be referred to as the positive edge of CK. Commands (address and control signals) are registered at every positive edge of CK. Input data is registered on both edges of DQS, and output data is referenced to both edges of DQS, as well as to both edges of CK.

Read and write accesses to the DDR SDRAM are burst oriented; accesses start at a selected location and continue for a programmed number of locations in a programmed sequence. Accesses begin with the registration of an ACTIVE command, which may then be followed by a READ or WRITE command. The address bits registered coincident with the ACTIVE command are used to select the bank and row to be accessed. The address bits registered coincident with the READ or WRITE command are used to select the bank and the starting column location for the burst access.

The DDR SDRAM provides for programmable READ or WRITE burst lengths of 2, 4, or 8 locations. An auto precharge function may be enabled to provide a self-timed row precharge that is initiated at the end of the burst access.

As with standard SDR SDRAMs, the pipelined, multibank architecture of DDR SDRAMs allows for concurrent operation, thereby providing high effective bandwidth by hiding row precharge and activation time.

An auto refresh mode is provided, along with a power-saving power-down mode. All inputs are compatible with the JEDEC Standard for SSTL_2. All full drive option outputs are SSTL_2, Class II compatible.

Table 9-1: DDR SDRAM Connections

Category	DDR SDRAM Signal Name	FPGA Pin Number	Function
Address	DDR_A<12>	H15	Address Inputs
	DDR_A<11>	G15	
	DDR_A<10>	F16	
	DDR_A<9>	E16	
	DDR_A<8>	H16	
	DDR_A<7>	G16	
	DDR_A<6>	E17	
	DDR_A<5>	D17	
	DDR_A<4>	G17	
	DDR_A<3>	F17	
	DDR_A<2>	G18	
	DDR_A<1>	F18	
	DDR_A<0>	B19	
	DDR_BA<1>	A19	Bank address inputs
	DDR_BA<0>	E20	

Table 9-1: DDR SDRAM Connections (*Continued*)

Control	DDR_CKE	D20	Clock Enable
	DDR_CS _n	B21	Chip select
	DDR_RAS _n	A21	Row address strobe
	DDR_CAS _n	D21	Column address strobe
	DDR_WEn	C21	Write enable
	DDR_Clk<1>	F15	Clock for upper 16-bits
	DDR_Clk _n <1>	E15	
	DDR_Clk<0>	D7	Clock for lower 16-bits
	DDR_Clk _n <0>	E7	
	DDR_CLK_FB_OUT	B20	Clock output feedback
	DDR_CLK_FB_IN	AD13	Clock input feedback
	DDR_DM<3>	H14	Data mask. Upper and lower data masks for each DDR chip
	DDR_DM<2>	A12	
	DDR_DM<1>	H13	
	DDR_DM<0>	D6	
	DDR_DQS<3>	G13	Data strobe. Upper and lower data strobes for each DDR chip
	DDR_DQS<2>	G11	
	DDR_DQS<1>	A6	
	DDR_DQS<0>	A3	

Table 9-1: DDR SDRAM Connections (Continued)

Data	DDR_D<31>	B15	Data input/output (U4)
	DDR_D<30>	A15	
	DDR_D<29>	G14	
	DDR_D<28>	F14	
	DDR_D<27>	E14	
	DDR_D<26>	A14	
	DDR_D<25>	D13	
	DDR_D<24>	E13	
	DDR_D<23>	B12	
	DDR_D<22>	C12	
	DDR_D<21>	E12	
	DDR_D<20>	F12	
	DDR_D<19>	G12	
	DDR_D<18>	H12	
	DDR_D<17>	D11	
	DDR_D<16>	E11	
	DDR_D<15>	E10	Data input/output (U3)
	DDR_D<14>	F10	
	DDR_D<13>	F9	
	DDR_D<12>	G9	
	DDR_D<11>	A8	
	DDR_D<10>	B8	
	DDR_D<9>	A7	
	DDR_D<8>	B7	
	DDR_D<7>	B6	
	DDR_D<6>	C6	
	DDR_D<5>	E6	
	DDR_D<4>	A5	
	DDR_D<3>	B5	
	DDR_D<2>	C5	
	DDR_D<1>	B4	
	DDR_D<0>	C4	

UCF

Address

Listed below are the UCF constraints for the DDR SDRAM address pins, including the I/O pin assignment and the I/O standard used.

Net DDR_A<12>	LOC = H15	IOSTANDARD = SSTL2_I;
Net DDR_A<11>	LOC = G15	IOSTANDARD = SSTL2_I;
Net DDR_A<10>	LOC = F16	IOSTANDARD = SSTL2_I;
Net DDR_A<9>	LOC = E16	IOSTANDARD = SSTL2_I;
Net DDR_A<8>	LOC = H16	IOSTANDARD = SSTL2_I;
Net DDR_A<7>	LOC = G16	IOSTANDARD = SSTL2_I;
Net DDR_A<6>	LOC = E17	IOSTANDARD = SSTL2_I;
Net DDR_A<5>	LOC = D17	IOSTANDARD = SSTL2_I;
Net DDR_A<4>	LOC = G17	IOSTANDARD = SSTL2_I;
Net DDR_A<3>	LOC = F17	IOSTANDARD = SSTL2_I;
Net DDR_A<2>	LOC = G18	IOSTANDARD = SSTL2_I;
Net DDR_A<1>	LOC = F18	IOSTANDARD = SSTL2_I;
Net DDR_A<0>	LOC = B19	IOSTANDARD = SSTL2_I;
Net DDR_BA<1>	LOC = A19	IOSTANDARD = SSTL2_I;
Net DDR_BA<0>	LOC = E20	IOSTANDARD = SSTL2_I;

Control

Listed below are the UCF constraints for the DDR SDRAM control pins, including the I/O pin assignment and the I/O standard used.

Net DDR_CKE	LOC = D20	IOSTANDARD = SSTL2_I;
Net DDR_CS _n	LOC = B21	IOSTANDARD = SSTL2_I;
Net DDR_RAS _n	LOC = A21	IOSTANDARD = SSTL2_I;
Net DDR_CAS _n	LOC = D21	IOSTANDARD = SSTL2_I;
Net DDR_WEn	LOC = C21	IOSTANDARD = SSTL2_I;
Net DDR_Clk<1>	LOC = F15	IOSTANDARD = SSTL2_I; # U4 CK
Net DDR_Clk _n <1>	LOC = E15	IOSTANDARD = SSTL2_I; # U4 CK _N
Net DDR_Clk<0>	LOC = D7	IOSTANDARD = SSTL2_I; # U3 CK
Net DDR_Clk _n <0>	LOC = E7	IOSTANDARD = SSTL2_I; # U3 CK _N
Net DDR_CLK_FB_OUT	LOC = B20	IOSTANDARD = SSTL2_I;
Net DDR_CLK_FB_IN	LOC = AD13	IOSTANDARD = SSTL2_I;
Net DDR_DM<3>	LOC = H14	IOSTANDARD = SSTL2_I; # U4 UDM
Net DDR_DM<2>	LOC = A12	IOSTANDARD = SSTL2_I; # U4 LDM
Net DDR_DM<1>	LOC = H13	IOSTANDARD = SSTL2_I; # U3 UDM
Net DDR_DM<0>	LOC = D6	IOSTANDARD = SSTL2_I; # U3 LDM
Net DDR_DQS<3>	LOC = G13	IOSTANDARD = SSTL2_II; # U4 UDQS
Net DDR_DQS<2>	LOC = G11	IOSTANDARD = SSTL2_II; # U4 LDQS
Net DDR_DQS<1>	LOC = A6	IOSTANDARD = SSTL2_II; # U3 UDQS
Net DDR_DQS<0>	LOC = A3	IOSTANDARD = SSTL2_II; # U3 LDQS

Data

This section provides the UCF constraints for the DDR SDRAM data pins, including the I/O pin assignment and the I/O standard used.

Net DDR_D<31>	LOC = B15	IOSTANDARD = SSTL2_II; # U4 DQ15
Net DDR_D<30>	LOC = A15	IOSTANDARD = SSTL2_II;
Net DDR_D<29>	LOC = G14	IOSTANDARD = SSTL2_II;
Net DDR_D<28>	LOC = F14	IOSTANDARD = SSTL2_II;
Net DDR_D<27>	LOC = E14	IOSTANDARD = SSTL2_II;
Net DDR_D<26>	LOC = A14	IOSTANDARD = SSTL2_II;
Net DDR_D<25>	LOC = D13	IOSTANDARD = SSTL2_II;
Net DDR_D<24>	LOC = E13	IOSTANDARD = SSTL2_II;
Net DDR_D<23>	LOC = B12	IOSTANDARD = SSTL2_II;
Net DDR_D<22>	LOC = C12	IOSTANDARD = SSTL2_II;
Net DDR_D<21>	LOC = E12	IOSTANDARD = SSTL2_II;
Net DDR_D<20>	LOC = F12	IOSTANDARD = SSTL2_II;
Net DDR_D<19>	LOC = G12	IOSTANDARD = SSTL2_II;
Net DDR_D<18>	LOC = H12	IOSTANDARD = SSTL2_II;
Net DDR_D<17>	LOC = D11	IOSTANDARD = SSTL2_II;
Net DDR_D<16>	LOC = E11	IOSTANDARD = SSTL2_II; # U4 DQ0
Net DDR_D<15>	LOC = E10	IOSTANDARD = SSTL2_II; # U3 DQ15
Net DDR_D<14>	LOC = F10	IOSTANDARD = SSTL2_II;
Net DDR_D<13>	LOC = F9	IOSTANDARD = SSTL2_II;
Net DDR_D<12>	LOC = G9	IOSTANDARD = SSTL2_II;
Net DDR_D<11>	LOC = A8	IOSTANDARD = SSTL2_II;
Net DDR_D<10>	LOC = B8	IOSTANDARD = SSTL2_II;
Net DDR_D<9>	LOC = A7	IOSTANDARD = SSTL2_II;
Net DDR_D<8>	LOC = B7	IOSTANDARD = SSTL2_II;
Net DDR_D<7>	LOC = B6	IOSTANDARD = SSTL2_II;
Net DDR_D<6>	LOC = C6	IOSTANDARD = SSTL2_II;
Net DDR_D<5>	LOC = E6	IOSTANDARD = SSTL2_II;
Net DDR_D<4>	LOC = A5	IOSTANDARD = SSTL2_II;
Net DDR_D<3>	LOC = B5	IOSTANDARD = SSTL2_II;
Net DDR_D<2>	LOC = C5	IOSTANDARD = SSTL2_II;
Net DDR_D<1>	LOC = B4	IOSTANDARD = SSTL2_II;
Net DDR_D<0>	LOC = C4	IOSTANDARD = SSTL2_II; # U3 DQ0

Related Resources

Below listed are related resources. Click any of the following links to download the desired documentation.

[Xilinx Embedded Design Kit \(EDK\)](#)

[MT46V32M16 \(32M x 16\) DDR SDRAM Data Sheet](#)

[MicroBlaze OPB Double Data Rate \(DDR\) SDRAM Controller \(v2.00b\)](#)

SPI Serial Flash

Overview

The Spartan-3 Board for PCI Express includes a STMicroelectronics M25P40 4 Mb SPI serial Flash, useful in a variety of applications, such as:

- Simple non-volatile data storage
- Storage for identifier codes, serial numbers, IP addresses, etc.
- Storage of MicroBlaze processor code that can be shadowed into DDR SDRAM.

Operation

The M25P40 is a 4 Mb (512K x 8) Serial Flash Memory, with advanced write protection mechanisms, accessed by a high speed SPI-compatible bus.

The memory can be programmed 1 to 256 bytes at a time, using the Page Program instruction. The memory is organized as 8 sectors, each containing 256 pages. Each page is 256 bytes wide. Thus, the whole memory can be viewed as consisting of 2048 pages, or 524,288 bytes. The whole memory can be erased using the Bulk Erase instruction, or a sector at a time, using the Sector Erase instruction.

On the Spartan-3 Board for PCI Express, the SPI Serial Flash signals are placed in Banks 2 and 3. These banks have an adjustable I/O voltage to add flexibility to the EXP connector. Therefore, the FPGA I/O voltage could be either 2.5V or 3.3V, depending on the setting of JP5. Voltage translator buffers are used to enable the 3.3V serial flash to connect to the FPGA. The `SPROM_EN` signal turns on these translator buffers. If the design uses the serial flash, this signal must be asserted.

The SPI Serial Flash can be programmed directly by the FPGA using MicroBlaze™ or PicoBlaze™. The serial flash pins can also be accessed through connector J6. This is compatible with the Xilinx XSPI programming utility (see XAPP445). To enable serial flash programming via the J6 JTAG connector, the FPGA must drive the `SPROM_ENABLE` pin (K6) from the FPGA to a logic level low. When `SPROM_ENABLE` is driven high, the FPGA has access to the Serial Flash interface. [Figure 10-1](#) shows the Serial Flash connection options.

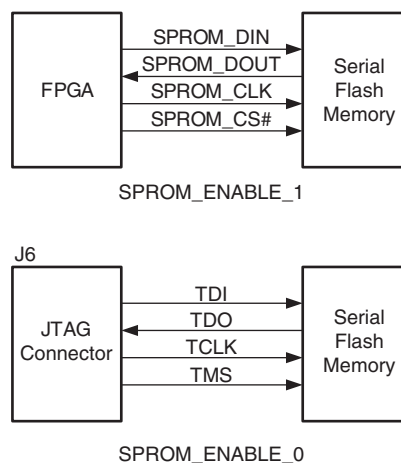


Figure 10-1: Serial Flash Interfaces

Connections

A UCF example of the SPI Serial Flash is shown below.

```
#=====
# SPI PROM - U33
#=====

# IOSTANDARD depends on JP5 jumper selection except "SPROM_EN"

Net SPROM_DIN      LOC = P19 | IOSTANDARD = LVCMOS25/33;
Net SPROM_DOUT     LOC = M22 | IOSTANDARD = LVCMOS25/33;
Net SPROM_EN       LOC = K6  | IOSTANDARD = LVCMOS33;
Net SPROM_CS[n]    LOC = P23 | IOSTANDARD = LVCMOS25/33;
Net SPROM_CLK      LOC = M21 | IOSTANDARD = LVCMOS25/33;
```

Related Resources

Click on the following links to download the desired supporting documentation.

[STMicro M25P40 datasheet](#)

[Xilinx Application Note XAPP445](#)

FPGA Configuration

Overview

The Spartan-3 Starter Kit Board for PCI-Express supports two FPGA configuration options:

- Download FPGA designs directly to the Spartan-3 FPGA via JTAG, using the download cable interface.
- Program the on-board 8 Mb Xilinx XCF08P parallel Platform Flash PROM, then configure the FPGA from the image stored in the Platform Flash PROM using either Master Serial or Master SelectMAP (Parallel) modes.

The configuration mode jumpers determine which configuration mode the FPGA uses when power is first applied, or whenever the PROG button is pressed. The DONE pin LED lights when the FPGA successfully finishes configuration. Pressing the PROG button forces the FPGA to restart its configuration process.

The JTAG download cable interface supports the Xilinx USB-JTAG and the Parallel Cable IV. Xilinx Parallel Cable III or Memec IJC-x cables are not supported. This interface also provides in-system programming for the on-board Platform Flash PROM. This interface is accessed through connector socket J2.

Note: Do not confuse the J2 socket with the Serial Flash programming socket (J6), which is identical.

The 4 Mb Xilinx Platform Flash PROM provides easy, JTAG-programmable configuration storage for the FPGA. The FPGA configures from the Platform Flash using either Master Serial or Master SelectMAP modes.

Operation

Configuration Mode jumpers

As shown in [Table 11-1](#), the JP3 jumper block settings control the FPGA configuration mode. Inserting a jumper grounds the associated mode pin. Insert or remove individual jumpers to select the FPGA configuration mode and associated configuration source.

Table 11-1: Configuration Mode Jumper Settings

Configuration Mode	Mode Pins M2:M1:M0	FPGA Configuration Image Source	Jumper Settings
JTAG	1:1:1	J6 programming socket	M1 installed
Master Serial	0:0:0	Platform flash in serial mode (DO only)	M0, M1, M2 installed
Master SelectMAP	0:1:1	Platform flash in parallel mode (D[07]used)	M2 installed

PROG Switch

SW1 causes a toggling of the PROG signal on the FPGA, which forces a reconfiguration based on the current setting of the Mode jumpers.

DONE LED

D14 is the DONE LED. If the FPGA successfully configures, then DONE is asserted, which causes D14 to be lit. This is the only blue colored LED on the board.

Platform Flash Design Revisioning

With the XCF08P Platform Flash, design revisioning is supported. The XC3S1000 FPGA requires 3,223,488 bits to configure. However, the XCF08P device has 8 Mbits of storage—more than twice as much as required to hold a single XC3S1000 bitstream.

The Spartan-3 PCI Express board gives you access to the EN_EXT_SEL_n and REV_SEL[1:0] pins through jumper block JP4. By creating a design-revisioned image and adding jumpers to these pins, you can have two alternative FPGA images for the board stored in the Platform Flash simultaneously.

Using the Platform Flash for User Storage

The Spartan-3 Board for PCI Express provides the necessary connections to allow you to use the Platform Flash for user storage (as discussed in [XAPP694](#)).

Jumper J4 makes the connection between PROM_CE# and FPGA_DONE. Under typical PROM operation, this jumper should be installed. However, to exercise the technique discussed in [XAPP694](#), this jumper is removed.

This board also provides signals FPGA_PROM_READ, FPGA_tο_PROM_CCLK, and FPGA_tο_PROM_CE#, also used when implemented as described in [XAPP694](#).

Reprogramming Platform Flash from the FPGA

The Spartan-3 Board for PCI Express provides connections between FPGA I/Os and the JTAG chain. This makes possible the creation of a design (like MicroBlaze) to reprogram the Platform Flash PROM from the FPGA (see [XAPP058](#)). These signals include FPGA_to_PROM, FPGA_TDI_to_PROM, FPGA_TCK_to_PROM, FPGA_TMS_to_PROM, and PROM_TDO_to_FPGA.

Connections

A UCF example listed below.

```
#####
# Platform Flash PROM Config (JTAG) - U24
#####

Net FPGA_to_PROM      LOC = AA10 | IOSTANDARD = LVCMOS25;
Net FPGA_TDI_to_PROM   LOC = W12  | IOSTANDARD = LVCMOS25;
Net FPGA_TCK_to_PROM   LOC = Y13  | IOSTANDARD = LVCMOS25;
Net FPGA_TMS_to_PROM   LOC = W13  | IOSTANDARD = LVCMOS25;
Net PROM_TDO_to_FPGA   LOC = AB10 | IOSTANDARD = LVCMOS25;

# IOSTANDARD depends on JP5 jumper selection
Net FPGA_PROM_READ     LOC = R20  | IOSTANDARD = LVCMOS25/33;
Net FPGA_to_PROM_CCLK  LOC = AB10 | IOSTANDARD = LVCMOS25/33;
Net FPGA_to_PROM_CE     LOC = AC25 | IOSTANDARD = LVCMOS25/33;
```

Related Resources

Click the links below to download related documentation.

[Xilinx Platform Flash datasheet](#)

[Xilinx Application Note XAPP694](#)

[Xilinx Application Note XAPP058](#)

Power Supplies

Overview

The Spartan-3 Board for PCI Express provides complete and stable power for all components on the board as well as the EXP expansion connectors. The board can be powered from multiple sources, such as:

- 3.3V PCI Express edge connector
- 5V ATX power supply disk drive socket (supply not provided in the Starter Kit)
- 5V barrel jack (supply not provided in the Starter Kit)

The input source is selected by the position of a fuse on the board.

Input Power Regulations

Several power rails are regulated based on the input power.

5V to 3.3V Regulation

A PTH050101W DC/DC switching module provides 5V to 3.3V conversion when either the disk drive or barrel jack is used to provide 5V input power. The PTH050101W is capable of producing 15A of output voltage. A TPS3828 provides input voltage tracking.

If a barrel jack is used, then SW6 allows the user to switch power on and off without unplugging the supply.

3.3V is used for the I/O voltage on the FPGA, source voltage for EXP, the Video DAC, the SPI Serial Flash, the clocks oscillators, RS-232, LEDs, switches, and PX1011A PHY VDD1.

3.3V to 2.5V Regulation

A PTH03000W DC/DC switching module provides 3.3V to 2.5V conversion with a maximum output of 6A.

2.5V is used for the I/O voltage on the FPGA, source voltage for EXP, the DDR, Platform Flash I/O, the JTAG chain, and PX1011A PHY VDD2.

3.3V to 1.2V Regulation

A PTH03000W DC/DC switching module provides 3.3V to 1.2V conversion with a maximum output of 6A.

1.2V is used for the FPGA internal voltage and the PX1011A PHY VDD4.

2.5V to 1.8V Regulation

A TPS72501 DC/DC linear regulator provides 2.5V to 1.8V conversion with a maximum output of 1A.

1.8V is used for the Platform Flash internal voltage.

1.25V Regulation

1.25V is needed for the PX1011A PHY reference voltage, the DDR reference voltage, and the PHY and DDR termination tracking voltage.

The TPS736125 linear regulator produces the 1.25V PHY reference voltage from 2.5V.

The TPS51100 produces both the 1.25V reference for the DDR as well as the 1.25V termination tracking voltage for both the PHY and the DDR. The TPS51100 has a 2.5V source input and a 5V reference input. [Figure 12-1](#) is a diagram of the SSTL2 termination.

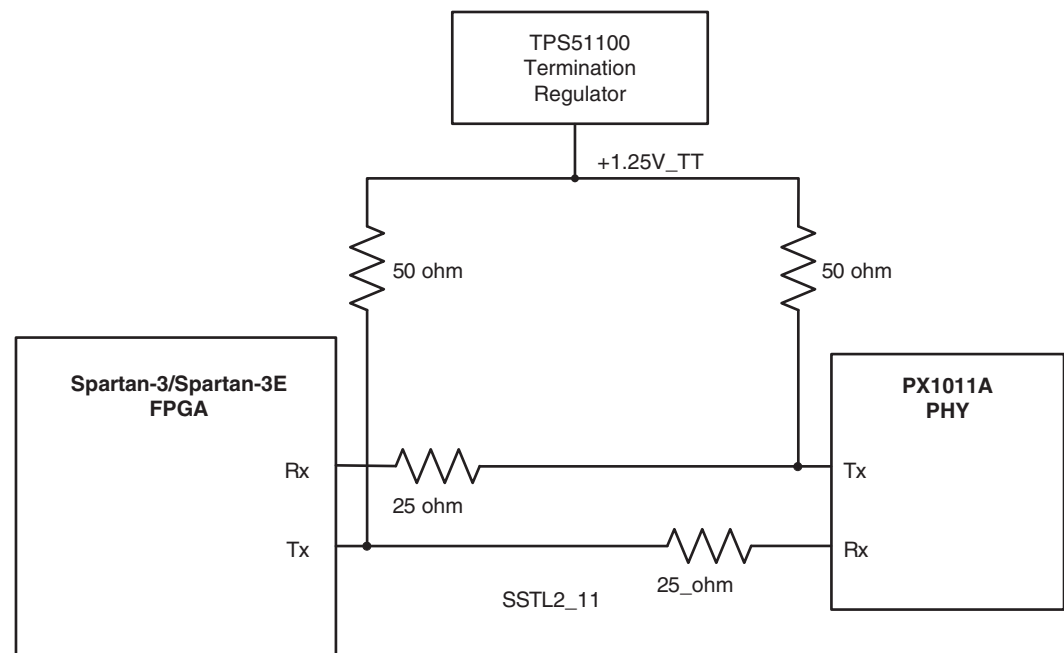


Figure 12-1: SSTL2 Termination Diagram

5V to 3.3V Regulation

A TPS60131 boost regulator provides 5V from 3.3V. This is necessary in the scenario that the PCI Express edge connector is the source voltage.

This 5V is used as a reference to the termination tracking voltage regulator. It is also used as a source voltage for the POWER GOOD LEDs.

Voltage Supervision

The TPS3126 and TPS3307 devices provide voltage supervision for the board. 1.2V, 1.8V, 2.5V, and 3.3V are monitored. When all of these supplies have regulated properly, the PROG signal to the FPGA is released such that configuration can begin. Pressing SW1 causes a reset on the TPS3126 supervisor, which in turn causes PROG to be toggled causing a reconfiguration.

Operation

When power is properly applied to the board, the following POWER GOOD LEDs are lit:

- D12 for 1.2V
- D15 for 1.25V Termination Voltage
- D11 for 2.5V
- D13 for 3.3V

PCI Express Edge Connector as Source

1. Place the fuse in socket F2.
Do NOT place a separate fuse in F1 as it may damage the board.
2. Plug the board into the PC.
3. When power is applied to the PC, power will immediately be distributed to the board.

Disk Drive Connector as Source

1. Place the fuse in socket F1.
Do NOT place a separate fuse in F2 as it may damage the board.
2. Plug the board into the PC.
3. Plug in a spare disk drive connector to socket J1 on the board.
4. When power is applied to the PC, power will immediately be distributed to the board.

Barrel Jack as Source

This mode is intended for benchtop operation only. This is useful for programming the PROM or working on a non-PCI Express design.

1. Place the fuse in socket F1.
Do NOT place a separate fuse in F2 as it may damage the board.
2. Insert a 5V source into the barrel jack.
3. Move the power switch to the ON position.

Related Resources

Click the link below to download related documentation.

[TI Power References](#)

Example User Constraints File (UCF)

```
#-----
# MASTER Constraints File for the Xilinx Spartan-3 PCIe Starter Board:
#
#-----

#-----
# Xilinx Technology : Spartan-3
# Part              : XC3S1000
# Package           : FG676
# Speed Grade       : -4
#-----
# Revision : 1.0 <Original Release>
# Date     : 3/01/2006
# Project  :
#=====

#*****
#
#
# DISCLAIMER:
#
# Avnet, Inc. makes no warranty for the use of this code*
# or design. This code is provided "As Is". Avnet, Inc *
# assumes no responsibility for any errors, which may *
# appear in this code, nor does it make a commitment *
# to update the information contained herein. *
# Avnet, Inc specifically disclaims any implied *
# warranties of fitness for a particular purpose. *
#
#*****

#####
# User Constraints (Pinout, Placement, Etc.)
#####

#=====
# CLOCKS, SWITCHES, LEDS, RS323 (UART)
#=====

Net CLK_50MHZ          LOC = A13 | IOSTANDARD = LVCMOS25; # U20 - 50MHz OSC
Net CLK_SOCKET         LOC = B14 | IOSTANDARD = LVCMOS25; # U10 - 3.3V OSC Socket
Net CLK_SMT            LOC = C14 | IOSTANDARD = LVCMOS25; # U26 - 3.3V OSC SMT

Net FPGA_RESETh        LOC = F21 | IOSTANDARD = LVCMOS25; # Push button SW4

Net PUSH_BUTTON<0>     LOC = N5  | IOSTANDARD = LVCMOS33; # SW3 - User PB1
Net PUSH_BUTTON<1>     LOC = K22 | IOSTANDARD = LVCMOS33; # SW2 - User PB2
```

```

Net DIP_SW<0>          LOC = F20 | IOSTANDARD = LVCMOS25; # SW5:1
Net DIP_SW<1>          LOC = G19 | IOSTANDARD = LVCMOS25; # SW5:2
Net DIP_SW<2>          LOC = B23 | IOSTANDARD = LVCMOS25; # SW5:3
Net DIP_SW<3>          LOC = F11 | IOSTANDARD = LVCMOS25; # SW5:4

NET LED<0>             LOC = H11 | IOSTANDARD = LVCMOS25; # LED1
NET LED<1>             LOC = C22 | IOSTANDARD = LVCMOS25;
NET LED<2>             LOC = C23 | IOSTANDARD = LVCMOS25;
NET LED<3>             LOC = N19 | IOSTANDARD = LVCMOS33;
NET LED<4>             LOC = A22 | IOSTANDARD = LVCMOS25; # LED5
NET LED<5>             LOC = A23 | IOSTANDARD = LVCMOS25;
NET LED<6>             LOC = D16 | IOSTANDARD = LVCMOS25;
NET LED<7>             LOC = E18 | IOSTANDARD = LVCMOS25; # LED8

Net UART_RX           LOC = AA11 | IOSTANDARD = LVCMOS25;
Net UART_TX           LOC = AB11 | IOSTANDARD = LVCMOS25;

#=====
# DDR SDRAM - U3 & U4
#=====

# Net Rst_DQS_Div_in   LOC = D10 | IOSTANDARD = SSTL2_I; # For MIG
# Net Rst_DQS_Div_out  LOC = C10 | IOSTANDARD = SSTL2_I; # For MIG

# Net DDR_A<13>        LOC = F13 | IOSTANDARD = SSTL2_I; # NC
Net DDR_A<12>          LOC = H15 | IOSTANDARD = SSTL2_I;
Net DDR_A<11>          LOC = G15 | IOSTANDARD = SSTL2_I;
Net DDR_A<10>          LOC = F16 | IOSTANDARD = SSTL2_I;
Net DDR_A<9>           LOC = E16 | IOSTANDARD = SSTL2_I;
Net DDR_A<8>           LOC = H16 | IOSTANDARD = SSTL2_I;
Net DDR_A<7>           LOC = G16 | IOSTANDARD = SSTL2_I;
Net DDR_A<6>           LOC = E17 | IOSTANDARD = SSTL2_I;
Net DDR_A<5>           LOC = D17 | IOSTANDARD = SSTL2_I;
Net DDR_A<4>           LOC = G17 | IOSTANDARD = SSTL2_I;
Net DDR_A<3>           LOC = F17 | IOSTANDARD = SSTL2_I;
Net DDR_A<2>           LOC = G18 | IOSTANDARD = SSTL2_I;
Net DDR_A<1>           LOC = F18 | IOSTANDARD = SSTL2_I;
Net DDR_A<0>           LOC = B19 | IOSTANDARD = SSTL2_I;

Net DDR_BA<1>          LOC = A19 | IOSTANDARD = SSTL2_I;
Net DDR_BA<0>          LOC = E20 | IOSTANDARD = SSTL2_I;

Net DDR_CKE            LOC = D20 | IOSTANDARD = SSTL2_I;
Net DDR_CSn            LOC = B21 | IOSTANDARD = SSTL2_I;
Net DDR_RASn           LOC = A21 | IOSTANDARD = SSTL2_I;
Net DDR_CASn           LOC = D21 | IOSTANDARD = SSTL2_I;
Net DDR_WEn            LOC = C21 | IOSTANDARD = SSTL2_I;

Net DDR_Clk<1>         LOC = F15 | IOSTANDARD = SSTL2_I; # U4 CK
Net DDR_Clk<1>         LOC = E15 | IOSTANDARD = SSTL2_I; # U4 CK_N
Net DDR_Clk<0>         LOC = D7  | IOSTANDARD = SSTL2_I; # U3 CK
Net DDR_Clk<0>         LOC = E7  | IOSTANDARD = SSTL2_I; # U3 CK_N
Net DDR_CLK_FB_OUT     LOC = B20 | IOSTANDARD = SSTL2_I;
Net DDR_CLK_FB_IN      LOC = AD13 | IOSTANDARD = SSTL2_I;

Net DDR_DM<3>          LOC = H14 | IOSTANDARD = SSTL2_I; # U4 UDM
Net DDR_DM<2>          LOC = A12 | IOSTANDARD = SSTL2_I; # U4 LDM
Net DDR_DM<1>          LOC = H13 | IOSTANDARD = SSTL2_I; # U3 UDM
Net DDR_DM<0>          LOC = D6  | IOSTANDARD = SSTL2_I; # U3 LDM

Net DDR_DQS<3>         LOC = G13 | IOSTANDARD = SSTL2_II; # U4 UDQS
Net DDR_DQS<2>         LOC = G11 | IOSTANDARD = SSTL2_II; # U4 LDQS
Net DDR_DQS<1>         LOC = A6  | IOSTANDARD = SSTL2_II; # U3 UDQS
Net DDR_DQS<0>         LOC = A3  | IOSTANDARD = SSTL2_II; # U3 LDQS

```

```

Net DDR_D<31>      LOC = B15 | IOSTANDARD = SSTL2_II; # U4 DQ15
Net DDR_D<30>      LOC = A15 | IOSTANDARD = SSTL2_II;
Net DDR_D<29>      LOC = G14 | IOSTANDARD = SSTL2_II;
Net DDR_D<28>      LOC = F14 | IOSTANDARD = SSTL2_II;
Net DDR_D<27>      LOC = E14 | IOSTANDARD = SSTL2_II;
Net DDR_D<26>      LOC = A14 | IOSTANDARD = SSTL2_II;
Net DDR_D<25>      LOC = D13 | IOSTANDARD = SSTL2_II;
Net DDR_D<24>      LOC = E13 | IOSTANDARD = SSTL2_II;
Net DDR_D<23>      LOC = B12 | IOSTANDARD = SSTL2_II;
Net DDR_D<22>      LOC = C12 | IOSTANDARD = SSTL2_II;
Net DDR_D<21>      LOC = E12 | IOSTANDARD = SSTL2_II;
Net DDR_D<20>      LOC = F12 | IOSTANDARD = SSTL2_II;
Net DDR_D<19>      LOC = G12 | IOSTANDARD = SSTL2_II;
Net DDR_D<18>      LOC = H12 | IOSTANDARD = SSTL2_II;
Net DDR_D<17>      LOC = D11 | IOSTANDARD = SSTL2_II;
Net DDR_D<16>      LOC = E11 | IOSTANDARD = SSTL2_II; # U4 DQ0

```

```

Net DDR_D<15>      LOC = E10 | IOSTANDARD = SSTL2_II; # U3 DQ15
Net DDR_D<14>      LOC = F10 | IOSTANDARD = SSTL2_II;
Net DDR_D<13>      LOC = F9 | IOSTANDARD = SSTL2_II;
Net DDR_D<12>      LOC = G9 | IOSTANDARD = SSTL2_II;
Net DDR_D<11>      LOC = A8 | IOSTANDARD = SSTL2_II;
Net DDR_D<10>      LOC = B8 | IOSTANDARD = SSTL2_II;
Net DDR_D<9>       LOC = A7 | IOSTANDARD = SSTL2_II;
Net DDR_D<8>       LOC = B7 | IOSTANDARD = SSTL2_II;
Net DDR_D<7>       LOC = B6 | IOSTANDARD = SSTL2_II;
Net DDR_D<6>       LOC = C6 | IOSTANDARD = SSTL2_II;
Net DDR_D<5>       LOC = E6 | IOSTANDARD = SSTL2_II;
Net DDR_D<4>       LOC = A5 | IOSTANDARD = SSTL2_II;
Net DDR_D<3>       LOC = B5 | IOSTANDARD = SSTL2_II;
Net DDR_D<2>       LOC = C5 | IOSTANDARD = SSTL2_II;
Net DDR_D<1>       LOC = B4 | IOSTANDARD = SSTL2_II;
Net DDR_D<0>       LOC = C4 | IOSTANDARD = SSTL2_II; # U3 DQ0

```

```

#=====
# Video DAC - U30
#=====

```

```

NET DAC_B<0>      LOC = W15 | IOSTANDARD = LVCMOS25;
NET DAC_B<1>      LOC = AA13 | IOSTANDARD = LVCMOS25;
NET DAC_B<2>      LOC = Y11 | IOSTANDARD = LVCMOS25;
NET DAC_B<3>      LOC = AB9 | IOSTANDARD = LVCMOS25;
NET DAC_B<4>      LOC = AA9 | IOSTANDARD = LVCMOS25;
NET DAC_B<5>      LOC = Y9 | IOSTANDARD = LVCMOS25;
NET DAC_B<6>      LOC = Y8 | IOSTANDARD = LVCMOS25;
NET DAC_B<7>      LOC = AB7 | IOSTANDARD = LVCMOS25;
NET DAC_G<0>      LOC = AA17 | IOSTANDARD = LVCMOS25;
NET DAC_G<1>      LOC = AA16 | IOSTANDARD = LVCMOS25;
NET DAC_G<2>      LOC = AC16 | IOSTANDARD = LVCMOS25;
NET DAC_G<3>      LOC = AB16 | IOSTANDARD = LVCMOS25;
NET DAC_G<4>      LOC = AB15 | IOSTANDARD = LVCMOS25;
NET DAC_G<5>      LOC = AA15 | IOSTANDARD = LVCMOS25;
NET DAC_G<6>      LOC = Y16 | IOSTANDARD = LVCMOS25;
NET DAC_G<7>      LOC = W16 | IOSTANDARD = LVCMOS25;
NET DAC_R<0>      LOC = AC21 | IOSTANDARD = LVCMOS25;
NET DAC_R<1>      LOC = AB21 | IOSTANDARD = LVCMOS25;
NET DAC_R<2>      LOC = AA20 | IOSTANDARD = LVCMOS25;
NET DAC_R<3>      LOC = AB20 | IOSTANDARD = LVCMOS25;
NET DAC_R<4>      LOC = Y18 | IOSTANDARD = LVCMOS25;
NET DAC_R<5>      LOC = AA18 | IOSTANDARD = LVCMOS25;
NET DAC_R<6>      LOC = AB17 | IOSTANDARD = LVCMOS25;
NET DAC_R<7>      LOC = AC17 | IOSTANDARD = LVCMOS25;
NET DAC_HSYNC     LOC = AB6 | IOSTANDARD = LVCMOS25;
NET DAC_VSYNC     LOC = AA7 | IOSTANDARD = LVCMOS25;

```

```

NET DAC_CSNC          LOC = AA6   | IOSTANDARD = LVCMOS25;
NET DAC_BLANK         LOC = AD4   | IOSTANDARD = LVCMOS25;
NET VIDEO_CLK         LOC = AF14  | IOSTANDARD = LVCMOS25;

#=====
# SPI PROM - U33
#=====

# IOSTANDARD depends on JP5 jumper selection except "SPROM_EN"

Net SPROM_DIN          LOC = P19 ;
Net SPROM_DOUT         LOC = M22 ;
Net SPROM_EN           LOC = K6   | IOSTANDARD = LVCMOS33;
Net SPROM_CSn          LOC = P23 ;
Net SPROM_CLK          LOC = M21 ;

#=====
# Platform Flash PROM Config (JTAG) - U24
#=====

Net FPGA_to_PROM       LOC = AA10 | IOSTANDARD = LVCMOS25;
Net FPGA_TDI_to_PROM   LOC = W12  | IOSTANDARD = LVCMOS25;
Net FPGA_TCK_to_PROM   LOC = Y13  | IOSTANDARD = LVCMOS25;
Net FPGA_TMS_to_PROM   LOC = W13  | IOSTANDARD = LVCMOS25;
Net PROM_TDO_to_FPGA   LOC = AB10 | IOSTANDARD = LVCMOS25;

#=====
# SPI PROM - U5
#=====

# IOSTANDARD depends on JP5 jumper selection

Net ID_DATA            LOC = T5;

#=====
# 8-bit AVR Microcontroller - U8
#=====

# IOSTANDARD depends on JP5 jumper selection

Net ATTINY_TX          LOC = V2;
Net ATTINY_RX          LOC = U6;
Net ATTINY_RSTn        LOC = U5;
Net ATTINY_CLK         LOC = V3;

#=====
# EXP I/O - Connector JX1
#=====

# IOSTANDARD depends on JP5 jumper selection

Net JX1_SE_IO_0        LOC = M3   ; # B6GIO_0
Net JX1_SE_IO_1        LOC = J7   ; # B7GIO_0
Net JX1_SE_IO_2        LOC = M7   ; # B6GIO_1
Net JX1_SE_IO_3        LOC = J6   ; # B7GIO_1
Net JX1_SE_IO_4        LOC = N7   ; # B6GIO_2
Net JX1_SE_IO_5        LOC = H5   ; # B7GIO_2
Net JX1_SE_IO_6        LOC = M8   ; # B6GIO_3
Net JX1_SE_IO_7        LOC = H2   ; # B7GIO_3
Net JX1_SE_IO_8        LOC = N8   ; # B6GIO_4
Net JX1_SE_IO_9        LOC = J5   ; # B7GIO_4
Net JX1_SE_IO_10       LOC = P8   ; # B6GIO_5
Net JX1_SE_IO_11       LOC = J4   ; # B7GIO_5
Net JX1_SE_IO_12       LOC = P2   ; # B6GIO_6
Net JX1_SE_IO_13       LOC = K7   ; # B7GIO_6

```

```

Net JX1_SE_IO_14      LOC = P7      ; # B6GIO_7
Net JX1_SE_IO_15      LOC = K5      ; # B7GIO_7
Net JX1_SE_IO_16      LOC = R1      ; # B6GIO_8
Net JX1_SE_IO_17      LOC = L8      ; # B7GIO_8
Net JX1_SE_IO_18      LOC = P1      ; # B6GIO_9
Net JX1_SE_IO_19      LOC = L7      ; # B7GIO_9
Net JX1_SE_IO_20      LOC = R2      ; # B6GIO_10
Net JX1_SE_IO_21      LOC = H1      ; # B7GIO_10
Net JX1_SE_IO_22      LOC = R3      ; # B6GIO_11
Net JX1_SE_IO_23      LOC = L1      ; # B7GIO_11
Net JX1_SE_IO_24      LOC = T1      ; # B6GIO_12
Net JX1_SE_IO_25      LOC = L2      ; # B7GIO_12
Net JX1_SE_IO_26      LOC = T2      ; # B6GIO_13
Net JX1_SE_IO_27      LOC = L4      ; # B7GIO_13
Net JX1_SE_IO_28      LOC = V6      ; # B7GIO_14
Net JX1_SE_IO_29      LOC = U7      ; # B7GIO_15
Net JX1_SE_IO_30      LOC = W5      ; # B6GIO_14
Net JX1_SE_IO_31      LOC = V7      ; # B6GIO_15
Net JX1_SE_IO_32      LOC = R8      ; # B7GIO_17
Net JX1_SE_IO_33      LOC = R7      ; # B7GIO_18

Net JX1_SE_CLK_IN     LOC = B13     ; # B0_GCLK7
Net JX1_SE_CLK_OUT    LOC = T4      ; # B7GIO_16

Net JX1_DIFF_N_0      LOC = AB4     ; # B6GPIOn_0
Net JX1_DIFF_N_1      LOC = W7      ; # B7GPIOn_0
Net JX1_DIFF_N_2      LOC = AC2     ; # B6GPIOn_1
Net JX1_DIFF_N_3      LOC = V5      ; # B7GPIOn_1
Net JX1_DIFF_N_4      LOC = W4      ; # B6GPIOn_2
Net JX1_DIFF_N_5      LOC = T8      ; # B7GPIOn_2
Net JX1_DIFF_N_6      LOC = W2      ; # B6GPIOn_3
Net JX1_DIFF_N_7      LOC = R6      ; # B7GPIOn_3
Net JX1_DIFF_N_8      LOC = U4      ; # B6GPIOn_4
Net JX1_DIFF_N_9      LOC = P6      ; # B7GPIOn_4
Net JX1_DIFF_N_10     LOC = U2      ; # B6GPIOn_5
Net JX1_DIFF_N_11     LOC = M6      ; # B7GPIOn_5
Net JX1_DIFF_N_12     LOC = P4      ; # B6GPIOn_6
Net JX1_DIFF_N_13     LOC = N3      ; # B7GPIOn_6
Net JX1_DIFF_N_14     LOC = N1      ; # B6GPIOn_7
Net JX1_DIFF_N_15     LOC = L5      ; # B7GPIOn_7
Net JX1_DIFF_N_16     LOC = K1      ; # B6GPIOn_9
Net JX1_DIFF_N_17     LOC = J2      ; # B7GPIOn_8
Net JX1_DIFF_N_18     LOC = K3      ; # B6GPIOn_10
Net JX1_DIFF_N_19     LOC = H3      ; # B7GPIOn_9
Net JX1_DIFF_N_20     LOC = G1      ; # B6GPIOn_11
Net JX1_DIFF_N_21     LOC = E3      ; # B7GPIOn_10

Net JX1_DIFF_P_0      LOC = AB3     ; # B6GPIOp_0
Net JX1_DIFF_P_1      LOC = W6      ; # B7GPIOp_0
Net JX1_DIFF_P_2      LOC = AC1     ; # B6GPIOp_1
Net JX1_DIFF_P_3      LOC = V4      ; # B7GPIOp_1
Net JX1_DIFF_P_4      LOC = W3      ; # B6GPIOp_2
Net JX1_DIFF_P_5      LOC = T7      ; # B7GPIOp_2
Net JX1_DIFF_P_6      LOC = W1      ; # B6GPIOp_3
Net JX1_DIFF_P_7      LOC = R5      ; # B7GPIOp_3
Net JX1_DIFF_P_8      LOC = U3      ; # B6GPIOp_4
Net JX1_DIFF_P_9      LOC = P5      ; # B7GPIOp_4
Net JX1_DIFF_P_10     LOC = U1      ; # B6GPIOp_5
Net JX1_DIFF_P_11     LOC = M5      ; # B7GPIOp_5
Net JX1_DIFF_P_12     LOC = P3      ; # B6GPIOp_6
Net JX1_DIFF_P_13     LOC = N4      ; # B7GPIOp_6
Net JX1_DIFF_P_14     LOC = N2      ; # B6GPIOp_7
Net JX1_DIFF_P_15     LOC = L6      ; # B7GPIOp_7
Net JX1_DIFF_P_16     LOC = K2      ; # B6GPIOp_9
Net JX1_DIFF_P_17     LOC = J3      ; # B7GPIOp_8

```

```

Net JX1_DIFF_P_18      LOC = K4   ; # B6GPIOp_10
Net JX1_DIFF_P_19      LOC = H4   ; # B7GPIOp_9
Net JX1_DIFF_P_20      LOC = G2   ; # B6GPIOp_11
Net JX1_DIFF_P_21      LOC = E4   ; # B7GPIOp_10

Net JX1_DIFF_CLK_P_IN  LOC = D2   ; # B6GPIOp_12
Net JX1_DIFF_CLK_N_IN  LOC = D1   ; # B6GPIOn_12

Net JX1_DIFF_CLK_P_OUT LOC = M2   ; # B6GPIOp_8
Net JX1_DIFF_CLK_N_OUT LOC = M1   ; # B6GPIOn_8

#=====
# EXP I/O - Connector JX2
#=====

# IOSTANDARD depends on JP5 jumper selection

Net JX2_SE_IO_0        LOC = M19  ; # B2GIO_0
Net JX2_SE_IO_1        LOC = P20  ; # B3GIO_0
Net JX2_SE_IO_2        LOC = M20  ; # B2GIO_1
Net JX2_SE_IO_3        LOC = T20  ; # B3GIO_1
Net JX2_SE_IO_4        LOC = K20  ; # B2GIO_2
Net JX2_SE_IO_5        LOC = P21  ; # B3GIO_2
Net JX2_SE_IO_6        LOC = J20  ; # B2GIO_3
Net JX2_SE_IO_7        LOC = R21  ; # B3GIO_3
Net JX2_SE_IO_8        LOC = H20  ; # B2GIO_4
Net JX2_SE_IO_9        LOC = P24  ; # B3GIO_4
Net JX2_SE_IO_10       LOC = J21  ; # B2GIO_5
Net JX2_SE_IO_11       LOC = P22  ; # B3GIO_5
Net JX2_SE_IO_12       LOC = H21  ; # B2GIO_6
Net JX2_SE_IO_13       LOC = R24  ; # B3GIO_6
Net JX2_SE_IO_14       LOC = H22  ; # B2GIO_7
Net JX2_SE_IO_15       LOC = R22  ; # B3GIO_7
Net JX2_SE_IO_16       LOC = J22  ; # B2GIO_8
Net JX2_SE_IO_17       LOC = T23  ; # B3GIO_8
Net JX2_SE_IO_18       LOC = J23  ; # B2GIO_9
Net JX2_SE_IO_19       LOC = T22  ; # B3GIO_9
Net JX2_SE_IO_20       LOC = L23  ; # B2GIO_10
Net JX2_SE_IO_21       LOC = U22  ; # B3GIO_10
Net JX2_SE_IO_22       LOC = M24  ; # B2GIO_11
Net JX2_SE_IO_23       LOC = T21  ; # B3GIO_11
Net JX2_SE_IO_24       LOC = T19  ; # B2GIO_12
Net JX2_SE_IO_25       LOC = V23  ; # B3GIO_12
Net JX2_SE_IO_26       LOC = N20  ; # B2GIO_13
Net JX2_SE_IO_27       LOC = V22  ; # B3GIO_13
Net JX2_SE_IO_28       LOC = W22  ; # B3GIO_14
Net JX2_SE_IO_29       LOC = U20  ; # B3GIO_16
Net JX2_SE_IO_30       LOC = AC26  ; # B2GIO_14
Net JX2_SE_IO_31       LOC = K21  ; # B2GIO_15
Net JX2_SE_IO_32       LOC = U21  ; # B3GIO_17
Net JX2_SE_IO_33       LOC = V20  ; # B3GIO_18

Net JX2_SE_CLK_IN      LOC = AE14  ; # B4_GCLK1
Net JX2_SE_CLK_OUT     LOC = V21  ; # B3GIO_15

Net JX2_DIFF_N_0       LOC = AB24  ; # B2GPIOn_0
Net JX2_DIFF_N_1       LOC = W26  ; # B3GPIOn_0
Net JX2_DIFF_N_2       LOC = Y26  ; # B2GPIOn_1
Net JX2_DIFF_N_3       LOC = U24  ; # B3GPIOn_1
Net JX2_DIFF_N_4       LOC = W24  ; # B2GPIOn_2
Net JX2_DIFF_N_5       LOC = N23  ; # B3GPIOn_2
Net JX2_DIFF_N_6       LOC = V25  ; # B2GPIOn_3
Net JX2_DIFF_N_7       LOC = N25  ; # B3GPIOn_3
Net JX2_DIFF_N_8       LOC = U26  ; # B2GPIOn_4
Net JX2_DIFF_N_9       LOC = M25  ; # B3GPIOn_4

```

```

Net JX2_DIFF_N_10      LOC = T26 ; # B2GPIO_n_5
Net JX2_DIFF_N_11      LOC = L25 ; # B3GPIO_n_5
Net JX2_DIFF_N_12      LOC = R26 ; # B2GPIO_n_6
Net JX2_DIFF_N_13      LOC = K25 ; # B3GPIO_n_6
Net JX2_DIFF_N_14      LOC = P26 ; # B2GPIO_n_7
Net JX2_DIFF_N_15      LOC = J24 ; # B3GPIO_n_7
Net JX2_DIFF_N_16      LOC = L19 ; # B2GPIO_n_9
Net JX2_DIFF_N_17      LOC = H25 ; # B3GPIO_n_8
Net JX2_DIFF_N_18      LOC = L21 ; # B2GPIO_n_10
Net JX2_DIFF_N_19      LOC = E23 ; # B3GPIO_n_9
Net JX2_DIFF_N_20      LOC = K23 ; # B2GPIO_n_11
Net JX2_DIFF_N_21      LOC = D25 ; # B3GPIO_n_10

Net JX2_DIFF_P_0       LOC = AB23 ; # B2GPIO_p_0
Net JX2_DIFF_P_1       LOC = W25 ; # B3GPIO_p_0
Net JX2_DIFF_P_2       LOC = Y25 ; # B2GPIO_p_1
Net JX2_DIFF_P_3       LOC = U23 ; # B3GPIO_p_1
Net JX2_DIFF_P_4       LOC = W23 ; # B2GPIO_p_2
Net JX2_DIFF_P_5       LOC = N24 ; # B3GPIO_p_2
Net JX2_DIFF_P_6       LOC = V24 ; # B2GPIO_p_3
Net JX2_DIFF_P_7       LOC = N26 ; # B3GPIO_p_3
Net JX2_DIFF_P_8       LOC = U25 ; # B2GPIO_p_4
Net JX2_DIFF_P_9       LOC = M26 ; # B3GPIO_p_4
Net JX2_DIFF_P_10      LOC = T25 ; # B2GPIO_p_5
Net JX2_DIFF_P_11      LOC = L26 ; # B3GPIO_p_5
Net JX2_DIFF_P_12      LOC = R25 ; # B2GPIO_p_6
Net JX2_DIFF_P_13      LOC = K26 ; # B3GPIO_p_6
Net JX2_DIFF_P_14      LOC = P25 ; # B2GPIO_p_7
Net JX2_DIFF_P_15      LOC = J25 ; # B3GPIO_p_7
Net JX2_DIFF_P_16      LOC = L20 ; # B2GPIO_p_9
Net JX2_DIFF_P_17      LOC = H26 ; # B3GPIO_p_8
Net JX2_DIFF_P_18      LOC = L22 ; # B2GPIO_p_10
Net JX2_DIFF_P_19      LOC = E24 ; # B3GPIO_p_9
Net JX2_DIFF_P_20      LOC = K24 ; # B2GPIO_p_11
Net JX2_DIFF_P_21      LOC = D26 ; # B3GPIO_p_10

Net JX2_DIFF_CLK_P_IN  LOC = H24 ; # B2GPIO_p_12
Net JX2_DIFF_CLK_N_IN  LOC = H23 ; # B2GPIO_n_12

Net JX2_DIFF_CLK_P_OUT LOC = N22 ; # B2GPIO_p_8
Net JX2_DIFF_CLK_N_OUT LOC = N21 ; # B2GPIO_n_8

#=====
# PCI Express Transceiver - U35
#=====

#
# SYS reset (input) signal. The sys_reset_n signal should be
# obtained from the PCI Express interface if possible. For
# slot based form factors, a system reset signal is usually
# present on the connector. For cable based form factors, a
# system reset signal may not be available. In this case, the
# system reset signal must be generated locally by some form of
# supervisory circuit. You may change the IOSTANDARD and LOC
# to suit your requirements and VCCO voltage banking rules.
#

NET "sys_reset_n"      LOC = "AE4" | IOSTANDARD = LVCMOS25 | IOBDELAY = NONE ;

#
# PIPE receive (input) signals. These inputs must be properly
# terminated SSTL2_I signals. Termination is required at both
# the driving device and the receiving device. When routing
# these signals on the board, they should be length matched to
# minimize skew. For FPGA devices that support DCI, you may

```

```
# elect to change the IOSTANDARD from SSTL2_I to SSTL2_I_DCI.
# Use of DCI requires proper setting of VRN and VRP reference
# resistors, but eliminates the need for external termination
# at the FPGA device. Note that the proper termination at the
# PHY device is still required, even when DCI is used. Please
# consult the core documentation and the FPGA device datasheet
# for additional information.
#

NET "rxclk" LOC = "AE13" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "phystatus" LOC = "AF12" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxvalid" LOC = "AD12" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxstatus<2>" LOC = "AC11" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxstatus<1>" LOC = "AD10" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxstatus<0>" LOC = "AC10" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<0>" LOC = "AF8" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<0>" LOC = "AE8" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<1>" LOC = "AC7" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<2>" LOC = "AF6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<3>" LOC = "AE6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<4>" LOC = "AD6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<5>" LOC = "AC6" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<6>" LOC = "AE5" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxdata<7>" LOC = "AD5" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;
NET "rxlecidle" LOC = "AF4" | IOSTANDARD = SSTL2_I | IOBDELAY = NONE ;

#
# PIPE transmit (output) signals. These outputs must be properly
# terminated SSTL2_I signals. Termination is required at both
# the driving device and the receiving device. When routing
# these signals on the board, they should be length matched to
# minimize skew. For FPGA devices that support DCI, you may
# elect to change the IOSTANDARD from SSTL2_I to SSTL2_I_DCI.
# Use of DCI requires proper setting of VRN and VRP reference
# resistors, but eliminates the need for external termination
# at the FPGA device. Note that the proper termination at the
# PHY device is still required, even when DCI is used. Please
# consult the core documentation and the FPGA device datasheet
# for additional information.
#

NET "resetn" LOC = "AF24" | IOSTANDARD = SSTL2_I ;
NET "rxpolarity" LOC = "AE24" | IOSTANDARD = SSTL2_I ;
NET "txlecidle" LOC = "AF23" | IOSTANDARD = SSTL2_I ;
NET "txcompliance" LOC = "AE23" | IOSTANDARD = SSTL2_I ;
NET "powerdown<1>" LOC = "AD23" | IOSTANDARD = SSTL2_I ;
NET "powerdown<0>" LOC = "AF22" | IOSTANDARD = SSTL2_I ;
NET "txdata<0>" LOC = "AE22" | IOSTANDARD = SSTL2_I ;
NET "txdetectrx_loopback" LOC = "AF21" | IOSTANDARD = SSTL2_I ;
NET "txclk" LOC = "AE21" | IOSTANDARD = SSTL2_I ;
NET "txdata<7>" LOC = "AD21" | IOSTANDARD = SSTL2_I ;
NET "txdata<6>" LOC = "AF20" | IOSTANDARD = SSTL2_I ;
NET "txdata<5>" LOC = "AE20" | IOSTANDARD = SSTL2_I ;
NET "txdata<4>" LOC = "AF19" | IOSTANDARD = SSTL2_I ;
NET "txdata<3>" LOC = "AE19" | IOSTANDARD = SSTL2_I ;
NET "txdata<2>" LOC = "AF15" | IOSTANDARD = SSTL2_I ;
NET "txdata<1>" LOC = "AE15" | IOSTANDARD = SSTL2_I ;
NET "txdata<0>" LOC = "AD15" | IOSTANDARD = SSTL2_I ;

#####
#
# Timing Constraints
#####
#
# Ignore timing on asynchronous signals.
```

```
#

NET "sys_reset_n" TIG ;

#
# Timing requirements and related constraints.
#

NET "rxclk" TNM_NET = "TNM_NET_RXCLK" ;
TIMESPEC "TS_RXCLK" = PERIOD "TNM_NET_RXCLK" 250 MHz HIGH 50 % PRIORITY 0 ;
PIN "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/kh2_bufg.I0" TIG ;

PIN "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/kh2_bufg.S" TPSYNC =
"SYNC_SELECT" ;
TIMESPEC "TS_SYNC_SELECT" = FROM FFS(*) TO "SYNC_SELECT" 4 ns ;

INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/*" AREA_GROUP = "AG_PLM" ;
AREA_GROUP "AG_PLM" RANGE = SLICE_X2Y0:SLICE_X77Y31 ;

INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/kh2_bufg" LOC =
"BUFGMUX3" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/mgt_bufg" LOC =
"BUFGMUX1" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/phy_bufg" LOC =
"BUFGMUX0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/dcm" LOC =
"DCM_X0Y0" ;

INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/reg_pipe_ckel"
RANGE = "SLICE_X44Y0:SLICE_X48Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata0"
RANGE = "SLICE_X44Y0:SLICE_X48Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata1"
RANGE = "SLICE_X45Y0:SLICE_X49Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata2"
RANGE = "SLICE_X45Y0:SLICE_X49Y0" ;

INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/reg_pipe_ckem"
RANGE = "SLICE_X60Y0:SLICE_X64Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata3"
RANGE = "SLICE_X60Y0:SLICE_X64Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata4"
RANGE = "SLICE_X60Y0:SLICE_X64Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata5"
RANGE = "SLICE_X61Y0:SLICE_X65Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata6"
RANGE = "SLICE_X61Y0:SLICE_X65Y0" ;

INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/reg_pipe_cker"
RANGE = "SLICE_X5Y0:SLICE_X90Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata7"
RANGE = "SLICE_X5Y0:SLICE_X90Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxdata8"
RANGE = "SLICE_X5Y0:SLICE_X91Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/nstxcompl"
RANGE = "SLICE_X5Y0:SLICE_X91Y0" ;

INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/reg_dcm_rst" LOC =
"SLICE_X38Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/_n00041" LOC =
"SLICE_X38Y0" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/reg_dcm_rst"
XBLKNM = "reg_dcm_rst_blk" ;
INST "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/_n00041"
XBLKNM = "reg_dcm_rst_blk" ;
```

```

NET "pcie_pipe/BU2/U0/pci_exp_1_lane_epipe_ep0/plm/kh2_mgt/reg_dcm_rst"
ROUTE="{3;1;3s1000fg676;38070486!-1;-560;-81464;S!0;-240;-1191!0;1881;" "-
520!1;3280;-638!1;-11216;-600!2;-2048;408!3;-2200;-1155;L!4;-20968;206!"
"5;167;0;L!7;-21656;0!9;-11613;-791!10;1390;-64!11;559;0;L!}";

#
# Timing requirements; input and output register
# packing into IOB. The following constraints are
# bogus constraints, used with the intent that they
# fail should the IOB flops not be properly packed.
# If you encounter timing failures, check that you
# are using map with the correct option, "-pr b".
# You should manually verify in the .mrp report
# file that the rxclk input register is properly
# packed into the IOB. It is illegal to specify
# an OFFSET constraint on a clock input with respect
# to itself, so no constraint is present for rxclk.
# The OFFSET constraint for txclk is larger than the
# other outputs to account for the fact that it is
# clocked by both edges of rxclk; the timing tools
# report this as an extra half-cycle worth of delay.
#

NET "phystatus"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxvalid"            OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxstatus<2>"        OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxstatus<1>"        OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxstatus<0>"        OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<0>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<0>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<1>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<2>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<3>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<4>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<5>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<6>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxdata<7>"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;
NET "rxlecidle"          OFFSET = IN 1.5 ns VALID 3.0 ns BEFORE "rxclk" ;

NET "resetn"             OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "rxpolarity"         OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txlecidle"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txcompliance"       OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "powerdown<1>"       OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "powerdown<0>"       OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<0>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdetectrx_loopback" OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<7>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<6>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<5>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<4>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<3>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<2>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<1>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txdata<0>"          OFFSET = OUT 4.5 ns AFTER "rxclk" ;
NET "txclk"              OFFSET = OUT 6.5 ns AFTER "rxclk" ;

#####
#
# Timing Budget Information, PX1011A-EL to XC3S1000
#####
#
# Timing info; output data valid window, PX1011A-EL.
# All output signals from PX1011A-EL are synchronous
# to rxclk. The rxclk is a source synchronous clock

```

```
# that is center aligned with the data. This creates
# a data valid window of 3.0 ns, with the data valid
# 1.5 ns before the rising edge of rxclk and 1.5 ns
# after the rising edge. This timing information is
# from the device datasheet and is assumed to account
# for all skew associated with the device.
#
#
# Timing info; input data setup/hold window, XC3S1000.
# All input signals to the XC3S1000 are synchronous
# to rxclk. The rxclk is a source synchronous clock
# that is center aligned with the data. This design
# uses an implementation of active phase alignment from
# Xilinx Application Note 268. The required input data
# setup/hold window at the device is based on parameters:
#
# * Tsamp, 800 ps
# This parameter indicates the total sampling error
# at the input registers across voltage, temperature,
# and process. For a single data rate interface, this
# number includes the DCM jitter, the DCM phase shift
# resolution, and DCM phase offset. This value is an
# estimate based on source synchronous characterization
# data.
#
# * package skew, +/- 150 ps
# This is the worst case package skew between pins
# used for rxclk and the inputs. This value is a
# conservative estimate based on published package
# skew data for another device family.
#
# * clock skew, +/- 250 ps
# This is the worst case clock skew between pins
# used for rxclk and the inputs. This value is
# measured for the clk2x signal using the delay
# function in FPGA Editor for this specific device
# and pinout.
#
# The input data valid window computed as the result:
# Rx = [800 + 150 + 250] = 1200 ps
#
#
# Timing budget, from PX1011A-EL to XC3S1000. The
# remaining slack is sufficient to cover skew in the
# signal routing, inter-symbol interference, and other
# analog considerations. Xilinx is not responsible
# for board-related failures and therefore strongly
# recommends all users perform simulations.
#
#####
#
# Timing Budget Information, XC3S1000 to PX1011A-EL
#####
#
# Timing info; output data valid window, XC3S1000.
# All output signals from XC3S1000 are synchronous
# to txclk. The txclk is a source synchronous clock
# that is center aligned with the data. The internal
# clock signals are generated from a 250 MHz reference
# (rxclk) that enters the DCM using the divide by two
# option. Three DCM outputs are used; clk2x is 250 MHz,
# clk0 is 125 MHz, and clkdv is 62.5 MHz. The clk0 is
```

```
# used as the DCM feedback clock. The clk2x signal is
# used to forward txclk via a DDR output register. The
# txclk yields an ideal data valid window of 4.0 ns,
# with the data valid 2.0 ns before the rising edge
# and 2.0 ns after the rising edge. This ideal window
# must then be adjusted for sources of skew:
#
# * DCM output jitter, +/- 200 ps
# This output jitter figure is for the 2x output,
# obtained from the device datasheet. Jitter is
# the only number in the timing budget defined
# with absolute magnitude; in this case, the jitter
# magnitude used in calculations is twice the value
# shown above.
#
# * duty cycle distortion, +/- 400 ps
# In this design, clk2x is used to forward txclk via
# a DDR output register using local clock inversion.
# The negative edge of clk2x causes the rising edge
# on txclk. Therefore, duty cycle distortion is an
# important consideration. This value is an estimate
# based on source synchronous characterization data.
#
# * package skew, +/- 150 ps
# This is the worst case package skew between pins
# used for txclk and the outputs. This value is a
# conservative estimate based on published package
# skew data for another device family.
#
# * clock skew, +/- 250 ps
# This is the worst case clock skew between pins
# used for txclk and the outputs. This value is
# measured for the clk2x signal using the delay
# function in FPGA Editor for this specific device
# and pinout.
#
# * DCM phase offset, +/- 400 ps
# This figure is the sum of the clkout_phase and
# clkln_clkfb_phase parameters. The clkout_phase
# specifies the phase relationship between the DCM
# outputs clk2x and clk0. The clkln_clkfb_phase
# defines the amount of phase offset between the
# clock input and the feedback input of the DCM.
#
# The output data valid window computed as the result:
# Tx = 4000 - [400 + 400 + 150 + 250 + 400] = 2400 ps
#
#
# Timing info; input data setup/hold window, PX1011A-EL.
# All input signals to the PX1011A-EL are synchronous
# to txclk. The txclk is a source synchronous clock
# that is center aligned with the data. The required
# input data setup/hold window at the device is 1.0 ns,
# with the data setup 0.5 ns before the rising edge of
# txclk and data hold 0.5 ns after the rising edge.
# This timing information is from the device datasheet
# and is assumed to account for all skew associated
# with the device.
#
#
# Timing budget, from XC3S1000 to PX1011A-EL. The
# remaining slack is sufficient to cover skew in the
# signal routing, inter-symbol interference, and other
```

```
# analog considerations.  Xilinx is not responsible
# for board-related failures and therefore strongly
# recommends all users perform simulations.
#
```

```
#####
#
# End
#####
```


Schematics

Please scroll down to the next page to view the schematics.

Spartan-3 PCIe Starter Board	
Avnet Engineering Services	
www.em.avnet.com/xilinx	
Function	Sheet Number
Cover Sheet	1
Block Diagram	2
FPGA - Banks 0 & 1	3
FPGA - Banks 2 & 3	4
FPGA - LVDS Banks 4 & 5	5
FPGA - PCIe Banks 6 & 7	6
FPGA Power	7
FPGA Configuration	8
PCI Express	9
DDR SDRAM	10
Video DAC	11
Board Power	12
Revision Notes	13



Powered By:



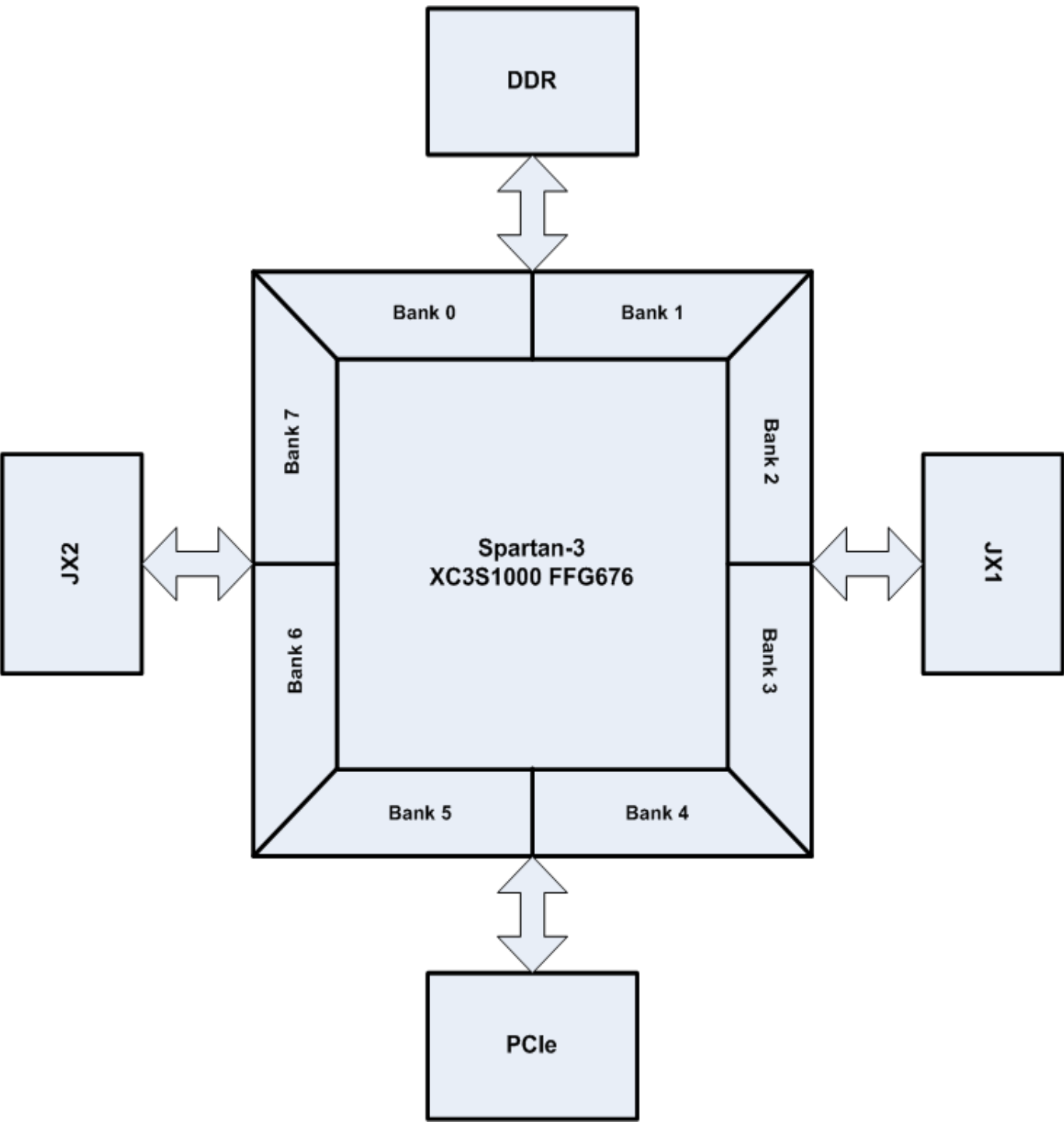
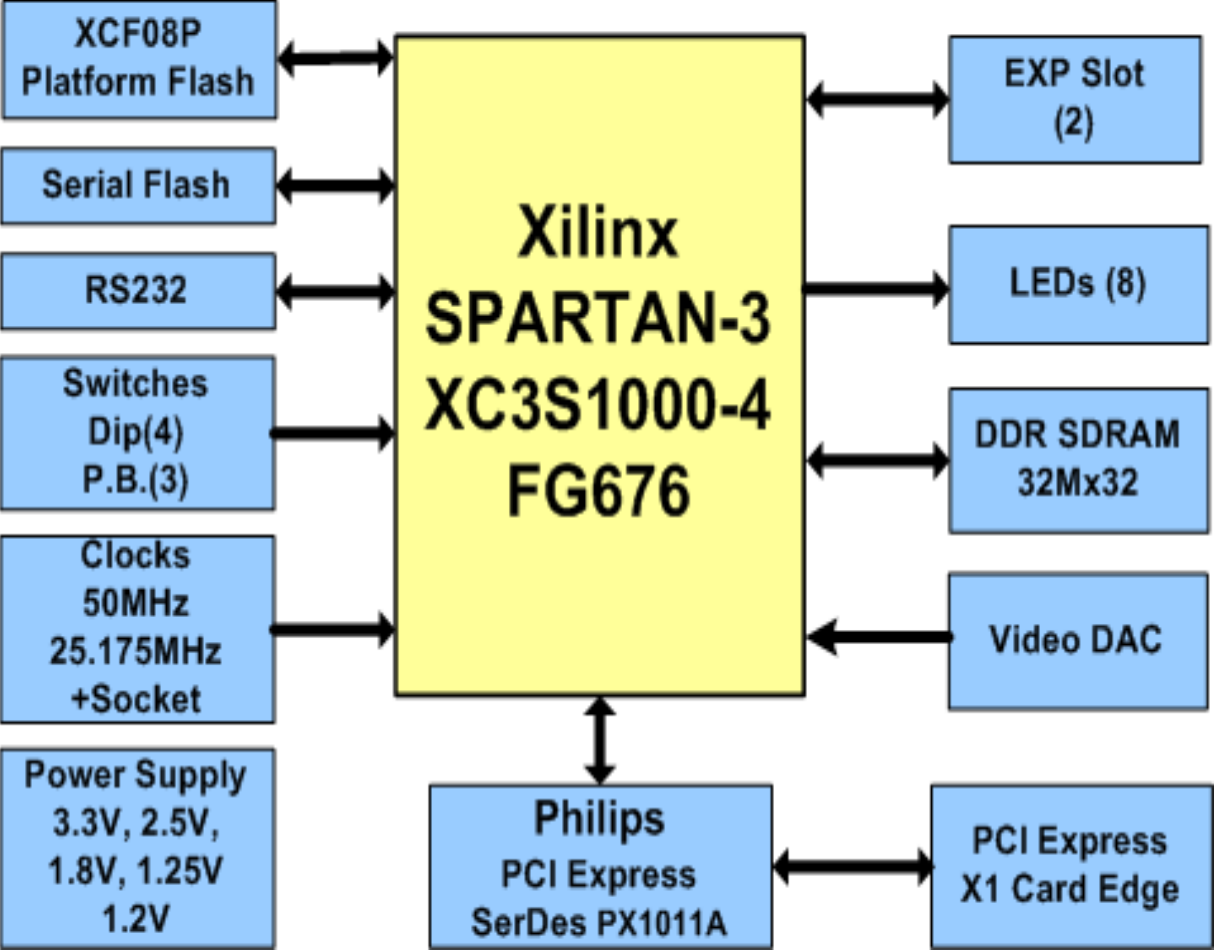
Copyright 2006, Avnet, Inc. All Rights Reserved.

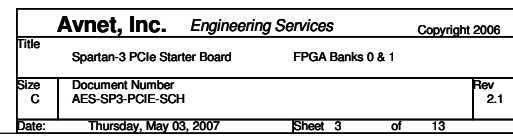
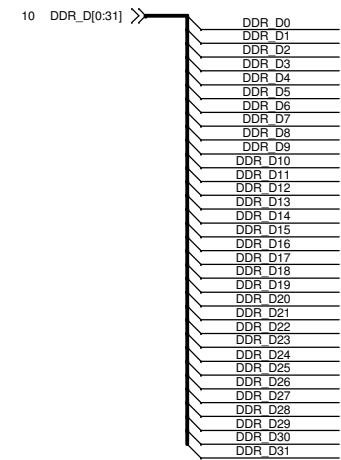
This material may not be reproduced, distributed, republished, displayed, posted, transmitted or copied in any form or by any means without the prior written permission of Avnet, Inc. AVNET and the AV logo are registered trademarks of Avnet, Inc. All trademarks and trade names are the properties of their respective owners and Avnet, Inc. disclaims any proprietary interest or right in trademarks, service marks and trade names other than its own.

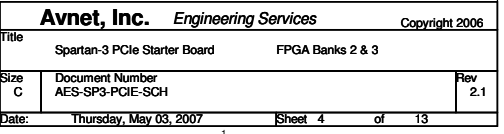
Avnet is not responsible for typographical or other errors or omissions or for direct, indirect, incidental or consequential damages related to this material or resulting from its use. Avnet makes no warranty or representation respecting this material, which is provided on an "AS IS" basis. AVNET HEREBY DISCLAIMS ALL WARRANTIES OR LIABILITY OF ANY KIND WITH RESPECT THERETO, INCLUDING, WITHOUT LIMITATION, REPRESENTATIONS REGARDING ACCURACY AND COMPLETENESS, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, SUITABILITY OR FITNESS FOR A PARTICULAR PURPOSE, TITLE AND/OR NON-INFRINGEMENT. This material is not designed, intended or authorized for use in medical, life support, life sustaining or nuclear applications or applications in which the failure of the product could result in personal injury, death or property damage. Any party using or selling products for use in any such applications do so at their sole risk and agree that Avnet is not liable, in whole or in part, for any claim or damage arising from such use, and agree to fully indemnify, defend and hold harmless Avnet from and against any and all claims, damages, loss, cost, expense or liability arising out of or in connection with the use or performance of products in such applications.

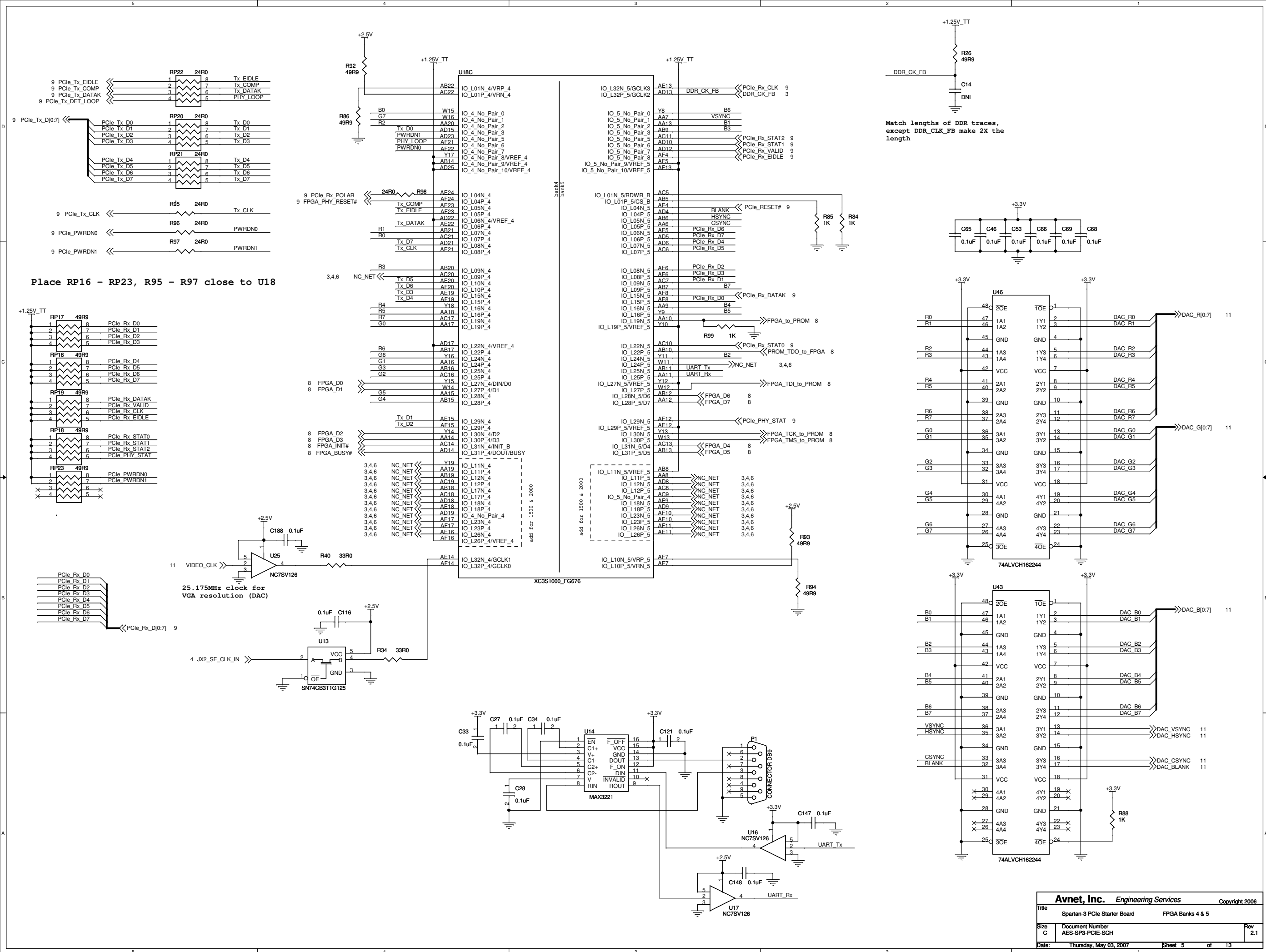
05/03/07

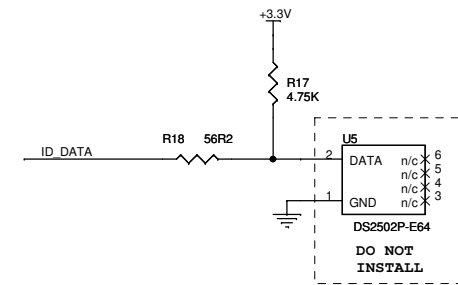
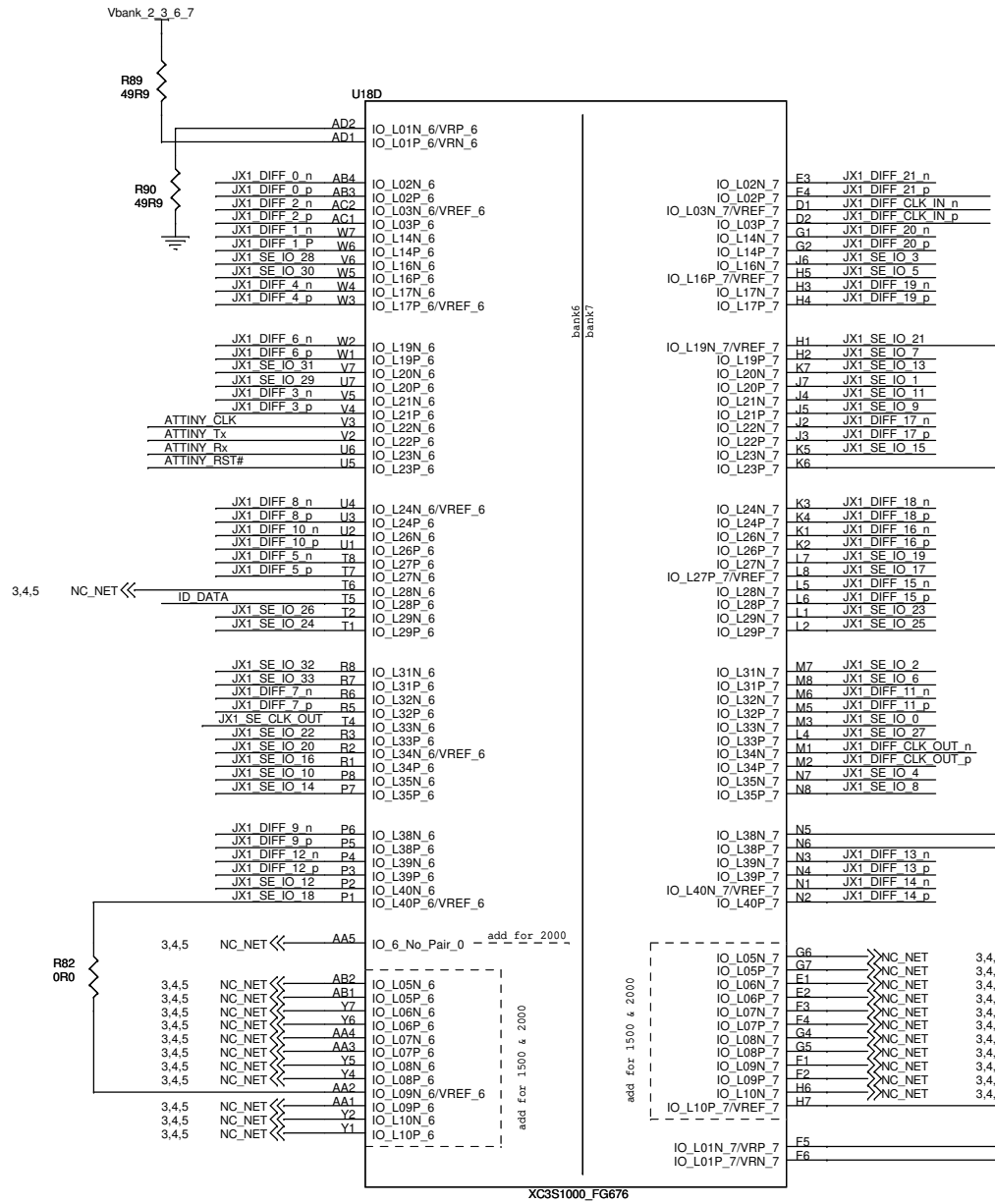
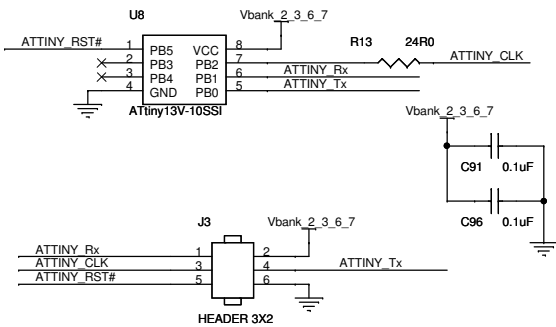
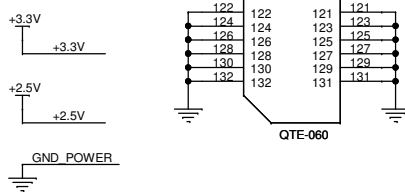
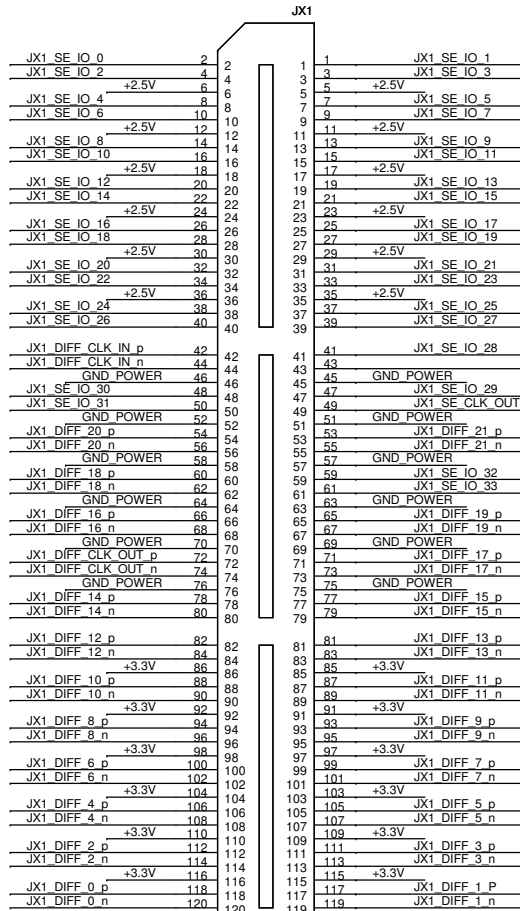
REVISION 2.1		
Avnet, Inc. Engineering Services Copyright 2006		
Title Spartan-3 PCIe Starter Board Cover sheet		
Size B	Document Number AES-SP3-PCIE-SCH	Rev 2.1
Date:	Thursday, May 03, 2007	Sheet 1 of 13

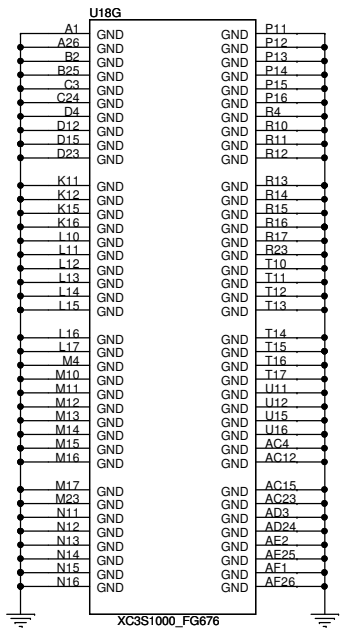
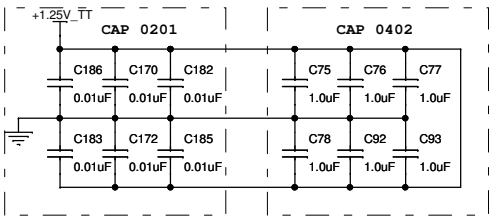
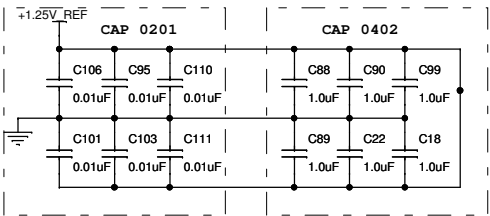
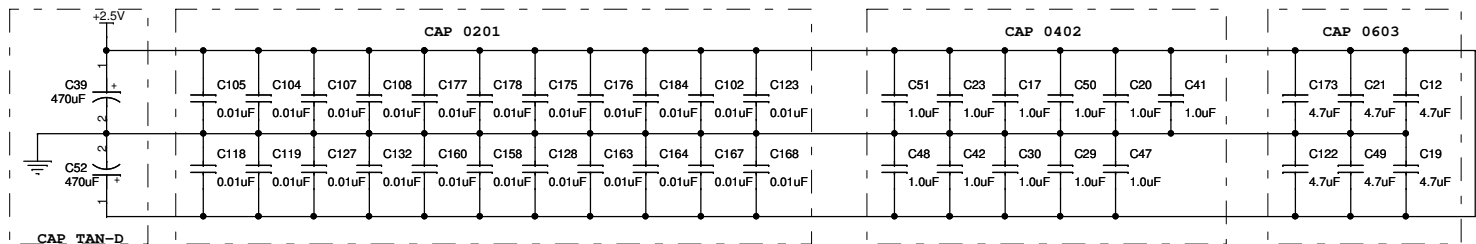
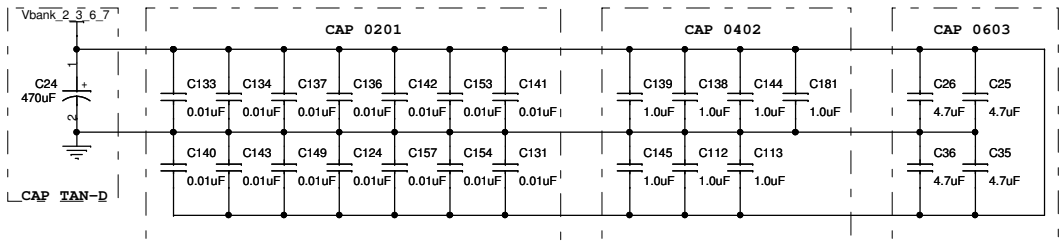
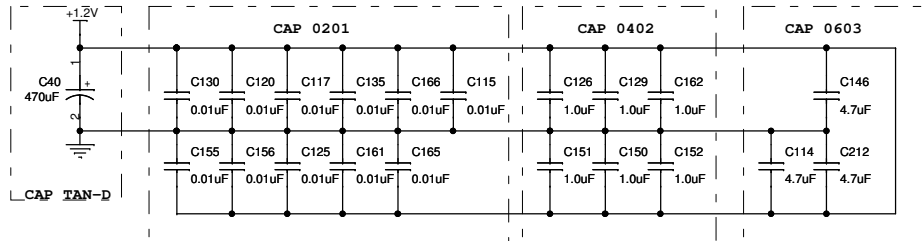
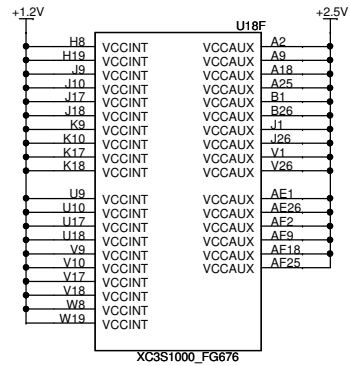
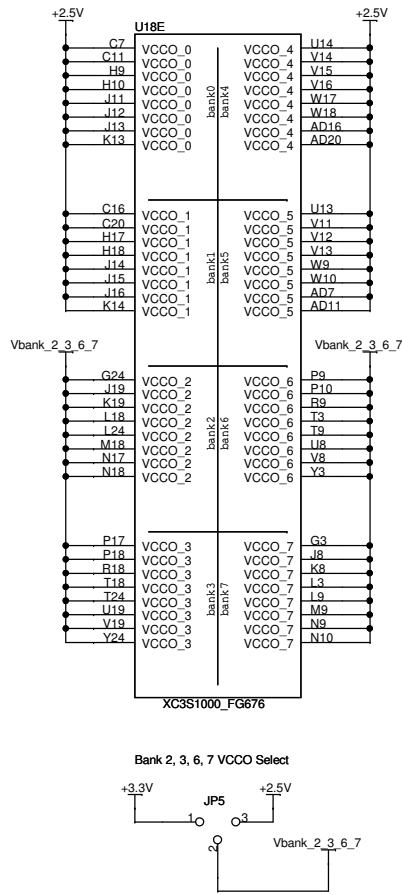


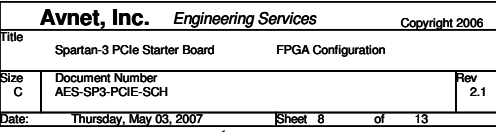
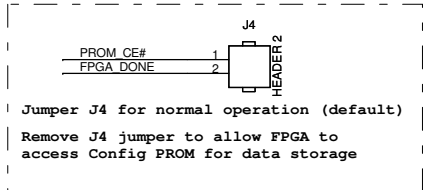
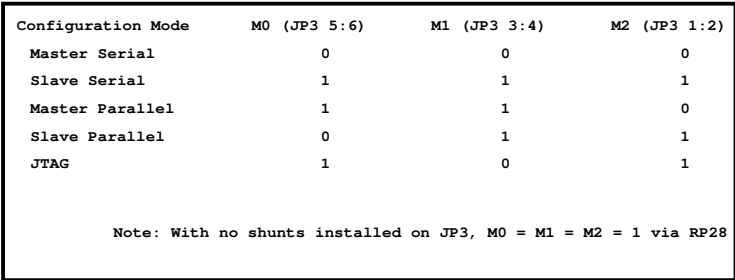
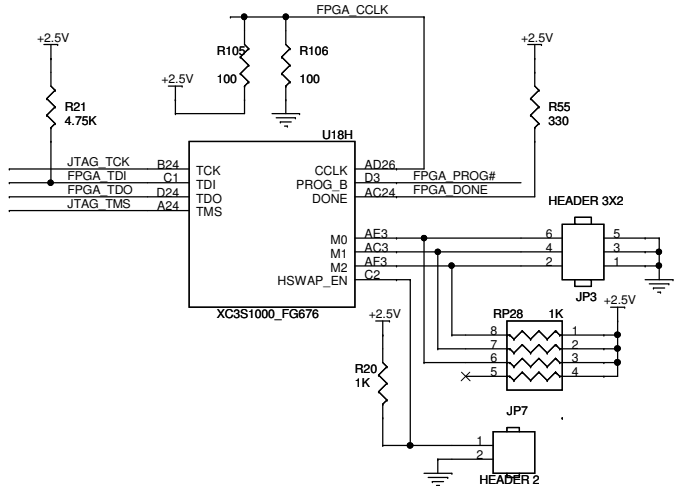


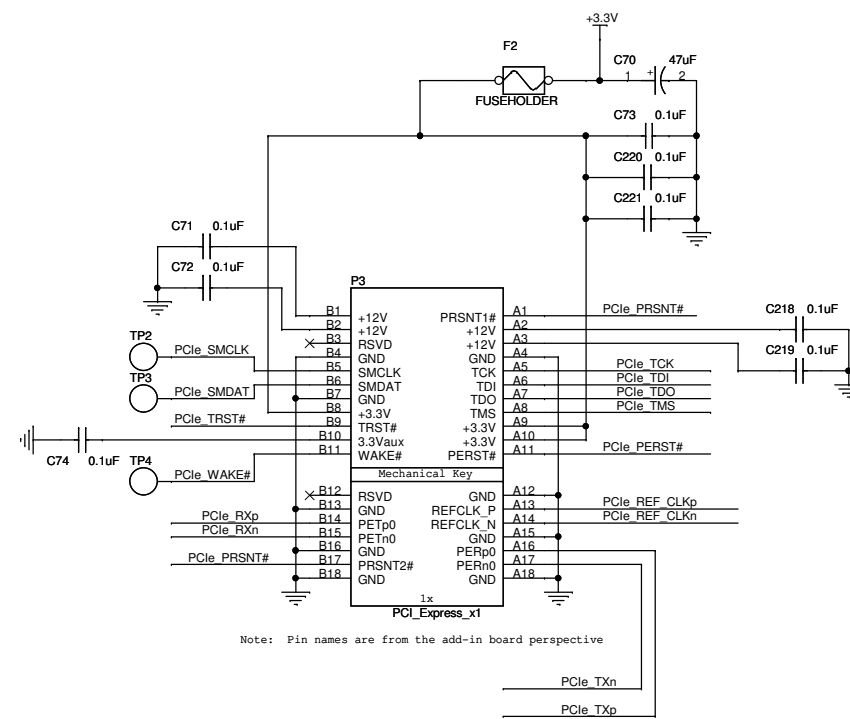
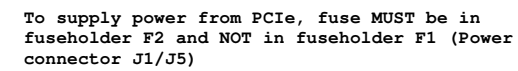
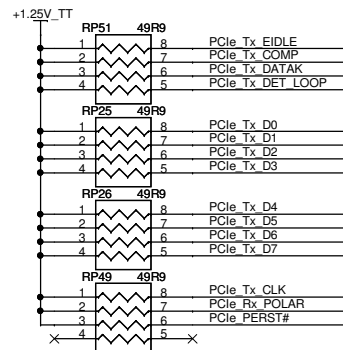
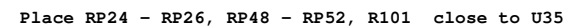




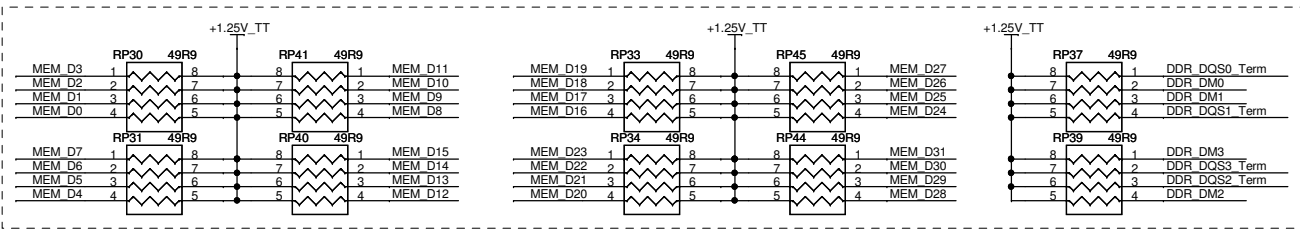




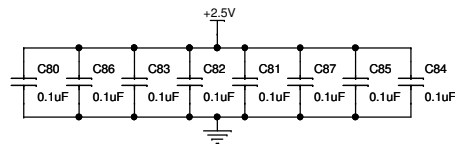
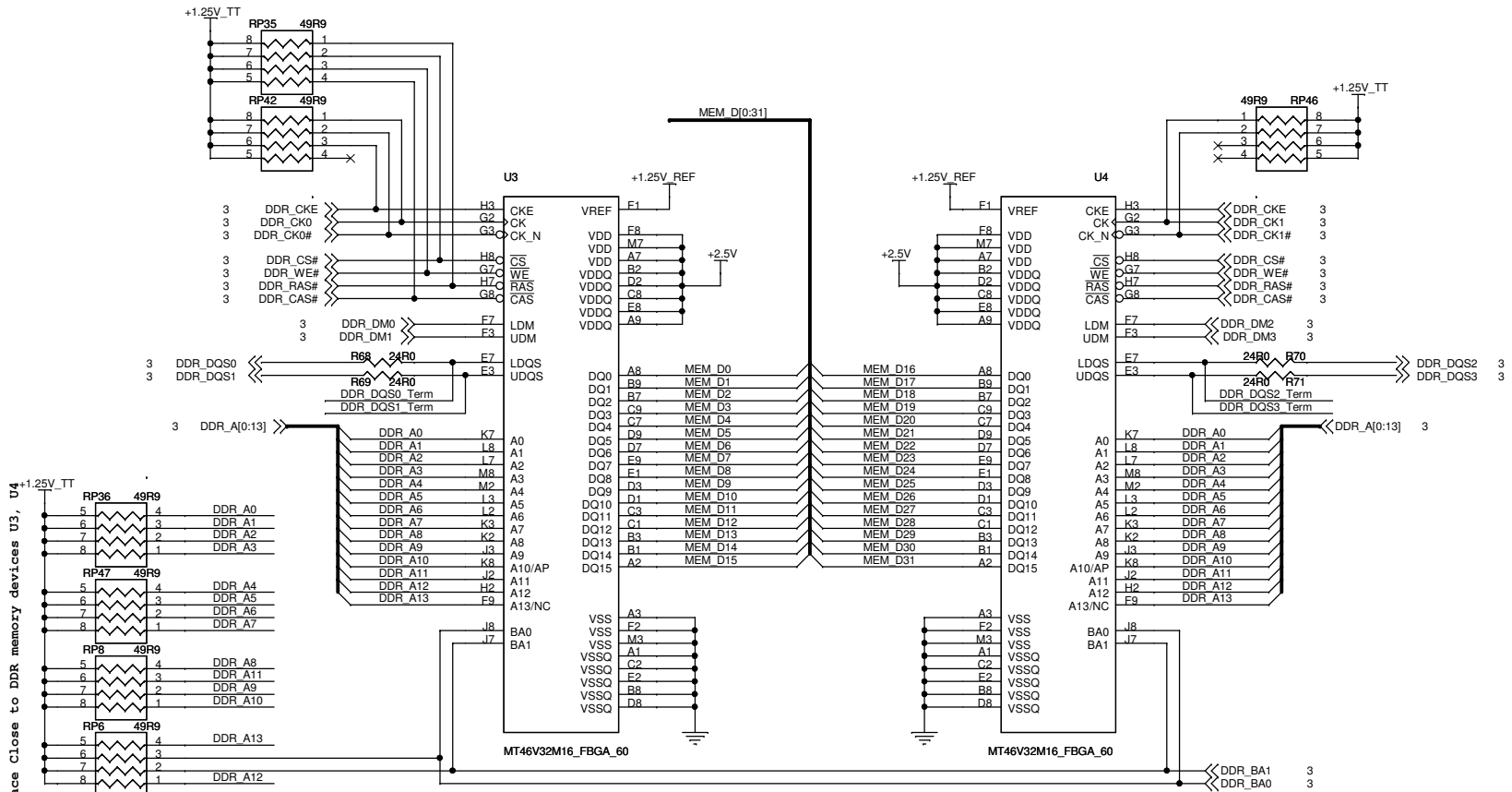
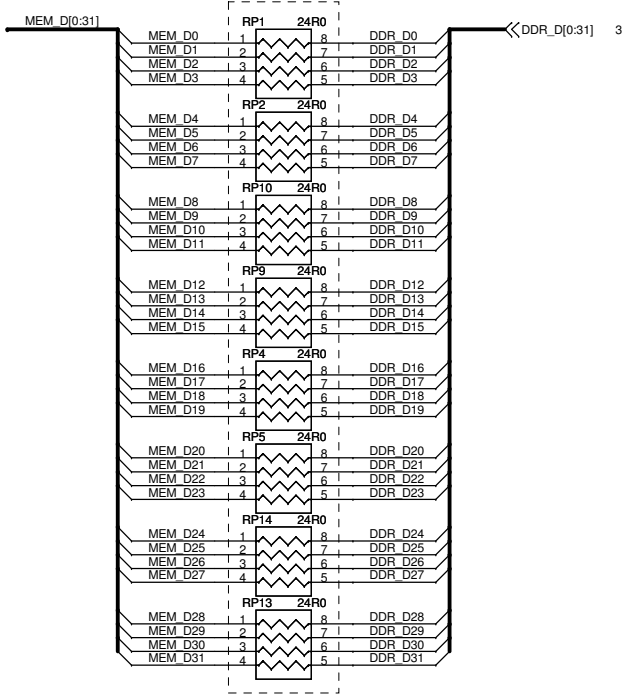




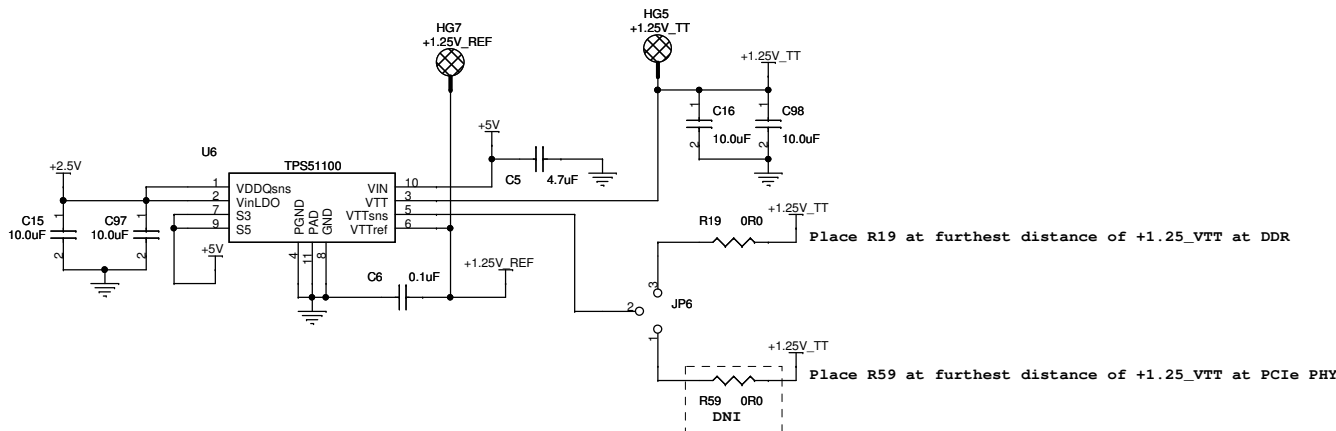
Install Terminators on Far Side of DDR memory devices U3 and U4

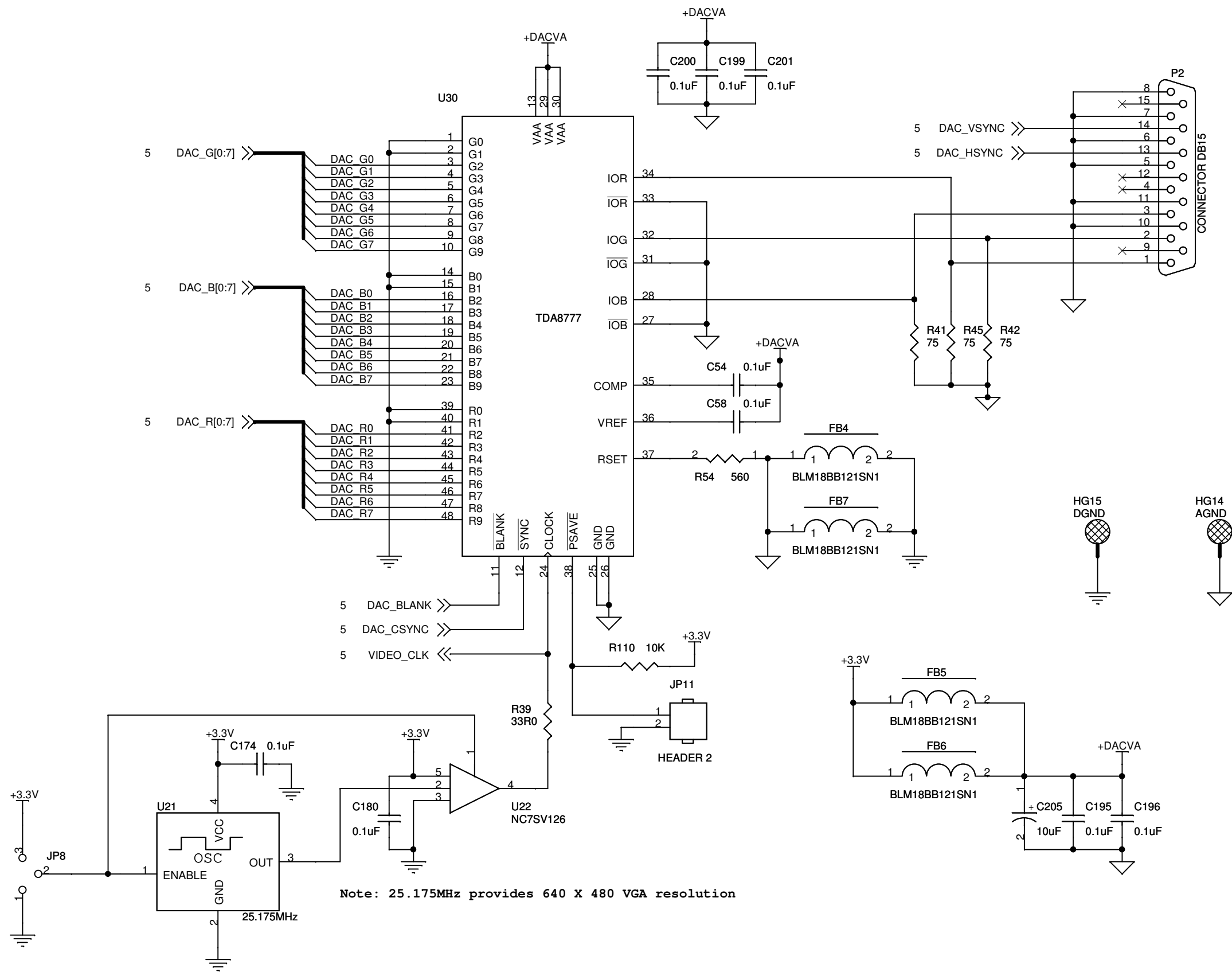


Place Close to DDR memory devices U3 and U4



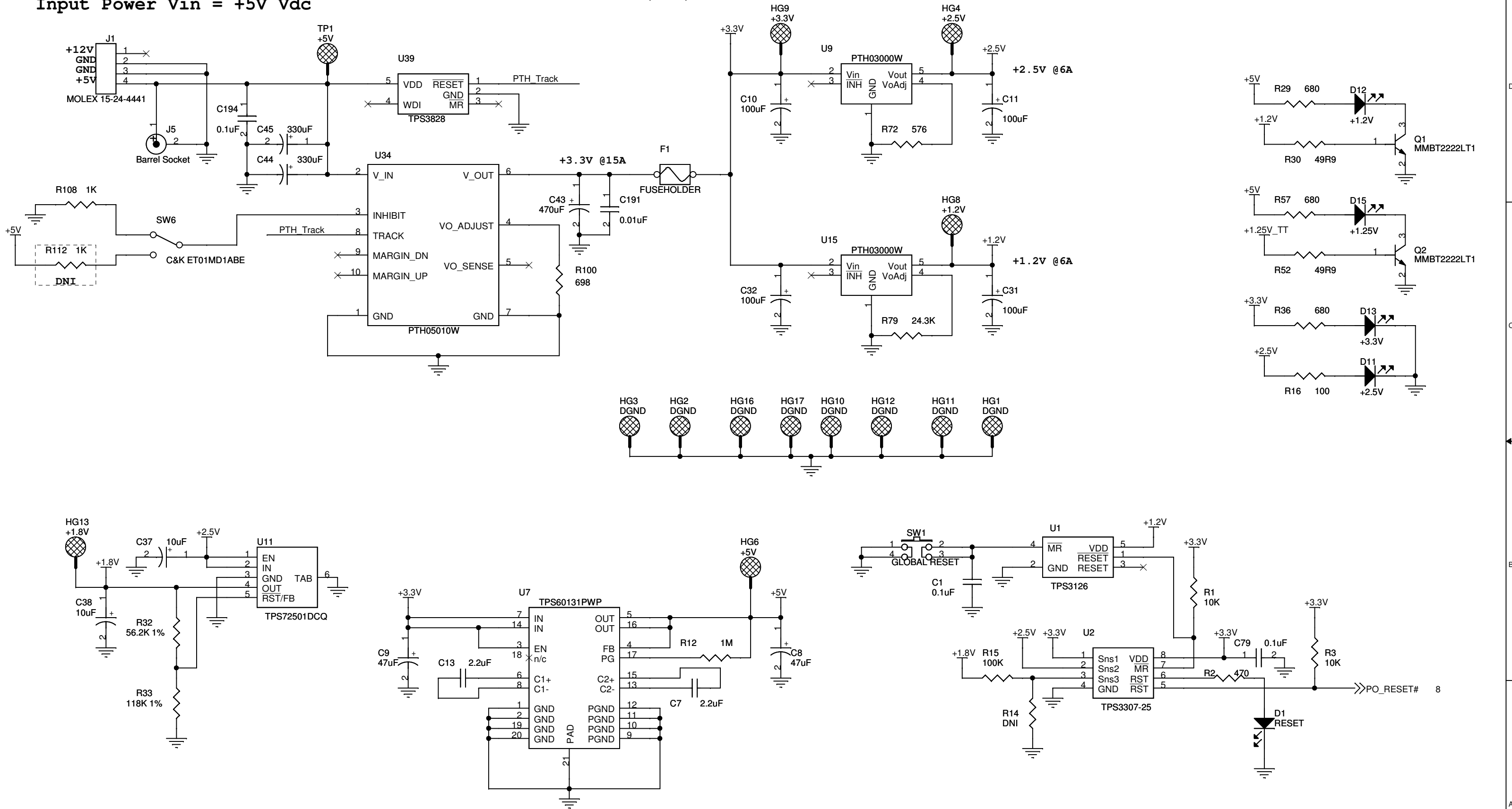
DDR Termination Voltage





To supply power from J1 or J5, fuse MUST be in fuseholder F1 and NOT in fuseholder F2 (PCIe)

Input Power Vin = +5V Vdc



Changes Rev1 to Rev2:

3/27/06 - Swapped power and ground connections to power switch SW6 to match PCB silk-screen (up = ON)

Clarification:

5/02/07 - Changed names of pin pairs B14/B15 and A16/A17 on PCIe connector P3 to match PCIe CEM Spec.

Avnet, Inc. <i>Engineering Services</i>			Copyright 2006
Title		Spartan-3 PCIe Starter Board Revision Notes	
Size A	Document Number AES-SP3-PCIE-SCH		Rev 2.1
Date:	Thursday, May 03, 2007	Sheet 13	of 13