

The *D**i**NI* Group

LOGIC Emulation Source

User Guide

DN9000K10PCIE4GL

LOGIC EMULATION SOURCE

DN9000K10PCIE4GL User Manual

Major Revision 1

Last Update November 26, 2008

© The Dini Group, 2006

1010 Pearl Street • Suite 6

La Jolla, CA92037 USA

Phone 858.454.3419 • Fax 858.454.1279

support@dinigroup.com

www.dinigroup.com

Table of Contents

CHAPTER 1: INTRODUCTION.....	1
1 MANUAL CONTENTS.....	1
INTRODUCTION	1
QUICK START GUIDE.....	1
CONTROLLER SOFTWARE.....	2
HARDWARE.....	2
THE REFERENCE DESIGN.....	2
ORDERING INFORMATION	2
2 CONVENTIONS.....	2
2.1 TYPOGRAPHICAL	2
2.2 MANUAL CONTENT	2
2.2.1 File names.....	2
2.2.2 Physical Dimensions	2
2.2.3 Part Pin Names.....	3
2.2.4 Schematic Clippings.....	3
2.3 TERMINOLOGY	3
3 RESOURCES.....	4
3.1 USER CD	4
3.2 DINIGROUP.COM.....	4
3.3 ERRATA AND CUSTOMER NOTIFICATIONS.....	5
3.4 SCHEMATICS AND NETLIST	5
3.4.1 Netlist.....	5
3.4.2 Net name conventions	5
3.5 DATASHEET LIBRARY	5
3.6 XILINX	6
3.7 DINI GROUP REFERENCE DESIGNS	6
3.8 BOARD MODELS.....	6
3.8.1 Base System Builder.....	6
4 EMAIL AND PHONE SUPPORT.....	6
CHAPTER 2: QUICK START GUIDE	9
1 PROVIDED MATERIALS	9
1.1 SYSTEM REQUIREMENTS	9
2 WARNINGS.....	10
2.1 ESD.....	10
2.2 OTHER.....	10
3 PRE-POWER ON INSTRUCTIONS	10
3.1 INSTALL MEMORY.....	11
3.2 PREPARE CONFIGURATION FILES	11
3.3 INSTALL DN9000K10PCIE4GL IN COMPUTER (OPTIONAL)	12
3.4 CABLES	13
3.4.1 Connect RS232 Cable	13
3.4.2 Connect USB Cable.....	13
3.4.3 Connect Power cable	13
4 POWER ON.....	14
4.1 VIEW CONFIGURATION FEEDBACK OVER RS232.....	14
4.2 CHECK LED STATUS LIGHTS	16
5 RUN USB CONTROLLER.....	16

INTRODUCTION

5.1	DRIVER INSTALLATION	17
5.2	OPERATING THE USB CONTROLLER PROGRAM	17
5.2.1	Configure an FPGA	18
5.2.2	Set Clock Frequencies.....	19
5.2.3	Check clock frequencies.....	19
5.2.4	Run Hardware Test (DDR2).....	19
5.3	GETTING DATA TO AND FROM THE FPGA.....	20
6	COMMUNICATING OVER THE SERIAL PORT.....	21
7	SCAN THE JTAG CHAIN	21
7.1	MOVING ON	22
CHAPTER 3: CONTROLLER SOFTWARE.....		23
1	USB CONTROLLER	24
1.1	MAIN WINDOW	24
1.1.1	Refresh Button.....	25
1.1.2	Disable/Enable USB.....	26
1.1.3	Log Window	26
1.1.4	Board Graphic	27
1.2	MENU OPTIONS	28
1.2.1	File Menu.....	28
1.2.2	Edit Menu.....	28
1.2.3	FPGA Configuration Menu	28
1.2.4	FPGA Reference Design	29
1.2.5	Main Bus.....	29
1.2.6	Settings/Info Menu	30
1.3	UNSUPPORTED FEATURES	32
1.3.1	Header Test.....	32
1.3.2	Programmer's Menu	32
1.4	LOG FILE	32
1.5	INI FILE.....	32
1.6	COMPILING USB CONTROLLER.....	32
2	AETEST USB.....	32
2.1	COMPILING AETEST_USB.....	33
2.1.1	Cygwin	33
3	ROLLING YOUR OWN SOFTWARE.....	33
3.1	DINI API.....	33
3.2	USB	33
4	UPDATING THE FIRMWARE.....	33
4.1	OBTAINING THE UPDATES.....	ERROR! BOOKMARK NOT DEFINED.
4.2	UPDATING THE EEPROM FIRMWARE	33
4.3	UPDATING THE MCU (FLASH) FIRMWARE.....	39
4.4	UPDATING THE SPARTAN (PROM) FIRMWARE	34
CHAPTER 4: HARDWARE		43
1	GENERAL OVERVIEW	43
1.1	MARKETING CRAP.....	43
1.1.1	Features	44
1.1.2	Description.....	45
2	VIRTEX 5.....	46
3	CONFIGURATION SECTION	46
3.1	CONFIGURATION SECTION FEEDBACK.....	47
3.2	FPGA CONFIGURATION	48
3.3	USB INTERFACES	49
3.4	COMPACTFLASH INTERFACE	49

INTRODUCTION

3.4.1	Main.txt	50
3.4.2	Limitations	53
3.4.3	Hardware	53
3.5	CONFIGURATION REGISTERS	53
3.5.1	Undocumented controls.....	54
3.6	FIRMWARE	54
4	CLOCK NETWORK.....	55
4.1.1	GC Pins.....	55
4.2	GLOBAL CLOCKS.....	55
4.3	G0, G1, G2 CLOCKS	57
4.3.1	Clock Synthesizers.....	57
4.3.2	Possible Outputs	57
4.3.3	Duty Cycle.....	58
4.4	EXT CLOCKS	58
4.4.1	EXT0	59
4.4.2	EXT1	59
4.4.3	Daughtercard zero-delay mode.....	59
4.4.4	SMA input	60
4.5	FB CLOCKS	61
4.6	NON-GLOBAL CLOCKS	61
4.6.1	Clock TP	61
4.6.2	Ethernet Clocks.....	62
4.6.3	DDR2 Clocks	62
4.6.4	SMA Clock E	62
5	TEST POINTS	63
5.1	POWER THRU-HOLE.....	63
5.2	POWER TP	64
5.3	DIMM POWER	66
5.4	“GC” TEST POINTS	67
5.5	CLOCK TEST POINTS	68
5.6	DIMM SIGNALS	68
6	USB INTERFACE	69
6.1	DIN.LH API.....	70
6.2	CONNECTING TO THE DN9000K10PCIE4GL	70
6.2.1	Windows XP	70
6.2.2	Windows Vista.....	71
6.2.3	Linux	71
6.2.4	Communication	71
6.3	VENDOR REQUESTS.....	71
6.3.1	VR_CLEAR_FPGA	72
6.3.2	VR_SETUP_CONFIG	72
6.3.3	VR_END_CONFIG.....	72
6.3.4	VR_SET_EP6TC (Read buffer size)	72
6.3.5	VR_MEM_MAPPED (Configuration Registers)	73
6.3.6	Other Vendor Requests.....	73
6.4	MAIN BUS ACCESSES	73
6.4.1	Important Note About Endpoints	74
6.4.2	Performance.....	74
6.5	FPGA CONFIGURATION	74
6.5.1	Readback.....	75
6.6	USB HARDWARE.....	75
6.6.1	Cypress CY7C68013A.....	75
6.6.2	Activity LED.....	76
6.6.3	Configuration FPGA.....	76
6.6.4	Power.....	77
6.7	TROUBLESHOOTING.....	77
6.7.1	USB Controller Freezes	77
6.7.2	Main Bus always returns 0x_____ (Error Codes).....	77
7	PCI EXPRESS INTERFACE	77
7.1	ELECTRICAL	78
7.2	TIMING	78

INTRODUCTION

7.3	HOST INTERFACE, MECHANICAL	78
7.4	HARDWARE DESIGN NOTES	78
8	UNUSABLE PINS.....	78
8.1.1	NC Pins.....	79
8.1.2	Configuration.....	79
8.1.3	DCI	80
8.1.4	VREF.....	81
8.2	EXTENDED MB.....	81
9	SYSTEM MONITOR/ADC.....	82
10	RESET	82
10.1	POWER RESET	82
10.2	USER RESET	83
11	JTAG	83
11.1	FPGA JTAG.....	83
11.1.1	Compatible Configuration Devices	84
11.1.2	Identify	85
11.1.3	ChipScope	85
11.2	FIRMWARE UPDATE HEADER	85
11.3	ETHERNET JTAG.....	85
11.4	TROUBLESHOOTING.....	85
12	RS232 INTERFACE.....	85
12.1.1	Configuration RS232.....	86
13	TEMPERATURE SENSORS.....	87
14	ENCRYPTION BATTERY.....	87
15	LED INTERFACE.....	89
15.1	CONFIGURATION SECTION LEDs.....	89
15.2	USER LEDs	89
15.3	ETHERNET LEDs.....	90
15.4	POWER LEDs	91
15.5	UNUSED LEDs	91
16	DDR2	92
16.1	POWER	93
16.1.1	Interface Voltages	93
16.1.2	Changing the DIMM voltage.....	94
16.2	CLOCKING	94
16.3	SIGNALING	96
16.3.1	Standards	96
16.3.2	Serial Interface.....	96
16.3.3	Timing.....	97
16.4	COMPATIBLE MODULES	97
16.5	TEST POINTS	98
17	FPGA INTERCONNECT.....	99
18	MAIN BUS	101
18.1	MB SIGNALS	101
18.1.1	Disambiguation.....	102
18.1.2	Electrical.....	102
18.1.3	Timing.....	102
18.1.4	FPGA Stuffing Options.....	102
18.2	ERROR CODES	102
18.3	FPGA INTERFACE	103
18.3.1	Conventional Memory map	104

INTRODUCTION

19	ETHERNET	105
19.1	MII	106
19.1.1	Electrical	106
19.1.2	Timing	106
19.2	JTAG	109
20	EPROM	109
21	POWER	110
21.1	12V	110
21.2	3.3V	111
21.3	2.5V	111
21.4	GROUND	111
21.5	VOLTAGE REGULATION	111
21.6	POWER CONNECTIONS	111
21.7	POWER MONITORS	113
21.8	HEAT	113
21.8.1	Fans	113
21.8.2	Removing Heatsinks	114
21.8.3	Fan Tachometers	114
22	CONNECTORS	115
22.1	FPGA USER INTERFACE CONNECTORS	115
22.1.1	Comments	116
22.2	NON-FPGA USER INTERFACE CONNECTORS	116
22.2.1	Comments	117
22.3	NOT-FOR-USE CONNECTORS	117
22.3.1	Comments	117
23	MECHANICAL	117
24	DAUGHTERCARD HEADERS	119
24.1	DAUGHTER CARD PHYSICAL	120
24.1.1	Daughter Card Locations and Mounting	120
24.1.2	Types 2 Short (400pin Short)	122
24.1.3	Insertion and removal	123
24.2	DAUGHTER CARD ELECTRICAL	125
24.2.1	Pin assignments	125
24.2.2	CC, VREF, DCI	126
24.2.3	Global clocks	126
24.2.4	Timing and Clocking	127
24.2.5	Power and Reset	128
24.2.6	VCCO Voltage	128
24.2.7	VCCO bias generation	128
24.3	ROLLING YOUR OWN DAUGHTERCARD	129
25	TROUBLESHOOTING	129
25.1	THE BOARD IS DEAD	129
25.2	THE FPGAS WON'T PROGRAM	130
25.3	MY DESIGN DOESN'T DO ANYTHING	130
25.4	THE DCMs WON'T LOCK	131
25.5	THE _____ SIGNAL ON MY BOARD IS GOING CRAZY ON MY OSCILLOSCOPE	131
CHAPTER 5: THE REFERENCE DESIGN		133
1	WHAT THE REFERENCE DESIGN DOES	133
1.1	EXAMPLE USAGE OF ALL INTERFACES	133
1.2	REFERENCE DESIGN TYPES	134
1.3	USING THE REFERENCE DESIGN	135
2	REFERENCE DESIGN MEMORY MAP	135
3	INTERCONNECT (DC)	137

INTRODUCTION

3.1	USING THE DESIGN.....	137
3.2	RUNNING THE TEST.....	138
4	DDR2 INTERFACE	138
4.1	PROVIDED FILES.....	138
4.2	USING THE DESIGN.....	138
4.3	RUNNING THE TEST.....	139
5	CLOCK COUNTERS.....	139
6	LEDS.....	139
7	SIMULATING THE REFERENCE DESIGN.....	139
8	COMPILING THE REFERENCE DESIGN.....	140
8.1.1	<i>The Xilinx Embedded Development Kit (EDK)</i>	<i>140</i>
8.1.2	<i>Xilinx XST</i>	<i>140</i>
8.1.3	<i>Xilinx ISE</i>	<i>140</i>
8.1.4	<i>The Build Utility: Make.bat.....</i>	<i>140</i>
8.2	BITGEN OPTIONS	141
8.3	VHDL	141
9	LVDS REFERENCE DESIGN	141
9.1	PROVIDED FILES.....	142
9.2	USING THE DESIGN.....	142
9.3	RUNNING THE TEST.....	142
10	ETHERNET AND PCI EXPRESS DESIGN	142
CHAPTER 6: ORDERING INFORMATION.....		145
1	SECTION TITLE	145
2	FPGA OPTIONS.....	145
2.1	FPGA A-F:.....	145
2.2	CES PARTS.....	145
2.2.1	<i>Hardware Errata Details</i>	<i>146</i>
2.3	“SMALL” FPGAS.....	146
2.4	SPEED GRADES.....	148
2.5	UPGRADE POLICY.....	149
3	OPTIONAL EQUIPMENT.....	149
3.1	COMPATIBLE DINI GROUP PRODUCTS	149
3.1.1	<i>Memories</i>	<i>149</i>
3.1.2	<i>Extenders</i>	<i>150</i>
3.1.3	<i>Daughterboards</i>	<i>150</i>
3.2	COMPATIBLE THIRD-PARTY PRODUCTS	151
4	COMPLIANCE DATA	152
4.1	COMPLIANCE.....	152
4.1.1	<i>EMI</i>	<i>152</i>
4.1.2	<i>PCI-SIG</i>	<i>152</i>
4.2	ENVIRONMENTAL.....	152
4.2.1	<i>Temperature.....</i>	<i>152</i>
4.3	EXPORT CONTROL.....	152
4.3.1	<i>Lead-Free.....</i>	<i>152</i>
4.3.2	<i>The USA Schedule B number based on the HTS.....</i>	<i>152</i>
4.3.3	<i>Export control classification number ECCN</i>	<i>152</i>
4.4	MISSION CRITICAL	152

Introduction

Congratulations on your purchase of the DN9000K10PCIE4GL logic emulation board. If you are unfamiliar with Dini Group products, you should read Chapter 2, Quick Start Guide to familiarize yourself with the user interfaces the DN9000K10PCIE4GL provides.

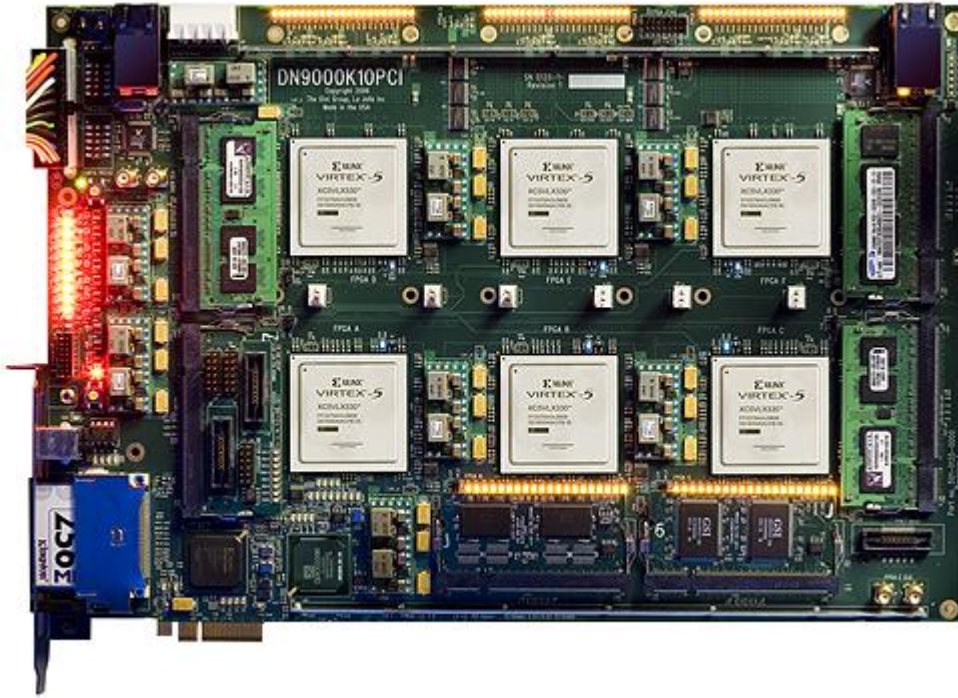


Figure 1 DN9000K10PCIE4GL – shown with optional memory modules installed; User LEDs, power fail LEDs, and Gigabit activity LEDs all glowing inexplicably. Heatsinks negligently left uninstalled.

1 Manual Contents

This manual contains the following chapters:

Introduction

Reader's Guide to this manual; List of available documentation and resources available.

Quick Start Guide

Step-by-step instructions for powering on the DN9000K10PCIE4GL, loading and communicating with a simple, provided FPGA design, and using the board's common control features.

Controller Software

A summary of the functionality of the provided software. Implementation details for the remote USB board control functions and instructions for developing your own USB host software.

Hardware

Detailed description and operating instructions of each individual circuit on the DN9000K10PCIE4GL, a description of each user-accessible interface and user features.

The Reference Design

Detailed description of the provided DN9000K10PCIE4GL reference design. Implementation details of the reference design interaction with DN9000K10PCIE4GL hardware features.

Ordering Information

Contains a list of the available options and available optional equipment. Some suggested parts and equipment available from third party vendors. Compatibility lists.

2 Conventions

This document uses the following conventions. An example illustrates each convention.

2.1 Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
Prefix “0x”	Indicates hexadecimal notation	Read from address 0x00110373, returned
Letter “#”, “N”	Signal is active low	INT# is active low RSTn is active low

2.2 Manual Content

2.2.1 File names

Paths to documents included on the User CD are prefixed with “D:\”. This refers to your CD drive’s root directory when the User CD is inserted in your Windows computer.

Alternately, copy the entire contents of the User CD to your hard drive, and allow D:\ to refer to this path. Due to limitations of the Xilinx ISE software, we recommend a path without space characters in it. (Bad places include C:/Documents and Settings/username/Desktop/)

2.2.2 Physical Dimensions

By convention, the board is oriented as shown in the above board photo, with the “top” of the board being the edge near FPGA D, E, and F with daughtercard headers. The “right” edge is near FPGA C and F. The “left” side is the side with the PCI bezel. “Top” side refers to the side of the PWB with FPGAs and fans; the “back” side is the side with the daughtercard connectors.

INTRODUCTION

The reference origin of the board is the center of the lower PCI bezel mounting hole. Physical dimensions are given in millimeters.

2.2.3 Part Pin Names

References to individual part's pin are given in the form <X><Y>.<Z>; The <X> is one of: U for ICs, R for resistors, C for capacitors, P or J for connectors, FB or L for inductors, TP for test points, MH for mounting structures, FD for fiducials, BT for sockets, DS for displays (light-emitting diodes), F for fuses, PSU for power supply modules, Q for discrete semiconductors, RN for resistor networks, G for oscillators, X for sockets, Y for crystals and the PCI bezel. <Y> is a number uniquely identifying each part from other parts of the same class. <Z> is the pin or terminal number or name, as defined in the datasheet of the part. Datasheets for all standard and optional parts used on the DN9000K10PCIE4GL are included in the Document library on the user CD.

2.2.4 Schematic Clippings

Partial schematic drawings are included in this document to aid quick understanding of the features of the DN9000K10PCIE4GL. These clippings have been modified for clarity and brevity, and may be missing signals, parts, net names and connections. Unmodified Schematics are included in the User CD as a PDF. *Please refer to this document when designing an interface in the FPGA.* Use the PDF search feature to search for nets and parts.

2.3 Terminology

Abbreviations and pronouns are used for some commonly used phrases. The user is assumed to know the meaning of the following:

Spartan

Spartan refers to the Spartan-3 FPGA device used by the DN9000K10PCIE4GL to perform configuration circuit functions. It is used interchangeably with “configuration circuit”

DCM, DLL, PLL

“Digital Clock Manager” or “Digitally-locked-loop”. This is a clock synthesis module in a Virtex-5 FPGA. PLL is “Phase-locked-loop”. See Xilinx documentation.

LVDS

“Low-Voltage differential signaling”. A signaling standard with a 1.2V DC, and 300mV AC level. In this manual and in advertisements, LVDS is often used where “Differential Signal” should be used instead.

Net, Signal, Plane, rail

A net is an electrically continuous piece of conductor on the PCB before assembly. Signal can refer to an electrically continuous conductor on the PCB, or to the logical meaning of that net. Plane is a net for voltage sources. Rail is also used to mean a power net.

GND, ground, grounded

GND is a net on the DN9000K10PCIE4GL, to which all voltages are referenced. “Ground” is

equivalent”. Grounded means “connected to GND”. There is a single ground net on the DN9000K10PCIE4GL

3 Resources

The following electronic resources will help you during development with your board.

3.1 User CD

The User CD contains all the electronic documents required for you to operate the DN9000K10PCIE4GL. These include schematics, the user manual, FPGA reference designs, and datasheets. The directory structure of the CD is as follows

3rdPartySoftware\Acrobat Reader 7.0\	Required to read .pdf documents (Windows)
Config_Section_Code\ code.	The DN9000K10PCIE4GL firmware source code.
ConfigFPGA\ MCU\	These sources are not intended to be used for development.
Datasheets\	A datasheet for every part used on the board. You will need these to interface successfully with resources on the DN9000K10PCIE4GL.
Documentation\Manual\	Contains this document
FPGA_Reference_Designs\ common\ DN9000K10PCIE4GL\ Programming_Files\ certify\	Contains the source and compiled programming files for the Dini group’s DN9000K10-PCI reference design. Also, board description files and simulation models.
PCI_Software_Applications\Aetest\	Source and binaries for the provided PCI-hosted controller software.
Schematics\Rev1\	Contains a PDF version of the board schematic. Search the PDF using control-F
USB_Software_Applications\ driver\ USB_CMD_Line_AETEST_USB\ USBController\	Contains source and binaries for the provided USB-hosted controller applications.

3.2 Dinigroup.com

The most recent versions of the following documents are found on the product web page <http://dini group.com/DN9000k10PCIE4GL.php>

INTRODUCTION

User's Manual (this document)
Certify Board Description File
Errata
USB Controller executable

3.3 Errata and Customer Notifications

The Errata sheet (available at www.dinigroup.com) lists all cases where the DN9000K10PCIE4GL is found to have failed to meet advertised specifications, or where an error in schematics or documentation is likely to cause a difficult-to-debug error by the user.

The customer is notified when there is an update to the Errata list after the board is shipped. Customers are not notified when changes are made to other documents including the reference design, USB Controller and User Manual. These documents change on a weekly basis or faster. You may always request a duplicate User CD. We will also be happy to provide the latest version of documents via email to customers.

As of May 23, 2007, there were no errata for this product.

3.4 Schematics and Netlist

Unmodified Schematics are included in the User CD as a PDF. Use the PDF search feature to search for nets and parts.

3.4.1 Netlist

In lieu of providing a machine-readable version of the schematic, the Dini Group provides a text netlist of the board. This netlist contains all nets on the board that connect to user IO on any FPGA. When interfacing with any device or connector on the DN9000K10PCIE4GL you should use either the provided ucf, or the netlist to generate the pinout. The netlist is located on the user CD at

D:\Schematics\Rev_01\DN9000K10PCIE4GL_customer_netlist.net

3.4.2 Net name conventions

All “power” nets begin with a +, - symbol, or GND

All clock signals begin with “CLK”

Two sides of a differential signal differ by one character “p” or “n”. This character is near the end of the net name.

Active low signals end in #. In the provided UCF files, the # is replaced by an “N”.

3.5 Datasheet Library

Datasheets for all parts used, or interfaced to, on the DN9000K10PCIE4GL are provided on the user CD. In order to successfully use the DN9000K10PCIE4GL, you will have to reference

INTRODUCTION

these datasheets. The interface descriptions given in this user manual typically end with electrical connectivity.

Especially read the Virtex-5 user guide. The copy provided on the user CD is only recent as of the DN9000K10PCIE4GL product announcement.

3.6 Xilinx

Virtex-5 is a brand-new device, and technical questions about getting the FPGA and ISE software to behave like you expect should be directed to a Xilinx FAE. Also use

WebCase	http://www.xilinx.com/support/clearxpress/websupport.htm
AnswerBrowser	http://www.xilinx.com/xlnx/xil_ans_browser.jsp
ISE Manual	http://www.xilinx.com/support/sw_manuals/xilinx82/index.htm
Virtex 5 Manual(s)	http://www.xilinx.com/xlnx/xweb/xil_publications_display.jsp?

3.7 Dini Group Reference Designs

The source code to the reference designs are on the User CD. Please copy and use any code you would like. The reference designs themselves are not deliverables, and as such receive limited support.

3.8 Board Models

Certify board models and other simulation models for the DN9000K10PCIE4GL are provided on the user CD.

D:\FPGA_Reference_Designs\DN9000K10PCIE4GL\certify

3.8.1 Base System Builder

We don't have a system builder file. As us if we can make one.

4 Email and Phone Support

Dini Group technical support for products can be reached via email at support@dinigroup.com. Our phone number is (USA) 858-454-3419. Please do not send .exe files, .vb files, or .zip files containing other .zip files as attachments, as we will not receive these emails. Please include the board's serial number in your email. This will allow us to reference our records regarding your board.

Before contacting support you should complete the following:

- 1) Follow the debugging steps in the troubleshooting sections at the end of the hardware chapter, and in any applicable interface sections.
- 2) Test the applicable interface(s) using the provided software and .bit files, to help rule out hardware failures.

INTRODUCTION

Quick Start Guide

The Dini Group DN9000K10PCIE4GL can be used and controlled using many interfaces. In order to learn the use of the most fundamental interfaces of the board (FPGA Configuration, USB data movement, etc.) please follow the instructions in this quick start guide. The guide will also show you how to run the board's hardware test to verify board functionality. (The board has already been tested at the factory).

1 Provided Materials

Examine the contents of your DN9000K10PCIE4GL kit. It should contain:

- DN9000K10PCIE4GL board
- CompactFlash card containing the FPGA configuration “.bit” files required to run the hardware test.
- USB CompactFlash card reader
- RS232 IDC header cable to female DB9
- A “Hard Drive-to-PCI express” power cable adapter
- USB cable, black.
- CD ROM containing:
 - Virtex 5 Reference Designs
 - User manual PDF
 - Board Schematic PDF
 - USB program (usbcontroller.exe)
 - PCI program (Aetest.exe)
 - Source code for USB program, PCI program and DN9000K10PCIE4GL firmware
 - Board netlist, certify model, and QL5064 simulation model

1.1 System Requirements

To compile Verilog designs for Virtex 5, ISE 8.2 (service pack 3 or later) is required. For some LX330 designs, you may need a Linux machine with a 64-bit processor and 6GB of RAM or greater. (Windows XP limits memory to 3GB)

To use the provided controller software, you need any Windows XP computer with USB 2.0.

To receive firmware updates to the DN9000K10PCIE4GL, you need a JTAG programming cable from Xilinx. Having this cable is basically mandatory, otherwise you would need to ship the board back to us to enable features or correct firmware bugs. You probably need a JTAG cable anyway for using embedded debugging software. We recommend the Xilinx Platform USB cable over their Parallel IV, because it sucks less.

2 Warnings

2.1 ESD

The DN9000K10PCIE4GL is sensitive to static electricity, so treat the PCB accordingly. The target markets for this product are engineers that are familiar with FPGAs and circuit boards. However, if needed, the following web page has an excellent tutorial on the “Fundamentals of ESD” for those of you who are new to ESD sensitive products:

<http://www.esda.org/basics/part1.cfm>



There are two large grounded metal rails on the DN9000K10PCIE4GL. The user should handle the board using these rails, as they are much less ESD sensitive than any other point on the board.

The 400-pin connectors are not 5V tolerant. No exposed surfaces on the board, except for the PCI edge connector are tolerant of voltages greater than 4V. According to the Virtex 5 datasheets, the maximum applied voltage to any IO signals on the FPGA is VCCO. This means you should not try to over-drive IOs in an FPGA interface above the interface voltage specified in this manual.

2.2 Other

Some parts of the board are physically fragile. Take extra care when handling the board to avoid touching the daughtercard connectors. Leave the covers on the daughtercard connectors whenever they are not in use. Use mounting hardware to secure daughtercards.

3 Pre-Power On Instructions

The image below represents your DN9000K10PCIE4GL. You will need to know the location of the following parts referenced in this chapter.

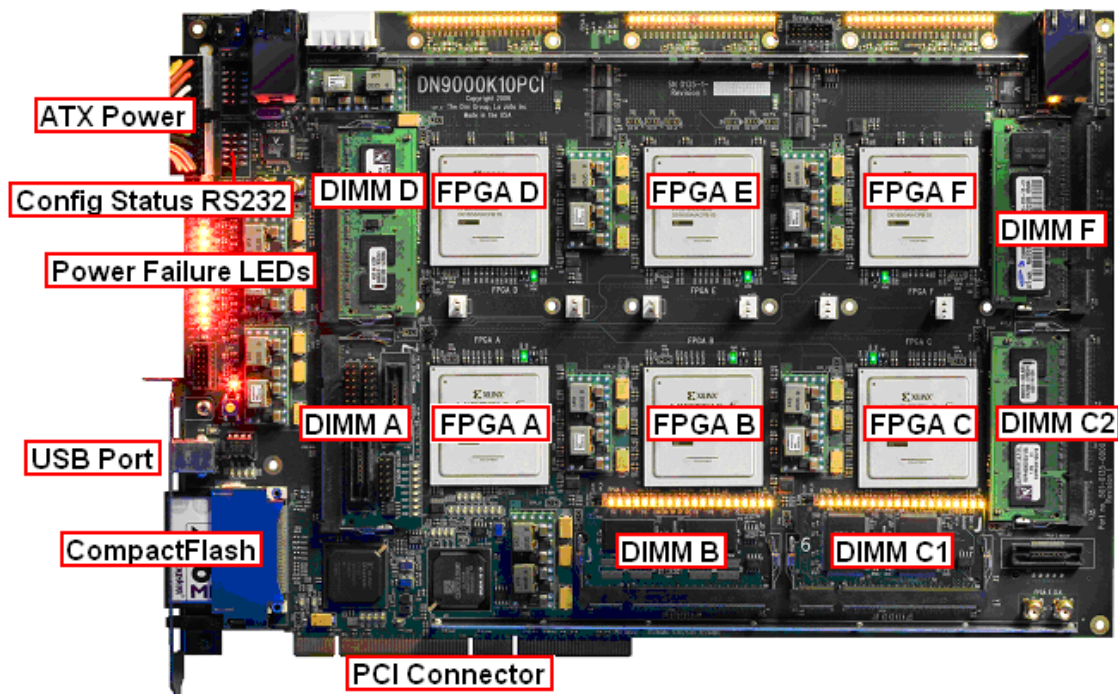


Figure 2 DN9000K10PCIE4GL configuration controls

The FPGAs on the board are names “FPGA A” through “FPGA F” FPGA A is in the lower left hand corner as shown in the above photo. FPGA B and C are to its right. The top row of FPGAs from left to right are named, “FPGA D”, “FPGA E”, and “FPGA F”

To begin working with the DN9000K10PCIE4GL, follow the steps below.

3.1 Install Memory

The DN9000K10PCIE4GL comes packaged without memory installed. If you want the Dini Group reference design to test your memory interfaces, you must install memory modules in the SODIMM slots on the board. The reference design supports DDR2 SODIMM modules in densities 256MB, 512MB, and 1GB with a CAS latency of 3. (Almost any modern off-the-shelf laptop memory)

The socket DIMMB is connected to FPGA B. The socket can accept any capacity DDR2 SODIMM module. Note that DDR1 modules will not work in these slots since they are a completely different pin-out and voltage level.

3.2 Prepare configuration files

The DN9000K10PCIE4GL reads FPGA configuration data from a CompactFlash card. To program the FPGAs on the DN9000K10PCIE4GL, FPGA design files (with a .bit file extension) put on the root directory of the CompactFlash card file using the provided USB card reader.

The DN9000K10PCIE4GL ships with a 256MB CompactFlash card preloaded with the Dini Group reference design. These “bit” files can also be found on the User CD. You can also compile the reference design source (provided on the CD) and place the generated .bit files on the Compact Flash card.

1. Insert the provided CompactFlash card labeled “Reference Design” into your USB card reader. Make sure the card contains the files:

```
FPGA_A.bit  
FPGA_B.bit  
FPGA_C.bit  
FPGA_D.bit  
FPGA_E.bit  
FPGA_F.bit  
main.txt
```

The files FPG_A-F.bit are files created by the Xilinx program bitgen, part of the ISE 8.2 tools. The file main.txt contains instructions for the DN9000K10PCIE4GL configuration circuitry, including which FPGAs to configure, and to which frequency the global clock networks should be automatically adjusted.

2. Take a look at the contents of main.txt. It might look something like this:

```
// Main.txt file for the DN9000K10PCIE4GL  
FPGA A: fpga_a.bit  
FPGA B: fpga_b.bit  
FPGA C: fpga_c.bit  
FPGA D: fpga_d.bit  
FPGA E: fpga_e.bit  
FPGA F: fpga_f.bit  
clock frequency: G0 350Mhz //LVDS frequency  
clock frequency: G1 250Mhz //DDR2 frequency  
clock frequency: G2 200Mhz //LVDS IDELAYCTRL  
//COMMENT TIME!
```

If you feel adventurous, you can change the clock frequencies by editing the main.txt file. Or you could cause an FPGA to not configure, by removing it's corresponding line.

3. Insert the CompactFlash card into the DN9000K10PCIE4GL's CompactFlash slot.

3.3 Install DN9000K10PCIE4GL in computer (Optional)

If you are not using the DN9000K10PCIE4GL in a PCIe slot, skip this step. You may instead choose to host the DN9000K10PCIE4GL over USB.

The DN9000K10PCIE4GL can be used with any sized PCIe slot. If you are using a 1x or 2x (does 2X exist?) slot, you will need an adapter card to physically fit.

Try buying there here:

<http://www.getcatalyst.com>
[PEA1621](#)

The board will work in a 8X or 16X slot, although PCI express may revert to 1x mode in this case.

This quick-start guide does not test the PCI express bus, and the DN9000K10PCIE4GL does not receive power from this slot. Therefore, placing a board in a PCI Express slot is really just using the PCI express slot as a place to put the board while we play with it.

3.4 Cables

3.4.1 Connect RS232 Cable

The configuration circuit displays status messages to an RS232 terminal. If (when) something goes wrong with configuration, this terminal will output error messages. Normally, you would only connect this cable when something is not working and you want to debug the problem.

Use the provided ribbon cable to connect the MCU RS232 port (P2) to a computer serial port to view feedback from the configuration circuitry during FPGA configuration. Run a serial terminal program on your PC (On Windows you can use HyperTerminal Start->Programs->Accessories->Communications->HyperTerminal) and make sure the computer serial port is configured with the following options:

- Bits per second: 19200
- Data bits: 8
- Parity: None
- Stop Bits: 1
- Flow control: None
- Terminal Emulation: VT100 (or None, if available)

3.4.2 Connect USB Cable

Use the provided USB cable to connect the DN9000K10PCIE4GL to a Windows computer (Windows XP is recommended).

If your board is installed in a PCI slot, the USB host is allowed to be the same computer as the non-USB host. The two computers should be connected to the same power outlet if this is the case.

3.4.3 Connect Power cable

The 6-pin power connector, P5, is required for operation. This is true whether or not the board is plugged into a PCI express slot. The type of power cable that is required is a 6-pin “PCI express graphics power” connector. At least one of these connectors is available on most new

ATX power supplies. For those customers who do not have a new ATX power supply, an adapter cable is provided so that the board can be powered from the 4-pin “hard drive” power connector instead.



Please note that the 4-pin and 8-pin “Molex” connectors coming off your power supply look like they would probably plug in to the 6-pin header on the board. However, if you somehow manage to do this let us assure you that it will result in a seventy-thousand dollar fireball.

4 Power On

Turn on the ATX power supply (Table-top USB Hosting) or the computer (PCIe hosting).

Some power supplies require a minimum load on the +5V rail. The DN9000K10PCIE4GL draws no current off this rail. In this case, the power supply may be unstable. You can work around this limitation by plugging a spare hard drive into one of the hard drive power connectors on the supply. Alternately, you could use a power supply with a zero minimum +5V load requirement.

When operating table-top, the power supply may not power on unless pin 14 (“PSON”) of the 20-pin “motherboard” connector is connected to ground. This wire is usually green.

When the DN9000K10PCIE4GL powers on, it automatically loads Xilinx FPGA design files (ending with a .bit extension), found on the CompactFlash card in the CompactFlash slot into the FPGAs, using the main.txt file as a guide.

4.1 View configuration feedback over RS232

As the DN9000K10PCIE4GL powers on, your RS232 terminal (connected to P2) will display information about the Configuration process. If FPGAs ever fail to configure using the Compact Flash card, this is the best place to look for help.

A typical RS232 power-on session is given below.

Rebooting from FLASH...please wait	This line has to do with the firmware-update mode
Setting G0...	
N 01 M: 000001000	
DONE	
Setting G1...	The board is setting the global clock frequencies according to the main.txt file on the CompactFlash card. The messages here are mostly

QUICK START GUIDE

<pre> N: 01 M: 000001000 DONE Setting G2... N: 01 M: 000001000 DONE ==-. DN9000K10PCIE4GL MCU FLASH BOOT -=.= -- FPGAS STUFFED -- A B -- COMPACTFLASH INFO -- MAKER ID: EC DEVICE ID: 75 SIZE: 32 MB -- FILES FOUND ON COMPACTFLASH CARD FPGA_B.BIT FPGA_A.BIT MAIN~1.TXT MAIN.TXT -- CONFIGURATION FILES -- FPGA A: FPGA_A.BIT FPGA B: FPGA_B.BIT --OPTIONS-- Message level set to default: 2 Sanity check is set to default: ON N: 00 M: 000001010 DONE Setting G0... N: 01 M: 000001100 DONE Setting G1... N: 01 M: 000001000 DONE *****CONFIGURING FPGA: A***** -- Performing Sanity Check on Bit File -- -- BIT FILE ATTRIBUTES -- FILE NAME: FPGA_A.BIT FILE SIZE: 003A943B bytes PART: 4v1x100ff151317:09:38 DATA: 2005/07/25 TIME: 17:09:38 Sanity check passedDONE WITH CONFIGURATION OF FPGA: A *****CONFIGURING FPGA B***** -- Performing Sanity Check on Bit File -- -- BIT FILE ATTRIBUTES -- FILE NAME: FPGA_B.BIT FILE SIZE: 003A943B bytes PART: 4v1x100ff151317:05:01 DATA: 2005/07/19 TIME: 17:05:01 Sanity check passedDONE WITH CONFIGURATION OF FPGA: B -- TEMPERATURE SENSORS -- A YES </pre>	only useful to whoever programmed the firmware Prints the FPGAs the configuration circuit thinks you have on your board. CompactFlash card debugging information. This lists the files found on the compact flash card. If this list is wrong there is something wrong with CompactFlash. The MCU reads the contents of the file MAIN.TXT and executes each instruction line. Here the MCU is setting the clocks according to instructions in MAIN.TXT The MCU is configuring FPGA A according to instructions in MAIN.TXT Debugging information about the .bit file. The MCU is configuring FPGA B according to instructions in MAIN.TXT The MCU is setting the temperature threshold (Causes the FPGAs to automatically disable when overheating)
--	---

B YES FPGA Temperature Alarm Threshold: 80 degrees C	
---	--

Figure 3 RS232 Output

4.2 Check LED status lights

The DN9000K10PCIE4GL has many status LEDs to help the user confirm the status of the configuration process.

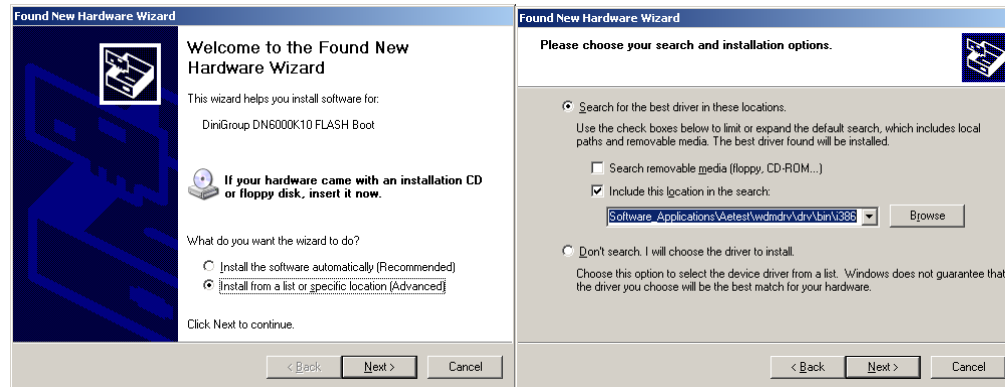
1. Check the power Failure LEDs to confirm that all voltage rails of the DN9000K10PCIE4GL are within tolerance. If the voltage of any critical power net on the DN9000K10PCIE4GL is too high or too low, the board will be held in reset and at least one of the red LEDs will light. The LEDs are located down the left side of the PCB. Each one is labeled with the voltage that it represents. Normally, all of these LEDs are off. If any of these LEDs light, there is a power problem with the board, and you should contact us. First, make sure that the output of the power supply is acceptable. If the 5V, 3.3V or 12V power fail LED is lit, you probably have a problem with the power supply you are connected to, and not with the DN9000K10PCIE4GL.
2. Reset LED. When the board is in reset for any reason, including power failure, or pressing the reset button, this LED will light RED. The LED is located below the RED voltage LEDs next to the logic-reset button.
3. Check the Spartan FPGA status LED located near pin 1 of the PCI edge connector. This LED should remain GREEN as long as the board is powered on, except for a quarter second as the board is powering on. This LED indicated the configuration and control FPGA is on. If this LED is not on, it indicates a problem with the board or firmware.
4. Check the “DONE” LEDs of each FPGA. When an FPGA is configured, a green LED labeled “DONE” will glow.
5. Check the FPGA A status LEDs located just below the FPGA A. These LEDs should be blinky and active if the Dini Group reference design is correctly loaded. Repeat this step for FPGAs B,C,D,E, and F. FPGA B and C LEDs are located just below their respective FPGAs. The LEDs for FPGA D,E and F are located along the top edge of the board.
6. Check the CF activity LED, located just below the CompactFlash socket. When the board is reading off the CompactFlash card during configuration, this LED should blink.

5 Run USB Controller

This section will get you started with USB and show you how to operate the provided software.

5.1 Driver Installation

When the DN9000K10PCIE4GL powers on, or you connect it to a USB port for the first time, the computer will ask you to install a driver.



In the window that appears, select “Install from a list or specific location”. Select Next.

Click “Include this location in the search” and browse to
D:\USB_Software_Applications\driver\windows_wdm
Select Next.

In the next window, select the item in the list “Dini Group ASIC Emulator”. Click FINISH.

After Windows installs the driver, you will be able to see the following device in the “USB Controllers” group in the Windows device manager: “Dini Group ASIC Emulator”.

5.2 Operating the USB Controller program

Run the USB controller application found on the product CD in
D:\USB_Software_Applications\USBController\USBController.exe

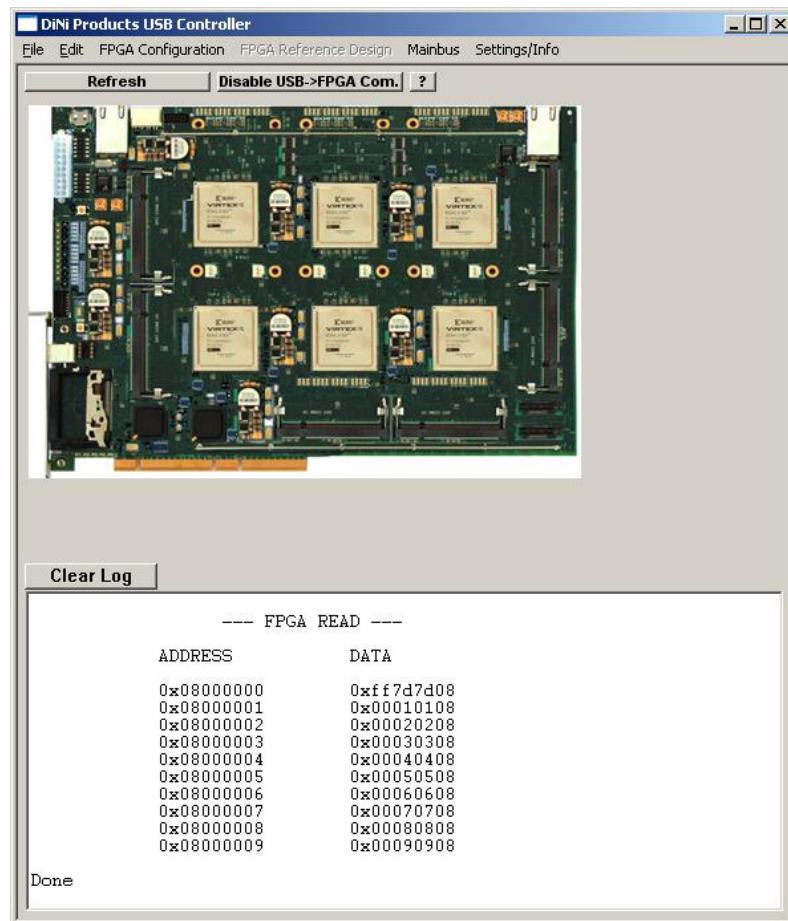


Figure 4 USB Controller Window

This window will appear showing the current state of the DN9000K10PCIE4GL. Next to each FPGA a green light will appear if that FPGA is configured successfully. The above window shows the USB Controller connected to a DN9000K10PCIE4GL with six LX110 FPGAs.

5.2.1 Configure an FPGA

Even though the reference design should already be loaded (because you had a CompactFlash card installed when the board powered on), let's configure an FPGA over USB.

Clear an FPGA of its configuration. Right-click on an FPGA and select from the popup menu, "Clear FPGA". The blue light above the FPGA on the GUI and on the board should turn off.

To re-configure that FPGA using the USB Controller program, right-click on the FPGA and select Configure FPGA via USB from the popup menu. The program will open a dialog box for you to select the configuration file to use for configuration. Browse to the provided user's CD "D:\FPGA_Reference_Designs\Programming_Files\DN9000K10PCIE4GL\MainTest\LX330\fpga_a.bit"

If you are configuring an LX220 or LX110 device you should select a bit file from the LX220 or

LX110 directories instead. Failing to select the correct type of bit file will result in the USB Controller program to warn you, and the FPGA fail to configure. The program will report the status of the configuration when it finishes. “DONE did not go high”. This refers to the DONE SelectMap signal, which is asserted by the FPGA when it is properly configured.

If you are configuring FPGA B or FPGA C, you should select fpga_b.bit or fpga_c.bit instead. Should you configure the wrong FPGA with the wrong bitfile, the FPGA will succeed to configure, but probably won't function properly. This is not recommended because it could lead to bus contention and excessive heat generation.

```
Done
FPGA B cleared successfully.
FPGA A cleared successfully.
Doing a sanity check...Sanity Check passed. Configuring FPGA
B via USB...please wait.
File
D:\\dn_BitFiles\\DN9000K10PCIE4GL\\MainTest\\LX330\\fpga_b.bit
transferred.
Configured FPGA B via USB
```

Figure 5 USB Controller Log Output

The message box below the DN9000K10PCIE4GL graphic should display some information about the configuration process. When the configuration is successful, the green LED should re-appear next to the FPGA.

5.2.2 Set Clock Frequencies

To change the clock frequencies of G0, G1 or G2, select the “Clock settings” option from the “Settings” menu.

A dialog box appears asking to which frequency you would like to set each clock. Enter 100, 100, 200 for G0, G1 and G2 respectively. The log window will display feedback including what frequencies the clocks were set to. The actual frequency to which each clock is set may differ from the frequency you entered because the frequency synthesizers have a limited granularity.

5.2.3 Check clock frequencies

The reference design will measure the frequency of each of the global clocks on the board and print them out if you select “read back clock frequencies” from the reference design menu. This will only work if the reference design is running and the “enable usb” option is selected.

5.2.4 Run Hardware Test (DDR2)

First, hit the “Enable USB->FPGA communication” button. This must be done before this (or any usb) program can interact with the reference design. You must also have the reference design loaded, and a DDR2 module installed in a memory socket connected to the FPGA using that reference design. Also, the clock settings must be correct. Follow the procedure in the previous section to accomplish this.

From the FPGA Memory menu, select Test DDR. A box will appear and ask which FPGA should be tested. The log window will report whether the test passed. If it fails, it will print a list of addresses and data that failed. If you have a board with LX110 or LX220, remember some of the DDR SODIMM connectors (A,C2,D,F) will not work on your board.

If you would like to simulate a failure, you could repeat this guide with a broken DDR2 module or a squirrel installed, or with the DDR2 module removed. Other tests that could be performed from the USB Controller (but aren't part of this quick-start) are FPGA interconnect tests, Ethernet tests, and some other thing. For information on running these tests, see the Software chapter.

5.3 Getting data to and from the FPGA

The USB Controller program also allows you to easily configure and transfer data to and from the user design on the emulation board. This data transfer occurs over the boards "MainBus". This interface is described in the Hardware chapter.

Some users may choose not to implement the "MainBus" interface, and use these signals for general-purpose FPGA interconnect. To allow this, by default the main bus is disabled, and the Host interface (USB in this case) is prevented from operating it. To override this setting, hit the "Enable USB->FPGA communication" button near the top of the window.

To read data from the FPGA design (the Dini Group reference design), select from the menu MainBus->Read

In the resulting dialog box, enter the address 0x08000000 in the address box, and "10" in the "number of DWORDs" box. Press OK, and then DONE. The result of the read is printed to the USB Controller log window.

```
FPGA READ
0x08000000 0x00000000
0x08000001 0x05000135
0x08000002 0xDEAD1234
0x08000004 0x1D000000
...
```

Figure 6 USB Controller Log Output

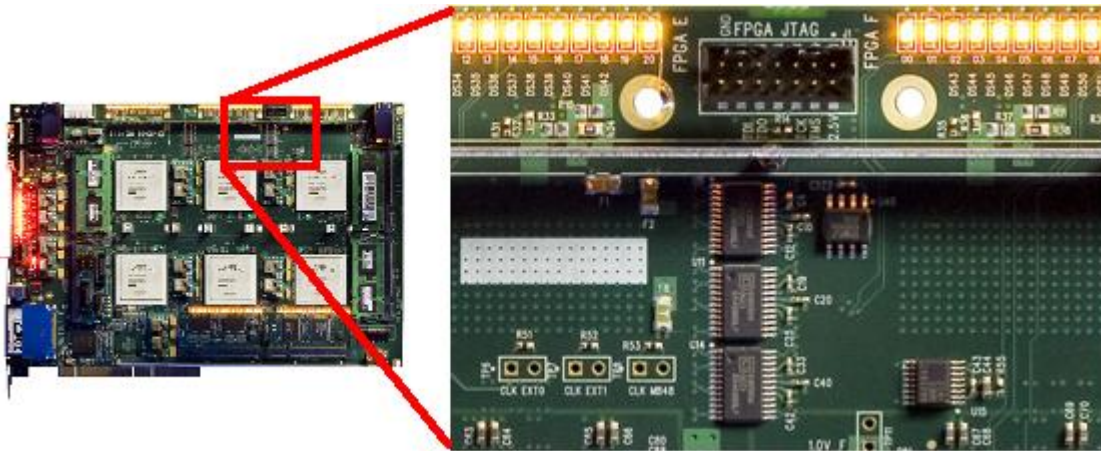
The address 0x08000000 is by convention of the "MainBus" interface to be mapped to registers physically residing within FPGA A on the DN9000K10PCIE4GL. If FPGA A is not loaded with the Dini Group reference design (or a design that implements the MainBus slave), then all DWORDs will return 0xDEADDEAD, which is the timeout sentinel value.

6 Communicating over the Serial Port

You may want to communicate with your design over the user serial port (P1). The MainTest reference design that you already loaded has an asynchronous loop back on this port. If you want to test this connection, connect an RS232 terminal to the header and type stuff. The port should echo back what you are typing.

7 Scan the JTAG chain

If you wish, you can program the FPGAs using their JTAG interface. Connect a Xilinx Platform USB cable into the FPGA JTAG port (J1), and open the IMPACT program that is installed with Xilinx ISE 8.2.



When you connect the Platform USB cable for the first time, the driver will install three times in a row, like a retarded parrot.

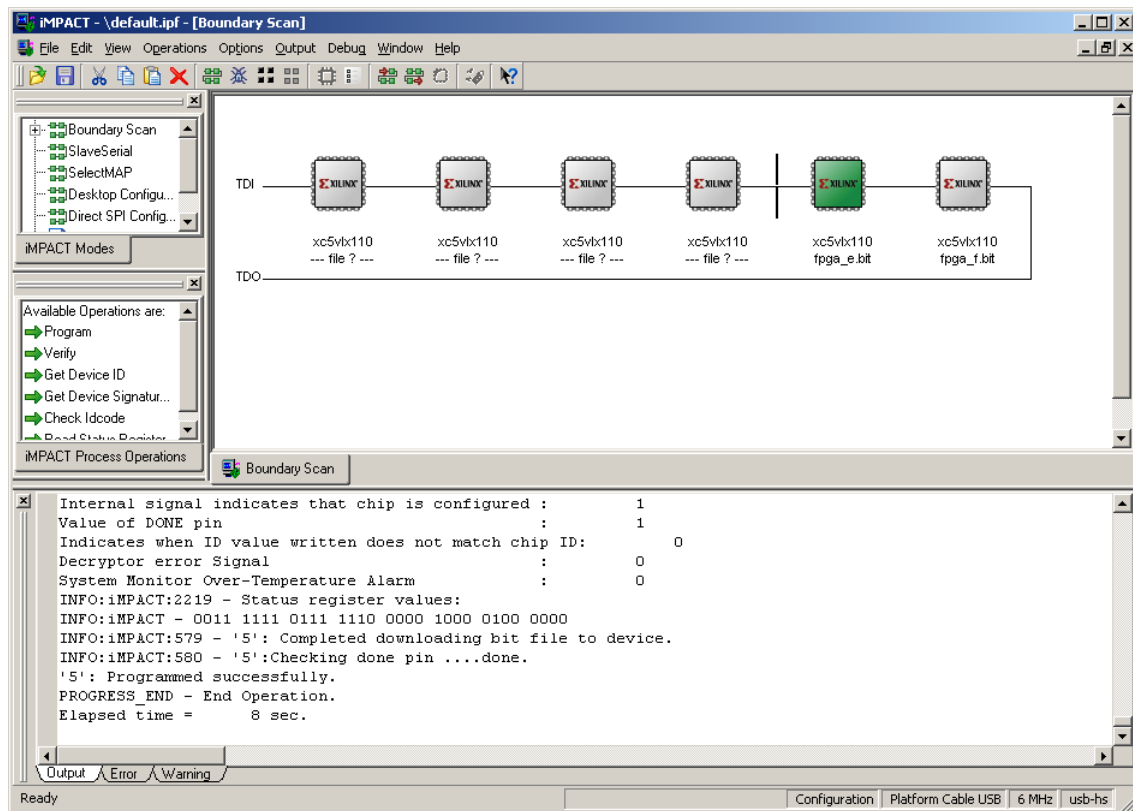
The program “scans the chain” to auto-detect the type and number of FPGAs installed on your board and display them on the screen. Right click on an FPGA and select “choose configuration file”. Browse to the bit files provided on the user CD

D:\FPGA_Reference_Designs\Programming_Files\DN9000K10PCIE4GL\MainTest\LX110\fpga_A.bit

Right-click on the FPGA again and select “Program”. Make sure “verify” is not selected, or it won’t work and it won’t tell you why.

This JTAG port should also be used for visibility products like Xilinx ChipScope.

QUICK START GUIDE



If you don't see the above screen, and instead see a XC3S1000 and an 18V04, then you plugged into the wrong header. Smart!!

7.1 Moving On

Congratulations! You have just programmed the DN9000K10PCIE4GL and learned all of the features that you have to know to start your emulation project. If you are new to Xilinx FPGA, you might want start by compiling the reference design using the provided ISE projects, and adding code to the reference design, until you are comfortable with the design flow. You should also use the provided UCF constraint file as a starting point for your UCF file.

Controller Software

The DN9000K10PCIE4GL can be hosted from USB or PCI. As an example to hosting using these interfaces, the Dini Group provides some controller software that allows configuring FPGAs, and changing the board settings. For more complex host behavior, such as interactively transferring data to and from the board from the host computer, you may have to develop your own host software, either USB or PCI. At the end of this chapter, there is a programmer's guide to help you interface to the DN9000K10PCIE4GL. This, along with the source code of the example software should be able to get you communicating with the DN9000K10PCIE4GL.

The software included with the DN9000K10PCIE4GL is

- USB Controller – A Windows XP only GUI application capable of configuring FPGAs, sending data to the user FPGA core via USB, changing board settings, and running hardware tests.
- AETest_usb - A cross-platform (Windows, DOS, Linux, Solaris) command-line application capable of configuring FPGAs, sending data via USB, and changing board settings.

These programs and the source code for them can be found on the user CD
D:\USB_Software_Applications\USB_CMD_Line_AETEST_USB\
D:\USB_Software_Applications\USBController\

Precompiled windows XP binaries for USB Controller, and AETest_usb, and AETest are provided on the user CD as a Microsoft Visual Studio 6 project. Visual Studio 6 or later is required to compile these programs.

All three programs use a driver provided by the Dini Group.

The USB driver can be found at
USB_Software_Applications\driver

1 USB Controller

The USB Controller program is intended to

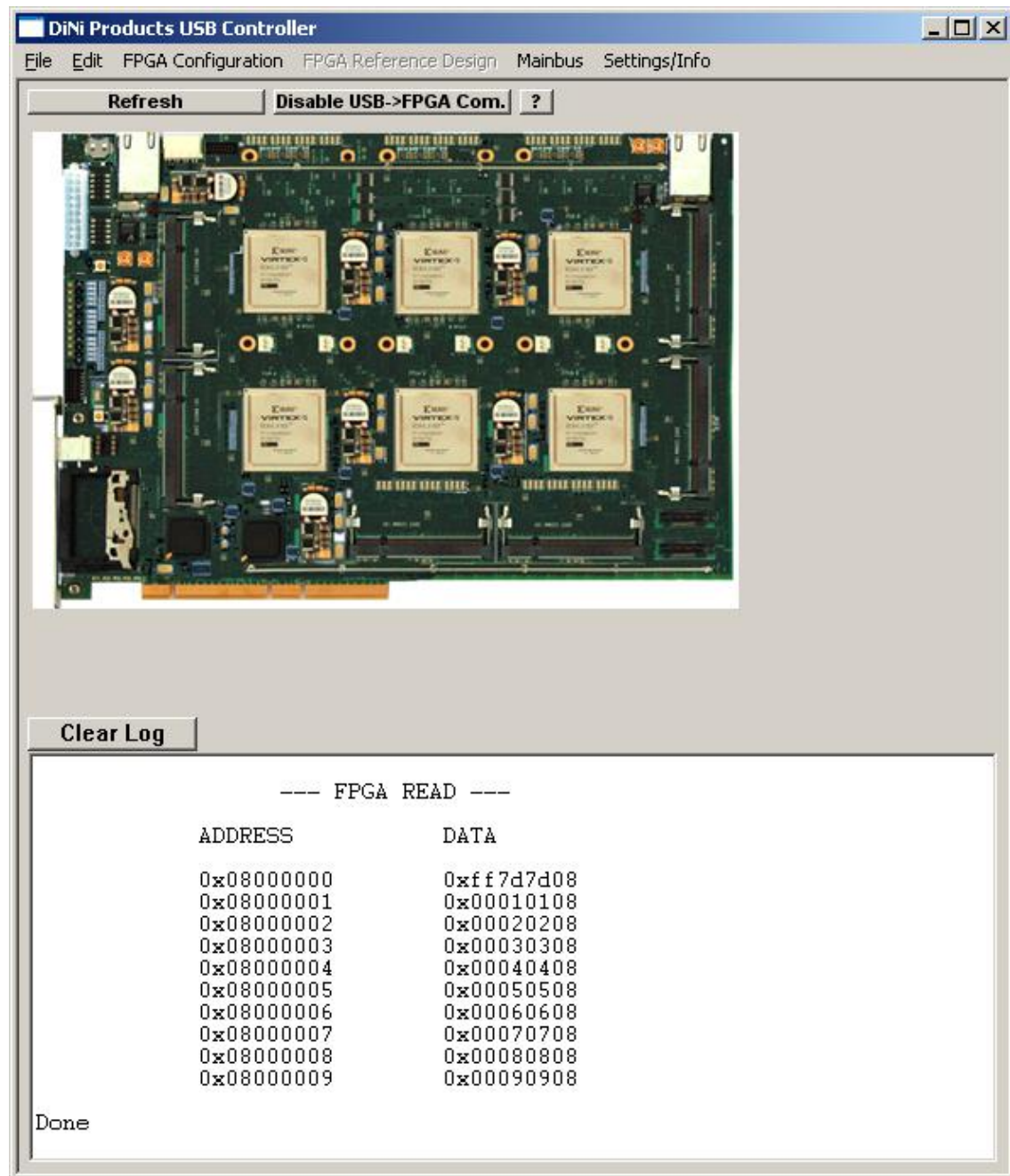
- Verify Configuration Status
- Configure FPGA(s) over USB
- Configure FPGAs via CompactFlash card
- Clear FPGA(s)
- Reset FPGA(s)
- Set Global clocks frequency
- Update MCU FLASH firmware

The following function interface with the Dini Group reference design.

- Read/Write to FPGA(s) – see the chapter Reference Design for a description of the Main bus interface.
- Test DDRs/FLASH/Reigsters/FPGA Interconnect

1.1 Main Window

The main USB Controller window has the following components: a menu bar, a refresh button, a “Disable USB” button, and board graphic, and a message log.



Each item in the menu bar is described later in this section.

1.1.1 Refresh Button

The Refresh button updates the board graphic by querying the DN9000K10PCIE4GL and reading back it's status. The USB Controller program does not poll the board, and only updates the status when there is some user command. Items that may be updated when the refresh button is hit are:

CONTROLLER SOFTWARE

- Type of board connected (DN9000K10PCIE4GL in this case)
- Number of FPGAs installed
- Whether or not the FPGAs are configured (blue DONE LED on/off)
- Whether the Dini Group reference design is loaded in one or more FPGAs (disable/enable the “FPGA Reference Design” menu)
- Check whether USB is enabled



1.1.2 Disable/Enable USB



To communicate to the FPGA design using USB, the “Main Bus” interface is used. See the hardware chapter for more information on this interface. Some users elect not to use the Main Bus for USB communication. To allow these users to make use of the signals in the Main Bus for their own purposes, the USB Controller is careful not to use the Main Bus unless explicitly given permission by the user. The user can give permission to use Main Bus by pressing the “Enable USB->FPGA communication” button. It can revoke that permission by pressing the “Disable USB->FPGA communication” button. When the DN9000K10PCIE4GL powers on, it begins in the disabled state. The state is stored on the board, so that multiple programs accessing the DN9000K10PCIE4GL may prevent each other from using the Main Bus.

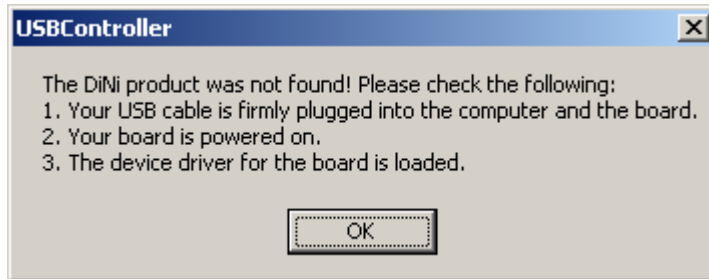
1.1.3 Log Window

This text box prints the result of each user command in USB Controller. There is a “clear log” button to clear the contents of this text box.

1.1.4 Board Graphic

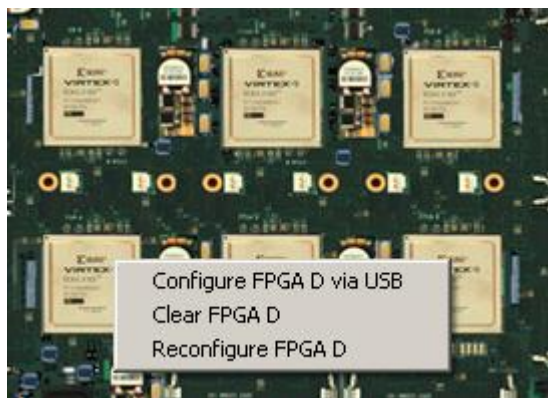
USB Controller's main window shows a graphic representing your DN9000K10PCIE4GL. The number of FPGAs that are installed on your board should appear in this graphic. If one or more FPGAs are configured on the board, a blue LED will glow next to the FPGA in this graphic window, just like on the actual board hardware itself.

If the USB Controller could not find a DN9000K10PCIE4GL connected to any USB port, this window will appear.



If the board is turned on and plugged in, the USB Controller should be able to detect it. If it does not, try opening the Device manager. You can right-click on the "My computer" icon and select "Hardware tab" and click the "Device Manager" button. This will display a list of the devices connected to your computer. If a Dini Group Logic Emulator appears in the USB section, then USB is working properly on the board, but the program is unable to connect to it. Select "Switch Device" from the File menu. If the board does not appear in the Hardware manager, then the DN9000K10PCIE4GL may be stuck in reset. See the "Troubleshooting" section in the Hardware chapter. Also, check the red "Reset" LED.

As well as providing visual feedback, the board graphic can be used to control configuration of the FPGAs. To do this, right-click on an FPGA in the graphic to show a contextual menu with the options: Configure, Clear and reconfigure.



Configure will show an Open... dialog for you to select the bit file you wish to use with the FPGA.. Clear FPGA will clear and reset the FPGA of its current configuration. Reconfigure

FPGA will configure the FPGA with whatever bit file that *this instantiation* of USB Controller used to configure that FPGA last.

1.2 Menu Options

The following sections describe each menu option and it's function.

1.2.1 File Menu

The File Menu has the following 2 options:

- a. Open – opens a file with the selected text editor (notepad by default). To change the text editor see Settings/Info Menu section
- b. About
Displays USB Controller version number, along with other things.
- c. Switch device
Displays a list of all Dini Group USB devices detects and allows the user to switch the “current” device. The USB Controller will behave as if the “current device” is the only attached Dini Group USB product. Under some situations, the USB Controller may automatically switch device when the “current device” is not valid.
- d. Exit – Closes the USBController application

1.2.2 Edit Menu

The Edit Menu performs the basic edit commands on the command log in the bottom half of the USBController window.

Copy, delete, select all

1.2.3 FPGA Configuration Menu

The FPGA Configuration Menu has the following options:

- (1) Refresh Window
This menu option is equivalent to hitting the “Refresh” button in the main window. It queries the board and updates the graphic for visual feedback.
- (2) Configure Via USB (individual)
This menu option allows you to configure an FPGA. It is equivalent to selecting an FPGA by clicking on it and selecting “Configure”, except that this menu option will display a dialog asking which FPGA to configure. Before any FPGA is configured in USB Controller, a “sanity check” is performed. This reads the header out of the binary bit file and determines whether the bit file is compatible with the FPGA installed on the DN9000K10PCIE4GL. It will prevent configuration if the “sanity check is not passed” This check can be disabled from the “Settings/Info” menu.

- (3) Configure via USB (using file)
This command allows the user to configure more than one FPGA over USB at a time. To use this option you must create a setup file that contains information on which FPGA(s) should be configured and what bitfiles should be used for each FPGA. The syntax of this file is similar or identical to the syntax of the CompactFlash main.txt interface. Details are found in the USB Controller manual on the user CD at
D:\USB_Software_Applications\USBController\doc\USBController_Manual.pdf
- (4) Configure via CompactFlash
This command causes the FPGAs to configure based on the instructions in the main.txt file on the CompactFlash card. It will also cause the commands and settings on the main.txt file to be re-issued.
- (5) Clear All FPGAs
This command resets all FPGAs, causing them to lose their configuration.
- (6) Reconfigure All FPGAs
This menu command is equivalent to selecting “reconfigure FPGA” in the context menu of each of the FPGAs. Each FPGA is cleared before being configured. The last bit file that was loaded via USB for each FPGA is loaded again into the FPGA. If an FPGA has not been loaded with a bit file using *this instance* of USB controller, it is skipped.
- (7) Reset
This command asserts the RESET# signal to all FPGAs simultaneously. This is the same signal that is asserted when the user hits the “Soft Reset” (User Reset) button. Its function in the user design is left for the user to define. In the reference design, it causes a global, asynchronous reset. This option also causes the SYS_RSTn signal on the daughtercards to be asserted.

1.2.4 FPGA Reference Design

- DDR Type/Size
When running the DDR2 test in the reference design, a register must be set to identify the type of memory installed in the DDR2 slot. This menu option allows you to set that register. When the FPGAs are reset, this register is automatically set to the correct value depending on the contents of the SPD interface of the installed memory.
- Read FPGA Clock Frequencies
This menu option measures and reads back the frequencies of the eight global clock networks, and displays them on the message log.

1.2.5 Main Bus

The way that user FPGA designs can communicate over USB is the “Main Bus” interface. The “Reference design” menu uses the main bus to read and write registers in the reference design to

control the board tests. These tests can be done by the using these menu options without the user having to understand the Main Bus interface or the main bus memory space and it's mapping to the reference design. The Main Bus menu allows direct control of the Main Bus. This can be useful if you are using your own FPGA core that implements the main bus.

- **Write DWORD**
This displays a dialog box for writing to the Main Bus address space. It includes some debugging features. All main bus transactions are of length 4 bytes (DWORD)
- **Read DWORD**
This redundantly displays the same dialog box as the option above, but with default settings appropriate for reading.
- **Write and Read DWORD**
This redundant command displays the same dialog box as above, but set up for reading and writing.
- **Test Address Space**
This writes and reads random data to the address range specified in a dialog box, and prints and error message when the read and write do not match.
- **Read Address Space to File**
This reads data from the main bus at the address specified, and writes the data to a binary file specified. Data on the main bus is in little-endian order. The address after each DWORD is implicitly incremented. This behavior can be turned off (contact support)
- **Write Address space from file.**
This reads data from a file and writes the data to the address on main bus specified. The data is written in little-endian order. The address is implicitly incremented after each DWORD of data. This behavior can be changed (contact support)
- **Send Command File**
Deprecated.

1.2.6 Settings/Info Menu

The Settings/Info Menu has the following options

- (1) **Change Text Editor**
This option changes the behavior or “Open” in the file menu and is otherwise undocumented. We would remove it, but exactly one user likes it.
(amy@dinigroup.com)

- (2) FPGA Stuffing information
Displays a list of the FPGAs on the board, and their type and speed grade. This information is stored in the firmware flash, and is basically just used for FYI.
- (3) Turn fans on/off
This command cannot be used with the DN9000K10PCIE4GL.
- (4) Board/Spartan/MCU version
This option is used to read the version number of the current board's firmware. There are two types of firmware, the "Flash" and the "Prom". The two types of firmware, the reference design, and the USB Controller application are only guaranteed to work when using corresponding versions of each. If you update one, you should update the others, or you risk being pestered by the USB Controller.
- (5) Read FPGA temperatures
Displays the current temperature of the on-die FPGA temperature sensors.
- (6) Force Memory Menu display
When the Dini Group reference design is not loaded in at least one FPGA, the "FPGA Reference Design" menu is disabled. This menu command causes that menu to be displayed in this situation. The USB Controller determines if the Dini Group reference design is loaded by reading a memory location on Main Bus and comparing the result to a predetermined value. This menu may also be disabled because the "USB->FPGA Communication" is disabled.
- (7) Toggle Sanity Check
This menu command changes the behavior of configuration where it reads the header in the binary bit file and determines if the file is compatible with the installed FPGA. This may be necessary if using bitstream encryption, or using a custom bitfile not created by ISE 8.2 bitgen.
- (8) Setup clock frequencies
This menu option displays a dialog box allowing the three global clock networks to be configured.
- (9) DN9000K10PCIE4GL clock sources
This allows you to select the source of G0, G1, G2, EXT0, and EXT1 clocks, which each have multiple available reference frequency options.
- (17) Test DDR2
This menu option displays a dialog box allowing testing of the DDR2 sockets on the DN9000K10PCIE4GL. If the Dini Group reference design is not loaded, the command will automatically load them into the FPGAs. If the clocks are not set, the command will automatically set them to settings compatible with the DN9000K10PCIE4GL reference design.

(18) One Shot Test

This menu option tests various functions on the board, automatically configuring the FPGAs and setting up the clocks as required. Note that for many of the tests to pass, a special test setup or fixture may be required. For tests that require hardware that you don't have (header test, Mictor test, SMA Test) you can disable using check boxes.

1.3 Unsupported Features

If you somehow discover these features, they are not intended for customer use.

1.3.1 Header Test

This test requires special hardware that you do not have.

1.3.2 Programmer's Menu

~~This menu is currently undocumented~~

~~Read Benchmark, Write Benchmark, Configuration Register, Stress Tests~~

1.4 Log File

If you run "one-shot test" a log file will be created in the same directory as USB Controller which will contain all of the output in the message window.

1.5 INI File

Some command considered "debugging" commands save persistence information in an "ini" file that gets created in the same directory as the USB Controller executable. This file should not be generated for most users. If it is generated, you can safely delete it, unless you like it for some reason. Some of the settings that can be stored in this file are the Text Editor selection settings, the location of (path to) the reference design programming files (for one-shot-test), and enabling the debug menu.

1.6 Compiling USB Controller

The code for USB Controller is provided, but we don't encourage you to look at it because it's terrible. If you want a starting point for a USB application, see AETest_usb.

If for some reason you must compile USB Controller (perhaps Dini Group was destroyed in a fire storm, or otherwise refuses to add whatever feature you want as a standard feature), then you will need Visual Studio 6. If you try to use VS.net, then the graphics will look funny, but you won't be laughing. The VS6 project file is provided.

2 AETest USB

AETest_usb is a program that has all of the functionality of the USB Controller, except there is no graphical user interface, it is cross-platform, and you have a little bit of a chance reading through and understanding the code.

2.1 Compiling AETest_usb

AETest_usb can be compiled using Microsoft Visual Studio 6 or later, or on any version of Linux that supports the usbdevfs library (not 4 years old).

A make file is provided, but you must un-comment one of the following lines to define which operating system you are running.

```
#DESTOS = WIN_WDM
#DESTOS = LINUX
#DESTOS = SOLARIS
```

Also uncomment one of these lines.

```
#include Makefile.make
#!INCLUDE Makefile.nmake
```

To start the compilations in Windows, you should run **nmake**. In any other OS, you should run **make**.

2.1.1 Cygwin

Nope. VS6 only.

3 Rolling Your Own Software

Most customers who need to use USB or PCI as a data interface to their FPGA designs write their own USB and PCI controller programs, since the USBController and AETest programs do not meet their requirements.

3.1 Dini API

This is a placeholder for the API. It doesn't exist yet.

3.2 USB

The behavior of the DN9000K10PCIE4GL with regard to the USB interface is given in the Hardware chapter.

4 Updating the Firmware

Dini Group may release firmware bug fixes or added features to the DN9000K10PCI. If a firmware update is released you will need to download this new code to the firmware flash of the DN9000K10PCIE4GL. There are three firmware files that Dini Group may release: EEPROM_FLP.iic for EEPROM; firmware.hex for FLASH and prom_flp.mcs for Spartan PROM

The first firmware update is for EEPROM, which stands for Electrically Erasable Programmable Read-Only Memory. The Firmware Mode is booted from here. This firmware is rarely changed. Please consult with us before updating this device.

The second firmware update is Micro Controller (MCU) software that is stored in a flash memory. The User Mode is booted from here. This update can be accomplished easily from the USBController or AETest_USB application.

The third update that may be required is a Spartan FPGA core update. The configuration data for the Spartan FPGA is contained in a Xilinx configuration PROM. This update can be accomplished with the Xilinx JTAG programming or iMPACT programs. This update requires a Xilinx JTAG cable. Either the [Xilinx Platform cable USB](#) (\$199) or the [Xilinx Parallel cable IV](#) (\$125) is appropriate.

If you don't have Xilinx cable and wish to update Spartan prom, you can use USBController or AETest_USB to program it through USB. You have to contact us for *.xsvf file.

When updating the firmware, the "Flash", "PROM" and USBController.exe should all be updated simultaneously, since Dini Group only verifies this code using corresponding versions of each.

4.1 Obtaining the updates

The firmware update files are not posted on the web site. In order to obtain them, you must request them from support@dinigroup.com. You may be required to perform a firmware update to your board to receive support and some features. When updating firmware, you should update in the following order:

- 1) USB Controller.exe (<http://www.dinigroup.com/product/common/USBController.zip>)
- 2) Configuration FPGA PROM firmware
- 3) EEPROM (option)
- 4) MCU Flash

4.2 Updating the Spartan (PROM) firmware

4.2.1 Using JTAG cable (Xilinx products)

Connect a Xilinx Platform USB configuration cable to your computer. When the cable is working properly, but not connected to a JTAG chain, the LED on the cable turns amber. When connected to the DN9000K10PCIE4GL, the LED turns green.

Connect the cable to the "Firmware" header, J16

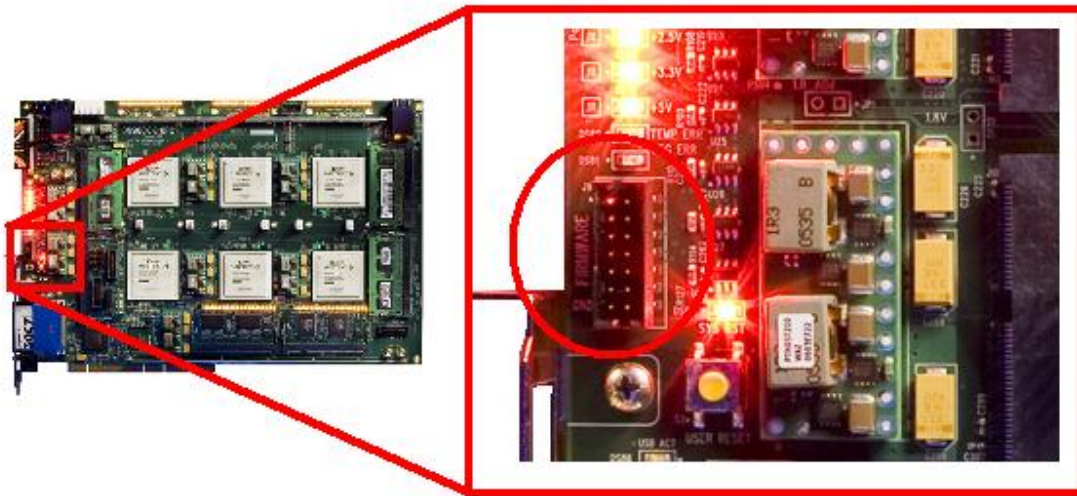


Figure 7 Firmware Update Header

Power on the DN9000K10PCIE4GL. When the Parallel IV cable is connected to a header, the status light turns green.

Open the Xilinx program Impact, usually found at
Start->Programs->Xilinx->ISE->Accessories->impact

iMPACT Project Dialog will appear, please hit Cancel.

Choose the menu option File->Initialize Chain.

Impact should detect 2 devices in the JTAG chain: XC3S1000 and XC18V04. For each item in the chain Impact will direct you to select a programming file for each. For the XC3S1000, Press Bypass. Impact will then ask for a programming file to program the XC18V04 device, select the Spartan Firmware update file provided by Dini Group. This file should be named "prom_flp.mcs". Hit Open.

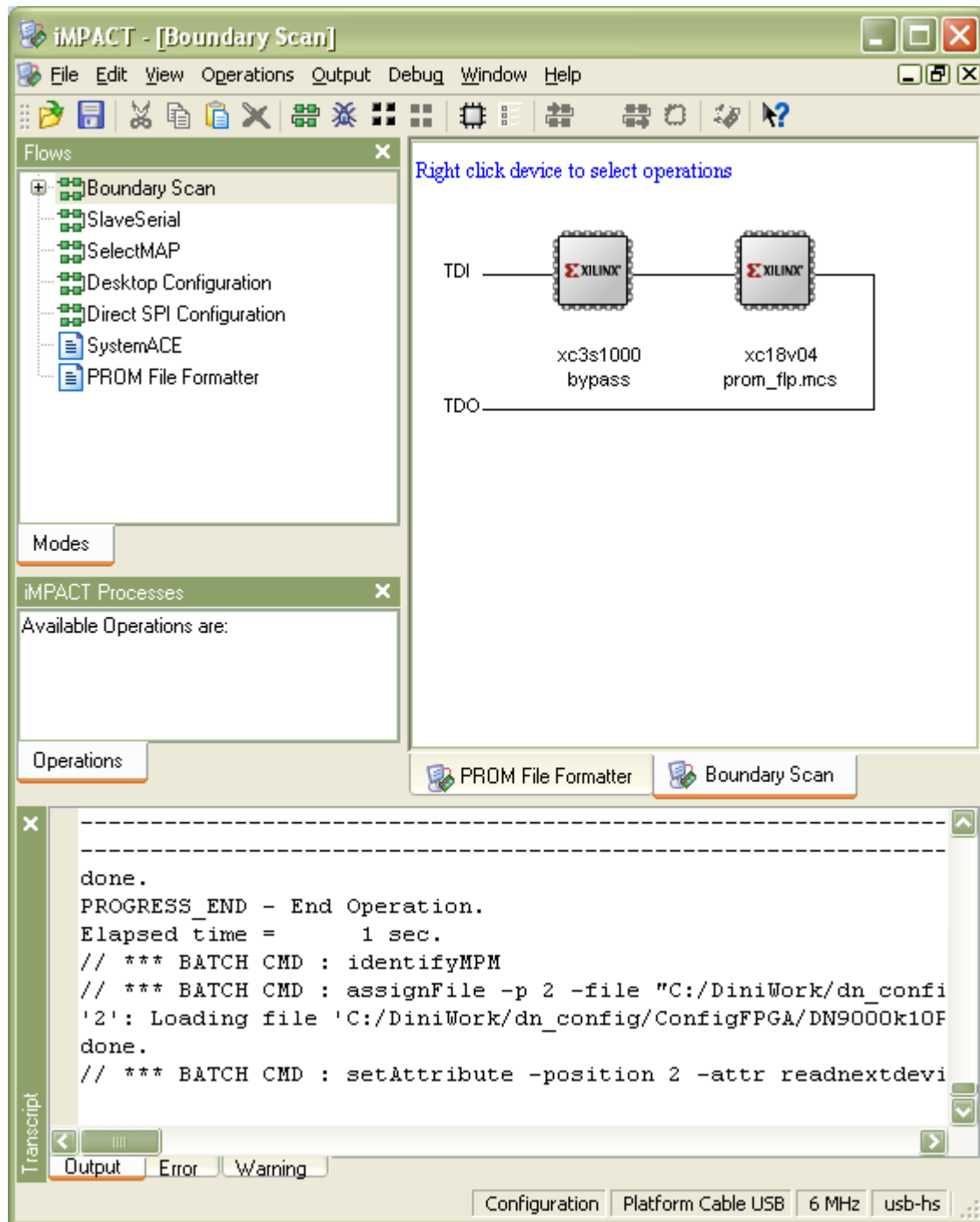


Figure 8 Impact Window

To program the prom, Right-click on the prom and select Program... from the popup menu. In the options dialog that follows, the options “Erase before programming” should be selected, and “Verify” should be selected. Press OK. The programming process takes about 35 seconds over the parallel port, and 5 seconds over USB.

Power cycle the DN9000K10PCIE4GL. The new firmware is now loaded. You can close impact and disconnect the Xilinx JTAG cable.

4.2.2 Using USBController

If you do not have a JTAG cable, you will need to use the following instructions to update your “Spartan PROM” firmware. This update is dependent on USBController and Flash firmware version. Please double check with us (support@dinigroup.com) to make sure that your current version (MCU version, USBController) supports this option and request *.xsvf file from us.

1. Open USBController.ini and add the line “service_mode=1”. You save and close the file.
2. Launch USBController, go to “Service” menu and select “Program/Update Spartan”. A warning message will appear to ensure that you want to update Spartan. If you do, hit “Yes” button.
3. Open file Dialog will appear. Please select the *.xsvf file that we provide you.
4. After selecting file, there will be debug level dialog. Please select debug level: 0
5. The process takes about 10-15 minutes, please leave the board and USBController alone. The process bar is on the bottom of USBController window.
6. When the execution is finished, power cycle the board.

4.2.3 Using AETest_USB

If you do not have a JTAG cable, you will need to use the following instructions to update your “Spartan PROM” firmware. This update is depending on AETest_USB and Flash firmware version. Please double check with us (support@dinigroup.com) to make sure that your current version (MCU version, AETest_USB) supports this option and request *.xsvf file from us.

1. Run aeusb_wdm.exe (or aeusb_linux).
2. At the main menu, please select option 3 “FPGA Configuration Menu”
3. In “Flash Boot Menu”, please select option ‘9’. Note: the option menu is not displayed for security purpose.
4. Please enter the full path filename for the *.xsvf file.
5. Verbose level is ‘0’. The higher verbose level, the slower the program runs.

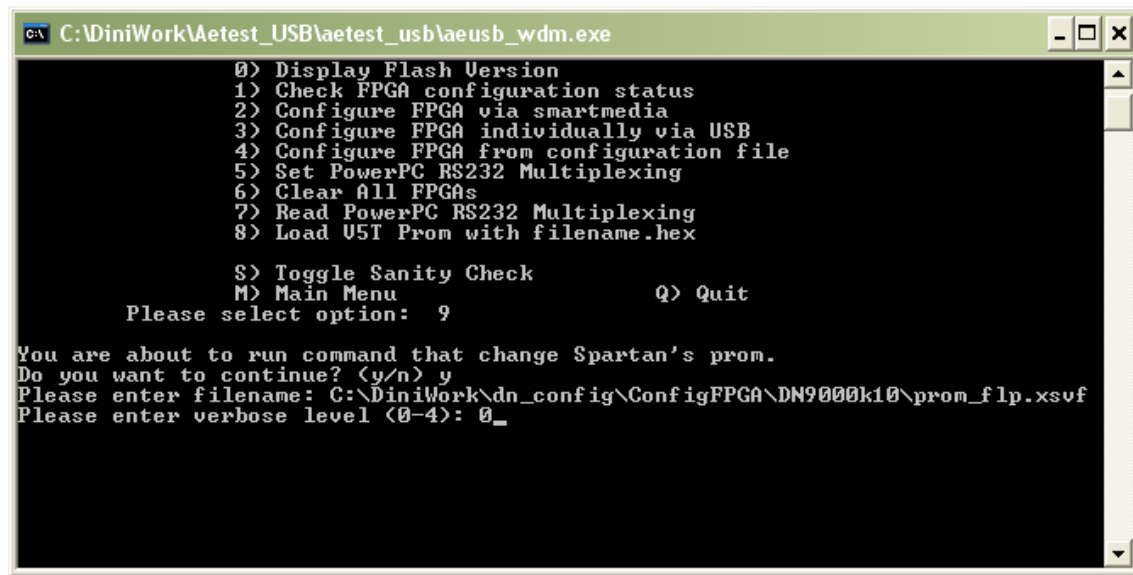


Figure 9 aetest_usb window

6. The progress will start from 0 to 100%. This will take long time to complete (10 minutes). Please do not disturb the process.
7. Power cycle the board when finish.

You can also use commend line: "aeusb_wdm_cmd.exe -XSVF <filename.xsvf>" (or "aeusb_linux_cmd.exe -XSVF <filename.xsvf>").

4.3 Updating EEPROM firmware (not recommend)

To protect against accidental erasure, the EEPROM firmware cannot be updated unless the board is put in firmware update mode during power-on. Find Switch S2 (“User Reset”) on the DN9000K10PCIE4GL.

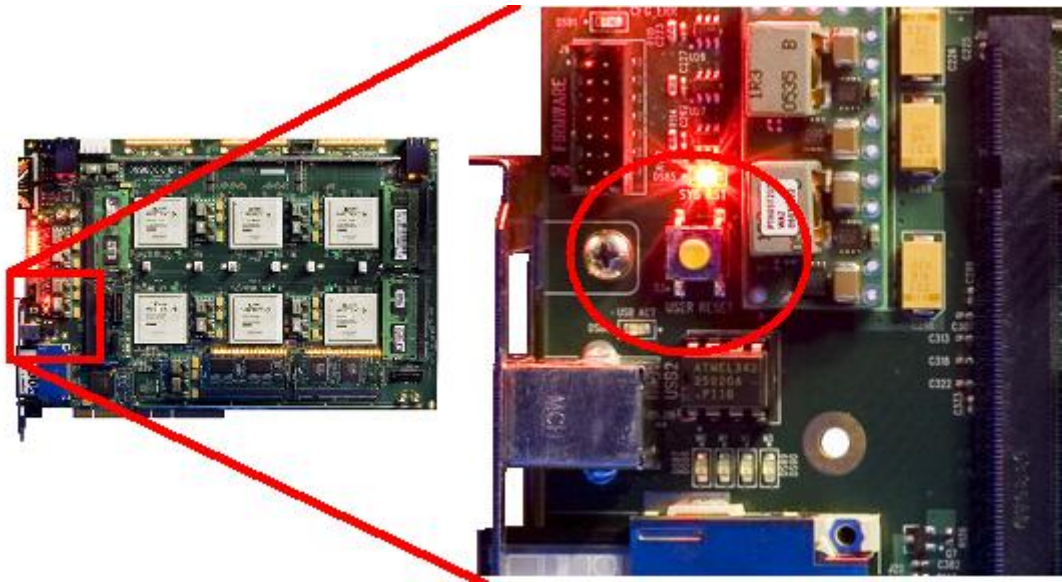


Figure 10 Switch S2

Hold down the “User reset” button while the DN9000K10PCIE4GL powers on. Or alternately, while holding down the “User reset” switch, press the “Hard reset” button. The DN9000K10PCIE4GL samples the user-reset button on power on to enter into firmware update mode.

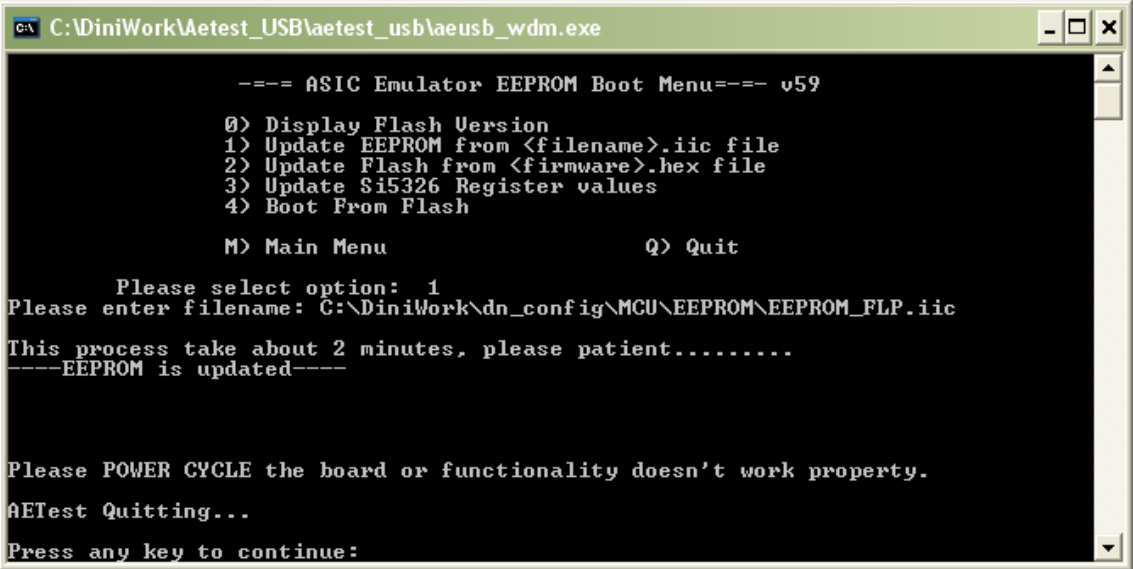
4.3.1 Using USBController

1. Put the board into Firmware Mode (see 4.3)
2. Open USBController.ini and add this line “service_mode=1”, save and close the file.
3. Run USBController. “Update Flash” dialog will appear, please select “NO” because we are doing update EEPROM
4. Go to “Service” menu, select “Program EEPROM”. This Process will take about 1 minute. Please hit OK
5. Select file EEPROM_FLP.iic. When USBController completes the update, please power cycle power the board.

4.3.2 Using AETest_USB

1. Put the board into Firmware Mode (See 4.3)

2. Run aeusb_wdm (aeusb_linux). Select option 3 “Firmware Menu”.
3. In EEPROM Boot Menu, please select option 1 “Update EEPROM from <filename>.iic file”
4. Enter filename (full path). The process should take about 2 minutes



```

C:\DiniWork\Aetest_USB\Aetest_usb\aeusb_wdm.exe

--== ASIC Emulator EEPROM Boot Menu== v59

0) Display Flash Version
1) Update EEPROM from <filename>.iic file
2) Update Flash from <firmware>.hex file
3) Update Si5326 Register values
4) Boot From Flash

M) Main Menu          Q) Quit

Please select option: 1
Please enter filename: C:\DiniWork\dn_config\MCU\EEPROM\EEPROM_FLP.iic
This process take about 2 minutes, please patient.....
----EEPROM is updated----

Please POWER CYCLE the board or functionality doesn't work property.
AETest Quitting...
Press any key to continue:
    
```

Figure 11 aeusb_wdm window

5. Please power cycle the board.

You can also run this on the commend line: aeusb_wdm_cmd.exe -EEPROM <filename.iic>

4.4 Updating the MCU (Flash) firmware

To protect against accidental erasure, the MCU (Flash) firmware cannot be updated unless the board is put in Firmware Mode during power-on (see 4.3). You can either use USBController or AETest_USB program to update MCU (Flash) firmware.

4.4.1 Using USBController

1. Put the board into Firmware Mode (see 4.3)
2. Run USBController.exe, Flash Update dialog will appear, please select “Yes”.
3. Please select firmware.hex (we provide you this file).
4. When finish, please recycle power the board or hit “Hard Reset” (S3) on the board to boot from User Mode.

4.4.2 Using AETest_USB

1. Put the board into Firmware Mode (See 4.3)
2. Run aeusb_wdm. Select option 3 “Firmware Menu”.
3. Please select option 2 “Update Flash from <firmware>.hex”
4. Enter the full path filename. It should be firmware.hex that we provide you
5. The process will take about 2 minutes. When it finishes, please hit “Hard Reset” (S3) on the board or recycle power the board so that DN9000K10PCIE4GL can boot from User Mode.

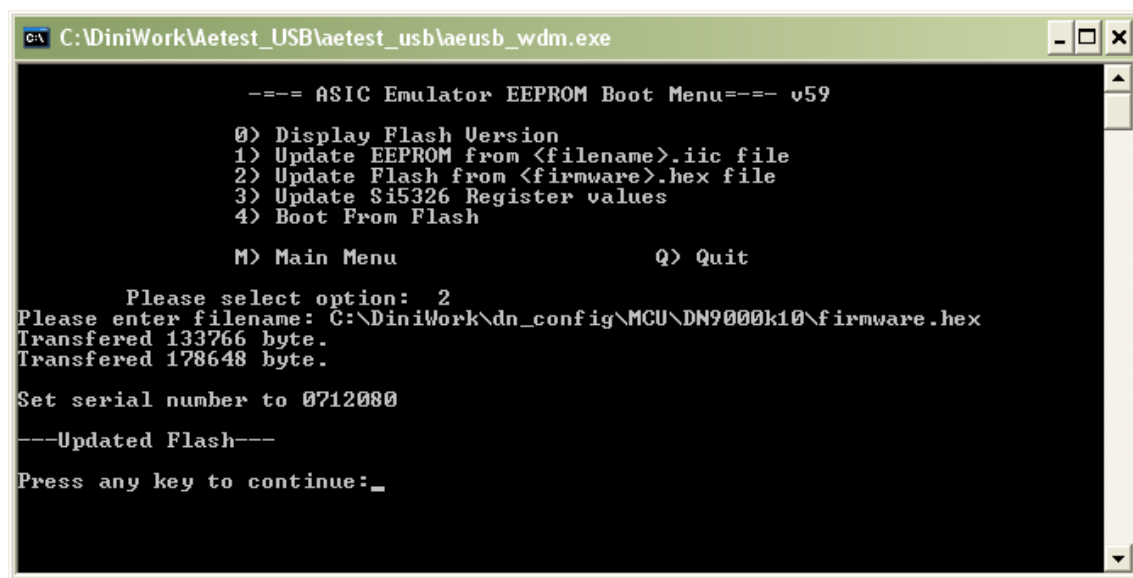


Figure 12 aeusb_wdm window

You can also run this on the command line: “aeusb_wdm_cmd.exe -FLASH <filename.hex>” (aeusb_linux_cmd.exe -FLASH <filename.hex>”).

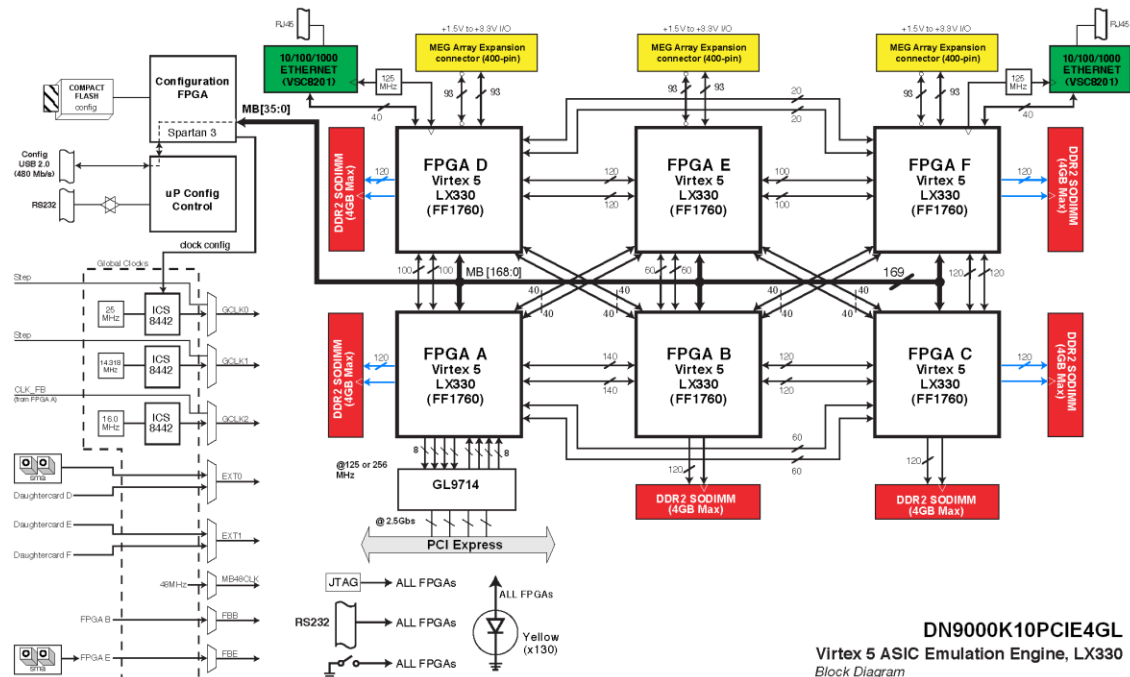
Chapter 4

Hardware

1 General Overview

The DN9000K10PCIE4GL ASIC emulation platform is optimized for providing the maximum amount of interconnect between six Virtex-5 FPGAs. It is the highest-density off-the-shelf development board using the Xilinx Virtex-5 FPGA.

Below is a block diagram of the DN9000K10PCIE4GL.



1.1 Marketing Crap

The following is the advertised feature list of this board. This manual is responsible for providing the information necessary to use these features.

1.1.1 Features

- Prototyping system with two to six Xilinx Virtex-5 FPGAs XC5VLX110, XC5VLX220, or XC5VLX330 (1760-pin package)
- 100% FPGA resources available for user application:
 - Configuration and clocks are controlled off-FPGADDR2, PCI Express and Ethernet do require FPGA cores.
- Nearly 12M ASIC gates (LSI measure) with 6 LX330s
- FPGA-to-FPGA interconnect is single-ended or LVDS
750Mhz LVDS chip to chip (DDR effective clock rate)
- Main Bus (MB) connects all LX FPGAs (169 signals)
Single-ended, 48Mhz
- 6 separate DDR2 SODIMMs
1SODIMM for FPGAs A,B,F,D
2 SODIMMs for FPGAs C
64-bit data width, 250MHz operation
DDR2 modules: PC2-3200/PC2-4200
4GB maximum density per SODIMM (when available)
aggregate data transfer rate: 32Gb/s
- 8 board-level global clock networks (G0, G1, G2, MB48, EXT0, EXT1, FBB, FBE)
Three separate programmable synthesizers (configurable via CompactFlash, USB, PCI)
Global clocks networks distributed differentially and balanced
Two single-step clocking available on each global clock network
5 external differential clock inputs. (Three daughtercard and two SMA cables)
- Flexible customization via daughter cards
Three 400-pin Meg-Array connectors (FCI)
Connected to FPGAs D,E,F
93 LVDS pairs + clocks (or 186 single-ended)
400MHz (DDR so 800Mbs) on all signal pairs
Supported IO Voltages: 1.5V, 1.8V, 2.5V, 3.3V
Power-over-header
- Built-in FPGA configuration
Compact Flash, USB, JTAG
Configuration
Readback supported
- Two tri-mode Ethernet ports

- RS232, Logic Analyzer, LEDs.
Support

1.1.2 Description

The DN9000K10PCIE4GL is a complete logic emulation system that enables ASIC or IP designers a vehicle to prototype logic and memory designs for a fraction of the cost of existing solutions. The DN9000K10PCIE4GL is hosted over USB. A single DN9000K10PCIE4GL configured with 6 Xilinx Virtex-5, XC5VLX330s can emulate up to 12 million gates of logic as measured by LSI (or at least how LSI used to measure ASIC gates when they manufactured ASICs). This number does not include the embedded memories and multipliers resident in each FPGA, all of which are 100% available to user application. The DN9000K10PCIE4GL achieves high gate density and allows for fast target clock frequencies by utilizing FPGAs from Xilinx's Virtex-5 FPGA family for logic and memory. All FPGA resources are available for the target application. Any subset of FPGAs can be stuffed.

The DN9000k10pcie4gl uses high I/O-count, 1760-pin, flip-chip BGA packages. Abundant fixed interconnects (either differential or single-ended) are provided between the FPGAs. All pins of all banks of each FPGA are utilized. FPGA to FPGA busses are routed and tested LVDS, run at 400MHz+ but can be used single-ended at a reduced speed. Example designs utilizing the integrated ISERDES/OSERDES with DDR for pin multiplexing are included. A 160-pin main bus (MB) is connected to all FPGAs including the Spartan configuration FPGA.

Three separate 400-pin FCI MEG-Array connectors allow for customization with daughter cards. Signals to/from these cards are routed differentially, and can run at the limit of the FPGA: 400MHz. Clocks, resets, and presence detection, along with abundant power are included in each connector.

Six separate DDR2 SODIMM sockets are stuffed and have connections to FPGAs A, B, D, F, and C (two separate sets). Each socket is tested to 250MHz with a DDR2 SODIMM. Standard, off-the-shelf DDR2 memory Dims (PC2-3200/PC2-4200) work nicely and we can provide these for a small charge. We have developed alternative SODIMMs that can be stuffed into these positions. Consult the factory for more details, but the list includes FLASH, SSRAM, QDR SSRAM, Mictor and others.

The configuration bit files for the FPGAs are copied onto a CompactFlash card (provided) and an on-board Cypress microprocessor controls the FPGA configuration process. FPGA configuration can also be controlled via the USB interface. Visibility into the configuration process is enhanced with an RS232 port. Sanity checks are performed automatically on the configuration bit files, streamlining the configuration process. FPGA configuration occurs at the fastest possible SelectMap frequency - 48MHz. Multiple LED's provide instant status and operational feedback.

As always, reference material such as a DDR SDRAM controller, code is included (in Verilog, VHDL) at no additional cost.

2 Virtex 5

The DN9000K10PCIE4GL allows use of each of the new features of the Virtex-5 FPGA. As well as exercises all of the external interfaces on the DN9000K10PCIE4GL, the included reference design also exercises all of the new (Compared to Virtex 4) Virtex-5 features listed below.

- Greater speed logic and internal routing speed
- Built-in PLLs
 - Example PLL usage found in the DDR2 reference design
- 1.25 Gbs maximum IO speed (**750Mbs design provided**)
 - LVDS design uses high-speed IO
- ODELAY output signal delay elements
 - LVDS design dynamically adjusts IDELAY to account for interfaces on the DN9000K10PCIE4GL where signals are not externally length-matched (FPGA interconnect)
- 6-input lookup tables
- Larger total-density parts (in terms of total LUT gates)
- More flexible IO

3 Configuration Section

Many functions on the DN9000K10PCIE4GL are done by circuitry on the DN9000K10PCIE4GL external to the FPGA. Collectively, these circuits are referred to in this document as the “Configuration Section”. The configuration section takes care of:

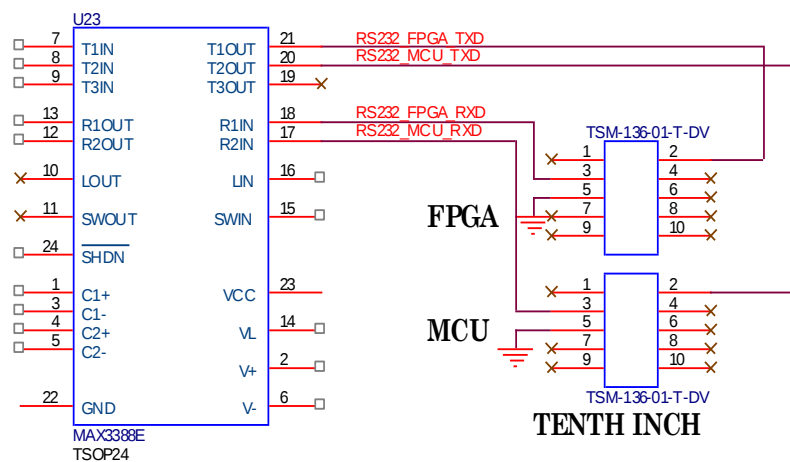
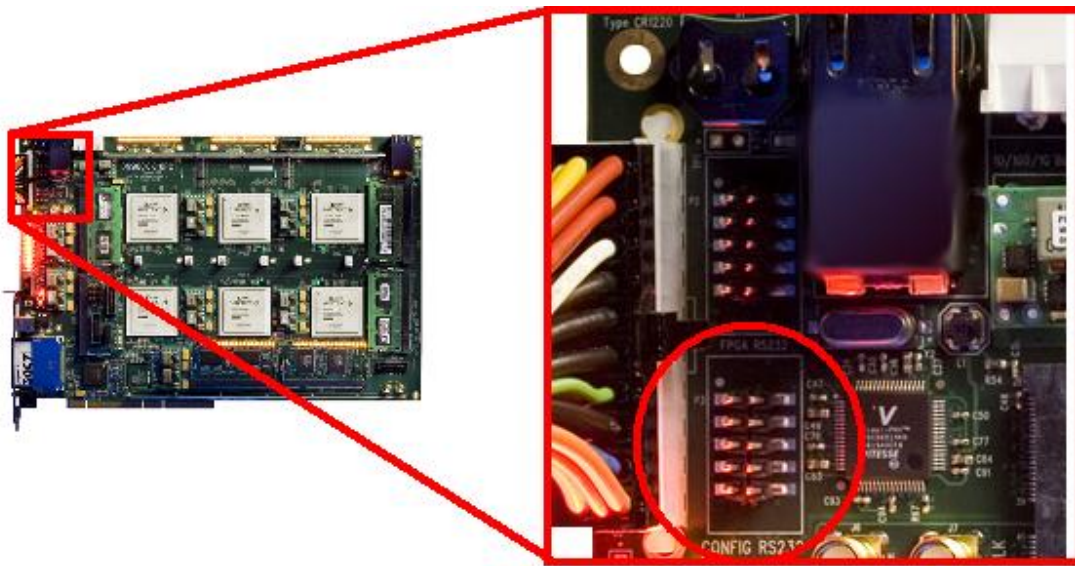
CompactFlash interface
 USB interface
 “Main Bus” interface master
 Temperature sensing
 Over/under voltage sensing
 Clock frequency and source configuration
 SelectMap configuration interface
 Blinking red and green LEDs

The Configuration Section is built around a Spartan 3 FPGA and Cypress microprocessor. These ICs are used by the configuration circuit and are not intended for user design. The code running these controlling ICs is collectively referred to as the “firmware”. The code for this firmware is provided, but I don’t know why. Development on these platforms is not directly supported, and correct function of these devices is only guaranteed as it is specified in this document or otherwise advertised. If you need special configuration section behavior, please contact Dini Group so that we can implement it as a supported feature. The technical details of the Configuration Section are omitted from this manual, since the user should not require it.

3.1 Configuration Section Feedback

During normal operation, and in error situations, the configuration section prints messages to the RS232 terminal header. The configuration section processes that can be monitored using this header are:

- Temperature sensor: FPGA overheat
- CompactFlash card reading
- USB configuration
- PCI configuration
- Main Bus reads/writes
- Fun Facts (like what files are on the CF card)
- Global clock settings



The configuration section RS232 terminal header, labeled “MCU” above, can be connected to a computer serial port, using the settings:

19200 Baud
No flow control
One stop bits
No parity

The content of the RS232 output changes with time as the debugging requirements on this platform are reduced, or when the firmware development team is bored.

3.2 FPGA Configuration

Normally, configuration of the Virtex-5 FPGA occurs over the Virtex-5 “SelectMap” interface. The only configuration method possible on the DN9000K10PCIE4GL that does not use this interface is the JTAG header. For a description of the SelectMap interface, see the Virtex-5 configuration guide.

The signals in the SelectMap bus are listed below

D[15:0]	SelectMap data signals. (DN9000K10PCIE4GL may only use [7:0])
PROGRAM_B	Active low asynchronous reset to the configuration logic. This will cause the FPGA to become un-configured. This documentation refers to this signal as PROG#
DONE	After the FPGA is configured, it is driven high, or tri-stated by the FPGA.
INIT	Low indicates that the FPGA configuration memory is cleared. After configuration, this could indicate an error.
RDWR_B	Active low write enable. This Documentation refers to this signal as RDWR#
BUSY	This signal is not used by the DN9000K10PCIE4GL
CS_B	SelectMap chip select. This documentation refers to this signal as CS#
CCLK	Clock driven to the FPGA by the configuration section. The SelectMap signals D[15:0], RDWR_B and CS_B synchronous to CCLK On this board, CCLK is fixed at 48MHz

USB, CompactFlash and PCI configuration occur over the SelectMap bus. The configuration section makes no modification of the “bit stream” sent to it over PCI or USB. It only copies the data to the SelectMap interface. The “bit stream” must contain all of the SelectMap commands necessary to configure and startup the FPGA. These SelectMap commands are created automatically by Xilinx tool bitgen (part of ISE). Not all of the bitstream generation options available in bitgen are compatible with the DN9000K10PCIE4GL:

Currently, before configuring the FPGA using any method (except JTAG), the configuration section asserts the PROG# signal of the FPGA to clear it. For this reason, the “disable SelectMap” option in bitgen has no effect.

On each FPGA, the DONE signal is connected to a blue LED located next to each FPGA. This signal gives a quick indication of whether each FPGA is configured or not.

The data signals, D[15-0] are dual-purpose signals and can be used as additional interconnect pins after all FPGAs have been configured. Care must be taken that the FPGA design does not drive these signals until after all FPGAs have been configured. The configuration section will assert the FPGA_RESET# signal until this occurs (CompactFlash configuration only). If you do use the SelectMap data signals as interconnect, the provided software (USB Controller) is not guaranteed to function properly (may interfere with your design). When using these signals as interconnect, the appropriate drive standard is LVCMOS25. The IO voltage is 2.5V

SelectMap Readback is possible on the DN9000K10PCIE4GL. This can be accomplished over PCI or USB. The user interface for obtaining this data is not defined. If you need this feature contact the Dini Group. Partial reconfiguration is also supported, however the interface is undefined, so you'll have to contact us so we can define it.

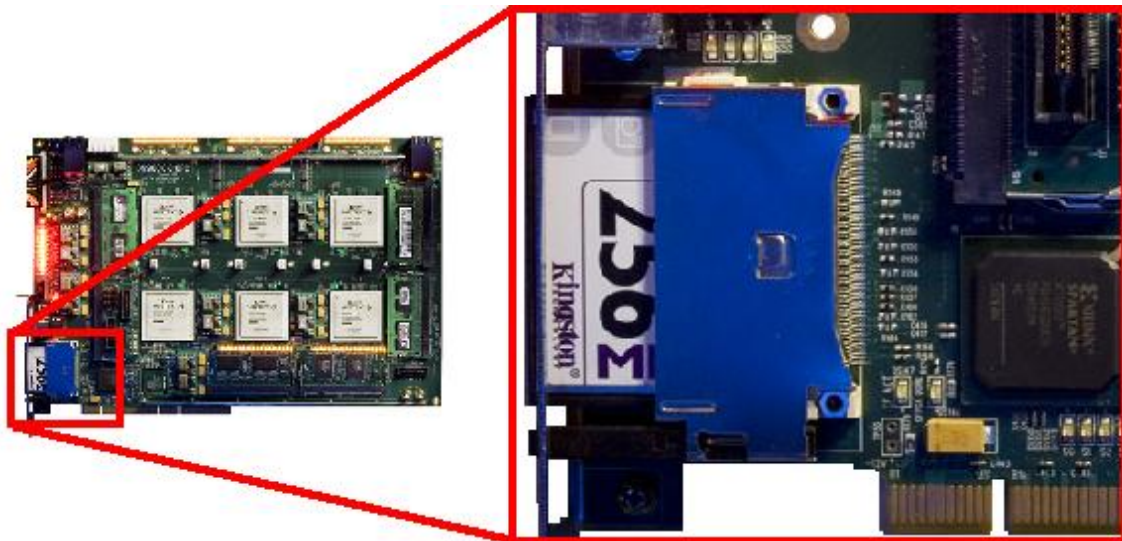
The JTAG configuration method does not go through the configuration circuit, but is connected directly to a header. See the JTAG interface section in the hardware chapter for details about this.

3.3 USB interfaces

The USB interfaces can be used for both configuration (FPGA configuration, and clock settings, etc.) or for direct communication with the user design in the FPGA. These interfaces are described individually in their own sections in the hardware chapter.

3.4 CompactFlash Interface

Most important settings on the DN9000K10PCIE4GL can be controller through the Compact Flash interface. This interface can also be used to configure FPGAs. The CompactFlash interface is not under the direct control of the user, but is accessed only by the configuration logic.



3.4.1 Main.txt

The main.txt interface is the primary method you will use to control settings on the DN9000K10PCIE4GL. From this interface, you can

- Configure FPGAs
- Set clock frequencies
- Write to MainBus

Other settings on the DN9000K10PCIE4GL can also be controlled via the main.txt file by accessing the configuration registers, using the MEMORY MAPPED command. (Basically anything that can be done with USB Controller can also be done from main.txt)

To use the main.txt interface, create a file called main.txt on the root directory of the Compact Flash card. Plug the card into the DN9000K10PCIE4GL. The DN9000K10PCIE4GL will execute commands contained within this file when the board powers on, when the Hard Reset button is pressed, or when instructed to do so by the USB interface (vendor request).

A main.txt file contains a list of commands, separated by newline characters. A list of valid main.txt commands is given below.

```
// <comment>
FPGA A:<filename>
FPGA B:<filename>
FPGA C:<filename>
FPGA D:<filename>
FPGA E:<filename>
FPGA F:<filename>
```

CLOCK FREQUENCY: 0 <number> [MHz]
CLOCK FREQUENCY: 1 <number> [MHz]
CLOCK FREQUENCY: 2 <number> [MHz]
SANITY CHECK: <yn>
VERBOSE LEVEL: <level>
MEMORY MAPPED: 0x<SHORTADDR> 0x<BYTE>
MAIN BUS 0x<WORDADDR> 0x<WORDDATA>

<comment> can be any string of characters except for newline.

<filename> can be the name of a file on the root directory of the CompactFlash Card.

<number> can be any positive number in decimal. Decimal points are allowed.

<yn> can be the letter y or the letter n

<level> can be 0,1,2 or 3

<SHORTADDR> is a 2-digit number in hexadecimal (16 bits)

<BYTE> is a 1-digit number in hexadecimal (8 bits)

<WORDADDR> 4-digit (32 bit) number in hexadecimal representing a main bus address

<WORDDATA> 4-digit (32 bit) number in hexadecimal containing data for a main bus transaction

The following table describes the function of each of the available main.txt commands.

Instruction	Function
// <comment>	The configuration circuitry performs no operation and moves to the next command.
VERBOSE LEVEL: <level>	This command will set the amount of output that will be produced over the RS232 port during configuration. When level is set to 0, the port will produce only error output.
FPGA A:<filename>	The Virtex 5 FPGA "A" will be configured with the file named by <filename>
FPGA B:<filename>	The Virtex 5 FPGA "B" will be configured with the file named by <filename>
FPGA C:<filename>	The Virtex 5 FPGA "C" will be configured with the file named by <filename>

HARDWARE

SANTITY CHECK: <yn>	If <yn> is set to y, then the MCU will examine the headers in the .bit files on the CompactFlash card before using them to configure each FPGA. If the target FPGA annotated in the .bit file header is not the same type as the FPGA the MCU detects on the board, it will reject the file and flash the error LED. Before this command is executed, <yn> is set to the default value y. If you want to encrypt or compress your bit files, you will need to set <yn> to n.
MAIN BUS 0x<WORDADDR> 0x<WORDDATA>	Writes data in <WORDDATA> to the address on the main bus interface at <WORDADDR>. This command only makes sense in the context of the Dini Group reference design, unless your design implements a compatible controller on the main bus pins. The Specification for this interface is in MainBus section
MEMORY MAPPED: 0x<SHORTADDR> 0x<BYTE>	Writes to a configuration Register. This command can be used to access features that do not have a main.txt command. Example applications include setting clock sources, settings the EXT0 or EXT1 clock buffers to zero-delay mode, or setting the clocks to frequencies lower than 31Mhz.
RS232: <port> <FPGA>	The RS232 port (P1) will be controlled by the FPGA <FPGA> if <port> is 1
CLOCK FREQUENCY: <clockname> <number> MHz	The MCU will adjust the clock synthesizer producing clock <clockname> to the frequency <number>. Currently, only frequency settings from 31Mhz to 700Mhz are supported. Contact support@dinigroup.com for frequencies from 2 to 31Mhz.

An example main.txt file is given below.

```
VERBOSE LEVEL:0
// This will prevent the MCU output over RS232 to speed up configuration
FPGA A:a.bit
//this will load the configuration a.bit into FPGA A
CLOCK FREQUENCY: G0 300Mhz
MAIN BUS: 0x08000000 0x00000001
//Writes to a register in FPGA A.
```

Even if you are not planning to configure your Virtex 5 FPGAs using a CompactFlash card, you may want to leave a CompactFlash card in the socket to automatically program your global

clock. (Clocks may also be programmed using the provided USB application, or over the MCU RS232 terminal.)

3.4.2 Limitations

Directories are not supported for FPGA configuration files. All bit files must be contained in the root directory of the card. File name lengths for bit files are limited to 8 characters. You can still use bit files with longer file names by specifying the DOS version of the file name in the main.txt file. For example, to configure FPGA A with the file

Com_apple_sj45_ethn0_fpga_a_121107b.bit

You can use the main.txt command

FPGA A: com_ap~1.bit

The FAT32 file system is not supported. Most OTS CF cards now-a-days come pre-formatted in FAT32.

3.4.3 Hardware

The Compact Flash interface is hot-swappable.

An activity LED, DS147, located next to the Compact Flash slot indicates activity on this interface.

Please contact support@dinigroup.com if you find an incompatible card, so that we can add software support for it. (Some CF cards, particularly ones marketed “for use in a <brand name> camera” come with “features” that prevent their use in a DN9000K10PCIE4GL)

3.5 Configuration Registers

The configuration control on the DN9000K10PCIE4GL is controlled by setting “configuration registers”. Basically, these are just locations in the memory space of the on-board micro controller that controls the board’s function. A full description of the function of this micro controller is omitted, but some of the registers in this space are required to be accessed over USB or PCI to control the board. For information on how to access this address space over USB or PCI, see the corresponding section in this chapter.

REGISTER	ADDRESS	FUNCTION
FPGA_RESET	DF22	Write 0x2 to hold reset. 0x0 to release
G0_N_VAL	DF29	Sets the divider value of G0
G0_M_VAL	DF30	Sets the 8442 multiplier of G0
G1_N_VAL	DF31	Sets the 8442 divider value of G1
G0_M_VAL	DF32	Sets the 8442 multiplier of G1
G2_N_VAL	DF33	Sets the 8442 divider value of G2
G2_M_VAL	DF34	Sets the 8442 multiplier of G2
UPDATE_CLOCK_FLAG	DF40	When high, each bit causes the configuration circuit to update the represented clock frequency with the

HARDWARE

		current M and N values. 0x01 is G0, 0x02 is G1, 0x04 is G2
FPGA_COMMUNICATION	DF39	Disables Main Bus interface
TEMP_SENSOR_A	DF50	Temperature of FPGA A
TEMP_SENSOR_B	DF51	Temperature of FPGA B
TEMP_SENSOR_C	DF52	Temperature of FPGA C
TEMP_SENSOR_D	DF53	Temperature of FPGA D
TEMP_SENSOR_E	DF54	Temperature of FPGA E
TEMP_SENSOR_F	DF55	Temperature of FPGA F
SERIAL_NUMBER	DFFA	
BOARD_TYPE	DFFE	

These registers are accessible using the main.txt “memory mapped” command, and over USB using the “config register” vendor request (or menu option in USB Controller)

3.5.1 Undocumented controls

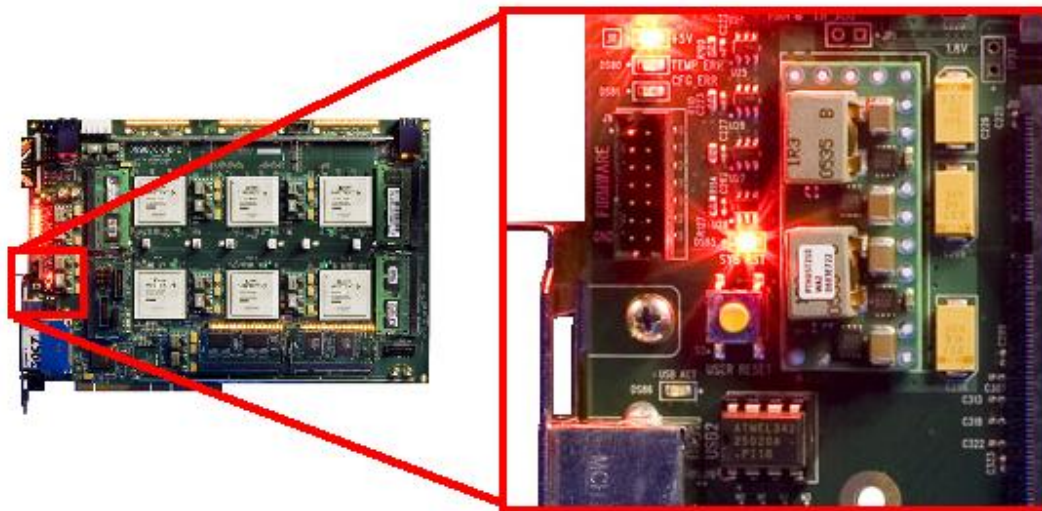
Most of the accessible registers to control board function (used by the AETest_usb and USB Controller programs) are not documented in the table above. This is because we do not anticipate a need for customer use. If there are board features that are accessible through USB Controller or AETEST programs that you feel you need access to in your own PCI or USB applications, contact support@dinigroup.com and we will provide details on using the interface you require. A list of these features is provided below

G0, G1, G2 frequencies below 31 MHz
Single-step clocking on G0, G1, G2 clock networks.
Zero-Delay Daughtercard clock network
External clock source selection
MainBus error counter

3.6 Firmware

A Spartan 3 FPGA and a Cypress micro controller control the configuration circuitry. The programming data for the FPGA is stored on a flash device, and the code for the micro controller is stored on a separate flash device. The instructions for updating the firmware are given in the software section. The flash that stores the Spartan FPGA programming information is made available via a JTAG header, which can be used with the Xilinx program impact. The Dini Group does not recommend doing any sort of development on this FPGA, because if you add custom code, you will not be able to use firmware updates from Dini Group without merging it with your custom code.

The code for the firmware is provided on the user CD in case you really want to mess with it, or if Dini Group is destroyed in a meteor strike and can no longer maintain it.



4 Clock Network

4.1.1 GC Pins

When this manual refers to a “clock input” of an FPGA, it means the “GC” pin described in the Virtex-5 user manual. These pins have the capability of driving a DCM, PLL, or BUFG input with a known (accounted for) delay within the FPGA. All “clock” signals on the DN9000K10PCIE4GL are connected to “GC” pins on the FPGA.

4.2 Global Clocks

All of the “global clock networks” on the DN9000K10PCIE4GL are LVDS, point-to-point signals. The arrival times of the clock edges at each FPGA are phase-aligned (length-matched on the PCB) within about 100ps. These clocks are all suitable for synchronous communication among FPGAs.

Since LVDS is a very low voltage-swing differential signal, you cannot receive these signals without using a differential input buffer. Single-ended inputs will not work. An example Verilog implementation of a differential clock input is given below.

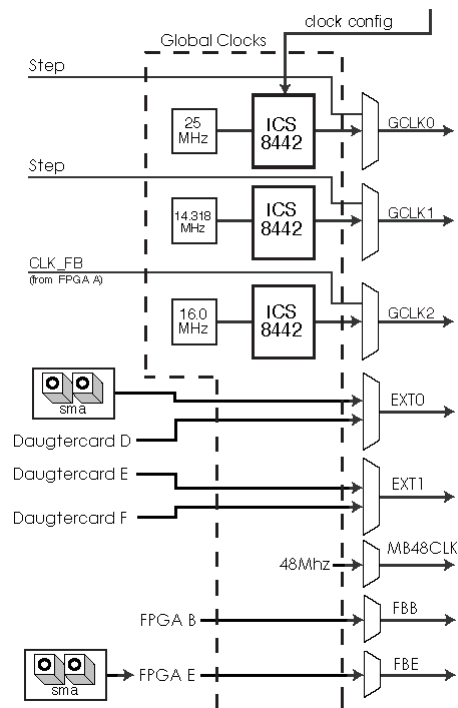
```
Wire aclk_ibufds;
IBUFGDS G0CLK_IBUFG
    (.O(g0clk_ibufg), .I(GCLK0p), .IB(GCLK0n)) ;

always@(g0clk_ibufg) begin
    // Registers
end
```

HARDWARE

Either in the UCF or using a synthesis directive, you should turn the DIFF_TERM attribute of the IBUFGDS to TRUE. This is recommended because there are no external termination resistors on the DN9000K10PCIE4GL.

All global clock networks have a differential test point. The positive side of the differential signal is connected to pin 1 (square) and the negative side is connected to pin 2 (circular).



A diagram of the global clock network is shown above. Each of the eight clock outputs of the clock network is distributed to all six FPGAs.

Some people don't understand some parts of this diagram, so here is a paragraph clarifying some things:

Each clock (GCLK0, GCLK1, GCLK2, EXT0, EXT1, MB48, FBB, FBE) arrives at all six FPGA simultaneously, so these clocks can be used for synchronous communication among FPGAs.

FPGA A can be set as the source of clock GCLK2. FPGA A can therefore be used to do frequency synthesis or anything else.

FPGA B drives the FBB clock, so FPGA B can be used to synthesize a frequency or whatever you want.

FPGA E drives the FBE clock, so FPGA E can be used to synthesize a frequency or whatever you want.

Two SMA connectors are attached to FPGA E. If you wanted to, you could pipe that clock through FPGA E to the FBE network, making those SMAs one more available external clock input.

The clock MB48 is fixed at 48Mhz, and cannot be changed ever ever.

4.3 G0, G1, G2 Clocks

The G0, G1 and G2 clocks are the primary clock resource for your FPGA design. Each of these clocks can be set to a wide range of frequencies between 2 and 550 MHz.

On the schematic, these signals are named

CLK_G*_*_n

where x is 0,1 or 2 and N is the name of the FPGA connected to that signal.

The possible source of G0 and G1 clock is either the ICS8442 frequency synthesizer or a “step clock”. The “step clock” is driven by the configuration circuit and can be toggled over USB by writing to the correct configuration register. 0xDF23[2], 0xDF23[1]. Before the Synthesizer or step clock drives the network, the correct source setting must be made in the GUI or in the main.txt file. By default the source is the synthesizer.

The syntax to set the clock source in the main.txt file is

(contact support@dinigroup.com. This syntax does not exist yet)

In USB Controller, from the settings menu, select DN9000K10PCIE4GL clock source settings

The possible sources of the G2 clock are the synthesizer, or FPGA A (using the CLK_FBA_INT signal. FPGA drives this 3.3V, LVCMOS clock signal. From the USB Controller program you can select CLK_FBA_INT from the settings->DN9000K10PCIE4GL clock source dialog. This setting can also be achieved over PCI or USB, but the interface is not documented here, so you must contact support if you require this function.

4.3.1 Clock Synthesizers

The G0, G1, and G2 clock synthesis source is driven by an ICS8442 clock synthesizer chip. This chip is capable of driving a wide range of output frequencies. If you use the USB Controller, CompactFlash card, or AETest program to set the clocks, you do not need to understand how the 8442 synthesizer works. The interface for USB or PCI to set the synthesizer frequencies directly controls the PLL unit within the ICS8442 synthesizer. The PLL only accepts a limited range of inputs. If you are creating your own USB or PCI software, you need to read the 8442 datasheet provided on the user CD.

In summary, the multiplication and division factors of the synthesizer need to be set.

4.3.2 Possible Outputs

Only some multiplication and division settings are possible on the clock synthesizers described above. (The synthesizer can only do integer multiplication, and power-of-two division).

HARDWARE

Limited resolution. When you command the DN9000K10PCIE4GL to set a global clock frequency, it automatically selects the closest frequency possible from the following lists.

G0, synthesized from a 25.0 MHz crystal

3.13, 6.25, 12.50, 25.00, 31.25, 34.38, 37.50, 40.63, 43.75, 46.88, 50.00, 53.13, 56.25, 59.38, 62.50, 65.63, 68.75, 71.88, 75.00, 78.13, 81.25, 84.38, 87.50, 93.75, 100.00, 106.25, 112.50, 118.75, 125.00, 131.25, 137.50, 143.75, 150.00, 156.25, 162.50, 168.75, 175.00, 187.50, 200.00, 212.50, 225.00, 237.50, 250.00, 262.50, 275.00, 287.50, 300.00, 312.50, 325.00, 337.50, 350.00, 375.00, 400.00, 425.00, 450.00, 475.00, 500.00, 525.00, 550.00

G1, Synthesized from a 14.318 MHz crystal

1.79, 3.58, 7.16, 14.32, 32.22, 34.01, 35.80, 37.58, 39.37, 41.16, 42.95, 44.74, 46.53, 48.32, 50.11, 51.90, 53.69, 55.48, 57.27, 59.06, 60.85, 62.64, 64.43, 66.22, 68.01, 69.80, 71.59, 73.38, 75.17, 76.96, 78.75, 80.54, 82.33, 84.12, 85.91, 89.49, 93.07, 96.65, 100.23, 103.81, 107.39, 110.96, 114.54, 118.12, 121.70, 125.28, 128.86, 132.44, 136.02, 139.60, 143.18, 146.76, 150.34, 153.92, 157.50, 161.08, 164.66, 168.24, 171.82, 178.98, 186.13, 193.29, 200.45, 207.61, 214.77, 221.93, 229.09, 236.25, 243.41, 250.57, 257.72, 264.88, 272.04, 279.20, 286.36, 293.52, 300.68, 307.84, 315.00, 322.16, 329.31, 336.47, 343.63, 357.95, 372.27, 386.59, 400.90, 415.22, 429.54, 443.86, 458.18, 472.49, 486.81, 501.13, 515.45, 529.77, 544.08

G2, Synthesized from a 16.0 MHz crystal

2.00, 4.00, 8.00, 16.00, 32.00, 34.00, 36.00, 38.00, 40.00, 42.00, 44.00, 46.00, 48.00, 50.00, 52.00, 54.00, 56.00, 58.00, 60.00, 62.00, 64.00, 66.00, 68.00, 70.00, 72.00, 74.00, 76.00, 78.00, 80.00, 82.00, 84.00, 86.00, 88.00, 92.00, 96.00, 100.00, 104.00, 108.00, 112.00, 116.00, 120.00, 124.00, 128.00, 132.00, 136.00, 140.00, 144.00, 148.00, 152.00, 156.00, 160.00, 164.00, 168.00, 172.00, 176.00, 184.00, 192.00, 200.00, 208.00, 216.00, 224.00, 232.00, 240.00, 248.00, 256.00, 264.00, 272.00, 280.00, 288.00, 296.00, 304.00, 312.00, 320.00, 328.00, 336.00, 344.00, 352.00, 368.00, 384.00, 400.00, 416.00, 432.00, 448.00, 464.00, 480.00, 496.00, 512.00, 528.00, 544.00

If there are some unusual requirements for specific frequencies, the crystal frequencies can be changed at the factory.

4.3.3 Duty Cycle

The G0, G1, and G2 clocks only have a 50% duty-cycle when set to frequencies below 350Mhz. Above this frequency the duty cycle is not guaranteed. Experimentally, it is shown to be better than 60/40 at high frequencies.

4.4 Ext Clocks

There are two clock networks on the DN9000K10PCIE4GL that are designed to provide clocks from an external frequency reference. EXT0 and EXT1. Each of these clocks is delivered synchronously to all 6 FPGAs and is suitable for synchronous communication among the FPGAs.

4.4.1 EXT0

This clock can be sourced from either the external clock input SMA connectors or daughtercard D.

By default, EXT0 is set to be sourced from the daughtercard D. The source setting can be made from the USB Controller by selecting settings->DN9000K10PCIE4GL clock source.

The syntax for setting this clock from the main.txt file is not yet specified. Contact support@dinigroup.com

If you are writing your own USB or PCI software, the method of changing the source is by writing to a configuration register. To change the current setting write to configuration register 0xDF27.

0xDF28[4:0] = S23, S1, S0, PLLSEL, CLKSEL

To understand the correct settings, you must read the ICS8745B datasheet on the user CD.

When the USB controller or main.txt file set the clock setting to daughtercard or SMA mode, they set the clock buffer on the DN9000K10PCIE4GL to bypass mode, which means that the skew between the SMA or daughtercard source, and the receiver FPGA is unknown. If you must use this clock for synchronous communication across the daughtercard, then the buffer must be set into zero-delay mode, by setting the S23, S1, S0 and PLLSEL signals to the correct value. The GUI does not currently support this. Contact support if you require this behavior.

4.4.2 EXT1

This clock can be sourced from either daughtercard E or daughtercard F.

By default, EXT1 is set to be sourced from the daughtercard F. The source setting can be made from the USB Controller by selecting settings->DN9000K10PCIE4GL clock source.

The syntax for setting this clock from the main.txt file is not yet specified. Contact support@dinigroup.com

If you are writing your own USB or PCI software, the method of changing the source is by writing to a configuration register. To change the current setting write to configuration register 0xDF27.

0xDF28[4:0] = S23, S1, S0, PLLSEL, CLKSEL

To understand the correct settings, you must read the ICS8745B datasheet on the user CD.

4.4.3 Daughtercard zero-delay mode

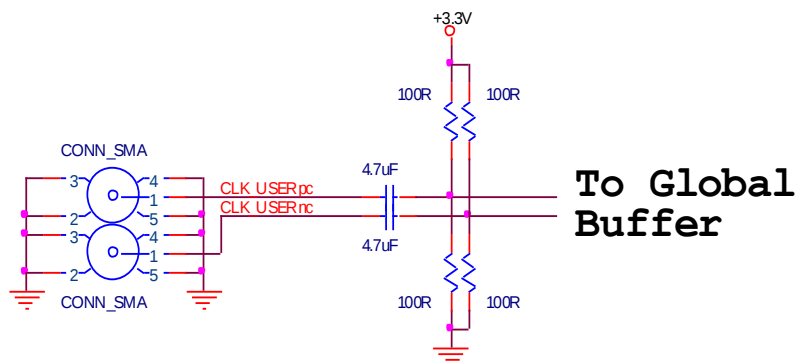
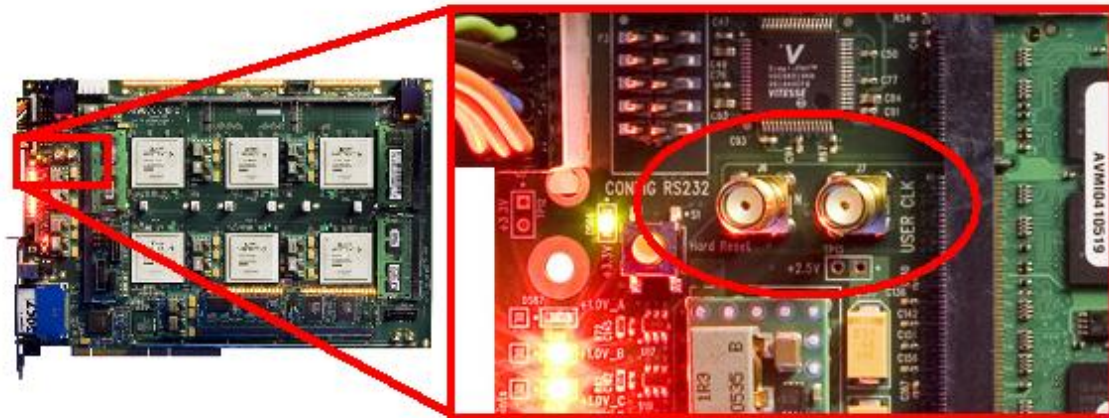
EXT0 and EXT1 can be set to zero-delay mode, where each FPGA is able to receive the clock synchronous to the daughtercard. This feature requires configuring the clock distribution network with the frequency of the clock. The interface by which the user can do this is not defined. Contact support@dinigroup.com if you require this feature.

HARDWARE

You probably don't want to do this though, instead you will probably use the CLKA and CLKB signals on the daughtercard interface, which directly connect the GC pins of the FPGA, and not to the global clock network.

4.4.4 SMA input

The EXT0 clock can be sourced from a pair of SMA inputs, J6, J7. These SMA connectors are designed to connect to a differential clock source.



The inputs are AC-coupled. This limits the minimum possible frequency of the clock input to around 4 kHz. If you require an external clock with a frequency lower than this, you should modify the board by removing the 4.7uF resistors shown above and replacing them with 0-ohm resistors. If you do this you should pay attention to the electrical requirements of the input buffer. The maximum recommended swing on the differential inputs is 3.3V.

To connect a single-ended clock source, you can connect to one of the SMA connectors, and leave the other unconnected.

4.5 FB Clocks

FPGA E and FPGA B each have an LVDS output connected to a global clock input pin of each FPGA. This connection is intended for the purpose of controlling an application-specific clocking requirement from within the FPGA. These networks can also be used to forward a clock, only available to one FPGA.

The clocks are designed to be used as LVDS clocks. If you are using this clock for synchronous communication, remember to drive a clock out of FPGA B back to itself, so that FPGA B receives its own clock synchronous to the other 5 FPGAs. All of these signals are length-matched.

The receiver of these clocks should use DIFF_TERM attribute on the input buffer. The name of these signals on the schematic is CLK_FBE_*p/CLK_FBE_*n.

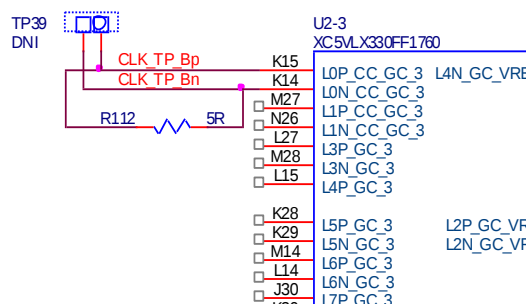
Note that these clock outputs include one feedback signal the is driven out of the FPGA back into the FPGA on a different (“GC”) pin. This clock input pin should be used for synchronous logic on that FPGA.

4.6 Non-Global Clocks

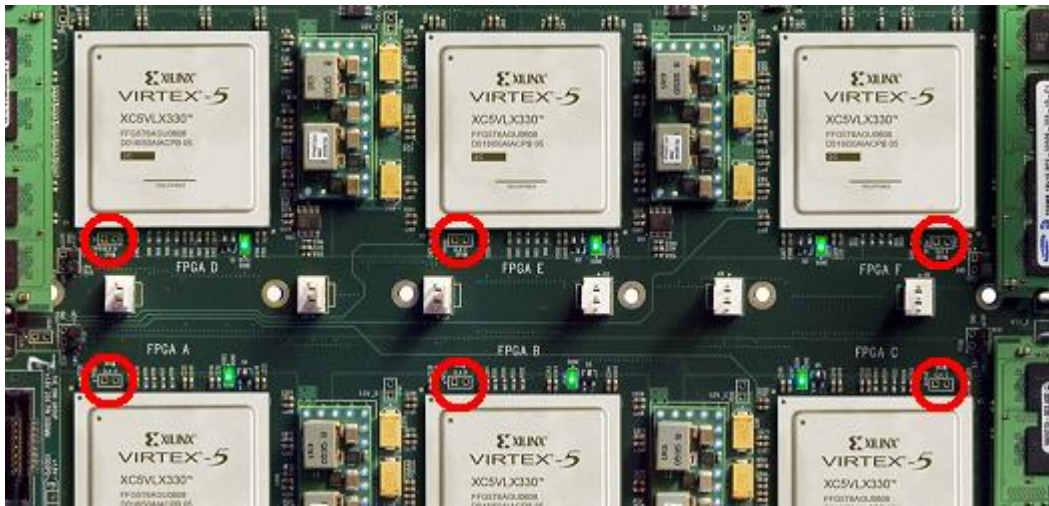
The following sections describe clocks that are not considered “global” because they do not distribute to all six FPGAs on the board. These clocks may be used for specific interfaces and details on the clocking required for those interfaces are found in a different section in the hardware chapter.

4.6.1 Clock TP

Each FPGA is connected to a two-pinned test point. This test point can be used to input a differential clock from off-board. Each of these test points has a 0-Ohm jumper installed shorting the negative and positive signals. To input or output differentially, you must remove this resistor.



The schematic clipping above shows FPGA B's test point, but all FPGAs use the same pinout. A list of all test points on the board can be found in the test points section.



This signal can also be used as an external feedback path for a DCM. When connecting the output of a DCM to K14, the DCM FB input can be connected to K15. Using this configuration, output flip-flops connected to CLK0 of the DCM will have an effective clock-to-out time of less than zero.

Test points for FPGA B, C, D, E and F are connected to “2.5V” banks, meaning they are compatible with signaling up to 2.5V (including LVDS, LVC MOS25, SSTL25, DIFF_SSTL18). FPGA A is on a “3.3V” bank (allowing LVDS input, LVC MOS33, SSTL25 input, SSTL18 input)

4.6.2 Ethernet Clocks

Each of the two VSC8601 Ethernet PHY devices outputs a 125Mhz clock. The signals in the schematic are CLK125_ETHD and CLK125_ETHF. These signals are LVC MOS25, single-ended signals. The frequency is fixed. Details about appropriate clock methodology for the Ethernet interfaces is in the Ethernet section.

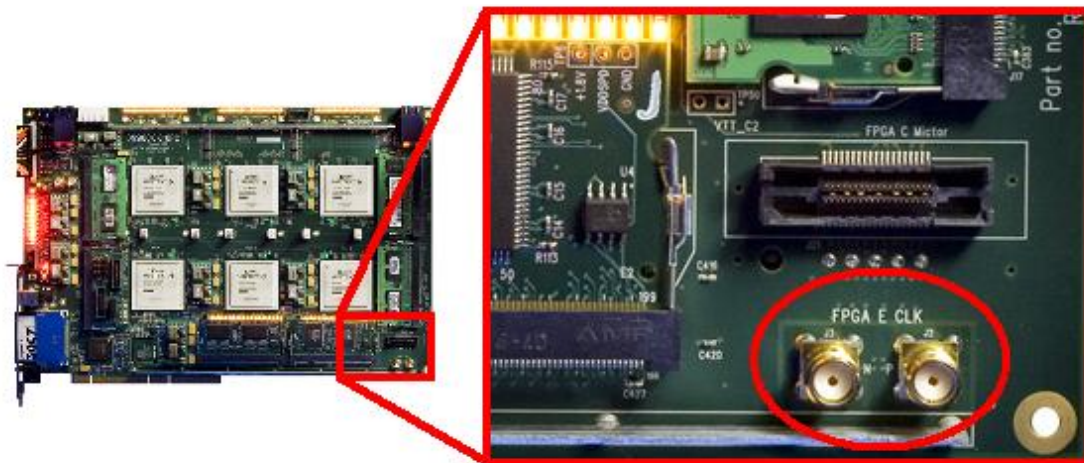
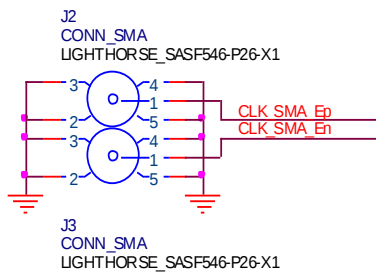
4.6.3 DDR2 Clocks

The CK signals in the DDR2 interface are described in the DDR2 interface section. Note that on the netlist, these signals connect to the FPGA twice: once on the DDR2 interface bank (1.8V), and once on the global clock input bank (2.5V, pins AM13, AM14. FPGA C also has a second pair connected to AP13, AN13). The 2.5V, clock bank connections should be used as inputs, and the 1.8V bank signals should be configured as outputs. For input signals, use the LVDSEXT standard with the DIFF_TERM attribute set to TRUE.

4.6.4 SMA Clock E

FPGA E has a pair of SMA connector connected directly to global clock inputs (AM13, AM14). The bank connected to these signals is a +2.5V bank. Allowed input standards are LVC MOS25, SSTL25, LVDS, DIFF_SSTL18.

HARDWARE



If you need access to this clock globally, you can use the “FB” clock networks described in the “Global Clocks” section.

These connections are DC-coupled, meaning the user must ensure that the levels received on this input are within the limits of the Virtex-5 device to prevent damage to the part.

This pair of SMA connectors can also be used as outputs, or for non-clock signals.

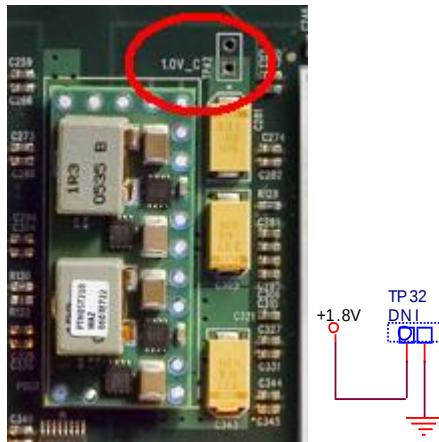
5 Test points

This section lists all of the test points on the DN9000K10PCIE4GL. A more detailed description may be found in the section about the system that the test point is part of, but all test points are listed here for reference.

5.1 Power Thru-hole

Each power rail requiring more than 100mA on the DN9000K10PCIE4GL has a dedicated test point associated with it. This test point is a through-hole, two-pin location, where pin one is the power rail, and pin two is a ground connection. These test point locations are suitable for supplying at least 2A, regardless of the power requirements or capabilities of the power net.

HARDWARE



Pin one is a square. Pin two is circular.

TP33	VTT_A
TP48	VTT_B
TP49	VTT_C1
TP50	VTT_C2
TP9	VTT_D
TP30	VTT_F

TP41	1.0V_B
TP43	1.0V_A
TP2	1.0V_D
TP42	1.0V_C
TP10	1.0V_E
TP11	1.0V_F

TP13	2.5V
TP32	1.8V
TP52	1.2V_S

TP53	-12V
TP1	+5.0V
TP12	+3.3V

Power for the -12V, +5.0V and +3.3V nets are generated off-board.

These test-points are suitable for wiring to if power is needed off-board for some reason. Or maybe if you need to bring power in from an external source.

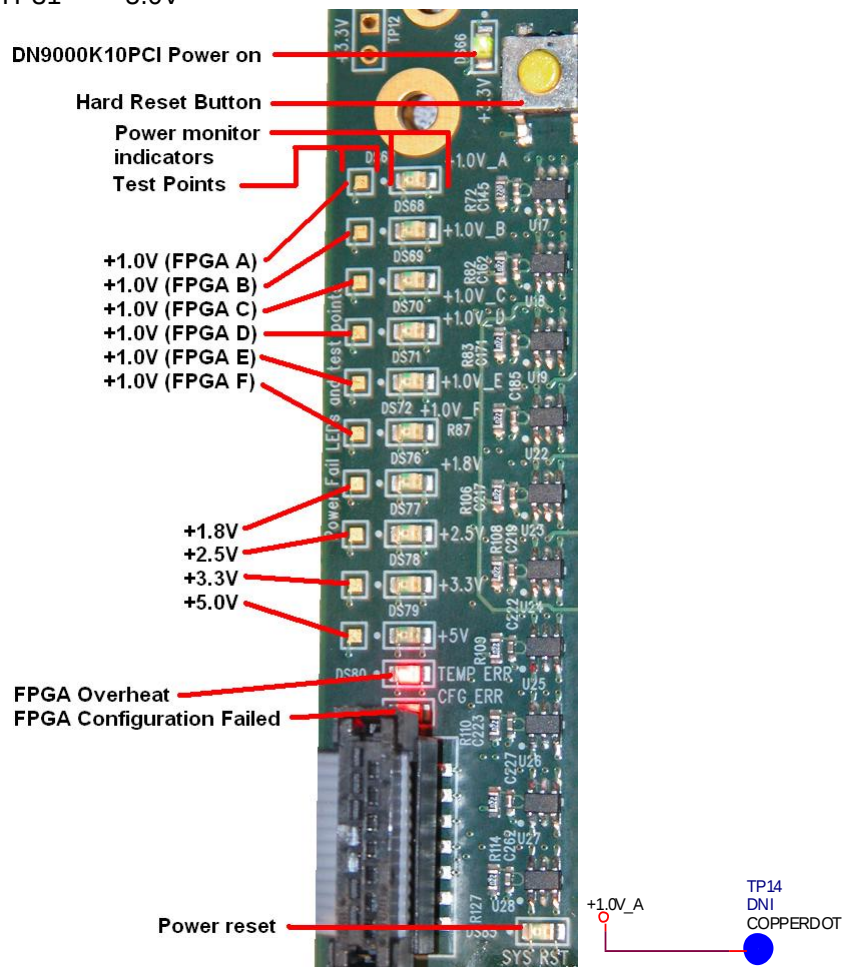
5.2 Power TP

The following test-points are located along the left edge of the board, next to an LED associated with that power net. These test points are square pads. They are not suitable for supplying power to the board, or off the board.

HARDWARE

TP14 1.0V_A
 TP15 1.0V_B
 TP16 1.0V_C
 TP17 1.0V_D
 TP18 1.0V_E
 TP22 1.0V_F

TP23 1.8V
 TP28 2.2V
 TP29 3.3V
 TP31 5.0V

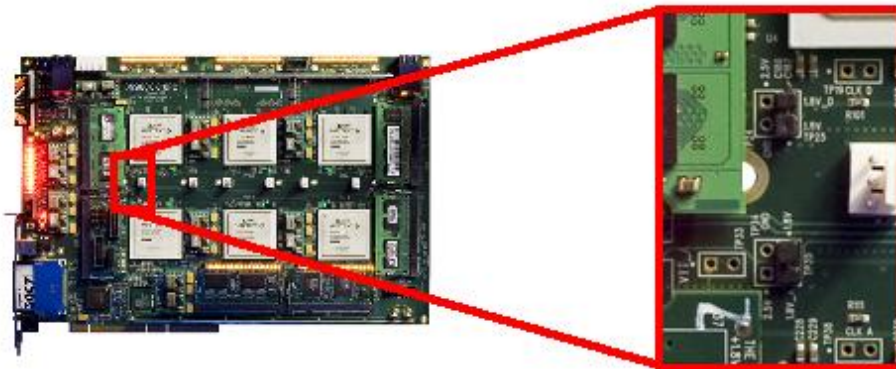
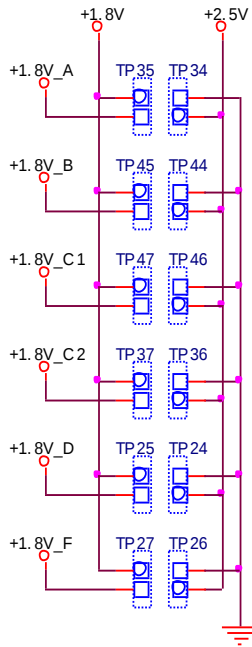


The test point reference designator is not visible on the silkscreen of the DN9000K10PCIE4GL. Instead, there is a label indicating which power net the test point is connected to. These test points are connected by thin traces that are not capable of conducting more than 100mA of current. You should only use these test points for probing. Also, due to the routing of these signals, noise may appear on these test points that is not present on the

actual power plane on the board. For noise measurements, use the through-hole test points provided.

5.3 DIMM Power

As described in the DDR2 Interface section, provisions have been made for the use of +2.5V modules in the memory sockets of the DN9000K10PCIE4GL. To allow this, a jumper point is provided for each DDR2 memory power net.



The DN9000K10PCIE4GL comes with a jumper installed in each of the “1.8V” test points, shorting the DIMM power supplies to the +1.8V power supply.

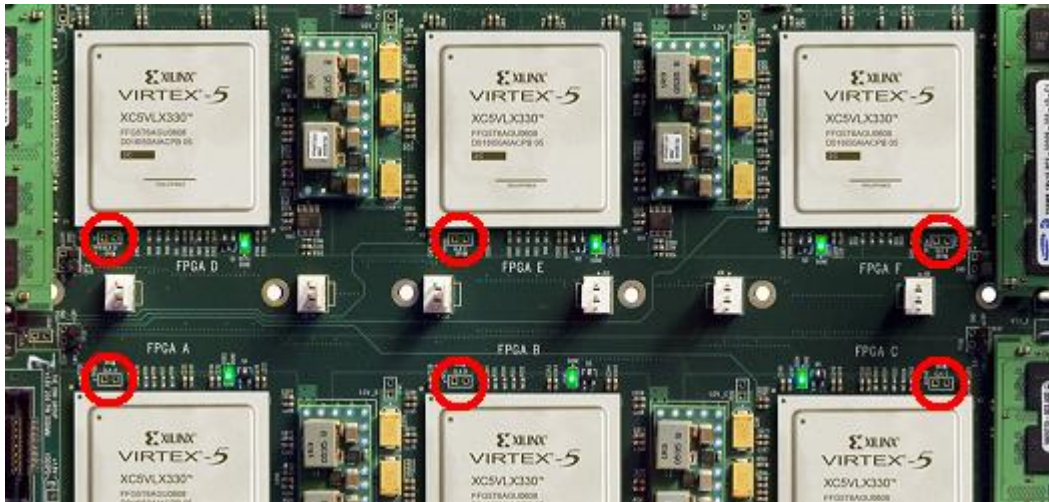
HARDWARE

1.8V	2.5V	DIMM
TP35	TP24	A
TP45	TP44	B
TP47	TP46	C1
TP37	TP36	C2
TP25	TP26	D
TP27	TP34	F
JP1	1.8V_ADJ	

If you require power on any DIMM other than +1.8V, you must remove the installed jumper and instead install a different jumper to connect the DIMM power to +2.5V. If you require +3.3V on this power net, no jumper point is provided, and you will have to run a wire.

5.4 “GC” Test points

Each FPGA is connected to a two-pin test point for debugging purposes. This test point is the same as the ones used for the “Power Thru-hole” test points. Each test point is connected to a “GC” pin on the FPGA, meaning it can be used as a differential clock input to the FPGA. By connecting a reference voltage to the “N” pin (circular), this test point can be used to input a single-ended signal.



TP38	A
TP39	B
TP40	C
TP19	D
TP20	E
TP21	F

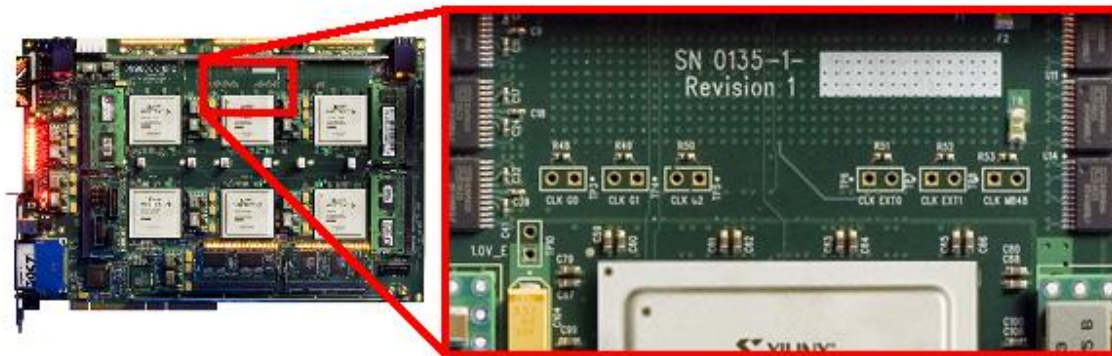
Note that TP38, the FPGA A test point is used for +3.3V signaling, the rest are +2.5V-only.

Note that the signals connected to either side of these test points are shorted together by a 5-ohm resistor. This connection allows the use of external clock feedback. If you need to use these test points as two separate I/Os, this resistor would have to be removed.

The reference design uses this connection for external clock feedback. The register is called “TPP”

5.5 Clock Test points

Each of the “Global clock” networks has a test point. These points are not length-matched with the global clock network, so there may be some phase offset between this point and the FPGA input.



TP8	MB48
TP3	G0
TP4	G1
TP5	G2
TP6	EXT0
TP7	EXT1
TP51	CLKP

All of these networks use LVDS signaling, except for TP51, which is a single-ended, +3.3V LVC MOS signal. LVDS test points have the “p” signal connected to pin 1 (square) and “n” connected to pin 2 (circular). LVC MOS signals are connected to pin 1, with pin 2 being ground.

A 100-ohm resistor connects the P and N side of these clock signals. This is excellent for probing with a high-impedance probe, but not so good for connecting wires. You can remove this resistor if needed.

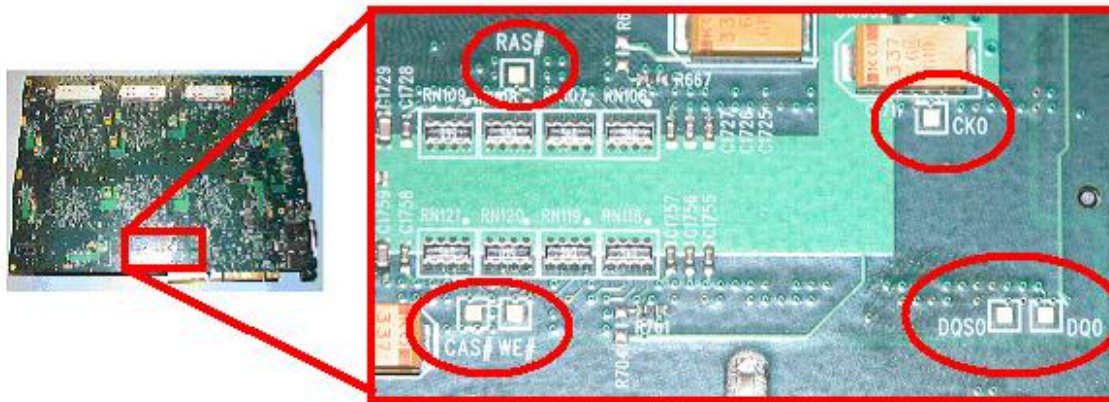
5.6 DIMM Signals

Some key signals on each DDR2 interface are connected to test points for debugging. The test point pad used is the same as on the “Power TP” test points.

HARDWARE

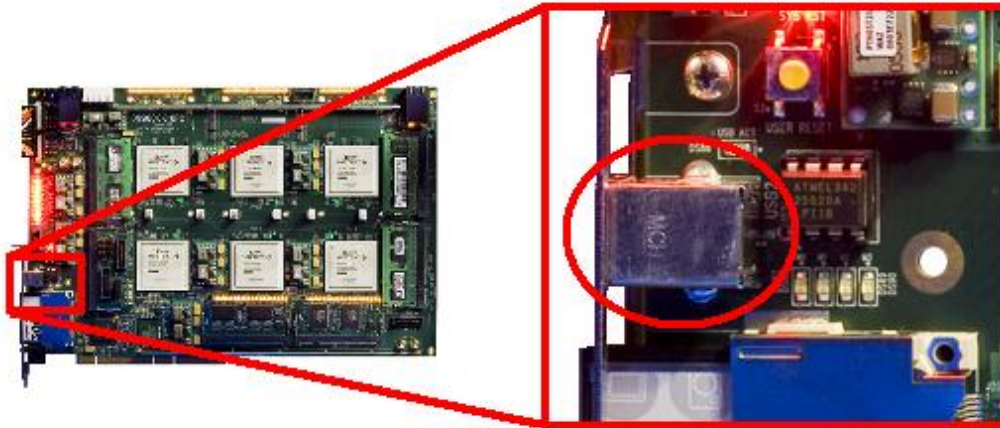
	A	B	C1	C1	D	F
WE	TP72	TP84	TP88	TP71	TP60	TP59
DQ0	TP66	TP82	TP86	TP77	TP54	TP65
DQS0(p)	TP67	TP83	TP87	TP76	TP55	TP64
CK0(p)	TP68	TP80	TP81	TP75	TP56	TP63
RAS#	TP73	TP78	TP79	TP70	TP61	TP58
CAS#	TP74	TP85	TP89	TP69	TP62	TP57

The test points are not labeled with their reference designators. Instead, they are labeled with the signal name. The reference designators above appear in the schematic.



6 USB interface

The DN9000K10PCIE4GL allows the user FPGA to communicate to a host PC over USB. The configuration circuitry allows this by bridging USB to the Main Bus interface. For most users, implementing USB communication will be as simple as making a Main Bus controller. In the reference design, there is an example Main Bus controller. See the Main Bus section of this chapter for more information on the Main Bus.



USB on the DN9000K10PCIE4GL also allows control of the configuration circuitry from a host PC. This includes configuring FPGAs, setting clock frequencies and others

This section will describe the software interface required to communicate to the DN9000K10PCIE4GL. In addition to reading this section, you may chose to modify the provided software (USB Controller and AETest_usb). The source code for these programs is on the user CD. These programs collectively implement all of the available controls on the DN9000K10PCIE4GL.

6.1 DINI.H API

This is a placeholder for the Dini API it doesn't exist yet.

6.2 Connecting to the DN9000K10PCIE4GL

Depending on the operating system, there are different methods of obtaining a software handle to the DN9000K10PCIE4GL in order to access it from software.

6.2.1 Windows XP

The driver used is the EzUSB driver.

```
HANDLE handle = CreateFile("\\\\.\\Ezusb-0", GENERIC_WRITE,
                           FILE_SHARE_WRITE, NULL, OPEN_EXISTING,
                           0, NULL);
```

The "EzUsb-0" device name is registered with Windows when installing the EzUSB device driver. The .ini file provided with the driver causes the driver to be assigned to any USB device with VendorID=0x1234 and ProductID=0x1234.

The HANDLE object returned by CreateFile is suitable for use with DeviceIoControl().

6.2.2 Windows Vista

Testing was not complete at print time. support@dinigroup.com

6.2.3 Linux

To use USB in Linux, use the provided usbdvrlinux.c file provided on the user CD in AETest_usb/driver

Connecting to the device occurs using the driver's usb_open function.

```
int handle = usb_open(0x1234, 0x1234, 0);
```

Note that the driver must be compiled on your system, and that any binaries provided, if any, probably won't work on your system. Compiling the driver requires the kernel source.

6.2.4 Communication

The USB interfaces that the DN9000K10PCIE4GL presents are separated into two types: The Vendor requests, and the Bulk Transfers. All other types of USB transactions are not supported. The vendor requests are low-bandwidth control signals used for controlling the board settings. The Bulk Transfers are used for configuring and reading back FPGAs and reading and writing to the main Bus interface.

6.3 Vendor Requests

Most of the “control” functions available over USB are accomplished using a “vendor request”. Programming a USB vendor request is out of the scope of this document, but you can copy the code provided in the USB Controller program.

The following table describes the USB interface presented to the host by the MCU micro controller.

Vendor Request Name	ID	Description
VR_GET_FLASH_REV	0xA6	Returns a revision code of the “MCU” firmware
VR_GET_FPGA_INFO	0xA7	
VR_REBOOT	0xAD	
VR_CONFIG	0xAF	Causes MCU to go through configuration sequence (Media Card)
VR_FLASH_VERSION	0xB2	Reads version of flash code
VR_DISPLAY_FPGA_INFO	0xB3	
VR_CHECK_FPGA_INFO	0xB4	
VR_CHECK_FPGA_CONFIG	0xB5	Returns a string representing if the selected FPGA is configured
FLASH_VERSION_ADDR	0x08	Value to go into upper address register (MCU_XADDR)
VR_SET_EP6TC	0xBB	Sets the size of the bulk transfer (Read) buffer. You must set this to a value equal to the SIZE field of the USB Bulk transfer

VR_SETUP_CONFIG	0xB7	This vendor request must be called to select an FPGA for configuration prior to a bulk transfer containing the configuration stream for that FPGA.
VR_END_CONFIG	0xBD	This vendor request de-selects an FPGA after configuration and returns the configuration status of that FPGA (DONE signal)
VR_MEM_MAPPED (Config Read) (Config Write)	0xBE	This vendor request reads or writes to the address space of the MCU. This vendor request can be used with the configuration register map above to accomplish any configuration task.
VR_CLEAR_FPGA	0x90	Clears the selected FPGA of configuration data.
VR_BOARD_VERSION	0xB9	Returns a byte representing the type of board

Each vendor request has a direction, request type, request, and value, size and buffer pointer fields. The request type is always TYPE_VENDOR. The request field is the ID listed in the table above. The value and data in the buffer pointer fields are vendor-request specific. The size field is the number of bytes in the buffer. The details of how to implement a vendor request are outside the scope of this manual.

6.3.1 VR_CLEAR_FPGA

This vendor request clears an FPGA.

Direction is OUT. Size is 0. Value represents which FPGA should be cleared. 0 is FPGA A. 1 is FPGA B... and so on.

6.3.2 VR_SETUP_CONFIG

This vendor request must be called before sending configuration data to an FPGA. It tells the DN9000K10PCIE4GL which FPGA should receive the next configuration stream sent over USB. It also clears that FPGA of its current configuration.

Direction is OUT. Size is 1. In the buffer is a number representing which FPGA should be selected. 0 is FPGA A, 1 is FPGA B, 2 is FPGA C... and so on.

6.3.3 VR_END_CONFIG

This vendor request de-selects an FPGA (so that configuration data sent will go to no FPGA) and checks the configuration status of an FPGA.

6.3.4 VR_SET_EP6TC (Read buffer size)

The SetReadBufferSize vendor request must be used before any “bulk read” bulk transfer. This sets the size (in bytes) of the data that will be requested by the bulk transfer. If this vendor request is not sent before the bulk read, the behavior is undefined.

The direction is OUT. The size is 0. The value is the number of bytes required for the next bulk transfer.

6.3.5 VR_MEM_MAPPED (Configuration Registers)

Some of the controls on the DN9000K10PCIE4GL do not have their own Vendor Request. These functions include setting the clock frequencies. In order to accomplish these tasks, you must use the “Configuration Registers”. The full list of registers is in the “Configuration Section” section. To write to a configuration register, use the VR_MEMORY_MAPPED vendor request.

The direction is OUT. The value field is the address you wish to write to (example 0xDF39, the disable Main Bus register). The size field should be 1. The buffer should contain a single byte containing the byte to be written to the Configuration Register. All configuration registers are one byte.

6.3.6 Other Vendor Requests

Many of the Vendor requests used by the USB Controller program are not documented. Dini Group does not support these requests for users. If you need a function that you feel is not described here, contact support@dinigroup.com

6.4 Main Bus accesses

The USB Controller control the DN9000K10PCIE4GL reference design using USB vendor requests and bulk transfers that access the configuration FPGAs registers. These registers cause “Main Bus” transactions with the user FPGAs. The host computer initiates all Main Bus transactions. To see a specification of the Main Bus interface, see *Reference Design*.

To request a Main Bus interface write transaction, the USB Controller program sends a USB bulk write to EP2 (endpoint 2). The first byte contains a code, either 0x00 or 0x01, determining whether the next 4 bytes contain an address or a datum. If this byte is a 0x00, the next 4 bytes in the bulk transfer are stored into an address register. All data transferred to and from the main bus is LSB first. The address 0x12345678 should be sent as a bulk transfer of 5 bytes: 0x00, 0x78, 0x56, 0x34, 0x12. To send a datum, send the code 0x01, followed by 4 bytes, LSB first. When the DN9000K10PCIE4GL receives a data word, it sends it onto the main bus interface to the address in the address register. It then increments the address register. Therefore, to send two words over main bus, 0x00000001 to address 0x00000001 and 0x00000002 to address 0x00000002, the USB Controller would send the following 15 bytes to USB EP2:

```
0x00 0x01 0x00 0x00 0x00
0x01 0x01 0x00 0x00 0x00
0x01 0x02 0x00 0x00 0x00
```

Note that the number of bytes sent to EP2 must be divisible by 5.

To request a main bus read operation, the USB Controller sends a USB bulk write to EP2 to set the address register, as described in the above paragraph. Then, the USB Controller sends a bulk read to EP6 (endpoint 6), with the USB bulk request SIZE field set to the number of bytes

requested. The number requested must be divisible by 4. After the bulk read is complete, the address register is incremented by SIZE/4. Read and write transactions use the same

Before starting a USB read using a bulk transfer, you must tell the DN9000K10PCIE4GL how many bytes are going to be read by using the VR_SET_EP6TC (0xBB) vendor request described in the *Vendor Requests* section.

6.4.1 Important Note About Endpoints

There is only one “endpoint” that the user should use: endpoint 2. Note that an endpoint is bi-directional. Using the driver that Dini group provides, the endpoint and direction fields are stuffed within the same byte. To write to endpoint 2, this byte should be 0x02. To read, it should be 0x08. Some people refer to these as two uni-directional endpoints, 2 and 8. And that’s cool with us.

6.4.2 Performance

Main Bus over USB runs at a maximum speed of 80Mbps for reads, and 32Mbps for writes. These numbers assume that the FPGA operates the Main Bus interface with zero wait cycles. If the FPGA design has more wait cycles, this speeds decreases. The approximate speed of Main Bus over USB is given below as a function of Main Bus wait states.

0 cycles	80Mbps read	32Mbps write
1 cycle	76Mbps read	31Mbps write
5 cycles	64Mbps read	29Mbps write
30 cycles	32Mbps read	16Mbps write
100 cycles	13Mbps read	11Mbps write
250 cycles	6Mbps read	5Mbps write

6.5 FPGA Configuration

The following procedure is used by software on the host computer to configure an FPGA over USB. This procedure is followed by the USBController program and AETest_usb program on the user CD.

- 1) USB Software gets a handle to a USB device with VID 0x1234 PID 0x1234.
- 2) USB host software sends vendor request VR_SETUP_CONFIG 0xB7 (see *Vendor Requests*) with 1 byte in the data buffer representing which FPGA to configure. (A is 0x01, B is 0x02, C is 0x03...)
- 3) The configuration circuit on receiving this vendor request asserts the PROG signal of the selected FPGA. This resets the FPGA and clears any configuration data it may already have. This Vendor request also selects the FPGA, so that SelectMap bus activity only affects the selected FPGA. Bulk transfers initiated after this command to endpoint 2 are interpreted as SelectMap transfers, rather than Main Bus transfers. (See *Main Bus access* above). This will be so until vendor request VR_SETUP_END (0xBD) is called.

4) USB host software sends a bulk write USB request to EP2. Each byte of data in the bulk write is sent to the selected FPGA over the SelectMap bus, and the FPGA signal CCLK is pulsed once for each byte of data sent. Note that the LSBit in the USB transaction is sent to the LSBit in the SelectMap interface, so bit swapping as described in the Virtex 5 Configuration Guide is not required. A standard .bit file from Xilinx bitgen can be transferred in binary over this USB interface to correctly configure an FPGA on the DN9000K10PCIE4GL. Make sure CCLK is selected as the startup clock in the bitgen settings. This is the default setting.

5) After an FPGA configures, the DONE signal will go high, lighting the blue LED next to the FPGA (labeled “DONE”).

6) The USB Controller sends a vendor request out VR_SETUP_END (0xBD). This request deselects the FPGA, so that further bulk requests are interpreted as Main Bus transactions.

6.5.1 Readback

This section should be updated when Readback is enabled. If this sentence is here, it means this section has not been verified yet.

Readback is performed in the same way that configuration, except that the direction of the bulk transfer is BULK_READ instead of BULK_WRITE. The commands required by the SelectMap interface to start a Readback must be sent using the configuration interface. For this reason, it is the programmer’s responsibility to understand and implement the SelectMap protocol.

6.6 USB Hardware

The actual hardware associated with performing USB communication with the DN9000K10PCIE4GL is briefly described here. Since the user is not required to understand how to operate the hardware from the FPGA, much detail is omitted.

6.6.1 Cypress CY7C68013A

A Cypress Micro controller (MCU) with a built-in USB controller provides the USB interface of the DN9000K10PCIE4GL. For a low-level understanding of the way the DN9000K10PCIE4GL communicates over USB, you should see the Cypress CY7C68013A datasheet. The driver that Dini Group provides is the free Cypress EzUSB driver, with customizations to the corresponding .ini file to identify the board as a “Dini Group Emulator” product.

As with all USB devices, communication with the DN9000K10PCIE4GL is initiated by the host (PC) and can be either a USB “vendor request” or “Bulk transfer”. All other types of USB transactions are not supported or documented with the DN9000K10PCIE4GL. In general, Bulk transfers are used for high-bandwidth data and vendor requests are used for all other control functions.

Bulk Transfer Functions:

Configure FPGA (SelectMap)

Readback FPGA (SelectMap)

MainBus read

MainBus write

Vendor requests can contain short (512Byte) messages in either direction, and cause the MCU to execute code. In response to most vendor requests, the MCU will modify or read values in the Configuration memory space (see next section).

Since vendor requests can contain only a limited amount of data, USB Bulk transfers are used to send configuration data to the DN9000K10PCIE4GL. The MCU is too slow to process USB 2.0 data at full speed, and so the bulk transfer data is sent to external pins on the Cypress MCU (see Cypress datasheet) and to the configuration FPGA (next section). Currently, this data is only used to configure FPGAs, and so the data is sent to the SelectMap pins of the Virtex 5 FPGAs.

To begin communication with the DN9000K10PCIE4GL, the USB Controller program creates a USB connection object in the host operating system, by opening Vendor ID 0x1234 product ID 0x1234. (For the purposes of updating the firmware, the DN9000K10PCIE4GL can come up in “EPROM” mode, where it loads a program capable of connecting over USB to a host, downloading firmware and writing it to the MCU flash memory, U201. The check the MCU makes on reset to determine which mode it should start in is the firmware update switch, S1 #4. This EPROM code is stored in the EPROM DIP installed in U203. When the MCU is in this mode, it registers itself to the operating system as Vendor ID 0x1234, product ID 0x1233. For firmware update instructions, see *USB Software: Firmware Update*. For information about the MCU boot up sequence, see *Hardware: Configuration Circuit: MCU*)

The source code for the MCU firmware (“Flash”) is provided in
D:\Source Code\MCU\FLASH
as a Keil Studios MicroVision 2.11 project file.

6.6.2 Activity LED

A yellow LED located next to the USB connector flickers playfully when there is USB activity.

6.6.3 Configuration FPGA

The MCU unit controls all of the configuration circuits on the DN9000K10PCIE4GL, but it does not have sufficient IO to access all of the configuration signals. For IO expansion, the MCU’s external memory bus is connected to a Virtex 5 LX40 FPGA. This FPGA provides a memory-mapped interface to all of its IO. This bus is called the ‘Configuration Space’.

The configuration FPGA is connected to all of the configuration signals of the Virtex 5 FPGAs, the temperature sensors, status LEDs, CompactFlash card, reset buttons, clock synthesizer control signals, global clock multiplexer control signals, FPGA clock inputs, the Main Bus.

The source code for the Configuration FPGA is provided in
D:\Config_Section_Code\ConfigFPGA

This project can be compiled using Xilinx ISE version 7.1i SP4 or later.

It also appears to require Synplify

6.6.4 Power

The DN9000K10PCIE4GL does not draw any power from the USB connector. Hot-plugging the DN9000K10PCIE4GL is acceptable.

6.7 Troubleshooting

If you cannot get USB to communicate with your design over Main Bus, please try using the USB Controller software with your design, and using the Dini Group reference design with your software. This will help determine whether the software or the hardware is causing the error.

6.7.1 USB Controller Freezes

The Vendor requests on the DN9000K10PCIE4GL are blocking. Only one can be completed at a time. This includes vendor requests that take a very long time like “Configure from CompactFlash” (10 seconds). During this time USB Controller, a single-threaded application, freezes when any Vendor Request is issued. (All the time).

6.7.2 Main Bus always returns 0x_____ (Error Codes)

0xDEADDEAD

Main Bus timeout. The VALID signal on Main Bus was never asserted. See the Main Bus section for details. Your FPGA may not be configured.

0x12345678

This error code may mean the “Enable USB->FPGA Communication” button in USB Controller has not be pressed (the Main Bus disable register is set)

0xDEAD5566

This error code is returned by the Dini Group reference design when there is a Main Bus read to a register that is not defined. (Default Main Bus output). This code is specific to the reference design.

0xDEAD1234

This sentinel value is no longer used and should not be returned.

0xABCDABCD

This error code is returned when a MainBus register corresponding to a memory is read, but the memory is not implemented in the Reference Design

7 PCI Express Interface

PCI express is available to FPGA A though the Genesys Logic GL9714 4-lane PHY device. The 4 PCI express lanes are connected to the 4x edge connector. The back-end interface to the FPGA is a standard PIPE interface, consisting of 8 data signals in each direction for each lane, operating at 250Mhz each.

There are four ports on the GL9714, named A through D. The channels are not in the order one might expect. Channel A corresponds to PCI express lane 2, channel B corresponds to PCI express lane 0, channel C corresponds to PCI express lane 1, and channel D corresponds to PCI express lane 3.

The GL9714 has three operating modes:

M1: 2-lane, 125Mhz with 8b/10b encoding-decoding

M0: 4-lane, 250Mhz with 8b/10b encoding-decoding

M3: 4-lane, 250Mhz without 8b/10b encoding-decoding

The DN9000K10PCI is designed to be compatible with any of these four modes. The reference design only uses M0. When in mode M1, only channels B and C are active (corresponding to lanes 0 and 1). The other lanes should be held inactive using the TXIDLE signals. In any mode, the user can operate in 1x or 2x mode by holding unused lanes inactive with the TXIDLE signal.

7.1 Electrical

The drive standard used on output signals from the FPGA on the PIPE interface should be LVDCI. Inputs on the PIPE interface should use SSTLII_DCI.

7.2 Timing

Inputs and outputs should be synchronous to the PCIE_PCLK signal driven from the GL9714. The phase of the clock used on the FPGA's input flip-flops and the phase of the clock used on the FPGA's output flip-flops should be independently determined. The reference design contains an example of a DCM instantiation that will work at 250Mhz.

7.3 Host Interface, Mechanical

Push harder.

7.4 Hardware Design Notes

There are three LEDs on the board that give the current status of the PCI express links. These LEDs are DS155, DS156, and DS157. They correspond to activity, link on lane 0 and link on lane 3 respectively.

8 Unusable pins

Some of the user IO available on the Virtex-5 FPGAs may not be used on the DN9000K10PCIE4GL. These include pins dedicated for DCI (IO calibration), VREF pins for interfaces with externally generated thresholds, configuration pins, and pins that are not connected on the DN9000K10PCIE4GL.

The provided UCF files for the board have a CONFIG PROHIBIT directive on these pins to prevent them from being used. Since your designs must always have 100% location-constrained IO, this directive is redundant.

HARDWARE

The user doesn't need to know anything about the way the signals in this section are connected on the DN9000K10PCIE4GL, just that these pins cannot be used. Ironically, these are some of the best-documented pins in this manual.

8.1.1 NC Pins

The following pins are not connected to anything on the DN9000K10PCIE4GL.

A	B	C1	D	E	F
M28		C18	AC35	L15	Y8
N6		C19	AB36	AP30	AP1
M4		A4	AL39	AD37	AV5
M28		F34	AM39	P36	G6
M1			AN38	AT37	F6
R3			AM38	AR38	F5
P2			H41	AR37	E5
C18			J41	AT36	V11
C19			AC41	AJ35	V10
A4			AD42	AK35	P3
			AP42	AB42	R3
			AP41	AC41	P2
			AT41	AD42	P1
			AU41	AE42	AA1
			M37	AD41	AA2
			V39	AU42	AD7
			W38	AV41	AD8
			AA34	N39	AJ6
			Y34	M39	AH5
			W36	V39	AM6
			W37	W38	AN5
			C18	AA34	AH30
			C19	Y34	AH29
			A4	C18	
				C19	
				A4	

Sometimes when I need just six more connections between FPGA E and F, I look at this chart and get really frustrated.

8.1.2 Configuration

The following pins (All FPGAs) are the SelectMap data pins, used to configure the FPGAs. These pins are connected to all six Virtex-5 FPGAs. Using these signals for FPGA interconnect is possible, but may interfere with the configuration circuitry on the DN9000K10PCIE4GL.

AJ13
AK13
AJ28

AK29
 AL15
 AL14
 AJ26
 AJ27
 AK28
 AK27
 AL16
 AK17
 AM29
 AN30
 AN16
 AM16

8.1.3 DCI

The “DCI” self-calibrating IO on the DN9000K10PCIE4GL’s FPGA interfaces uses the following pins. These interfaces include DDR2, daughtercards connectors. Each of these pins is connected to a 50-ohm resistor either to power or ground. I guess you could make use of these pins if you wanted to wire something to the resistor pads.

A	B	C1	D	E	F
K13	K13	K13	K13	K13	K13
J13	J13	J13	J13	J13	J13
AN15	AN15	AN15	AN15	AN15	AN15
AN14	AN14	AN14	AN14	AN14	AN14
C38	U3	C38	AP37	AG39	W7
D38	U2	D38	AP36	AH39	W8
A19	AL2	A19	AG39	R42	AL2
B19	AK3	B19	AH39	P42	AK3
A5	W7	A5	R42	H36	AM7
A6	W8	A6	P42	G37	AN6
	AJ5	AK40	AK40	AP37	T10
	AK4	AL40	AL40	AP36	R10
		R42	H36	AK40	U3
		P42	G37	AL40	U2
		AG39	V38	V38	AJ5
		AH39	U37	U37	AK4
		V38	C38	C38	
		U37	D38	D38	
		L34	A19	A19	
		K34	B19	B19	
			A5	A5	
			A6	A6	
			L34	F10	

K34 F11

8.1.4 VREF

The following pins are used for a “VREF” reference voltage required by the Virtex-5 FPGA to use some drive standards. These VREF pins are required when using SSTL signaling on DDR2 interfaces. Additionally, all global clock inputs are supplied with VREF to allow single-ended end termination.

A	B	C1	D	E	F
L16	L16	L16	L16	L16	L16
AN29	AN29	AN29	AN29	AN29	AN29
C33	M2	C33	C33	C33	
B38	W3	B38	B38	B38	
A25	AE3	A25	A25	A25	
A17	AM2	A17	A17	A17	
D8	L4	D8	D8	D8	
D5	W6	D5	D5	D5	
P28	AB7	P28	P28	P28	
	AL5	AG41			
		AM42			
		L41			
		T41			
		AD38			
		AK37			
		R38			
		U39			
		G34			
		J32			

8.2 Extended MB

In the netlist there are signals named MB[169] through MB[182]. These signals connect two or more FPGAs, but not to all six FPGAs. These signals are not advertised as part of the available interconnects on the DN9000K10PCIE4GL, but are nonetheless usable. The highest index of the “MB” bus connected to each FPGA is listed below.

A (U1): MB168
 B (U2): MB179
 C (U3): MB168
 D (U4): MB182
 E (U5): MB173
 F (U6): MB182

Signals MB180, 181 and 182 only connect between FPGA D and F.

9 System Monitor/ADC

The “System Monitor” function of the Virtex-5 was undocumented at time of the DN9000K10PCIE4GL design. I bet by using rework wires, this feature could be enabled. Dini Group doesn’t know of anyone who is interested in System Monitor. Call us I guess.

The temperature sensor function of the System monitor is replaced by on-board temperature sensing chips. More information is in the power section of this chapter. When an FPGA is operating above the recommended temperature range of the device (85 degrees C), the FPGA automatically un-configures.

10 Reset

There are two reset circuits on the DN9000K10PCIE4GL. One is the power-on reset, or “Hard Reset”, that holds the board, including the configuration circuitry in reset until all power supplies on the board are within their tolerances. The second reset circuit is the user reset, or “Soft reset”.

10.1 Power Reset

The power-reset signal holds the configuration circuit (including a micro controller and Spartan 3 FPGA) in reset. It also causes the FPGAs to become un-configured, and causes the RSTn signal on the daughtercards to be asserted. When the board is “in reset”, the “Hard Reset” LED, DS85, is lit red. It is located about an inch above the USB connector. It is red.

When the board is in reset, FPGAs cannot be configured, USB does not function (the host computer will not be able to communicate with the device). When in reset, the Spartan configuration FPGA remains configured, but all of the logic in the device is cleared.

Pressing the “HARD RESET” button, S1, located near the ATX power connector, can trigger the Power reset. This reset cannot be triggered over PCI or USB. It is also triggered with one or more voltages on the board fall below, or above a certain threshold. These thresholds are given below:

Voltage	Min	Max
1.0V (A-F):	0.94V	1.1V
1.8V:	1.67V	3.8V
3.3V:	2.7V	3.8V
5.0V:	4.0V	5.6V
12V:	--	--
2.5V	2.25V	2.7V

When the board comes out of reset, the micro controller goes through an initialization process that will cause all current settings to be lost, including clock settings. Also, the configuration

circuit will act as if the board has just powered on and read from the main.txt file to configure FPGAs.

When reset is triggered, it remains triggered until 55ms after all trigger conditions are removed. This behavior prevents USB from behaving in such a way to permanently disable USB on the host machine.

Under some conditions, the DN9000K10PCIE4GL can fail to be responsive after rapidly asserting and de-asserting reset, or if the board is powered off and back on very quickly. This behavior is caused due to a flaw in the micro controller used for the DN9000K10PCIE4GL configuration circuit.

10.2 User Reset

The “USER RESET” circuit is intended for use by the user. When this reset is asserted, the RESET_# signal (from the schematic), is asserted to each FPGA. After at least 200ns, this signal is de-asserted simultaneously to each FPGA. This signal is connected to a regular user IO on the FPGA, so it is up to the FPGA designer to implement reset correctly within his design.

The User Reset is asserted whenever the “User Reset” button is pressed. This button, S2, is located just above the USB connector. There is no LED indicating the state of user reset. User reset is also asserted when the reset vendor request is sent over USB.

When User reset is asserted, the RSTn signal to each daughtercard is also asserted.

The rise time of the reset signal is fairly slow (10s of nanoseconds), and the delay within the FPGA of the reset signal cause the actual de-assertion time of the logic within the FPGA to be uncertain by as many as 20ns. (the timing of a synchronous reset within a single FPGA is guaranteed) This means that if this signal is used to reset circuitry used for inter-FPGA communication, care needs to be taken that a synchronous reset is not required for the multiple-FPGA system to operate correctly. Alternately, you design can re-generate a synchronous reset and distribute this signal using a MB* signal.

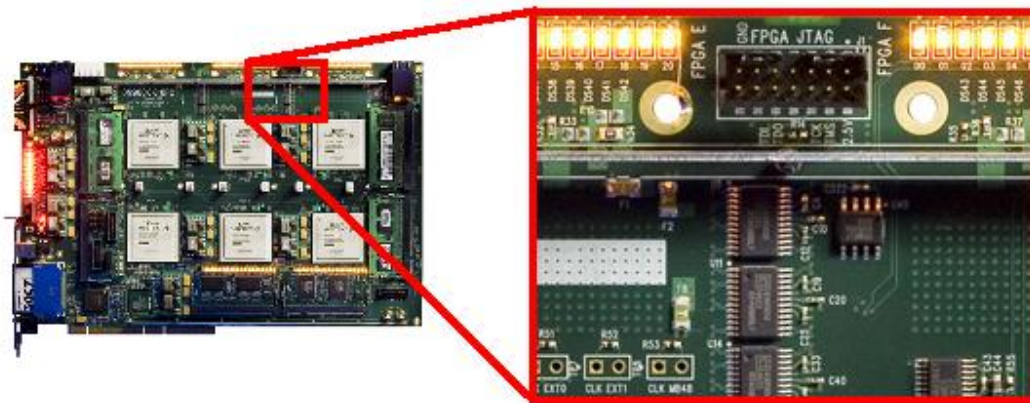
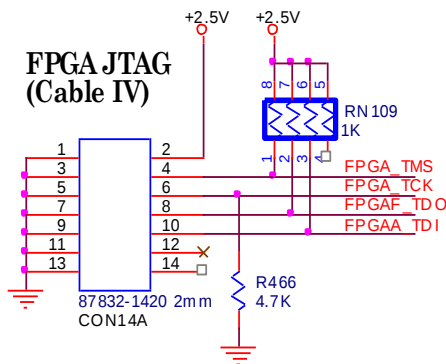
11 JTAG

There are two JTAG headers on the DN9000K10PCIE4GL. The first is used only to update the board’s firmware. The second, J1 is connected to the JTAG port of the six Virtex-5 FPGAs. This interface can be used for configuring the FPGAs, or using debugging tools like ChipScope or Identify.

11.1 FPGA JTAG

The connector for FPGA JTAG is shown below.

HARDWARE



Note that the signal “TDO” on the header and in the schematic refers to the “TDO” port of the FPGA, not the connector.

The order of the FPGA JTAG chain is FPGA A->FPGA B->FPGA C->FPGA D->FPGA E->FPGA F. There are no other components in the chain. If you received your board with fewer than six FPGAs installed, then the chain will be shorter.

The voltage of the JTAG chain is fixed at 2.5V and cannot change. Hot-plug on this header is allowed.

11.1.1 Compatible Configuration Devices

The JTAG header is designed to work with the Xilinx Parallel IV or Platform USB cable. The JTAG chain is tested at manufacture using a Platform USB cable at 12Mhz.

The driver installation process for the Platform USB cable is relatively difficult for a USB device. Follow the instructions carefully.

In order to achieve high-speed configuration using a Parallel IV cable, you need to enable ECP mode on your parallel port. This is probably a BIOS setting on your computer.

11.1.2 Identify

In order to use JTAG debugging tools on the DN9000K10PCIE4GL, you do not need to configure via JTAG.

11.1.3 ChipScope

In order to use JTAG debugging tools on the DN9000K10PCIE4GL, you do not need to configure via JTAG.

11.2 Firmware Update Header

The firmware update JTAG header J16, should not be used unless you are updating the DN9000K10PCIE4GL firmware. This header is used with a Xilinx Platform USB or Parallel IV cable. The instructions for updating the firmware are in the Controller software chapter.

11.3 Ethernet JTAG

See the Ethernet section.

11.4 Troubleshooting

If you are having problems getting JTAG to work, try connecting the Xilinx Platform USB cable to the JTAG header and running the Xilinx program Impact. Impact will generate a failure log that you can email to support@dinigroup.com. If you have an upgraded board, please mention this in your email.

12 RS232 Interface

RS232 access is available to all six FPGAs through the header P2 "FPGA RS232". To connect to this header, use the provided .1" header-to-DB9 cable to connect to a PC's serial port.

The RS232 transmit and receive signals are connected to each FPGA's pins:

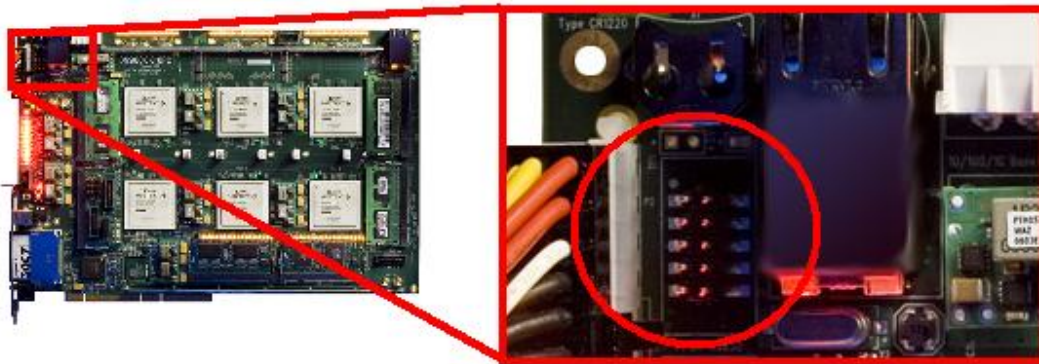
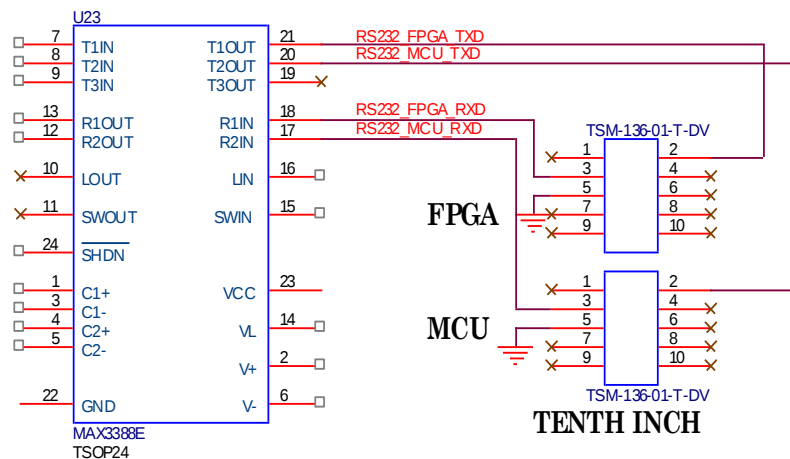
AJ16 (TX from FPGA)

AJ17 (RX to FPGA)

The TX and RX signals use the RS232 data protocol, so the FPGA will have to implement a UART in its logic.

All six FPGA share the same RX and TX signals, so only one FPGA should use the interface at a time. RS232 requires a 12V to -12V signaling level, which is not available on Virtex5 FPGAs, so an external RS232 buffer is used.

HARDWARE



On the board, pin 1 is marked with a white circle dot. On the provided cable, pin one is marked with a red stripe on the cable. Hot-plugging this connector is acceptable.

The port settings required on the serial ("COM") port of your computer are dependent on the UART in the FPGA. Since the flow-control signals on the serial cable are not connected to the FPGA, you cannot use "hardware handshaking".

12.1.1 Configuration RS232

A second RS232 header (P3) is for the configuration circuitry to give feedback to the user. It is described in the section "Configuration Section".

13 Temperature sensors.

Each FPGA is connected to a temperature monitor. This monitor can internally measure the temperature of the FPGA silicon die. The maximum recommended operating temperature of the FPGA is 85 degrees. The accuracy of the temperature sensor is about +0 to +5 degrees. When the configuration circuitry measures the temperature of any FPGA rise above 80 degrees, it will immediately un-configure the hot FPGA, and prevent it from re-configuring. When the temperature drops below 80, the configuration circuitry will again allow the FPGA to configure.

When this occurs a message will appear on the CONFIG RS232 port (P3). An example test output is given below.

```
*****
TEMPERATURE ALERT: FPGA A
CURRENT TEMPERATURE: 81 DEGREES C
THRESHOLD TEMPERATURE: 80 DEGREES C
THE FPGA IS BEING CLEARED IN AN ATTEMPT TO PREVENT HEAT DAMAGE.
SOFTWARE WILL PREVENT RECONFIGURATION UNTIL THE TEMPERATURE
DROPS A FULL DEGREE BELOW THE THRESHOLD TEMPERATURE.
*****
*****
TEMPERATURE ALERT: FPGA A
CURRENT TEMPERATURE: 79 DEGREES C
THRESHOLD TEMPERATURE: 80 DEGREES C
THE FPGA HAS DROPPED BELOW THE ALARM THRESHOLD
AND MAY NOW BE RECONFIGURED.
*****
```

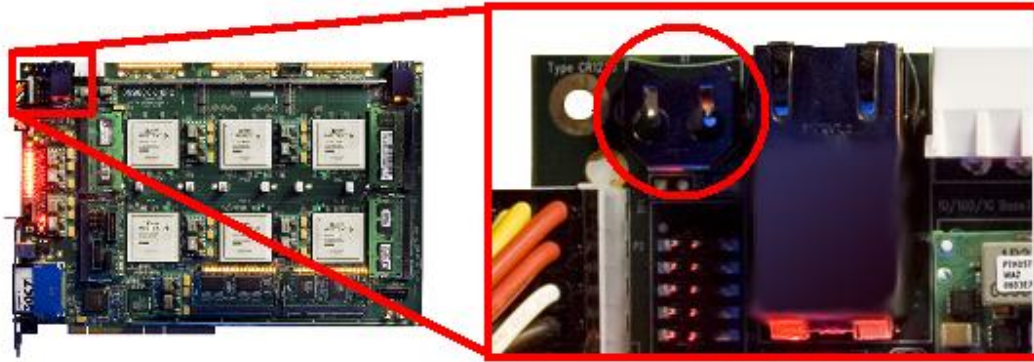
The FPGA can safely operate as hot as 120 degrees, but timing is not guaranteed. You can use the temperature setting in the ISE place and route tool to make timing allowances for operating the FPGA out-of-range. If you want to disable the temperature limit on the DN9000K10PCIE4GL, you can do that using a menu option in the configuration RS232 interface.

14 Encryption Battery

The Virtex5 FPGA supports bit stream encryption. When using encryption, the FPGA must decode the bitstream using a secret key that is stored in a persistent memory in the FPGA. When the DN9000K10PCIE4GL is powered off, a voltage is supplied to the FPGA by a battery installed in socket X1.

HARDWARE

X1 is designed to house a CR1220-type lithium coin-cell battery. Typically, these batteries produce 3.0V. The socket may also work with battery types DB-T13, L04, PA. These however, have not been tested. Insert the battery positive side up.



The same battery is used for all six FPGAs. Removing the battery will cause the FPGAs to lose their encryption memories, and will have to be re-programmed before they can work with encrypted bitfiles again.

To create encrypted bitfiles, turn on the “encryption” option in bitgen. The program will produce an additional output file with an .nky extension. Use the program impact with a Platform USB JTAG cable (plugged into the FPGA JTAG connector on the DN9000K10PCIE4GL) to load this .nky file into each FPGA.

When using a bitfile with encryption enabled, the DN9000K10PCIE4GL will not be able to read the FPGA type out of the bitstream. It will therefore prevent your FPGA design from loading into the FPGA. To disable this behavior, you must disable sanity check. Adding the following line to your main.txt file can do this

Sanity check: n

Also, when using encryption, you must be careful to correctly set the "startup clock" option correctly in bitgen, or the FPGA will fail to configure.

The battery should last 4 years of operation, or 8 years of inoperation.

15 LED Interface

This section lists all of the LEDs. More detailed explanations of the LED functions may be in the sections describing the board system that contains the LED.

15.1 Configuration Section LEDs

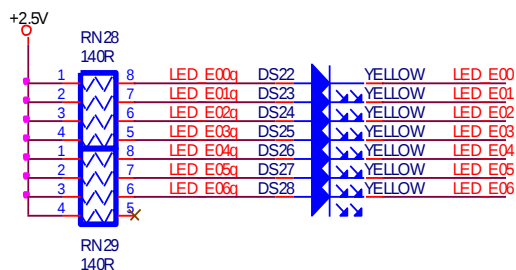
These LEDs are controlled by the configuration section and give the status of the board.

LED Reference Designator	LED Color	Signal Name	The LED indicates the following when ON.
DS66	GREEN	3.3V	The board is powered on
DS148	GREEN	SPARTAN_DONE	Configuration circuit is on (Firmware loaded)

LED Reference Designator	LED Color	Signal Name	The LED indicates the following when ON.
DS82	BLUE	A DONE	FPGA A is configured
DS83	BLUE	B DONE	FPGA B is configured
DS84	BLUE	C DONE	FPGA C is configured
DS73	BLUE	D DONE	FPGA D is configured
DS74	BLUE	E DONE	FPGA E is configured
DS75	BLUE	F DONE	FPGA F is configured
DS86	YELLOW	USB ACT	There is USB activity
DS147	YELLOW	CF ACT	There is CompactFlash activity
DS149	YELLOW	PCI_ACT	There is PCI activity

15.2 User LEDs

These LEDs are connected to an FPGA and are controller by the user. The meaning of the LED is design-dependent. Below is the general circuit used to connect user LEDs. To turn the LED on, drive the signal low. To turn off, tri-state or drive-high the signal.



The user LEDs may be connected to banks where the “Bank Voltage” does not match the LED voltage. In this case, use the drive standard corresponding to the bank, and not the LED. For example, on LEDs connected to the “DDR2 interface banks”, use the drive standard LVCMOS18, even though the LEDs are connected to the +2.5V power. Do not use DCI on

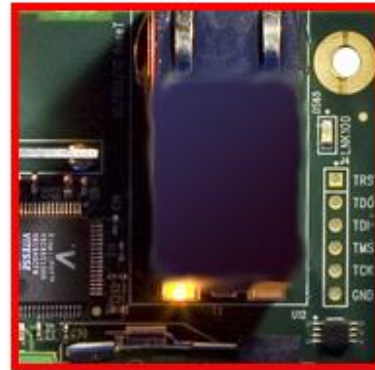
LED signals. You can control the brightness of LEDs by either using a low-drive setting (DRIVE=2ma in the .ucf file), or by rapidly toggling the LED signal high and low.

LED Reference Designator	LED Color	Signal Name	The LED indicates the following when ON.
DS91-94	YELLOW	LED A	User LEDs. Drive with LVCMOS33
DS135-146	YELLOW	LED A	User LEDs. Drive with LVCMOS33
DS95-114	YELLOW	LED B	User LEDs. Drive with LVCMOS18
DS115-134	YELLOW	LED C	User LEDs. Drive with LVCMOS18
DS1-21	YELLOW	LED D	User LEDs. Drive with LVCMOS
DS22-42	YELLOW	LED E	User LEDs. Drive with LVCMOS
DS43-63	YELLOW	LED F	User LEDs. Drive with LVCMOS

The number of LEDs available for the user on FPGA B,C,D,E and F is 24. The number of LEDs available on FPGA A is 16. User LED's on FPGA B-F are numbered 0 to 24. The location of the IOs to use for these LEDs can be found in the provided UCF file or the netlist. The name of each LED is labeled in silkscreen next to the LED.

15.3 Ethernet LEDs

These LEDs are controlled by the Ethernet PHYs connected to FPGA D and FPGA F. They can also be user-controller by setting registers in the serial interface of the PHYs.

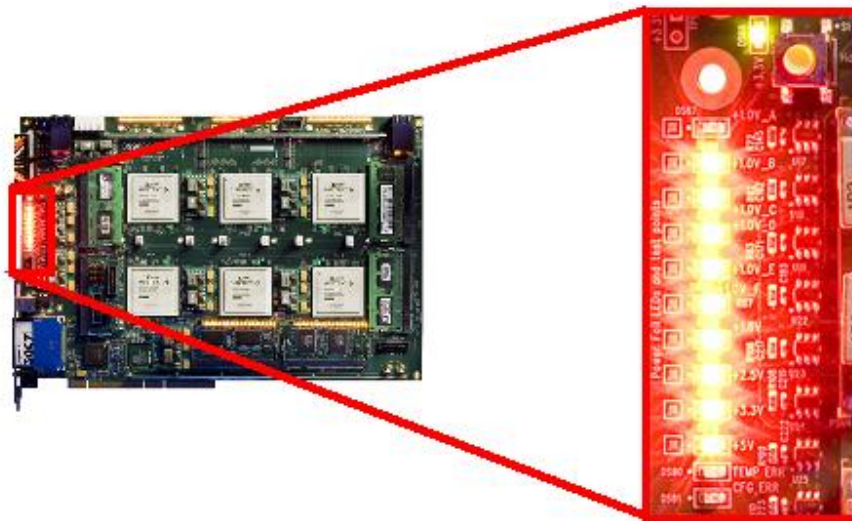


T1 and T2 are the RJ45 jacks on the top edge of the board. There is a yellow and a green LED embedded in this connector, facing the board edge.

LED Reference Designator	LED Color	Signal Name	The LED indicates the following when ON.
T2	GREEN	ETHF_LINK1000	Ethernet PHY F has established link, 1000Base-T
T2	YELLOW	ETHF_ACT	Ethernet PHY F has detected activity
DS65	GREEN	ETHF_LINK100	Ethernet PHY F has established link, 100Base-

T1	GREEN	ETHD_LINK1000	Ethernet PHY D has established link, 1000Base-T
T1	YELLOW	ETHD_ACT	Ethernet PHY D has detected activity
DS64	GREEN	ETHD_LINK100	Ethernet PHY D has established link, 100Base-T

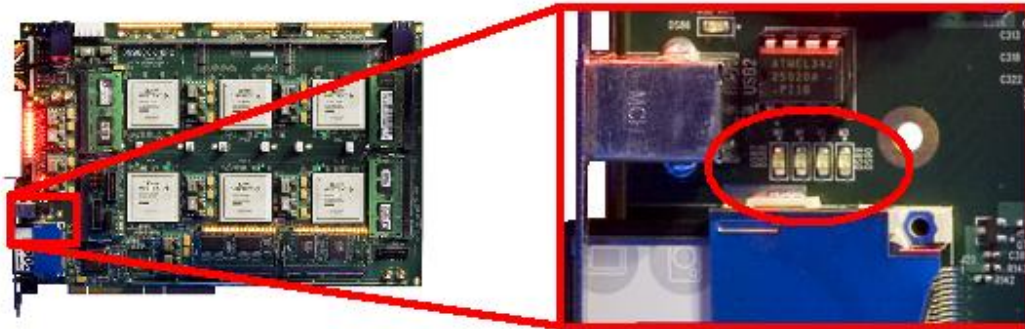
These LEDs indicate is one or more power supplies fail, either outputting a voltage that is too high or too low. The voltage that the LED indicates is marked in silkscreen near the LED.



15.5 Unused LEDs

These LEDs often blink just when you least expect them to.

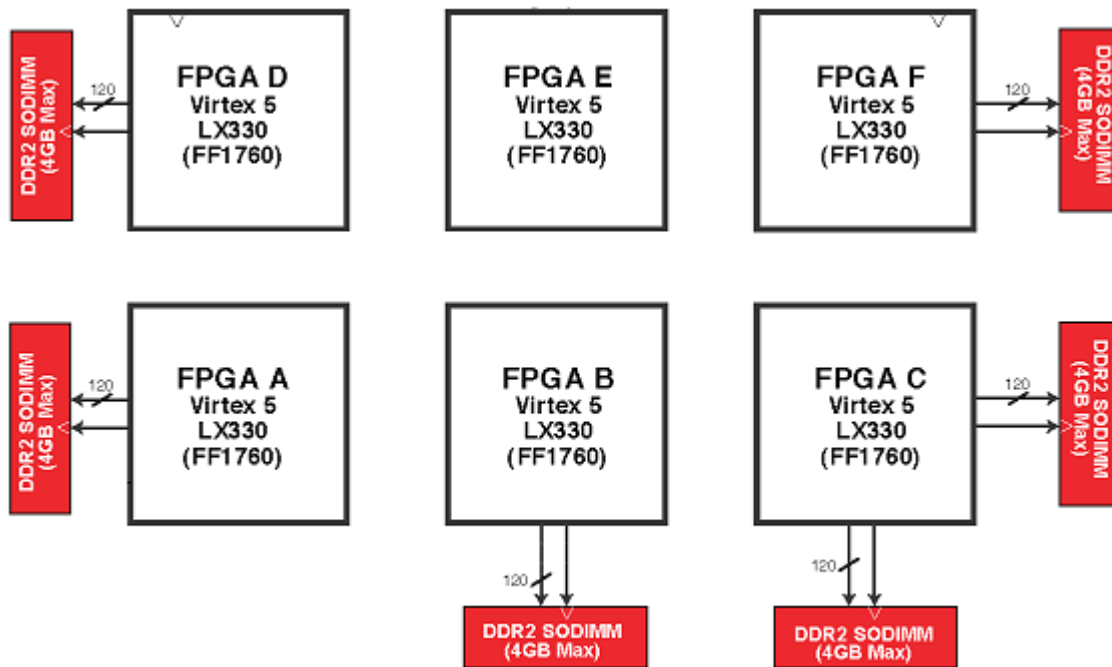
HARDWARE



LED Reference Designator	LED Color	Signal Name	The LED indicates the following when ON.
DS87-90	RED	MCU	No meaning.
DS154	YELLOW	HOST_ACT	No meaning.
DS150-153	GREEN	LED_SPARTAN	No meaning.
DS80	RED	ERR_TEMP	No meaning.
DS81	RED	ERR_CONF	No meaning.

16 DDR2

There are six “DDR2” memory socket interfaces on the DN9000K10PCIE4GL. The connections of these interfaces are shown in the block diagram below.



By convention, the names of these interfaces are DIMMA, DIMMB, DIMMC1, DIMMC2, DIMMD and DIMMF. The letter in the name indicating which FPGA it is connected to. FPGA C has two DIMM interfaces, C1 and C2. C1 is located along the bottom edge of the board, DIMM C2 is located along the right edge.

In this section, the interfaces may be called “DIMM” or “DDR2” interface interchangeably.

Signal names given in this section, and in other documentation (ucf files) are given in the form DIMM_<signal name>. The actual name of the signal on the schematic will replace the “DIMM_” in the signal name with “DIMMA_”, “DIMMB_”, “DIMMC1_”... etc depending on which DIMM interface the signal is used for.

16.1 Power

Each DDR2 SODIMM is capable of drawing 5A of current when in auto-precharge mode. Since the DN9000K10PCIE4GL is not capable of providing this amount of current for all six DIMMs simultaneously, you should not use auto-precharge on any of the SODIMM interfaces. Using this restriction, each DIMM will draw a maximum of 1A when running at 200Mhz.

16.1.1 Interface Voltages

The “standard” DDR2 interface voltage is +1.8V. The banks that connect to the DIMM interface are powered by 1.8V, and the power pins on the socket is connected to this same power net. In a DDR2 interface, these signals are driven using the SSTL18_DCI drive standard. There are some exceptions, listed below.

DIMM_SDA, DIMM_SCL, DIMM_CK2

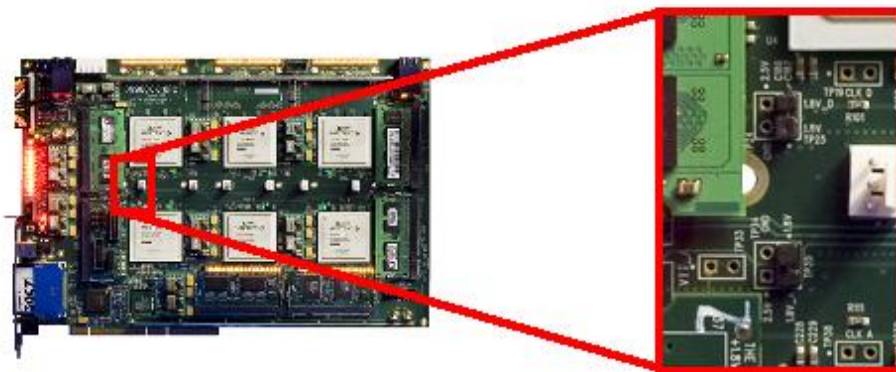
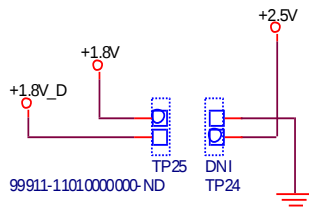
HARDWARE

These signals are connected to a 2.5V clock bank on the FPGA. DIMM_SDA and DIMM_SCL should be driven using the LVC MOS25 standard. For details on the DIMM_C2 signal, see the clocking section below.

The DIMM interfaces are not designed for hot-plug.

16.1.2 Changing the DIMM voltage

If you need to change the voltage of the DIMM interface, there is a set of jumper points provided for each interface allowing power to be redirected from a source other than the on-board 1.8V power supply. When the DN9000K10PCIE4GL is shipped, a jumper is installed connecting the DIMM/FPGA Bank power to the 1.8V power rail. Next to each of these jumpers is a 2.5V test point suitable for jumper-ing to the DIMM power rail, if necessary. Some Dini Group products (DNSODM_SDR, DNSODM_DDR1) require this jumper to be installed. When installing this jumper, remove the 1.8V jumper to prevent shorting 1.8 and 2.5V supplies together.



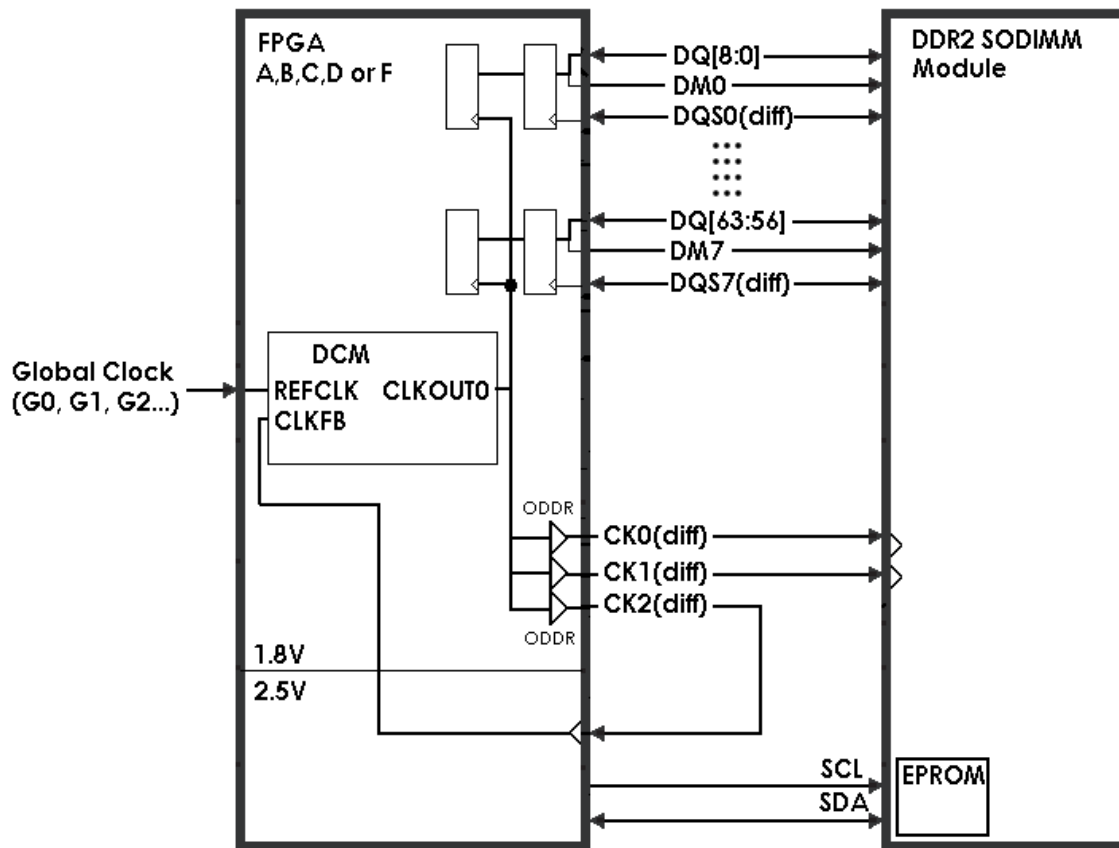
For example, to change the DIMMD interface to 2.5V, remove the jumper installed in TP25 and install a jumper from TP25.2 to TP24.1

16.2 Clocking

The data signals in the DDR2 interface are clocked source-synchronously. In order to clock in and out the “DQ” data signals, the DQS signal is used as a clock using the Virtex-5 “BUFIO”

clock driver. Details on how to implement a DDR2 controller are in the Xilinx application note XAPP858. You can also see the provided DDR2 reference design for example code.

A basic block diagram of the clocking is given below.



Note that the DIMM_CK2 signal is driven by the FPGA from a 1.8V bank. The output should be a DIFF_SSTL18. It is received by a global clock ("GC") pin on the Virtex-4 device. To receive the signal, use an LVDS_EXT input with DIFF_TERM attribute set to TRUE.

The CK0, CK1 and CK2 signals are length-matched, so this input should be synchronous to the clock input of the DIMM module.

The DQ and DM signals are synchronous to the DQS signals in each bank. See the DDR2 SODIMM module specification for information on the timing of this interface.

DQS timing

In order to clock the DQ and DM inputs using the DQS signal, you must use a BUFIO clock buffer on the DQS signal.

16.3 Signaling

16.3.1 Standards

DQ, and DM signals should use the SSTL18_II_T_DCI drive standard. The required VREF, VRP and VRN connections required for this standard are provided on all DIMM interface banks.

DQS signals should use the DIFF_SSTL18_II drive standard. External differential termination is provided on these signals at the FPGA.

DDR2 clock signals should be driven by the DIFF_SSTL18_II standard.

DDR2 “Control” signals (Address, BA, S#, RAS#, CAS#, WE#) should be driven by the SSTL18_I_DCI standard. The following signals are exceptions to this requirement. On four of the DIMM interfaces, external termination resistors are provided. The signals with external termination are listed below.

DIMMA_A00	DIMMC2_A00	DIMMD_A00	DIMMF_A00
DIMMA_A01	DIMMC2_A01	DIMMD_A01	DIMMF_A01
DIMMA_A02	DIMMC2_A02	DIMMD_A02	DIMMF_A02
DIMMA_A03	DIMMC2_A03	DIMMD_A03	DIMMF_A03
DIMMA_A04	DIMMC2_A04	DIMMD_A04	DIMMF_A04
DIMMA_A05	DIMMC2_A05	DIMMD_A05	DIMMF_A05
DIMMA_A06	DIMMC2_A06	DIMMD_A06	DIMMF_A06
DIMMA_A07	DIMMC2_A07	DIMMD_A07	DIMMF_A07
DIMMA_A08	DIMMC2_A08	DIMMD_A08	DIMMF_A08
DIMMA_A09	DIMMC2_A09	DIMMD_A09	DIMMF_A09
DIMMA_A10	DIMMC2_A10	DIMMD_A10	DIMMF_A10
DIMMA_A11	DIMMC2_A11	DIMMD_A11	DIMMF_A11
DIMMA_A12	DIMMC2_A12	DIMMD_A12	DIMMF_A12
DIMMA_A13	DIMMC2_A13	DIMMD_A13	DIMMF_A13
DIMMA_A14	DIMMC2_A14	DIMMD_A14	DIMMF_A14
DIMMA_A15	DIMMC2_A15	DIMMD_A15	DIMMF_A15
DIMMA_CAS#	DIMMC2_CAS#	DIMMD_CAS#	DIMMF_CAS#
DIMMA_CS#0	DIMMC2_CS#0	DIMMD_CS#0	DIMMF_CS#0
DIMMA_ODT0	DIMMC2_ODT0	DIMMD_ODT0	DIMMF_ODT0

For signals in this list, use the SSTL18_II drive standard.

16.3.2 Serial Interface

The SDA and SCL interfaces are connected to 2.5V LVCMOS buffers. External pull-ups are provided on these signals. The address of all six DIMMs on the DN9000K10PCIE4GL is set to zero.

16.3.3 Timing

The length matching of the DDR2 interface signals includes all signals except for DIMM_SCL and DIMM_SDA signals.

Due to the source-synchronous clocking techniques used by the DDR2 interface, the delay from FPGA to DIMM should not be needed, but is provided here anyway.

	DIMMA	DIMMB	DIMMC1	DIMMC1	DIMMD	DIMMF
Length	85mm	81mm	85mm	77mm	77mm	77mm
Delay	535ps	510ps	535ps	485ps	485ps	485ps

The trace impedance to each of the connectors is controlled to 50-ohms. All signals in the interface are ground-referenced. Note that this is contradictory to the recommendations of the DDR2 SODIMM specification.

To increase the setup time available for control signals, modules may be set into T2 mode. In the reference design, the modules are in T1 mode.

Address Control signals:

FPGA:

Assume a DCM in system-synchronous mode.

Worst clock-to-out time of Virtex 5 : 3.37 with DCM. No phase-shift.

Worst setup time: 0.097

Worst hold time: 0.21

DIMM:

setup 600ps

hold 600ps

DQ signals:

DIMM:

DQS must be within 350ps of DQ, DM

setup 400ps

Hold 400ps

FPGA:

IDELAY

setup -1.23

hold 2.14

clock-to-out 5.34

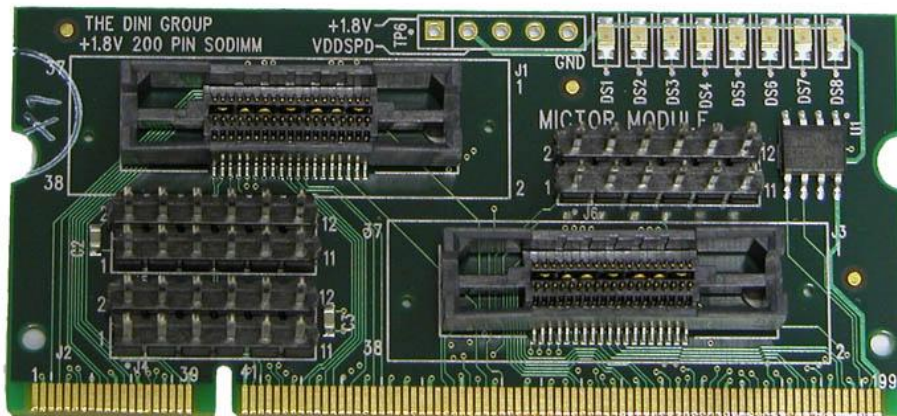
16.4 Compatible Modules

The DDR2 interfaces are compatible with standard PC2-2700 or faster memory modules up to a capacity of 4GB. The greatest capacity modules available at print time are 2GB. The interface has been tested with modules with a CAS latency of 3. The interface is characterized to 250Mhz,

HARDWARE

although faster designs may be possible. Xilinx is advertising a maximum DDR2 interface for the Virtex-5 of 333Mhz.

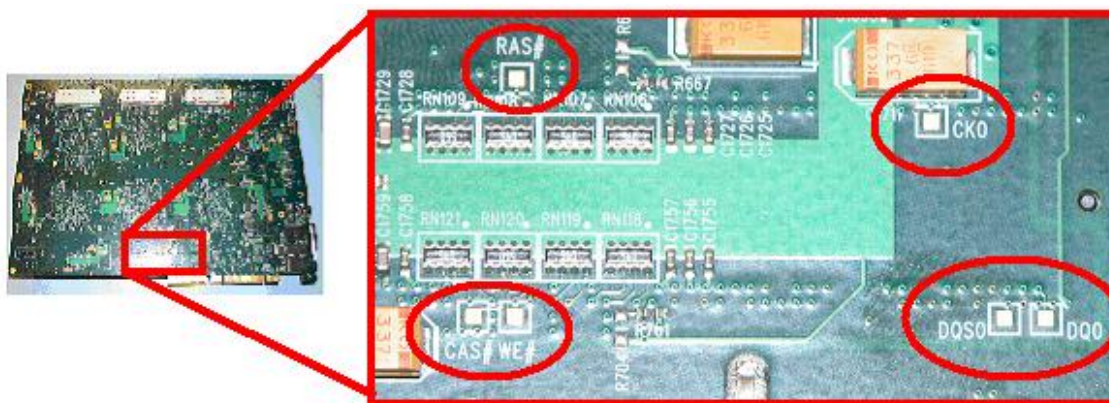
The DDR2 memory interface can also be used with SRAM, Flash and other types of memory modules. See the chapter on Ordering Options for a list of compatible memory modules.



The interface implementation on these modules is not provided. The customer must design the memory interface including timing and clocking.

16.5 Test points

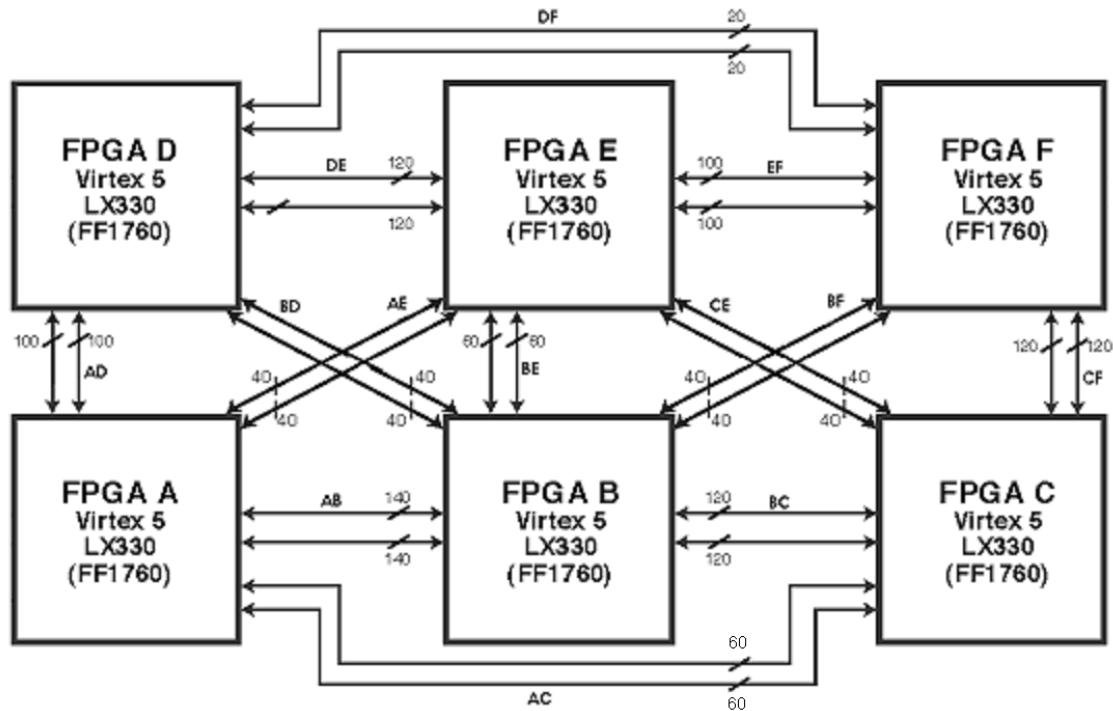
Each DDR2 interface exposes five signals as test points, located on the bottom of the PCB right under the SODIMM connector. These signals are DQ0, DQS0p, CK0p, RAS# and CAS#. The test points are labeled in silkscreen. The test points near DIMMA implicitly are part of the DIMMA interface, and so on.



17 FPGA Interconnect.

The point-to-point interconnect on the DN9000K10PCIE4GL is designed to operate at the maximum switching frequency possible on the DN9000K10PCIE4GL. The fastest switching standard available on the Virtex 5 FPGA is LVDS. Using this standard on the interconnect of a DN9000K10PCIE4GL, we have demonstrated switching frequencies as high as 950Mbs.

A block diagram of the point-to-point interconnect is below.



The interconnect in the above diagram is confusingly described as sets of two busses. Marketing explained why this was, but I forget the rationale now. "DE" is the bus between FPGA D and FPGA E. It contains 120 "p" signals and 120 "n" signals. This means there are 240 total signals between D and E. If you use LVDS and pair the "p" and "n" signals, you would have 120 LVDS signals between these two FPGAs.

The above diagram is only valid when the board is installed with only LX330 FPGAs (the largest available size). When any LX220 or LX110 FPGAs are installed, the interconnect available between FPGAs drops significantly. In the "Ordering Information" chapter of this manual, there is a block diagram showing the available features on a board loaded with LX220 or smaller FPGAs. If the board has mixed LX220 and LX330 FPGAs, then the interconnect available between any two FPGAs is the lesser of the signal counts shown in these diagrams.

Every interconnect bus on the DN9000K10PCIE4GL is guaranteed 650Mbs performance using LVDS signaling, using any speed grade FPGA. Performance up to 800Mbs per second is

HARDWARE

possible. High-speed grade FPGAs have not yet been tested due to availability, but interconnect switching speeds in excess of 1Gbs are expected.

Information on how to achieve this interconnect switching speed can be obtained by examining the provided MainTest reference design, or by reading the Xilinx application note XAPP855. Other methods of implanting high-bandwidth interconnect are described in XAPP860.

In a synchronous system between two FPGAs and a DCM in zero-delay mode, the following timing is possible.

Clock to Out	3.37	NS
Trace Delay	1.70	NS
Rise-time adjustment	0.30	NS
Clock skew	0.20	NS
duty cycle	0.05	NS
jitter	0.05	NS
setup time	1.00	NS
Min Period	6.67	NS
Max Frequency	0.15	GHZ

If LVDS is used, make sure to assign the DIFF_TERM attribute to the IBUFDS in the receiver FPGA.

As the frequency of synchronous communication between FPGAs increases, the user must implement more difficult techniques. As a general guide, these techniques are described below.

0	MHz	
20	MHz	The user should use the “Pack the IOBs” by using synthesis attributes. The output delay for each output and setup time for each input is a known value.
100	MHz	Use DCMs in each FPGA to eliminate the variation of clock network skew internal to each FPGA. The clock must be free-running
150	MHz	Use the phase-adjustment feature of the DCM to use the optimal phase for transmit and receive clocks.
250	MHz	Use DDR clocking, and DDR IO buffers
300	MHz	Use source-synchronous clocking between FPGAs. The clock is driven with the data for each bus. The receiving FPGA uses the clock signal, received on a “CC” pin to clock the IOs in the bus. An IDELAY element on the CC pin input delays the clock with respect to the data by a fixed amount to allow some setup time.
550	MHz	Use the Virtex 5 build in ISERDES and OSERDES modules. Use Virtex 5 PLL devices to reduce cycle-to-cycle jitter on the clocks.
600	MHz	Individually de-skew each bit using IDELAY elements. Use a training pattern or hard-code the correct delay values for each input.
700	MHz	Use LVDS signal standard

HARDWARE

800 MHz dynamically de-skew each bit to account for temperature and voltage variation
1+ GHz use error-correcting encoding.

Note that for speeds above 550Mhz, you must use the ISERDES and OSERDES modules, adding latency to your interconnect. (At speeds greater than 500Mhz, there is more than one clock-cycle of latency in board trace delay alone).

For the maximum bandwidth, use single-ended signaling at 700Mhz. For single-ended signaling, an IOSTANDARD of LVCMOS25 is appropriate. Use a drive strength of 6mA or 8mA.

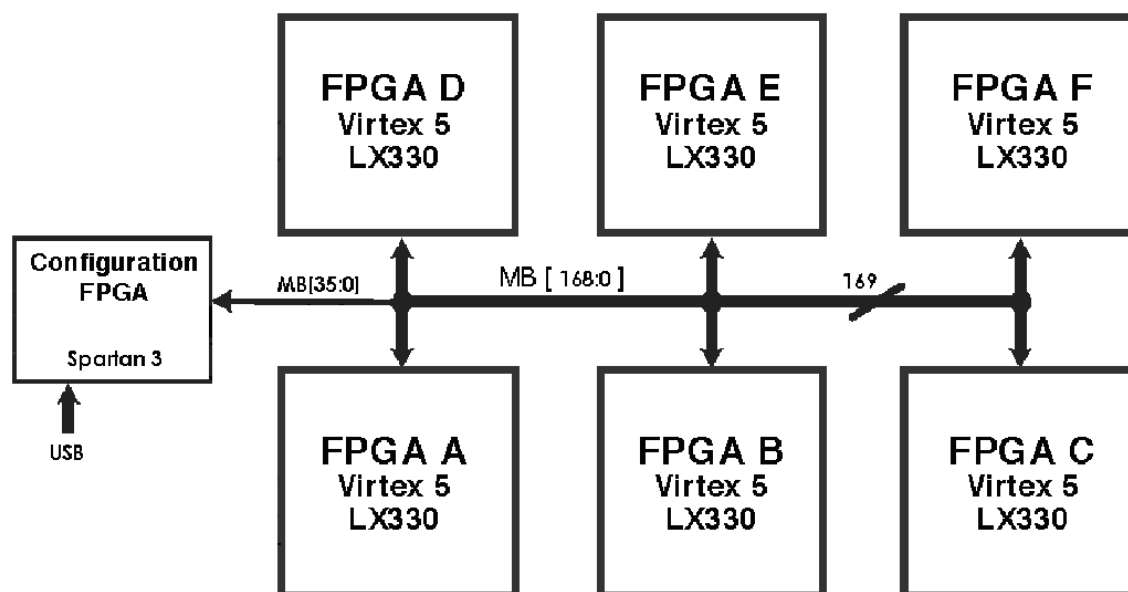
18 Main bus

Main Bus is the interface that the DN9000K10PCIE4GL uses to bring USB and PCI access to all six of the Virtex-4 FPGAs. If you want to use USB in your design, or want PCI access without implementing PCI in FPGA, then you must implement a Main Bus slave in your FPGAs. The reference designs include one such controller, and you are free to use it.

Drive strength.

18.1 MB Signals

The DN9000K10PCIE4GL, in addition to the dense interconnect available between FPGAs in a point-to-point topology, provides a 169-signal-wide “MB” bus that is connected to all six Virtex-4 FPGAs.



A subset of these signals (MB[35:0]) are used to implement Main Bus.) These 36 signals on the schematic have an underscore appendix on their net names describing their function in this interface.

18.1.1 Disambiguation

The “MainBus” has two meanings. In this document it usually refers to the interface connecting the FPGAs to USB and PCI via the configuration circuitry. It can also mean the group of 169 signals on the DN9000K10PCIE4GL that connects the six Virtex-4 FPGAs. It just so happens that the “MainBus” interface is implemented using 36 of the “MB* signals”. In this document “MB” will be used when referring to the signals themselves, and “MainBus” when referring to the Dini-Group-defined, 36-signal interface description.

18.1.2 Electrical

The MB signals are fixed at a 2.5V signaling level. LVCMOS25 is an appropriate signaling standard. Due to very heavy capacitive loads on the MB signals, you must use drive strength of 24mA to use main bus. DCI should not be used because the signals are not impedance-controlled. Although not required, by convention, data on the MB signals is synchronous to the MB48 clock. In order to use the “Main Bus” interface to communicate with USB or PCI, you must use the MB48 clock. This clock runs at a fixed 48Mhz.

Note that as well as the 169 “MB” signals, there are also 16 signals in the “Selectmap_D[15:0]” that connect to all six FPGAs that could be used for user data. Dini Group does not directly support using these signals. If you chose to use these signals, note that the FPGA design can interfere with the programming of FPGAs. You would have to keep the outputs on these signals tri-stated until all FPGA configurations are complete.

18.1.3 Timing

As described above, the MB signals are typically run synchronous to the 48Mhz MB48 bus. This is the highest speed that the MB signals are guaranteed to run using a system-synchronous clocking method. You may be able to achieve performance from FPGA-to-FPGA on this bus as high as 75Mhz, if you adjust input and output clocks and perform a timing analysis. Using LVCMOS25 with a drive strength of 24mA, you can assume there is a 10ns rise time/flight time for signals on this bus.

No length matching is done on the MB signals.

Virtex 5 clock-to-out time: 3.37ns with DCM

Virtex 5 setup time: 0.97ns

Flight time: 10ns (includes rise-time adjustment for capacitive load)

Total: 14.34ns (69 MHz)

The MB* signals are tested at 48Mhz

18.1.4 FPGA Stuffing Options

Note that the size of the MB bus is only 85 signals when one or more FPGAs installed are not an LX330. (i.e. LX110, and LX220 are only connected to the lower 85 MB signals)

18.2 Error Codes

The Main Bus interface has no way of signaling an error condition on read requests, but some errors will result in the same sentinel values being returned. Following is a list of these values.

0xABCDABCD: The Main Bus read timed out. (PCI only)

0xDEADDEAD : The Main Bus read times out (USB only). When this condition occurs, a register, accessible as part of the “configuration register” space, gets incremented. In this way, it is possible for a Main Bus access program to verify that a MainBus transaction has succeeded.

0xFFFFFFFF : The PCI bus timed out. This is not a value returned by the DN9000K10PCIE4GL. The PCI request was not returned. The QL5064 may not be configured correctly.

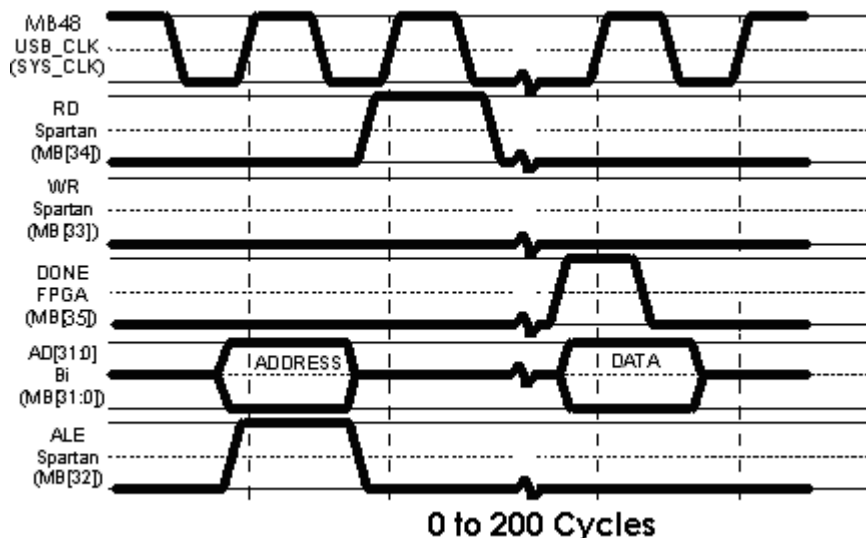
0xDEAD5566: This value is returned by the Dini Group reference design as a default value, when a read request is to an address that has no registers associated with it.

0xBABABABA: <unknown. Contact support>

0x12345678: The Main Bus is disabled. This is the default state of the DN9000K10PCIE4GL when it powers on. To set the DN9000K10PCIE4GL to enable, a configuration register must be written. This behavior is intended to protect users who do not wish to implement Main Bus interface, but who wish to use the MB0-MB35 signals for their own purposes.

18.3 FPGA Interface

All memory-mapped transactions in the reference design occur over the MB bus. This 36-signal bus connects to all Virtex 5 FPGAs and to the Spartan 3 configuration FPGA. The Configuration circuit (Spartan 3) is the master of the bus. All access to the MB bus (reads and writes) is initiated by the Spartan 3 FPGA when the reference design is in use.

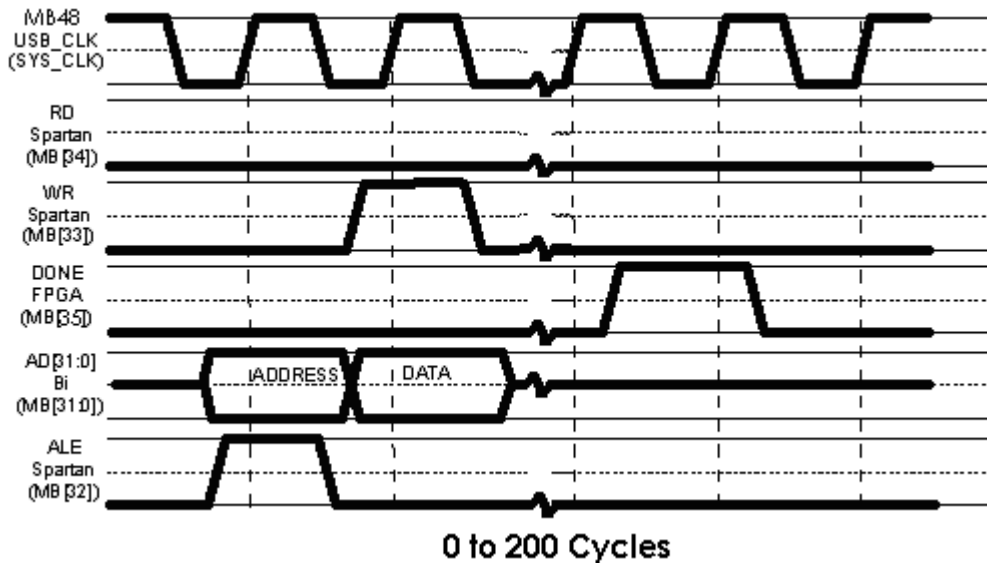


All transfers are synchronous to the USB_CLK (or SYS_CLK) signal. This clock is fixed at 48Mhz, and cannot be changed by the user. This clock is LVCMOS, single-ended. When the configuration circuit asserts the ALE signal, the slave device on the bus (the FPGA) is required

HARDWARE

to register the data on the on AD bus. This is the “main bus address”. All future transfers over the main bus are said to be at this address, until a new address is latched. On a later clock cycle, the master may assert the “RD” signal. Some time after this, (within 256 clock cycles), the FPGA should assert DONE for one clock cycle. On this cycle, the master (Spartan) will register the data on the AD bus, and that will be the read data. If DONE is not asserted, then a timeout will be recorded and the transaction cancelled.

Here is a write transaction:



When the Spartan asserts the “WR” signal, the FPGA should register the data on the AD bus.

Some time after this, the FPGA should assert the DONE signal. This will allow the Spartan to begin more transactions. The FPGA may delay this for up to 256 clock cycles before a timeout is recorded and the transaction is cancelled.

Main bus can be controlled from the USB Controller program. (Read and write single addresses, or to/from files) It can also be written from the main.txt configuration method. The main.txt syntax is

MAIN BUS 0x<address> 0x<data>

Where <address> and <data> are 8-digit (32-bit) hexadecimal numbers.

18.3.1 Conventional Memory map

By convention, FPGAs on the main bus interface are assigned address ranges. Assigning address ranges is required because the “FPGA sourced” signals (DONE) need to be driven by only one FPGA at a time.

The convention that Dini Group uses is to reserve the upper four bits in the address as an FPGA-select address. The address range (hex)
0x00000000 – 0x0FFFFFFF is reserved for FPGA A,
0x10000000 – 0x1FFFFFFF is reserved for FPGA B,
and so on.

The user need not follow this convention, but unless you really need 32-bit addresses, we recommend using it. Only one FPGA has “control” of the DONE signal. If the last address latched by ALE was not for a given FPGA, it should tri-state the output. Before tri-stating any signal with a pull-up or pull-down resistor, it is good practice to drive the signal to the DC value before tri-stating. (So that simulation will match emulation result).

19 Ethernet

Ethernet connections are available to FPGA D and FPGA F. Each of these FPGAs is connected to a Vitesse VSC8601 tri-mode Ethernet PHY. The RJ45 connector can be used to connect to a regular 10Base-T, 100Base-TX, or 1000Base-T Ethernet network connection.



The VCS8601 device does not contain an Ethernet MAC. The FPGA must implement a complete network stack to make use of the Ethernet connection. Sorry! I know that's retarded, and the DN10,000K10PCI will be better. Until then, check out OpenCores Tri-mode Mac controller

http://www.opencores.org/projects.cgi/web/ethernet_tri_mode/overview

19.1 MII

The 4-bit GMII interface is the only required interface on the PHY device. The EEPROM, MDIO, and other signals are only required if you want to put the PHY into a mode that is not default.

The SMI (MDC, MDIO signals) address is set to 0000. Each Ethernet interface (one for FPGA D, one for F) is on its own SMI interface.

19.1.1 Electrical

The appropriate electrical standard to use is LVDCI_25. In Gigabit mode (default), the MII interface runs at 125MHz, DDR.

The CLK_ETH125. Signal should use the SSTL_II_25_DCI signaling standard.

19.1.2 Timing

The board is designed such that when using a DCM in zero-delay mode on the clock, CLK125_ETH, the interface will meet timing, clocking all IOs on this clock. Alternately, you can use the CLK_ETH_RX to clock inputs using a BUFIO and clock CLK_ETH_TX on the same clock as the rest of your transmit signals.

By default, the 8601's internal clock compensation mode is enabled. This causes the timing of the device to be based on a clock that is delayed 2ns from the clock on the external TX_CLK and RX_CLK pins. This makes synchronous operation of the interface possible.

Length-Matched: 500ps

FPGA:

Assume a DCM in system-synchronous mode.

Worst clock-to-out time of Virtex 5: 3.37 (with DCM.) No phase-shift.

Worst setup time: 0.097

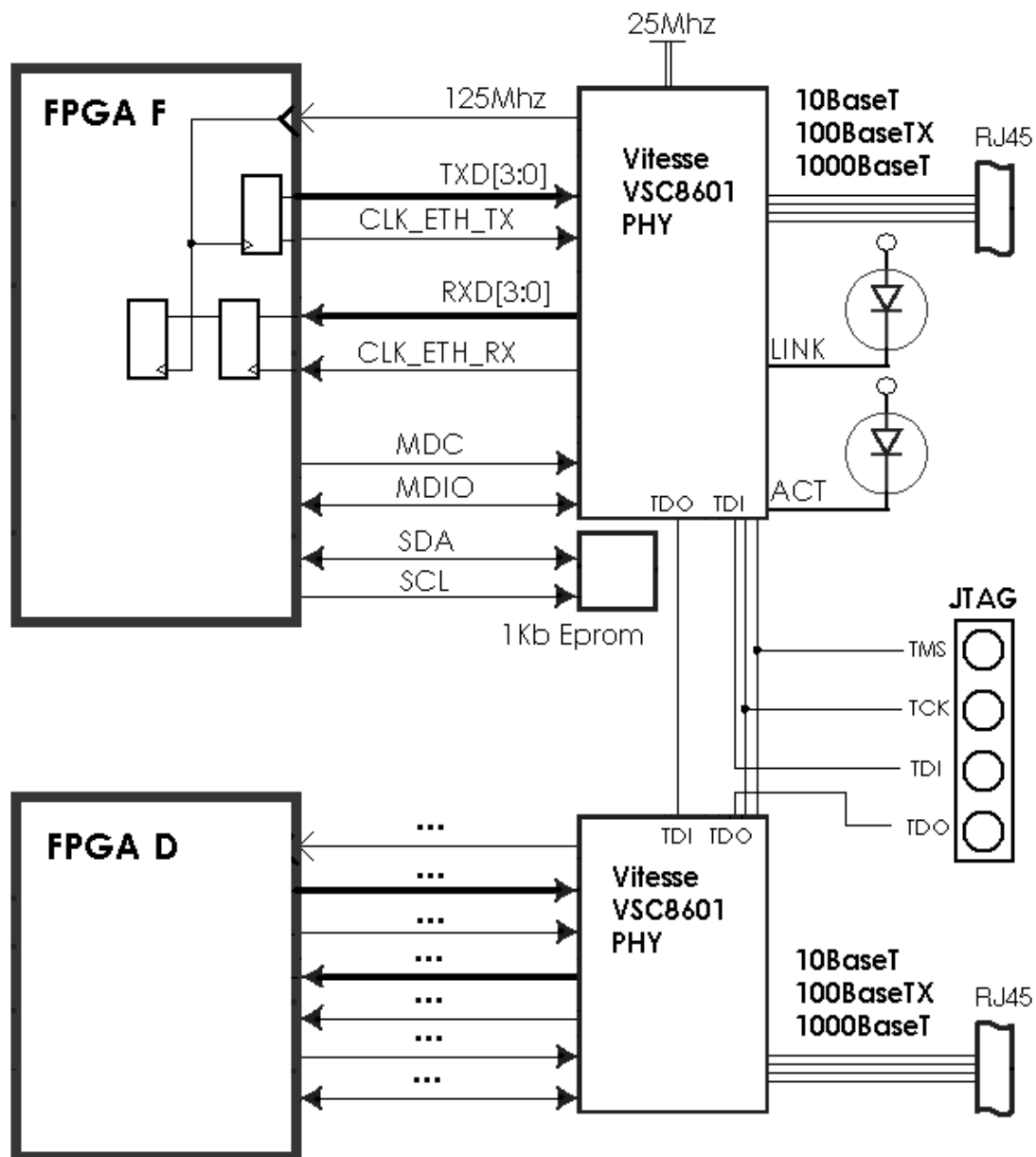
Worst hold time: 0.21

PHY: (clock measured at PHY pin)

clock-out 2ns

setup 2ns

valid: 1.2ns



The EEDAT and EECLK signals are intended to connect the PHY to an EPROM that would contain configuration settings for the device (LED behavior, MII timing, Link speed, duplex, auto negotiation, etc.). Since the MDIO interface is connected to the FPGA, it is unlikely you would ever use these signals, unless you just like emulating EPROMs on weekends and vacations.

If you do not implement the MDIO interface, then the default settings are used for the device. This includes settings that are specified by multi-level inputs connected to resistors.

HARDWARE

The CMODE options of the Ethernet PHYs has been set as follows

CMODE0 – 0100 (8.25K resistor)
 CMODE1 – 0000 (0 Ohm resistor)
 CMODE2 – 0001 (2.2K resistor)
 CMODE3 – 0000 (0 Ohm resistor)

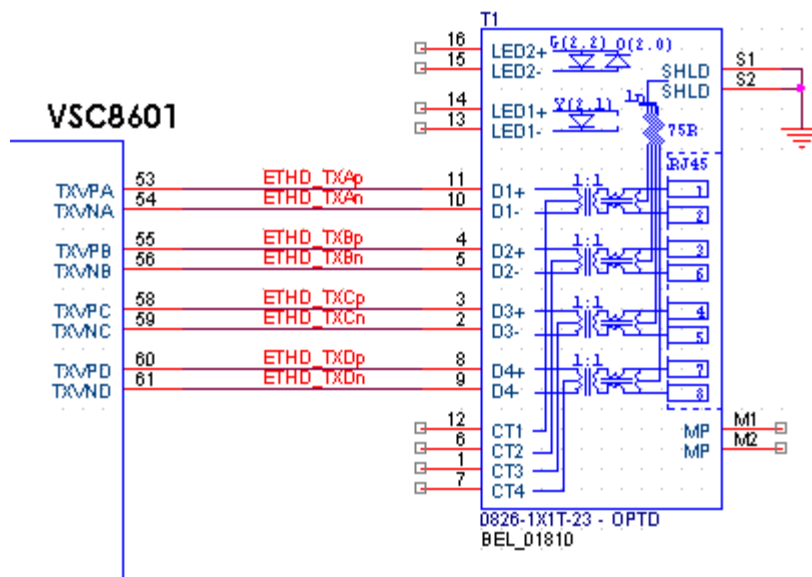
This results in the following settings

ADDR	=	00000	MDIO address
CLKOUT	=	TRUE	Drives the CLK_ETH_125 signal
PAUSE	=	00	I don't know
DOWNSHIFT	=	FALSE	I don't know
SPEED	=	00	Gigabit mode
ACTIPHY®	=	FALSE	
SKEW	=	11	ETH125 clock
MAC CALIBRATION MODE	=	00	????????

The LEDs on the RJ45 connector are controlled by the PHY. The Amber LED indicates activity and the Green LED indicates link in gigabit. The LED, DS64, located next to the RJ45 connector, indicates link in 100Mbit mode. The 10Mb link LED is not configured.

Hot plug is acceptable on a 1000Base-T connection.

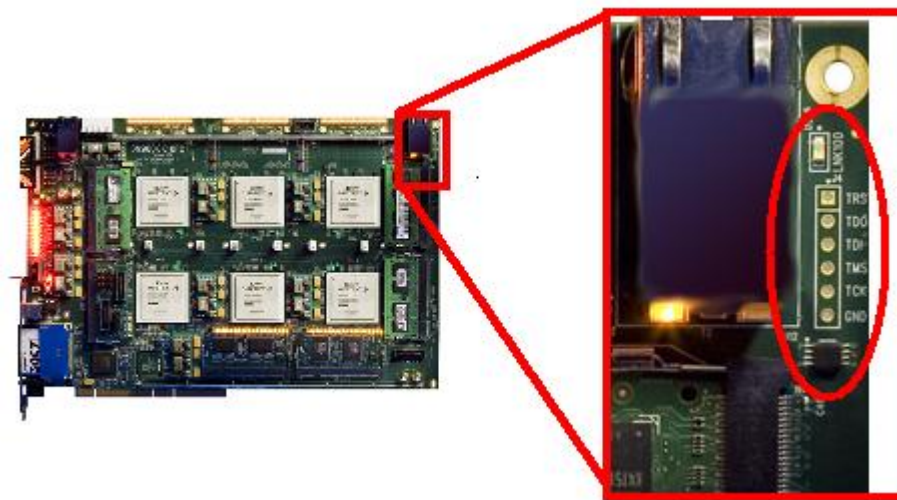
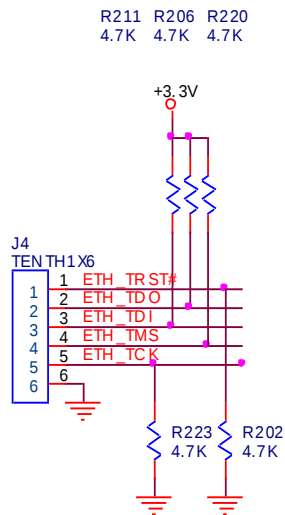
The Ethernet PHY works with the Xilinx Ethernet IP, but only in 10 and 100Mbit modes.



The above schematic clipping is useless but looks cool and technological.

19.2 JTAG

The two VSC8601 devices are attached to a JTAG chain. The schematic clipping showing this connection is given below.



I don't know why you would need access to this. It isn't tested or thought about ever. This JTAG chain does not connect to the FPGA JTAG chain.

The maximum speed of operation of this JTAG chain is 750 kHz.

20 EPROM

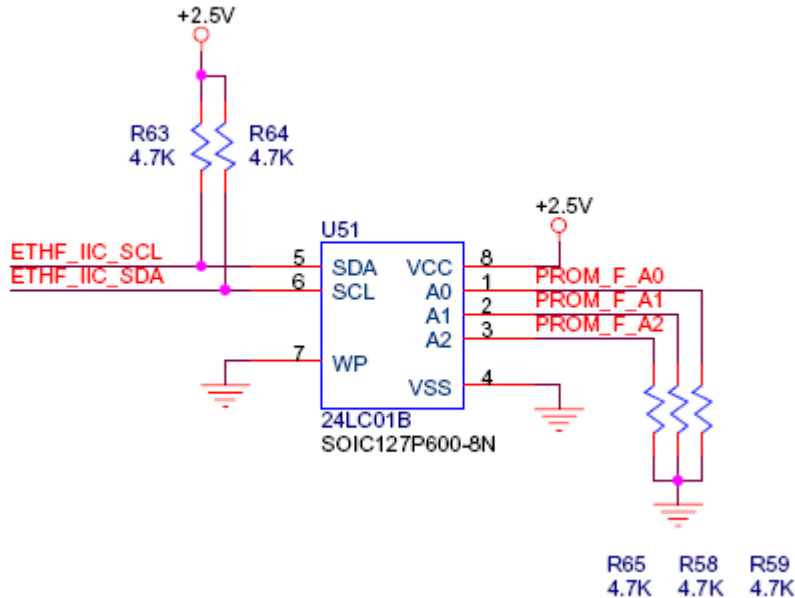
A small EPROM (64K) is attached to FPGA D and FPGA F. These devices are intended to store identification data for generating a unique MAC address for the Ethernet interfaces.

HARDWARE

However, the EPROM can be used for any user-defined purpose requiring static-memory intensive tasks, like remembering your name and birthday.

The interface to the EPROM is a standard IIC at 2.5V. The IIC address of the devices is (binary) 1010 000

The maximum clock speed of the IIC interface is 400 kHz



The pins used to access the EPROM are given below

ETHF_IIC_SCL E8
ETHF_IIC_SDA E7
ETHD_IIC_SCL R32
ETHD_IIC_SDA R33

21 Power

The power used by the DN9000K10PCIE4GL is derived from external 12V voltage supplies. The current at these voltages is supplied through P5, the pci express graphics power connector. The maximum power draws on each of these rails is given below.

21.1 12V

The 12V rail is used to generate all other voltages on the board. The only places where 12V is used directly are the daughtercards.

Below is a list of the maximum power draw of each of the 12V loads on the DN9000K10PCIE4GL.

HARDWARE

Rail	Max Current	Uses	5V current
1.0V_A	25	Internal FPGA power	2
1.0V_B	25	Internal FPGA power	2
1.0V_C	25	Internal FPGA power	2
1.0V_D	25	Internal FPGA power	2
1.0V_E	25	Internal FPGA power	2
1.0V_F	25	Internal FPGA power	2
1.8V	6	DIMM A,B,C1,C2,D,F	0.9
2.5V	25	Spartan 3 (1.2V)	2
		FPGA IO	
		FPGA Aux power	
Daughterboards	4		4
TOTAL			18.9

The total possible power requirement of the DN9000K10PCIE4GL is 18.9A on 12V (226W). This rate of power dissipation is well beyond the cooling capabilities of the heat dissipation provided. Therefore, the user must limit the power dissipated by his design. Typically, each FPGA would only use 10A, and daughterboards would use little or no power on 12V. Under these conditions, the 12V power requirement is only 8A (96W).

21.2 3.3V

3.3V is used by the DN9000K10PCIE4GL to supply the clock distribution network, the configuration logic (Micro controller and Spartan 3 FPGA), and daughterboard power.

The maximum power requirement for the DN9000K10PCIE4GL on 3.3V is 1A. 3.3V derived from 12V

21.3 2.5V

2.5V power is generated from the 12V using a 30A power supply.

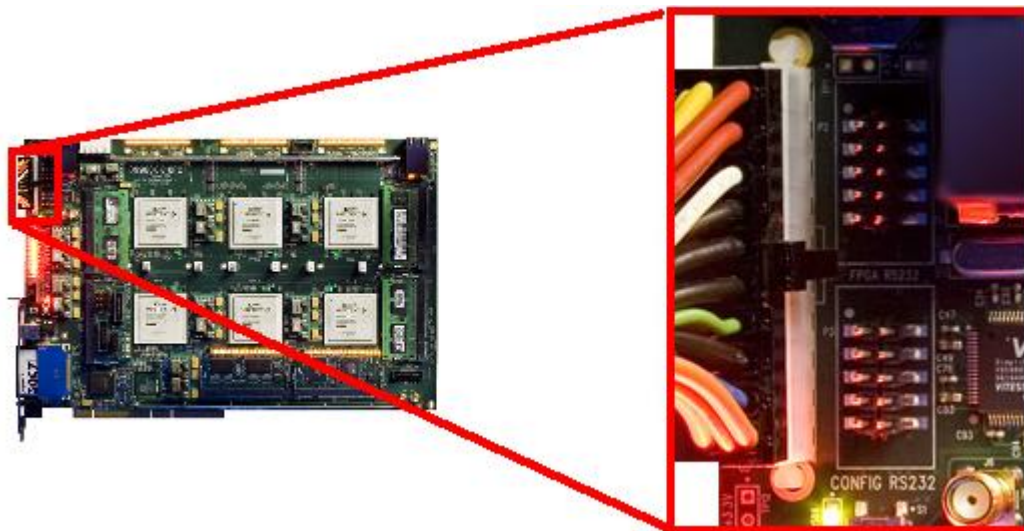
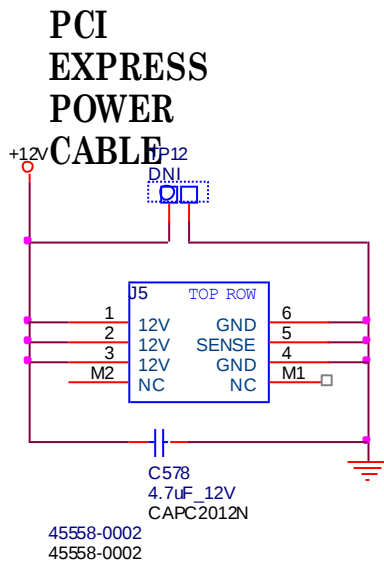
21.4 Ground

All ground (0V) voltages on the DN9000K10PCIE4GL are shared. A monolithic ground design strategy was used. The nets GND_SHIELD and GND_ANALOG are directly connected to the ground plane.

21.5 Voltage Regulation

21.6 Power Connections

The only source of power for the DN9000K10PCIE4GL is p5, the PCI express graphics power connector. No current is drawn from the PCI express slot. All voltages on the board are generated from 12V.



This connector will work only with a 6-pinned “PCI Express Graphics power” cable. Cables for this purpose are common on newer ATX power supplies. Many have multiple cables so that several DN9000K10PCIE4GL boards can operate in the same system. If you do not have this cable on your power supply, use the adapter cable that came with your board.

It's important not to attempt to plug anything other than a PCI express graphics power cable into P5. The pin-out of the 4- and 8-pin cable connectors is such that you will probably destroy the DN9000K10PCIE4GL if you try to connect it.

Power monitors (time constants)

21.7 Power Monitors

The DN9000K10PCIE4GL monitors the voltage levels on the board to ensure they are within tolerance. If they fall out of tolerance (above or below voltage) the board will enter a reset state. These tolerance ranges are listed below.

1.0V	(0.95 to 1.21)
1.8V	(1.65 to 3.00)
2.5V	(2.20 to 2.90)
3.3V	(2.89 to 4.00)
5.0V	(3.99 to 6.02)

The voltage monitors filter the voltage at a frequency of about 1KHz.

The following voltages are not monitored.

1.2, VCCO_B0, VCCO_B1, VCCO_B2, DIMM_VTT, DIMM_VREF

When a power supply voltage falls out of tolerance, the board is put in reset (the SYS_RST# signal is asserted), and SYS_RSTn LED glows, and an LED along the right hand side of the board will light to indicate which power rail has failed.

The 12V rail is monitored by the power supply modules. These modules will lock-out the generation of any other voltages unless the 12V rail is above 7V

21.8 Heat

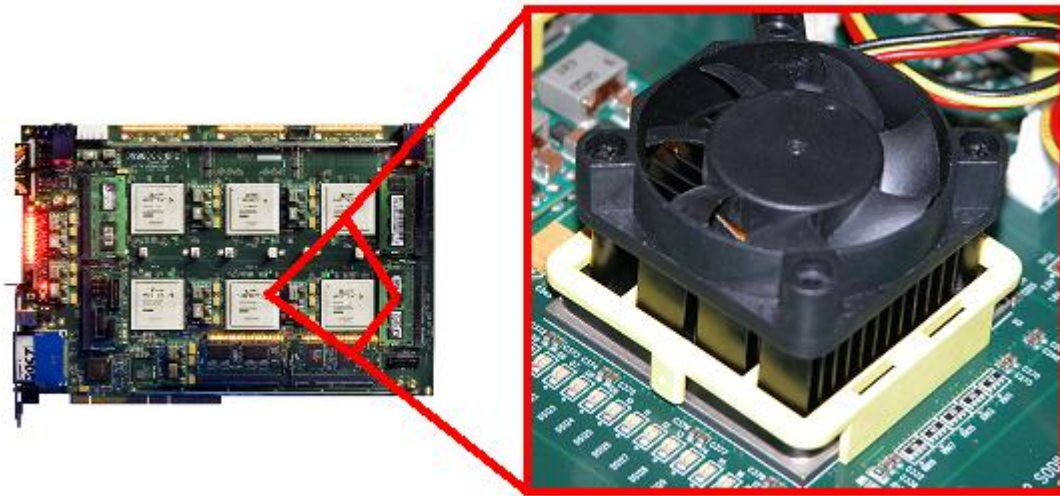
The maximum power dissipation supported for each FPGA is 25W. Using the provided heat sink and fan assemblies, FPGAs will remain under the maximum recommended junction temperature (85 degrees C). If your design exceeds this limit, you can assume the temperature of the device rises 2 degrees for each watt above this amount your design uses. Put this number in the settings of the timing analyzer.

Power requirements of a design can be estimated using the power estimator tool in ISE 9.1.

For this calculation the board is assumed to be in an ambient temperature of 35 degrees. In a closed computer case, the ambient temperature will increase. The ambient temperature in a server rack can vary from 35 to 60 degrees, and deployment may require special consideration.

21.8.1 Fans

The fan units attached above the heat sinks are powered by 5V. Each fan has its own power connector.

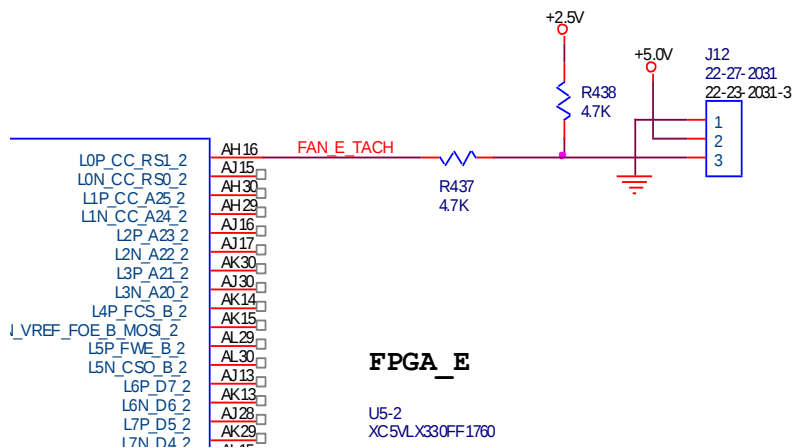


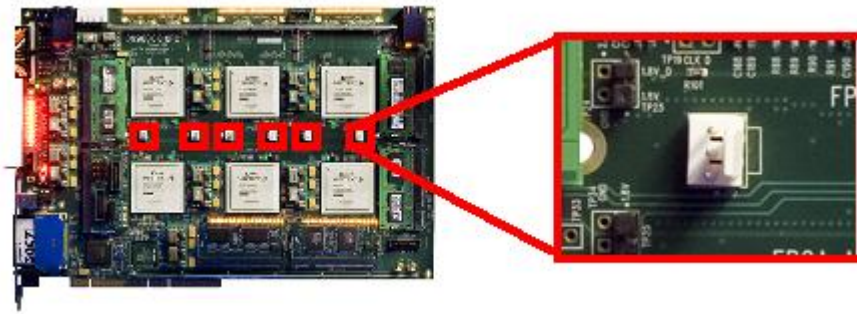
21.8.2 Removing Heatsinks

The heat sink/fan assemblies are attached using a plastic clip. There is a thermal interface material between the FPGA and heat sink that is slightly adhesive. Forcibly removing the heat sink will not damage the FPGA.

21.8.3 Fan Tachometers

Each FPGA fan has a tachometer connected to it for the detection of fan failure. If you intend to use this system in a rack or production system, you may want to monitor the fans.





The fan tachometer inputs (AH16) can be LVC MOS25. The fan will produce 2 rising edges per revolution. You may need to de-bounce the signal.

22 Connectors

This section provides a list of all connectors on the DN9000K10PCIE4GL. Items considered “test points”, including the “clock TP” points are listed in the test point section.

22.1 FPGA User Interface Connectors

The following connectors are directly connected to the FPGA, and the user needs to know the interface requirements in detail. All of these connectors should be fully described in the manual section indicated below.

FPGA Interface Connectors

Reference	Manufacturer	Part Number	Connector description	FPGA	Manual Section
J2, J3	Lighthouse	LTI-SASF546-P26-X1	SMA Jacks (differential)	E	Clocks
J25	AMP	2-767004-2	Mictor logic analyzer connector	ALL	Mictor
P5-2	FCI	84520102LF	MEG-Array 400-pin plug	D	Daughtercards
P6-4	FCI	84520102LF	MEG-Array 400-pin plug	E	Daughtercards
P7-4	FCI	84520102LF	MEG-Array 400-pin plug	F	Daughtercards
J18	JAE	MM50-200B2-1E	DDR2 200-pin SODIMM socket	A	DDR2
J22	JAE	MM50-200B2-1E	DDR2 200-pin SODIMM socket	B	DDR2
J23	JAE	MM50-200B2-1E	DDR2 200-pin SODIMM socket	C	DDR2
J17	JAE	MM50-200B2-1E	DDR2 200-pin SODIMM socket	C	DDR2
J9	JAE	MM50-200B2-1E	DDR2 200-pin SODIMM socket	D	DDR2
J8	JAE	MM50-200B2-1E	DDR2 200-pin SODIMM socket	F	DDR2
P2	Samtec	TSM-136-01-T-DV	Dual-row 0.1" RS232-header	ALL	RS232
J11	Molex	22-27-2031	3-pin Fan Power	A	Power

HARDWARE

J13	Molex	22-27-2031	3-pin Fan Power	B	Power
J15	Molex	22-27-2031	3-pin Fan Power	C	Power
J10	Molex	22-27-2031	3-pin Fan Power	D	Power
J12	Molex	22-27-2031	3-pin Fan Power	E	Power
J14	Molex	22-27-2031	3-pin Fan Power	F	Power
S2	Omron	B3S-1002	"Reset" Pushbutton	ALL	Reset

22.1.1 Comments

If you have a board with fewer than six FPGAs installed, connectors associated with the missing FPGA will be not be installed.

P2 Connections to this RS232 header are through a 12V buffer

J11-15 Pin 1: GND, Pin 2: 5V, Pin 3: Tachometer

22.2 Non-FPGA User Interface Connectors

The following connectors are not directly connected to FPGA IO, and therefore the user does not need to know detailed information about them. The interfaces relating to these connectors are functionally described in the manual section indicated. The FPGA indicated is used to access the connector's interface, but is not directly connected to the connector pins.

Non-User connectors

Reference	Manufacturer	Part Number	Connector description	FPGA	Manual Section
J7,J6	Lighthouse	LTI-SASF546-P26-X1	SMA Jack (differential)	ALL	Clocks
J20	Molex	538-53856-5070	CompactFlash socket		CompactFlash
J19	Molex	67068-8000	USB Type B female		USB
S1	Omron	B3S-1002	"Hard Reset" Pushbutton		Reset
J1	Molex	87832-1420	14Pos 2MM JTAG header	ALL	JTAG
X1	Keystone	3001	Coin Batter retainer		Encryption
T2	Belfuse	0826-1X1T-23-F	RJ45 w/ LEDs	D	Ethernet
T1	Belfuse	0826-1X1T-23-F	RJ45 w/ LEDs	F	Ethernet
J4	NONE	NONE	0.1" pitch mounting positions		Ethernet
J16	Molex	87832-1420	14Pos 2MM header		
P3	Samtec	TSM-136-01-T-DV	Dual-row 0.1" RS232 header		RS232

J5 Molex 4558-0002 6-pin Right angle "PCI Express" Power

22.2.1 Comments

If you have a board with fewer than 6 FPGAs installed, connectors with missing FPGAs as associated FPGA will also not be installed.

22.3 Not-For-Use Connectors

The following connectors are not intended for use by the user.

Not-for-use Connectors

Reference	Manufacturer	Part Number	Connector description	
Y3	Gompf	93340115	PCI Bracket	
TP35,45, TP47,37, TP25,27	Mill-Max	999-11-210-10-0	Jumper	1.8V connects to DIMMS
X2	AMP/Tyco	2-641260-1	8-pin DIP socket	Used by the DN9000K10PCIE4GL to store firmware data
MP2	CCI	B-250-6-1.5	Board Stiffeners	Board stiffeners
MP1	CCI	B-250-6-1.5	Board Stiffeners	Board stiffeners
M1-5	NONE	NONE	M3 Holes	Mounting Holes

22.3.1 Comments

Y3 The pins of Y3 are part of the “DN9000K10PCIE4GL mechanical drawing” in the

Mechanical section. The two pins are grounded.

JP1 JP1 can be installed to change the 1.8V power rail on the DN9000K10PCIE4GL to 2.5V

TP35 et al. These jumpers connect the DIMM interfaces to the 1.8V rail. They can be removed to change the DIMM voltage

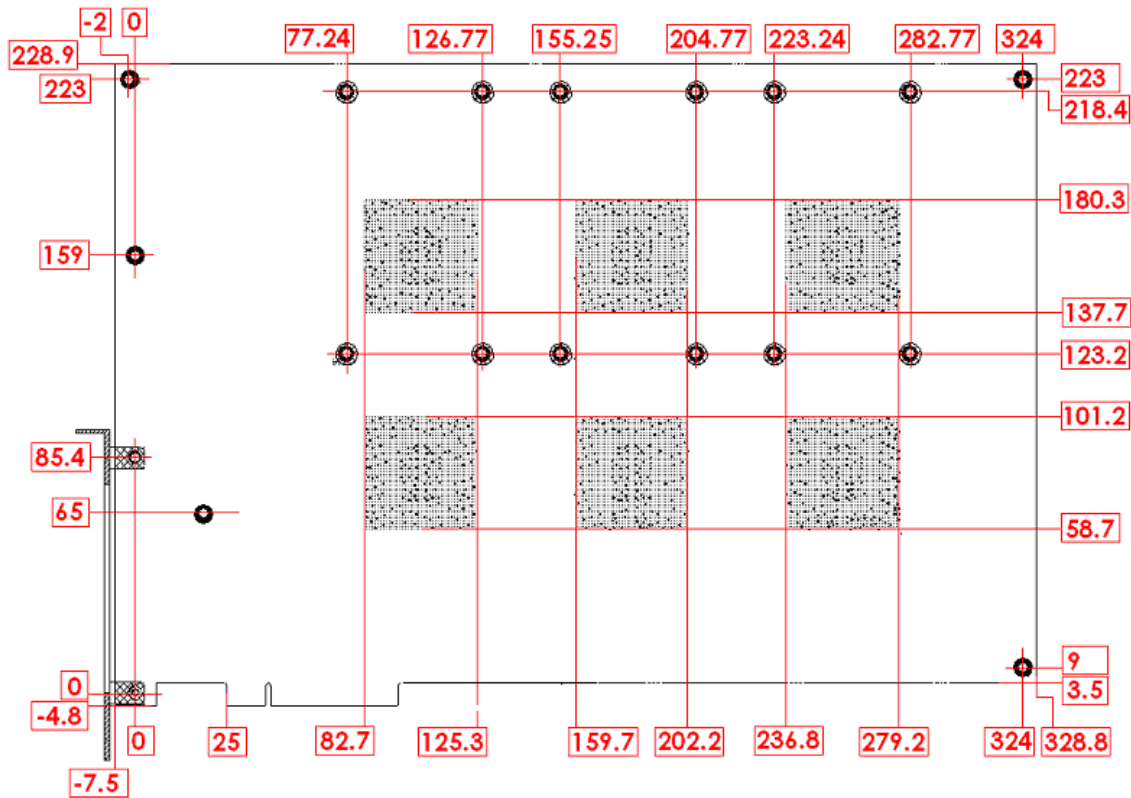
M1-5 These are connected to GND

23 Mechanical

The DN9000K10PCIE4GL is larger than the PCI specification allows, and is not guaranteed to fit into any ATX case. It will certainly fail to fit into a rack mount server enclosure. The vertical clearance with the fans installed and the ATX power connector not connector is 30mm. Lower-

HARDWARE

profile fans are available (14mm) but they may not have enough thermal performance for very power-hungry designs.



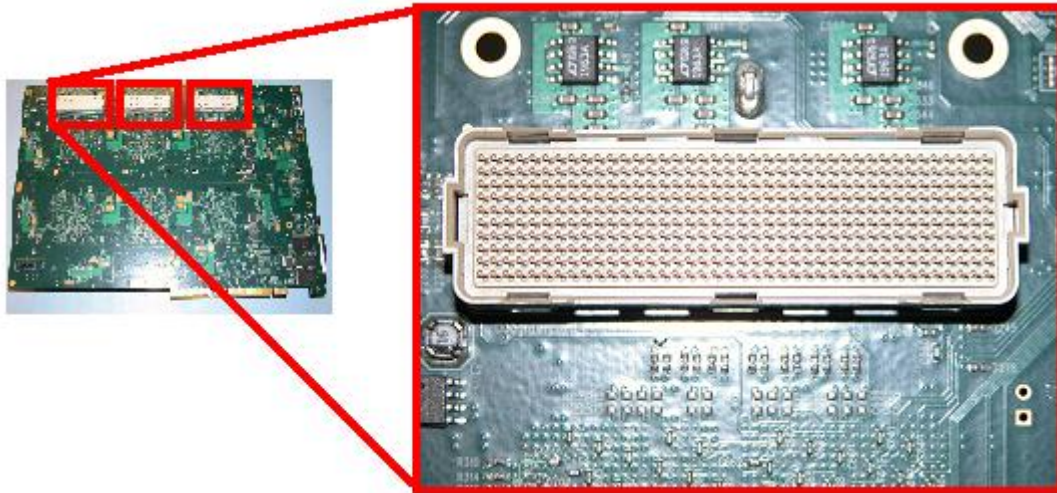
Mounting holes are all over the place. These are grounded.

Metal runners are along both edges of the board. These are for ground oscilloscope probe ground clips. You should also handle the DN9000K10PCIE4GL by its ground bars to help prevent ESD damage to the FPGAs.

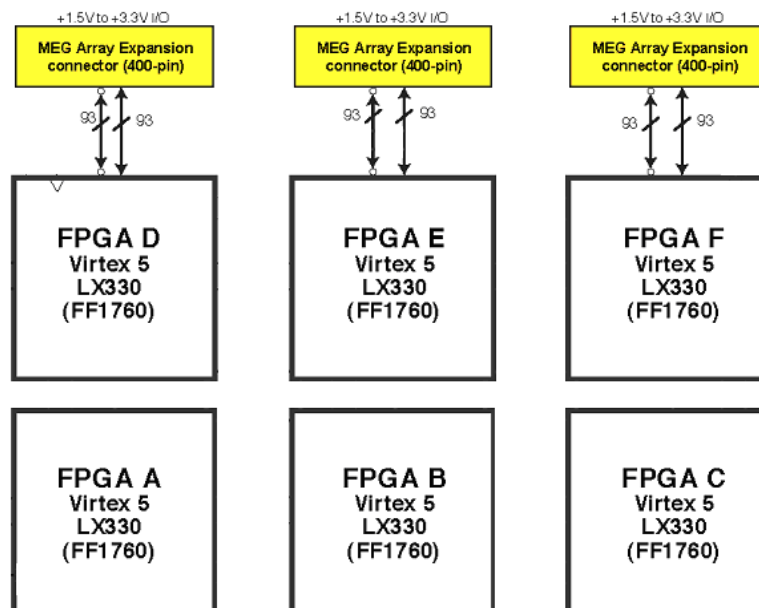


24 Daughtercard Headers

The daughter card expansion capability of the DN9000K10PCIE4GL is provided by three FCI ‘MEG-Array’ family connectors. Even though it uses the same FCI connector, it is NOT compatible with the 300-pin MSA standard.



Each daughtercard connector provides 186 signals to its associated FPGA. The signals can be used with just about any setting of IOSTANDARD, and can be used differentially.



The daughter card interface includes a 400-pin MEG-Array connector, made by FCI. The daughter card header is arranged into three “Banks”, correlating to the banks of IO on the

Virtex 5 FPGA.. Each of these banks connects to one or more “IO Banks” on the Virtex 5 FPGA. This allows three different sets of voltage or timing requirements to be met on a single daughter card simultaneously. Each Bank on the daughter card is 62 signals.

Other connections on the daughter card connector system include three dedicated, differential clock connections for inputting global clocks from an external source, power connections, bank VCCO power, a buffered power on reset signal.

24.1 Daughter Card Physical

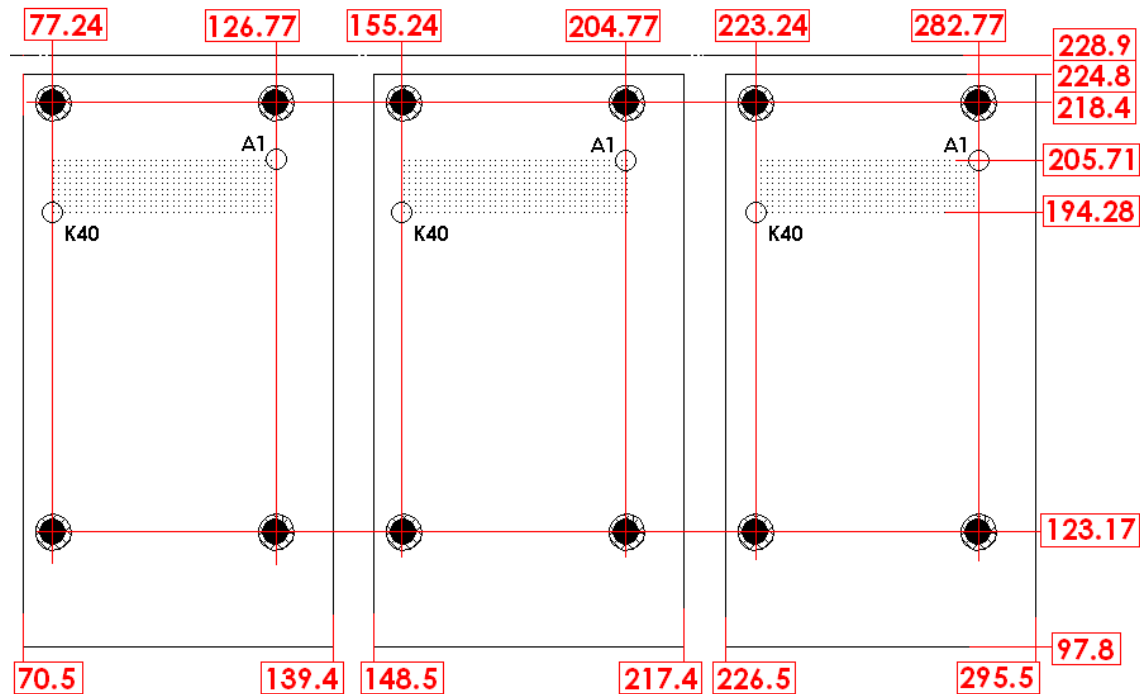
The connectors used in the expansion system are FCI MEG-Array 400-pin plug, 6mm, part #84520-102. This connector is capable of as much as 10Gbs transmission rates using differential signaling.

All daughter card expansion headers on the DN9000K10PCIE4GL are located on the bottom side of the PWB. This is done to eliminate the need for resolving board-to-board clearance issues, assuming the daughter card uses no large components on the backside.

The “Plug” of the system is located on the DN9000K10PCIE4GL, and the “receptacle” is located on the expansion board. This selection was made to give a greater height selection to the daughter card designer.

24.1.1 Daughter Card Locations and Mounting

The 400-pin daughtercard header is located on the bottom (solder) side near the right side of the board. Each MEG-Array header on a Dini Group product has four standard-position mounting holes. The drawing below shows the location of the daughter card header and it’s associated mounting holes.



This view of the DN9000K10PCIE4GL daughter card locations is from the top of the PCB, looking through to the bottom side. The Dini Group standard daughtercard, DNMEG_OBS400 is compatible with the DN9000K10PCIE4GL.

The mounting holes are designed to be used with 14mm, M3 standoffs. Dini Group has available appropriate mounting hardware on request:

Standoffs (Male-to-Female):

Harwin R30-3001402

(Mouser 855-R30-3001402)

“M3 x 14mm HEX 5mmA/F Harwin Metric Spacers RoHS: Compliant. Box/100”

Nuts:

LMI HN4600300

“M3 x 0.5mm

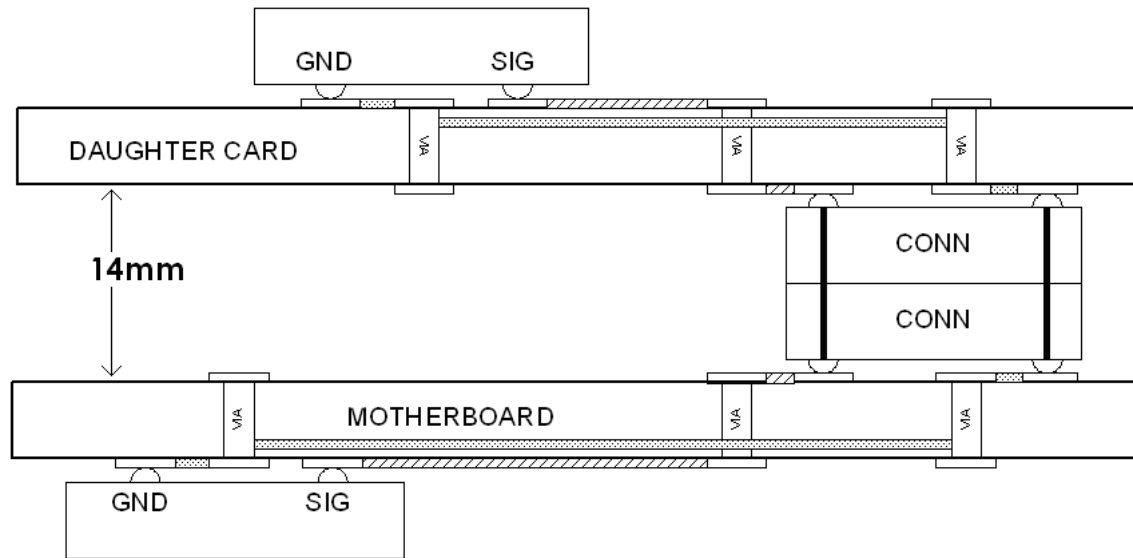
Bolts:

MPMS 003-0005-PH

(Digi-key H742-ND)

“SCREW MACHINE METRIC PH M3x5MM”

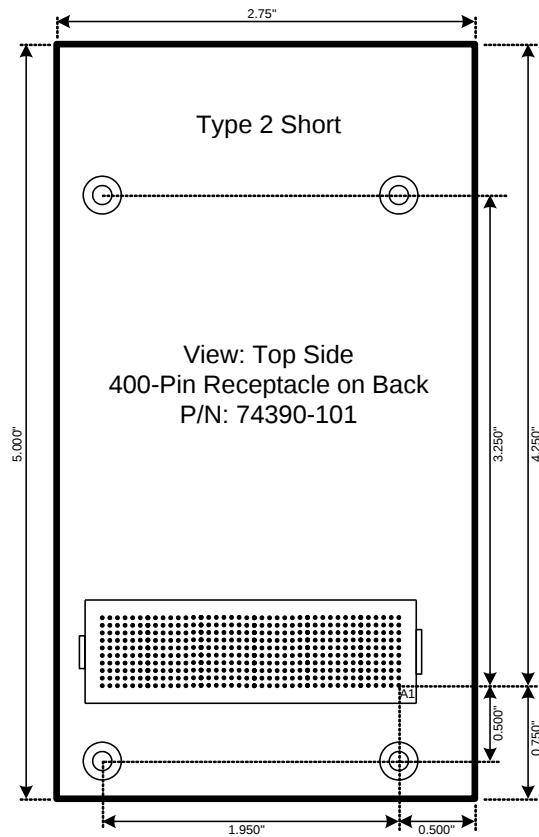
With this host-plate-daughter card arrangement, there is a limited Z dimension clearance for backside components on the daughter card. This dimension is determined by the daughter card designer’s part selection for the MEG-Array receptacle.



Note that the components on the topside of the daughter card and DN9000K10PCIE4GL face in opposite directions.

24.1.2 Types 2 Short (400pin Short)

The daughtercard mechanical provisions on the DN9000K10PCIE4GL are for a “type 2 short” daughtercard. The DNMEGOBS-400 is of these dimensions. This standard daughtercard provides headers and Mictor connectors for signal access to the DN9000K10PCIE4GL

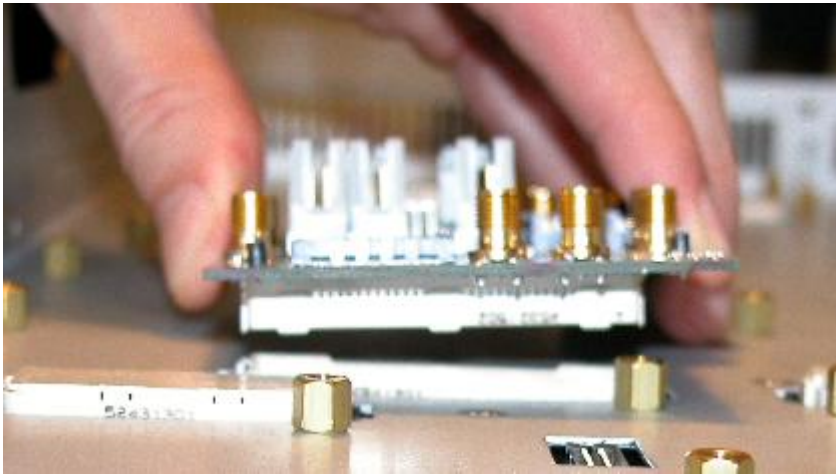


The mounting hole positions are standard, and the DN9000K10PCIE4GL has mounting holes to accommodate these holes. See the diagram in the section above.

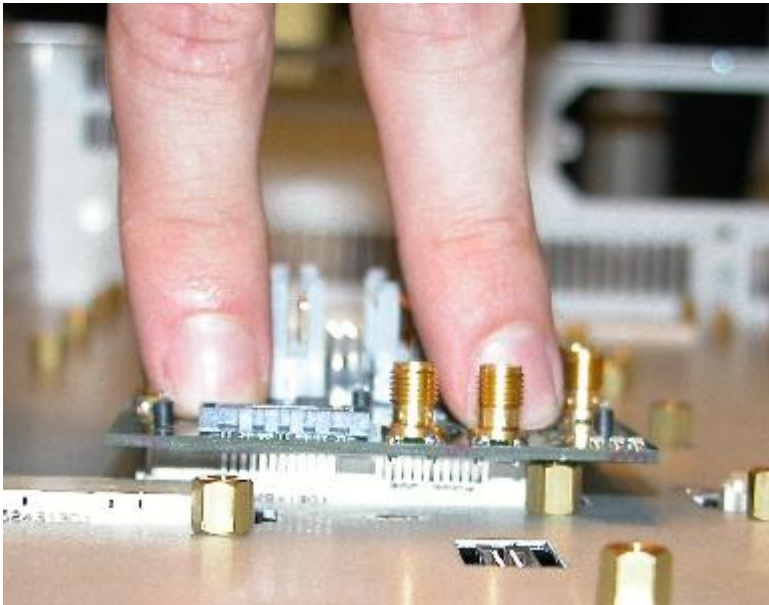
24.1.3 Insertion and removal

Due to the small dimensions of the very high speed Meg Array connector system, the pins on the plug and receptacle of the Meg Array connectors are very delicate.

When plugging in a daughter card, make sure to align the daughter card first before pressing on the connector. *Be absolutely certain that both the small and the large keys at the narrow ends of the Meg Array line up BEFORE applying pressure to mate the connectors!*



Place it down flat, then press down gently.



The following two excerpts are taken from the FCI application guide for the Meg Array series of connectors.

A part can be started from either end. Locate and match the connector's A1 position marking ("A1") for both the Plug and Receptacle. (Markings are located on the long side of the housing.) Rough alignment is required prior to connector mating as misalignment of >0.8mm could damage connector contacts. Rough alignment of the connector is achieved through matching the Small alignment slot of the plug housing with the Small alignment key of the receptacle housing and the large alignment slot with the large alignment key. Both connector housings have generous lead-in around the perimeter and will allow the user to blind mate assemble the connectors. Align the two connectors by feel and when the receptacle keys start into the plug slots, push down on one end and then move force forward until the receptacle cover flange bottoms on the front face of the plug.
Dec 09, 2004

Like mating, a connector pair can be unmated by pulling them straight apart. However, it requires less effort to un-mate if the force is originated from one of the slot/key ends of the assembly. (Reverse procedure from mating) Mating or un-mating of the connector by rolling in a direction perpendicular to alignment slots/keys may cause damage to the terminal contacts and is not recommended.

24.2 Daughter Card Electrical

The daughter card pin out and routing were designed to allow use of the Virtex 5's 1 Gbps general purpose IO. All signals on the DN9000K10PCIE4GL are all routed as differential, 50-Ohm transmission lines. Signals can be used as single-ended also. Proper electrical levels are explained in the VCCO section.

No length-matching is done on the PCB for daughter card signals, (except between two ends of a differential pair), because the Virtex 5 is capable of variable-delay input or output using the built-in IDELAY or ODELAY modules.

24.2.1 Pin assignments

The pin out of the DN9000K10PCIE4GL expansion system was designed to reduce cross talk to manageable levels while operating at full speed of the Virtex 5. The ground to signal ratio of the connector is 1:1. General purpose IO is arranged in a GSGS pattern to allow high speed single-ended or differential use. On the DN9000K10PCIE4GL (host), these signals are routed as loosely-coupled differential signals, meaning when used differentially, they benefit from the noise-resistant properties of a differential pair, but when used single-ended-ly, do not interfere with each other excessively.

All high-speed signals on the DN9000K10PCIE4GL, including daughter card signals, are routed against a ground potential reference plane. When creating a daughter card, it is recommended that these signals remain against a ground plane to maintain trace impedance.

The central columns of the connector pin out use a closely coupled, differential pair pin arrangement, which is uniformly surrounded by ground pins.

Below is a graphic representation of the pin assignments for the 400-pin connectors. Note that this is a view from the backside of the connector. The green boxes represent ground connections.

	A	B	C	D	E	F	G	H	J	K	
1	+12V		+5V		GCAP	GCAN		+5V		+12V	1
2		+3.3V		+3.3V			+3.3V		RSTn		2
3	B0 L1P		B0 L2P		GCBP	GCBN		B0 L3P		B0 L4P	3
4		B0 L1N		B0 L2N			B0 L3N		B0 L4N		4
5	B0 L5P		B0 L6P		GCCP	GCCN		B0 L7P		B0 L8P	5
6	VCC0 0	B0 L6N		B0 L6N			B0 L7N		B0 L8N	VCC0 0	6
7	B0 L9P		B0 L10P		B0 L27P	B0 L27N		B0 L11P		B0 L12P	7
8		B0 L9N		B0 L10N			B0 L11N		B0 L12N		8
9	B0 L13P		B0 L14P		B0 L28P	B0 L28N		B0 L15P		B0 L16P	9
10		B0 L13N		B0 L14N			B0 L15N		B0 L16N		10
11	B0 L17P		B0 L18P		B0 L29P	B0 L29N		B0 L19P		B0 L20P	11
12		B0 L17N		B0 L18N			B0 L19N		B0 L20N		12
13	B0 L21P		B0 L22P		B0 L30P	B0 L30N		B0 L23P		B0 L24P	13
14		B0 L21N		B0 L22N			B0 L23N		B0 L24N		14
15	B0 L25P		B0 L26P		B0 L31P	B0 L31N		B1 L1P		B1 L2P	15
16		B0 L25N		B0 L26N			B1 L1N		B1 L2N		16
17	B1 L3P		B1 L4P		B1 L27P	B1 L27N		B1 L5P		B1 L6P	17
18		B1 L3N		B1 L4N			B1 L5N		B1 L6N		18
19	B1 L7P		B1 L8P		B1 L28P	B1 L28N		B1 L9P		B1 L10P	19
20	VCC0 1	B1 L7N		B1 L8N			B1 L9N		B1 L10N	VCC0 1	20
21	B1 L11P		B1 L12P		B1 L29P	B1 L29N		B1 L13P		B1 L14P	21
22		B1 L11N		B1 L12N			B1 L13N		B1 L14N		22
23	B1 L15P		B1 L16P		B1 L30P	B1 L30N		B1 L17P		B1 L18P	23
24		B1 L15N		B1 L16N			B1 L17N		B1 L18N		24
25	B1 L19P		B1 L20P		B1 L31P	B1 L31N		B1 L21P		B1 L22P	25
26		B1 L19N		B1 L20N			B1 L21N		B1 L22N		26
27	B1 L23P		B1 L24P		B2 L25P	B2 L25N		B1 L25P		B1 L26P	27
28		B1 L23N		B1 L24N			B1 L25N		B1 L26N		28
29	B2 L1P		B2 L2P		B2 L26P	B2 L26N		B2 L3P		B2 L4P	29
30		B2 L1N		B2 L2N			B2 L3N		B2 L4N		30
31	B2 L5P		B2 L6P		B2 L27P	B2 L27N		B2 L7P		B2 L8P	31
32	VCC0 2	B2 L6N		B2 L6N			B2 L7N		B2 L8N	VCC0 2	32
33	B2 L9P		B2 L10P		B2 L28P	B2 L28N		B2 L11P		B2 L12P	33
34		B2 L9N		B2 L10N			B2 L11N		B2 L12N		34
35	B2 L13P		B2 L14P		B2 L29P	B2 L29N		B2 L15P		B2 L16P	35
36		B2 L13N		B2 L14N			B2 L15N		B2 L16N		36
37	B2 L17P		B2 L18P		B2 L30P	B2 L30N		B2 L19P		B2 L20P	37
38		B2 L17N		B2 L18N			B2 L19N		B2 L20N		38
39	B2 L21P		B2 L22P		B2 L31P	B2 L31N		B2 L23P		B2 L24P	39
40		B2 L21N		B2 L22N			B2 L23N		B2 L24N		40
	A	B	C	D	E	F	G	H	J	K	

	Special Signal
	Ground
	Power
	FPGA I/O
	Bank Power

Special purpose pins are described below.

24.2.2 CC, VREF, DCI

Some of the signals connected to the daughter card expansion headers are “clock-capable”; the inputs on the Virtex 5 FPGA can be used for source-synchronous clocking. In the schematic and customer netlist on the user CD, these pins contain a “_C” in the pin name.

Pins declared in the above diagram that are underlined are connected to “VREF” pins on the Virtex 5 FPGA. These FPGA pins are used to supply a voltage reference used as the threshold voltage for the signals on that bank. The use of these pins is only necessary when using threshold standards, such as SSTL.

DCI is used on all FPGA IO banks connected to a daughter card header. The reference resistance is 50 Ohms. Each Virtex 5 bank that is connected to a header DCI in enabled.

24.2.3 Global clocks

The daughter card pin out defines 6 clock output pins. These clock outputs are intended to be used a 3 differential signals (LVDS). Two clock signals GCA and GCB connect to the “GC” clock inputs on the FPGA. These clocks can be used only by the FPGA that is

associated with the header.

The GCC signal driven from each FPGA connects to a global clock buffer and can be used by all of the FPGAs on the DN9000K10PCIE4GL. (EXT0 and EXT1 networks). Since daughter

cards E and F share the same clock network (EXT1), only one of these two daughtercards can drive a global clock at one time.

24.2.4 Timing and Clocking

Signal from the FPGAs to the daughtercard connector are not length-matched. The maximum trace length on the DN9000K10PCIE4GL board for these signals is 800ps.

Each daughtercard has a global clock output pair “DCCLKCp/n”. This LVDS output is distributed on the DN9000K10PCIE4GL to all six Virtex-5 FPGAs. The clock buffer on the host board is designed to deliver the clock edge to all six FPGA synchronized with the CCLK pin on the daughtercard header. The daughtercard is expected to distribute clocks on it so that ICs on the daughtercard receive the clock signal synchronized with the pin on the daughtercard header. In this way, the host and daughter boards should be able to communicate synchronously with equal, large IO periods in each direction.

There are three methods of communicating FPGA-to-FPGA across the daughtercard interface.

Local Synchronous

The daughtercard generates a clock and drives it over the GCAp/n or GCBp/n clock pins to the host board FPGA. The daughtercard drives a synchronized clock to the logic on the daughtercard, adding 0.5ns delay to account for the trace delay on the DN9000K10PCIE4GL. The host FPGA will use a DCM in zero-delay mode, and the logic on the daughtercard should have a low clock-to-out and setup times. (or use a DCM). This method has the disadvantage of only allowing the one FPGA attached to the daughtercard to use this frequency. To communicate globally across the DN9000K10PCIE4GL, the user would have to pass the data across clock domains.

Global Synchronous

The daughter card generates a clock and drives it over the GCCp/n pins to the DN9000K10PCIE4GL host board. The user will select the daughtercard source for either the EXT0 or EXT1 networks as appropriate. The user sets the EXT0 or EXT1 network into zero-delay mode. (contact support@dinigroup.com). The disadvantage of this method is that the EXT0 or EXT1 network must be used, and that the zero-delay configuration has to be calculated for you by us. The advantage is that the entire system can be operated on a single clock domain.

Zero-delay on the DN9000K10PCIE4GL is allowed by enabling PLL devices (zero-delay buffers) connected to the GCC pins of each daughtercard header. To allow for a very wide range of clock frequencies sourced from the daughtercard, the PLL bandwidth of these buffers must be manually set. This can be done via USB, PCI or Compact Flash. The PLL can also be bypassed, allowing a global system-synchronous clock to be used without configuring this PLL. To use this method, the user will have to experimentally find the proper clock phase to use on the IO of the daughter card.

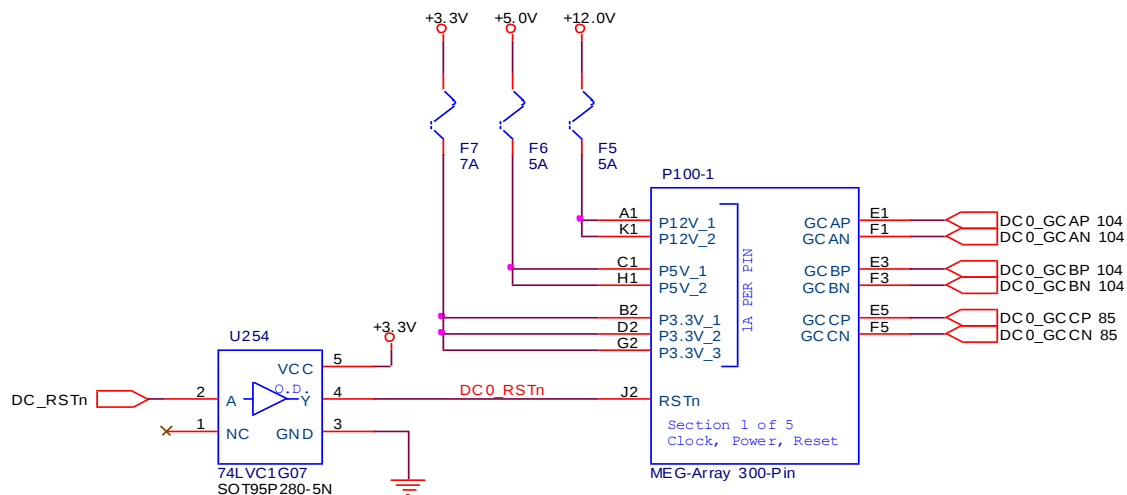
Source Synchronous

The daughtercard drives a clock into the CC pins of the daughtercard connector. This clock is

used to latch IOs. This method should be used for frequencies exceeding 150Mhz, because the phase-tolerance of the Virtex 5 FPGA and the clock buffer devices on the DN9000K10PCIE4GL EXT0 and EXT1 signals will prevent a reliable system-synchronous design at high speeds.

24.2.5 Power and Reset

The +3.3V, +5.0V and +12V power rails are supplied to the Daughter card headers. Each pin on the MEG-Array connector is rated to tolerate 1A of current without thermal overload. Most of the power available to daughter cards through the connector comes from the two 12V pins, for a total of 24W. Each power rail supplied to the Daughter card is fused with a reset-able switch. Daughter cards are required to provide their own power supply bypassing and onrush current limiting.



The RSTn signal to the daughter card is an open-drain, buffered copy of the SYS_RST# signal. It is also asserted when the User Reset is active. When RSTn is de-asserted, the +3.3V, +5.0V and +12V power rails are guaranteed to be within the DN9000K10PCIE4GL tolerance. If there are additional power requirements, the daughter card is required to ensure these.

24.2.6 VCCO Voltage

The daughter card is required to provide a voltage on the VCCO pin on the connector. This voltage is used on the DN9000K10PCIE4GL to power the FPGA IOs that are connected with that daughter card. In this way, the daughter card can control what voltage the interface will use.

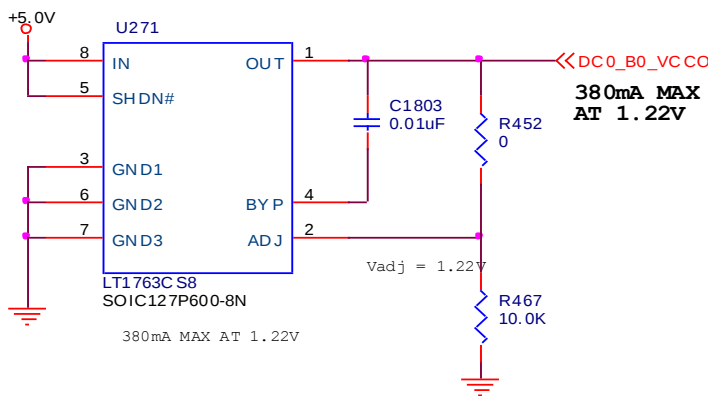
Each bank of the connector (B0, B1, or B2) uses a separate VCCO pin, and can have a different voltage applied to it. When designing a daughter card, you must determine the current requirements for the DN9000K10PCIE4GL and supply enough current capacity on these pins.

The VCCO voltage impressed by the daughter card should be less than 3.75 to prevent damage to the Virtex 5 IOs connected to that daughter card. Additionally, the voltage applied to the header pins from a daughtercard or external source, should be equal to or less than the VCCO

voltage of the bank that contains the IO. For example, a 2.5V daughtercard (one that uses 2.5V on each VCCO pin) should not drive a 3.3V signal onto the daughtercard pins.

24.2.7 VCCO bias generation

Since a daughter card will not always be present on a daughter card connector, a VCCO bias generator is used on the motherboard for each daughter card bank to keep the VCCO pin on the FPGA within its recommended operating range. The VCCO bias generators supply +1.2V to the VCCO pins on the FPGAs, and are back-biased by the daughter card when it drives the VCCO rails.



The output voltage of this regulator can be adjusted if needed. This will require changing the resistors on the ADJ pin of the regulators. The bias regulators can provide up to 1.5A of current. Some low-speed designs may not need more than this. Dini Group recommends placing the IO voltage regulators on the daughtercards, because this does not require modification of the DN9000K10PCIE4GL.

24.3 Rolling your own daughtercard

Small quantities of the connectors required for building a daughtercard can be obtained at cost from the Dini Group.

If you need help designing a daughtercard, we will be happy to review your schematic for errors. Send it.

25 Troubleshooting

25.1 The board is dead

If the board is not responding at all (when connected to a Windows XP computer, there is no “Dini Emulation Engine” in the hardware manager) the board may be stuck in reset. Check the power failure LEDs. If any of them are red, then the board is stuck in reset due to a power problem. If the failing voltage is 3.3V, 5V, or 12V, then the problem is probably caused by your

power supply. Check the voltages of these power rails and make sure they are within at least 5% of their nominal voltages. If the power supply was the one supplied by Dini Group, make sure that the voltage trim faceplate is connected. This faceplate allows trimming the 3.3, 5.0 and 12V outputs up and down for performance reasons. If the plate is not connected, all of the power supply's outputs default to their lowest settings. This will cause the Dini Group board to reset due to under-voltage.

If the board is not in reset, the RS232 terminal will be active. Connect a computer serial port to this header and open a terminal program on the computer. Start->Programs->Accessories->Communication->HyperTerminal is a suitable program. We use VanDyke software SecureCRT program, because it doesn't suck. Hopefully the RS232 configuration status dump will tell you exactly what the problem is. In any case, the Dini Group will need this capture to diagnose the problem.

25.2 The FPGAs won't program

First, connect the RS232 terminal and follow the instructions in the preceding paragraph. Usually, when an FPGA fails to program, the configuration section will detect the problem and print an error message to this terminal. Common problems the configuration section might report are:

- The syntax in the main.txt file is incorrect
- The bit file on the CompactFlash card is for the wrong type of FPGA
- The CompactFlash card is not formatted with a file system that the DN9000K10PCIE4GL can read

If the DN9000K10PCIE4GL reports about one or more FPGAs that "DONE did not go high", then there is a problem with the bit file. The bit file may have been generated using bitgen options that are not compatible with the DN9000K10PCIE4GL.

See if the FPGAs will configure using USB, PCI or JTAG.

When you contact Dini Group for support, we will need a capture of the RS232 terminal output.

25.3 My design doesn't do anything

Make sure that the clock your design uses is running. Output the clock to an LED and probe it with an oscilloscope.

Check the pinout in your constraint file. Check the .PAR report file to make sure that 100% of your IOBs used have LOC constraints. There is never a reason not to constrain an IO.

Use the .PAD report to make sure your constraints were all applied. Some situations may cause constraints to be ignored.

Double-check that the connections match between your FPGA pins and the daughtercard pins using the schematic.

If “MainBus” interface is not working, make sure that none of the other FPGAs are driving those MB pins.

Make sure that the "Unused IOBs" option in bitgen is set to "Float

Check for Timing errors in the timing report

Route the clock signal to a pin and observe it with an oscilloscope.

25.4 The DCMs won't lock

1) The DCMs are required to be set in a frequency mode compatible with the frequency of the reference clock input. Check the following attributes of the DCMs.

DFS_FREQUENCY_MODE

DFS_FREQUENCY_MODE

2) All clock inputs of the DCM are required to be stable for a certain number of microseconds before releasing the DCMs reset signal. If you are generating the reference clock from an FPGA (or another DCM), you will need to build a delayed-reset circuit to reset the second DCM.

3) Make sure the global clock you are using is being received with an LVDS receiver, not a single-ended one. Make sure the DIFF_TERM attribute is turned on. (especially is the problem is with high-frequency clocks)

25.5 The _____ signal on my board is going crazy on my oscilloscope

Make sure the ground clip is attached to the probe.

If there is an oscillation on the signal near 60Hz, there is a problem with the oscilloscope setup.

Capture the oscilloscope view and email it to support@dinigroup.com.

The Reference Design

This chapter introduces the DN9000K10PCIE4GL Reference Design, including information on what the reference design does, how to build it from the source files, and how to modify it for another application.

1 What the reference design does

1.1 Example usage of all interfaces

The reference design helps users by showing them how using each interface is possible. Code is provided as-is, and is intended as proof-of-concept on each interface advertised for the DN9000K10PCIE4GL product. The Dini Group warrants only that the DN9000K10PCIE4GL hardware is functional and usable. The interfaces that the Dini Group design exercises and provides examples for are:

- Access to the DDR2 SDRAM Modules At 250Mhz
- Known-good bitfiles for programming over USB, CompactFlash or JTAG
- RS232 Communication
- FPGA Interconnect, using IO Flip-flops
- FPGA Interconnect using advanced source-synchronous techniques
- Main Bus interface (for USB communication)
- Main Bus memory-mapping example (to DDR2, block RAM)
- Blink LEDs in cool patterns.
- Reset Button
- New internal Virtex-5 features (PLL, ODELAY, 550Mhz clocking, 800Mbs IO)
- Set global clocks
- Ethernet Hardware test
- PCI Express hardware test
- Mictor cable (for LVDS communication. Mictor also can be used for logic analyzers)

All source code for the reference design is included on the CD and may be used freely in customer development. Precompiled bit files for the most common stuffing options are also

included and can be used to verify board functionality before beginning development. A build utility, described in the section [Compiling The Reference Design](#), can be used to generate new bit files, or to generate bit files for less common configurations of the DN9000K10PCIE4GL.

The reference design was created using

Here are the default main.txt file lines.

```
// Main.txt file for the DN9000K10PCIE4GL

FPGA A: fpga_a.bit
FPGA B: fpga_b.bit
FPGA C: fpga_c.bit
FPGA D: fpga_d.bit
FPGA E: fpga_e.bit
FPGA F: fpga_f.bit

clock frequency: G0 350Mhz //LVDS frequency
clock frequency: G1 250Mhz //DDR2 frequency
clock frequency: G2 200Mhz //LVDS IDELAYCTRL

//MAIN BUS 0x10000000 0x00000001
//MEMORY MAPPED: 0xDF49 0x0
```

1.2 Reference Design Types

“The Reference Design” in this chapter refers to the FPGA designs located on the user CD at
D:\FPGA_Reference_Designs\DN9000K10PCIE4GL\MainRef\
D:\FPGA_Reference_Designs\Programming_Files\DN9000K10PCIE4GL\MainTest\

Four other self-contained designs are on the CD and described in this manual. These four designs are described in their own sections later in this chapter. The remaining sections describe the “MainTest” design. “MainTest”, “The reference design” and “The Dini Group reference design” are the same thing.

The four additional designs are

PCI Interface Design: Tests the 64-bit interface between FPGA A and the QL5064 (PCI)
LVDS Reference Design: Characterizes the FPGA interconnect using source-synchronous
Ethernet Reference Design: Tests the Ethernet PHY.

Other features of the board, such as memory sockets and daughtercard headers are tested using the Main Test.

1.3 Using the reference design

Setting the clocks, what the clocks do.

How to run the tests in USB Controller.

2 Reference Design Memory Map

The Dini Group reference design memory allows access to the main hardware features of the DN9000K10PCIE4GL to the host Main Bus interface. Main Bus can be accessed through PCI or USB. Main Bus is described in the Hardware chapter.

To read this chart:

- * Addresses are 32-bits.
- * Each address contains one 32-bit word.
- * Each FPGA has a base address. FPGA A will respond to all MB accesses in the range 0x00000000 – 0x0FFFFFFF. FPGA B will respond to accesses from 0x10000000-0x1FFFFFFF, etc.
- * The addresses given below are offsets from the base address of any given FPGA.
- * Base addresses are 0x00000000 for FPGA A, 0x100000000 for FPGA B, etc.
- * Some registers are not valid for all FPGAs.
- * Some addresses are not valid for all of the Dini Group's reference designs.
- * Some of the address bits are decoded as “Don't care” bits. Therefore, there may be multiple addresses on Main Bus that access the same registers.

Address Range	Register Name	Register Contents
0x00000000 - 0x07FFFFFF	DDR2	The data contained in the DDR2 SODIMM memory
0x08000001	DDR2HIADDR	The upper bits of DDR2 address (MainBus memory space is smaller than most DDR2 SODIMMs)
0x08000002	IDCODE	0x05000135
0x08000003	DDR2HIADDRSIZE	The number of valid addresses in DDR2HIADDR
0x08000004	INTERCONTYPE	An ID code used to identify which design is loaded 0x34561111 – Interconnect, Single 0x34562222 – Interconnect, LVDS 0x34563333 – Interconnect, LVDS (reversed) 0x34560000 – Any Other Design (PCI, Ethernet, etc.)
0x08000005	DDR2SIZE	A code to control how DDR2 memory is coded into MainBus memory

THE REFERENCE DESIGN

0x08000006	RWREG	Read/Write Scratch Register for testing
0x08000007	DDR2TAPCNT0	The current “tap” settings of the IODELAY elements in the DQ IO buffers on the DDR2 interface (lower bytes)
0x08000008	DDR2TAPCNT1	The current “tap” settings of the IODELAY elements in the DQ IO buffers on the DDR2 interface (upper bytes)
0x0800000A - 0x08000011		This range of addresses is reserved for manufacturing tests (Daughtercards)
0x08000012	SODIMM_SEL	For FPGA C, this controls which DDR2 interface is active
0x08000013	FAN_TACH	The current input value of the fan tachometer (0 or 1)
0x08000014	IS_LX_330	0x1 if the FPGA is an LX330, 0x0 is it is not.
0x0800001B	SODIMM_RANK	Data read from the SODIMM IIC interface
0x0800001C	SODIMM_COL	-
0x0800001D	SODIMM_ROW	-
0x0800001E	SODIMM_BANK	-
0x0800001F	SODIMM_CAS	-
0x08000021	CLK_COUNTER	Contains contents of G0 counter /4
0x08000022	CLK_COUNTER	Contains contents of G1 counter
0x08000023	CLK_COUNTER	Contains contents of G2 counter
0x08000024	CLK_COUNTER	Contains contents of CLK48 counter
0x08000025 - 0x08000032	RCLK_COUNTER	LVDS source-synchronous clock counters (LVDS design only)
0x08000033 - 0x0800003F	MCLK_COUNTER	Clock counters for (in backwards order!): DDR2 clock, EXTCLK0, EXTCLK1, SMACLK, CLK_FBE, CLK_FBB, CLK125_ETH, CLKP, CLK_TpP
0x08000040 - 0x08000043	DDR2TESTTAPCNT	Reserved for manufacturing tests (DDR2)
0x08000044	LED_OE	Controls LED output enables.
0x08000045	LED_OUT	Controls LED output values.
0x08000046	DDR2SIZE_SODIMM2	Controls address mapping order on second DIMM interface (FPGA C only)
0x08000047	HIADDRSIZE_SODIMM2	Number of unique addresses in HIADDR for second DIMM interface (FPGA C only)
0x0800004B	SODIMM2_RANK	IIC data retrieved from the SODIMM in socket 2
0x0800004C	SODIMM2_COL	(FPGA C only)

0x0800004D	SODIMM2_ROW	-
0x0800004E	SODIMM2_BANK	-
0x0800004F	SODIMM2_CAS	-
0x0800007C	ETH_EEPROM	Read to get the values of the EPROM SDA (bit 0) Write to modify {SCL, SCL_oe, SDA, SDA_oe} FPGA D and F only. MainTest only
0x0800007E	VRP_ALL	Contains input signals on the “VRP” pins
0x0800007F	VRN_ALL	Contains input values on the “VRN” pins
0x0B000000 - 0x0B0003FF	BLOCKRAM	The contents of an internal-FPGA block RAM
0x0C000XX0	BUS XX OUT	XX can be 0-21 hex. Output status of IOs on bus XX.
0x0C000XX4	BUS XX OE	XX can be 0-21 hex. OE status of IOs
0x0C000XX8	BUS XX IN	XX can be 0-21 hex. The input values
0x0C000XXC	BUS XX Name	A unique name of the bus (schematic)
0x0xxxxxxx	REG_DEFAULT	0xDEAD5566 Any undefined register

3 Interconnect (DC)

The “single-ended” interconnect test tests the DC connectivity of FPGA-to-FPGA interconnect, and the “MB” signals.

Presented on the MainBus, are registers allowing the interface to control the output value, output enable, and input value of each FPGA-to-FPGA interconnect pin. Each pin on the FPGAs is pulled high. This allows a test program to find single-stuck-at faults, open faults, and stuck-together faults.

3.1 Using the Design

The design can be controller over the MainBus. The register banks connected to the IO are arranged into “busses”. Each bus has an ID code, an OE register bank, an ENABLE register bank, and an IN register bank.

The addresses of the IO registers are as follows:

FpgaNum (4-bit) | MB_SEL_INTERCON (4 bit) | busnum (20-bit) | reg_offset (4-bit)

FPGA NUM is 0x0 for FPGA A, 0x1 for FPGA B, 0x2 for FPGA C...

MB_SEL_INTERCON is 0xC

busnum is any number, but only low-values (less than LAST_ADDR) will constrain valid busses

reg_offset is 0x0 for REG_OUT, 0x4 for REG_OE, 0x8 for REG_IN, and 0xC for REG_ENABLED

To determine which bits (if any) in a bus are valid, read the REG_ENABLED register. The 32-bits returned ‘1 are a mask for which of the bits in the REG_OUT, REG_OE, and REG_IN registers are meaningful.

To get the bus ID of a bus, write value 0x1 (32-bit) to REG_ENABLED, then read REG_ENABLED, then write 0x0 (32-bit) to REG_ENABLED. The value returned will be a coded name for the bus. Bits 0-15 are ASCII characters representing FPGA names. Bits 16-31 are an arbitrary unique integer distinguishing the bus. Connecting busses from two different FPGAs have the same bus ID.

To cause an FPGA to output signals on a bus, write 0xFFFFFFFF on REG_OE. To set the outputs all to “high” write 0xFFFFFFFF to REG_OUT.

To read the current received value from the bus’ inputs, read from REG_IN

3.2 Running the Test

In the USB Controller program, select Settings->OneShot Test. From the dialog box, check the Interconnect Test box. The program will automatically load the bit files, set the clocks and run the test.

4 DDR2 Interface

The DDR2 interface design is an example DDR2 controller running at 250Mhz. You can use this controller as an example, especially for the purpose of required IO logic, timing and clocking. The controller bandwidth is most of what is possible on the DN9000K10PCIE4GL.

4.1 Provided Files

The DDR2 reference design is part of the “MainTest” reference design, and the MainTest files should be used.

4.2 Using the Design

The DDR2 memory interfaces are mapped to the address range

0xNXX00000 – 0xNXXFFFFFF

Where the 4-bit “N” represents an FPGA ID, as described in the MainBus interface description. X are “don’t-care”. Since the remaining 19 bits are insufficient to address an entire 4GB DRAM, there is a register DDR2HIADDR that selects the highest address bits of the DRAM. Each address refers to a 32-bit location in the DRAM. The lowest bit is not mapped to DRAM address, but instead selects between the upper and lower 32 bits of the DRAM data. This is

necessary because MainBus is a 32-bit interface, and the DN9000K10PCIE4GL DRAM interfaces are 64 bits wide.

The bank and side controls are also mapped to the DDR2HIADDR register. The location of the DDR2HIADDR register is given in the Reference Design Memory Map section.

The clock that this design uses (G1) must be set to between 180 and 250Mhz. <verify this number>.

4.3 Running the Test

To run the hardware test, in the USB Controller application, select Settings->OneShotTest and check the DDR2 box. The program will automatically load the bit files, set the clocks and run the test, reporting any errors.

5 Clock Counters

Each clock available to the FPGA is connected to a counter register, and the value of this register is available on MainBus. In this way, the user can determine if each clock input is working properly.

6 LEDs

All of the LEDs are connected to an output enable register. When the LEDs are not enabled, the blink a pattern representing which FPGA the design is for. When enabled, each LED is controlled by the LED value register.

7 Simulating the Reference Design

The simulation environment the Dini Group uses is ModelSim. A ModelSim project file is provided, but it may not be compatible with your version of ModelSim. When you create a ModelSim project, add only the top-level design file (sim_single.v).

Source can be found on the user CD:

D:\FPGA_Reference_Designs\DN9000K10PCIE4GL\MainTest\source\

Also, you must add to the project a simulation library. Simulation models of all of the primitives used in the reference design are found in the Xilinx ISE install directory in the unisims directory.

Simulation models are also provided of the DN9000K10PCIE4GL as a whole board, along with DDR2 modules, headers and the MainBus interface.

8 Compiling the Reference Design

The MainTest reference design (for which bit files are included on the user CD and the provided CompactFlash card) can be found on the user CD here.

```
D:\FPGA_Reference_Designs\
    \common\DDR2\controller_ver\*
    \ddr2_to_mb\*
    \DN9000K10PCIE4GL
        \MainTest\source\*
```

The top module is

```
D:\FPGA_Reference_Designs\DN9000K10PCIE4GL\MainTest\source\fpga.v
```

This module includes all of the other required sources and expects the directory structure found on the CD.

8.1.1 The Xilinx Embedded Development Kit (EDK)

The DN9000K10PCIE4GL does not use the EDK because it has no embedded processor.

8.1.2 Xilinx XST

The Dini Group uses XST software to for design synthesis. The XST projects for each of the 3 FPGAs on the DN9000K10PCIE4GL can be found at 'buildxst/*.xst'. These projects have been create and tested using XST version 9.1.

8.1.3 Xilinx ISE

Xilinx ISE version 9.1.02 (service pack 3 or later) is required to use the DN9000K10PCIE4GL. Earlier versions may work, but are not supported. Use HDL files as input. Modification of the ISE project may also require modification of the DHL, timing constraints are in files 'buildsxt/*.xcf'.

8.1.4 The Build Utility: Make.bat

The Build Utility is found at 'DN9000K10PCIE4GL/build/make.bat'. This batch file can be used to run ISE and bitgen. You may need to run make.bat from inside of a Cygwin session because the script runs the program sed. You may also need to add the Xilinx bin directory to your path so the command "par" calls the correct program. Send the command which par to make sure that "par" is the place-and-route program in the Xilinx bin directory.

The build script creates a directory called "out" and places its output files there. After the script completes you will find files for each FPGA that was built. fpga_*.bit is the file to be downloaded to the FPGA.

The script contains a hard-coded value for the type of FPGA installed in each location on the board.

```
:implement_a
call :implement_x A xc5vlx110-1ff1760 lx220
goto end
```

This must be changed to lx330 if you have an LX330 board. To run the script, see the comments near the top of the file.

Example invocation:
./make.bat SINGLE A

8.2 Bitgen Options

The Make.bat script correctly sets all bitgen options that are compatible with the DN9000k10PCI. The following options should be used with the DN9000K10PCI. Options that are not listed here can be selected by the user, or left to their default settings.

Compress:	OFF	(Or you can disable “sanity check” option on board)
UnusedPin:	Pullnone	
Persist:	Yes	(Only require is Readback is used)
Encrypt:	No	(Or you can disable “sanity check” option on board)
DonePipe:	No	(“Yes” Can cause configuration errors)
DriveDone:	Yes	(“No” can cause configuration Errors)

8.3 VHDL

VHDL source is generated from the Verilog code. Therefore at any moment the Verilog code is more recent.

9 LVDS Reference Design

The "LVDS Intercon" design is to show the user how to implement source-synchronous communication between FPGAs. Using this method, the advertised 800Mbs system speed can be achieved. If you do not wish to use source-synchronous interconnect, ignore this reference design.

All FPGA-to-FPGA interconnect in this design is constantly being driven by one FPGA sending (uni-directionally) a test pattern. The receiving FPGA checks the test pattern for correctness against a known pattern.

The design is intended to characterize the bandwidth of the interconnect between FPGAs. Access to test status is provided over the MainBus interface.

Note that there are two designs, “ADC” and “CBA”. In the design, the directions of LVDS connections between FPGAs are uni-directional. In the “CBA”, all of the signals are in a direction opposite to the “ABC” design signals.

9.1 Provided Files

The source is located at:

D:\FPGA_Reference_Designs\DN9000K10PCIE4GL\MainRef

Note that this is the same source as the “Main Reference Design”. To compile the design for LVDS, #define statements in the Verilog code must be added or removed. The make.bat utility described in the “compiling the reference design” section automatically adds and removes these directives. The pre-compiled bitfiles for this design are located at

D:\FPGA_Reference_Designs\Programming_Files\DN9000K10PCIE4GL\LVDSIntercon\

9.2 Using the Design

The design’s MainBus interface is undocumented

The IOs in the LVDS reference design are clocked using the G0 clock. A clock setting of 300Mhz on G0 results in data transmission from FPGA to FPGA of 600Mbs per signal pair.

9.3 Running the Test

In the USB Controller program, select Settings->OneShot Test. From the dialog box, check the Interconnect Test box. The program will automatically load the bit files, set the clocks and run the test.

10 Ethernet and PCI Express Design

The bit files provided for the Ethernet and PCI Express test are for the purpose of testing hardware, and should not be used as a starting point for PCI Express or Ethernet designs.

Ethernet test setup

- 1) Set G0 clock to 100Mhz.
- 2) Configure FPGA D and E with the Ethernet design
- 3) Connect the RJ45 to a computer using a UTP cable
- 4) Set the IP address of this computer to 254.169.1.1
- 5) Make sure the “1G LINK” LED on the DN9000K10PCIE4GL is on
- 6) Make sure the LINK LED on the computer is on.
- 7) Hook up the RS232 port to the USER RS232 port of the DN9000K10PCIE4GL
- 8) From the computer, ping 254.169.1.2

PCI Express test setup

- 1) Plug the board into a motherboard with a 4x PCI slot
- 2) Configure FPGA A with the bitfile PCI_EXPRESS_MOTHERBOARD_LOOPBACK
- 3) Run AETest_usb

THE REFERENCE DESIGN

- 4) Select the menu option to run PCI Express loopback test
- 5) The test should report pass/fail for each of the 4 PCI Express lanes. Note that some PCI Express chipsets do not implement loopback for all four lanes, and instead will only loop back lane 0. If this is the case, then only channel B will report a pass.

Ordering Information

Part Number
DN9000K10PCIE4GL

1 Section Title

Request quotes by emailing sales@dinigroup.com.

For technical questions email support@dinigroup.com

2 FPGA Options

Any subset of FPGAs can be installed on the DN9000K10PCIE4GL. Any unneeded FPGA positions can ship empty to reduce the total price.

2.1 FPGA A-F:

Select an FPGA part to be supplied in each position, A-F. Possible selections are

NONE

LX110 -1 -2 -3

LX220 -1 -2 -3

LX330 -1 -2

2.2 CES Parts

The DN9000K10PCIE4GL may ship with CES “engineering sample” parts. This is often the case early in the Xilinx product release cycle. If your board will ship with CES parts, the quote will state the Xilinx part number of each FPGA on your board, indicating a CES revision. It is important that the user knows that CES parts may have limitations that are not listed in the Virtex-5 datasheet. To read about these limitations, see the Xilinx website and search for Virtex-5 errata. In general, it is the responsibility of the user to determine if the board is suitable for his application prior to ordering a board. Details about the interfaces on the board that are not in this manual and characterizations of interfaces, if available can be requested.

2.2.1 Hardware Errata Details

This list may not be complete. See the Xilinx website for an up-to-date list

Block RAM

- 1) In simple dual port and cascade mode, the data output register cannot be initialized by configuration (INIT), and it cannot be restored to the INIT register content using the Global Set/Reset (GSR).
- 2) Violating timing requirements on the address bus can cause the block RAM to cease normal operation until the device is reconfigured. Asynchronous resetting of the block RAM in FIFO mode can cause a similar outcome.
- 3) FMAX_FIFO is 400 MHz when using the attribute EN_SYN = TRUE.

Clocks

- 1) Only BUFG and BUFGCE are supported. The remaining global clock buffer primitives are not supported.

OSERDES

- 1) Optional inversion for the divided clock (CLKDIV) in the OSERDES is not supported.

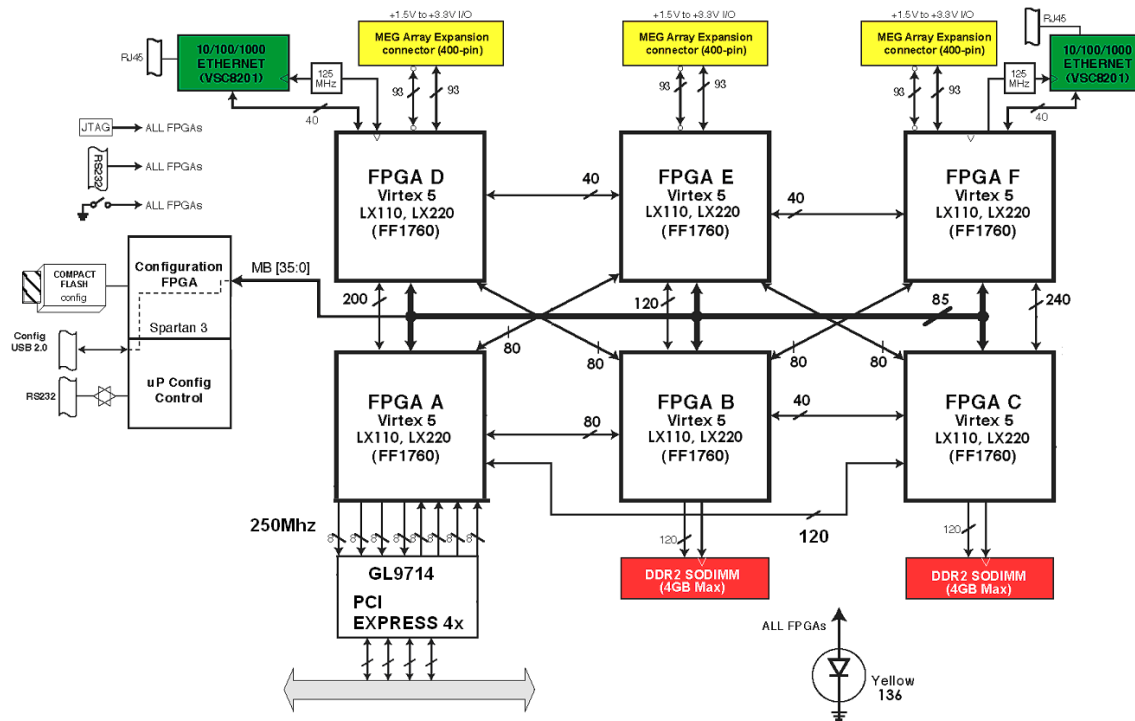
Configuration

- 1) Configuration bitstream compression and fallback reconfiguration are not supported.

2.3 “Small” FPGAs

The DN9000K10PCIE4GL is optimized for six Xilinx Virtex-5 LX330 FPGAs. Optionally, it can be ordered with LX110 or LX220 FPGAs instead. When installed with one or more LX110 or LX220 FPGAs, the amount of available interconnect is reduced due to some IOs in those devices being "no-balled".

ORDERING INFORMATION



The diagram above shows the block diagram representing the resources available on a board installed with six LX110 or LX220 FPGAs. The size of the MB bus is reduced from 169 to 85 signals. The interconnect between A-B, D-E, E-F, and B-C is greatly reduced. The DIMMs connected to FPGA A, D, and F are not present. One of the DIMMs on FPGA C is not present.

The Ethernet, daughtercards, and PCI are not affected by FPGA selection. Also, you should analyze your design to determine if the internal resources available in the LX110 and LX220 are sufficient to meet your needs. The FPGA selection guide from Xilinx is printed below.

ORDERING INFORMATION

			LX110	LX220	LX330
Part Number			XC5VLX110	XC5VLX220	XC5VLX330
CLB Array Size (Row x Column)			160 x 54	160 x 108	240 x 108
Slices ⁽²⁾			17,280	34,560	51,840
Logic Cells ⁽³⁾			110,592	221,184	331,776
CLB Flip-Flops			69,120	138,240	207,360
Maximum Distributed RAM (kbits)			1,120	2,280	3,420
Block RAM/FIFO w/ECC (36kbits each)			129	192	288
Total Block RAM (kbits)			4,608	6,912	10,368
DSP48E Slices			64	128	192
FF1760	42.5 x 42.5 mm	I/O	800	800	1200

2.4 Speed Grades

The interface performance characterizations included in this manual and in advertisements are valid for all shipped FPGAs, regardless of speed grade. These numbers are characterizations, and not guaranteed under all operational conditions. Every shipped board has passed this characterization test under some operational conditions.

If there are any interfaces where performance is only characterized for specific speed grade parts, this is noted in the advertisement and in this document. Below is a list of all such interfaces:

1) PCI Express

Some interfaces may run at increased speeds above and beyond Dini Group's advertised performances when used with –2 or –3 speed grade parts. Some performance numbers that are advertised by Xilinx are listed here. These characterizations have not been performed on the DN9000K10PCIE4GL, but we have no reason to think the DN9000K10PCIE4GL is a limiting factor on these interfaces:

FPGA-to-FPGA interconnect, LVDS	1.25 Gbs (625 MHz)
FPGA-to-FPGA interconnect, single-ended	800 Mbps (400 MHz)
DDR2 Interface	667 Mbps (333 MHz)

2.5 Upgrade Policy

Upgrading (adding FPGAs) to a DN9000K10PCIE4GL.

Call for a quote.

3 Optional Equipment

The following tools are suggested for use with the Dini Group DN9000K10PCIE4GL.

3.1 Compatible Dini Group products

The Dini Group supplies standard daughtercards and memory modules that you can use with the DN9000K10PCIE4GL.

3.1.1 Memories

The memory module solutions from Dini Group allow the user to install whichever type of memory his application requires.

- **DNSODM200_SRAM**
Memory module for use in the 200-pin SODIMM sockets.
Standard memory configuration: Two GS8320V32 memories
(1M x 32 each)
Performance up to 175Mhz (SDR)
Small EPROM.
- **DNSODM200_QDR**
QDR SRAM module, compatible with the DDR2 SODIMM sockets.
Two times 32 bit wide
Small EPROM.
- **DNSODM200_NOBL**
Cypress “no bus latency” SRAM (64 bit wide)
Compatible with 200-pin SODIMM sockets
Ask for availability.
Small EPROM.
- **DNSODM200_RLDRAM**
Reduced latency DRAM (Micron) 64 bit wide
compatible with the 200-pin SODIMM sockets.
Small EPROM.
- **DNSODM200_MICTOR**
Provides 2 Mictor-38 connectors.
Compatible with the DDR2 SODIMM sockets.
User LEDs. Small EPROM.

ORDERING INFORMATION

- **DNSODM200_DDR1**
DDR1 memory module compatible with the 200-pin SODIMM sockets
(Jumper rework required)
Comes with 512MB standard.
Allows use of standard PC2700 modules (up to 1GB)
175Mhz performance
- **DNSODM200_SDR**
SDR memory module compatible with 200-pin SODIMM sockets.
Accepts PC133 modules up to 512MB.
(Jumper rework required)
Comes with 256MB standard.
75Mhz performance
- **DNSODM200_FLASH**
Spansion S29WS064J memory (x2). Each is 4Mx16 bit flash
16Mb SRAM memory (512k x 32)
Compatible with DDR2 SODIMM sockets.
66Mhz performance (read burst)

3.1.2 Extenders

- The DNPCIEXT-S3/5 is an extender card designed to aid in the debug and test of PCI-based circuit boards. This is an active extender card; an Intel® 21154 PCI to PCI bridge is used to isolate the primary PCI bus from the three secondary PCI bus slots. Since primary and secondary busses are electrically isolated, a much cleaner electrical signaling environment exists, and a single host slot can be expanded to contain up to three plug-in PCI cards. The primary PCI frequency can range from 0 to 66.66 MHz. The secondary PCI frequency is configurable to be the primary frequency or one half the primary frequency. DIP switches are provided to force the primary or secondary busses to 33MHz.

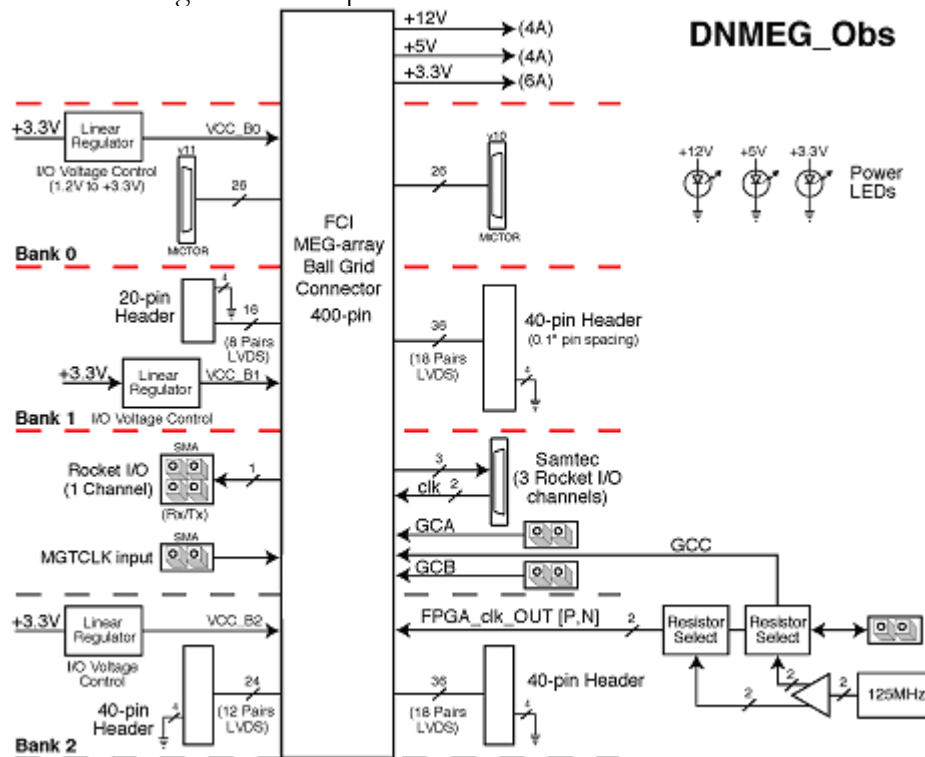
3.1.3 Daughtercards

Dini Group daughtercards connect to the MEG-Array connector (400-pin) using the standard Dini Group interface description.

- **DNMEG_PCIE**
8-lane PCI express PHY card. Host or downstream mode.
DDR2 memory module.
Virtex-4 FPGA. (LX40-LX160)
- **DNMEG_ADC**
High-speed Analog-Digital daughtercard
Virtex-4 FPGA
DDR2 memory module
250Msps, 12-bit ADC. 60dB SNR (10 bits) 200kHz-75Mhz

ORDERING INFORMATION

- **DNMEG_ARM**
ARM processor core development daughtercard
Altera Excalibur FPGA
PC133 memory module
- **DNMEG_S2GX**
Altera Stratix 2 GX FPGA. DDR2 Memory.
High-speed serial interfaces. SMA, SATA, SFP, others
- **DNMEG_OBS**
Adjustable-voltage tenth-inch pitch headers
User LEDs
Two Mictor-38 connectors.
SMA global clock inputs for host board.



3.2 Compatible third-party products

The following products have been shown to work with the DN9000K10PCIE4GL.

Intel Entry Server board SE7230NH1-E

<http://www.intel.com/design/servers/boards/se7230nh1-e/index.htm>

Standard DDR2 modules (256 MB \$44, 512 MB \$79, 1GB \$141, 2GB \$722, 4GB eventually)

<http://www.crucial.com/store/listmodule/DDR2/list.html>

Xilinx Platform USB Cable (required for JTAG FPGA programming, firmware update, ChipScope Pro)
HW-USB-G
<http://nuhorizons.com>

4 Compliance Data

4.1 Compliance

4.1.1 EMI

Since the DN9000K10PCIE4GL is not intended for production systems, it has not passed EMI testing. Compliance is only done by special request.

4.1.2 PCI-SIG

4.2 Environmental

4.2.1 Temperature

The DN9000K10PCIE4GL is designed to operate within an ambient temperature range of 0 – 50 degrees C.

All components used on the DN9000K10PCIE4GL are guaranteed to operate within a temperature range of 0 – 80 degrees C (measured on the device die).

4.3 Export Control

4.3.1 Lead-Free

The DN9000K10PCIE4GL meets the requirements of EU Directive 2002/95/EC, “RoHS”. Specifically, the DN9000K10PCIE4GL contains no homogeneous materials that:

- a) contains lead (Pb) in excess of 0.1 weight-% (1000 ppm)
- b) contains mercury (Hg) in excess of 0.1 weight-% (1000 ppm)
- c) contains hexavalent chromium (Cr VI) in excess of 0.1 weight-% (1000 ppm)
- d) contains polybrominated biphenyls (PBB) or polybrominated dimethyl ethers (PBDE) in excess of 0.1 weight-% (1000 ppm)
- e) contains cadmium (Cd) in excess of 0.01 weight-% (100 ppm)

No exemptions are claimed for this product.

4.3.2 The USA Schedule B number based on the HTS

8471 60 7080

4.3.3 Export control classification number ECCN

EAR99

4.4 Mission Critical

DN9000K10PCIE4GL and supporting hardware and software are not intended for use on human subjects, in life support or mission-critical systems.