# Table of Contents

## ABOUT THIS MANUAL

1. **MANUAL CONTENTS**
   - About this Manual: 10
   - Quick Start Guide: 10
   - Controller (USB) Software Guide: 10
   - Board Hardware Description: 10
   - Reference Design Guide: 10
   - FPGA Design Guide: 10
   - Ordering Information: 10

2. **ADDITIONAL RESOURCES**

3. **CONVENTIONS**
   3.1 **Typographical**
   3.2 **Content**
     3.2.1 File names: 13
     3.2.2 Physical orientation and Origin: 14
     3.2.3 Part Pin Names: 14
     3.2.4 Schematic Clippings: 14
     3.2.5 Media card interface: 14
     3.2.6 Config FPGA: 14
     3.2.7 Terminology: 14
     3.2.8 FPGA Numbering: 15

## QUICK START GUIDE

1. **PROVIDED MATERIALS**

2. **ESD WARNING**

3. **POWER-ON INSTRUCTIONS**
   3.1 Check Power Jumper
ABOUT THIS MANUAL

3.2 Memory and heat sinks ................................................................. 19
3.3 Prepare configuration files .......................................................... 20
3.4 Connect cables ............................................................................. 22
3.5 Power on ....................................................................................... 22
3.6 Check Power indicator LEDs ...................................................... 23
3.7 View configuration feedback over RS232 .................................... 23
  3.7.1 Watch startup sequence over RS232 ....................................... 23
  3.7.2 Interactive configuration ......................................................... 25
  3.7.3 Read temperature sensors ....................................................... 26
  3.7.4 User Serial port .................................................................... 27
3.8 Check LED status lights .............................................................. 28

4 USING THE USB CONTROLLER PROGRAM ........................................ 29
4.1 Operating the USB controller program ....................................... 30
4.2 Using AETEST to run hardware tests ......................................... 32
  4.2.1 AETest on Linux or Solaris .................................................... 32
  4.2.2 Use AETest .......................................................................... 32

5 BOARD CONTROLS ........................................................................ 35

6 MOVING ON .................................................................................. 36

CONTROLLER SOFTWARE .......................................................... 37

1 USB CONTROLLER ....................................................................... 37
1.1 Visual display .............................................................................. 38
1.2 Log window ................................................................................. 38
1.3 Menu Options ............................................................................. 38
  1.3.1 Contextual Menu ................................................................. 38
  1.3.2 File Menu ........................................................................... 38
  1.3.3 Edit Menu .......................................................................... 39
  1.3.4 FPGA Configuration Menu ................................................ 39
  1.3.5 FPGA Memory Menu .......................................................... 40
  1.3.6 Settings Info ....................................................................... 41
  1.3.7 Settings/Info Menu .............................................................. 43

2 AETEST ......................................................................................... 44
2.1 Running AETEST ....................................................................... 45
2.2 Compiling AETEST ................................................................. 45
ABOUT THIS MANUAL

3 UPDATING THE DN8000K10 FIRMWARE ......................................................................................................................... 45
3.1 Updating the MCU (flash) firmware ................................................................................................................................. 46
3.2 Updating the Configuration FPGA (PROM) firmware ....................................................................................................... 47

4 PROGRAMMER’S GUIDE .................................................................................................................................................. 50
4.1 Cypress CY7C68013A ...................................................................................................................................................... 51
4.2 Configuration FPGA .......................................................................................................................................................... 51
4.2.1 Configuration Register Map ........................................................................................................................................ 52

4.3 Vendor Request List ...................................................................................................................................................... 54

4.4 USB Reference Design Control ..................................................................................................................................... 55
4.4.1 Main Bus accesses ....................................................................................................................................................... 55
4.4.2 Configuration ............................................................................................................................................................... 56
4.4.3 Readback .................................................................................................................................................................... 56

HARDWARE ..................................................................................................................................................................... 57

1 OVERVIEW ..................................................................................................................................................................... 57

2 CONFIGURATION CIRCUIT ........................................................................................................................................... 58

2.1 Overview ...................................................................................................................................................................... 58

2.2 Configuration Options .................................................................................................................................................. 59
2.2.1 CompactFlash.............................................................................................................................................................. 59
2.2.2 Jtag ................................................................................................................................................................................. 63
2.2.3 SelectMap ................................................................................................................................................................. 65
2.2.4 IDE (Remote Compact Flash) ................................................................................................................................... 65
2.2.5 USB ............................................................................................................................................................................ 66

2.3 The Configuration FPGA ............................................................................................................................................... 66
2.3.1 Config FPGA Configuration .................................................................................................................................... 66
2.3.2 Smart Media / Compact Flash ................................................................................................................................. 68
2.3.3 MCU communication .............................................................................................................................................. 69
2.3.4 Clock control ............................................................................................................................................................ 69
2.3.5 RS232 ........................................................................................................................................................................ 71
2.3.6 Main Bus control ..................................................................................................................................................... 72
2.3.7 LEDs .......................................................................................................................................................................... 72
2.3.8 Remote access ....................................................................................................................................................... 72
2.3.9 IIC .............................................................................................................................................................................. 72
2.3.10 Signal Descriptions ........................................................................................................................................... 72

2.4 FPGA configuration Process ........................................................................................................................................ 72

2.5 MCU ............................................................................................................................................................................. 75
2.5.1 RS232 ....................................................................................................................................................................... 75
2.5.2 Clocks ....................................................................................................................................................................... 76
2.5.3 LEDs ....................................................................................................................................................................... 77
2.5.4 Memory space ....................................................................................................................................................... 77
2.5.5 Config FPGA Memory Space ................................................................................................................................... 79
2.5.6 Flash and SRAM memory space ............................................................................................................................ 81
2.5.7 USB ....................................................................................................................................................................... 81

DN8000K10 User Guide www.dinigroup.com 4
### ABOUT THIS MANUAL

2.5.8 CompactFlash........................................................................................................................................................................84

3 CLOCKING ..................................................................................................................................................................................85

3.1 Global Clocks..............................................................................................................................................................................87
  3.1.1 ICS 8442 Phases.................................................................................................................................................................88
  3.1.2 SS (Single Step) Clocks..................................................................................................................................................89
  3.1.3 DIV Clocks.....................................................................................................................................................................90
  3.1.4 User Clock...................................................................................................................................................................91

3.2 Reference Clock......................................................................................................................................................................92

3.3 Daughter card clocks ............................................................................................................................................................92

3.4 FPGA clock banks.....................................................................................................................................................................94

3.5 Expansion CLPD.....................................................................................................................................................................94
  3.5.1 FX CLPD.....................................................................................................................................................................96

4 RESET TOPOLOGY ..............................................................................................................................................................96

4.1 Reset Configuration .............................................................................................................................................................96

5 POWER .................................................................................................................................................................................98

5.1 Switching power supplies ....................................................................................................................................................101

5.2 Secondary Power Supplies ................................................................................................................................................102
  5.2.1 DDR2 Termination Power................................................................................................................................................102
  5.2.2 MGT Power................................................................................................................................................................103
  5.2.3 Optical Module Power................................................................................................................................................104
  5.2.4 VBATT.........................................................................................................................................................................105

5.3 Power distribution...............................................................................................................................................................107
  5.3.1 Bypassing....................................................................................................................................................................108
  5.3.2 VCCAUX..................................................................................................................................................................108

5.4 Cooling................................................................................................................................................................................108

6 FPGA INTERCONNECT ........................................................................................................................................................110

6.1 High speed Serdes support ..............................................................................................................................................110

6.2 Main Bus ..........................................................................................................................................................................113

7 FPGA DRAM MEMORY INTERFACE ....................................................................................................................................115

7.1 Clocking..............................................................................................................................................................................115

7.2 Signaling...........................................................................................................................................................................116
  7.2.1 Termination...............................................................................................................................................................116
  7.2.2 Source-synchronous clocking....................................................................................................................................116

7.3 SODIMM Power supply .....................................................................................................................................................116

7.4 Alternate Memory modules..............................................................................................................................................117
ABOUT THIS MANUAL

13 TEST POINTS AND CONNECTORS .................................................................................................................. 162
13.1 Test points .......................................................................................................................................................... 162
13.2 Connectors ....................................................................................................................................................... 164

THE REFERENCE DESIGN ............................................................................................................................... 168

1 EXPLORING THE REFERENCE DESIGN ........................................................................................................... 168
1.1 What is the Reference Design? ......................................................................................................................... 168
1.2 Running the Reference Design.......................................................................................................................... 170
1.2.1 The Precompiled Bit files ............................................................................................................................... 170
1.2.2 Load FPGAs with the reference design ........................................................................................................... 171
1.2.3 Run the AETest program ............................................................................................................................... 171
1.2.4 Run the RocketIO Test ................................................................................................................................ 171
1.3 Compiling the Reference Design (Windows)..................................................................................................... 174
1.3.1 Modify Source ................................................................................................................................................. 174
1.3.2 Xilinx ISE ......................................................................................................................................................... 175
1.3.3 The Build Utility: Make.bat ......................................................................................................................... 176

2 IMPLEMENTATION DETAILS ............................................................................................................................ 177
2.1 Main Test ............................................................................................................................................................ 177
2.2 RocketIO V4 Test ............................................................................................................................................. 179
2.3 LVDS Test .......................................................................................................................................................... 180
2.3.1 Clock Summary ............................................................................................................................................. 180

3 ADDRESS MAPS ............................................................................................................................................. 181
3.1 Memory Space ................................................................................................................................................... 181
3.1.1 Decoder ......................................................................................................................................................... 183
3.2 Main Bus Signaling .......................................................................................................................................... 185

ORDERING INFORMATION ............................................................................................................................... 188

1 FPGA OPTIONS .................................................................................................................................................. 188
1.1 FPGA F0, F12 ..................................................................................................................................................... 188
1.2 FPGA F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F13, F14, F15: ................................................................. 188
1.3 Configuration FPGA: ...................................................................................................................................... 189

2 MULTI-GIGABIT SERIAL OPTIONS .................................................................................................................. 189
2.1 Serial Clock Crystals ............................................................................................................................................................... 189

3 OPTIONAL EQUIPMENT ............................................................................................................................................................... 189

3.1 Rack mount Chassis ................................................................................................................................................................ 189

3.2 Daughter cards ........................................................................................................................................................................ 190

4 THIRD PARTY EQUIPMENT ............................................................................................................................................................. 191

5 COMMON PROBLEMS ................................................................................................................................................................. 192

5.1.1 Simulation Does not match Synthesis .................................................................................................................................... 192

5.1.2 XST is giving errors because it cannot find virtex4.v or resync.v ............................................................................................ 193

5.1.3 Signal name[0] exists in my code, but place and route MAP step complains the signal name[0] from my ucf file does not exist 193

5.1.4 Certify fails because the ports on my Virtex 4 do not match the specified part .............................................................................. 193

5.1.5 The DN8000K10 cannot read my SmartMedia card. The RS232 port prints: Bad SM card format .............................................. 193

5.1.6 There is a lot of deterministic jitter (DJ) on my RocketIO channel .......................................................................................... 193

5.1.7 There is a lot of deterministic jitter (DJ) on my RocketIO clock signals ................................................................................... 194

5.1.8 My design works correctly in all the FPGAs except for one ....................................................................................................... 194

5.1.9 RocketIO is getting a high bit error rate .................................................................................................................................... 194

5.1.10 RocketIO is not working ........................................................................................................................................................... 194

5.2 Common Questions ....................................................................................................................................................................... 194

5.2.1 Where are all of the debug pins .................................................................................................................................................. 194

5.2.2 Where can I get the daughter card connectors? ........................................................................................................................ 194

5.2.3 Can I use two Dini Products at the same time? ........................................................................................................................ 194

5.2.4 Do you provide a board description file for ____? ..................................................................................................................... 194

5.2.5 Where is the VHDL reference design? ...................................................................................................................................... 194
About this Manual

Welcome to DN8000K10 Logic Emulation Board

Congratulations on your purchase of the DN8000K10 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 DN8000K10 provides.
1 Manual Contents

About this Manual
List of available documentation and resources available. Reader’s Guide to this manual

Quick Start Guide
Step-by-step instructions for powering on the DN800K10, loading and communicating with a simple provided FPGA design and using the board controls.

Controller (USB) Software Guide
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.

Board Hardware Description
Detailed description and operating instructions of each individual circuit on the DN800K10

Reference Design Guide
Detailed description of the provided DN800K10 reference design. Implementation details of the reference design interaction with DN800K10 hardware features.

FPGA Design Guide
Information needed to use the DN800K10 with third-party software, including Xilinx ISE, Certify, and Identify. Some commonly asked questions and problems specific to the DN800K10

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

2 Additional Resources

For additional information, go to http://www.dinigroup.com. All of the electronic information provided on the User CD is updated frequently; your User CD contains the latest files available at the time your board was shipped.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN800K10 User Guide</td>
<td>This is your first source of technical information.</td>
</tr>
<tr>
<td>Resource</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DN8000K10 User Guide appendicies</strong></td>
<td>The appendicies are distributed with the User Guide on the user CD and are available from the Dini Group website: <a href="http://www.dinigroup.com">www.dinigroup.com</a></td>
</tr>
<tr>
<td></td>
<td>PIN_OTHER – Pin-to-pin connection information for Daughter cards, clocks, memory modules, Multi-gigabit transceivers (MGT), LED and Main Bus</td>
</tr>
<tr>
<td></td>
<td>PIN_DIAG – Pin-to-pin connections for inter-FPGA interconnect “Diagonal” connections.</td>
</tr>
<tr>
<td></td>
<td>PIN_HORIZ - Pin-to-pin connections for inter-FPGA interconnect “Horizontal” connections.</td>
</tr>
<tr>
<td></td>
<td>PIN_VERT - Pin-to-pin connections for inter-FPGA interconnect “Vertical” connections.</td>
</tr>
<tr>
<td></td>
<td>Schematics.pdf – Abbreviated schematics of the DN8000K10. The PDF file can be searched for net names and part numbers using the PDF search.</td>
</tr>
<tr>
<td></td>
<td>ASSY_TOP.pdf – A drawing of the DN8000K10 top side showing part placement</td>
</tr>
<tr>
<td></td>
<td>ASSY_BOT.pdf – A drawing of the DN8000K10 bottom side, reversed, showing part placement</td>
</tr>
<tr>
<td></td>
<td>DRILL.pdf – A drawing showing the DN8000K10’s phystical dimensions and mounting hole locations</td>
</tr>
<tr>
<td><strong>DN8000K10 document library</strong></td>
<td>Datasheets for all parts used on the DN8000K10, Application notes providing implementation suggestions, design documents, specifications of implemented interfaces</td>
</tr>
<tr>
<td></td>
<td>Xilinx Virtex 4 User Guide (UG70)</td>
</tr>
<tr>
<td></td>
<td>Xilinx RocketIO User Guide</td>
</tr>
<tr>
<td></td>
<td>Xilinx Virtex 4 Errata</td>
</tr>
<tr>
<td><strong>DN8000K10 reference design</strong></td>
<td>Example code for the DN8000K10 showing how to implement memory controllers, RocketIO, etc. The design is provided both as source, and as compiled bit file configuration streams for use with the FPGAs installed on your board.</td>
</tr>
<tr>
<td></td>
<td>Constraining (.ucf) files are included specifying pin outs, IO standards, and location constraints. You should modify and hold for your project.</td>
</tr>
<tr>
<td>Resource</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>use these constraint files</td>
<td>for your own design.</td>
</tr>
<tr>
<td>Dini Group website</td>
<td>The web page will contain the latest manual, application notes, FAQ, articles, and any device errata and manual addenda. Please visit and bookmark: <a href="http://www.dinigroup.com">http://www.dinigroup.com</a></td>
</tr>
<tr>
<td>E-Mail technical support</td>
<td>You may direct questions and feedback to the Dini Group using the following e-mail address: <a href="mailto:support@dinigroup.com">support@dinigroup.com</a></td>
</tr>
<tr>
<td>Phone technical support</td>
<td>Call us at <strong>858.454.3419</strong> during the hours of 8:00am to 5:00pm Pacific Time.</td>
</tr>
</tbody>
</table>
3 Conventions

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

3.1 Typographical

The following typographical conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning or Use</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Courier font</td>
<td>Messages, prompts, and program files that the</td>
<td>speed grade: - 100</td>
</tr>
<tr>
<td></td>
<td>system displays</td>
<td></td>
</tr>
<tr>
<td>Courier bold</td>
<td>Literal commands that you enter in a syntactical</td>
<td>ngdbuild design_name</td>
</tr>
<tr>
<td></td>
<td>statement</td>
<td></td>
</tr>
<tr>
<td>Garamond bold</td>
<td>Commands that you select from a menu</td>
<td>File Open</td>
</tr>
<tr>
<td></td>
<td>Keyboard shortcuts</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Italic font</td>
<td>Variables in a syntax statement for which you</td>
<td>ngdbuild design_name</td>
</tr>
<tr>
<td></td>
<td>must supply values</td>
<td></td>
</tr>
<tr>
<td></td>
<td>References to other manuals</td>
<td>See the Development System Reference Guide</td>
</tr>
<tr>
<td></td>
<td>for more information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Emphasis in text</td>
<td>If a wire is drawn so that it overlaps the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pin of a symbol, the two nets are not</td>
</tr>
<tr>
<td></td>
<td></td>
<td>connected.</td>
</tr>
<tr>
<td>Braces [ ]</td>
<td>An optional entry or parameter. However, in</td>
<td>ngdbuild [option_name] design_name</td>
</tr>
<tr>
<td></td>
<td>bus specifications, such as bus[7:0], they are</td>
<td></td>
</tr>
<tr>
<td></td>
<td>required.</td>
<td></td>
</tr>
<tr>
<td>Braces { }</td>
<td>A list of items from which you must choose one</td>
<td>lowpwr = {on</td>
</tr>
<tr>
<td></td>
<td>or more.</td>
<td></td>
</tr>
<tr>
<td>Vertical bar</td>
<td>Separates items in a list of choices</td>
<td>lowpwr = {on</td>
</tr>
<tr>
<td>Vertical ellipsis</td>
<td>Repetitive material that has been omitted</td>
<td>IOB #1: Name = QOUT'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOB #2: Name = CLKIN'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>Horizontal ellipsis</td>
<td>Repetitive material that has been omitted</td>
<td>allow block block_name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>loc1 loc2... locn;</td>
</tr>
<tr>
<td>Prefix “0x” or suffix “h”</td>
<td>Indicates hexadecimal notation</td>
<td>Read from address 0x00110373, returned 4552494h</td>
</tr>
<tr>
<td>Letter “#” or “_n”</td>
<td>Signal is active low</td>
<td>INT# is active low</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fpga_inta_n is active low</td>
</tr>
</tbody>
</table>

3.2 Content

3.2.1 File names

Paths to documents included on the User CD are prefixed with “D:\”. This refers to your CD drive’s root directory.
3.2.2 Physical orientation and Origin
By convention, the board is oriented as show on page 3, with the “top” of the board being the edge near FPGAs F0, F1, F2 and F3. The “bottom” is near FPGAs F12, F13, F14 anf F15. The “right” edge is near the USB and CompactFlash sockets, the “left” side is the side with the SMA connectors. “topside” refers to the side of the PWB with FPGAs soldered to it, “backside” is the side with the daughtercard connectors. The reference origin of the board is the lower left-hand corner of the board.

3.2.3 Part Pin Names
Pin names are given in the form XY.Z. 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 diodes, BS for fuses, HS for mechanicals, PSU for power supply modules, Q for discreet semiconductors, RN for resistor networks, X for oscillators, Y for crystals. Y is a number uniquely identifying each part from other parts of the same X class on the same PWB. 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 DN8000K10 are included in the Document library on the provided User CD.

3.2.4 Schematic Clippings
Partial schematic drawings are included in this document to aid quick understanding of the features of the DN8000K10. 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 document library as Appendix Schematics. Please refer to this document. Use the PDF search feature to search for nets and parts.

3.2.5 Media card interface
There are three Media card interfaces that can be used to configure FPGAs on the DN8000K10: CompactFlash, SmartMedia and IDE. IDE is intended to be used with a CompactFlash-to-IDE adapter module, like the one mounted on the face panel of the optional DN8000K10 chassis. The instructions for using all three interfaces are identical. See Hardware: Configuration: CompactFlash section. In this manual the all of the media card interfaces are referred to as “CompactFlash”.

3.2.6 Config FPGA
Some Dini Group documentation refers to the Configuration FPGA as Spartan. The configuration FPGA on the DN8000K10 is an LX40 or LX80 Virtex 4 FPGA.

3.2.7 Terminology
Abbreviations and pronouns are used for some commonly used phrases.

Host is the DN8000K10, as opposed to a daughter card connected to it.

MCU is the Cypress FX2 Microcontroller, U200

MGT and RocketIO are used interchangeably. MGT is multi-gigabit transceiver. RocketIO is the Xilinx trademark on their multi gigabit transceiver hardware.
**FPGA array** include all of the 16 user-programmable Virtex 4 FPGAs on the DN8000K10. These are F0-F15.

### 3.2.8 FPGA Numbering

The Virtex 4 FPGAs are named from the top left in a row major order, F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15. The configuration FPGA may be referred to in some cases as F16. For historical reasons, in some documentation, namely the reference design source code, the FPGAs may be lettered rather than numbered, where F0 is A, F1 is B… F15 is P.
Chapter 2

Quick Start Guide

The Dini Group DN8000K10 is the biggest general-purpose FPGA system available using the Xilinx Virtex 4 FPGA. The built-in configuration circuitry makes configuration management very easy. However, due to the number of features and flexibility of the board, it will take some time to become familiar with all the control and monitor interfaces equipped on the DN8000K10. Please follow this quick start guide to become familiar with the board before starting your logic emulation project.

1 Provided materials

Examine the contents of your DN8000K10 kit. It should contain:

- DN8000K10 board mounted on metal base plate
- EPS Power supply
- One 128MB CompactFlash card with reference design
- USB CompactFlash/SmartMedia card reader
- RS 232 IDC header cable to female DB9
- 6 Foot computer serial cable
- USB cable
- CD ROM containing
  - Virtex 4 Reference Design source code and .bit files
  - User manual, FPGA pin list in excel format
  - Datasheets for all parts on the DN8000K10
  - Board Schematic PDF
  - USB controller program (usbcontroller.exe)
2 ESD Warning

The DN8000K10 is sensitive to static electricity, so treat the PCB accordingly. The target market 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

The DN8000K10 is shipped in a metal carrier case designed to protect the board from physical and electrical damage. When you handle the DN8000K10, contact the handles of the carrier to ground yourself before touching the PWB.

The 300 and 400-pin connectors are not 5V tolerant. According to the Virtex 4 datasheets, the maximum applied voltage to these signals is VCCO + 0.5V (3.0V while powered on). On the DN8000K10, FPGA IO signals directly drive connectors and other exposed nets on the PWB. Be especially careful when working with cables and connectors.
3 Power-On Instructions

In the following sections, you will need to know the location of the following DN8000K10 features.

Figure 1

3.1 Check Power Jumper

The DN8000K10 can be installed in an optional chassis with a remote power-on switch. When not installed in its chassis, the remote control connector needs to have a jumper installed on its power-on signal. This jumper is installed before the DN8000K10 is shipped. Check the jumper installed in P203 pin 3 to pin 4. This jumper connects the EPS power supply PSON signal to GND.
3.2 Memory and heat sinks
There should be an active heatsink installed on each FPGA on the DN8000K10. Virtex 4 FPGAs are capable of dissipating 15W or more, so you should always run them with heat sinks installed. There is a fan 12V power connector next to each FPGA for an active heatsink.

The DN8000K10 comes packaged without memory installed. If you want to use the Dini Group reference design to test your memory modules, you can install them now in the 1.8V DDR2 DIMM sockets.

![Diagram of FPGA and memory connections]

The socket DIMM0 is connected to FPGA F1. DIMM1 is connected to FPGA F2. DIMM2 is connected to FPGA F13. DIMM3 is connected to FPGA F14.

The socket can accept any capacity DDR2 SODIMM module. Note that DDR1 modules will not work in these slots since DDR1 requires a completely different pin-out. For other memory options for the DN8000K10, see Ordering Information: Optional Equipment: Memory.

FPGAs are numbered in a row-major order, from F0 to F15. By convention, Dini Group references board dimensions as shown above. The top of the board is near FPGAs F0-F3 and the left of the board is near FPGAs F0, F4, F8, F12 and the RocketIO connectors.

Since F0 and F12 are both Virtex 4 FX family parts, they are also referred to as “FX0” and “FX1” respectively.
3.3 Prepare configuration files
The DN8000K10 reads FPGA configuration data from a CompactFlash card. To program the
FPGAs on the DN8000K10, FPGA design files (with a .bit file extension) put on the root
directory of the CompactFlash card file using the provided USB media card reader. The
DN8000K10 ships with a 128MB CompactFlash card preloaded with the Dini Group reference
design. This card is labeled “DN8000K10 Ref Design”. You can skip the “prepare config files”
step if you wish. Just use the preloaded Ref Design card. Also note that there is a SmartMedia
card interface on the DN8000K10 that behaves identically to the CompactFlash interface.

1. Insert a blank 128MB CompactFlash card (provided) into your USB card reader. If you
have fewer than 16 FPGAs on your DN8000K10, you may be able to use a lower
capacity card. The only supported file system is FAT16. (This is the standard file format
for CompactFlash)

2. Copy a configuration stream file (with a .bit file extension generated by the Xilinx tool
bitgen) for each of the 16 Virtex 4 FPGAs on to the Compact Flash card. The compiled
.bit files of the Dini Group DN8000K10 reference design are provided on the User CD
in the directory
D:\FPGA Programming Files\Standard_Reference_Design
There are provided programming files for all types of supported FPGAs, you must
select the ones that are compiled for the FPGAs on your board.

For more information about generating configuration streams, see FPGA Design Guide
and Hardware: Configuration Section

3. Create a file on the root of the Compact Flash card called main.txt. The main.txt file
contains instructions for the configuration circuitry of the DN8000K10. It also contains
settings required by the reference design to work properly like clock frequencies and
main bus enable. See Hardware: Configuration Section: Compact Flash for a detailed
description of the available main.txt file commands.
4. Copy the following text into main.txt and save. Eject the Compact Flash card.

Default Main.txt file contents:

```plaintext
//main.txt use with DN8000K10 reference design
verbose level: 2
sanity check: y

8442 PH0 CLOCK FREQUENCY: 150MHz
8442 PH1 CLOCK FREQUENCY: 200MHz
8442 PH2 CLOCK FREQUENCY: 66MHz
8442 REF CLOCK FREQUENCY: 200MHz

PH2 DIVIDE BY: 2^1
GCLK0 SELECT: 8442 // 350MHz
GCLK1 SELECT: 8442 // 200MHz
GCLK2 SELECT: DIV // 66MHz / 2 = 33MHz

// configure all 16 FPGAs
FPGA 0: fpga_f0.bit
FPGA 1: fpga_f1.bit
FPGA 2: fpga_f2.bit
FPGA 3: fpga_f3.bit
FPGA 4: fpga_f4.bit
FPGA 5: fpga_f5.bit
FPGA 6: fpga_f6.bit
FPGA 7: fpga_f7.bit
FPGA 8: fpga_f8.bit
FPGA 9: fpga_f9.bit
FPGA 10: fpga_f10.bit
FPGA 11: fpga_f11.bit
FPGA 12: fpga_f12.bit
FPGA 13: fpga_f13.bit
FPGA 14: fpga_f14.bit
FPGA 15: fpga_f15.bit

DGCLK0 Select: DC0 62.5MHz // or DC1
DGCLK1 Select: DC2 250Mhz // or DC3
DGCLK2 select: DC5 // or DC6,
DGCLK3 select: DC7 // or DC8,
```

5. Insert the CompactFlash card labeled “Reference Design” into the DN8000K10’s CompactFlash (CF) slot. If the DN8000K10 is in a chassis, there is a remote CF slot on the faceplate of the chassis.
3.4 Connect cables

The configuration circuitry can accept user input to control FPGA configuration or provide feedback during the configuration process. The configuration circuitry IO can also be used to transfer data to and from the user design. This can be done over USB with the provided software, or over RS232 with a serial port terminal.

1. Use the provided ribbon cable to connect the MCU RS232 port (P204) to a computer serial port to view feedback from the configuration circuitry during FPGA configuration. Using the cables provided, the red stripe on the cable indicates pin one. Pin one is labeled “TX” on P204.

2. Run a serial terminal program on your PC. Windows XP users can use Microsoft’s HyperTerminal
   Start->Programs->Accessories->Communications->HyperTerminal
   however we recommend using a good terminal program such as SecureCRT (Vandyke.com). 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)

3. Connect a USB cable (provided) to connect the DN8000K10 to a Windows XP computer. Older Windowses may work, but are not supported. Use connector J203 on the DN8000K10, or if it is installed in a chassis, you can use the remote USB power on the faceplate of the chassis.

4. Connect power supply cables. If you are using an EPS power supply (provided) connect all three power connectors (4, 8 and 24 pin) to P202, P201, and P200. If the DN8000K10 is installed in its chassis, use the red power switch on the faceplate to power on the DN8000K10. If you are operating without the chassis, a jumper must be installed on P203 pin 3 to pin 4, or the EPS power supply will not turn on. There is no power switch on the supplied power supply, so the DN8000K10 will power on immediately after you plug it in. You can also use a standard ATX power supply for the DN8000K10. See Hardware: Power

3.5 Power on

If the board is in a chassis, you can use the front panel “POWER” switch to turn on the DN8000K10. If the board is not in a chassis you must use the power supply’s on/off switch to control the DN8000K10 power. When using the supplied EPS power supply without the
chassis, the DN8000K10 is always powered on as long as a jumper is installed on P203. Use a wall switch or install a toggle switch on the power supply to control power.

When the DN8000K10 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. See Hardware: Configuration for a detailed description of the boot-up process.

3.6 Check Power indicator LEDs
The DN8000K10 monitors all power rails on the board for under voltage. If any of the power supplies are not above their threshold voltages, then the board will be held in reset. Each power rail has a green LED indicator next to the power supply generating that voltage. If your board is held in reset, check these LEDs to see which power supply is failing. A complete list of LEDs on the DN8000K10 is listed in the Hardware: LEDs section.

If your board is being held in reset, a ref LED, DS 17 will flash. This LED is located near the upper right-hand side of the board.

3.7 View configuration feedback over RS232
As the DN8000K10 powers on and the configuration circuitry reads configuration instructions from the CompactFlash card, your RS232 terminal (connected to the serial connector on the Chassis, or directly to the connector P204) will display useful information about the Configuration process. If your Dini Group product ever fails to configure an FPGA, this is the best place to look for diagnostic information.

3.7.1 Watch startup sequence over RS232
The following is a capture from a successful configuration.

<table>
<thead>
<tr>
<th>RS232 Output</th>
<th>Description of Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- FPGAS STUFFED --</td>
<td>The MCU is pre-programmed with the optional equipment that is installed on your DN8000K10.</td>
</tr>
<tr>
<td>00 01 04 05</td>
<td></td>
</tr>
<tr>
<td>-- 8442 INPUT FREQUENCIES --</td>
<td></td>
</tr>
<tr>
<td>PH0:  25.0</td>
<td></td>
</tr>
<tr>
<td>PH1:  16.0</td>
<td></td>
</tr>
<tr>
<td>PH2: 14.318</td>
<td></td>
</tr>
<tr>
<td>FX0_0: 25.5</td>
<td></td>
</tr>
<tr>
<td>FX0_1: 25.0</td>
<td></td>
</tr>
<tr>
<td>FX1_0: 25.0</td>
<td></td>
</tr>
<tr>
<td>FX1_1: 25.0</td>
<td></td>
</tr>
<tr>
<td>CF CTRL = 0xC0</td>
<td></td>
</tr>
<tr>
<td>SMART MEDIA CARD DETECTED</td>
<td></td>
</tr>
<tr>
<td>Reading SM info</td>
<td></td>
</tr>
<tr>
<td>-- SMART MEDIA INFO --</td>
<td></td>
</tr>
<tr>
<td>MAKER ID: BC</td>
<td></td>
</tr>
<tr>
<td>DEVICE ID: 75</td>
<td></td>
</tr>
<tr>
<td>SIZE: 32 MB</td>
<td></td>
</tr>
</tbody>
</table>

SMART MEDIA CARD DETECTED
Reading SM info
-- SMART MEDIA INFO --
MAKER ID: BC
DEVICE ID: 75
SIZE: 32 MB

The MCU searches the media card slots, Compact Flash, Smart Media and IDE (Remove chassis CompactFlash). If a media card is plugged into one of these slots, the Configuration circuit reads the card and follows the configuration commands on it.

If the MCU cannot detect your SmartMedia card, make sure you have not reformatted the card using Windows. If you need to reformat a
**FILEs FOUND ON SMART MEDIA CARD**

- FPGA_F0.BIT
- FPGA_F1.BIT
- FPGA_F4.BIT
- FPGA_F5.BIT
- MAIN~1.TXT
- MAIN.TXT

--- OPTIONS ---

- **Message level set to: 2**
- **Sanity check option for bit files: ON**
- Setting 8442 PH0 CLOCK FREQUENCY to 100
  - M val = 0x0010 N val = 0x0004
- Setting 8442 PH1 CLOCK FREQUENCY to 100
  - M val = 0x0019 N val = 0x0004
- Setting 8442 PH2 CLOCK FREQUENCY to 100
  - M val = 0x001C N val = 0x0004
- **PH CLK DIVIDE VALUE: PH0 = 2**
- **PH CLK DIVIDE VALUE: PH1 = 2**
- **PH CLK DIVIDE VALUE: PH2 = 2**
- **GCLK: 0 MUX = DIV**
- **GCLK_MUX_SEL = 2**
- **GCLK: 1 MUX = DIV**
- **GCLK_MUX_SEL = 2**
- **GCLK: 2 MUX = DIV**

--- CONFIGURATION FILES ---

- FPGA 0: FPGA_F0.BIT
- FPGA 1: FPGA_F1.BIT
- FPGA 4: FPGA_F4.BIT
- FPGA 5: FPGA_F5.BIT

--- CONFIGURING FPGA 0 ---

- Performing Sanity Check on Bit File
- **-- BIT FILE ATTRIBUTES --**
  - FILE NAME: FPGA_F0.BIT
  - FILE SIZE: 00280F9A bytes
  - PART: 4vfX60ff1152
  - DATA: 2005/11/09
  - TIME: 19:14:01
  - Sanitiy check passed

--- CONFIGURING FPGA 1 ---

- Performing Sanity Check on Bit File
- **-- BIT FILE ATTRIBUTES --**
  - FILE NAME: FPGA_F1.BIT
  - FILE SIZE: 003A943B bytes
  - PART: 4vlx100ff151319
  - DATA: 2005/11/09
  - TIME: 19:42:39

--- CONFIGURING FPGA ---

- Performing Sanity Check on Bit File
- **-- BIT FILE ATTRIBUTES --**
  - FILE NAME: FPGA_F2.BIT
  - FILE SIZE: 003A943B bytes
  - PART: 4vlx100ff151319
  - DATA: 2005/11/09
  - TIME: 19:42:39

SmartMedia card, use the utility included on the user CD.

By default, the Configuration circuit reads a file named “Main.txt” on the media card for configuration commands.

The global clocks (G0CLK, G1CLK, G2CLK, REFCLK) are frequency-configurable. The binary sequence M represents the multiplication applied to the installed crystal. The N represents the division applied. See the Hardware: Clocking for details.

FPGA pin assignments for clocks are found in the appendix PINS_OTHER.

The MCU reads the configuration stream file assignments to each FPGA.

Before configuring an FPGA, the configuration circuit (MCU) reads the header information in the .bit file. If the target device in the header does not match the FPGA type on the board, the configuration stream is rejected, and the MCU prints and error message. This check can be disabled using the sanity check:n option. See the Hardware:Configuration:Media card section for more information on this command.

The MCU is configuring FPGA 0 according to instructions in MAIN.TXT

The MCU is configuring FPGA F1 according to instructions in MAIN.TXT
Sanity check passed

........................................
.DONE WITH CONFIGURATION OF FPGA: 1

DN8000k10 MAIN MENU (Nov 11 2005 15:48:09)

1.) Configure FPGAs using "MAIN .TXT"
2.) Interactive configuration menu
3.) Check configuration status
4.) Change MAIN configuration file
5.) List files on Smart Media
6.) Display Smart Media text file
z) READ FROM FLASH REGISTER
x) WRITE TO FLASH REGISTER
v) READ SECTOR
w) READ SECTORS (starting at 0)
o) SEARCH FOR MB RECORD
r) RESET FLASH
g.) Display FPGA Temperatures
h.) Set FPGA Temperature Alarm Threshold

ENTER SELECTION: v

This is the DN8000K10 main menu. A discussion of the available commands are given later this section.

You should see the DN8000K10 MCU main menu. If the reference design is loaded in the Virtex 4 FPGAs, then you should see the above on your terminal. Try pressing 3 to see if the configuration circuit was successful in programming the FPGAs.

ENTER SELECTION: 3

************* CONFIGURATION STATUS ***************
FPGA 0 configured with file: FPGA_F0.BIT
FPGA 1 configured with file: FPGA_F1.BIT
FPGA 4 configured with file: FPGA_F4.BIT
FPGA 5 configured with file: FPGA_F5.BIT

Figure 5

You can verify each FPGA has been successfully configured with a design by looking at the green LED labeled F_DONE next to each FPGA. F0 DONE is DS46. Each green LED is lit when the FPGA next to it is successfully configured. This LED is controlled by the DONE signal of the Virtex 4 SelectMap interface. See the Virtex 4 User Guide.

3.7.2 Interactive configuration
You can save multiple design configuration files for each FPGA on a single CompactFlash card, and use the serial interface’s interactive configuration menu to select which .bit file to use on each FPGA. Select menu option 2.

---= INTERACTIVE CONFIGURATION MENU -=--
HOLD DONES = 0x02 BITS_1 = 0x12
1) Select bit files to configure FPGA(s)
2) Set verbose level (current level = 2)
3) Enable sanity check for bit files
From the Interactive Configuration Menu select option 1, then select a bit file on the CompactFlash card that you would like to use for FPGA F0.

You can also automate this by writing multiple configuration .txt files with alternate configuration settings and use the RS232 menu to select among them. DN8000K10 main menu option 4.) Change Main Configuration File allows this.

3.7.3 Read temperature sensors
The DN8000K10 is equipped with temperature sensors to measure and monitor the temperature on the silicon die of the Virtex 4 FPGAs. If the internal temperature of any FPGA increases beyond a set threshold, the FPGA will become de-configured to protect the FPGA from potential damage, and to warn the user.

According to the Virtex 4 datasheet, the maximum recommended operating temperature of the die is 85°C. If the DN8000K10’s FPGA monitor circuit measures by default sets its reset threshold to 80°C.

If the DN8000K10 is resetting due to temperature overload, you can use the temperature monitor menu to measure the current junction temperature of each FPGA.
ENTER SELECTION: g

-- FPGA TEMPERATURES (Degrees Celsius [+/- 4]) --
F0 29

-- Set FPGA Temperature Alarm Threshold --
(degrees C, decimal values, range [1-127]): 85
Old Threshold: 80
New Threshold: 85
Threshold Updated: 85 Degrees C

Figure 8

The Virtex 4 FPGA can operate at temperatures as high as 120°C without permanently damaging the part, although timing specifications are not guaranteed. The MCU allows you to change the reset threshold from the default of 80°C.

More information about the temperature sensor system can be found in the Hardware: Power: Cooling section

3.7.4 User Serial port

The DN8000K10 has four serial ports (P206, P207, P208, P209) for user use. These ports can be accessed through the MB64B Bus. See Appendix PINS_OTHER for the pin locations of the MB64B signals on each FPGA.

Figure 9

<table>
<thead>
<tr>
<th>Signal (from provided .ucf file) and Schematic</th>
<th>Header Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 RX</td>
<td>P1 TX</td>
</tr>
<tr>
<td>P2 RX</td>
<td>P2 TX</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Remember to enable the Main Bus MB64B bus switches to access the RS232 ports from the FPGA. The switches are enabled by default. See Hardware: Interconnect: Main Bus.

3.8 Check LED status lights

The DN8000K10 has many status LEDs to help the user confirm the status of the configuration process.

1. Check the Reset indicator LED located near the upper, right-hand corner of the board, DS17. If it is flashing red, the board is in reset indicating a power failure or a firmware problem.

2. Check the EPS power supply voltage indication LEDs to confirm that all externally supplied power rails of the DN8000K10 are within 5% of their nominal voltage. From the top, these green LEDs indicate the presence of 3.3V, 12V, 5.0V, and “ATX POWER OK”. A green lit “ATX power OK” indicates that the voltage monitor inside the EPS power supply are within acceptable operating ranges (5V is 4.5 – 5.5V, 3.3V is 3.0-3.6V).

3. Check the Configuration FPGA status green LEDs. These are located along the top right corner of the board.

4. Check the Configuration FPGA DONE status LED, DS108. This LED indicates that the Configuration FPGA has been configured. If this LED is not lit soon after power on, then there may be a problem with the firmware on the DN8000K10. This LED off or blinking may indicate a problem with one of the board's power supplies.

5. Check the FPGA F0 green DONE LED, DS26 to the top of FPGA F0 (If F0 is installed). This green LED is lit when FPGA F0 is configured and operational. This light should be on if you loaded the reference design from the CompactFlash card.
6. Check the DONE green LED for each of the 15 other FPGAs.

7. Check the 4 green user LEDs on FPGA F0. If the Dini Group DN8000K10 reference design was properly loaded in the correct FPGA, then these LEDs will be blinking.

8. If you suspect one or more FPGAs did not configure properly, check the configuration circuitry’s “MCU” status lights. These are red LEDs (DS81 DS82 DS83 DS84) are located near the USB connector. If there has been an error, these four LEDs blink, indicating in binary the number of the FPGA that caused the failure. If there has been no error, there should be two LEDs on and two off. If there was an error, the easiest way to determine the cause of the error is to connect a terminal to the RS232 serial “MCU” port (P204) and try to configure again. Configuration feedback will be presented over this port.

A complete listing of LEDs and their function is found in Chapter Hardware: Test points, LEDs and Connectors

4 Using the USB Controller program

To change settings of the DN8000K10, or to communicate with the reference design (or user design), you can use the provided USB Controller program.

Like the RS232 interface, the USB interfaces allow configuration of the FPGAs, changing clock and other settings. The USB program can also be used to transfer data to and from the User design at high speed.

This section will get you started with the provided software. For detailed information about the reference design and its USB interface, see Reference Design. For detailed information about how the USB program operates and USB drivers, see chapter Controller Software
4.1 Operating the USB controller program

Use the provided USB monitoring software to verify that the design is loaded into the FPGAs.

1. Connect the provided USB cable to your DN8000K10 and to a Windows XP computer, either before or after the DN8000K10 has powered on.

2. When you connect the DN8000K10 via USB to your PC for the first time, Windows XP detects the DN8000K10 and asks for a driver. The board should identify itself as a “DiNi Prod FLASH BOOT”. When the new device detected window appears, select the option "install from a list" -> select "search for the best driver in these locations". Select "include the location in the search" and browse to the product CD in “USB_Software_Applications\driver\windows_wdm\”. Select "finish"

3. After Windows installs the driver, you will be able to see the following device in the USB section of Windows device manager: “DiniGroup DN8000K10 FLASH boot”.

4. Run the USB controller application found on the product CD in “Source Code\USBController\USBController.exe”.

5. This window will appear showing the current state of the DN8000K10. Next to each FPGA a green light will appear if that FPGA is configured successfully. The visual feedback in the window will display which FPGAs are present, which FPGAs are configured, the source of the globals clocks, the frequency of each global clock network at the FPGA input pins, the state of the Main Bus switches.
6. Clear an FPGA of its configuration. Right-click on the image of an FPGA to get a contextual menu. Select Clear FPGA from the menu. The green LED on the board and in the USB Controller window should turn off.

![Figure 12](image)

7. Now configure the FPGA using the contextual menu. Right-click on the FPGA image and select Configure FPGA. In the Open dialog box, select the DN8000K10 reference design from the User CD and click open. “D:\Programming Files\Standard_Reference_Design\LX100\fpga_a.bit” If you are configuring an LX200 or FX60 devices you should select a bit file from the LX200 or FX60 directories instead. After a couple seconds, the USB Controller window should show a green LED appear next to the FPGA to show that it configured successfully.

8. The message box below the DN8000K10 graphic should display some information about the configuration process

```
Done
FPGA F3 cleared successfully.
Doing a sanity check...Sanity Check passed. Configuring FPGA F3 via USB...please wait.
File D:\Programming Files\Standard_Reference_Design \LX100\fpga_a.bit transferred.
Configured FPGA F3 via USB
```

![Figure 13](image)

1. The USB Controller program also allows you to easily configure the clock settings on the DN8000K10. The DN8000K10 reference design requires a 48Mhz or slower clock on GCLK2 to function properly. First, check the current clock frequencies. Select from the menu bar Setting and Info->Read Clock Frequencies

The message window should print frequencies for the four global clocks: GCLK0, GCLK1, GCLK2, and REFCLK, the daughter card-sourced global clocks DC0, DC1, DC2, DC3.
2. Set GCLK2 to 50Mhz. From the menu, select
Settings and Info->Set Global Clock Frequency
From the dialog box, select GCLK2, and type in 48 in the frequency box.

The USB controller program will calculate the PLL settings required to obtain the
closest achievable frequency to 48Mhz using the clocking resources available.

3. Reset the reference design. Since we have changed settings, we should send a logic reset
to the FPGA designs. You can do this from the USB Controller program. From the
menu, select
FPGA Configuration->Reset Logic
This menu option will assert the RESET_FPGA# signal to each FPGA. See the
Appendix Pins Other for the connection of this signal.

More details about each available function of the USB Controller software is found in the USB Software Chapter.

4.2 Using AETEST to run hardware tests

In addition to the Windows GUI application, the Dini Group provides a command line
interface program that provides the same functions. If you will be using Linux or if you plan to
write your own USB software driver, you will be using the source code for this program as a
reference.

AETest is the program that you can use to verify the hardware on the DN8000K10, as well as
to demonstrate the reference design function. The following instructions assume you have a PC
running the Windows XP operating system. The user CD includes a compiled Windows version
of the AETest program. Connect the DN8000K10 to your Windows XP computer with a USB
cable and use aetest_usb in D:\Source Code\USB_Software\aetest_usb\aeusb_wdm.exe. If the
computer asks for a driver, click “Have Disk” and browse to
D:\Source Code\USB_Software\driver\win_wdm\dndevusb.inf

Dini Group does all of its development on Windows XP.

4.2.1 AETest on Linux or Solaris

To use the AETest application on Linux or Solaris, you must compile the source code included
on the User CD. Instructions for compiling AETest are found in Chapter USB Software.

To install the AETest drivers on Linux, <JACK>

4.2.2 Use AETest

The AETest_usb application is compatible with other Dini Group products, and so before
displaying its main menu it displays some USB debug information.
The AETest application then displays its main menu.

Select menu option 2) to interact with the “main bus” interface of the Dini Group DN8000K10 reference design.
The Main Bus menu will only work properly when the Dini Group reference design is loaded, or if your user design has implemented a compatible controller.

Select Option
3) MainBus Write/Read DWORD

0x1000_0000 is address 0 of the DDR2 memory attached to FPGA F1. Enter that address and a data pattern. AETest_usb should report back that it read back the test value. This test will only pass if there is a DDR2 SODIMM installed in DIMM0 and the reference design, including required clock settings, is loaded.

Menu option
5) MainBus Display
dumps 16, 32-bit words from the reference design memory space to the screen.

Enter option M) to return to the main menu.

The AETest_usb application can also be used to configure FPGAs over USB.
Select option
4) Configure FPGA individually via USB
AETest_usb will ask for the path to a configuration .bit file, and an FPGA number to configure. Select a .bit file from the user CD.

5 Board Controls

The DN8000K10 is designed to be operated remotely via the chassis front panel to protect your hardware investment. As a result, the DN8000K10 has very few controls located on the PWB itself.
Press the Hard Reset button. You will see all of the 16 Virtex 4 FPGAs immediately become un-configured. This behavior is exactly the same as if the board were powered off and back on. When you let go of the button, the configuration circuitry will read the CompactFlash card again and attempt to configure the FPGAs.

After the FPGAs are again configured, press and hold the Soft Reset button. You will notice that the blinking LEDs controlled by the reference design will stop blinking. This is because the Soft reset button causes the 16 Virtex 4 FPGAs to receive the FPGA_RESETn signal (active low). This is the signal that the reference design uses to reset its internal logic.

If you have the DN8000K10 installed in the DN8000K10 Chassis, you can also use the front panel buttons to the same effect.

6 Moving On

Congratulations! You have just configured FPGAs on the DN8000K10 and used all of the configuration control interfaces that you must know to start your emulation project. You should use Appendix PINS to create your design constraint (.ucf) files, or you can just use the .ucf files that were included as part of the reference design. All of the source code for the reference design in Verilog is included on the provided CD.
Chapter 3: Controller Software

1 USB Controller

USB Controller application is used to communicate with the DN8000K10.

All USB Controller source code is included on the CD-ROM shipped with the DN8000K10. The USB Controller can be installed on Windows 2000/3/XP. Linux and Solaris users must use the command line interface version, AEtest_usb.

The USB Controller Application contains the following functionality:

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

The following are designed to work with Dini Group’s reference Design.

- Read/Write to FPGA(s)
- Test DDRs/Registers/FPGA Interconnect/Rocket IO

Before shipping a DN8000K10, the Dini Group uses this program to test all of the IO signals, memory interfaces, serial interfaces, clocks, and connectors of your board. The program is compatible with all Dini Group products in the 5000K, 6000K, 7000K and 8000K series products.
1.1 Visual display
The main window of the Dini Group USB Controller program shows the DN8000K10 Graphic.

1.2 Log window
1.3 Menu Options
All of the menu commands available in the USB Controller program can also be set using the CompactFlash card or RS232 interfaces. The details about the implementation of each of the commands available is listed in the Chapter Hardware, section CompactFlash. Note that there is a SmartMedia card interface that behaves identically to the CompactFlash interface. Both of these interfaces are referred to in this section as “CompactFlash”.

1.3.1 Contextual Menu
Right-click on one of the images of the FPGAs to display the FPGA contextual menu. From this menu you can configure and clear FPGAs. The FPGA image that you click determines which FPGA is selected.

a. Configure FPGA Fn
   This menu option allows you to select a configuration stream (.bit file) in an Open dialog box. The selected FPGA will be programmed with that stream.

b. Clear FPGA Fn
   Choosing this menu item causes the selected FPGA to become unconfigured.

1.3.2 File Menu
The File Menu has the following 1 option:
c. Exit
   Closes the USB Controller application

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

1.3.4 FPGA Configuration Menu
The FPGA Configuration Menu has the following options:

a. Refresh
   This menu item refreshed the image displaying the DN8000K10 with the current configured status of each FPGA, the main bus switch settings and current global clock frequency values.

b. Configure via USB (individually)
   After selecting this item, a window will appear and ask which FPGA you want to configure and then which configuration stream (.bit file) you want to configure the selected FPGA with from your computer’s file system. The status of the FPGA configuration process will be logged in the log window and the DN8000K10 image and clock frequencies will be updated after the .bit file has been transferred.

c. Configure via USB using file
   This option 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 which .bit files should be used for each FPGA. The file should be in the following format: The first two characters of each line represents which FPGA you want configured (F0 – F15), this letter should be followed by a colon and then the path to the .bit file to use for this FPGA. The path to the bit file is relative to the directory where this setup file is, or you can use the full path. Below is an example of an accepted setup file:

   F0: fpga_zero.bit
   F1: ../fpga_one.bit
   D:\FPGA
   Programming\Files\Standard_Reference_Design\LX200\fpga_F15.bit

d. Configure via SmartMedia/CompactFlash Card
   This option causes the DN8000K10 to go through its startup sequence by reading configuration instructions from the CompactFlash card. The Section Hardware: Configuration: CompactFlash contains instructions for creating a configuration CF card.
USB SOFTWARE

e. Clear All FPGAs
   This option will immediately un-configure all FPGAs.

f. Reset
   This option sends an active low reset (active for 1ms) to all FPGAs on the signal RESET_FPGAn. Check Appendix Pins Other for the FPGA-side pin out of these signals.

1.3.5 FPGA Memory Menu
This menu contains commands designed to work with the DN8000K10 reference design. All of the commands in this menu cause read and write instructions to happen over the reference design’s main bus interface. Status and control registers for the reference design are all memory-mapped over this main bus interface. For a description of the main bus interface and memory map, see the chapter Reference Design.

a. Write DWORD
   Displays a dialog box that allows you to send data to a signal address in the reference design’s address space. This operation occurs over the “main bus” interface. See the Chapter Reference Design.

b. Read DWORD
   Displays a dialog box that allows you to read data from a signal address in the reference design’s address space. This operation occurs over the “main bus” interface. See the Chapter Reference Design.

c. Write and Read DWORD
   This displays a dialog box that tests the “main bus” interface by writing a value to a location in the reference design’s memory space and immediately reading that address back.

d. Test Address Space
   This item shows a dialog box allowing you to test a range of locations in the memory space of the reference design. This is used to test memory connected to individual FPGAs that is mapped to the reference design’s “main bus” interface.

e. Display Address Space
   This item dumps a range of addresses in the reference design’s “main bus” to the log window under the DN8000K10 image.

f. Test DDR (may be called “Test DN8000K10/PCI DDR”)
   This item automates a test of the memory space on the “main bus” interface that the reference design maps to DDR2 memory.
g. Test DDR single FPGA (may be called “Test DN8000K10/PCI DDR”)  
   This item automates a test of the memory space on the “main bus” interface  
   but allows you to specify which FPGA(s) you would like to test.

h. Set DDR config  
   This item allows you to set the size of the installed DDR2 memory module.  
   (and hence the address range to test)

i. Read DDR config  
   This item displays the current size of the installed DDR2 memory module.

j. Display Memory Map  
   This item displays for reference a table showing the memory map of the default  
   Dini Group reference design.

k. Send Command File  
   This item allows the user to select a script containing “main bus” transactions to  
   automate testing tasks. The menu item displays an Open dialog and asks the  
   user to select a file. The contents of the file should be ASCII text:

   AD 00000000 // sets current address to 0  
   WR 12345678 // writes hex 12 34 56 78 to address 0  
   WR 12345678 // writes hex 12 34 56 78 to address 1  
   RD 1000 // writes 1000 DWORDs (4 byte) to log window

1.3.6 Settings Info

a. Set RocketIO Frequency  
   This item shows a dialog box that allows you to set the clock source and  
   PLL settings of the MGT clocks. The clocks FX0_0 and FX0_1 feed the  
   MGTs of FPGA F0. The clocks FX1_0 and FX1_1 feed the MGTs on  
   FPGA F12.

b. Set Clock input frequency  
   The DN8000K10 firmware is preprogrammed knowing the frequency of  
   the reference crystals installed on the inputs of the PLL circuits on the  
   board. These numbers can be overwritten using this menu option if you  
   have removed these crystals and replaced them with a different frequency  
   crystal.

c. FPGA stuffing information  
   The DN8000K10 firmware comes pre-programmed with information  
   about the optional equipment that is installed on your board. This menu  
   item reads back that information and displays it.
d. Turn Fans On/Off
   The fans on the DN8000K10 cannot be turned off. This menu item cannot be enabled.

c. MCU firmware version
   This menu item reads back the firmware version of the MCU to help the Dini Group debugging.

e. Spartan version
   This menu item reads back the firmware version of the Configuration FPGA to aid the Dini Group debugging.

f. DN8000K10 MB Switch Setup
   This menu item allows the user to change the MB80B and MB64B bus switch settings. These busses can be used as global interconnect between all 16 FPGAs, or they can be disconnected by selecting MB Switch settings to form lower fan-out regional busses. This selection can be made with a byte-wide resolution. For a description of the MB80B and MB64B busses, see Chapter Hardware: Interconnect: Main Bus.

h. DN8000K10 MB Switch Read
   This menu item displays the current state of the MB switches.

i. DN8000K10 Global Clock Synth. Setup
   This menu item allows the user to select the source of the 8 board-global scope clock networks on the DN8000K10. The user can also change the PLL frequency settings of the networks that have PLLs. See section Hardware: Clocks for details.

j. DN8000K10 Global Clock Setup Read
   This option reads back the current clock settings.

k. DN8000K10 Set N dividers

l. DN8000K10 Read N dividers

m. Set Phase Muxes

n. Calculate GCLK Freqs
   This menu item measures and displays the clock frequencies of each of the 8 board-global scope clock networks on the DN8000K10. These measurements will differ slightly from the calculated values.

o. DN8000K10/PCI interconnect test
   This menu runs an automated connectivity test of the inter-FPGA interconnect on the DN8000K10.
p.  DN8000K10/PCI interconnect Menu
This menu is used for operating the inter-FPGA interconnect
characterization test. This test is designed to operate at 350Mhz. The
following sub-menu options are available:
Display Registers AB -?
Display Registers BC -?
Reset TX -?
Reset RX -?
Restart test -?

1.3.7 Settings/Info Menu
The Settings/Info Menu has the following options

(1) Set FPGA RocketIO CLK Frequency
When the DN8000K10 is first powered up the RocketIO Synthesizer MGTCLK inputs
to the FPGAs are inactive, unless programmed using the main.txt file on a
CompactFlash card. (The Epson Oscillators are active.) This menu option allows the
user to specify what frequency the RocketIO Synthesizers should supply to each FPGA.
The supported frequency range is 103MHz – 260MHz. After selecting this option, a
pop-up window will ask which FPGA’s RocketIO Frequency you want to set (or you
can choose to set all to the same frequency), and then what frequency you want. The
USB Controller program will calculate the best PLL settings to achieve this output
frequency. Check the log window to verify what frequency the synthesizers were
actually set at.

(2) Set Global clock frequencies
The clocks on the DN8000K10 are automatically adjusted to the user's desired
frequency by reading the setup file on the CompactFlash card. If you wish to change the
frequency after power-on, or do not want to use a CF card, you can set the frequency in
the USB program.

GCLK0) GCLK0 is generated from a 25MHz crystal. Possible output frequencies are:

<table>
<thead>
<tr>
<th>Frequency</th>
<th>31.25</th>
<th>34.375</th>
<th>37.5</th>
<th>40.625</th>
<th>43.75</th>
<th>46.875</th>
<th>50</th>
<th>53.125</th>
<th>56.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>59.375</td>
<td>62.5</td>
<td>65.625</td>
<td>68.75</td>
<td>71.875</td>
<td>75</td>
<td>78.125</td>
<td>81.25</td>
<td>84.375</td>
</tr>
<tr>
<td>Frequency</td>
<td>93.75</td>
<td>100</td>
<td>106.25</td>
<td>112.5</td>
<td>118.75</td>
<td>125</td>
<td>131.25</td>
<td>137.5</td>
<td>143.75</td>
</tr>
<tr>
<td>Frequency</td>
<td>156.25</td>
<td>162.5</td>
<td>168.75</td>
<td>175</td>
<td>187.5</td>
<td>200</td>
<td>212.5</td>
<td>225</td>
<td>237.5</td>
</tr>
<tr>
<td>Frequency</td>
<td>262.5</td>
<td>275</td>
<td>287.5</td>
<td>300</td>
<td>312.5</td>
<td>325</td>
<td>337.5</td>
<td>350</td>
<td>375</td>
</tr>
<tr>
<td>Frequency</td>
<td>425</td>
<td>450</td>
<td>475</td>
<td>500</td>
<td>525</td>
<td>550</td>
<td>575</td>
<td>600</td>
<td>625</td>
</tr>
<tr>
<td>Frequency</td>
<td>675</td>
<td>700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GCLK1) GCLK1 is generated from a 14.318 MHz crystal. Possible output
frequencies are:
USB SOFTWARE

GCLK2) GCLK2 is generated from a 16.0 crystal. Possible output frequencies are:

32  34  36  38  40  42  44  46  48  50
52  54  56  58  60  62  64  66  68  70
72  74  76  78  80  82  84  86  88  92
96 100 104 108 112 116 120 124 128 132
136 140 144 148 152 156 160 164 168 172
176 184 192 200 208 216 224 232 240 248
256 264 272 280 288 296 304 312 320 328
336 336 344 352 368 384 400 416 432 448
464 480 496 512 528 544 560 576 592 608
624 640 656 672 688

RefClk) RefClk is generated from a 25.0 MHz crystal. Possible output frequencies are the same as GCLK0

(3) FPGA Stuffing Information – This option will display the type of FPGAs that are stuffed on the DN8000K10.

(4) MCU Firmware Version – This option will display the MCU Firmware version in the log window.

(5) Hardware/Firmware Version – This option will display the Board Version along with the Configuration Fpga version.

2 AETest

The AETEST utility program contains the following tests:

- Memory Tests (DDR2)
- Daughter Card Test
USB SOFTWARE

- BAR Memory Range Tests
- RocketIO error rate test
- FPGA interconnect error rate test
- Clock frequency readback

2.1 Running AETEST
AETest_usb.exe is a Windows executable distributed on the user CD ROM. The program can be run in Windows XP.

For the windows version of AETest, a usb driver is required. The driver can be found on the user CD. Dndev.inf. There is no "driver" for linux USB. The Linux version of the usb software runs in user mode. Just run the aetest application aeusb_linux.

2.2 Compiling AETEST
The source for the USB version is found on the User CD
D:\Source Code\USB_Software\aetest_usb

A Make file is provided for compiling on Linux, Solaris, Windows XP, and DOS. To compile, open Make file and change the line
#DESTOS = LINUX

to
DESTOS=LINUX

to target Linux, change
#DESTOS = WIN_WDM

to target windows XP.

Targeting Windows XP required Microsoft Visual Studio 5. Targeting DOS requires DJGPP.

3 Updating the DN8000K10 Firmware
Dini Group may release firmware bug fixes or added features to the DN8000K10. If a firmware update is released you will need to follow the instructions in this chapter.

There are two firmware files that Dini Group may release; the first is a Micro controller (MCU) software update that is stored in a flash memory. This update can be accomplished easily from within the USB Controller application.

The second update that may be required is a Configuration FPGA core update. The configuration data for the Config FPGA is contained in a Xilinx configuration PROM. This update can be accomplished with the Xilinx JTAG programming program, Impact. You will need a JTAG cable like Xilinx Parallel cable IV, or platform USB cable.
### 3.1 Updating the MCU (flash) firmware

To protect against accidental erasure, the MCU firmware cannot be updated unless the board is put in firmware update mode during power-on. Find Switch block 1 (S1) on the DN8000K10.

Move switch S1 #1 to the ON position. Power on the DN8000K10.

Open the USBController program. If the DN8000K10 powered on in firmware update mode, there will be an update flash dialog box as figure on left. Click “Yes” to open flash_flp.hex file.
If the USBController is already opened, click “Refresh” button, there will be an “Update Flash” button near the top of the USBController window. Click on this button (as figure below).

![USBController Window](image)

When the Open… dialog box appears, navigate to the Firmware image file supplied by Dini Group. The file name should be “flash_flp.hex”. Press OK.

The USB Controller should freeze for about 10 seconds while the firmware update is taking place. When the download is complete, the Log window should print, “Update Complete”

Move Switch block S1 #1 to the OFF position to put the DN8000K10 back into normal operation mode. Power cycle the board.

### 3.2 Updating the Configuration FPGA (PROM) firmware

Connect a Xilinx Parallel IV, or Platform USB configuration cable to the parallel port of your computer. The Parallel IV cable requires external power to operate, so you may need to connect the keyboard connector power adapter. When the Parallel IV cable has power, the status LED on Parallel IV turns amber.
Use a 14-pin 2mm IDC cable to connect the Parallel IV cable to the DN8000K10 connector J208.

Power on the DN8000K10. 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 may ask you to open an impact project. Hit cancel.

Choose the menu option File>Initialize Chain

Impact should detect 2 devices in the JTAG chain xcf32p and xc4vlx80. For each item in the chain Impact will direct you to select a programming file for each. For the xcf32p device, select the Configuration FPGA Firmware update file provided by Dini Group. This file should be named prom_flp.mcs. Hit Open. Impact will then ask for a programming file to program the xc4vlx80. Press Bypass.

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 deselected. Press OK. The programming process takes about 2 minutes over the parallel port.
Power cycle the DN8000K10. The new firmware is now loaded. You can close Impact and disconnect the Parallel IV or Platform USB cable.

4 Programmer’s Guide

This section contains information to help the development of your own USB software for use with the DN8000K10. If you do not need to develop your own USB control software, or modify the Dini Group USB controller, you can skip this section. All of the code for AETest and USB Controller is provided on the user CD. Precompiled Windows drivers are also provided for Windows XP. You should also read the section Hardware: Configuration Section before attempting to modify the USB software.

The source code for the Windows version of the USB Controller is provided in D:\Source Code\USB_Software\USBController as a Microsoft Visual Studio 5 project. Visual Studio 5 or later is required to compile this program.
4.1 Cypress CY7C68013A
A Cypress Microcontroller (MCU) with built-in USB support provides the USB interface of the DN8000K10. All communication with the DN8000K10 over USB is initiated by the host (PC) and consists either of a USB vendor request (See USB specification and Cypress datasheet) or a USB bulk transfer.

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 DN8000K10. 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 4 FPGAs.

To begin communication with the DN8000K10, 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 DN8000K10 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.

4.2 Configuration FPGA
The MCU unit controls all of the configuration circuits on the DN8000K10, 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 4 LX40 FPGA. This FPGA provides a memory-mapped interface to all of its IO. This bus is called the ‘Config Bus’.

The configuration FPGA is connected to all of the configuration signals of the Virtex 4 FPGAs, the temperature sensors, status LEDs, SmartMedia card, CompactFlash card, reset buttons, Main Bus switches, RS232 ports, clock synthesizer control signals, global clock multiplexer control signals, FPGA clock inputs, the Main Bus, and an 300-pin expansion header.

The source code for the Configuration FPGA is provided in D:\Source Code\ConfigFPGA
This project can be compiled using Xilinx ISE version 7.1i SP4 or later. Your board may have been built using an LX80 FF1148 or an LX40 FF1148 for the configuration FPGA.

### 4.2.1 Configuration Register Map

The DN8000K10 firmware is updated constantly to add compatibility for new products and add features. The information in this section may change after this manual is printed. The memory space of the MCU is 16 bits wide.

This table describes registers within the Configuration FPGA that are accessible from the memory space of the MCU.

<table>
<thead>
<tr>
<th>Address Range</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x0000-0x1FFF</td>
<td>EEPROM</td>
<td></td>
</tr>
<tr>
<td>0x2000-0xFFFFF</td>
<td>FLASH</td>
<td></td>
</tr>
<tr>
<td>XDATA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XXXX-XXXX</td>
<td>SRAM</td>
<td></td>
</tr>
<tr>
<td>0xDF10</td>
<td>FPGA_BE:</td>
<td>// select byte in addr, read, and data bytes</td>
</tr>
<tr>
<td>0xDF11</td>
<td>FPGA_RD_DATA</td>
<td></td>
</tr>
<tr>
<td>0xDF12</td>
<td>FPGA_WR_DATA</td>
<td></td>
</tr>
<tr>
<td>0xDF13</td>
<td>FPGA_ADDR</td>
<td></td>
</tr>
<tr>
<td>0xDF14</td>
<td>FPGA_ERROR</td>
<td>This register contains an error code after a Main Bus transaction</td>
</tr>
<tr>
<td>0xDF15</td>
<td>GPIF_DATA</td>
<td></td>
</tr>
<tr>
<td>0xDF16</td>
<td>GPIF_ERROR</td>
<td></td>
</tr>
<tr>
<td>0xDF20</td>
<td>HOLD_DONES</td>
<td>This register (1bit) determines if the FPGAs should be held in reset until all FPGAs are configured</td>
</tr>
<tr>
<td>0xDF21</td>
<td>STATES</td>
<td>The state of the state machines in the Configuration FPGA that control FPGA configuration [7:4] = PIF_STATE, [3:0] = FPGA_STATE</td>
</tr>
<tr>
<td>0xDF22</td>
<td>FPGA_FREQ_H</td>
<td></td>
</tr>
<tr>
<td>0xDF23</td>
<td>FPGA_FREQ_L</td>
<td></td>
</tr>
<tr>
<td>0xDF24</td>
<td>MCU_STUFFING1</td>
<td>This register contains a code representing the type of FPGA installed in F0-F7</td>
</tr>
<tr>
<td>0xDF25</td>
<td>MCU_STUFFING2</td>
<td>This register contains a code representing the type of FPGA installed in F8-F15</td>
</tr>
<tr>
<td>0xDF26</td>
<td>SERIAL_SCLK</td>
<td>This register (1 bit) controls the SCLK output connected to the clock synthesizers on the DN8000K10</td>
</tr>
<tr>
<td>0xDF27</td>
<td>SERIAL_CLK_CTRL_1</td>
<td>This register controls the control outputs connected to all of the clock synthesizers on the DN8000K10</td>
</tr>
<tr>
<td>0xDF28</td>
<td>MB80_1_CTRL0</td>
<td>This register holds the output values of the Main Bus switches for MB80B, section 1. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>0xDF29</td>
<td>MB80_1_CTRL1</td>
<td>This register holds the output values of the Main Bus switches for MB80B, section 1. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>Register</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x DF38</td>
<td>MB80_2_CTRL0</td>
<td>This register holds the output values of the Main Bus switches for MB80B, section 2. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>0x DF39</td>
<td>FPGA_COMMUNICATION</td>
<td>This register (1 bit) set enables access to the “Internal Main” registers (below). This register should be set to 0 when the INTERNAL_MAIN registers are not being accessed, or when accessing address space within the 16 user FPGAs.</td>
</tr>
<tr>
<td>0x DF40</td>
<td>MB80_2_CTRL1</td>
<td>This register holds the output values of the Main Bus switches for MB80B, section 2. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>0x DF41</td>
<td>MB64_1_CTRL</td>
<td>This register holds the output values of the Main Bus switches for MB64B, section 1. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>0x DF42</td>
<td>MB64_2_CTRL</td>
<td>This register holds the output values of the Main Bus switches for MB64B, section 2. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>0x DF43</td>
<td>MB64_3_CTRL</td>
<td>This register holds the output values of the Main Bus switches for MB64B, section 3. See Hardware: Interconnect: Main Bus for the effect of this. Each bit enables or disables 8 bits of the Main Bus</td>
</tr>
<tr>
<td>0x DF44</td>
<td>CPLD_CS_N_CTRL</td>
<td>This register holds the (_ bit) value that will be sent to the selected CPLD. See Hardware: CPLD chain</td>
</tr>
<tr>
<td>0x DF45</td>
<td>CPLD_DATA</td>
<td>This holds the (_ bit) value that will be sent to the selected CPLD. See Hardware: CPLD chain</td>
</tr>
<tr>
<td>0x DF46</td>
<td>CPLD_ADDR</td>
<td>This holds the address of the selected register in the CPLD chain. The address also determines which CPLD is selected. See Hardware: CPLD</td>
</tr>
<tr>
<td>0x DF47</td>
<td>GCLK_MSEL_CTRL</td>
<td>Holds the temporary multiplication value that will be sent over the 2-wire bus to the selected global clock synthesizer the next time clocks are set.</td>
</tr>
<tr>
<td>0x DF48</td>
<td>FPGA_PH0_DVAL</td>
<td>Holds the division value that the FPGA should apply to the PH0 clock to generate the PH0 DIV output.</td>
</tr>
<tr>
<td>0x DF49</td>
<td>FPGA_PH1_DVAL</td>
<td>Holds the division value that the FPGA should apply to the PH1 clock to generate the PH1 DIV output.</td>
</tr>
<tr>
<td>0x DF50</td>
<td>FPGA_PH2_DVAL</td>
<td>Holds the division value that the FPGA should apply to the PH2 clock to generate the PH2 DIV output.</td>
</tr>
<tr>
<td>0x DFE</td>
<td>CF_REG_OFFSET</td>
<td>The Registers in the “Internal Main” interface are accessible from the Main Bus interface. See Reference Design. These registers must be enabled by setting the FPGA_COMMUNICATION register (above)</td>
</tr>
<tr>
<td>0x0002</td>
<td>REG_IDCODE</td>
<td>This register returns a known value so the USB Controller program can identify it as the Dini Group reference design.</td>
</tr>
<tr>
<td>0x0004</td>
<td>REG_SCRATCH</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td>REG_HEADERTEST</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>REG_HEADERTEST_STATUS1</td>
<td></td>
</tr>
<tr>
<td>0x07</td>
<td>REG_HEADERTEST_STATUS2</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>REG_HEADERTEST_STATUS3</td>
<td></td>
</tr>
<tr>
<td>0x09</td>
<td>REG_HEADERTEST_STATUS4</td>
<td></td>
</tr>
<tr>
<td>0x0A</td>
<td>REG_HEADERTEST_STATUS5</td>
<td></td>
</tr>
<tr>
<td>0x0B</td>
<td>REG_HEADERTEST_STATUS6</td>
<td></td>
</tr>
<tr>
<td>0x0C</td>
<td>REG_HEADERTEST_STATUS7</td>
<td></td>
</tr>
</tbody>
</table>
4.3 Vendor Request List

The USB Program is updated constantly to add compatibility to new products and to add features. There may be changes to the application after this manual is printed that affect this section.

The following table describes the USB interface presented to the host by the MCU microcontroller. The USB Device identification numbers are Vendor: 0x1234, Device: 0x1234.

<table>
<thead>
<tr>
<th>Vendor Request Name</th>
<th>ID Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VR_UPLOAD</td>
<td>0xc0</td>
<td>Does nothing on DN8000K10</td>
</tr>
<tr>
<td>VR_DOWNLOAD</td>
<td>0x40</td>
<td>Downloads data to the Cypress EEPROM, or to RAM</td>
</tr>
<tr>
<td>VR_ANCHOR_DLD</td>
<td>0xa0</td>
<td>Loads (uploads) EEPROM</td>
</tr>
<tr>
<td>VR_RAM</td>
<td>0xa2</td>
<td>Loads (uploads) external ram</td>
</tr>
<tr>
<td>VR_SET_I2C_ADDR</td>
<td>0xa4</td>
<td>8 or 16 byte address</td>
</tr>
<tr>
<td>VR_GET_I2C_TYPE</td>
<td>0xa6</td>
<td>Returns a revision code of the DN8000K10 MCU firmware</td>
</tr>
<tr>
<td>VR_GET_FPGA_INFO</td>
<td>0xa8</td>
<td>The Cypress MCU behaves as if it were removed and reconnected to USB.</td>
</tr>
<tr>
<td>VR_DBG_FX</td>
<td>0xa9</td>
<td>Force use of double byte address EEPROM (for FX)</td>
</tr>
<tr>
<td>VR_I2C_100</td>
<td>0xaa</td>
<td>Put the I2C bus in 100KHz mode</td>
</tr>
<tr>
<td>VR_I2C_400</td>
<td>0xab</td>
<td>Put the I2C bus in 400KHz mode</td>
</tr>
<tr>
<td>VR_NOSDP_AUTO</td>
<td>0xac</td>
<td>Test code. Does uploads using SUDPTR with manual length override</td>
</tr>
<tr>
<td>VR_RESET</td>
<td>0xad</td>
<td>Causes MCU to go through configuration sequence (Media Card)</td>
</tr>
<tr>
<td>VR_FLASH_ERASE</td>
<td>0xae</td>
<td>Erases MCU Flash firmware</td>
</tr>
<tr>
<td>VR_FLASH_ACCESS</td>
<td>0xb0</td>
<td>Write a byte to flash</td>
</tr>
<tr>
<td>VR_FLASH_SECTOR_ERASE</td>
<td>0xb1</td>
<td>Erases a single sector from the flash</td>
</tr>
<tr>
<td>VR_FLASH_VERSION</td>
<td>0xb2</td>
<td>Reads version of flash code</td>
</tr>
<tr>
<td>VR_DISPLAY_FPGA_INFO</td>
<td>0xb3</td>
<td></td>
</tr>
<tr>
<td>VR_CHECK_FPGA_INFO</td>
<td>0xb4</td>
<td>Returns a string representing if the selected FPGA is configured</td>
</tr>
<tr>
<td>VR_PPC_RS232</td>
<td>0xb6</td>
<td>This Does nothing on the DN8000K10</td>
</tr>
<tr>
<td>FLASH_VERSION_ADDR</td>
<td>0xb8</td>
<td>Value to go into upper address register (MCU_XADD)</td>
</tr>
<tr>
<td>VR_SET_EP0TC</td>
<td>0xbb</td>
<td>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</td>
</tr>
<tr>
<td>VR_SETUP_CONFIG</td>
<td>0xbd</td>
<td>This vendor request must be called to select an FPGA for configuration prior to a bulk transfer containing the configuration stream for that FPGA.</td>
</tr>
<tr>
<td>VR_END_CONFIG</td>
<td>0xbe</td>
<td>This vendor request de-selects an FPGA after configuration and returns the config status of that FPGA (DONE signal)</td>
</tr>
</tbody>
</table>
4.4 USB Reference Design Control

4.4.1 Main Bus accesses

The USB Controller control the DN8000K10 reference design using USB vendor requests and bulk transfers that access the configuration FPGA’s registers. These registers cause “Main Bus” transactions with the user FPGAs. All Main Bus transactions are initiated by the configuration FPGA. 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 DN8000K10 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 address pointer.

Before starting a USB read, or series of reads, you should set the size of the Cypress USB read buffer to be equal to the size of the bulk transfer. This can be accomplished using the VR_SET_EP6TC (0xBB) vendor request described in the Vendor Requests section. If this step is skipped, you may experience slow USB response, or even system instability, depending on the operating system.
4.4.2 Configuration

To access the 16 FPGA configuration interface (SelectMap), a USB interface is provided using Vendor Requests and bulk transfers. The basic configuration process is as follows:

USB Controller sends VR_SETUP_CONFIG (see Vendor Requests) with data representing which FPGA to configure. (F0 is 0x01, F1 is 0x02, F2 is 0x03…)

MCU on receiving this vendor request sets 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 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 is called.

USB Controller 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. For more on the SelectMap interface, see Hardware: Configuration: SelectMap. 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 4 Configuration Guide UG071 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 DN8000K10.

Unless the HOLDDONES option has been activated, the Virtex 4 FPGA will activate, following the activation command imbedded in the .bit stream file. The DONE signal will go high, lighting the green LED next to the FPGA labeled “FPGA Done”.

The USB Controller sends a vendor request VR_SETUP_END. This request deselects the FPGA, so that further bulk requests are interpreted as Main Bus transactions. See Main Bus accesses.

4.4.3 Readback

Not recommended over SelectMap. Suggest Xilinx ChipScope Pro, which work over JTAG.
Chapter 4

Hardware

The DN8000K10 was designed to be the densest emulation platform in the world. To achieve this goal, the FPGA chosen was the Virtex 4 LX200 FPGA, the largest FPGA available. Sixteen of these FPGAs were crammed onto the same PCB for ultra-high performance and maximum interconnect. Every general purpose IO on the largest available package (Flip-Chip BGA 1513) of each FPGA was connected as inter-FPGA interconnect or to an expansion header. The clock and memory interfaces are designed to operate at the full potential of the Virtex 4.

In order to support enough bandwidth to deliver real time data to your design at speed, the DN8000K10 is equipped with two optional Xilinx Virtex 4 FX100 with RocketIO Multi-Gigabit Transceivers. Serial connections over Fibre, Coax ribbon cable, and Coax SMA cables allow for a total aggregate 150 Gb/s off-board communication.

Every new feature offered by the Virtex 4 is fully supported including 1Gbs differential interconnect using Xilinx serdes pin multiplexing. The new 10Gbs MGTs on the DN8000K10 are connected to high-speed off-board connectors. SFP module connectors allow the use of the new Xilinx EMAC modules included in Virtex 4 FX parts.

1 Overview

The resources available to your emulation project include excessive inter-FPGA interconnect, daughter card signals, four memory interfaces.
Below is a block diagram of the DN8000K10

The following sections describe in detail each circuit on the DN8000K10. Note that Schematics appearing in this section are illustrative and may have had details omitted or have been modified for clarity and brevity. If you need to probe, modify or design around the DN8000K10 you will need to examine the complete schematics. See Appendix Schematics. An assembly drawing has also been provided to help you find probe points on the DN8000K10. See Appendix Assembly.

## 2 Configuration Circuit

### 2.1 Overview

The primary purpose of the configuration circuit on the DN8000K10 is to allow the user to configure the 16 Virtex 4 FPGAs using USB, JTAG, or automatically using a CompactFlash card. Secondary functions of the configuration circuit are to provide a USB interface to the user design, provide automatic configuration of the boards flexible clock sources, monitor power and temperature.

The circuit is designed to provide an easy configuration solution that will work out-of-the-box for most users. For special configuration requirements, the configuration circuitry is programmable. The Verilog code for the configuration FPGA and the C code for the microcontroller are both provided on the reference CD. This is provided for information only,
and any development work on these parts of the board should be done with the help of the Dini Group.

2.2 Configuration Options
The DN8000K10 allows the user to select from three FPGA configuration methods.

When a Virtex 4 FPGA is configured, the DONE pin on the FPGA is pulled high. The DN8000K10 has a green LED on each FPGA DONE pin to indicate the configuration status of each Virtex 4 FPGA, and on the configuration FPGA.

2.2.1 CompactFlash
The CompactFlash configuration option allows the user to store FPGA configuration files on a CompactFlash (or SmartMedia) card in the DN8000K10’s media card slot.

When the DN8000K10 powers on, the microcontroller reads the contents of the CompactFlash card. If there is a file called “main.txt” on the root directory of the card, then the DN8000K10 will follow initialization instructions on that file.

Instructions in the main.txt file are read line-by-line and executed in order. The format of the file is non-case-sensitive.

A valid instruction is one of the following:

```
// <comment>
FPGA <fpga name>: <filename>
8442 <synth name> Clock Frequency: <number>Mhz
PH<phase number> Divide By: 2^<n>
GCLK<global clock> Select: <gc source>
DCGCLK<dc clock> select: <dc source> <number>Mhz
FX CLOCK FREQUENCY: <fx clockname> <number>Mhz
FX CLOCK SELECT: <fx clockname> <01>
Verbose Level: <level>
Sanity Check: <yn>
MAIN BUS 0x<address> 0x<data>
```
**H A R D W A R E**

---

**MCU REGISTER WRITE**

0x<short addr> 0x<byte> // configuration register

---

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

<fpga name> can be either F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14 or F15

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

<synth name> can be PH0, PH1, PH2, REF.

<fx clock name> can be FX0_0, FX0_1, FX1_0, FX1_1

PH0-2 feed the global clock networks G0-G2.

REF feeds a dedicated global clock network.

FX0-1_0-1 are four synthesizers feeding the RocketIO MGTs on FPGA F0 and F12.

<number> can be a decimal integer or non-integer between 0 and 800. If a synthesizer is set outside its output range, the configuration circuit will program the synthesizer to the closest frequency that is within range. The FPGA clock inputs can only operate up to 500Mhz.

<phase number> is the number of the synthesizer feeding one of the global clock networks. This can be 0, 1 or 2.

<n> can be an integer from 1 to 15.

<global clock> can be 0, 1 or 2.

<gc source> can be 8442, DIV, SMA or SS.

<dc clock> can be 0, 1, 2 or 3.

<dc source> can be DC0, DC1, DC2, DC3, DC5, DC6, DC7, or DC8.

<fx clock name> can be FX0_0, FX0_1, FX1_0, FX1_1

<01> can be 0 or 1

<level> can be 0, 1, 2, 3, or 4

<yn> can be Y or N

<address> can be an eight-digit hexadecimal number (32 bit)

<data> can be a eight-digit hexadecimal number (32 bit)

<short addr> can be a four-digit hexadecimal number (16 bit)

<byte> can be a two-digit hexadecimal number (8 bit).

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

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>// &lt;comment&gt;</td>
<td>The MCU performs no operation and moves to the next command.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>VERBOSE LEVEL</strong>: <code>&lt;level&gt;</code></td>
<td>This command will set the amount of output the MCU will produce over the RS232 port during configuration. When level is set to 0, the MCU will produce only error output. Before this command is executed, <code>level</code> is set to the default value 3.</td>
</tr>
<tr>
<td><strong>FPGA</strong>: <code>&lt;fpga name&gt;</code>:<code>&lt;filename&gt;</code></td>
<td>The Virtex 4 FPGA specified by <code>&lt;fpga name&gt;</code> will be configured with the file named by <code>&lt;filename&gt;</code>.</td>
</tr>
<tr>
<td><strong>SANITY CHECK</strong>: <code>&lt;yn&gt;</code></td>
<td>If <code>&lt;yn&gt;</code> is set to <code>y</code>, then the MCU will examine the headers in the .bit files on the SmartMedia 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, <code>&lt;yn&gt;</code> is set to the default value <code>y</code>. If you want to encrypt or compress your bit files, you will need to set <code>&lt;yn&gt;</code> to <code>n</code>. Encrypting bit files is not supported or recommended by Dini Group. Previous revisions of Xilinx parts have been vulnerable to permanent damage caused by bugs in the encryption circuitry.</td>
</tr>
</tbody>
</table>
| **GCLK**: `<global clock>` Select: `<gc source>` | The MCU will set the source of the global clock network specified by `<global clock>` to the source specified by `<gc source>`. The default setting is 8442.  

- 8442 causes the clock network to be sourced directly from the output of the clock synthesizer.  
- DIV causes the clock network to be sources from the post-synthesizer divider. This divider value can be set using the PH DIVIDE BY: setting.  
- SMA causes the clock network to be supplied by the SMA clock inputs.  
- SS is reserved for future use. Currently this setting will output 48Mhz onto the global clock network. |
| **8442**: `<synth name>` Clock Frequency: `<number>` MHz | The MCU will set the clock synthesizer specified by `<synth name>` to `<number>` MHz if possible. See the ICS8442 clock synthesizer datasheet for the capabilities of the clock synthesizer. The maximum output frequency of the 8442 clock synthesizer is 800Mhz, although the DN8000K10 will limit this to 500Mhz, because the Virtex 4 global clock inputs cannot operate above this frequency. When the clock synthesizer is outputting a frequency above 401Mhz, the duty cycle is not guaranteed to be 50%. This can be corrected using the Virtex 4 DCMs. If the synthesizer is not capable of outputting the frequency specified by `<N>`, the MCU will set the synthesizer to the closest output frequency available. |
| **MAIN BUS**: `0x<address>` `0x<data>` | The MCU will write the 32 bit value `<data>` to the address `<address>` using the Main Bus interface. This interface is primarily designed for use with the reference design provided, but it can also be used by the user design. See the chapter Reference Design for a description of the Main Bus interface. |
**HARDWARE**

<table>
<thead>
<tr>
<th>HARDWARE</th>
<th>Description</th>
</tr>
</thead>
</table>
| **MCU REGISTER WRITE** | The MCU’s memory space is written at the address `<short addr>` with the data `<byte>`.

This instruction is not designed to be used by most users. See the USB Software chapter for the structure of the MCU address space. |
| **FX CLOCK FREQUENCY:** | The RocketIO clock synthesizer specified by `<fx clockname>` will be set to the frequency specified by `<number>` in MegaHertz. For Synthesizer FX0_1 and FX1_1 to be used by the RocketIO, they will have to be selected using the FX CLOCK SELECT instruction. |
| `<fx clockname>` `<number>`Mhz | |
| **FX CLOCK SELECT:** | The RocketIO clock inputs will be set to one of two possible sources. Only FX0_1 and FX1_1 can have their sources selected. |
| `<fx clockname>` `<01>` | |
| **DCGCLK<de clock> select:** | Each of the four global clock networks supplied by daughtercards can have their sources selected. `<de clock>` specifies to which network the instruction applies. `<de source>` selects which daughtercard clock input pin drives the network. `<number>` Should be set to the known frequency of the clock input. This allows the MCU to correctly configure the PLLs used to de-skew the clock network. If de-skewing is not desired, set this number to 0. |
| `<de clock>` `<de source>` `<number>`Mhz | DCGCLK0 can be sourced from DC0 or DC1. DCGCLK1 can be sourced from DC2 or DC3. DCGCLK2 can be sourced from DC5 or DC6. DCGCLK3 can be sourced from DC7 or DC8. |
| **PH<phase number> Divide By:** | The “Divide” clock specified by `<phase number>` will be set to 2 to the power of `<n>`. The “PH0” divide clock feeds global clock GCLK0, PH1 feeds GCLK1, and PH2 feeds GCLK2. The global clock networks are only supplied with this divided clock when the clock source is set to DIV. Otherwise, this setting will have no effect. |
| `2^<n>` | |
Even if you are planning to configure your Virtex 4 FPGAs using the USB interface, you may want to leave a CompactFlash card in the socket to automatically program your global and MGT clock settings. (Clocks may also be programmed using the provided USB application, or over the MCU RS232 terminal.)

2.2.2 Jtag

Jtag is the only configuration method on the DN8000K10 that does not use the Virtex 4 SelectMap configuration interface. When programming the user FPGAs over a JTAG cable plugged into J13, the DN8000K10 configuration circuitry is not used.

JTAG is a relatively slow interface, and most users should choose to configure over USB, SmartMedia. However, some users like JTAG configuration because it is simple and allows some debugging features not available over SelectMap.

To configure using JTAG, use a Xilinx Parallel cable IV, or Xilinx platform USB cable. The Xilinx program Jtag configuration program Impact can be run from within the ISE software. You should set the configuration speed of your JTAG cable to 4Mhz or below.
The JTAG signals TMS and TCK are buffered and distributed point-to-point to each FPGA. TDO connects to FPGA pin TDO on F15, the TDI pin of J200 connects to the TDI pin of FPGA F3. The order of the JTAG Chain is F3, F2, F1, F0, F4, F5, F6, F7, F11, F10, F9, F8, F12, F13, F14, F15.

If you ordered your DN8000K10 with one or more FPGAs not installed, then a bypass jumper is installed connecting the TDI pin to the TDO pin of the uninstalled FPGA. In this way, the JTAG chain remains intact.

The signal TCK is buffered in a 1:16 buffer. U209.
The signal TMS is buffered in a 1:4 buffer, and fanned out by 4

### 2.2.3 SelectMap
All other configuration methods use the Virtex 4 SelectMap interface. SelectMap is an 8-bit interface described in the Virtex-4 Configuration guide.

![SelectMap Interface](image)

The SelectMap interface on the DN8000K10 is split into four separate interfaces, for electrical reasons. FPGA F0, F1, F2, F3 are on one segment, FPGA F4, F5, F6, F7 are on the next segment. FPGA F8, F9, F10, F11 are on a segment, and FPGA F12, F13, F14, F15 are on the last segment. All selectmap signals on a segment are point-to-point except for data and busy, which are bussed among the four.

The selectmap data signals can be used for the user application as interconnect, but this requires special consideration. The signals must be tri-stated until all FPGAs are done configuring, and re-configuration might be impaired.

### 2.2.4 IDE (Remote Compact Flash)
2.2.5 USB
The USB interface on the DN8000K10 is provided by the Cypress microcontroller unit. To use USB to configure the FPGAs, see Chapter X, The USB application.

USB can also be used to send information to and from your Virtex 4 user design. See Chapter X, the USB Application.

2.3 The Configuration FPGA

The configuration circuitry of the DN8000K10 is built around a Xilinx LX40 FPGA. The SelectMap interface of the user FPGAs is connected directly to the general purpose IOs of the Config FPGA, allowing the maximum flexibility of configuration. The Config FPGA also shares connectivity with the three user FPGAs over a 40-bit Main bus, allowing fast transfers from a computer to the user design over USB.

A powerful FPGA design comes preloaded in the Configuration FPGA allowing users full access to these features right out of the box. For those users who need special configuration behavior, the Configuration FPGA is easily programmed over a JTAG interface. All of the source code for the Configuration FPGA is provided on the user CD.

The Config FPGA is connected to the Cypress microcontroller's address and data busses, and all of the Config FPGA IOs are memory mapped into the Cypress microcontroller's address space. In this way, the microcontroller can monitor and control all configuration processes on the DN8000K10.

2.3.1 Config FPGA Configuration
The Config FPGA is hard-wired into Master Serial mode. After power up, the Config FPGA automatically clocks an external PROM, which serially programs the FPGA over the serial configuration pin (D_IN).

A green LED lights when pin DONE is high to show that the Config FPGA has configured successfully.

Both the Config FPGA and the serial prom are connected in a JTAG chain attached to J14.

As soon as the Config FPGA is configured, it resets the Cypress microcontroller and waits for instructions over the Microcontroller’s address/data bus.
Through the Configuration FPGA, the microcontroller is able to read configuration settings in the main.txt file. When the microcontroller program determines that the user wants to program a user FPGA, not even over JTAG.

This makes it so that if the Configuration FPGA is not active, then none of the FPGAs can configure, not even over JTAG.

2.3.2 Smart Media / Compact Flash

In order to allow high-speed configuration of the user FPGAs from a SmartMedia card, the Config FPGA is connected directly to the data bus of the SmartMedia card socket.

Through the Configuration FPGA, the microcontroller is able to read configuration settings in the main.txt file. When the microcontroller program determines that the user wants to program the Virtex 4 FPGAs from files in the SmartMedia or Compact Flash card, it instructs the Config...
FPGA to activate the Virtex 4's SelectMap interface and load configuration data from the SmartMedia card. The Config FPGA reads data out of the bit files in the Smart media card and sends them over the SelectMap bus to the user FPGAs.

In the schematics, you may notice the Smart Media data bus, SM[7:0], also connects to the microcontroller. These 8 data signals are also used to communicate USB bulk transfer data to the Configuration FPGA. The MCU does not have the ability to communicate with the Smart Media card directly.

### 2.3.3 MCU communication

The MCU communicates to the Config FPGA over its external memory interface, pins D[0:7] and A[0:15]. The Config FPGA is assigned an address range in the microcontroller's memory space.

The 480Mbs data rate of USB 2.0 is too fast for the microcontroller to pass over the memory mapped interface, so data going from USB 2.0 to the main bus or SelectMap interface through the microcontroller instead uses the Cypress CY7C68013 GPIF interface. The GPIF interface is capable of transferring data to and from USB without relying on the processor. The interface is clocked externally by the signal MCU_IFCLK, which is driven at 48Mhz from the Config FPGA.

### 2.3.4 Clock control

The Config FPGA connects to all of the control signals that configure the global clocking network on the DN8000K10. All of these signals are either connected directly to an IO on the Config FPGA, or to an IO expansion CPLD that the Config FPGA controls over a 4-wire bus. For a description of the interface between the Config FPGA and the CPLD IO expansion, see the section *Hardware: Clocking: Expansion CPLD*.

The clock control signals are:

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLCLK_SDATA;</td>
<td>AN15</td>
</tr>
<tr>
<td>ALLCLK_SLOAD;</td>
<td>AC19</td>
</tr>
<tr>
<td>ALLCLK_SRST;</td>
<td>AB18</td>
</tr>
<tr>
<td>PH0CLK_SCLK;</td>
<td>AE21</td>
</tr>
<tr>
<td>PH0_MUXSEL0_2.5V;</td>
<td>AN2</td>
</tr>
<tr>
<td>PH0_MUXSEL1_2.5V;</td>
<td>AN3</td>
</tr>
<tr>
<td>PH1CLK_SCLK;</td>
<td>AF21</td>
</tr>
<tr>
<td>PH1_MUXSEL0_2.5V;</td>
<td>AK6</td>
</tr>
<tr>
<td>PH1_MUXSEL1_2.5V;</td>
<td>AL6</td>
</tr>
<tr>
<td>PH2CLK_SCLK;</td>
<td>AP15</td>
</tr>
<tr>
<td>PH2_MUXSEL0_2.5V;</td>
<td>AL13</td>
</tr>
<tr>
<td>PH2_MUXSEL1_2.5V;</td>
<td>AK13</td>
</tr>
<tr>
<td>FX0_CLK0_SCLK;</td>
<td>AJ22</td>
</tr>
<tr>
<td>FX0_CLK1_SCLK;</td>
<td>AJ21</td>
</tr>
</tbody>
</table>
The following control signals are found connected to five expansion CPLDs. The interface specification between the Configuration FPGA and the expansion CPLDs is found in the Hardware: Configuration: Expansion CPLD section.

The control signals connected to the expansion CPLDs contain power monitor signals and daughter card clock multiplexer signals.

+1.2V\textsubscript{N} _OK\textsuperscript{#} These 16 signals, where \textit{N} is 0-15 are outputs from the power supply monitors. These inputs are currently ignored.
+2.1V\_OK\textsuperscript{#} This signal monitors the RocketIO +2.1V rail. It is currently ignored
+2.5V\_N\_OK\textsuperscript{#} These 4 signals, where \textit{N} is 0-3, monitors the four 2.5V rails. These inputs are currently ignored
+1.8V\_N\_OK\textsuperscript{#} These 2 inputs, where \textit{N} is 0 or 1, monitors the two 1.8V DIMM rails. These inputs are currently ignored.

The next set of signals connects to the five clock buffers used to distribute the daughter card clock network from each daughter card to each FPGA. Since there are only four global clock networks, and 8 daughter card clock sources, a multiplexer is used to select from pairs of two daughter cards.

\textbf{DC\_GCLK\{N\}_\textless X\textgreater\_PLL\_BYPASS}\textsuperscript{#} This signal connects to the PLL\_BYPASS signal of the clock buffer
DC\_GCLK\{N\}_\textless X\textgreater\_SEL0
DC\_GCLK\{N\}_\textless X\textgreater\_SEL1
DC\_GCLK\{N\}_\textless X\textgreater\_SEL2
DC\_GCLK\{N\}_\textless X\textgreater\_SEL3
\textit{<N>} is \{0|1|2|3\}, corresponding to the 4 global DC clocks. See Hardware: Clocks: Daughter card Clocks
\textit{<X>} is \{L0, L1A, L1B, L1C, L1D\}, corresponding to the 2 levels of the clock network.

DC\_GCLK\{0|1|2|3\}_\textunderscore MR Master reset signal to reset the buffers’ PLLs.
DC\_GCLK\{0\}_I0\_DC1\_DC0\# Selects between DC1 and DC0 to source GCLK0
DC\_GCLK\{1\}_I0\_DC3\_DC2\# Selects between DC3 and DC4 to source GCLK1
DC\_GCLK\{2\}_I0\_DC6\_DC5\# Selects between DC5 and DC6 to source GCLK2
DC\_GCLK\{3\}_I0\_DC8\_DC7\# Selects between DC7 and DC8 to source GCLK3
PWR_UP

The signal physical connection can be found in *Appendix Pins Other: Config FPGA*.

For information about how these control signals control the clock network, see *Hardware: Clocking resources*.

### 2.3.5 RS232

The DN8000K10 has two RS232 headers. One (P2) is reserved for use by the microcontroller unit. The other (P1) is connected to the Config FPGA. The Config FPGA has one RX and one TX signal connected to each Virtex 4 FPGA. The Config FPGA will multiplex the RX and TX signals to the Virtex FPGAs to the RS232 header P1. To change the Virtex 4 FPGA that has access to the RS232 headers, you can use the provided USB application program, or you can change the setting on a terminal connected to the Microcontroller unit’s RS232 port (P2).

Since RS232 uses a 12V signal levels, the RS232 signals from the Config FPGA are first buffered through a voltage translation buffer shown below.

On the underside of the DN8000K10, there are two duplicate RS232 ports (P7 and P8) that can be used if an installed daughter card is covering the headers on the front. These duplicate headers are not installed by default, but can be installed on request.

This goes to the front panel LCD display.
2.3.6 Main Bus control
The Configuration FPGA controls main bus using these registers:

See the section Hardware: FPGA Interconnect: Main Bus

2.3.7 LEDs
Here's what they mean:

2.3.8 Remote access
LCD

Buttons

PSON connected to EPS connector.

2.3.9 IIC
There is a single IIC bus on the DN8000K10 connecting all IIC enabled chips on the board. On this bus are three MAX1617A temperature sensing chips (U3, U4, U24), two DDR2 SODIMM sockets, and a serial EPROM. The IIC bus is polled constantly by the MCU for temperature information. Functions for the DDR2 SODIMM IIC and serial prom are currently unimplemented.

2.3.10 Signal Descriptions
The signal pin out list for the Configuration FPGA can be found in the Appendix Pins Other. A brief description of those signals is found here.

2.4 FPGA configuration Process
This is what the DN8000K10 does after power up:

1) EEPROM
2) FLASH
3) Config FPGA
4) SmartMedia, CompactFlash, IDE
5) Load stuff
6) USB
7) RS232

For information regarding the JTAG interface and configuration, See Xilinx publication UG071, Virtex 4 configuration guide.

When configuring over USB or CompactFlash, the FPGAs are configured over the Virtex 4 SelectMap bus.

All SelectMap signals are connected directly to the Configuration FPGA. The SelectMap signals are:

D[0-7] SelectMap data signals.

PROGRAM_B Active low asynchronous reset to the configuration logic. This will cause the FPGA to become un-configured. The documentation refers to this signal as PROGn

DONE After the FPGA is configured, it is driven high 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. The Documentation refers to this signal as RDWR

BUSY When busy is high, the SelectMap configuration stream must stop until BUSY goes low.

CS_B SelectMap chip select. The documentation refers to this signal as CSn

CCLK Signals D[0:7], DONE, RDWR_B and CS_B are clocked on CCLK

Each Virtex 4 FPGA has a complete set of SelectMap signals connected point-to-point to the Config FPGA, except for FPGA B and C, who share signals D[0-7]. All signals are 2.5V CMOS signals except for D0-7 of FPGA A (Signals SELECTMAP_3V_D[0-7])
After a Virtex 4 FPGA is configured, it asserts the signal DONE. On the DN8000K10, these signals have an LED attached to each DONE signal placed near the upper corner of each FPGA.

FPGA F0 LED is DS46, F1 is DS52, and F2 is DS53

If your Virtex 4 FPGA design is failing to produce the intended (or any) results, you should check the DONE light above the FPGA to make sure it is configured correctly. The design files created by Xilinx bitgen software contain a CRC check, so if the Virtex 4 FPGA detects a CRC failure, there was a transmission error during configuration and the DONE light will not glow. The DN8000K10 microcontroller also checks the design files you send to make sure they are compiled for the FPGAs that are installed on your board. If they are not, then the microcontroller unit halts the configuration process. As a result, when the DONE light goes on, you will know that the configuration process was successful.
2.5 MCU

The operation of the Config FPGA is monitored and controlled by a Cypress CY7C68013 (FX2) microcontroller. The microcontroller also has a USB 2.0 interface that can be used to monitor the board, control configuration, or transfer data to and from the user FPGA design. Basic operation can be controlled over an RS232 link from a computer terminal.

The MCU is a standard 8051 instruction-set computer, except that all instructions are executed 4 cycles per instruction. The source code provided by the Dini Group on the user CD is supplied with a Keil microVision IDE project file suitable for creating the firmware binaries for use with the DN8000K10. For firmware update instructions see Controller Software: Updating the Firmware.

2.5.1 RS232

The primary method of user interaction with the DN8000K10 configuration circuitry is the MCU’s RS232 port (P2). The Cypress CY7C68013 has two RS232 pins that are buffered through a 12V voltage translation buffer for use with a standard computer serial port.
The RS232 port will be able to communicate with a standard PC serial port set to 19200 baud, 8 data bits, no parity, and no handshaking. When you connect a computer terminal to the port and power on the DN8000K10, the firmware loaded on the microcontroller unit will display a menu on the terminal. This menu will allow you to control the basic configuration options of the DN8000K10 including configuration, clock frequencies, and the Virtex 4 FPGA RS232 ports.

The RS232 has a built-in UART and generates an interrupt when a character is received.

### 2.5.2 Clocks

The Cypress CY7C68013 is also responsible for configuring the global clocks and rocketIO clock of the DN8000K10. The Cypress CY7C68013 MCU reads the file “main.txt” from the SmartMedia card in the socket (J24), and follows the users clock configuration commands.

See Chapter X, Section X, Clock Resources for clock use.
The 5 ICS8442 clock synthesizers on the DN8000K10 share a serial configuration bus, allowing the MCU to program them. Each synthesizer can be programmed with a different multiplication value and division value. The MCU is connected to this bus on general-purpose IO pins and bit-bangs the ICS8442 serial programming signals. The SDATA, SRST, and SLOAD signals are bussed among all 8 synthesizers (G0, G1, G2, REFCLK, FX0_0, FX0_1, FX1_0, FX1_1). There is a separate SCLK signals for each synthesizer. Since the SLOAD signal is bussed, all 8 synthesizers must be set at the same time.

### 2.5.3 LEDs
The MCU is connected to 4 red LEDs that flash this code:

When LEDs are flashing, there has been an FPGA programming, or CompactFlash card error.

![Figure 22 Config FPGA LEDs](image)

### 2.5.4 Memory space
The Cypress microcontroller has two, 16-bit address spaces for instructions and data. The instruction address space, and the XDATA address space. In the code, memory locations in the XDATA address space are declared with the XDATA modifier. Externally, when the MCU is accessing XDATA memory, it asserts the MEM_OE signal. Both XDATA and instruction memory spaces use the MCU_DATA[7:0] signals to input data into the MCU. On the DN8000K10, this signal is used to select between an FLASH, and the Config FPGA and a SRAM. XDATA is mapped to the Config FPGA and SRAM, and the instruction space is mapped to the Flash.

The Cypress microprocessor has 8KB of internal RAM that is by default mapped to the first 8KB of addresses in the instruction address space. When the microprocessor code reads or writes to this memory, the external MCU_DATA bus is not used, but the internal memory. The
internal memory address range is from 0x0000 to 0x1FFF

When the Cypress MCU is reset (which happens after the Config FPGA is configured), it loads its boot code into its 8kB of internal memory from a serial EPROM (U13). The code in the EPROM instructs the MCU to copy the contents of the FLASH to the internal address range 0x0000 to 0x1FFF. In this way, the external flash can be reprogrammed to allow Dini Group to update the firmware of the DN8000K10.

The format of the data in the EPROM is as follows:
2.5.5 Config FPGA Memory Space

The Configuration FPGA is connected to the MCU_DATA[7:0] signals, the MCU_ADDR[15:0] signals and the MEM_OE signal, allowing it to decode address accesses of the MCU. The Configuration FPGA is programmed to respond to accesses in the XDATA address space in the address range of 0xDF00 to 0xDFFF.

Communication over the MCU memory bus to the Config FPGA is synchronized to the 24Mhz MCU_CLK (X3). For information regarding the timing of transactions on this bus, see the Cypress CY7C68013 user manual.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>XDATA Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>DF00</td>
<td>Used when reading from SM but not configuring</td>
</tr>
<tr>
<td>COMMAND</td>
<td>DF01</td>
<td>Commands for the SM</td>
</tr>
<tr>
<td>ROW_LADDR</td>
<td>DF02</td>
<td>Holds lower 8-bits of SM address</td>
</tr>
<tr>
<td>ROW_HADDR</td>
<td>DF03</td>
<td>Holds upper 8-bits of SM address</td>
</tr>
<tr>
<td>ROW_XADDR</td>
<td>DF04</td>
<td>Holds extra bits of SM address</td>
</tr>
<tr>
<td>Symbol</td>
<td>Address</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NUM_BYTES_0</td>
<td>DF05</td>
<td>Holds lower 8-bits of the number of bytes to read</td>
</tr>
<tr>
<td>NUM_BYTES_1</td>
<td>DF06</td>
<td>Holds upper bits of number of bytes to read in</td>
</tr>
<tr>
<td>BITS_1</td>
<td>DF07</td>
<td>BIT7: mcu_fpga_config_rd BIT6:</td>
</tr>
<tr>
<td>BITS_2</td>
<td>DF08</td>
<td>BIT4: FPGA_DONE BIT3 CPLD_idle BIT2:</td>
</tr>
<tr>
<td>SM_SIGNALS</td>
<td>DF09</td>
<td></td>
</tr>
<tr>
<td>MCU_XADDR</td>
<td>DF0A</td>
<td>Address register for upper FLASH/SRAM bits</td>
</tr>
<tr>
<td>MCU_CNTL</td>
<td>DF0B</td>
<td>Address register for upper FLASH/SRAM bits</td>
</tr>
<tr>
<td>FPGA_SELECT</td>
<td>DF0C</td>
<td>FPGA_select[5:0] = bits 5:0</td>
</tr>
<tr>
<td>PPC_RS232_ABSELECT</td>
<td>DF0D</td>
<td>Does nothing on the DN8000k10</td>
</tr>
<tr>
<td>PPC_RS232_CDSELECT</td>
<td>DF0E</td>
<td>Does nothing on the DN8000K10</td>
</tr>
<tr>
<td>FPGA_CNTRL</td>
<td>DF0F</td>
<td>bits[1:0] = 01 (write address), 10 (data write), 11</td>
</tr>
<tr>
<td>FPGA_BE</td>
<td>DF10</td>
<td>select byte in addr, read, and data bytes</td>
</tr>
<tr>
<td>FPGA_RD_DATA</td>
<td>DF11</td>
<td></td>
</tr>
<tr>
<td>FPGA_WR_DATA</td>
<td>DF12</td>
<td></td>
</tr>
<tr>
<td>FPGA_ADDR</td>
<td>DF13</td>
<td></td>
</tr>
<tr>
<td>FPGA_ERROR</td>
<td>DF14</td>
<td></td>
</tr>
<tr>
<td>GPIF_DATA</td>
<td>DF20</td>
<td></td>
</tr>
<tr>
<td>GPIF_ERROR</td>
<td>DF21</td>
<td></td>
</tr>
<tr>
<td>HOLD_DONES</td>
<td>DF22</td>
<td></td>
</tr>
<tr>
<td>FPGA_FREQ_H</td>
<td>DF24</td>
<td></td>
</tr>
<tr>
<td>FPGA_FREQ_SEL</td>
<td>DF25</td>
<td></td>
</tr>
<tr>
<td>FPGA_FREQ_L</td>
<td>DF26</td>
<td></td>
</tr>
<tr>
<td>MCU_STUFFING1</td>
<td>DF27</td>
<td></td>
</tr>
<tr>
<td>MCU_STUFFING2</td>
<td>DF28</td>
<td></td>
</tr>
<tr>
<td>SERIAL_CLK_CTRL_0</td>
<td>DF29</td>
<td></td>
</tr>
<tr>
<td>SERIAL_CLK_CTRL_1</td>
<td>DF30</td>
<td></td>
</tr>
<tr>
<td>MB80_1_CTRL0</td>
<td>DF36</td>
<td></td>
</tr>
<tr>
<td>MB80_1_CTRL1</td>
<td>DF37</td>
<td></td>
</tr>
<tr>
<td>MB80_2_CTRL0</td>
<td>DF38</td>
<td></td>
</tr>
<tr>
<td>MB80_2_CTRL1</td>
<td>DF39</td>
<td></td>
</tr>
<tr>
<td>FPGA_COMMUNICATION</td>
<td>DF40</td>
<td></td>
</tr>
<tr>
<td>MB64_1_CTRL</td>
<td>DF41</td>
<td></td>
</tr>
<tr>
<td>MB64_2_CTRL</td>
<td>DF42</td>
<td></td>
</tr>
<tr>
<td>MB64_3_CTRL</td>
<td>DF43</td>
<td></td>
</tr>
<tr>
<td>CPLD_CS_N_CTRL</td>
<td>DF44</td>
<td></td>
</tr>
<tr>
<td>CPLD_DATA</td>
<td>DF45</td>
<td></td>
</tr>
<tr>
<td>CPLD_ADDR</td>
<td>DF46</td>
<td></td>
</tr>
<tr>
<td>GCLK_MSEL_CTRL</td>
<td>DF47</td>
<td></td>
</tr>
<tr>
<td>FPGA_PH0_DVAL</td>
<td>DF48</td>
<td></td>
</tr>
<tr>
<td>FPGA_PH1_DVAL</td>
<td>DF49</td>
<td></td>
</tr>
<tr>
<td>FPGA_PH2_DVAL</td>
<td>DF50</td>
<td></td>
</tr>
</tbody>
</table>
These registers can be written to from the USB interface. See *USB Software: Programmers Guide*.

### 2.5.6 Flash and SRAM memory space

The XDATA memory range 0x1FFF to 0xDEFF is mapped to an external SRAM.

<table>
<thead>
<tr>
<th>Inside FX2</th>
<th>Outside FX2 (External RAM 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5 kbytes USB registers and endpoint buffers</td>
<td>56 kbytes external code memory</td>
</tr>
<tr>
<td>0.5 kbytes Scratch RAM</td>
<td>46 kbytes external data memory (RD/WR)</td>
</tr>
<tr>
<td>8 kbytes RAM Code &amp; Data</td>
<td></td>
</tr>
</tbody>
</table>

The XDATA memory range 0x1FFF to 0xDEFF is mapped to an external Flash.

### 2.5.7 USB

The MCU usb interface is built-in to the Cypress FX2 chip hardware. The USB protocol including timing, packetization buffering, error correction and mandatory USB device features are all implemented in hardware, without any code or external hardware interaction. The FX2 supports USB 1.1 (12Mbs) or USB 2.0 (480Mbs). The FX2 is capable of handling USB Control, Bulk, Isochronous and Vendor type transactions. Interrupt type transactions are not supported. The DN8000K10 firmware supports Bulk and Vendor type transfers. The FX2 responds to certain mandatory Control type transfers without requiring microcontroller code. Other than these, the DN8000K10 uses no Control transfers.

Bulk transfers are used to configure FPGAs using the Virtex-4 selectmap bus, and to communicate with the FPGA design using the MB80B bus. The FX2 allows Bulk transfers to 5 endpoints:

- EP0 is controlled by FX2 hardware, and is not used by the DN8000K10 firmware.
- EP1, 4 and 8 are supported by the FX2 and defined, but are not used by the DN8000K10 firmware.
- EP2 is input. (USB Bulk Write). This is used to configure FPGAs and to communicate to the
Main Bus
EP6 is output (USB Bulk read). This is used to read from the Main Bus. A hardware buffer in the FX2 of configurable size (0 to 1024 bytes)

The USB type B connector on the DN8000K10 (J203) is connected directly to the USB pins on the Cypress MCU. Some transient protection is provided.

Figure 24 USB Connector

The Cypress receives a 24Mhz clock from an oscillator (X3). The Cypress internally multiplies this clock to 480Mhz for USB 2.0 and 48Mhz for GPIF operation. The core runs at 24Mhz along with the external memory interface. Communication over this external memory interface is clocked using the MCU_IFCLK signal driven from the MCU at 48Mhz. (The Config FPGA communicates over main bus with the Virtex 4 FPGAs using a separate 48Mhz oscillator (X1) and distributes this clock to each FPGA including itself)

DN8000K10 draws no power from USB. It uses VBUS to signal MCU about USB.
Data transfer through FX2 device starts with a request from a user mode program. Firmware may reply this request immediately or performs data transfer with an external component before it replies the user. As stated before firmwares are free to use their own logic for data transfer, however GPIF is the most powerful and fastest way for communicate with the external world for FX2.

Following is 2 different data transfer processes through GPIF:

3 – Reading data with GPIF FIFO read transitions:
   a. User mode application sends a GPIF FIFO read request to the firmware with bulk data transfer request.
   b. Firmware should have the ability to start GPIF for FIFO read, starting the transition, checking the ready signals if available, checking the amount of data retrieved, ending a request either with error or with some data and so on. If so, firmware reads data from the external component if the ready signal(s) requirement is met and GPIF internal FIFOs are not full, regardless of the user request. For doing this, GPIF toggles FIFO read signal. Toggling depends on the GPIF waveform programmed. Notice that FIFO operations do not use addressing.
   c. External component connected to the GPIF replies read requests and puts data on the data bus sequentially. GPIF also takes this data and puts it into its internal FIFOs.
   d. Firmware replies to the user mode request if there is data to be sent in the GPIF FIFOs. Some firmwares have the ability to skip a request and return with 0 byte or less than the required amount of data.

4 – Writing data with GPIF FIFO write transitions:
   a. User mode application sends a GPIF FIFO write request to the firmware with bulk data transfer request.
   b. Firmware should have the ability to start GPIF for FIFO write, starting the transition, checking the ready signals if available, checking the amount of data retrieved, ending a request either with error or success and so on. If so, firmware sends data to the external component if the ready signal(s) requirement is met and GPIF internal FIFOs are not empty, regardless of the
user request. For doing this GPIF toggles FIFO write signal. With every toggle it puts the next
data byte on the bus. Toggling depends on the GPIF waveform programmed. Notice that FIFO
operations do not use addressing.
c. External component connected to the GPIF replies write requests and gets the data on the
data bus sequentially.
d. Firmware replies to the user mode request with success or error.

A waveform descriptor in internal RAM describes the behavior of each of the GPIF signals. The
waveform descriptor is loaded into the GPIF registers by the FX2 firmware during
initialization, and it is then used throughout the execution of the code to perform transactions
over the GPIF interface. FX2 software enables loading another user supplied waveform
descriptor, (see CeUsb2 API or CeUsb2 generic firmware interface documentation) if this
feature is implemented in the firmware.

The windows WDM drivers for the DN8000K10 are general-purpose kernel-mode drivers
(ezusb.sys) supplied by Cypress. The source code of the driver module is included on the user
CD.

2.5.8 CompactFlash
The CompactFlash card socket data pins are bussed between the Cypress MCU GPIF pins, and
Configuration FPGA IOs on the signals (SMD0-SMD7). These signals lines are used by the
Configuration FPGA to read the main.txt file from the CompactFlash card. They are connected
to the MCU only to preserve pins. The MCU uses these signals as data pins in the GPIF
interface between the MCU and the configuration FPGA. The MCU does not access the
CompactFlash, SmartMedia, or IDE interfaces directly.

See Configuration Options, CompactFlash
3 Clocking

The clocking circuitry on the DN8000K10 is designed for high-speed operation. The flexible clock design should meet the most difficult clocking needs, allowing 7 totally asynchronous, controllable clock sources for the entire sixteen FPGA array.

All global clock networks are differential, LVDS signaled, low skew, low jitter clocks. The programmable clock sources provided by the DN8000K10 are suitable for running 250Mhz DDR2 memory interfaces, and inter-FPGA interconnects as fast as 1Gb/s.

In addition, each Virtex 4 FX100 FPGA is provided with two ultra-low jitter reference clocks suitable for 10Gb/s serial IO for off-board communication.
The primary clocks on the DN8000K10 are the three “global clocks phases” G0, G1 and G2. Each of the global clock networks can be driven by a ICS8442 frequency synthesizer, a pair of differential SMA inputs, or the configuration FPGA for special clock requirements (single-stepping…) The Configuration FPGA connection also allows the division of the outputs from the frequency synthesizer to well below the range of the ICS8442.

Refclk is a global clock tree that is driven from an ICS8442 frequency synthesizer.

DCCLK0, DCCLK1, DCCLK2 and DCCLK3 are global clock networks that are sourced from the MegArray daughtercard connectors.

A feedback clock from the outputs of each of the global clock networks feeds back into the configuration FPGA to allow it to synchronize it’s IO over the Main Bus interface, and to provide a clock counter for the USB Application to display.

The configuration FPGA is supplied with a dedicated 48Mhz clock, making all of the global clock networks available for user.

<table>
<thead>
<tr>
<th>FPGA NAME</th>
<th>F1, F2, F5, F6</th>
<th>F4, F7</th>
<th>F5, F15</th>
<th>F0 (&quot;FX0&quot;)</th>
<th>F9, F10, F13, F14</th>
<th>F8, F11</th>
<th>F12 (&quot;FX1&quot;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 3.1 Global Clocks

The three main global clocks are driven by ICS8442 clock synthesizers, each capable of producing frequencies from 31 to 500Mhz. The clock synthesizers can be programmed from a CompactFlash card, from the USB GUI application (See Controller Software: USB Controller) or left at their default values (GCLK0 100Mhz, GCLK1 100Mhz, GCLK2 38.8Mhz).

<table>
<thead>
<tr>
<th>Clock</th>
<th>Pin 1</th>
<th>Pin 2</th>
<th>Pin 3</th>
<th>Pin 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC_GCLK0_P</td>
<td>P22</td>
<td>P22</td>
<td>P22</td>
<td>H19</td>
</tr>
<tr>
<td>DC_GCLK0_N</td>
<td>P21</td>
<td>P21</td>
<td>P21</td>
<td>H18</td>
</tr>
<tr>
<td>DC_GCLK1_P</td>
<td>AJ21</td>
<td>AJ21</td>
<td>AJ21</td>
<td>AE21</td>
</tr>
<tr>
<td>DC_GCLK1_N</td>
<td>AJ20</td>
<td>AJ20</td>
<td>AJ20</td>
<td>AF21</td>
</tr>
<tr>
<td>DC_GCLK2_P</td>
<td>AG20</td>
<td>AG20</td>
<td>AG20</td>
<td>AE18</td>
</tr>
<tr>
<td>DC_GCLK2_N</td>
<td>AF20</td>
<td>AF20</td>
<td>AF20</td>
<td>AE17</td>
</tr>
<tr>
<td>DC_GCLK3_P</td>
<td>L20</td>
<td>L20</td>
<td>L20</td>
<td>J14</td>
</tr>
<tr>
<td>DC_GCLK3_N</td>
<td>L19</td>
<td>L19</td>
<td>L19</td>
<td>K14</td>
</tr>
<tr>
<td>GCLK_PH0_P</td>
<td>AH20</td>
<td>AH20</td>
<td>AH20</td>
<td>AD21</td>
</tr>
<tr>
<td>GCLK_PH0_N</td>
<td>AH19</td>
<td>AH19</td>
<td>AH19</td>
<td>AD20</td>
</tr>
<tr>
<td>GCLK_PH1_P</td>
<td>L21</td>
<td>L21</td>
<td>L21</td>
<td>L15</td>
</tr>
<tr>
<td>GCLK_PH1_N</td>
<td>K21</td>
<td>K21</td>
<td>K21</td>
<td>L14</td>
</tr>
<tr>
<td>GCLK_PH2_P</td>
<td>AF19</td>
<td>AF19</td>
<td>AF19</td>
<td>AF16</td>
</tr>
<tr>
<td>GCLK_PH2_N</td>
<td>AF18</td>
<td>AF18</td>
<td>AF18</td>
<td>AE16</td>
</tr>
<tr>
<td>REFCLK_P</td>
<td>M21</td>
<td>J21</td>
<td>P20</td>
<td>J16</td>
</tr>
<tr>
<td>REFCLK_N</td>
<td>M20</td>
<td>J20</td>
<td>N20</td>
<td>J15</td>
</tr>
<tr>
<td>GC_SPARE_P</td>
<td>AH18</td>
<td>AH18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GC_SPARE_N</td>
<td>AG18</td>
<td>AG18</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram of Global Clocks](image-url)
The GCLK0, GCLK1 and GCLK2 networks are sourced by a multiplexer allowing the user to select from the ISC8442 clock synthesizer, SMA inputs and the Configuration FPGA. The outputs of the multiplexers are buffered 1:18 to each of the 16 Virtex 4 FPGAs, into the Configuration FPGA, and to a differential testpoint located near the center of the DN8000K10 (labeled “PH0”, “PH1” and “PH2”). The arrival of the clocks at each of these destinations is synchronized.

### 3.1.1 ICS 8442 Phases

The synthesizers are called PH0, PH1, and PH2.

Each ICS8442 has an internal multiplication PLL that can operate between 250 and 700 MHz. With 1, 2, 4, or 8x division on the output, the possible output frequencies are 31.25 – 700MHz. Only 500Mhz output frequencies are allowed by the software, because Virtex-4 FPGAs cannot accept a faster clock on the “GC” pins to which the global clock networks are connected.

The synthesizers are always configured in serial mode. The Serial configuration bus of the ICS8442 is connected to the Cypress MCU GPIF pins and controlled through software. The TEST output is in “FOUT” mode and connected to a test point labeled “PH0CLK”, “PH1CLK” or “PH2CLK”.

The crystal inputs are parallel resonant, fundamental mode. HC49-UP, surface mount crystals. Suitable custom crystals can be found at gedlm.com.
Each global clock is delivered to the FPGA as an LVDS, differential clock. The IO input on this clock should be configured as a differential clock input (the IBUFGDS primitive).

The example below shows the Verilog instantiation of this module, using the ACLK signal.

Wire aclk_ibufds;
IBUFGDS ACLK_IBUFG (.O(aclk_ibufg), .I(ACLK), .IB(ACLKn))

The signal aclk_ibufds should then be fed to either a BUFG or a DCM before being used as an internal clock for FPGA logic.

To set the frequency of these synthesizers via the Main.txt file (described in the Hardware:Configuration Circuit:Options:CompactFlash) use the following syntax

```bash
// main.txt file
8442 <synth name> Clock Frequency: <number>Mhz
```

Where `<synth name>` is PH0, PH1, PH2, REF,

Also, when using the synthesizer for GCLK0 (PH0), GCLK1 (PH1), GCLK2 (PH2), you must correctly set the source of GCLK0,1,2 to “8442”.

```bash
// main.txt file
GCLK<global clock> Select: 8442
```

<global clock> must be 0,1, or 2.

### 3.1.2 RocketIO Clock Synthesizers

The RocketIO clock synthesizers are named FX0_0, FX0_1, FX1_0, FX1_1.

```bash
// main.txt file
FX Clock Frequency: <clock name> <number>Mhz
```
3.1.3 SS (Single Step) Clocks
This feature has not been implemented. Contact support@dinigroup.com for assistance.

Compact Flash Card syntax

There is a break signal. This causes the single-step clock to stop. The signals are negative logic and wire-or’ed together, 4 FPGAs per signal.

BREAK_POINT0# is for FPGAs F0, F1, F2, and F3.
BREAK_POINT1# is for FPGAs F4, F5, F6, and F7.
BREAK_POINT2# is for FPGAs F8, F9, F10, and F11.
BREAK_POINT3# is for FPGAs F12, F13, F14, and F15.

These four signals are then Or’d again and read by the configuration FPGA.

This feature has not been implemented. Contact support@dinigroup.com for assistance.

3.1.4 DIV Clocks
If a clock is required below the frequency threshold of the ICS8442 (31Mhz), you must select the “DIV” as the source of the global clock network. This can be done from the USB Controller software in the clock setup control panel.

// main.txt file
GCLK0 SELECT: DIV // enables divide clock

Figure 27 DIV Clock selection syntax example
3.1.5 User Clock
The DN8000K10 has an SMA pair for each of the three global clock networks G0, G1 and G2 for inputting clocks. The expected signaling standard for the input is LDVS.

J226 G0 UserClock+
J227 G0 UserClock-
J228 G1 UserClock+
J229 G1 UserClock-
J230 G2 UserClock+
J231 G2 UserClock-

Figure 28 User clock input SMAs

The SMA pairs can be found at the lower, right-hand corner of the board. They are labeled “EXTCLK_PH0”, “EXTCLK_PH1”, and “EXTCLK_PH2”
### 3.2 Reference Clock

A fourth ICS8442 synthesizer drives its own, dedicated global clock network. This Synthesizer’s output can be programmed like PH0, Ph1 and Ph2, but cannot be divided by the configuration FPGA and cannot be driven from SMA inputs.

By convention, this clock is set to 200Mhz, and used to provide the reference clock for the Virtex-4 IDELAYCTL module, which requires it. However, REFCLK can be used for any purpose and set to any frequency in the range 31-700Mhz. The reference crystal is 25.0Mhz.

The IDELAYCTL module requires a clock in the range 190-210Mhz. This frequency can also be generated easily from any of the other global clock networks (at least 7.5Mhz) using the FPGA's frequency synthesis capabilities. (The “FX” output of the DCM).

### 3.3 Daughter card clocks

Four global clock networks are provided that are sourced from the daughtercard headers. The network “DC0CLK” can be sourced from the headers DC0 or DC1, “DC1CLK” can be sourced from DC2 or DC3. “DC2CLK” can be sourced from DC5 or DC6. “DC3CLK” can be sourced from DC7 or DC8. The headers DC4 and DC9 have no global clock sourcing capabilities.
The selection of daughtercard clock sources can be made from the USB GUI application, or by entering a command on the CompactFlash main.txt file.

```plaintext
// CompactFlash
// Main.txt file
DCLK: DCB 200MHz
```

Figure 30 Example DC clock select syntax

The clock tree is distributed through two levels of 1:4 PLL buffers with multiplexer inputs. The first level distributes the clock to the second level, and feeds back to the source daughtercards to allow the daughtercard to synchronize the output. The PLL in the buffers has a wide frequency range, but the PLL mode of each PLL must be set in order for the PLL to lock. This setting is made in the main.txt file. The PLL can also be bypassed for low-speed operation, or if synchronization is not needed or desired.
3.4 FPGA clock banks

This is F10 as an example.

NOTE: THIS NO-LOAD RESISTOR IS USED TO PROVIDE PADS FOR ACCESS TO A SPARE GC CLOCK INPUT. RESISTOR LOADED ON TOPSIDE OF BOARD.

DCI is hooked up.

Reset and BREAK are in this bank.

DDR FB should be from the size of the RefClk.

3.5 Expansion CLPD
<table>
<thead>
<tr>
<th>CPLD Name</th>
<th>MCU Accessed Functions</th>
<th>FX Accessed Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q0 (Quadrant 0)</td>
<td>• Control of leaf-level, zero-delay buffers in daughter card global clock trees.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• Monitoring of power regulator comparator outputs: +1.2V_0, +1.2V_1, +1.2V_4, +1.2V_5,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+2.5V_0, +2.5V_2</td>
<td></td>
</tr>
<tr>
<td>Q1 (Quadrant 1)</td>
<td>• Control of leaf-level, zero-delay buffers in daughter card global clock trees.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• Monitoring of power regulator comparator outputs: +1.2V_2, +1.2V_3, +1.2V_6, +1.2V_7,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+2.5V_1, +1.8_0</td>
<td></td>
</tr>
<tr>
<td>Q2 (Quadrant 2)</td>
<td>• Control of leaf-level, zero-delay buffers in daughter card global clock trees.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• Monitoring of power regulator comparator outputs: +1.2V_10, +1.2V_11, +1.2V_14, +1.2V_15,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+2.5V_2, +1.8_1</td>
<td></td>
</tr>
<tr>
<td>Q3 (Quadrant 3)</td>
<td>• Control of leaf-level, zero-delay buffers in daughter card global clock trees.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>• Monitoring of power regulator comparator outputs: +1.2V_6, +1.2V_9, +1.2V_12, +1.2V_13,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+2.5V_3</td>
<td></td>
</tr>
<tr>
<td>CNTR (Center)</td>
<td>• Control of top-level, zero-delay buffers in daughter card global clock trees.</td>
<td>None</td>
</tr>
</tbody>
</table>
Table 6.1  I/O Expansion CPLD Functions

3.5.1 FX CLPD

See: RocketIO: CPLD

4 Reset Topology

4.1 Reset Configuration

The system reset configuration implements the following reset policies:

- If any regulator output is below threshold, the board system reset will be held asserted. This is accomplished by connecting all regulator monitor comparators onto a “wired-OR” bus line, which is sensed by the supervisory chip.
- If any rail connected to the config section FPGA and/or the config FPGA configuration PROM is below threshold, then the FPGA/PROM system is held in an “initialize” or reset state.
- Prior to the config FPGA being configured, the array FPGAs is held in an “initialize” or reset state. Figure 10.2 illustrates the reset arrangement used on the Triton board.

![Diagram of reset topology]
The user may also assert reset by pressing S3, “Hard reset”. This will trigger the reset signal “SYS_RSTn” which is monitored by the Config FPGA. When SYS_RST is asserted, the Config FPGA resets the Virtex 4 FPGAs, causing them to lose their configuration data and deactivate. The Config FPGA also causes a reset on the Microcontroller unit, which will cause the microcontroller to reload configuration instructions from the Smart Media card. USB contact will be lost with the USB host, and the DN8000K10 will have to re-enumerate.

There is a second button, S2 called “Soft Reset”. When this button is pressed, the signal “RESET_FPGAn” is asserted. This signal is sent to the Virtex 4 FPGAs on a user IO pin, and could be used by the user design as a reset signal. This signal is also asserted to all FPGAs after any FPGA becomes configured. RESET_FPGAn is an asynchronous signal.
The above circuit shows how two LTC2900 voltage monitors are daisy chained together to monitor 5 different voltages.

Each FPGA is also connected to a temperature monitor. The Virtex 4 FPGA can easily overheat if a heatsink and fan are not used. The recommended operating temperature for the Virtex 4 is 85°C. The absolute maximum temperature for operation is 125°C. If at any time the junction temperature of the Virtex 4 exceeds 85°C, the Microcontroller will reset the FPGAs, causing them to lose their configuration data. An overheating FPGA could be the result of a misconfiguration, a clock that is set incorrectly, or an inadequate heatsink unit. The heatsink and fan assembly that comes with the DN8000K10 is appropriate for dissipating the amount of heat energy that almost any useful application would be capable of generating.

This circuit shows the MAX1617 temperature monitor. The IIC bus is connected to the Cypress microcontroller.

5 Power

The DN8000K10 gets its power from the 12V rail on 3 EPS power connectors (P200, P201, P202). A 500W EPS power supply is supplied with your board.
PS ON goes to P203.3

ATX_OK lights the LED DS137

The main rails of the DN8000K10 are:
- **1.2V_0, 1.2V_1, 1.2V_2, 1.2V_3, 1.2V_4, 1.2V_5, 1.2V_6, 1.2V_7, 1.2V_8, 1.2V_9, 1.2V_10, 1.2V_11, 1.2V_12, 1.2V_13, 1.2V_14, 1.2V_15, 1.2V_16** – This is the supply rail used for the internal digital logic of Virtex 4 FPGAs. There is one dedicated power supply for each of the 16 user FPGAs, and one for the Configuration FPGA. Separate supply rails for each FPGA provides good isolation between FPGAs from switching noise.

- **1.8V_0, 1.8V_1** – This is used for IO signaling and internal logic of DDR2 SDRAM memory. It is also used to supply some Gigabit optical modules.

- **2.5V_0, 2.5V_1, 2.5V_2, 2.5V_3** – This is used to power FPGA interconnect with low-power LVDS. It is also used as the analog power supply on the Virtex 4 FPGAs. 2.5V_0 provides current to F0, F1, F4, F5. 2.5V_1 provides current to F2, F3, F6, F7. 2.5V_2 provides current to F10, F11, F14, F15, and the configuration FPGA. 2.5_3 provides current to F8, F9, F12, F13. +2.5V_0 supplies current for VCC_FX0_MGT15
+2.5V_3 supplies current for VCC_FX1_MGT15

- **3.3V** – This voltage supplies the LVDS clock distribution trees. It is also used to power the LVTTL interfaces of the Cypress microcontroller, Smart Media and Compact Flash cards.

- **5.0V** – provides signal voltage for IDE interface
  +5.0V supplies current for VCC_FX0_MGT25 and VCC_FX1_MGT25

- **12V** All switching power regulators draw their current from the 12V rail. This rail is drawn directly from the EPS power supply.

- **2.1V** – To provide high-frequency isolation of the MGT RocketIOs on FX0 and FX1, all analog power rails for Multi-gigabit transceivers are derived from a 2.1V switching power regulator PSU140. The following nets are derived from the +2.1V net:
  VCC_FX0_MGT12_0
  VCC_FX0_MGT12_1
  VCC_FX0_MGT12_2
  VCC_FX1_MGT12_0
  VCC_FX1_MGT12_1
  VCC_FX1_MGT12_2

The DN8000K10 also has these secondary rails:

- **VTT0 (0.9V)** – This voltage is used to terminate the SSTL18 signaling of the DDR2 memory module.
- RocketIO 1.2V top, 1.2V right, 1.2V bottom – These linear regulated rails are very low noise supplies for the RocketIO CML inputs and outputs. They are isolated from each other to improve the isolation of multiple RocketIO channels operating simultaneously.

- RocketIO 1.5V – This linearly regulated voltage rail supplies the internal digital logic of the RocketIOs.

- RocketIO 2.5V – This linearly regulated voltage rail supplies the internal analog circuits of the RocketIO.

- XFP VEE.5 – Power for this rail is not supplied by the DN8000K10, but is required for the operation of PECL optical modules. To power this rail, you will need to connect an external power connector to the board from a low-noise voltage supply.

There are test points for measuring the voltage levels of each rail near the top left of the DN8000K10. Each rail is monitored by a voltage monitor circuit, and will cause a reset if any of the primary supplies drop 5% or more below their set points.

There are also LEDs next to each test point to indicate the presence of each voltage rail. These LEDs do not indicate that a rail is within 5% of its set point, only that the rail is present and above 1.6V. A power OK led shows the status of the ATX power supply’s PWR_OK signal. If this LED is lit, then +5.0V and +3.3V (and +12V –12V) are within 5% of their set points.

### 5.1 Switching power supplies

The main power rails for the Virtex 4 FPGAs are produced on board with three 20A switching power supplies, one for each of 1.8V, 2.5V, and 1.2V.

The DN8000K10 is shipped with a fan mounted above the power supplies to help keep them cool. If you need to remove this fan, the DN8000K10 will function properly without it, but be careful not to touch the power supplies with your fingers because they will burn!
Each power supply is protected with a 15A fuse on the inputs. If you need to operate the DN8000K10 with more than 15A of current for a power supply, you can change this fuse, but you need to find a heatsink solution for keeping the Virtex 4 FPGAs cool. The heatsink and fan provided are appropriate for a power consumption of about 10-15W per FPGA.

Each of the primary power rails (5.0, 3.3, 2.5, 1.8, 1.2) is monitored for under voltage. If the voltage monitor circuit detects a low voltage, it will hold the board in reset until the supply is back within 5% of its set point. See section X, Reset Circuit for information on reset.

Reset set to 15us

There are fuses on the power supplies.

5.2 Secondary Power Supplies

The secondary power supplies are derived from a primary supply.

5.2.1 DDR2 Termination Power

DDR2 memory modules use the SSTL18 signaling standard. Properly terminating SSTL18 requires a termination power supply of 0.9V. Since as much as 1.6 Amps of termination current are needed, a switching power supply is required.
The ML6554 produces up to 3A of the required 0.9V termination power rail along with a stable 0.9V reference voltage supply.

5.2.2 MGT Power

Power for the Virtex 4 MGTs are isolated from other supply noise through linear regulators.

U326, U323, U324, U325 supply power for FPGA F0’s RocketIO.
U331, U329, U328, U330 supply power for FPGA F1’s RocketIO.

All analog supplies are referenced against the GND net. GND on the 8000K10 is shared among analog, digital and chassis.
Five linear rails

5.2.3 Optical Module Power

Optional optical modules have a variety of power supply requirements, most of which are met by the DN8000K10.

### XFP power filtering

```
+5.0V
+3.3V
```

```
C1331 0.1uF 10V 20% TANT
C1351 0.1uF 10V 20% TANT
C1354 0.1uF 10V 20% TANT
C1356 0.1uF 10V 20% TANT

L3 4.7uH
L5 4.7uH

VCC50_FX0_XFP0 0.5A
VCCSS_FX0_XFP0 0.75A
```

Since the DN8000K10 has no negative voltage supply, it cannot generate the −5.2V required to supply ECL-based optical transceiver modules. Auxiliary power connectors, J278 (F0) and J280 (F12), is provided to connect to an external voltage supply if ECL signaling is required.
The 1.8V XFP power is supplied from the same +1.8V power regulators that supply the SODIMM modules. Since this supply might be adjusted by the customer for daughter card compatibility, a jumper, R3 and 230 has been added to allow the XFP headers to be disconnected from the SODIMM power nets. Headers, J277 (F0) and J279 (F12) have been connected to the XFP power net to provide an alternative means to supply the XFP headers.

5.2.4 VBATT
The DN8000K10 supports bit stream encryption by providing a battery socket. VBATT is connected to all 16 VBATT pins on the FPGAs. Use battery size 364, Positive side up.
### 5.3 Power distribution

The power system on the DN8000K10 is designed to minimize power loss by distributing current on the 12V net supplied by the EPS power supply connector to each FPGA, and using dedicated point-of-load power converters generate all of the FPGA power requirements. In this way, each lower-voltage rail is able to adjust output voltage independently, and FPGAs’ transient current draws do not affect the voltage available to other FPGAs.

Each FPGA has a dedicated 1.2V switching power supply module associated with it. Each group of 4 FPGAs has a 2.5V switching power supply associated with it. The configuration
FPGA also receives 2.5V power from one of these four 2.5V supplies. FPGAs F1 and F2 share one 1.8V power supply and FPGAs F13 and F14 share one 1.8V switching power supply. The few devices on the board that require 5.0V or 12V power receive this power directly from the EPS power supply connector.

Power for the RocketIO MGT power input pins is derived from a 2.1V switching power supply module. More details on the MGT power system can be found in the section Hardware: MGT Serial Resources: MGT Power.

5.3.1 Bypassing
The power supply bypassing on the DN8000K10 on each of the 1.2V internal rails is sufficient for a fully loaded FPGA design to operate at 500Mhz.

The 2.5V power rails have sufficient bypassing for all inter-FPGA and daughter card signals to be switching simultaneously using the LVDS standard at 500Mhz.

The 1.8V rails have sufficient bypassing for each FPGA bank operating at 1.8V to use the maximum allowed number of IOs using SSTL18_I operating at 350Mhz. See the Virtex 4 User Guide for SSO restrictions.

12V and 5V have no high-speed circuitry attached.

5.3.2 VCCAUX
The FPGA VCCAUX power pins are each supplied by the 2.5V rail associated with each FPGA. The VCCAUX power net is filtered through two ferrite inductors in parallel per FPGA and ceramic capacitors.

5.4 Cooling
Each of the seventeen V4 FPGAs is cooled using a heat sink/fan assembly. These assemblies are mounted to the PWB to avoid placing mechanical stress on the BGA top cases. The selected assembly, Cofan KEM-202B-12, has a $\theta_{ca}$ rating of 1.2$^\circ$C per Watt. The Xilinx packaging specification UG075 shows a maximum junction-to-case temperature of 0.5$^\circ$C/W. They total thermal resistance of the 1.7$^\circ$C/W should allow total device dissipation of 20.5W at 50$^\circ$C ambient with a max junction temperature of 85$^\circ$C.

The chassis also features two cooling fans mounted to the front panel, which exhaust out the back panel. These fans are required when running the DN8000K10 within the enclosure.

According to Xilinx online power estimator tool, a fully utilized FPGA running at 300Mhz can draw more than 30W of power. With this much power used in each FPGA, the DN8000K10 can dissipate 480 or more Watts of heat. For non-trivial designs, a heatsink must be used with the Virtex 4 FPGA.

The configuration circuitry on the DN8000K10 monitors core temperatures of each FPGA, so users should not have to manually check FPGA temperature readings. By default, the
configuration circuit monitors FPGA temperatures continuously, and de-configures any FPGA that exceeds the temperature reset threshold of 80°C. The Xilinx datasheets shows a maximum recommended operating temperature of 85°C, but the default is 80°C to warn the user of the approaching problem. The threshold can be changed to a different value using the RS232 interface, or via CompactFlash card instructions in main.txt.

You can derate timing for temperature at the rate of 0.35% per degree over 85. Xilinx Answer 1116 on the Xilinx website.

Above: The FPGA temperature monitor circuit. The MAX1617’s IIC bus is connected to the Configuration FPGA. The MCU polls all 16 FPGA once every second. The SMBCLK and SMBDATA pins on the temperature monitor device are connected to the DN8000K10 IIC bus. See Hardware: IIC.

Each FPGA active heatsink requires 12V power to operate. Each FPGA has a fan power connector next to it. The fan tachometer signal is connected to the configuration FPGA.

Two three-pin right angle fan connectors are located on the right edge of the board for use with a case fan. These are unused in the optional DN8000K10 chassis assembly.
6 FPGA Interconnect

The DN8000K10 was designed to maximize the amount of interconnect between the two primary Virtex 4 FPGAs A and B. This interconnect was routed as tightly coupled differential LVDS to provide the best immunity to power supply and cross talk noise so that your interconnect can operate at the full switching speed of the output buffers. Following Xilinx recommendations, the interconnect on the DN8000K10 was designed to operate at 1Gb/s for every LVDS pair. (Note 1Gb/s operation requires the fastest speed-grade part, LX200 –12) In order to achieve such breakneck speeds, you will need to operate the busses of signals using a source-synchronous clocking scheme. The interconnect signals on the DN8000K10 have been optimized to operate in data lanes. There are 2 or 3 lanes connecting each horizontal or vertically adjacent FPGA in the 4x4 FPGA array. Each lane has 4 differential LVDS source-synchronous clocks in each direction. These clock signals can also be used as additional data signals, but can only be operated in one direction. For a complete pin out of the Virtex 4 FPGA interconnect, along with a breakdown of lane assignments, see Appendix FPGA pins.

6.1 High speed Serdes support

Clocking incoming data at high speeds required the used of each input’s delay buffer to align each bit. The incoming clock needs to be adjusted and used to clock the inputs within its lane. This process can be automated by the use of the new Virtex 4 feature IDELAYCTL.
For detailed description of the required user design to achieve 1Gbs operation, see Xilinx Application note XAPP704, “High Speed SDR LVDS Transceiver”.

Synchronous clocking and single-ended signaling are still possible on the DN8000K10; you are not required to use high-speed serial design techniques.

Single ended interconnect is recommended for signaling below 300Mhz.

Source-Synchronous clocking (whether single-ended or differential signaling is used) is recommended on interconnect running speeds greater than 180Mhz.

Signals used as source-synchronous clocks in the reference design can also be used as general-purpose data signals, so long as they are not required for a source-synchronous clock.

Figure 8: RX_CLK_AND_DAT Module Block Diagram

Interconnect between FPGAs is arranged into groups of 62 signal data lanes.

In the provided .ucf files and the Appendix Pins, the signal naming convention is the following:

F[0-15] is the index of an FPGA in the 16 Virtex 4 FPGA array.

For example, the signal F0F5_B0n13 connects between FPGA F0 and F5. It is in byte lane 0 (so the appropriate source synchronous clock must be in byte lane 0). This signal is the complement of F0F5_B0p13.

Some pins on the Virtex 4 FPGA are designed to receive source synchronous clocks. These pins have been designated in the provided .ucf files and Appendix Pins with a _CC name extension. These signals can be used for data or clock signals, but must be used in their designated direction.

An appropriate signal to use for a source-synchronous clock for byte lane F0F5 B0 would be the LVDS pair
F5F0_CC_B0p0
F5F0_CC_B0n0
In the direction of F5 as a transmitter, F0 as a receiver. If the entire F0F5 byte lane 0 is in the same direction at the same frequency, then the signal pair F5F0_CC_B0p1, F5F0_CC_B0n1 can be used as data. F0F5_CC_B0p0, F0F5_CC_B0n0 cannot be used since this signal is unidirectional (F0 must be the transmitter)

The total interconnect counts between FPGAs is shown in the figure above.

- F0-F1, F1-F2, F2-F3, F0-F4, F3-F7, F4-F8, F8-F12, F11-F15, F12-F13, F13-F14, F14-F15:
  186 (3, 62-signal data lanes)

- F1-F5, F2-F6, F4-F5, F5-F6, F6-F7, F5-F9, F6-F10, F7-F11, F8-F9, F9-F10, F10-F11, F9-F13, F10-F14:
  124 (2, 62-signal data lanes)

- F0-F5, F1-F4, F1-F6, F2-F5, F2-F7, F5-F6, F8-F13, F8-F5, F9-F4, F9-F6, F9-F12, F9-F14, F10-F5, F10-F7, F10-F13, F10-F15, F11-F14:
  62 (1, 62-signal data lane)
6.2 Main Bus

The main bus “MB” is a 144-bit bus, which interconnects all of the FPGAs in the Virtex 4 array. This bus is logically divided into two sub-busses, one containing 80 bits and the other containing 64 bits. Single-ended, LVCMOS signaling is used on this bus.

In the Appendix Pins and the provided .ucf files, these signals are referred to as
MB80B[0-79]
MB64B[0-63]

The Dini Group reference design uses the signals MB80B[0-36]. Some options in the provided software may drive and read from these signals. Also, for the reference design to work, these signals must not be driven from a user design in another FPGA.

If you implement a design that uses these signals for interconnect you should read the section on the reference design MB interface. The reference design implements a USB interface that you might want to use as-is for debugging your design.

The MB80B bus is also connected to the Configuration FPGA. The MB64B bus can be connected to the Configuration FPGA if an LX80 part is installed in the configuration FPGA slot.
The MB64B section of the main bus does not connect to the FX parts, F0 and F12.

Since not all customers will use the main bus, bus switches are used to isolate branches of the main bus to reduce loading and increase the speed at which these signals can operate. See the above diagram for a drawing of the main bus connections. The bus switches can be opened and closed with an 8-bit resolution. The setting of these switches can be done through the software controller program or the configuration file on the configuration CompactFlash card.

The bus switches are bi-directional. The control signals for the switches are connected to the configuration FPGA.
7 FPGA DRAM Memory Interface

There are four standard 200-pin DDR2 SODIMM module sockets on the DN8000K10. These sockets are supplied with 1.8V power and keyed for use with DDR2 SDRAMs. These four sockets connect to FPGA F1, F2, F13 and F14.

You can use any capacity standard DDR2 SODIMM module with the DN8000K10.

7.1 Clocking

An external 1.8V SSTL buffer is provided to clock the DRAM modules. The differential signals CK0 CK1 and DDR_CLK_FB are length-matched.
A list of the pin outs of the FPGA signal connections to the SODIMM interfaces is in Appendix Pins. For a signal description of the DDR2 interface, see the DDR2 SODIMM module specification.

7.2 Signaling

7.2.1 Termination
External termination given on DQS signals: also CC signals on FPGA.

All signals from DDR memory that don’t have ODT have a 50-ohm termination to ½ the 1.8V power supply (0.9V).

You should use SSTL18_DCI for your DDR2 controller. (DQS signals are SSTL18_II)

7.2.2 Source-synchronous clocking
The bits in the byte lanes are arranged so that internally, the DQS signals can be used as a source-synchronous clock for the DQ signals.

7.3 SODIMM Power supply
The SODIMM slots are provided with 1.8V power as required by the DDR2 SODIMM specification. This voltage can be adjusted if the customer would like to design a custom daughter card for use in the memory sockets.

20A power supply.
HARDWARE

See schematic first.

R45 – dimm 0 and 1
R194 – dimm 2 and 3
NL: 1.8V (default)
11.0: 2.5V
4.75k: 3.3V
Remember to disable XFPs
R2 – XFP FX0
R230 – XFP FX1
LED indicates power more than 1.8V
DS23 dimms 0 and 1
DS143 dimms 2 and 3

-VREF is connected to external 0.9V

7.4 Alternate Memory modules
Dini Group has alternate memory modules available to provide SRAM, RLDRAM and Flash memory. These are compatible with the 1.8V SODIMM slots on the DN8000K10
The expansion system of the DN8000K10 is designed to provide the highest total aggregate bandwidth possible. The connector, pin out and signaling, has been selected to achieve this. The Source-synchronous interface requirements of the Virtex 4 FPGA have been met by the daughter card expansion interface to allow use of the built-in serdes modules. See Xilinx Appnote XAPP704.

The daughter card interface includes 11 MEG-Array connectors, made by FCI. The expansion headers come in two flavors, a 300 and 400 pin varieties. The Triton board mounts four 300-pin connectors on the left side of the array, and four 400-pin connectors on the right side of the array. Two additional 300-pin connectors are used on the right side of the array on the corner FPGAs. One, 300-pin connector is used in the configuration section. Each of the daughter card headers is arranged in “Banks”, correlating to the banks of IO on the Virtex 4 FPGA. Each 300-pin connector contains two full banks of IO, 62 signals, including the special-purpose CC, and VREF pins. Each 400-pin connector contains 3 full Virtex 4 IO banks of 62 signals each.

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, and 10Gbs RocketIO signals.

The total general-purpose IO signal count on the expansion system accessible from the array of 16 user FPGAs is 1240 signals arranged in 20 banks.
8.1 Daughter card Physical

The connectors used in the expansion system are FCI MEG-Array 300-pin plug, 6mm, part #84578-102 and 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 DN8000K10 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. Since the DN8000K10 comes in a metal carrier, it can be operate upside-down to allow access to backside-mounted expansion cards.

The “Plug” of the system is located on the DN8000K10, and the “receptacle” is located on the expansion board. This selection was made to give a greater height selection to the daughter card designer.

8.1.1 Daughter Card Locations

The Triton board mounts four, 300-pin connectors on the left side of the array, and four 400-pin connectors on the right side of the array. Two additional 300-pin connectors are used on the right side of the array on the corner FPGAs. One, 300-pin connector is used in the configuration section.

The drawing below shows a rough location of each daughter card header and it’s associated FPGA number.

This view of the DN8000K10 daughter card locations is from the top of the PCB, looking through to the bottom side. The number in parenthesis indicates the number of “Banks” connected to each expansion header. For physical information for planning an expansion system, see Appendix Assembly.
Every Dini Group product with a MegArray 300 or 400 pin daughter card connector has a standard mounting point position to allow standard daughter cards to be interchangeable among the 8000 series of Dini Group products.

### 8.1.2 Daughter card mounting

The DN8000K10 features a standard metal base plate that gives the board mechanical stability, and provides plenty of mounting points for daughter cards. The daughter card receptacle on the daughter card itself will also be mounted on the backside of the board.

The daughter card should use standoffs to secure itself to the backside of the base plate. The standard chassis that comes with the DN8000K10 will allow it to operate FPGA side down, or on its side to allow physical access to the daughter card and the controls of the DN8000K10.

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 MegArray receptacle.

Note that the components on the topside of the daughter card and DN8000K10 face in opposite directions.
8.1.3 Insertion and removal
Due to the small dimensions of the very high speed MegArray 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 ("A") 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.

8.2 Daughter Card Electrical

The daughter card pin out and routing were designed to allow use of the Virtex 4’s 1 Gbps general purpose IO, and 10Gbs MGT signaling. All signals on the DN8000K10 are all routed as differential, 50-Ohm transmission lines, with means to properly terminate.

No length-matching is done on the PCB for daughter card signals, (except between two ends of a differential pair), because the Virtex 4 is capable of variable-delay input using the built-in IDELAY module.

8.2.1 Pin assignments

The pin out of the DN8000K10 expansion system was designed to reduce cross talk to manageable levels while operating at full speed of the Virtex 4. 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 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-endedly, do not interfere with each other excessively.

All high-speed signals on the DN8000K10, including daughter card signals, are routed against a ground potential reference plane.

The RocketIO signals on daughter cards DC0 and DC3 are arranged in a GSSG. These signals can only be used in a differential configuration, and cross talk between the two signals is complementary and beneficial. On the Host, these signals are routed as 110-Ohm differential signals. 110 Ohm signaling was chosen because the Meg Array connector system in the 14mm stack height configuration is slightly inductive. For the 35ps rise time of a Virtex 4 RocketIO CML signal, the Meg Array connector appear very much like a 110Ohm transmission line with a 70ps transmission delay. Daughter cards designed to work with RocketIO at the highest data rates should account for this during design.
You may want to read the following references for designing a daughter card using 110Ohm RocketIO signals:

Howard Johnson, High-Speed Signal Propagation, p. 315 Matching Pads
Xilinx Virtex 4 MGT Users Guide See: TXTERMTRIM

The central columns of the connector pin out use a closely coupled, differential pair pin arrangement, which is uniformly surrounded by ground pins. These differential pins are used for RocketIO connections on FX parts. All other signals use a “checkerboard” type of ground arrangement. This allows the signals to be used as high-speed, single-ended, or as loosely coupled differential pairs.

There are two types of connectors on the DN8000K10, 300 and 400 pins. The first 300 pins on both types of connectors are identical. This should allow a 300-pin connector to be installed on a 400-pin land pattern on a daughter card to allow limited functionality in 300-pin daughter card positions. The “Banks” of signals are segregated. On the 300-pin connector, there are extra signals in the checkerboard pattern that are left as NC.
Below is a graphic representation of the pin assignments for the 300- and 400-pin connectors. Note that this is a view from the backside of the connector. The green boxes represent ground connections.
Special purpose pins are described below.

8.2.2 CC, VREF, DCI
Some of the signals connected to the daughter card expansion headers are “clock-capable”; the inputs on the Virtex 4 FPGA can be used for source-synchronous clocking. In the Appendix Pins Other and provided constraints (.ucf) files, these signals are post pended with “_CC”. See Appendix Pins Other or the above diagram for the location of these pins.

Pins declared as “VREF” pins by Xilinx have a defined placement on the daughter card pin out to allow the daughter card to define a logic threshold as required by some standards.

DCI is used on all FPGA IO banks connected to a daughter card header. The reference resistance is 50 Ohms. A Virtex 4 bank has 64 pins. Of each bank connected to a daughter card header, 62 signals are connected to the header, and 2 are used as DCI reference pins.

8.2.3 Global clocks
The daughter card pin out defines 6 clock input pins. These clock inputs are intended to be used a 3 differential signals. Two clock signals GCA and GCB connect to the “GC” clock inputs in the FPGA. These clocks can be used as global clocks from within the FPGA code of the FPGA that connects to the daughter card, but not globally to the entire DN8000K10.

The GCC signal on every daughter card except DC4 and DC9 connects to the “Daughter card Global Clock” network. This clock input can be distributed to all 16 FPGAs on the DN8000K10. For more information on the daughtercard clock network, see Hardware: Clocks: Daughter card Clocks.

For distributing an FPGA-global clock to the entire board, the Dini Group standard daughter card DNMEGOBS-300 or DNMEGOBS-400 is capable of driving the global clock network from its GCC pin.

8.2.4 Power and Reset
The +3.3V, +5.0V and +12V power rails are supplied to the Daughter card headers. Each pin on the MegArray 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_RSTn signal. This signal causes the entire DN8000K10 to reset, losing all FPGA configuration data and resetting the configuration circuitry.

8.2.5 MGT Signals

Also see Hardware: MGT Serial Resources: The connections: Daughter cards.

8.2.6 VCCO Voltage

The signal voltage on the Daughter card interface is defined by the daughter card by setting the voltage on the VCCO0, VCCO1 and VCCO2 pins. Since the daughter card provides all the current necessary for the FPGA on the DN8000K10 to communicate over the daughter card interface, the daughter card designer will have to determine the current requirements of the interface.

The each VCCO_ net supplies power for the host board FPGA for one entire bank. Bank 0 (VCCO0) includes the signals B0L[0-31]. Bank 1 (VCCO1) includes B1L[0-31]. Bank 2 (VCCO2) includes signals B2L[0-31] (on the 400 pin headers only)

FPGA VCCO power is provided by the daughter card for each connected bank. This allows the daughter card to define the I/O standard to be used on the bank.

8.2.7 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 VCCO voltage impressed by the daughter card should be less than 3.75 to prevent destruction of the Virtex 4 IOs connected to that daughter card.
8.3 Daughter card Types

To avoid incompatibilities with future products, the Dini Group has defined daughter card sizes that it uses for all of its standard daughter cards using the MegArray connector system.
8.3.1 Types 1 and 2 Short (300pin & 400pin Short)
The DNMEGOBS-300 is Type 1. The DNMEGOBS-400 is Type 2 short. See Ordering Information: Option Equipment: Daughter cards

The mounting hole positions are standard, and the DN8000K10 has holes in its base plate to accommodate these holes. See Appendix: Assembly

8.3.2 Type 3 (300pin)
The 300-pin connectors connected to FPGAs F3 and F15 have the GCLKC pins, however, these pins do not connect to the global clock network as described in the Global Clock section of the Daughter card electrical specification. Instead, these pins connect to FPGA-global input pins on the associated FPGA. The two type 3 connectors on the DN8000K10 are:

P104, DC4, F3
P109, DC9, F15

These two daughter card headers also do not follow the header spacing requirement and a type-2 daughter card plugged in to these connectors will extend slightly beyond the edge of the DN8000K10's outline.
The 400-pin connectors connected to these FPGAs are normal type 2, 400 pin connectors.

### 8.3.3 Type 0 (300 FX)

300-pin daughter cards DC0 and DC3 (connected to FX0 and FX12) use a different pin out. These connectors do not have general-purpose IO. Instead, Virtex 4 MGT signals are provided.

<table>
<thead>
<tr>
<th>Connector Pin</th>
<th>Signal Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>E7 CH1_RXP</td>
<td>F7 CH1_RXN</td>
</tr>
<tr>
<td>E9 CH1_TXP</td>
<td>F9 CH1_TXN</td>
</tr>
<tr>
<td>E11 CH2_TXP</td>
<td>F11 CH2_TXN</td>
</tr>
<tr>
<td>E13 CH2_RXP</td>
<td>F13 CH2_RXN</td>
</tr>
<tr>
<td>E15 CH3_RXP</td>
<td>F15 CH3_RXN</td>
</tr>
<tr>
<td>A29, B30</td>
<td>MGTCLK</td>
</tr>
<tr>
<td>E1, F1</td>
<td>GCLKA</td>
</tr>
<tr>
<td>E3, F3</td>
<td>GCLKB</td>
</tr>
<tr>
<td>E5, F5</td>
<td>GCLKC</td>
</tr>
<tr>
<td>J2</td>
<td>RSTn</td>
</tr>
</tbody>
</table>

The RX and TX pins are connected to the Virtex 4 RocketIO inputs and outputs. See Hardware: MGT Serial Resources: Connections: Daughter card. The GCLKA-GCLKC connects as described in the Daughter card electrical section, including GCLK’s connection to the global clock distribution network.

### 9 LEDs

The following table lists all of the LEDs on the DN8000K10.

LED conventions:

GREEN GOOD

RED BAD.

<table>
<thead>
<tr>
<th>Assembly number</th>
<th>Led name</th>
<th>Color</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS25</td>
<td>+1.2V_0_OK#</td>
<td>GREEN</td>
<td>The net +1.2V_0 is &gt; 1.04V</td>
</tr>
<tr>
<td>DS45</td>
<td>+1.2V_1_OK#</td>
<td>GREEN</td>
<td>The net +1.2V_1 is &gt; 1.04V</td>
</tr>
<tr>
<td>DS104</td>
<td>+1.2V_10_OK#</td>
<td>GREEN</td>
<td>The net +1.2V_10 is &gt; 1.04V</td>
</tr>
</tbody>
</table>
**HARDWARE**

DS85  +1.2V_11_OK# GREEN The net +1.2V_11 is > 1.04V
DS133 +1.2V_12_OK# GREEN The net +1.2V_12 is > 1.04V
DS130 +1.2V_13_OK# GREEN The net +1.2V_13 is > 1.04V
DS134 +1.2V_14_OK# GREEN The net +1.2V_14 is > 1.04V
DS146 +1.2V_15_OK# GREEN The net +1.2V_15 is > 1.04V
DS58  +1.2V_16_OK# GREEN The net +1.2V_16 is > 1.04V (Config FPGA power)
DS43  +1.2V_2_OK# GREEN The net +1.2V_2 is > 1.04V
DS1   +1.2V_3_OK# GREEN The net +1.2V_3 is > 1.04V
DS56  +1.2V_4_OK# GREEN The net +1.2V_4 is > 1.04V
DS86  +1.2V_5_OK# GREEN The net +1.2V_5 is > 1.04V
DS61  +1.2V_6_OK# GREEN The net +1.2V_6 is > 1.04V
DS75  +1.2V_7_OK# GREEN The net +1.2V_7 is > 1.04V
DS106 +1.2V_8_OK# GREEN The net +1.2V_8 is > 1.04V
DS107 +1.2V_9_OK# GREEN The net +1.2V_9 is > 1.04V
DS42  +1.8V_0_GT_2.2V N RED The net +1.8V_0 is > 2.2V
DS23  +1.8V_0_OK# GREEN The net +1.8V_0 is > 1.6V
DS144 +1.8V_1_GT_2.2V N RED The net +1.8V_1 is > 2.2V
DS143 +1.8V_1_OK# GREEN The net +1.8V_1 is > 1.6V
DS87  +2.1V_OK# GREEN The net +2.1V is > 1.6V
DS80  +2.5V_0_OK# GREEN The net +2.5V_0 is > 2.2V
DS22  +2.5V_1_OK# GREEN The net +2.5V_1 is > 2.2V
DS140 +2.5V_2_OK# GREEN The net +2.5V_2 is > 2.2V
DS105 +2.5V_3_OK# GREEN The net +2.5V_3 is > 2.2V
DS50  +3.3V_OK# GREEN The net +3.3V is > 2.9V
DS49  +5.0V_OK# GREEN The net +5.0V is > 4.0V
DS27  F0_LED0 GREEN User-controlled LED from FPGA F0.
DS28  F0_LED1 GREEN User controlled LED from FPGA F0.
DS29  F0_LED2 GREEN User controlled LED from FPGA F0.
DS26  F0_LED3 GREEN User controlled LED from FPGA F0.
DS30  F1_LED0 GREEN User controlled LED from FPGA F1.
DS31  F1_LED1 GREEN User controlled LED from FPGA F1.
DS32  F1_LED2 GREEN User controlled LED from FPGA F1.
DS33  F1_LED3 GREEN User controlled LED from FPGA F1.
DS96  F10_LED0 GREEN User controlled LED from FPGA F10.
DS97  F10_LED1 GREEN User controlled LED from FPGA F10.
DS98  F10_LED2 GREEN User controlled LED from FPGA F10.
DS99  F10_LED3 GREEN User controlled LED from FPGA F10.
DS100 F11_LED0 GREEN User controlled LED from FPGA F11.
DS101 F11_LED1 GREEN User controlled LED from FPGA F11.
DS102 F11_LED2 GREEN User controlled LED from FPGA F11.
DS103 F11_LED3 GREEN User controlled LED from FPGA F11.
DS113 F12_LED0 GREEN User controlled LED from FPGA F12.
<table>
<thead>
<tr>
<th>Device</th>
<th>LED Number</th>
<th>Color</th>
<th>FPGA Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS114</td>
<td>F12_LED1</td>
<td>GREEN</td>
<td>F12</td>
</tr>
<tr>
<td>DS115</td>
<td>F12_LED2</td>
<td>GREEN</td>
<td>F12</td>
</tr>
<tr>
<td>DS116</td>
<td>F12_LED3</td>
<td>GREEN</td>
<td>F12</td>
</tr>
<tr>
<td>DS117</td>
<td>F13_LED0</td>
<td>GREEN</td>
<td>F13</td>
</tr>
<tr>
<td>DS118</td>
<td>F13_LED1</td>
<td>GREEN</td>
<td>F13</td>
</tr>
<tr>
<td>DS119</td>
<td>F13_LED2</td>
<td>GREEN</td>
<td>F13</td>
</tr>
<tr>
<td>DS120</td>
<td>F13_LED3</td>
<td>GREEN</td>
<td>F13</td>
</tr>
<tr>
<td>DS121</td>
<td>F14_LED0</td>
<td>GREEN</td>
<td>F14</td>
</tr>
<tr>
<td>DS122</td>
<td>F14_LED1</td>
<td>GREEN</td>
<td>F14</td>
</tr>
<tr>
<td>DS123</td>
<td>F14_LED2</td>
<td>GREEN</td>
<td>F14</td>
</tr>
<tr>
<td>DS124</td>
<td>F14_LED3</td>
<td>GREEN</td>
<td>F14</td>
</tr>
<tr>
<td>DS125</td>
<td>F15_LED0</td>
<td>GREEN</td>
<td>F15</td>
</tr>
<tr>
<td>DS126</td>
<td>F15_LED1</td>
<td>GREEN</td>
<td>F15</td>
</tr>
<tr>
<td>DS127</td>
<td>F15_LED2</td>
<td>GREEN</td>
<td>F15</td>
</tr>
<tr>
<td>DS128</td>
<td>F15_LED3</td>
<td>GREEN</td>
<td>F15</td>
</tr>
<tr>
<td>DS34</td>
<td>F2_LED0</td>
<td>GREEN</td>
<td>F2</td>
</tr>
<tr>
<td>DS35</td>
<td>F2_LED1</td>
<td>GREEN</td>
<td>F2</td>
</tr>
<tr>
<td>DS36</td>
<td>F2_LED2</td>
<td>GREEN</td>
<td>F2</td>
</tr>
<tr>
<td>DS37</td>
<td>F2_LED3</td>
<td>GREEN</td>
<td>F2</td>
</tr>
<tr>
<td>DS38</td>
<td>F3_LED0</td>
<td>GREEN</td>
<td>F3</td>
</tr>
<tr>
<td>DS39</td>
<td>F3_LED1</td>
<td>GREEN</td>
<td>F3</td>
</tr>
<tr>
<td>DS40</td>
<td>F3_LED2</td>
<td>GREEN</td>
<td>F3</td>
</tr>
<tr>
<td>DS41</td>
<td>F3_LED3</td>
<td>GREEN</td>
<td>F3</td>
</tr>
<tr>
<td>DS63</td>
<td>F4_LED0</td>
<td>GREEN</td>
<td>F4</td>
</tr>
<tr>
<td>DS64</td>
<td>F4_LED1</td>
<td>GREEN</td>
<td>F4</td>
</tr>
<tr>
<td>DS65</td>
<td>F4_LED2</td>
<td>GREEN</td>
<td>F4</td>
</tr>
<tr>
<td>DS66</td>
<td>F4_LED3</td>
<td>GREEN</td>
<td>F4</td>
</tr>
<tr>
<td>DS67</td>
<td>F5_LED0</td>
<td>GREEN</td>
<td>F5</td>
</tr>
<tr>
<td>DS68</td>
<td>F5_LED1</td>
<td>GREEN</td>
<td>F5</td>
</tr>
<tr>
<td>DS69</td>
<td>F5_LED2</td>
<td>GREEN</td>
<td>F5</td>
</tr>
<tr>
<td>DS70</td>
<td>F5_LED3</td>
<td>GREEN</td>
<td>F5</td>
</tr>
<tr>
<td>DS71</td>
<td>F6_LED0</td>
<td>GREEN</td>
<td>F6</td>
</tr>
<tr>
<td>DS72</td>
<td>F6_LED1</td>
<td>GREEN</td>
<td>F6</td>
</tr>
<tr>
<td>DS73</td>
<td>F6_LED2</td>
<td>GREEN</td>
<td>F6</td>
</tr>
<tr>
<td>DS74</td>
<td>F6_LED3</td>
<td>GREEN</td>
<td>F6</td>
</tr>
<tr>
<td>DS76</td>
<td>F7_LED0</td>
<td>GREEN</td>
<td>F7</td>
</tr>
<tr>
<td>DS77</td>
<td>F7_LED1</td>
<td>GREEN</td>
<td>F7</td>
</tr>
<tr>
<td>DS78</td>
<td>F7_LED2</td>
<td>GREEN</td>
<td>F7</td>
</tr>
<tr>
<td>DS79</td>
<td>F7_LED3</td>
<td>GREEN</td>
<td>F7</td>
</tr>
<tr>
<td>DS88</td>
<td>F8_LED0</td>
<td>GREEN</td>
<td>F8</td>
</tr>
<tr>
<td>DS89</td>
<td>F8_LED1</td>
<td>GREEN</td>
<td>F8</td>
</tr>
<tr>
<td>DS90</td>
<td>F8_LED2</td>
<td>GREEN</td>
<td>F8</td>
</tr>
<tr>
<td>DS91</td>
<td>F8_LED3</td>
<td>GREEN</td>
<td>F8</td>
</tr>
<tr>
<td>DS92</td>
<td>F9_LED0</td>
<td>GREEN</td>
<td>F9</td>
</tr>
<tr>
<td>DS93</td>
<td>F9_LED1</td>
<td>GREEN</td>
<td>F9</td>
</tr>
<tr>
<td>DS94</td>
<td>F9_LED2</td>
<td>GREEN</td>
<td>F9</td>
</tr>
<tr>
<td>LED ID</td>
<td>Description</td>
<td>Color</td>
<td>Notes</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>DS95</td>
<td>F9_LED3</td>
<td>GREEN</td>
<td>User controlled LED from FPGA F9</td>
</tr>
<tr>
<td>DS108</td>
<td>FPGA_DONE_Q_DK</td>
<td>GREEN</td>
<td>The Configuration FPGA is configured</td>
</tr>
<tr>
<td>DS47</td>
<td>FX0_QSFPO_FAULT</td>
<td>RED</td>
<td>“TXFAULT” Output by SFP module (J237). See SFF specification</td>
</tr>
<tr>
<td>DS48</td>
<td>FX0_QSFPO_LOS</td>
<td>RED</td>
<td>“LOS” Output by SFP module (J237). See SFF specification</td>
</tr>
<tr>
<td>DS54</td>
<td>FX0_QSFPO1_FAULT</td>
<td>RED</td>
<td>“TXFAULT” Output by SFP module (J236). See SFF specification</td>
</tr>
<tr>
<td>DS55</td>
<td>FX0_QSFPO1_LOS</td>
<td>RED</td>
<td>“LOS” Output by SFP module (J236). See SFF specification</td>
</tr>
<tr>
<td>DS24</td>
<td>FX0_XFP0_QXFP1_LO</td>
<td>RED</td>
<td>“LOS” Output by XFP module (U405). See XFI specification</td>
</tr>
<tr>
<td>DS44</td>
<td>FX0_XFP0_QXFP1_LO</td>
<td>RED</td>
<td>“TXFAULT” Output by SFP module (J236). See SFF specification</td>
</tr>
<tr>
<td>DS135</td>
<td>FX1_QSFPO0_FAULT</td>
<td>RED</td>
<td>“LOS” Output by SFP module (J239). See SFF specification</td>
</tr>
<tr>
<td>DS136</td>
<td>FX1_QSFPO0_LO</td>
<td>RED</td>
<td>“LOS” Output by SFP module (J239). See SFF specification</td>
</tr>
<tr>
<td>DS131</td>
<td>FX1_QSFPO1_FAULT</td>
<td>RED</td>
<td>“TXFAULT” Output by SFP module (J238). See SFF specification</td>
</tr>
<tr>
<td>DS132</td>
<td>FX1_QSFPO1_LO</td>
<td>RED</td>
<td>“LOS” Output by SFP module (J238). See SFF specification</td>
</tr>
<tr>
<td>DS138</td>
<td>FX1_XFP0_QXFP1_LO</td>
<td>RED</td>
<td>“LOS” Output by XFP module (U409). See XFI specification</td>
</tr>
<tr>
<td>DS145</td>
<td>FX1_XFP0_QXFP1_LO</td>
<td>RED</td>
<td>“LOS” Output by XFP module (U408). See XFI specification</td>
</tr>
<tr>
<td>DS18</td>
<td>QCFG_SLED0</td>
<td>GREEN</td>
<td>Smart Media Card is being read</td>
</tr>
<tr>
<td>DS19</td>
<td>QCFG_SLED1</td>
<td>GREEN</td>
<td>USB is in use</td>
</tr>
<tr>
<td>DS20</td>
<td>QCFG_SLED2</td>
<td>GREEN</td>
<td>SelectMap bus is in use (FPGAs are configuring)</td>
</tr>
<tr>
<td>DS21</td>
<td>QCFG_SLED3</td>
<td>GREEN</td>
<td>CompactFlash is being read</td>
</tr>
<tr>
<td>DS2</td>
<td>Q_CLED0</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS3</td>
<td>Q_CLED1</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS4</td>
<td>Q_CLED2</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS5</td>
<td>Q_CLED3</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS6</td>
<td>Q_CLED4</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS7</td>
<td>Q_CLED5</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS8</td>
<td>Q_CLED6</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS9</td>
<td>Q_CLED7</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS10</td>
<td>Q_CLED8</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS11</td>
<td>Q_CLED9</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS12</td>
<td>Q_CLED10</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS13</td>
<td>Q_CLED11</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS14</td>
<td>Q_CLED12</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS15</td>
<td>Q_CLED13</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS16</td>
<td>Q_CLED14</td>
<td>GREEN</td>
<td></td>
</tr>
<tr>
<td>DS17</td>
<td>Q_CLED15</td>
<td>RED</td>
<td>Blinks when the board is in reset</td>
</tr>
<tr>
<td>DS46</td>
<td>Q_F0_DONE</td>
<td>GREEN</td>
<td>FPGA F0 is configured</td>
</tr>
<tr>
<td>DS52</td>
<td>Q_F1_DONE</td>
<td>GREEN</td>
<td>FPGA F1 is configured</td>
</tr>
<tr>
<td>DS53</td>
<td>Q_F2_DONE</td>
<td>GREEN</td>
<td>FPGA F10 is configured</td>
</tr>
<tr>
<td>DS51</td>
<td>Q_F3_DONE</td>
<td>GREEN</td>
<td>FPGA F11 is configured</td>
</tr>
</tbody>
</table>
Each FPGA has 4 green user LEDs. Above shows FPGA F3 and its three green user LEDs, F3_LED0, F3_LED1, F3_LED2 and F3_LED3.

<table>
<thead>
<tr>
<th>Signal</th>
<th>FPGA NAME</th>
<th>F0</th>
<th>F1,F2,F3,F8,</th>
<th>F4,F5,F6,F7,</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>(&quot;FX0&quot;)</td>
<td>F9,F10,F11</td>
<td>F13,F14,F15</td>
<td>(&quot;FX1&quot;)</td>
<td></td>
</tr>
<tr>
<td>LED[0]</td>
<td>G13</td>
<td>N24</td>
<td>AE22</td>
<td>AK17</td>
<td></td>
</tr>
</tbody>
</table>

DN8000K10 User Guide  www.dinigroup.com  133
HARDWARE

<table>
<thead>
<tr>
<th>LED[1]</th>
<th>F13</th>
<th>T20</th>
<th>AD21</th>
<th>AK18</th>
</tr>
</thead>
</table>

There are also green LEDs to indicate that each power rail is present.

10 MGT Serial Resources

10.1 RocketIO

The FX parts are used to implement multiple channels of high-speed serial I/O. The FX60 or FX100 FPGAs used on the DN8000K10 provide either 16 (FX60) or 20 (FX100) Multi-Gigabit Transceiver (MGT) channels on two corners of the board. The DN8000K10 allows the use of Xilinx new 11Gbs transceivers. High-speed I/O connections provided include the following:

- **XFP socket (4)** – used for high-speed (9.5-11 Gbps) optical modules
- **SFP socket (4)** – used for medium-speed (1-4Gbs) optical modules
- **SMA (4 channels)** – RF frequency connectors with bandwidth beyond 20Ghz. Each channel provides up to 10Gbs in both directions.
- **Daughter card connectors (2)** – the 300-pin daughter card connectors associated with each FX part provide four (FX60/100) MGT channels per connector. The daughter card connector is FCI the MegArray series 300-pin high-speed connector. These interfaces are expected to support medium- to high-speed serial I/O links.
- **Samtec QSE connectors (4)** – these small, differential connectors have off-the-shelf coaxial ribbon cables available (Samtec EQDP) capable of 10Gb operation. (See Xilinx publication RPT015)
### Table 7.1 RocketIO Tile Assignments for FPGA 0 (FX60/100)

<table>
<thead>
<tr>
<th>“Right” Column Tiles</th>
<th>Functional Assignment</th>
<th>“Left” Column Tiles</th>
<th>Functional Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>109</td>
<td>Daughter card channels 3 and 4</td>
<td>101</td>
<td>Samtec channels 5 and 6 (FX100 only)</td>
</tr>
<tr>
<td>110</td>
<td>XFP Modules (production parts only)</td>
<td>102</td>
<td>Samtec channels 3 and 4</td>
</tr>
<tr>
<td>112</td>
<td>Daughter card channels 1 and 2</td>
<td>103</td>
<td>Samtec channels 1 and 2</td>
</tr>
<tr>
<td>113</td>
<td>SFP modules</td>
<td>105</td>
<td>Straight SMA connector pairs</td>
</tr>
<tr>
<td>114</td>
<td>Samtec channels 7 and 8 (FX100 only)</td>
<td>106</td>
<td>End-Launch SMA connector pairs</td>
</tr>
</tbody>
</table>

### Table XXX RocketIO Tile Assignments for FPGA 12 (FX60/100)

<table>
<thead>
<tr>
<th>“Right” Column Tiles</th>
<th>Functional Assignment</th>
<th>“Left” Column Tiles</th>
<th>Functional Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>109</td>
<td>SFP modules</td>
<td>101</td>
<td>Samtec channels 5 and 6 (FX100 only)</td>
</tr>
<tr>
<td>110</td>
<td>XFP Modules (production parts only)</td>
<td>102</td>
<td>End-Launch SMA connector pairs</td>
</tr>
<tr>
<td>112</td>
<td>Daughter card channels 3 and 4</td>
<td>103</td>
<td>Straight SMA connector pairs</td>
</tr>
<tr>
<td>113</td>
<td>Daughter card channels 1 and 2</td>
<td>105</td>
<td>Samtec channels 3 and 4</td>
</tr>
<tr>
<td>114</td>
<td>Samtec channels 7 and 8 (FX100 only)</td>
<td>106</td>
<td>Samtec channels 1 and 2</td>
</tr>
</tbody>
</table>

### 10.2 RocketIO signaling

![RocketIO signaling diagram](image-url)
Figure 6-7: Transmit Termination
10.3 RocketIO Clock Resources
AC-coupled.

10.3.1 Daughter card input

10.3.2 Synthesizers
Own power supply.
10.3.3 Samtec

10.3.4 Oscillators

Own power supply from 5.0 to 3.3. Epson EG-2102CA or Vectron VS-500 on top of each other

Since it is impossible to determine during manufacturing the clocking requirements of every possible end application, the DN8000K10 comes with a flexible clock network capable of a wide range of serial frequencies, while maintaining the tight jitter requirements of the 10 Gigabit serial transceivers.
The RocketIO clock tree for each Virtex 4 FX part is selectable via two differential clock multiplexers

is driven by a synthesizer and two oscillators, and dedicated multiplexers inside the Virtex 4 FPGA allow the user to switch between these clock sources.

To select a clock source in your design, use the CLOCK SOURCE command in the configuration file, main.txt on the Compact Flash card. The following example sets

```
//SET “FX1” (F12) clock sources
CLOCK SOURCE: FX1_0 SYNTH1
CLOCK FREQUENCY: FX1_0 300Mhz
```

All of the Source options for FX1_0 (F12) and FX0_0 (F0) are:

SYNTH0, SYNTH1, QSE0, QSE1

The available options for FX1_1 (F12) and FX0_0 (F0) are:
HARDWARE

DC, SYNTH1, QSE1
The MGTs on the Virtex 4 FPGA are divided into two columns, X0 (right) and X1 (left). The clock network of each column is separate and clocks may not be shared between the two columns. Each column has two FPGA internal clock distribution trees and two clock input pins. Either clock input can drive each tree. Finally, each tile has a multiplexer than can select from one of the two clock trees to clock that entire tile. Each tile contains two RocketIO transceivers, so each pair of channels must share a single transmit clock.

Once a clock is routed to an MGT tile, that clock can be multiplied and divided by the MGT tile.

Most users will want to use the frequency synthesizer for generating RocketIO reference clocks. The ICS843020-01 synthesizer is very low jitter and should suitable for operation up to 6Gbs RocketIO operation. The frequency of the synthesizer can be adjusted through the main.txt file on the SmartMedia card, or through the USB GUI program.
The LVPECL outputs of the ICS843020 are terminated through a resistor network to meet the input requirements of the MGTCLK inputs.

An output from the ICS843020-01 is also converted to LVDS and driven to J3 pins 19 and 21, the Samtec QSE-DP connector. This can be used to forward a RocketIO clock off board along with RocketIO signals to support standards that require an exact reference clock, like PCI Express. J3 may also drive pins 20 and 22. The ICS843020-01 can receive this clock and use it to generate a frequency for the MGTCLK inputs.

The ICS843020-01 Frequency Synthesizer is a very low phase noise. With the default 25Mhz oscillator, the frequency synthesizer is capable of producing frequencies in the ranges 71.875-84.375, 143.75-168.75, 287.5-337.5, and 575-675 MHz.

For 10Gb serial transmission rates, you should use one of the low-jitter fundamental frequency SAW oscillators. These oscillators operate at 250Mhz and so cover the gaps in the frequency synthesis options given by the ICS843020-01.
Each FPGA has two Epson 2101CA SAW oscillators connected directly to a MGT clock input on each column of the Virtex 4.

10.3.5 XFP REFCLK

XFP modules may require a low-jitter clock at a frequency 1/64 of the data rate. The only clock source on the DN8000K10 capable of meeting these requirements are the MGT outputs. You should use the same transmit clock as you are using for the XFP data MGT. Set the output data pattern such that it becomes a clock at 1/64 of the XFP bit rate. See Xilinx publication XAPP656.

J272, J273 – FX0

J274, J275 – FX1

LVPECL see schematic to change to LVDS.

10.4 MGT Power network

The RocketIO strict power supply constraints require the use of heavy power supply isolation. The RocketIO’s three power rails are each generated by a linear voltage regulator. The 1.2V
MGT analog and digital supply voltages and the 1.5V termination supply are isolated from the high-frequency digital noise produced by the 16 FPGAs in the DN8000K10 array section.

Each MGT power supply input pin is further protected from switching noise and supply current variation by a passive power filter network.

The termination power requirement of all 10 MGT tiles is supplied by a single 1.5V linear power supply, one per FX FPGA. This termination supply voltage can be changed. To be compatible with Virtex II Pro RocketIO in DC-coupled mode, this voltage must be changed to 1.8V. See the Virtex 4 RocketIO Users Guide. These input pins are named, VTTXB, VTTXA, VTRXB, and VTRXA.

The 1.2V analog and digital supply voltages are provided by three 1.2V linear regulators per FX FPGA. The MGT tiles on the FPGA were roughly split into three groups, with one 1.2V regulator supplying each. These input pins are named AVCCAUXRXB, AVCCAUXRXA, and AVCCAUXTX
The 2mA 2.5V requirement by the MGT tiles is met by a small reference voltage generator. These input pins are named AVCCAUXMGT.

The 1.2V and 1.5V linear regulators each have a small surface-mount heatsink installed on them.

In all other ways, the DN8000K10 follows all of the recommendations made by the Virtex 4 RocketIO user guide, UG076.

10.4.1 RTERM and MGTREF
These inputs can be used to change the default termination used in the Virtex 4 RocketIO drivers and receivers. The DN8000K10 implements these as suggested in the Virtex 4 Users Guide. The Xilinx software may disable this feature currently. Further documentation is not available for these as of this printing.

10.4.2 FX CES2 power supplies.
If your DN8000K10 came with a CES2 (engineering sample) FX part for FX0 and FX1 (F0 and F12), then a Virtex 4 erratum require the MGT analog 1.2V rail to be 1.1V. This setting may not be reflected by Appendix Schematics.

10.5 Connections
The following sections list the individual RocketIO connections. Here is a connection summary:

<table>
<thead>
<tr>
<th>RIGHT COLUMN</th>
<th>FPGA Pin</th>
<th>Signal Name</th>
<th>Connector</th>
<th>Conn. Pin</th>
<th>Signal Name</th>
<th>Connector</th>
<th>Conn. Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>MGT LOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GT11_X1Y9</td>
<td>A15</td>
<td>QSE07_TXP</td>
<td>SAMTEC QSE13</td>
<td>3</td>
<td>QSE07_TXP</td>
<td>SAMTEC QSE</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>A14</td>
<td>QSE07_TXN</td>
<td>CABLE</td>
<td>1</td>
<td>QSE07_TXN</td>
<td>CABLE</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>A18</td>
<td>QSE07_RXP</td>
<td>(J234)</td>
<td>2</td>
<td>QSE07_RXP</td>
<td>(J235)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>A17</td>
<td>QSE07_RXN</td>
<td></td>
<td>4</td>
<td>QSE07_RXN</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>(114)</td>
<td>A13</td>
<td>QSE08_TXP</td>
<td></td>
<td>9</td>
<td>QSE08_TXP</td>
<td></td>
<td>39</td>
</tr>
<tr>
<td>GT11_X1Y8</td>
<td>A12</td>
<td>QSE08_TXN</td>
<td></td>
<td>7</td>
<td>QSE08_TXN</td>
<td></td>
<td>37</td>
</tr>
</tbody>
</table>
## HARDWARE

<table>
<thead>
<tr>
<th></th>
<th>A10</th>
<th>QSE08_RXP</th>
<th>8</th>
<th>QSE08_RXP</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A9</td>
<td>QSE08_RXN</td>
<td>10</td>
<td>QSE08_RXN</td>
<td>40</td>
</tr>
<tr>
<td>GT11_X1Y7</td>
<td>A4</td>
<td>SFP0_TXP</td>
<td>SFP Connector</td>
<td>DC_CH1_TXP DC3</td>
<td>E9</td>
</tr>
<tr>
<td>(113)</td>
<td>A3</td>
<td>SFP0_TXN</td>
<td>(J237)</td>
<td>DC_CH1_TXN (P103)</td>
<td>F9</td>
</tr>
<tr>
<td></td>
<td>A7</td>
<td>SFP0_RXP</td>
<td></td>
<td>DC_CH1_RXP</td>
<td>E7</td>
</tr>
<tr>
<td></td>
<td>A6</td>
<td>SFP0_RXN</td>
<td></td>
<td>DC_CH1_RXN</td>
<td>F7</td>
</tr>
<tr>
<td>GT11_X1Y6</td>
<td>C1</td>
<td>SFP1_TXP</td>
<td></td>
<td>DC_CH2_TXP</td>
<td>E11</td>
</tr>
<tr>
<td></td>
<td>D1</td>
<td>SFP1_TXN</td>
<td></td>
<td>DC_CH2_TXN</td>
<td>F11</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>SFP1_RXP</td>
<td></td>
<td>DC_CH2_RXP</td>
<td>E13</td>
</tr>
<tr>
<td></td>
<td>G1</td>
<td>SFP1_RXN</td>
<td></td>
<td>DC_CH2_RXN</td>
<td>F13</td>
</tr>
<tr>
<td>GT11_X1Y5</td>
<td>R1</td>
<td>DC_CH1_TXP</td>
<td>DC0</td>
<td>E9</td>
<td>DC_CH3_TXP DC3</td>
</tr>
<tr>
<td>(112)</td>
<td>T1</td>
<td>DC_CH1_TXN</td>
<td>(P100)</td>
<td>F9</td>
<td>DC_CH3_TXN (P103)</td>
</tr>
<tr>
<td></td>
<td>M1</td>
<td>DC_CH1_RXP</td>
<td></td>
<td>E7</td>
<td>DC_CH3_RXP</td>
</tr>
<tr>
<td></td>
<td>N1</td>
<td>DC_CH1_RXN</td>
<td></td>
<td>F7</td>
<td>DC_CH3_RXN</td>
</tr>
<tr>
<td>GT11_X1Y4</td>
<td>U1</td>
<td>DC_CH2_TXP</td>
<td></td>
<td>E11</td>
<td>DC_CH4_TXP</td>
</tr>
<tr>
<td></td>
<td>V1</td>
<td>DC_CH2_TXN</td>
<td></td>
<td>F11</td>
<td>DC_CH4_TXN</td>
</tr>
<tr>
<td></td>
<td>Y1</td>
<td>DC_CH2_RXP</td>
<td></td>
<td>E13</td>
<td>DC_CH4_RXP</td>
</tr>
<tr>
<td></td>
<td>AA1</td>
<td>DC_CH2_RXN</td>
<td></td>
<td>F13</td>
<td>DC_CH4_RXN</td>
</tr>
<tr>
<td>GT11_X1Y2</td>
<td>AH1</td>
<td>XFP0_TXP</td>
<td>XFP Connector</td>
<td>XFP0_TXP</td>
<td>XFP1_TXP</td>
</tr>
<tr>
<td>(110)</td>
<td>AJ1</td>
<td>XFP0_TXN</td>
<td>(U405)</td>
<td>XFP0_TXN</td>
<td>(U112)</td>
</tr>
<tr>
<td></td>
<td>AL1</td>
<td>XFP0_RXP</td>
<td></td>
<td>XFP0_RXP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AM1</td>
<td>XFP0_RXN</td>
<td></td>
<td>XFP0_RXN</td>
<td></td>
</tr>
<tr>
<td>GT11_X1Y3</td>
<td>AF1</td>
<td>XFP1_TXP</td>
<td></td>
<td>XFP1_TXP</td>
<td></td>
</tr>
<tr>
<td>THIS TILE IS NON-FUNCTION ON CES PARTS</td>
<td>AG1</td>
<td>XFP1_TXN</td>
<td></td>
<td>XFP1_TXN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AC1</td>
<td>XFP1_RXP</td>
<td></td>
<td>XFP1_RXP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AD1</td>
<td>XFP1_RXN</td>
<td></td>
<td>XFP1_RXN</td>
<td></td>
</tr>
<tr>
<td>GT11_X1Y1</td>
<td>AP9</td>
<td>DC_CH3_TXP</td>
<td>DC0</td>
<td>E17</td>
<td>SFP0_TXP</td>
</tr>
<tr>
<td>(109)</td>
<td>AP10</td>
<td>DC_CH3_TXN</td>
<td>(P100)</td>
<td>F17</td>
<td>SFP0_TXN</td>
</tr>
<tr>
<td></td>
<td>AP6</td>
<td>DC_CH3_RXP</td>
<td></td>
<td>E15</td>
<td>SFP0_RXP</td>
</tr>
<tr>
<td></td>
<td>AP7</td>
<td>DC_CH3_RXN</td>
<td></td>
<td>F15</td>
<td>SFP0_RXN</td>
</tr>
<tr>
<td>GT11_X1Y0</td>
<td>AP11</td>
<td>DC_CH4_TXP</td>
<td></td>
<td>E19</td>
<td>SFP1_TXP</td>
</tr>
<tr>
<td></td>
<td>AP12</td>
<td>DC_CH4_TXN</td>
<td></td>
<td>F19</td>
<td>SFP1_TXN</td>
</tr>
<tr>
<td></td>
<td>AP14</td>
<td>DC_CH4_RXP</td>
<td></td>
<td>E21</td>
<td>SFP1_RXP</td>
</tr>
<tr>
<td></td>
<td>AP15</td>
<td>DC_CH4_RXN</td>
<td></td>
<td>F21</td>
<td>SFP1_RXN</td>
</tr>
</tbody>
</table>

### RIGHT COLUMN CLOCKS

<p>|        | J1   | OSC0 (U379) | EG-2101CA | 250Mhz | OSC0 (U379) | EG-2101CA | 250Mhz |
|        | K1   | 250Mhz      |          |        | 250Mhz      |          |        |
|        | AP3  | SYNTH1 (U395) | 25.0Mhz | SYNT1 | 25.0Mhz |          |
|        | AP4  | (&quot;FX0_1&quot;)  |          |        | (&quot;FX1_1&quot;) |          |        |
|        |      | DC0 (P100) | A29,B30 | DC3 (P103) | A29,B30 |        |</p>
<table>
<thead>
<tr>
<th>MGT LOC</th>
<th>QSE1 (J234)</th>
<th>QSE1 (J235)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT11_X0Y9 (101)</td>
<td>A23 QSE05_TXP Samtec QSE1 3</td>
<td>QSE05_TXP 3</td>
</tr>
<tr>
<td></td>
<td>A24 QSE05_TXN Connector 1</td>
<td>QSE05_TXN 1</td>
</tr>
<tr>
<td></td>
<td>A20 QSE05_RXP (J234) 2</td>
<td>QSE05_RXP 2</td>
</tr>
<tr>
<td></td>
<td>A21 QSE05_RXN 4</td>
<td>QSE05_RXN 4</td>
</tr>
<tr>
<td>GT11_X0Y8</td>
<td>A25 QSE06_TXP 9</td>
<td>QSE06_TXP 9</td>
</tr>
<tr>
<td></td>
<td>A26 QSE06_TXN 7</td>
<td>QSE06_TXN 7</td>
</tr>
<tr>
<td></td>
<td>A28 QSE06_RXP 8</td>
<td>QSE06_RXP 8</td>
</tr>
<tr>
<td></td>
<td>A29 QSE06_RXN 10</td>
<td>QSE06_RXN 10</td>
</tr>
<tr>
<td>GT11_X0Y7 (102)</td>
<td>D34 QSE03_TXP Samtec QSE0 33</td>
<td>J266 SMA RF</td>
</tr>
<tr>
<td></td>
<td>E34 QSE03_TXN Connector 31</td>
<td>J267 Connector</td>
</tr>
<tr>
<td></td>
<td>A31 QSE03_RXP (J232) 32</td>
<td>J264 Right-Angle</td>
</tr>
<tr>
<td></td>
<td>A32 QSE03_RXN 34</td>
<td>J265</td>
</tr>
<tr>
<td>GT11_X0Y6</td>
<td>F34 QSE04_TXP 39</td>
<td>J268</td>
</tr>
<tr>
<td></td>
<td>G34 QSE04_TXN 37</td>
<td>J269</td>
</tr>
<tr>
<td></td>
<td>J34 QSE04_RXP 38</td>
<td>J270</td>
</tr>
<tr>
<td></td>
<td>K34 QSE04_RXN 40</td>
<td>J271</td>
</tr>
<tr>
<td>GT11_X0Y5 (103)</td>
<td>V34 QSE01_TXP Samtec QSE0 3</td>
<td>J258 SMA RF</td>
</tr>
<tr>
<td></td>
<td>W34 QSE01_TXN Connector 1</td>
<td>J259 Connector</td>
</tr>
<tr>
<td></td>
<td>R34 QSE01_RXP (J232) 2</td>
<td>J256 Straight</td>
</tr>
<tr>
<td></td>
<td>T34 QSE01_RXN 4</td>
<td>J257</td>
</tr>
<tr>
<td>GT11_X0Y4</td>
<td>Y34 QSE02_TXP 9</td>
<td>J260</td>
</tr>
<tr>
<td></td>
<td>AA34 QSE02_TXN 7</td>
<td>J261</td>
</tr>
<tr>
<td></td>
<td>AC34 QSE02_RXP 8</td>
<td>J262</td>
</tr>
<tr>
<td></td>
<td>AD34 QSE02_RXN 10</td>
<td>J263</td>
</tr>
<tr>
<td>GT11_X0Y3 (105)</td>
<td>AJ34 J242 SMA RF QSE03_TXP Samtec QSE 33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AK34 J243 Connectors QSE03_TXN Connector 31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AF34 J240 Straight QSE03_RXP 32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AG34 J241 QSE03_RXN 34</td>
<td></td>
</tr>
<tr>
<td>GT11_X0Y2</td>
<td>AL34 J244 QSE04_TXP 39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AM34 J245 QSE04_TXN 37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AP32 J246 QSE04_RXP 38</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AP31 J247 QSE04_RXN 40</td>
<td></td>
</tr>
<tr>
<td>GT11_X0Y1 (106)</td>
<td>AP23 J250 SMA RF QSE01_TXP Samtec QSE 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AP22 J251 Connectors QSE01_TXN Connector 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AP26 J248 Right-Angle QSE01_RXP 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AP25 J249 QSE01_RXN 4</td>
<td></td>
</tr>
</tbody>
</table>
### Hardware

| GT11_X0Y0 | AP21 | J252 | QSE02_TXP | 9   |
| AP20      | J253 |     | QSE02_TXN | 7   |
| AP18      | J254 |     | QSE02_RXP | 8   |
| AP17      | J255 |     | QSE02_RXN | 10  |

#### Left Column Clocks

| (GT11CLK_X0Y3) | M34 | N34 | OSC1 (U382) | 250Mhz | OSC3 (U388) | 250Mhz |
| (GT11CLK_X0Y1) | AP29 | AP28 | SYNTH0 ICS84321 25.5Mhz | SYNTH0 ICS84321 25.5Mhz |
| (102)          |     |     | ("FX0_0") | ("FX1_0") |
| (105)          |     |     | QSE0 (J232) Samtec QSE 20,22 | QSE0 (J233) Samtec QSE 20,22 |
|                |     |     | SYNTH1 ICS84321 25.0Mhz | SYNTH1 ICS84321 25.0Mhz |
|                |     |     | ("FX0_1") | ("FX1_1") |
|                |     |     | QSE1 (J234) Samtec QSE 20,22 | QSE1 (J235) Samtec QSE 20,22 |

#### 10.5.1 Samtec Multi Gigabit Cable Connector

For board-to-board high-density connections, two Samtec ribbon cable connectors per FX part are connected to RocketIO. The pin outs on the cable allow two DN8000K10 boards to be connected to each other for 4 bi-directional channels operating at 5Gbs or more per channel, per direction.

The Samtec part number for the connector installed on the host is QSE-014-01-F-D-DP-A. An appropriate crossover cable for cabling two DN8000K10s together is the Samtec EQDP-014-09.00-TBR-TBL-4. The appropriate mating parts are from the Samtec QTE-DP, or DP EQCD HFEM-DP series.
The pin out is arranged such that a crossover cable (pin 1 to 40) can be used to connect two DN8000K10s together. Note that the grounded pins 5, 11, 17, 23, 29, 35, 6, 12, 18, 24, 30, 36 are NC (no pin present) on the connector. These are grounded in the DN8000K10 for reverse-compatibility. The pins 41, 42, 43, 44 are ground blades built into the connector. Pins 45 and 46 are non-plated plastic alignment pins.

The Samtec cable EQDP-014-09.00-TBR-TBL-4 (9 inch version) is capable of 10Gbs operation for lengths of up to 1 meter according to the Samtec Appnote.
HARDWARE

See Appendix Pins Other for pin out information

Each connector also has a clock input that can be routed to an MGT CLK input of it’s FX FPGA to allow cabling standards that require transmitting at an exact frequency, such as PCI Express. See Hardware: MGT Serial Resources: Clocks: QSE.

10.5.2 Optical Modules

The DN8000K10 comes with eight optical module connectors. If you need to interface to a specific physical standard, the easiest way is to buy an SFP or XFP module that supports that standard.

The optical module interfaces contain a high-speed connector and a metal EMI and mechanical cage. Each connector has one transmit and one receive differential signal (4 wires), and some low-speed support signals.

10.5.3 SFP

SFP modules are available supporting 1-4.5Gbs serial transmission rates.

Two red LEDs show the status of the channel. The RXLOS LED indicates the loss-of-signal in the SFF specification. The TXFAULT LED indicates a transmission laser failure, or an unsecured module.

For the connectivity to the FPGA, see the Appendix Pins Other: RocketIO.

The support signals in the SFF interface are TXFAULT, TXDIS, MOD-DEF2, MOD-DEF1, MOD-DEF0, RATESEL, and LOS. The MOD-DEF1 and MOD-DEF2 signals usually make up a 2-wire interface that can be used to communicate with logic imbedded on the module itself. These support signals do not connect directly to an FPGA, instead they connect to an IO expansion CPLD, which connects to the FPGA through a four-wire interface.

Most SFP modules can be operated without having to use any of the low-speed SFF signals.

10.5.4 SFP Expansion CPLD
The SFF signals from the SFP modules are connected to the IOs of an IO Expansion CPLD. The interface for interacting with the IO expansion CPLD is as follows:

FXx_CPLDFXx_SCLK_i
FXx_CPLDFXx_SDATA_io
FXx_CPLDFXx_RD_WRn_i
FXx_CPLDFXx_CS_n_i

4data + 5 addr

SCLK should be a free-running clock no faster than 1Mhz. To begin a read or write hold CS low and RD_WRn low synchronously with SCLK. Serially transmit a 5-bit sequence to the CPLD using the SDATA pin. The sequences are used for the CPLD to determine which optical module to access. The 5-bit sequence can be one of the following (binary):
HARDWARE

S_RD_SFP0 10011  Read from SFP0
S_RD_SFP1 10100  Read from SFP1
S_RD_XFP0 10101  Read from XFP0
S_RD_XFP1 10110  Read from XFP1
S_WR_SFP0 00111  Write to SFP0
S_WR_SFP1 01000  Write to SFP1
S_WR_XFP0 01001  Write to XFP0
S_WR_XFP1 01010  Write to XFP1

After clock 5, if you selected a Read command, leave RD_WRn low. If you selected a Write, assert RD_WRn high after the 5th clock cycle.

During the next 4 clock cycles, if a Read command was sent, capture the SDATA signal bits during those cycles and interpret them as follows:

1: TX_FAULT (SFP), INTn (XFP)
2: LOS (SFP), MOD_ABS (XFP)
3: MOD_SEL0 (SFP), MOD_NR (XFP)
4: MOD_SEL2 (SFP), MOD_RXLOS (XFP)

If the sent command was a Write command, during those four cycles, transmit

1: MOD_SEL1 (SFP), MOD_DES1 (XFP)
2: MOD_SEL2 (SFP), TXDIS (XFP)
3: MOD_SEL2_outputenable (SFP), PDOWN(XFP)
4: RATESEL (SFP)
5: TXDIS (SFP)

The source code for the expansion CPLD is included on the User CD. An expansion CPLD controller is included in the /Source Code/ directory of the User CD. For the signal description of the SFF or XFI interfaces, see the SFF and XFI specifications.

<table>
<thead>
<tr>
<th>CPLD registers</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>XFP0_MOD_DESEL</td>
<td>0x001</td>
</tr>
<tr>
<td>XFP0_INTn</td>
<td>0x002</td>
</tr>
<tr>
<td>XFP0_TXDIS</td>
<td>0x003</td>
</tr>
<tr>
<td>XFP0_MOD_ABS</td>
<td>0x004</td>
</tr>
<tr>
<td>XFP0_MOD_NR</td>
<td>0x005</td>
</tr>
<tr>
<td>XFP0_RX_LOS</td>
<td>0x006</td>
</tr>
<tr>
<td>XFP0_PDOWN</td>
<td>0x007</td>
</tr>
<tr>
<td>SFP0_TXFAULT</td>
<td>0x001</td>
</tr>
<tr>
<td>SFP0_TXDISABLE</td>
<td>0x002</td>
</tr>
<tr>
<td>SFP0_MOD_DEF</td>
<td>0x003</td>
</tr>
<tr>
<td>SFP0_MOD_DEF</td>
<td>0x004</td>
</tr>
</tbody>
</table>
### XFP

XFP modules are the fastest available removable serial modules that are protocol independent. The XFP modules provide externally a XFI signaling interface. This interface only operates between the rates of 9.5-10.3 Gbps. It may be possible for a module to operate at lower speeds.

Note that in order to obtain speeds compatible with the XFI specification, Virtex 4 FX production parts (non-CES) are required in the –12 speed grade.

Above is a simplified schematic capture of the XFP connector. The high speed serial IOs of the XFP connector connect directly to the MGT IOs on the Virtex 4 FPGA.

Some XFP modules may require a reference clock for retiming the transmitted signal and jitter reduction (See signal REFCLK in the XFI specification). The REFCLK signal on the DN8000K10 is sourced from one pair of SMAs for each Virtex 4 FX FPGA. The REFCLK signal should be 1/64 of the data rate driven onto the XFP’s TX pins. You can generate this clock using one of the SMA outputs of the Virtex 4 FX MGT. See Xilinx Application note XAPP656. More description of this circuit is in Hardware: MGT Serial Resources: RocketIO clock: XFP REFCLK.

The XFP specification allows for modules to require an optional –5.2V power supply to be provided by the host board. The DN8000K10 provides no –5.2V power, so a mounting point (U1) is provided for the use of a bench supply if ECL signaling is required.
Most XFP modules require 1.8V power from the host. The 1.8V Voltage provided by the DN8000K10 come from the 1.8V_0 and 1.8V_1 power rails, shared by the SODIMM module sockets.

Power supply filtering for each XFP module is provided, following the recommendations in the XFP specification.

The XFI specification defines some low-speed signals for monitoring status. These signals, like the SFP low-speed signals, with the exception of the SDATA and SCLK signals, are connected to an IO expansion CPLD. These signals are 3.3V, open-drain signals with external pull-up resistors. The SDATA and SCLK signals connect directly to the associated FX FPGA. The two XFP optical modules’ SDATA and SCLK signals are bussed, so in order to use the serial status interface of two XFP modules simultaneously, the CPLD interface must be implemented for control of the MOD_DESEL signal.

**10.5.6 XFP IIC**

There is an IIC bus on the XFI signal interface for each XFP module. The IIC signals are connected to the FPGA and are bussed between the two modules. In order to access this bus, the XFP CPLD IO expansion must be used to cable MOD_SEL on the module you with to communicate with.

- **XFP IIC signals**
  - (Bussed between both XFP modules)
  - XFP_SCL
  - XFP_SDA

**10.5.7 XFP expansion CLPD**

The XFI interface includes some low-speed signals. None of these signals are required to use the XFP modules, however they are made available to the user FPGA through the IOs of an IO expansion CPLD. The interface used for this IO is the same as described in the section above. See Hardware: MGT Serial Resources: SFP: Expansion CPLD for an interface description.
All of the source code required to perform this IO expansion is provided on the user CD.

**10.5.8 The daughter card**

On each FX FPGA, 4 MGT channels connect to a 300-pin FCI Meg Array connector. These connectors are capable of data rates up to 10Gbs. The meg-array connector itself is a BGA grid of conductors, providing controlled-impedance connections with very low cross talk. The pin out of the MegArray connectors on the DN8000K10 is in a GSG pattern, with extra ground pins surrounding the signals to bring pair-to-pair cross talk to less than 1%.

More information about the daughter card connector is in the section Hardware: Daughter card Interface.

The two daughter card connectors with RocketIO signals DC0 and DC3 differ from the other 300-pin MegArray connectors because they have no general-purpose IO
Nothing is connected to DC0 and DC3 except for MGT signals, power, VCCO0 and VCCO1 (unused), MGTCLK, GCA, GCB, GCC.

Descriptions of these signals are found in *Hardware: Daughter cards*
### 10.5.9 The SMAs

SMA RF connectors are the most robust connector on the DN8000K10 for very-high data rates. For MGT channels connected to the SMA interface, there is one SMA connector for each of Transmit Negative, Transmit Positive, Receive Negative, and Receive Positive.

You must use matched cables for a P/N pair. Signals are routed on the host as 50Ohm, loosely coupled differential signals.

The vertical SMA connectors.

![Image of vertical SMA connectors](image)

The right-angle SMA connectors extend beyond the back panel of the carrier assembly.
SMA connectors have a bandwidth of 16-24Ghz and are suitable for Virtex 4 RocketIO signals up to 10Gbs.

11 FPGA System monitor/ADC

The System Monitor and ADC functions of the Virtex 4 FPGA are no longer supported by Xilinx. One important function of the System Monitor, temperature sensing, has been added to the configuration circuitry. The DN8000K10 will automatically monitor and prevent thermal overload in the sixteen Virtex 4 FPGA array. No user action is required. A Maxim MAX1617A temperature monitor uses a current sensing voltage source connected to the TDN and TDP pins of the Virtex 4 FPGA to measure changes in internal temperature. The MAX1617A has an IIC interface to the Microcontroller. The microcontroller polls each of the sixteen temperature monitors about once a second. If any of the temperature sensors measure beyond a user-specified temperature, the microcontroller causes a board reset.

The power supply pins of the system monitor are connected. As recommended by the Virtex 4 User Guide
12 Mechanical

12.1 Overview.
The dimensions of the PWB are 580mm long by 381mm tall.

There are 4 metal stiffener bars. These are connected to the GND net and are convenient for grounding oscilloscope probes and your hand static. User control connections are located on the right edge of the board, or the front of the chassis. High speed serial connectors are located on the left edge of the board, or the back of the chassis. Grounded mounting holes are distributed.

12.2 Base Plate
The DN8000K10 is shipped on a steel base plate to provide protection, stability and an easy way to transport the board.

The reverse side of the base plate has cutaway holes to allow access to the daughter card headers and FPGA JTAG header from the bottom side. The base plate provides standoff holes for mounting daughter cards. For exact dimensions of these stand-offs see the section Hardware, Daughter cards.
Optionally, the DN8000K10 can ship in a 4U rack mount chassis assembly. See Section *Ordering information: Optional Equipment: Chassis.*

The DN8000K10 can also be operated outside of the carrier if desired.
13 Test points and Connectors

13.1 Test points
The following table lists all of the test points on the DN8000K10 and the corresponding net. See Appendix Schematic.

<table>
<thead>
<tr>
<th>Assembly Label</th>
<th>Label</th>
<th>Net name</th>
<th>Location</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>J278</td>
<td>LVEE5_XFP</td>
<td>70,370 top</td>
<td>-5.2V supply access point for FX0's XFP0 and XFP1 modules</td>
<td></td>
</tr>
<tr>
<td>J277</td>
<td>+1.8V_XFP</td>
<td>70,370 top</td>
<td>+1.8V supply access point for FX0's XFP0 and XFP1 modules. Shorted to +1.8V_0 through R3 (0 Ohm)</td>
<td></td>
</tr>
<tr>
<td>TP3</td>
<td>DIMM1 CK</td>
<td>DDR1_CK_TEST</td>
<td>270,380 top</td>
<td>Single-ended copy of CK0p and CK1p to socket J101</td>
</tr>
<tr>
<td>TP1</td>
<td>DIMM0 CK</td>
<td>DDR0_CK_TEST</td>
<td>270,380 top</td>
<td>Single-ended copy of CK0p and CK1p to socket J100</td>
</tr>
<tr>
<td>Assembly Label</td>
<td>Label</td>
<td>Net name</td>
<td>Location</td>
<td>Comment</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>----------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>TP13</td>
<td></td>
<td>+1.2V_4</td>
<td></td>
<td>VCCINT of FPGA F4</td>
</tr>
<tr>
<td>TP21</td>
<td></td>
<td>-1.2V_5</td>
<td></td>
<td>VCCINT of FPGA F5</td>
</tr>
<tr>
<td>TP12</td>
<td></td>
<td>+1.2V_7</td>
<td></td>
<td>VCCINT of FPGA F7</td>
</tr>
<tr>
<td>TP11</td>
<td></td>
<td>+1.2V_6</td>
<td></td>
<td>VCCINT of FPGA F6</td>
</tr>
<tr>
<td>TP2</td>
<td>DIMM_VTT0</td>
<td></td>
<td>330,380, top</td>
<td>Termination voltage for DIMMS J100, J101 (DIMM0, DIMM1).</td>
</tr>
<tr>
<td>TP7</td>
<td>GND</td>
<td>GND</td>
<td>400,340</td>
<td>Monolithic ground net.</td>
</tr>
<tr>
<td>TP17</td>
<td></td>
<td>REPCLKTEST</td>
<td></td>
<td>A single-ended copy of the REFCLK global clock network.</td>
</tr>
<tr>
<td>TP37</td>
<td></td>
<td>+5.0VSB</td>
<td></td>
<td>The EPS signal “SSB”. Unconnected on DN8000K10</td>
</tr>
<tr>
<td>TP43</td>
<td>GND</td>
<td>GND</td>
<td></td>
<td>Monolithic Ground</td>
</tr>
<tr>
<td>TP18</td>
<td>GND</td>
<td>GND</td>
<td></td>
<td>Monolithic Ground</td>
</tr>
<tr>
<td>TP45</td>
<td></td>
<td>+12.0V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP35</td>
<td></td>
<td>-5.0V</td>
<td></td>
<td>Unconnected on DN8000K10</td>
</tr>
<tr>
<td>TP32</td>
<td></td>
<td>-12.0V</td>
<td></td>
<td>Unconnected on DN8000K10</td>
</tr>
<tr>
<td>TP31</td>
<td></td>
<td>+3.3V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP33</td>
<td></td>
<td>+5.0V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP47</td>
<td>DDR2_CK_TEST</td>
<td></td>
<td></td>
<td>Copy of CK0p and CK1p signal to DIMM2 (J102) (F13)</td>
</tr>
<tr>
<td>TP48</td>
<td>DDR3_CK_TEST</td>
<td></td>
<td></td>
<td>Copy of CK0p and CK1p signal sent to DIMM3 (F14) (J103)</td>
</tr>
<tr>
<td>TP15</td>
<td>PH0_TEST_P</td>
<td></td>
<td></td>
<td>Copy of PH0 global clock network output</td>
</tr>
<tr>
<td>TP14</td>
<td>PH0_TEST_N</td>
<td></td>
<td></td>
<td>Copy of PH0 global clock network output</td>
</tr>
<tr>
<td>TP36</td>
<td>PH0CLKTEST</td>
<td></td>
<td></td>
<td>Single-ended copy of PH0 8442 clock synthesizer output (U310)</td>
</tr>
<tr>
<td>TP23</td>
<td>PH1_TEST_N</td>
<td></td>
<td></td>
<td>Copy of PH1 global clock network output. Use differentially with TP22</td>
</tr>
<tr>
<td>TP22</td>
<td>PH1_TEST_P</td>
<td></td>
<td></td>
<td>Copy of PH1 global clock network output. Use differentially with TP23.</td>
</tr>
<tr>
<td>TP40</td>
<td>PH1CLKTEST</td>
<td></td>
<td></td>
<td>Single-ended copy of PH1 8442 clock synthesizer output (U315)</td>
</tr>
<tr>
<td>TP44</td>
<td>PH2CLKTEST</td>
<td></td>
<td></td>
<td>Single-ended copy of PH2 8442 clock synthesizers output (U330)</td>
</tr>
<tr>
<td>TP26</td>
<td>PH2_TEST_N</td>
<td></td>
<td></td>
<td>Copy of PH2 global clock network output. Use differentially with TP27.</td>
</tr>
<tr>
<td>TP27</td>
<td>PH2_TEST_P</td>
<td></td>
<td></td>
<td>Copy of PH2 global clock network output. Use differentially with TP26.</td>
</tr>
<tr>
<td>TP8</td>
<td></td>
<td>+1.2V_0</td>
<td></td>
<td>VCCINT of FPGA F0</td>
</tr>
<tr>
<td>TP10</td>
<td></td>
<td>+1.2V_1</td>
<td></td>
<td>VCCINT of FPGA F1</td>
</tr>
<tr>
<td>FP9</td>
<td></td>
<td>+1.2V_2</td>
<td></td>
<td>VCCINT of FPGA F2</td>
</tr>
<tr>
<td>TP5</td>
<td></td>
<td>+1.2V_3</td>
<td></td>
<td>VCCINT of FPGA F3</td>
</tr>
</tbody>
</table>
### H A R D W A R E

| TP30 | +1.2V_9 | VCCINT of FPGA F9 |
| TP29 | +1.2V_8 | VCCINT of FPGA F8 |
| TP24 | +1.2V_11 | VCCINT of FPGA F9 |
| TP28 | +1.2V_10 | VCCINT of FPGA F10 |
| TP34 | +1.2V_13 | VCCINT of FPGA F13 |
| TP38 | +1.2V_12 | VCCINT of FPGA F12 |
| TP42 | +1.2V_15 | VCCINT of FPGA F15 |
| TP39 | +1.2V_14 | VCCINT of FPGA F14 |
| TP19 | +1.2V_16 | VCCINT of Configuration FPGA |
| TP20 | +2.1V | This net supplies current for RocketIO nets |
| TP16 | +2.5V_0 | This net supplies VCCO and AVCC AUX for FPGA F0, F1, F4, F5 |
| TP4 | +2.5V_1 | This net supplies VCCO and AVCC AUX for FPGA F2, F3, F6, F7 |
| TP25 | +2.5V_3 | This net supplies VCCO and AVCC AUX for FPGA F8, F9, F12, F13 |
| TP46 | +2.5V_2 | This net supplies VCCO and AVCC AUX for FPGA F10, F11, F14, F15 |
| TP49 | DIMM_VTT1 | Termination voltage for DIMM1 (J103) |
| TP41 | +1.8V_1 | Internal and IO voltage for DDR2 SODIMM in sockets DIMM0 and DIMM1 (J100, J101) |

Grounded test points are distributed to make grounding oscilloscope probes easier.

## GND Test Points

![GND Test Points Diagram]

### 13.2 Connectors

The following table lists all of the connectors on the DN8000K10. Also see the Schematics provided on the user CD.

<table>
<thead>
<tr>
<th>Assy Num</th>
<th>Purpose</th>
<th>Connector Part Number</th>
<th>Specification</th>
<th>Signaling</th>
<th>Top/ Bot</th>
</tr>
</thead>
<tbody>
<tr>
<td>J204</td>
<td>IDE, Use to connect remote</td>
<td>AMP 103310-8</td>
<td>ATA-4 <a href="http://www.t13.org/project/d1153r18-ATA-ATAPI-4.pdf">http://www.t13.org/project/d1153r18-ATA-ATAPI-4.pdf</a></td>
<td>5V TTL</td>
<td>T</td>
</tr>
<tr>
<td>J201</td>
<td>CompactFlash socket</td>
<td>Hirose Electronic Co. MI21-50PD-SF-EJR</td>
<td>Compact Flash 3 <a href="http://www.computflash.org/">http://www.computflash.org/</a></td>
<td>3V LVTTL</td>
<td>T</td>
</tr>
<tr>
<td>J203</td>
<td>USB</td>
<td>Molex 67068-1000</td>
<td>USB 2 <a href="http://www.usb.org/developers/docs/usb_20_02212005.zip">http://www.usb.org/developers/docs/usb_20_02212005.zip</a></td>
<td>5V differential</td>
<td>T</td>
</tr>
<tr>
<td>J206</td>
<td>Case Fan</td>
<td>Molex 22-05-3031</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open Drain Tachometer input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P204</td>
<td>MCU terminal</td>
<td>AMP 103310-10</td>
<td>Pin 2 TX (output), Pin 3 RX (input), Pin 5 GND</td>
<td>RS232 (12V)</td>
<td>T</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
<td>Access</td>
<td>Reference</td>
<td>Voltage</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>P206 User RS232_2</td>
<td>AMP 103310-10</td>
<td>Pin 2 TX, Pin 3 RX, Pin 5 GND</td>
<td>RS232 (12V)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J207 Case Fan</td>
<td>Molex 22-05-3031</td>
<td>Pin 2 12V, Pin 1 GND, Pin 3 Open drain Tachometer</td>
<td>12V</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>P200 EPS 24</td>
<td>Molex 39-29-1248</td>
<td><a href="http://www.ssiforum.org/Power%20Supplies/EPS12V_Spec%202_1.pdf">http://www.ssiforum.org/Power%20Supplies/EPS12V_Spec%202_1.pdf</a></td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P201 EPS 8</td>
<td>Molex 39-29-1088</td>
<td><a href="http://www.ssiforum.org/Power%20Supplies/EPS12V_Spec%202_1.pdf">http://www.ssiforum.org/Power%20Supplies/EPS12V_Spec%202_1.pdf</a></td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3 Hard Reset</td>
<td>Omron SW416</td>
<td>Causes the board including the configuration circuit to reset</td>
<td>3.3V Open drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>S2 Logic Reset</td>
<td>Omron SW416</td>
<td>Sends the “Logic Reset” signal to the 16 user FPGAs.</td>
<td>3.3V Open drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>BT1 Battery Socket</td>
<td>Keystone 31C0589</td>
<td>Use Type 364 watch battery</td>
<td>1.5V</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J200 FPGA JTAG</td>
<td>Molex 87832-1420</td>
<td>Compatible with Xilinx Parallel IV cable</td>
<td>2.5V Open Drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>S1 MCU switch</td>
<td>CTS 219-4MST</td>
<td>Compatible with Xilinx Parallel IV cable</td>
<td>3.3V</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J225 Configuration CPLD JTAG</td>
<td>Molex 87832-1420</td>
<td>Compatible with Xilinx Parallel IV cable</td>
<td>3.3V Open drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>U203 Boot code EPROM socket</td>
<td>Mill-Max 110-93-308-41-001</td>
<td>Compatible with 24LC64 EPROM and Cypress CY7C68013/TQFP128</td>
<td>3.3V Open drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>P203 Chassis Controls remote connector</td>
<td>CON10A TEST??</td>
<td>Pin 1 Logic reset, Pin 3 PS_ON, Pin 5 Hard reset, Pins 2,4,6 GND</td>
<td>3.3V Open drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>P206 User RS232 4</td>
<td>MOLEX_71349-1003</td>
<td>Pin 2 TX, Pin 3 RX, Pin 5 GND</td>
<td>RS232 (12V)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>P209 User RS232 5</td>
<td>MOLEX_71349-1003</td>
<td>Pin 2 TX, Pin 3 RX, Pin 5 GND</td>
<td>RS232 (12V)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>P207 User RS232 3</td>
<td>MOLEX_71349-1003</td>
<td>Pin 2 TX, Pin 3 RX, Pin 5 GND</td>
<td>RS232 (12V)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J205 Case Fan</td>
<td>Molex 22-05-3031</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td>2.5V Open drain</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J228 Extclk PH0p</td>
<td>Johnson Components 142-0701-201</td>
<td>Feeds global clock 0 (GCLK0)</td>
<td>LVDS (modify for LVPECL or CML)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J229 Extclk PH1n</td>
<td>Johnson Components 142-0701-201</td>
<td>Feeds global clock 1 (GCLK1)</td>
<td>LVDS (modify for LVPECL or CML)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J230 Extclk PH2p</td>
<td>Johnson Components 142-0701-201</td>
<td>Feeds global clock 2 (GCLK2)</td>
<td>LVDS (modify for LVPECL or CML)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J231 Extclk PH2n</td>
<td>Johnson Components 142-0701-201</td>
<td>Feeds global clock 2 (GCLK2)</td>
<td>LVDS (modify for LVPECL or CML)</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>U408 “FX1_XFP1”</td>
<td>AMP 1367500-1</td>
<td><a href="http://www.xfpmsa.org/XFP_SFF_INF_8074i_Rev4_0.pdf">http://www.xfpmsa.org/XFP_SFF_INF_8074i_Rev4_0.pdf</a></td>
<td>CML</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>U409 “FX1_XFP0”</td>
<td>AMP 1367500-1</td>
<td><a href="http://www.xfpmsa.org/XFP_SFF_INF_8074i_Rev4_0.pdf">http://www.xfpmsa.org/XFP_SFF_INF_8074i_Rev4_0.pdf</a></td>
<td>CML</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J239 “FX1_SFP0”</td>
<td>Molex 74441-0001</td>
<td>SFP specification INF-8074.PDF</td>
<td>CML</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>J238 “FX1_SFP1”</td>
<td>Molex 74441-0001</td>
<td>SFP specification INF-8074.PDF</td>
<td>CML</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Hardware Item</td>
<td>Description</td>
<td>Manufacturer</td>
<td>Part Number</td>
<td>Specification Link</td>
<td>Type</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
<td>---------------------</td>
<td>------</td>
</tr>
<tr>
<td>U405</td>
<td>“FX0_XFP0”</td>
<td>AMP</td>
<td>1367500-1</td>
<td><a href="http://www.xfpmsa.org/XFP_SFF_INF_8077i_Rev4_0.pdf">http://www.xfpmsa.org/XFP_SFF_INF_8077i_Rev4_0.pdf</a></td>
<td>CML</td>
</tr>
<tr>
<td>U404</td>
<td>“FX0_XFP1”</td>
<td>AMP</td>
<td>1367500-1</td>
<td><a href="http://www.xfpmsa.org/XFP_SFF_INF_8077i_Rev4_0.pdf">http://www.xfpmsa.org/XFP_SFF_INF_8077i_Rev4_0.pdf</a></td>
<td>CML</td>
</tr>
<tr>
<td>J237</td>
<td>“FX0_SFP0”</td>
<td>Molex</td>
<td>74441-0001</td>
<td>SFP specification INF-8074.PDF</td>
<td>CML</td>
</tr>
<tr>
<td>J236</td>
<td>“FX0_SFP1”</td>
<td>Molex</td>
<td>74441-0001</td>
<td>SFP specification INF-8074.PDF</td>
<td>CML</td>
</tr>
<tr>
<td>J272</td>
<td>“FX0” XFP RefClk</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>LVPECL</td>
</tr>
<tr>
<td>J247</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J246</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J245</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J244</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J243</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J242</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J241</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J240</td>
<td>FX0 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J262</td>
<td>FX1 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J263</td>
<td>FX1 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J261</td>
<td>FX1 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J258</td>
<td>FX1 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J259</td>
<td>FX1 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J256</td>
<td>FX1 SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>J257</td>
<td>FX1 VERT SMA</td>
<td>Johnson</td>
<td>142-0701-201</td>
<td><a href="http://www.xilinx.com/bvdocs/userguides/ug076.pdf">http://www.xilinx.com/bvdocs/userguides/ug076.pdf</a></td>
<td>CML (RocketIO)</td>
</tr>
<tr>
<td>Part Number</td>
<td>Description</td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J234</td>
<td>SAMTEC QSE</td>
<td>CML (RocketIO)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J221</td>
<td>SAMTEC QSE</td>
<td>CML (RocketIO)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J233</td>
<td>SAMTEC FX1</td>
<td>CML (RocketIO)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J235</td>
<td>SAMTEC FX1</td>
<td>CML (RocketIO)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J100</td>
<td>DIMM0 F1</td>
<td><a href="http://www.jedec.org/download/search/N03-NM0.pdf">www.jedec.org/download/search/N03-NM0.pdf</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J101</td>
<td>DIMM1 F2</td>
<td><a href="http://www.jedec.org/download/search/N03-NM0.pdf">www.jedec.org/download/search/N03-NM0.pdf</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J102</td>
<td>DIMM2 F3</td>
<td><a href="http://www.jedec.org/download/search/N03-NM0.pdf">www.jedec.org/download/search/N03-NM0.pdf</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J103</td>
<td>DIMM3 F4</td>
<td><a href="http://www.jedec.org/download/search/N03-NM0.pdf">www.jedec.org/download/search/N03-NM0.pdf</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J209</td>
<td>FAN HEADER 0</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J210</td>
<td>FAN HEADER 1</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J211</td>
<td>FAN HEADER 2</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J212</td>
<td>FAN HEADER 3</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J213</td>
<td>FAN HEADER 4</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J214</td>
<td>FAN HEADER 5</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J215</td>
<td>FAN HEADER 6</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J216</td>
<td>FAN HEADER 7</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J217</td>
<td>FAN HEADER 8</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J218</td>
<td>FAN HEADER 9</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J219</td>
<td>FAN HEADER 10</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J220</td>
<td>FAN HEADER 11</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J221</td>
<td>FAN HEADER 12</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J222</td>
<td>FAN HEADER 13</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J223</td>
<td>FAN HEADER 14</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J224</td>
<td>FAN HEADER 15</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J225</td>
<td>FAN HEADER 16</td>
<td>Pin 1 GND, Pin 2 12V, Pin 3 Open drain Tachometer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J226</td>
<td>FPGA JTAG</td>
<td>87332-1420</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P210</td>
<td>“DC10”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P109</td>
<td>“DC9”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P108</td>
<td>“DC8”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P107</td>
<td>“DC7”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P106</td>
<td>“DC6”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P105</td>
<td>“DC5”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P104</td>
<td>“DC4”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P103</td>
<td>“DC3”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P102</td>
<td>“DC2”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P101</td>
<td>“DC1”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P100</td>
<td>“DC0”</td>
<td>FCI 84578-102</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 5: The Reference Design

This chapter introduces the DN8000K10 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 Exploring the Reference Design

1.1 What is the Reference Design?

The reference design is a fully functional Virtex 4 FPGA design capable of demonstrating most of the features available on the DN8000K10. Features exercised in the reference design include:

- Access to the DDR2 modules at 200MHz
- Interaction with the Configuration FPGA and MCU
- Interact with external USB interface
- Access to external LEDs
- Implement 2.5Gbs Rocket I/O Transceivers
- Test daughtercard headers for connectivity.
- RS232 Communication
- Pin-multiplexed FPGA interconnect using LVDS at 700Mbs per signal pair
- Low-speed FPGA interconnect connectivity test

All source code for the FPGA reference design is included on the user CD and may be used freely in customer development. Precompiled configuration streams (.bit files) for your board are included can be used to verify board functionality before beginning development. A build script, described in the section Compiling the Reference Design can be used to generate new .bit files.

The Directory Structure of the User CD is as follows:
/Source Code
Contains All Dini Group-written Verilog, VHDL and C code

/... /USB_Software
Contains C source code and MSVS project for the USB Controller

/... /MCU
Contains C code for Cypress MCU, U200

/... /FPGACode
Contains the code for the DN8000K10 reference design

/... /RocketIO
Contains Verilog and VHDL code for the DN8000K10

/... /DN8000K10
Contains Verilog and VHDL code for the DN8000K10

/... /common
Contains general Verilog and VHDL code used in the DN8000K10 reference design that is not specific to DN8000K10.

/... /ConfigFPGA
Contains Verilog source for the configuration FPGA. Since this code is revised frequently, consult the Dini Group before modifying this code and check for firmware updates.

/... /certify_modules
Contains board description files for the DN8000K10.

/Schematics
/... /RevAX1
Contains an abbreviated Schematic of the DN8000K10 in PDF Format

/FPGA Programming Files
Contains compiled configuration streams for the 16 Virtex 4 FPGAs on the DN8000K10. These files were created using the Verilog source code in the Source Code directory, synthesizing, place and route in Xilinx ISE 7.1i, and generated as a .bit format file in bitgen.

/... /Main_Ref_Design
This contains the .bit files corresponding to the Main_ref branch of the source code.

/... /LVDS_interconnect
This contains the .bit files corresponding to the LVDS branch of the source code

/... /RocketIO
This contains the .bit files corresponding to the ROCKETIO branch of the source code.

/Documentation
/... /Manual
This directory contains the latest version of this document available when your DN8000K10 was shipped.

/... /Datasheets
This directory contains manufacturer’s datasheets for all of the standard or optional parts used on the DN8000K10

/... /Standards
This directory contains specifications that were used to design the DN8000K10

/Daughter Card
This directory contains files relating to the standard test daughter card available from Dini Group for use with the DN8000K10, the DNMEG300 and DNMEG400.

/... Schematics
This directory contains the Schematic of the DNMEG300/400
1.2 Running the Reference Design
Running the reference design requires default configuration settings. Create a main.txt file with the following contents and load it onto your configuration CompactFlash card.

Verbose level: 2 // Configuration RS232 port (P204) prints messages
Sanity check: y // prevents programming FPGAs with wrong bit streams

8442 PH0 Clock Frequency: 125.125Mhz
8442 PH1 Clock Frequency: 141Mhz
FX Clock Frequency: FX0_1 132Mhz
FX Clock Frequency: FX0_0 125Mhz
FX Clock Frequency: FX1_0 100Mhz
FX Clock Frequency: FX1_1 700Mhz
/PH0 Divide By: 2^1
PH1 Divide By: 2^2
/PH2 Divide By: 2^15
GCLK0 Select: 8442 // possible option SMA/8442/DIV/SS
GCLK1 Select: DIV // Selects DIVIDE clock
GCLK2 Select: SS / Select Single Step Clock (48Mhz right now)

For more information about the CompactFlash main.txt file, see Hardware: Configuration: CompactFlash.

1.2.1 The Precompiled Bit files
The Bit files on the user CD (D:\FPGA Programming Files\) are broken into three groups, the Main_test, RocketIOtest_v4, and LVDS_interconnect. To run each reference design test, the correct .bit file needs to be loaded into the tested FPGA. The tests have been split into three branches to reduce the effort required to place and route the designs.

The main_test files contain memory controllers for the DDR2 SODIMM modules, and some memory mapped internal-FPGA memory. It also contains IO registers accessible through the Main Bus interface for testing inter-FPGA interconnect at low speed.

The RocketIO_test files are only supplied for FPGA F0 and F12, the Virtex 4 FX parts. These designs send a test pattern out each RocketIO channel and compare transmitted data to received data. To use this test, you must connect each RocketIO channel with an external loop back.

The LVDS reference design is designed to run the inter-FPGA interconnect at 350Mhz, double-data rate. Each output sends a test pattern out and each input checks its received data against the expected test pattern.

For technical information on the reference design’s implementation of the Main Bus interface, see Reference Design: Address Maps. For more information about how the USB Controller program interacts with the reference design, see USB Software: Programmer’s Guide.

1.2.2 Load FPGAs with the reference design
Either compile the reference design following the steps provided in Compiling the reference design, or copy the compiled reference design provided on the user CD to a compact flash card, following the steps given in Quick Start Guide.

Turn on the DN8000K10 and allow the configuration circuitry to load the configuration streams into all 16 FPGAs.

1.2.3 Run the AETest program
The compiled windows version of AETest_usb is provided on the user CD. For Linux and Solaris users, follow the instructions in USB Software: Compiling AETEST.

Since there are three versions of the compiled reference design, you must do this step three times to use all of the options in the AETest program.

1.2.4 Run the RocketIO Test
Make sure the RocketIO version of the reference design is loaded into FPGAs F0 and F12.

From the AETest main menu, select option 4, MGT Menu. The MGT test sends a repeating test pattern out all of the RocketIO transmit pairs, and compares the input of each RocketIO channel to that pattern. To run the test, you must loop back each RocketIO pair.
You can easily loop back the SMA channels by connecting the RX and TX connectors of each MGT pair together with an SMA cable. The SFP modules can be tested with an LR loop back attenuator. (For LR modules)

Option 5 of the MGT menu allows you to invert the polarity of the Samtec QSE RocketIO tiles. This must be done if you are using the QSE loopback cable.

The MGT menu also allows you to modify the settings of the MGT tile. Some commonly-used testing features, like internal loopback and clock source settings are available from the MGT menu in the AETest application. Note that for many changes to take effect, the MGT tiles must be reset. All MGT settings can be changed from the AETest application using the DRP interface of the MGIs. Using this interface requires using a memory-map index provided in the Virtex 4 RocketIO User Guide.

To verify the function of each MGT tile and it’s error rate, select the Display registers COL 0 and 1 menu items in the MGT menu. This will display a dump of the IO registers controlling each MGT channel.
To cross-reference which MGT COL and TILE is connected to which external connector, see the *Appendix Pins Other*, or see *Hardware: MGT Serial Resources*. The connectivity is different from the F0 to the F12 FPGAs.

Frame Cnt: Shows a 64-bit counter clocked off the MGT tile’s USER_CLK. This counter is reset using option 9) Reset All in the MGT menu.
1.3 Compiling the Reference Design (Windows)

This section deals with the source code to the Reference Design, which can be found on the CD-ROM. All file references are with respect to the root directory of the Reference Design source code (/Source Code/FPGAcode). Files that are specific to the DN8000K10 design are found in the DN8000K10 subdirectory, whereas general application code is found in the common subdirectory.

1.3.1 Modify Source

Copy the FPGAcode directory structure to your hard drive working directory. The Xilinx tools will not allow spaces in the path to your working directory.

Open the top-level source file, fpga.v. There are several defined parameters in the top-level design file that can be used to change the features included in the compiled design. Each possible define statement is in the fpga.v file and commented out.

You should uncomment ONE of the following lines. This line ensures the correct external interfaces are implemented for the FPGA you wish to compile.

```vhd
//`define FPGA_F0
//`define FPGA_F1
//`define FPGA_F2
...
//`define FPGA_F4
```

You should leave the following line uncommented. This define is used by Dini Group for testing.

```
`define BOARD_DN8000K10
```
You may uncomment one of the following lines. The INTERCON_SINGLE define enables single ended testing of the inter-fpga interconnects. INTERCON_LVDS_DIR_ABC and INTERCON_LVDS_DIR_CBA are used to test high-speed inter-fpga interconnect. DIR_ABC and DIR_CBA versions both use fixed-direction IO between FPGAs. In each version, the direction is reversed to allow characterization of each bus in both directions.

`define INTERCON_SINGLE
//`define INTERCON_LVDS_DIR_ABC
//`define INTERCON_LVDS_DIR_CBA

The following define enables a DDR2 controller. This should be uncommented when compiling FPGA F1, F2, F13 and F14.

`define INCLUDE_DDR2

1.3.2 Xilinx ISE
A Xilinx ISE 7.1i project file is not included because ISE is updated so often. To place and route the reference design, create a Xilinx ISE project file.

Get the latest Xilinx ISE service pack from http://www.xilinx.com/xlnx/xil_sw_updates_home.jsp
Open Xilinx project navigator 7.1i SP4 or later. Earlier versions may not place and route the design as intended by the reference designer.

Select File->New Project
If you would like to use Xilinx built-in XST synthesis, select HDL. Select the speed grade of the FPGAs on your board. This information was specified in the packing slip, or can be obtained from the USB Controller software menu option Get Stuffing Options. In the Add Existing Source Dialog (if you are using XST) select only the file fpga.v. This Verilog source file will include all Verilog source files necessary. Make sure not to check the “add to project” box, or the Verilog compiler will not be able to find the included files in the directory structure.

If using XST for synthesis, the following paths must be added to the project:
`define synthesis
`define synthesis
This will disable all simulation-only code.

Add a .ucf file containing all of the signal Pin outs, IO settings, and location constraints. An example, working .ucf file has been provided for each FPGA in the directory /standard_reference_design/ucf/
The same .ucf file can be used for reference design branches Main_test and LVDS. Use /RocketIOtest_v4/ucf_8k10/ for compiling the RocketIO branch reference design.
A build utility is included that completes all the above steps using the command-line version of Xilinx ISE. The Build Utility is found at `FPGAcode/DN8000K10/standard_reference_design/build/make.bat`.

Make sure the following:

%\%XILINX\%/bin\nt;

are in your Path Environment variable.

Make.bat can be called with the following command line options.

- `[no option]` same as all
- `all` synthesizes, implements, and updates for all 9 FPGAs.
- `clean` delete all intermediate files, but leave out directory intact.
- `clean_all` delete all generated files
- `synthesize_all` synthesizes for all 16 FPGAs
- `implement_all` implements for all 16 FPGAs (edif files must exist in rev_1 directory)
- `update_all` updates for all 16 FPGAs (un-initialized bit files and bmm files must exist in out directory)
- `A` synthesize, implement, and update for fpga F0
- `B` synthesize, implement, and update for fpga F1
- `C` synthesize, implement, and update for fpga F2
- etc...
- `synthesize_a` synthesize for fpga F0
- `synthesize_b` synthesize for fpga F1
- `synthesize_c` synthesize for fpga F2
- etc...
- `implement_a:` implement for fpga F0
- `implement_b:` implement for fpga F1
- `implement_c:` implement for fpga F2
- etc...
- CBA
- ABC
- SINGLE

Outputs are generated and placed in a new directory `/out/`. During the place and route step, Xilinx ISE produces output files. These are placed in the `/ise/` directory.

To use Make.bat effectively, you need to use command line parameters. You can run a command line program from Windows explorer with command line options by making a shortcut to the Make.bat file, then right click and select properties. In the Target text input form, you can add command line parameters.

The reference design must support any subset of the 16 Virtex 4 FPGAs in any combination of LX100, LX160, and LX200 sizes. Compiler constants are used to include/exclude code, including sections specific to certain FPGAs, sections specific to memory controllers, or may
switch between the LX100 and LX200 part. There are four places where changes must be made to source and project settings to get the desired configuration:

I. XST synthesis project file
II. UCF files in 'source/ucf'
III. Dini Group Source code.

The Make.bat utility makes all of these changes based on the command line parameters supplied.

2 Implementation Details

Each test can be controlled and monitored by reading and writing the state of control registers in each design. These registers are made accessible to the world by memory-mapping them to the “Main Bus” interface. The main bus interface uses the DN8000K10 signals “MB80B[37:0]”. These signals are common to all 16 FPGAs. Note that these signals are broken by bus switches. In order to operate the main bus, and hence the reference design, the switches enabling the lower 5 bytes of the MB80B bus must be enabled. See Hardware: Interconnect: Main Bus for more information about the MB80B signals. See Address Maps below for details about the Main Bus interface.

The following subsections describe the function of each of the three reference design sets.

2.1 Main Test

This reference design tests FPGA interconnect at low speed, and tests the DDR2 SODIMM modules.

The IO buffers’ output and enable are memory mapped to the main bus interface. The registers REG_OUT, REG_OE, REG_IN, REG_EN control the FPGAs IOs that are used for inter-FPGA interconnect. See Address Maps for the locations of these registers on the main bus.
The test logic is performed in the USB Controller software. The test writes a test pattern to the output register of a selected FPGA and a selected bus, enables the output, and reads back the connected register on the connected FPGA. The test then alters the bit pattern and repeats.

The memory test portion of the Main_test reference design is implemented by mapping the DDR2 memory in the DIMM interfaces to the MainBus interface. This allows software on the host computer to read and write test patterns to the memory during the hardware test.

This portion of the test can be enabled in the source code by setting `define INCLUDE_DDR2. The memory controller is based on the DDR2 controller generated using the Xilinx memgen Verilog generator. The memory controller from Xilinx was modified to extend the bus width to 64 bits, adaptive clock phase alignment was added, and the clock structure was changed to fit within the clock resources on the DN8000K10. The contents of the DDR2 SODIMM memory interface are memory mapped to the main bus interface. See Address Maps for the location of the DDR2 memory interface in the Main Bus. Since the maximum address space of the memory is far greater than the memory space on the Main Bus, an address extension register, REG_DDR2HIADDR, is accessible from the Main bus (see Address Maps) that contains the upper bits of address used for all accesses to DDR memory. The REG_HIADDRSIZE register contains the number of bits in REG_DDR2HIADDR that are physically present. For the location of the REG_DDR2HIADDR and REG_HIADDRSIZE registers on the main Bus, see Address Maps.

The above diagram shows the clocking structure of the DDR2 interface. For the Main_test reference design, the DDR2 interfaces are run at 200MHz (400Mb/s x 64Bit). GCLK1 is used as the reference clock for a DLL (digital PLL) inside the FPGA. The output of this clock is fed onto a global clock (low fan-out) network to the signals DDR_CLK_OUTp/n. This signal is externally routed SSTL25 through an ICS855 clock driver configured as a non-PLL buffer. Matched outputs from this buffer are routed as SSTL18 signals to the CK0p/n and CK1p/n.
inputs of the DDR2 SODIMM, and the signal DDR_CLK_FBp/n back to a clock input of the Virtex 4 FPGA. The DDR_CLK_FB net routed internal to the FPGA on a dedicated clock feedback path to the DLL, where it ensures the DDR2 controller logic is clocks synchronous to the DDR2 SODIMM.

The latencies of the DDR2 interface are hard-coded to the maximum allowable DDR2 latencies 3-3-3-8.

The register REG_IDCODE returns a constant allowing the USB Controller program to recognize the board as a Dini Group reference design. The USB Controller program can then display reference design status and controls.

### 2.2 RocketIO V4 Test

This reference design is only provided in a compiled form for FPGAs F0 and F12, the only FPGAs on the DN8000K10 with RocketIO circuits. The RocketIOtest_v4 reference design sends a test pattern, either semi-random, 1010, worst case, or user-specified of the RocketIO outputs on the device, and tests the RocketIO input for that same pattern. The number of successful and failed “frames” of transferred data are counted and stored in the FRAMCNT and ERRCNT registers. (See *Address Maps*) These registers are 64 bits wide, and therefore have two addresses each on the Main Bus. The AETEST_WDM program polls and displays the value of these registers. For a functional link, FRAMECNT should be increasing and ERRCNT should be static. ERRCNT may be small non-zero because it counts errors that occur immediately after MGT reset).

In order to achieve a loop back link, a loop back cable needs to be installed on each of the MGT connectors. Loop back cable for XFP and SFP modules can be found at [www.fiberdyne.com](http://www.fiberdyne.com). Samtec QSE loop back cables are not available, although a QTE connector can be easily modified into a bit-inverting loop back. Standard SMA cables can be used to loop back the SMA connectors. The daughter card MGT connections can be looped back using the Standard Dini Group DNMEG-300 daughter card (See *Ordering Information: Optional Equipment*). Internal loop back can be enabled for MGTs using the LOOPBACK register.
The reference design uses the ICS84321 synthesizers to produce a reference clock for the MGT circuits. The test patterns are 8 bits wide, and converted to a 10-bit frame using built-in 8B10B encoding. The serial transmit frequency is 2.5Gbs. The MGTs internally provide a parallel clock output at 250Mhz that is used to clock the reference design logic. (The main bus interface is still run off GCLK2 at 48Mhz).

All of the configuration options available (without recreating the configuration stream) are brought out to registers accessible from main bus. (See Address Maps). This includes the powerful DRP interface, which allows modifications of the MGTs internal PLL and data routing control registers while the MGT is in operation. Operating links at 10Gbs will most likely require changing internal settings of the MGT circuit to work with your particular link. It is impossible for Dini Group to predict the link conditions of all users, so the default settings are intended to work well on a very short, well-matched noise free link.

### 2.3 LVDS Test

The LVDS test is made up of two sets of files, the ABC and CBA versions. The IOs on the FPGA pins are bi-directional, and signal integrity issues could be different in the two directions, so two compiled designs are supplied to each bus can be tested in both directions.

The IOs for the test are configured as 10:1 pin multiplexed IOs using the Xilinx ISERDES and OSERDES modules. (See Hardware: FPGA Interconnect). GCLK0 supplies the serial clock to the serdes modules at 350Mhz. The serdes modules provide a backend parallel interface at 70Mhz. Data is generated pseudo-randomly using a LFSR.

The design reference for this reference design is Xilinx App note XAPP704.

The external signals are configured as LVDS differential IO. Each external IO uses 2 OSERDES modules in ganged mode, and 2 ISERDES modules in ganged mode. The latency of the OSERDES-LVDS-ISERDES system is 2 clock cycles.

#### 2.3.1 Clock Summary

The clocks used in the Dini Group DN8000K10 reference design are:
3 Address Maps

3.1 Memory Space

The DN8000K10 reference design is controlled from the configuration FPGA over the MB80B[79:0] bus. All reference design functions are controlled by a memory-mapped register interface. This memory-mapped interface can be accessed through the USB controller program, by main.txt file commands, and through the RS232 port. See the next section for the reference design Main Bus interface specification. The memory map for the reference design functions is decoded as follows.

This map is only valid for the “MainTest” reference design.

DDR registers are only meaningful in FPGAs F1, F2, F13, F14.

The upper 4 bits are used by the Dini Group to distinguish FPGAs. F0 is hex ‘0’, F1 is hex ‘1’… F15 is hex ‘A’.

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Address</th>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0</td>
<td>0x08000002</td>
<td>IDCODE</td>
<td>0x05000121</td>
</tr>
<tr>
<td>F0</td>
<td>0x08000004</td>
<td>INTERCONTYPE</td>
<td>0x34561111</td>
</tr>
<tr>
<td>F0</td>
<td>0x08000006</td>
<td>RWREG</td>
<td>Scratch Register for testing</td>
</tr>
<tr>
<td>F0</td>
<td>0x08000010</td>
<td>LED_OE</td>
<td>Controls LED output enables</td>
</tr>
<tr>
<td>F0</td>
<td>0x08000011</td>
<td>LED_OUT</td>
<td>Controls LED outputs</td>
</tr>
<tr>
<td>F0</td>
<td>0x08100001</td>
<td>CLK_COUNTER</td>
<td>Contains contents of ACLK counter</td>
</tr>
<tr>
<td>F0</td>
<td>0x08100002</td>
<td>CLK_COUNTER</td>
<td>Contains contents of BCLK counter</td>
</tr>
<tr>
<td>F0</td>
<td>0x08100003</td>
<td>CLK_COUNTER</td>
<td>Contains contents of DCLK counter</td>
</tr>
<tr>
<td>F0</td>
<td>0x08100004</td>
<td>CLK_COUNTER</td>
<td>Contains contents of SYSCLK counter</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C000000</td>
<td>ABP0 OUT</td>
<td>W; the output state of FPGA IOs connected to the ABP0 interconnect bus</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C000004</td>
<td>ABP0 OE</td>
<td>W; The output enable of each FPGA IO on the ABP0 interconnect bus</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C000008</td>
<td>ABP0 IN</td>
<td>The input state of each FPGA IO… …on the ABP0 interconnect bus</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C00000C</td>
<td>ABP0 Name</td>
<td>“ABP0” (ascii)</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C000010</td>
<td>ABP1 OUT</td>
<td>W; ABP1 IO output values</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C000014</td>
<td>ABP1 OE</td>
<td>W; Output enable of ABP1 bus</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C000018</td>
<td>ABP1 IN</td>
<td>R; ABP1 input values</td>
</tr>
<tr>
<td>F0</td>
<td>0x0C00001C</td>
<td>ABP1 Name</td>
<td>“ABP1” (ascii)</td>
</tr>
</tbody>
</table>
FPGA F0 0x0C000XX0 BUS XX OUT XX can be 0-21 hex. Output status of IOs on bus XX.
FPGA F0 0x0C000XX4 BUS XX OE XX can be 0-21 hex. OE status of IOs
FPGA F0 0x0C000XX8 BUS XX IN XX can be 0-21 hex. The input values
FPGA F0 0x0C000XXC BUS XX Name The name of the bus XX (schematic)
FPGA F1 0x10000000 - DDR2 B space… Mapped to DDR2 SODIMM…
FPGA F1 0x17FFFFFF … …interface
FPGA F1 0x18000002 IDCODE 0x05000121
FPGA F1 0x18000004 INTERCONTYPE 0x34561111
FPGA F1 0x18000006 RWREG Scratch Register for testing
FPGA F1 0x18000010 LED_OE Controls LED output enables
FPGA F1 0x18000011 LED_OUT Controls LED outputs
FPGA F1 0x181000001 CLK_COUNTER Contains contents of ACLK counter
FPGA F1 0x181000002 CLK_COUNTER Contains contents of BCLK counter
FPGA F1 0x181000003 CLK_COUNTER Contains contents of DCLK counter
FPGA F1 0x181000004 CLK_COUNTER Contains contents of SYSCLK counter
FPGA F1 0x18000001 DDR2HIADDR upper address bits for DDR2 interface
FPGA F1 0x18000003 HIADDRSIZE number of bits in DDR2HIADDR
FPGA F1 0x18000005 DDR2SIZEHIADDR The size of the DDR2 module.
FPGA F1 0x18000007 DDR2TAPCNT0 Current IDelay values of DDR2…
FPGA F1 0x18000008 DDR2TAPCNT1 …interface
FPGA F1 0x1C000XX0 BUS XX OUT XX can be 0-21 hex. Output status of IOs on bus XX.
FPGA F1 0x1C000XX4 BUS XX OE XX can be 0-21 hex. OE status of IOs
FPGA F1 0x1C000XX8 BUS XX IN XX can be 0-21 hex. The input values
FPGA F1 0x1C000XXC BUS XX Name The name of the bus XX (schematic)
FPGA F2 0x20000000- DDR2 C space… Mapped to DDR2 SODIMM…
FPGA F2 0x27FFFFFF … … interface
FPGA F2 0x28000002 IDCODE 0x05000121
FPGA F2 0x28000004 INTERCONTYPE 0x34561111
FPGA F2 0x28000006 RWREG Scratch Register for testing
FPGA F2 0x28000010 LED_OE Controls LED output enables
FPGA F2 0x28000011 LED_OUT Controls LED outputs
FPGA F2 0x281000001 CLK_COUNTER Contains contents of ACLK counter
FPGA F2 0x281000002 CLK_COUNTER Contains contents of BCLK counter
FPGA F2 0x281000003 CLK_COUNTER Contains contents of DCLK counter
FPGA F2 0x281000004 CLK_COUNTER Contains contents of SYSCLK counter
FPGA F2 0x28000001 DDR2HIADDR upper address bits for DDR2 interface
FPGA F2 0x28000003 HIADDRSIZE number of bits in DDR2HIADDR
3.1.1 Decoder
The following is an address decoder diagram.
## REFERENCE DESIGN

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>FPGANUM</th>
<th>INTERCONSEL</th>
<th>REGSEL</th>
<th>DDRSEL, INTERCONSEL, REGSEL</th>
<th>DDRSEL, INTERCONSEL, REGSEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:28</td>
<td></td>
<td></td>
<td></td>
<td>Determines which FPGA contains the register</td>
<td>Selects DDR2 Memory map, External IO Memory map, or Internal control registers</td>
</tr>
<tr>
<td>27:25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td>F0</td>
<td>0xx</td>
<td>DDRSEL</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
</tr>
<tr>
<td>0001</td>
<td>F1</td>
<td>110</td>
<td>IOSEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>F2</td>
<td>100</td>
<td>REGSEL</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
</tr>
<tr>
<td>0011</td>
<td>F3</td>
<td>111</td>
<td>ROCKETIO</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
<td>DDRSEL, INTERCONSEL, REGSEL</td>
</tr>
<tr>
<td>1111</td>
<td>F15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### DDRSEL

- **Address Range:** 0000 - FF
- **Description:** Memory mapped to DDR2 memory module (32-bit words) (F1, F2, F13, F14 only)

### IOSEL

- **Address Range:** 0x0 - 0xF
- **Values:**
  - 0x0: LVDS_GENREG
  - 0x1: LVDS_BANK0_REG

### REGSEL

- **Address Range:** 11:4
- **Width:** 32 bits
- **Bus Select:**
  - 9D:9E: MB64
  - 9A:9C: MB80
  - 00:99: FPGA-FPGA
- **Register Select:**
  - REG_OUT: The output bit for the selected IO
  - REG_OE: Enables the output Buffer for the IO
  - REG_IN: (Read only) the value of the interconnect signal
  - REG_EN: 0xFFFFFFFF (30-bit values)

### ROCKETIO

- **Address Range:** 21:20
- **Width:** 19:16
- **Values:**
  - DDRHIADDR: 13 bits (Haddr)
  - DDRSIZE: 8 bits (its x 128MB)
  - IDCODE: 32 bits (returns ID_CODE)
- **Other registers:**
  - CLKCOUNTER: "DEAD5566"

## DN8000K10 User Guide

- **RESET**
  - Bit0: reset
  - Bit1: reset
- **LOOPBACK**
  - 2 bits – Loopback mode
- **POWERRDOWN**
  - 1 bit – powerrdown mode
  - 2 bits – select test pattern
  - 2 bits
- **CLKSTABLE**
  - 2 bits
- **POLARITY**
  - 3 bits – MGT RXSTATE
  - 32 bits (Last frame’s data)
4 Main Bus Signaling

The bus is implemented over the shared MB80B bus (80 bits). This bus has connection to all 16 Virtex 4 FPGAs. See Appendix Pins Other for the pin connection of these signals.

The reference design transactions over main bus are synchronous to GCLK2. For proper operation, this clock should be set to a frequency 50Mhz or below.

All transactions are initiated by the configuration FPGA. To complete a read, the Configuration FPGA presents a 32-bit address on the AD bus (MB80B[31:0]) and asserts an address latch enable signal, ALE (MB80B[32]). Each FPGA using the interface should register the value on the AD bus. On the next clock cycle, the Configuration FPGA asserts RD to begin a read command. Each FPGA using the interface should decode the address to determine if it should begin to control the DONE and VALID signals. For 1-255 clock cycles, the user FPGA may extend the read command by holding VALID and DONE low. During this time, the configuration FPGA will not attempt to begin a new transaction.

To send data back to the configuration FPGA, the user FPGA presents the 32 bit data value on AD[31:0] and asserts VALID (MB80B[35]). To release control of the main bus, the user FPGA one cycle later asserts DONE (MB80B[33]). It must then tri-state the AD, DONE, and VALID signals. The user should use the Xilinx PULLDOWN on the DONE and VALID IO buffers to prevent these signals from changing state during an idle main bus.

A write command is initiated by the configuration FPGA.
One cycle after presenting the address, the configuration FPGA presents a 32 bit data on the AD bus. The user FPGA then asserts DONE 1-255 clock cycles late.
Ordering Information

Dini Group part number
DN8000K10

1 FPGA Options

1.1 FPGA F0, F12
Select an FPGA part to be supplied in the F0 and F12 position. These FPGAs slots are the only available with the FX family of Virtex 4 FPGAs. Installing these slots enables use of RocketIO MGT 10Gb serial transceivers.

In order to use SFP FX1_SFP[0/1], SAMTEC J235, the FX60 or FX100 part is required for F12.
In order to use daughter card DC0 channel 3,4 or SMAs Channel 2,3, the FX60 or FX100 part is required for F0
In order to use the SAMTEC channel 5,6,7,8 the FX100 part is required or F0 and F12

In order to achieve 10Gbs operation on the MGT transceivers, speed grade –12 is required. In order to achieve 1Gb operation of the inter-FPGA interconnect, the –12 speed grade is required.

NONE
FX40 –10 –11 –12
FX60 –10 –11 –12
FX100 –10 –11 –12

1.2 FPGA F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F13, F14, F15:
Select an FPGA part to be supplied in the B position. This FPGA is connected to an expansion header, a memory module socket, and can source global clocks. The –12 speed grade is required for full speed operation (1Gbs/pair) of the interconnect between FPGAs.

NONE
1.3 Configuration FPGA:
The configuration FPGA (Xilinx Virtex 4 LX 40)

For special configuration requirements, an expansion header is connected to the configuration FPGA. Connectivity to this header requires upgrading the configuration FPGA to an LX80.

2 Multi-Gigabit Serial Options

2.1 Serial Clock Crystals
The DN8000K10 is equipped with frequency synthesizers (ICS84321) for its high-speed serial (MGT) interfaces. These synthesizers are appropriate for serial transmission speeds of up to 10Gbs. By default, the crystals supplied are 25.0000Mhz (MGT left or right columns) and 25.5000Mhz (MGT left column only)

These selections are suitable to meet most common serial protocols that are supported by Xilinx MGTs and the DN8000K10. If you have specific MGT requirements, be sure to contact Dini Group about your needs before placing your order.

Dini Group keeps the following crystals in stock for MGTs:
The default option is 25.000 MHz.

3 Optional Equipment

3.1 Rack mount Chassis
The DN8000K10 is shipped on a steel base plate “carrier” to provide protection, stability and an easy way to transport the board. Optionally, the DN8000K10 can ship inside a 4U rack mount chassis. This carrier can be used externally to the chassis, as well as mounted inside the chassis. The carrier allows the assembly to rest either FPGA-side up or FPGA-side down. The steel base plate provides strain relief for daughter card insertion and removal, and provides a heat-sinking path for the motherboard.

The chassis front panel provides an LCD display, a remote power switch, a configuration reset switch, a logic reset switch, a remote Compact Flash drive, and cooling fans. Apertures with cover plates are used to provide access for daughter card cables.
The fans mounted on the front panel of the chassis provide cooling when the DN8000K10 is operated with the chassis lid on. A 500W EPS power supply mounts inside the chassis and exhausts out the backside of the chassis.

Controls available on the front face of the Chassis are logic reset button, system reset button, power on button, an LCD readout with input buttons, USB port and four RS232 ports.

3.2 Daughter cards
The Dini Group supplies standard daughter cards and memory modules that you can use with the DN8000K10.

- DN8000K10DC300 card – 48 signals on Mictor connectors, 62 signals on .1” TP headers. 4 RocketIO channels, global clock input.
- DN8000K10DC400 card – 48 signals on Mictor connector, 124 signals on .1” pitch TP headers, 4 RocketIO channels, global clock input.

- SRAM module for use in the 200-pin SODIMM sockets of the DN8000K10. QDRII, 300Mhz 64x2Mb

- SRAM module for use in the 200-pin SODIMM socket. 64x2Mb Standard SDR SRAM. Pipelined or Flow through, NoBL available

- RLDRAM module for use in the 200-pin SODIMM socket. 64x16Mb, 300Mhz DDRII

- Flash module for use in the 200-pin SODIMM header.

- Mictor module for use in the 200-pin SODIMM header. (2 Mictor 38 connectors for use with logic analyzer)

### 4 Third Party Equipment

You may also want to obtain from a third party vendor

- DDR2 SDRAM SODIMM (200 pin)
  PC2-3200 CL=3 Unbuffered
  Acceptable Part List from crucial.com
  CT12864AC40F (1GB)
  CT6464AC40F (512MB)
  CT3264AC40F (256MB)
**REFERENCE DESIGN**

- **SFP modules**
  
  IBM part 13N1796 from insight.com $180 (FibreChannel)
  Asante GBIC 1000SX insight.com $104 (1000Base-SX)
  3Com 3CSFP81 insight.com $131 (100Base-FX)
  3Com 3CSFP93 insight.com $154 (1000Base-T)

- **XFP modules**
  
  - Intel part TXN181070850X18 from insight.com $692
  - XFP heatsink/clip – Tyco part 1542992-2
  - 5.2V bench supply for powering ECL-based XFP modules (if required)

- **Xilinx Parallel IV cable** $95, or **Xilinx Platform USB cable** $149
  Xilinx.com

- **LVPECL oscillators for RocketIO MGT clocking. (The DN8000K10 comes with a 250Mhz oscillator standard)**
  Epson Part EG-2102CA PECL digikey.com $40

- **Xilinx ChipScope** for embedded logic analyzer functionality.

- **Surface-mount reference crystals for RocketIO (ISC84321) synthesizers or global (ICS8442) synthesizers.**

## 5 Common Problems

### 5.1.1 Simulation Does not match Synthesis

Make sure that the clock your design uses is running with an Oscilloscope or the USB Controller program.

Check the pin out in your constraint file against the schematic or the Dini Group supplied USF files. Common pin assignment mistakes are with the daughter card headers.

Check the place and route (.PAR) report file to make sure that 100% of the IOBs you used have LOC constraints. Even if one IOB is not constrained, it could be placed in a location that causes other problems with the board or your design.

Read the .PAD report to make sure your constraints were applied correctly.

For problems with the Main Bus, make sure that none of the other FPGAs are driving those MB pins.
Make sure tri-state signals with pull-ups are driven before they are released. The timing in simulation will be different than in operation.

Pack all IOs in IOB registers. The compiler hint to do this in XST is `xc_props="IOB=true"`. Check the map report (.mrp) and check the % IO packed in IOB. This should be 100%. Sometimes the place and route tool removes registers from the IO block to meet timing constraints. This can be stopped by adding clock-to-out and clock-to-in constraints. See Xilinx documentation for the OFFSET constraint.

Make sure that the "Unused IOBs" option in the ISE bitgen settings is set to "Float." If it is set to "Pull down," then be aware that FPGAs are driving any pin that is not assigned in the source code.

5.1.2 XST is giving errors because it cannot find virtex4.v or resync.v
Add the line
`define synthesis
when using XST.

The provided XST projects include ../../../common/ddr2/ddr2_to_mb/ in the path. You must include this path in your ISE project, or copy the contained files to your source directory.

5.1.3 Signal name[0] exists in my code, but place and route MAP step complains the signal name[0] from my ucf file does not exist
The XST compiler renames bus indexes using <> brackets instead of [] brackets. Now the mapping step cannot see the signals. Rename the IO constraints in the ucf file to use <> brackets.

If the output of a module is not used, your synthesis program may remove the module completely. A UCF constraint on that module will fail.

5.1.4 Certify fails because the ports on my Virtex 4 do not match the specified part
There is an error in the Certify part description files for FX60 parts in the 1153 package. Copy these files from the user CD in the /certify/ directory.

5.1.5 The DN8000K10 cannot read my SmartMedia card. The RS232 port prints: Bad SM card format
If you formatted the SmartMedia card using Windows, the DN8000K10 will no longer be able to read the file structure on the card. Reformat the card using the program on the User CD D:/3rdParty/SMFormat

5.1.6 There is a lot of deterministic jitter (DJ) on my RocketIO channel
For long links, or links exhibiting loss over poor connectors (all connectors). The default preemphasis settings need to be changed. Read the Virtex 4 RocketIO Users Guide, UG076. The DRP registers that need to be adjusted are TXPRE, TX, TXPOST
5.1.7 There is a lot of deterministic jitter (DJ) on my RocketIO clock signals
Estimate the deterministic jitter on your clock by sending a 101010 pattern over a RocketIO TX pin, or using an active probe directly on the outputs of the RocketIO differential oscillator. Routing clocks through FPGA fabric or using a non-MGT clock input will cause a large duty cycle distortion.

5.1.8 My design works correctly in all the FPGAs except for one
Make sure you are meeting all your design constraints. A marginal timing margin will work in most FPGAs, but not all.

5.1.9 RocketIO is getting a high bit error rate
5.1.10 RocketIO is not working

5.2 Common Questions
5.2.1 Where are all of the debug pins
DNMEG-300 and DNMEG-400 daughter cards for debug access are available from Dini Group for a nominal fee.

5.2.2 Where can get the daughter card connectors?
Dini Group can supply these in small quantity at cost.

5.2.3 Can I use two Dini Products at the same time?
The USB utility, AETest_usb can select among any number of Dini Group products at once. You must run a separate copy of AETest for each board. There is a menu option "change current device", that lets you switch to another installed device.

The feature will be added to USB Controller soon. Contact support@dinigroup.com for requests.

5.2.4 Do you provide a board description file for ____?
No. Just Certify.

5.2.5 Where is the VHDL reference design?
The VHDL reference design will not be available at the initial release of the DN8000K10. Email support@dinigroup.com for a schedule