# XtremeDSP Development Kit User Guide

NT107-0132 – Issue 9

Nallatech is the leading FPGA-Centric Systems provider, with unrivalled industry expertise in the provision of complete systems including hardware, IP and software.

Nallatech's industry-leading products and services are used in many application areas, including DSP & Imaging, Telecommunications & RF, Aerospace & Defence, and Networking & Storage. Nallatech offers design services for complete systems, in addition to the modular DIME™ and DIME-II™ product ranges for electronic systems.

## Contacting Nallatech and Xilinx:

### Support:

***WWW:***
If your DSP kit was purchased directly from Xilinx, go to http://support.xilinx.com/.
If the kit was purchased via Nallatech go to www.nallatech.com and click 'support'.

### Headquarters

***Europe & Asia-Pacific:***
Nallatech
Boolean House
One Napier Park
Cumbernauld
Glasgow G68 0BH
United Kingdom

***North America:***
Nallatech Inc
12565 Research Parkway
Suite 300
Orlando
Florida 32826
United States of America

### Phone/Fax

***Europe & Asia-Pacific:***
Phone: +44 (0)1236 789500
Fax: +44 (0)1236 789599

***North America:***
Phone: +1 407 384 9255
Fax: +1 407 384 8555

### Email:

sales@nallatech.com

### WWW:

www.nallatech.com

**Document Name**: XtremeDSP Development Kit User Guide

**Document Number**: NT107-0132

**Issue Number**: 9

**Date of Issue**: 21-05-2003

**Revision History:**

| Date | Issue Number | Revision |
|---|---|---|
| 14/05/02 | 1 | Initial release |
| 30/05/02 | 2 | Pinout table fixes, inclusion of information on 'Interface FPGA to User FPGA Communications'. |
| 28/06/02 | 3 | Pinout table fixes |
| 15/07/02 | 4 | Pinout table fixes, additional example application. |
| 25/08/02 | 5 | Update to documentation – various |
| 26/09/02 | 6 | Section 5.11 added. Interface FPGA to User FPGA information added. |
| 04/10/02 | 7 | Updated documentation |
| 24/02/03 | 8 | Clocking pinouts for DACs and ADCs updated in BenADDA section, power supplies LED table updated in BenONE section. |
| 21/05/03 | 9 | Update to FPS power calculation. |

# Contents

# List of Figures

# List of Tables

# Part 1

# Introduction

This part of the User Guide provides an introduction to the XtremeDSP Development Kit and contains information on the structure of the User Guide, followed by a Getting Started Guide. The following Sections include:

- Section 1: Preface
- Section 2: Getting Started

# Section 1

# Preface

In this Section:

- About this User Guide
- User Guide format
- References
- Abbreviations

## 1.1 About this User Guide

This User Guide provides detailed information on using the DIME-II XtremeDSP Kit. The User Guide is designed to provide information that allows you to become acquainted with the kit and the functionality it provides.

Throughout this document there are symbols to draw attention to important information:

The blue 'i' symbol indicates useful or important information.

The red '!' symbol indicates a warning, which requires special attention.

## 1.2 User Guide Format

The User Guide is divided into **Sections**, which are grouped into **Parts.** The Parts divide the document as follows:

- Introduction: Provides an introduction to the User Guide and a Getting Started Section so you can get the kit up and running as quickly as possible.
- BenONE: Provides detailed instructions for installing and using the BenONE DIME-II motherboard.
- BenADDA: Provides detailed instructions for installing and using the BenADDA DIME-II module.

The Parts and Sections of the document are summarised below:

| | |
|---|---|
| PART 1 - Introduction | Section 1: Preface<br>Section 2: Getting Started |
| PART 2 – BenONE | Section 3: BenONE Overview<br>Section 4: Installation Guide |
| PART 3 – BenADDA | Section 5: Implementation Guide<br>Section 6: Reference Guide<br>Section 7: BenADDA Overview<br>Section 8: Installation Guide<br>Section 9: Implementation Guide<br>Section 10: Reference Guide |

## 1.3 References

There are a number of additional sources of information about specific aspects of the products used in the XtremeDSP kit. Generally these are included in the documents folder in the CDs indicated in brackets.

- Analog Devices — AD6644 ADC Datasheet Rev.0 (XtremeDSP Kit CD)
- Analog Devices — AD9772A DAC Datasheet Rev.0 (XtremeDSP Kit CD)
- Maxim — 1617 Datasheet (XtremeDSP Kit CD)
- Micron — ZBT SRAM Datasheet (XtremeDSP Kit CD)
- Nallatech Ltd — NT107-0068 FUSE C/C++ API Developers Guide (FUSE CD)
- Nallatech Ltd — NT107-0068V2 FUSE System Software User Guide (FUSE CD)
- Nallatech Ltd — NT107-0103 DIMEScript User Guide (FUSE CD)
- Nallatech Ltd (FUSE CD) — NT302-0000 PCI to User FPGA Interface Application Note
- Xilinx — Virtex-II Datasheet (available on Xilinx Website)

## 1.4 Abbreviations

- **ADC:** Analogue to Digital Converter
- **DAC:** Digital to Analogue Converter
- **DIME**: DSP and Image Processing Modules for Enhanced FPGAs
- **DLL:** Delay Locked Loop
- **ESD:** Electro-Static Discharge
- **FPGA**: Field Programmable Gate Array
- **ILA:** (Xilinx) Integrated Logic Analyser
- **PCI:** Peripheral Component Interconnect
- **RAM:** Random Access Memory
- **VHDL:** VHSIC (Very High Speed IC) Hardware Description Language
- **ZBT** Zero Bus Turnaround

# Section 2

# Getting Started

## 2.1 Overview

The XtremeDSP Development Kit serves as an ideal development platform for Virtex-II and provides an entry into the scalable DIME-II systems available from Nallatech. Its dual channel high performance ADCs and DACs, as well as the user programmable Virtex-II device are ideal to implement high performance signal processing applications such as Software Defined Radio, 3G Wireless, Networking, HDTV or Video Imaging.

This 'Getting Started Guide' covers the installation of the Nallatech hardware and software components that are provided in the DIME-II XtremeDSP kit. This document details how to connect your DIME II XtremeDSP kit to your PC using the supplied USB cable and the driver and software installation process that follows. The hardware portion of the kit consists of a BenONE (PCI motherboard DIME-II carrier card) and a BenADDA DIME II module that is plugged into the available DIME-II slot on the BenONE. Full details of the board specifics are available in the full documentation provided on the XtremeDSP Development Kit CDROM.

## 2.2 XtremeDSP Development Kit Requirements

The following Minimum System requirements are recommended for the software and this card:

- Pentium PII 233
- 32MB RAM
- 1GB Hard Disk
- USB v1.1 compatible port
- Windows 98/ME/2000/XP operating system. (NT4 is supported but not over USB)

If you wish to install the BenONE – PCI card in a PC using a PCI slot please note that in the default configuration provided with the DSP Kit the BenONE will only function correctly in a 5V PCI Signalling environment.

## 2.3 Unpacking Your Starter Kit

### 2.3.1 Hardware Installation Features

The XtremeDSP Development Kit contains Electro Static Discharge (ESD) sensitive devices. ESD handling procedures must be observed during handling of the cards.

The physical features of the XtremeDSP Development Kit referred to in these instructions are highlighted below:

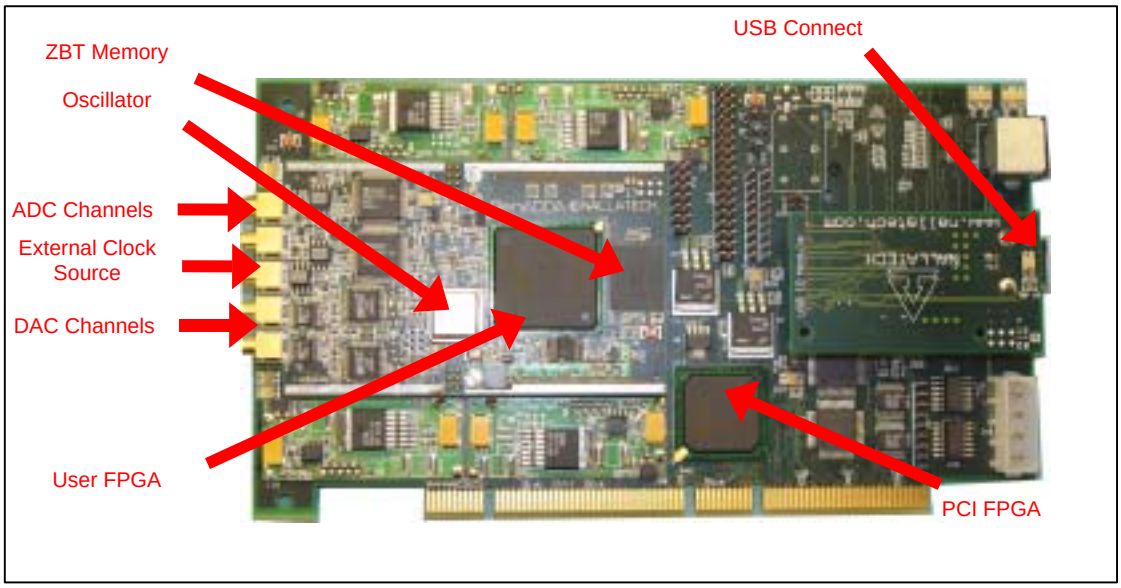



**Figure 1: Hardware features relating to connection and installation**

## 2.3.2 XtremeDSP Development Kit Contents

This kit provides a motherboard populated with a daughter card also called a DIME-II module. It is shipped in a stand-alone case, and includes all necessary power supplies as well a USB cable. In addition, it includes the FUSE software for configuration and control as well as a set of evaluation software for the design and implementation tools.

- Hardware
  - BenONE Motherboard populated with a BenADDA DIME-II module in a stand-alone case
  - External power supply (US Mains cable with separate UK, European or Australian mains adaptors)
  - Wide ranging input (90 – 264Vac), multiple output, power supply, generating;
    - +5 Volts @ 2A
    - +12 Volts @ 1A
    - -12 Volts @ 200mA.
  - USB v1.1 compatible cable, 2 metres long
  - 3 MCX to BNC cables for connecting to the ADC / DAC and external clock connectors.
  - Carrying Case
- Software
  - Nallatech FUSE (Field Upgradeable Systems Environment) Software CD. Provides the ability to control and configure FPGA, and provides facilities to transfer data between the BenONE motherboard and a host PC).
  - Nallatech XtremeDSP Development Kit CD that provides documentation on the BenONE and BenADDA products as well as supporting files for their use within the FUSE framework.
  - Xilinx Evaluation CD kit

- 30 days evaluation for the Xilinx Foundation ISE
- 90 days evaluation for the Xilinx System Generator for DSP
- 30 days evaluation for Matlab Simulink

- Reference Design
  - A design example is provided with the kit and reference designs are also made available on the Xilinx web site – http://www.xilinx.com/dsp

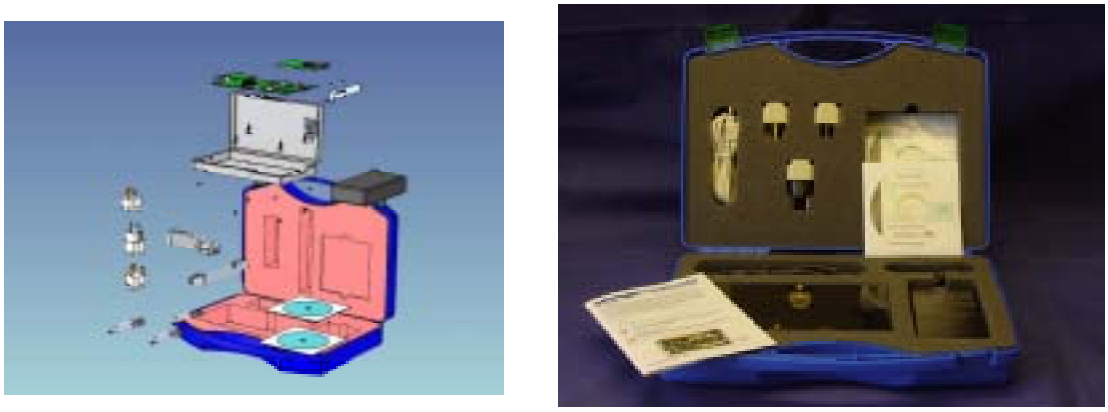


**Figure 2: Kit Case and Contents**

### 2.3.3 Specifications

The dual-channel high-performance ADCs and DACs, as well as the user-programmable Virtex-II FPGA, are ideal for implementing high-performance signal processing applications such as Software Defined Radio, 3G Wireless, networking, HDTV or video imaging.

**Motherboard – (BenONE card)**

- Supports one DIME-II slot for any DIME-II Module
- Spartan 2 FPGA for 3.3V/5V PCI or USB interface
- Host interfacing via 3.3V/5V PCI 32-bit/33-MHz or USB v1.0 interfaces.

**Daughter Board – (BenADDA module)**

- Virtex-II user FPGA: XC2V2000-4FG456
- 2 ADC channels: AD664 ADC (14-bits up to 65 MSPS)
- 2 DAC channels: AD9772 DAC (14-bits up to 160MSPS)
- Support for external clock, on board oscillator and programmable clock
- One bank of ZBT-SSRAM (133Mhz, 256Kx16 bits)

## 2.4 BenONE/BenADDA connectivity

The XtremeDSP kit consists of the BenONE and the BenADDA DIME-II module. The following diagram shows the connectivity between the BenONE motherboard and a DIME-II module site.

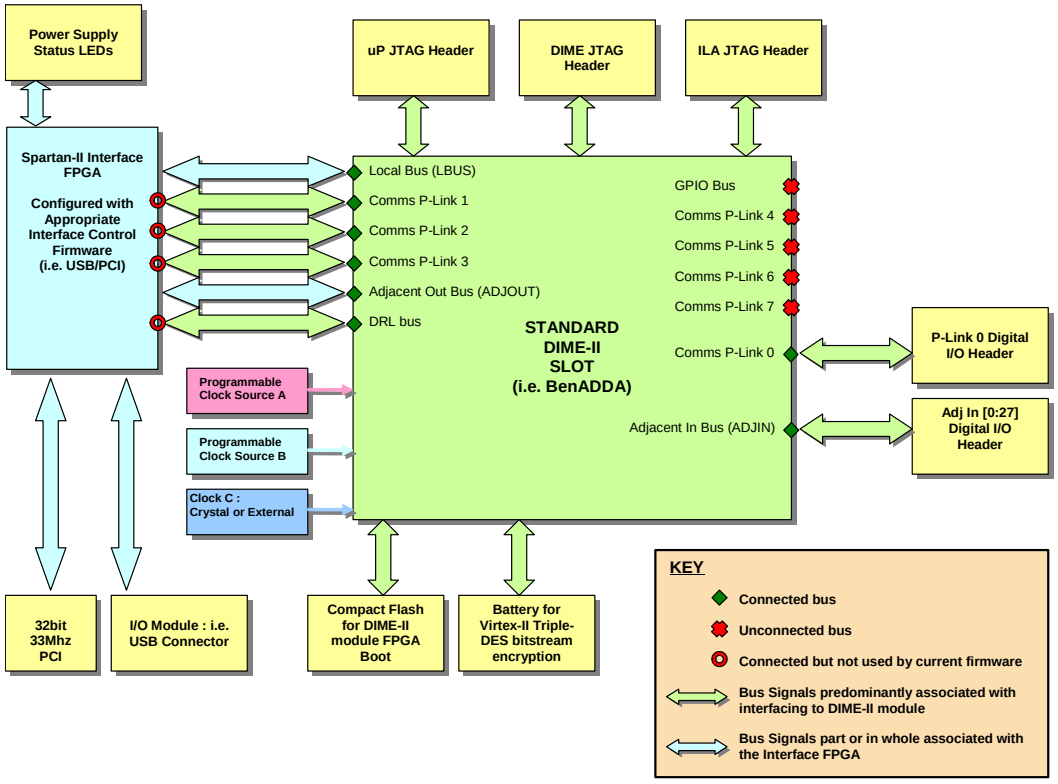



**Figure 3: BenADDA/BenONE connectivity**

The connectivity to the FPGA on the specific resources, such as FPGAs on the DIME-II module, will depend upon the DIME-II module itself. In the case of the XtremeDSP development kit the available I/O from the User FPGA on the BenADDA is defined in Table 1. Note that certain configurations of the BenADDA DIME-II module provide greater I/O resources than can be supported on the BenONE itself. This is due to the fact that the BenONE can be fitted to other Nallatech DIME-II carrier cards that have onboard resources with sufficient pin resources to make use of these additional bus signals.

| Communication Bus | 2V250 (FG456) | 2V1000 (FG456) | 2V2000/ 2V3000 (FG676) | 2V3000 – 2V8000 (FF1152) |
|---|---|---|---|---|
| Adjacent IN | 16 bits | 26 bits | 40 bits | 64 bits |
| Adjacent OUT | 16 bits | 26 bits | 40 bits | 64 bits |
| Comm P-Link 0 | 12 bits | 12 bits | 12 bits | 12 bits |
| Comm P-Link 1 | N/a | N/a | N/a bits | 12 bits |
| Comm P-Link 2 | 12 bits | 12 bits | 12 bits | 12 bits |
| Comm P-Link 3 | N/a | N/a | N/a bits | 12 bits |
| Comm P-Link 4 | N/a | N/a | N/a bits | 12 bits |
| Comm P-Link 5 | N/a | N/a | 12 bits | 12 bits |
| Comm P-Link 6 | N/a | N/a | N/a bits | 12 bits |
| Comm P-Link 7 | N/a | N/a | 12 bits | 12 bits |
| Local Bus | 32 bits | 54 bits | 64 bits | 64 bits |
| GP IO Bus | N/a | N/a | N/a bits | 21 bits |

**Table 1: Communication Bus Summary**

In addition, certain buses are in part used to provide a communications channel between the Interface FPGA (Spartan-II) and the DIME-II module site (in effect the User FPGA on the BenADDA). On the BenONE, part of the Local Bus and the ADJOUT bus are used for this purpose.

## 2.5 Installation Guide

The installation process is essentially split into 3 stages

(1) Install the Nallatech FUSE environment software provided on the FUSE CD

(2) Connecting the hardware to the PC and installing the device driver for the BenONE when detected under Windows.

(3) Install the FUSE supporting software for the BenONE and BenADDA boards provided on the XtremeDSP Development Kit CD

### 2.5.1 FUSE Software Installation

Before installing FUSE ensure that the BenONE motherboard is NOT powered up.

Insert the supplied FUSE CD which autoruns to load the CD menu. If the CD does not auto-run, run the following program: *autorun.exe*. When the program runs, the screen shown below will appear:


**Figure 4: FUSE CD Autoplay Menu**

1. Click the first option 'Install FUSE Application Software'
2. The installation process begins. It uses a standard installation interface, with which most users will be familiar. Work through the dialog boxes, filling in details as required until the 'Finish' dialog box is reached.
3. Press 'Finish' to install the software.
4. The PC then needs restarted prior to continuing the installation.

## 2.5.2 Hardware Driver Installation

The following driver installation can be performed on power up or with the host PC already powered and the operating system's desktop visible.

1. Connect the supplied power supply to a suitable mains supply. The power supply accepts mains input from 90 – 264Vac.
2. Apply power to the BenONE card (see Figure 1)
3. Connect the BenONE card to your host PC with the supplied USB cable.
4. Found New Hardware Window will appear followed by the Step 1 screen (or similar) – Press Next
5. Step 2 screen should appear. Choose the top option (search for suitable drivers option) then press Next
6. Step 3 screen should appear. If not already ticked, select the CD-ROM choice – Press Next.
7. Windows will inform you that it has located a valid USB driver, BenONE - PCI Loader Firmware – Press Next. If you are presented with an intermediate error that it is not a Windows Certified driver please continue anyway.
8. Step 4 screen should appear indicating that the installation has been completed successfully - Press Finish.
9. At this stage you will notice that one of the bank of 3 LEDs on the BenONE will switch off to indicate that the hardware is successfully initialised in terms of the USB interfacing to the PC.
10. To ensure correct installation you may view the Device Manager window. This is accessed by a right mouse click on the My Computer Icon on your desktop.
11. Select Properties from the drop down list that appears.
12. Click on the Device Manager button (or TAB).
13. Screen opposite will appear. At the bottom of the tree structure expand the Universal Serial Bus Controllers branch. You should see 'BenONE - PCI Firmware Loader' driver present.
14. Remove the USB cable form either the host PC or the BenONE - PCI should dynamically remove the driver from this branch.





15. Reconnect the cable. The driver should be added to this branch only this time it should read 'BenONE - PCI'. If the driver still reads 'BenONE – PCI Firmware Loader', remove the USB cable, then the power cable on the BenONE card.

16. Reconnect the power cable and USB cable to the BenONE, the driver should now read 'BenONE - PCI'.

### 2.5.3 BenONE / BenADDA FUSE Supporting Software

Insert the supplied XtremeDSP Development Kit CD that contains the required files for the BenADDA and BenONE boards. This CD should autorun and display the Menu screen. If the CD does not auto-run, run the following program:
*autorun.exe*.


**Figure 5: DSP Kit CD Autoplay Menu**

1. Click on the first option 'Install Product FUSE Software'

2. The installation process begins. It uses a standard installation interface, with which most users will be familiar. Work through the dialog boxes, filling in details as required until the 'Finish' dialog box is reached.

3. It is then recommended to restart the PC at this stage.

## 2.6 Confidence Test

In order to verify the correct installation of the software and hardware a simple confidence test can be performed. In the Windows 'Start Menu' under Programs a new entry will have been created for FUSE.

Run the program FUSE→Software→FUSE Probe #. # (where #. # refers to the distributed version number).

You will see the GUI shown in Figure 6. If it does not appear correctly make sure the window is maximised.

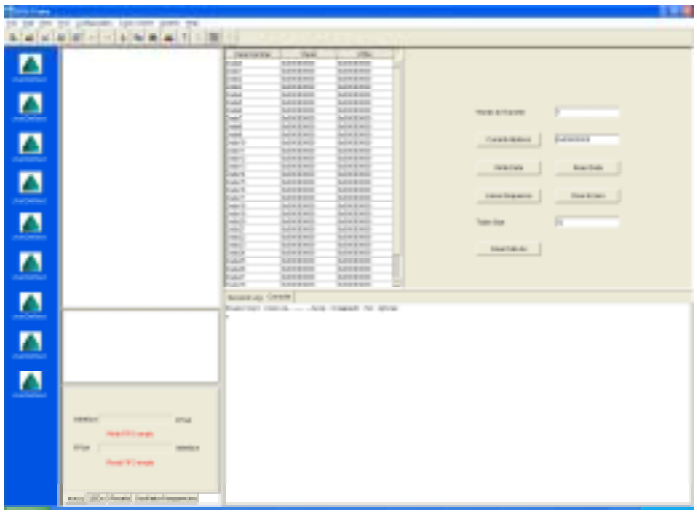


**Figure 6: FUSE Probe #.'#**

The simplest way to make sure that the software and hardware has been installed correctly is to open the card in FUSE Probe. Open the cards with the following steps:

(1) From the 'Card Control' Menu select 'Open Card'

(2) This will open the dialog box for selecting the type of card and interface over which you wish to open. In this case please select USB and 'All card types' as show in Figure 7.

(3) Press the 'Locate Cards' button to carry out the search for cards on the selected interface.

(4) You should then be presented with the location results specifying what cards were found on the selected interface. With the kit you should see the BenONE listed

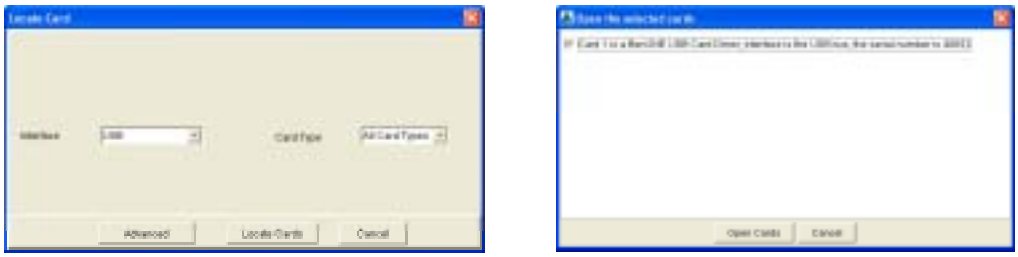


**Figure 7: Locate Cards prompt/Detected Cards**

(5) Press the 'Open Cards' button to open the selected cards. These will then be listed in the card viewer in the FUSE Probe GUI. In the case of the DSP kit you will see the BenADDA and the BenONE listed in the tree structure window.

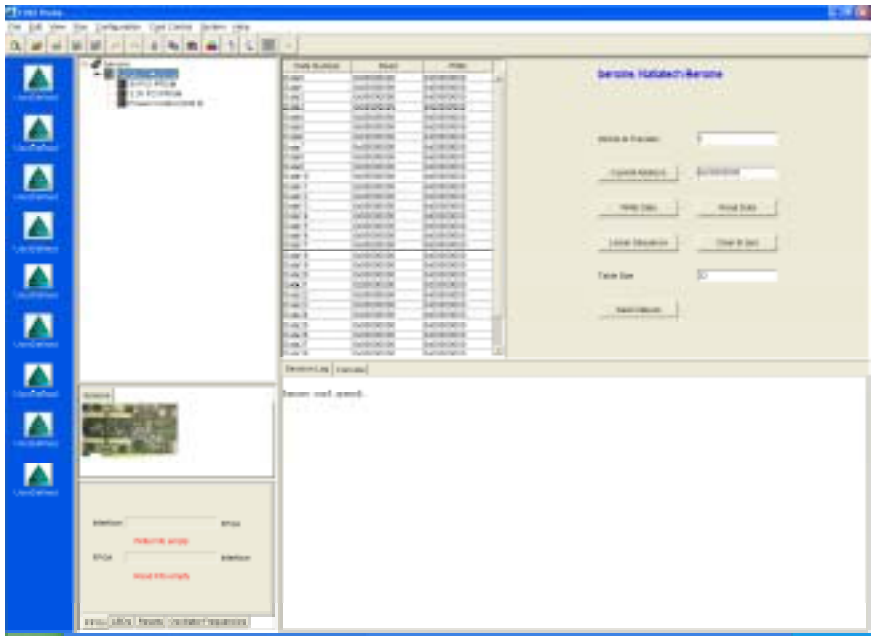


**Figure 8: FUSE Probe Tool with Open Cards**

Once the BenONE and BenADDA have been opened, the power LEDs for the supplies used, change from red to green as shown below in Figure 9:

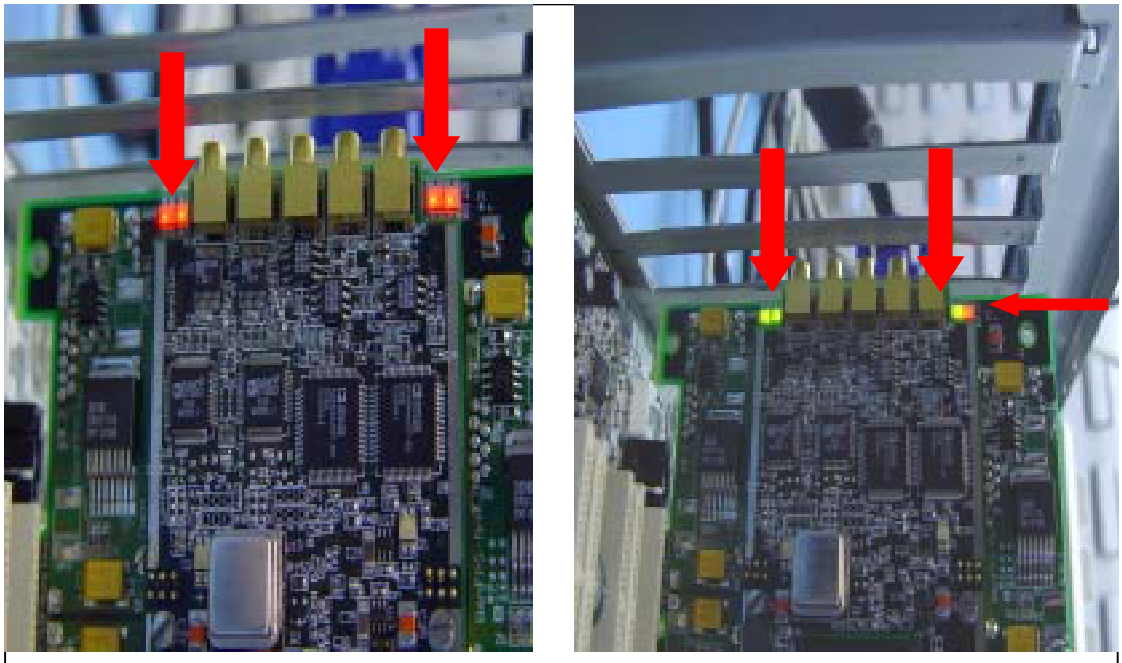


**Figure 9: Power LEDs**

Note that if a power supply is not used then the power indicator for that supply will remain RED indicating it is not switched on. For example, on the BenADDA only three supplies are required and so only three of the four LEDs will change from red to green.

## 2.7 Support

There are several other sources of information on using the Nallatech hardware and FUSE software. These are included on the CD in the documentation folder but may also be installed onto the local machine as part of the installation process.

### 2.7.1 Document References

- NT107-0068V2 FUSE System Software User Guide
    - This document is intended to give detailed instructions on how to use Nallatech's FUSE (Field Upgradeable Systems Environment) Software, in particular the FUSE Probe tool. This user Guide is included on the FUSE CD supplied in the folder: <CDROM>\ Documentation\FUSE
- NT107-0068 FUSE C/C++ API Developers Guide
    - This developers guide provides detailed information on installing and using the FUSE (Field Upgradeable Systems Environment) C/C++ API. The main focus of the guide is to provide information that allows the user to become acquainted with the FUSE C/C++ API and the functionality that it provides. This user Guide is included on the FUSE CD supplied in the folder: <CDROM>\ Documentation\FUSE
- NT107-0103 DIMEScript User Guide
    - This guide provides information on the use of a script language called DIMEScript that is installed with FUSE. This is a simple yet powerful scripting tool that can be invoked from within FUSE Probe. This User Guide is included on the FUSE CD supplied in the folder: <CDROM>\ Documentation\FUSE

### 2.7.2 FUSE CD Structure

The structure of the supplied FUSE CD takes the following format:

- Application Notes – contains applications notes general to FUSE with some specific card examples
- DIMEScript – contains further information and source on DIMEScript.
- Documentation – contains FUSE (non card specific) documentation as noted in section 2.7.1
- FUSE API Examples
- Software

### 2.7.3 XtremeDSP Development Kit CD Structure

The structure of the supplied XtremeDSP Development kit takes the following format:

- Documentation - contains documentation in the form of Adobe PDF documents
- Drivers - contains driver files for use under various operating systems
- Examples - provides examples in the form of VHDL source and pre-generated bitfiles
- UCFs - contains UCF (User Constraint Files) for the user FPGA provided on the BenADDA module
- Application Notes

### 2.7.4 Technical Support References

- Xilinx support available on the internet at http://support.xilinx.com/
- Nallatech DSP Kit support lounge. Access to this lounge is available on establishment of a maintenance agreement. This lounge provides access to Nallatech software updates and relevant application notes as they become available.

# Part 2

# BenONE

This part of the User Guide provides you with information on installing and using the BenONE DIME-II motherboard. In the following Sections:

- Section 3: BenONE Overview
- Section 4: BenONE Installation Guide
- Section 5: BenONE Implementation Guide
- Section 6: BenONE Reference Guide

# Section 3

# BenONE Overview

In this Section:



## 3.1 Description

The BenONE - PCI is a PCI, single slot DIME-II motherboard. It is classed as an entry-level motherboard and is capable of hosting a single width DIME-II module. The board has no FPGA resources available to the user; all resources are addressed on an attached module. It does however have the capability of a secondary connection to a host PC, for example USB (primary connection being PCI). This is achieved by the addition of an IO module on the motherboard. Another feature of the BenONE – PCI is that it can be used in standalone solutions using Compact Flash technology. The Compact Flash is a specific option that is not included as standard in the XtremeDSP Development Kit. The BenONE also performs housekeeping functions of the Programmable Power Supplies and PCI bus. Finally, connection to further Nallatech motherboards and modules is made possible by the inclusion of a pin header connection direct to the module site.

## 3.2 Key Features

The key features of the BenONE - PCI are:

- PCI/Control Xilinx® Spartan-II FPGA, pre-configured with PCI/Control Firmware
- Single DIME-II module expansion slot
- 32 bit/33MHz PCI interface with expansion to 64bit/33Mhz via firmware upgrade.
- Two on-board clock nets
- 2 Programmable clock sources
- 1 Fixed Oscillator socket
- Status LEDs
- JTAG configuration headers
- User selectable pin headers
- Fixed or fully programmable power supplies
- Nallatech FUSE Software for FPGA configuration over PCI
- Nallatech FUSE Software Library for board interfacing & control

***Build Options***

- USB 1.1 I/O Module interface
- Battery Backup for Virtex-II® Encryption Keys
- Compact Flash using Xilinx® System Ace chipset
- External oscillator input via mini coax connector

## 3.3 Functional Diagram

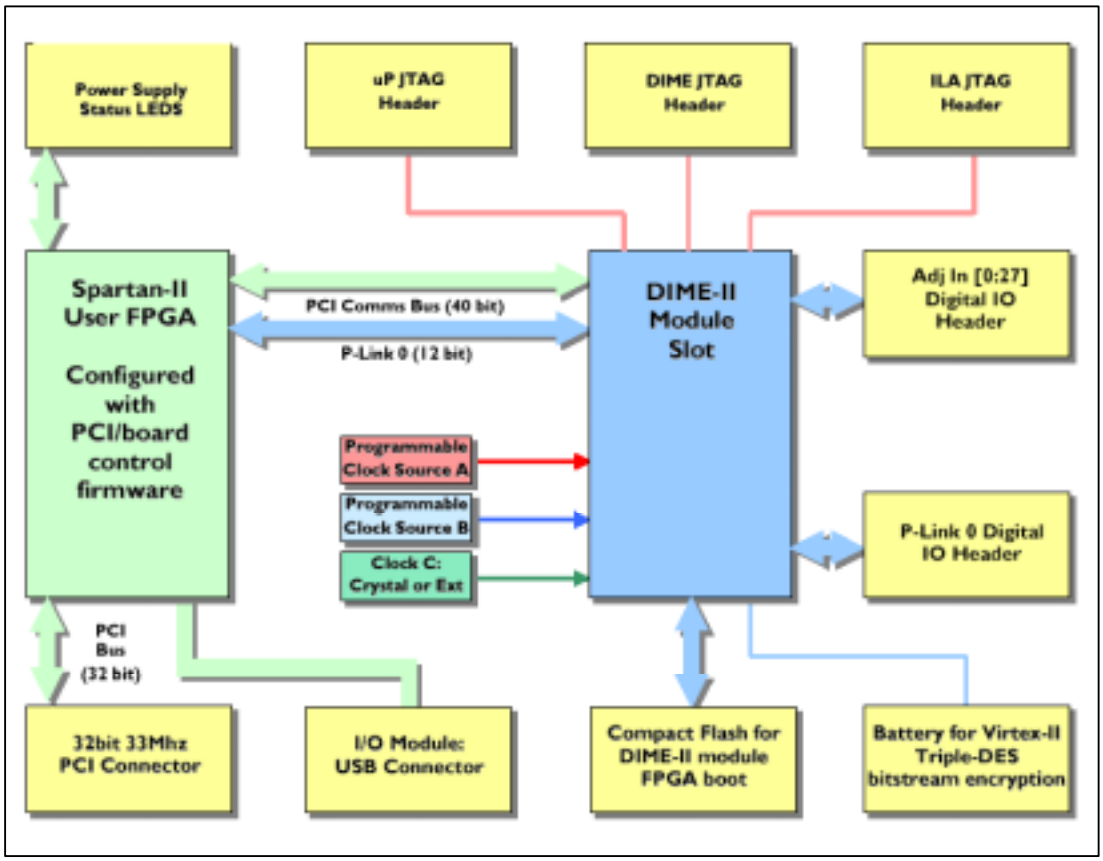



**Figure 10: BenONE - PCI Functional Diagram**

# Section 4

# Installation Guide

In this Section:

- Hardware Installation
- Software Installation

# Hardware Installation

- Hardware requirements
- Hardware installation features
- Hardware installation instructions for both PCI and USB

## 4.1 BenONE-PCI host PC requirements

- Pentium PII 233
- 32MB RAM
- 1GB Hard Disk
- Windows 95/98/Me/NT/2000/XP operating system or Red Hat Linux 6.2 or above.

### 4.1.1 PCI Connection

PCI defines two types of signalling environment, which operate at either 3.3v or 5v. The BenONE-PCI is a universal card and thus can be used in either signalling environment.

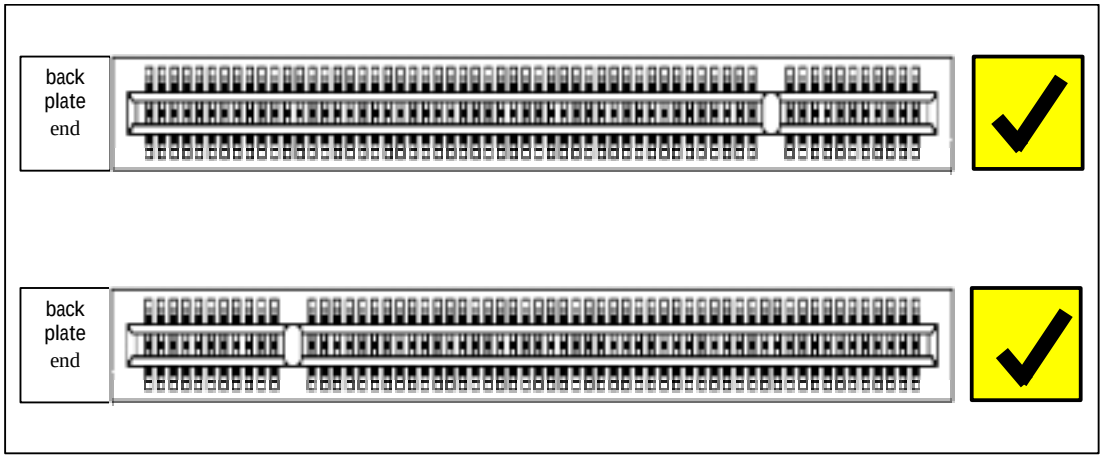



**Figure 11: 5V (top) and 3.3V (bottom) signalling PCI Connectors**

If you wish to install the BenONE – PCI card in a PC using a PCI slot please note that in the default configuration provided with the XtremeDSP Development Kit User Guide, the BenONE will only function correctly in a 5V PCI Signalling environment. This is because one of the XC1800 proms is programmed with a 5VIO PCI bitstream and the other PROM that can contain the 3.3VIO PCI bitstream has been used for the USB bitstream.

### 4.1.2 USB Connection

If you will be using the USB connection to connect the BenONE – PCI to a host PC you will require in addition to the above minimum requirements

- USB v1.1 compatible port

Windows NT or earlier versions of Win95 do not support USB.

Please note. Although Linux supports USB, at present Nallatech do not support this option.

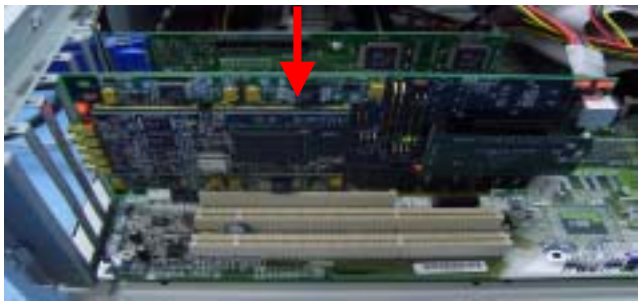
## 4.2 Hardware Features

The BenONE is an Electro-Static Discharge (ESD) sensitive device. ESD handling procedures must be observed during the handling and installation of the BenONE - PCI card.

## 4.3 PCI Installation

1. Ensure the PC you are using has the power switched off prior to installation.
2. Having removed the PC cover, locate a free PCI slot for the BenONE - PCI:

3. Fit the BenONE - PCI into the PCI slot you have selected and push down firmly:



4. Fit the fixing screw to secure the backplate of the BenONE - PCI to the chassis of the PC.
5. The PC power can then be switched on.

6. As a confidence test that the BenONE - PCI is correctly installed, the power Status LED and PCI/Control FPGA configuration LEDs illuminate to indicate the card has configured correctly.

7. On the BenONE - PCI, the 3.3V status LED on the 'front' of the card also illuminates to indicate that a 3.3V supply is available from the PCI slot. The illuminated LEDs are shown below:

The hardware installation of the BenONE - PCI is now complete. It should be noted that at this stage the card will not function, until the BenONE - PCI drivers and software have been installed. Instructions on how to do this are given in 'Software Installation' on page 25.

## 4.4 USB Installation

The following installation can be performed on power up or with the host PC already powered with the operating system's desktop visible.

Windows 98/ME/2000/XP Installation

1. Connect the supplied power supply to a suitable mains supply. The power supply accepts mains input from 90 – 264Vac.

2. Apply power to the BenONE.

3. As a confidence test that the BenONE - PCI is correctly installed, the power Status LED and PCI/Control FPGA configuration LEDs illuminate to indicate the card has configured correctly.

4. Connect the BenONE card to your host PC with the supplied USB cable.

The hardware installation of the BenONE is now complete. It should be noted that at this stage the card will not function, until the BenONE - PCI drivers and software have been installed. Instructions on how to do this are given in 'Software Installation' on page 25.

# Software Installation

- PCI driver installation
- USB driver installation
- FUSE software installation

## 4.5 PCI Driver Installation

### 4.5.1 Windows 95/98/2000/ME/XP (Pro) Installation

Administrative Privileges are required for Windows 2000 / XP installation

1. Power up PC with BenONE – PCI installed.
2. During boot up the PC should report that a new device has been found, a PCI co-processor CPU.
3. The following screen (or similar) should appear:



4. Click Next…the following screen should appear:

5. Choose the top option (the recommended choice) then press Next…the following screen should appear:


6. If not already ticked, select the bottom choice. Browse and locate the *CD_Drive*:\Drivers folder on the installation CD, then select the relevant folder (Win9x/Win2k/WinNT/WinXP) as appropriate for your installation.
7. Windows will now load the new drivers and acknowledge the installation
8. On the last screen Click finish.
9. Next click YES to restart the PC, allowing the drivers to be loaded


The PC will now restart and the driver installation is complete.

Under Windows XP Pro you may see the warning "Windows cannot initialise the device driver for this hardware". This is expected and the warning is taken care of when installing the DIME software as directed in 'FUSE Software Installation' on page 30.

Once you have installed the drivers you should proceed to 'FUSE Software Installation' on page 30 to install the DIME Software that is required to communicate with the card.

## 4.5.2 PCI - Windows NT Installation

Administrative Privileges are required for Windows NT installation

The Windows NT drivers need to be installed by running the Nallatech software from Windows. These drivers are now installed automatically during the installation of the card software.

## 4.6 USB Driver Installation

The following driver installation can be performed on power up or with the host PC already powered and the operating system's desktop visible.

### 4.6.1 Windows 98/ME/2000/XP Installation

1. Connect the supplied power supply to a suitable mains supply. The power supply accepts mains input from 100 – 240vac.
2. Apply power to the BenONE - PCI card (see Figure 1.)
3. Connect the BenONE - PCI card to your host PC with the supplied USB cable.
4. Found New Hardware Window will appear (or similar) – Click Next


5. Choose the top option (search for suitable drivers option) – Click Next

6. If not already ticked, select the CD-ROM choice – Click Next.

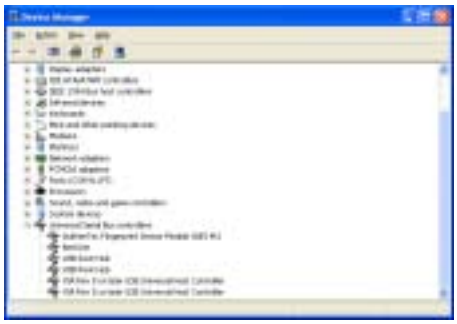
7. Windows will inform you that it has located a valid USB driver, BenONE - PCI Loader Firmware – Click Next.

8. The screen below should appear indicating that the installation has been completed successfully - Click Finish.


### 4.6.2 Checking USB Installation

To ensure correct installation you may view the Device Manager window. This is accessed by a right mouse click on the My Computer Icon on your desktop.

1. Select Properties from the drop down list that appears.
2. Click on the Device Manager button (or TAB).
3. The screen below will appear. At the bottom of the tree structure expand the Universal Serial Bus Controllers branch. You should see 'BenONE Firmware Loader' driver present.



4. Removing the USB cable from either the host PC or the BenONE should dynamically remove the driver from this branch.
5. Reconnect the cable. The driver should be added to this branch only this time it should read 'BenONE'. If the driver reads 'BenONE Firmware Loader', remove the USB cable, then remove power from the BenONE card.
6. Reconnect the power cable and USB cable to the BenONE, the driver should now read 'BenONE'.

An additional visual indicator of the card being successfully initialised over USB is the bank of 3 LEDs on the BenONE. Initially when the card boots on power being applied, all three of these LEDs should be illuminated. When the USB cable is connected and the OS successfully communicates with the USB interface on the board, one of these LEDs will turn off, as shown below in Figure 12. This process normally takes around 5 seconds. If the LED never turns off on connection of the USB cable, there are two common causes:

1. The USB drivers have not be installed properly as described in 'USB Driver Installation' on page 27.

2. It is not commicating properly with the PC USB interface. In this case try unplugging the USB cable from the BenONE, cycling the power to the BenONE and then reconnecting the USB cable.

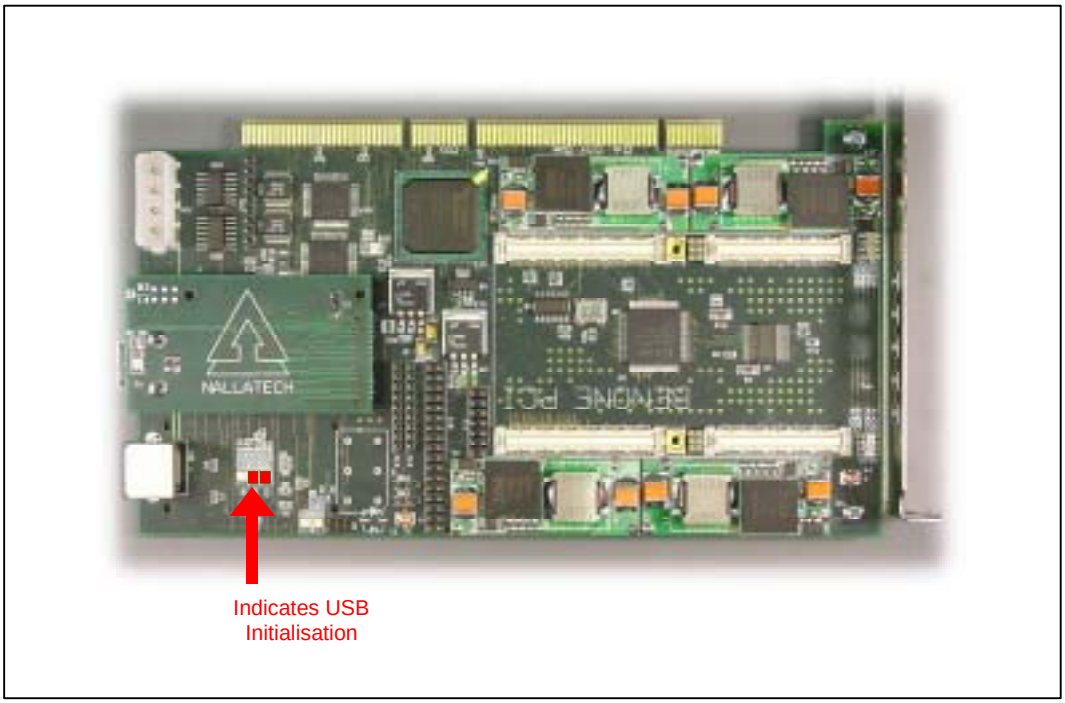



**Figure 12: LEDs indicating USB Successful Initialisation**

## 4.7 FUSE Software Installation

Prior to installing the product CD it is necessary to have installed the FUSE Software that is supplied on a separate CD. Simply insert the FUSE CD into the drive and select 'Install to FUSE Application Software' from the menu that appears. See 'Getting Started' Section on page 5 for more information.

### 4.7.1 Linux

Linux is supported for the XtremeDSP Development Kit. Support files for FUSE under Linux are included on this CD. However it is necessary to purchase FUSE for Linux to use this. Please contact Nallatech for more details.

Tested on **red**hat Standard installation 6.2.

1. Insert the installation CD
2. Mount the installation CD
   ```
   > mount –t iso9660 /dev/cdrom /mnt/cdrom
   ```
3. Go to the root of the area in which you installed the FUSE software e.g.
   ```
   > cd /usr/local/nallatech
   ```
4. Unzip the files of the LinuxBenDSPK.tgz
   ```
   > tar –xvzf /mnt/cdrom/linux/Ballynuey2/LinuxBenDSPK.tgz
   ```
5. Once you have installed them to test the BenDSPK installation (assuming you have a BenDSPK card installed).
   ```
   > cd /bin
   > sh/fuse
   ```

This will load up the FUSE configuration and probe tool. Use this tool to open the BenDSPK. See the FUSE System Software User Guide for more details on how to do this.

This installation also covers LINUX support for the BenADDA module.

Note that the driver is a reloadable module which should be installed at start-up. This can be done by typing the command '> insmod windrvror' putting this command in the start-up script '/etc/profile'. Note also that this must be done with root privilege. The device created '/dev/windrvr' has root access. For other users the permissions will have to be changed.

# Section 5

# Implementation Guide

In this Section:

- FPGA Configuration
- Clock Configuration

## 5.1 FPGA Configuration Options

Any FPGAs on the DIME-II module can be configured via a variety of methods.

- Using the FUSE Software GUI (Graphical User Interface)
- Using DIMEScript
- Using FUSE Software APIs
- Using an external JTAG programmer, via the ALT JTAG chain and pin header

Additionally, on-module FPGAs can also be configured as follows:

- Using external JTAG programmer, via the uP JTAG chain and pin header, assuming the module supports uP JTAG.

## 5.2 FPGA Configuration using FUSE

The FUSE Software provides the following functionality:

- Configuring FPGAs on DIME-II modules
- Controlling reset signals
- Controlling programmable clock frequencies
- Sending data to designs running in the User FPGA
- Reading data from designs running in the User FPGA
- Resetting PCI FIFOs

This functionality is provided in the following guises:

- FUSE Probe Application
- DIMEScript
- FUSE Software development APIs

### 5.2.1 Device/Module ordering for configuration

When configuring FPGAs using FUSE Software, it is necessary to use **Module ID** and **Device ID** numbers to target the correct FPGAs for configuration, as most systems will have multiple FPGAs. The

only exception to this is when using the FUSE GUI, where the software allows FPGAs to be targeted graphically.

Each FPGA must be uniquely identified, in order that the user can target the correct bit-file to each device. The system used for identifying FPGAs (and also any PROMs in the DIME JTAG chain) consists of a **module ID,** identifying the module the FPGA is on and a **device ID,** identifying the device within that module.

The **module ID** and **device ID** are determined by the order in which the devices are configured and how many modules and devices are present. The **module ID** always starts at 0 and increments for each additional module to be configured. It should be noted therefore, that the maximum **module ID** depends upon the number of modules fitted.

The **device ID** works in a similar way to the **module ID**. The **device ID** always starts at 0 and increments for each additional device to be configured within a module or virtual module. It should be noted therefore, that the maximum **device ID** depends upon the number of device present. The ordering of the module configuration on the BenONE is shown in Figure 13. In order to clarify this concept, some example configurations follow in this Section.

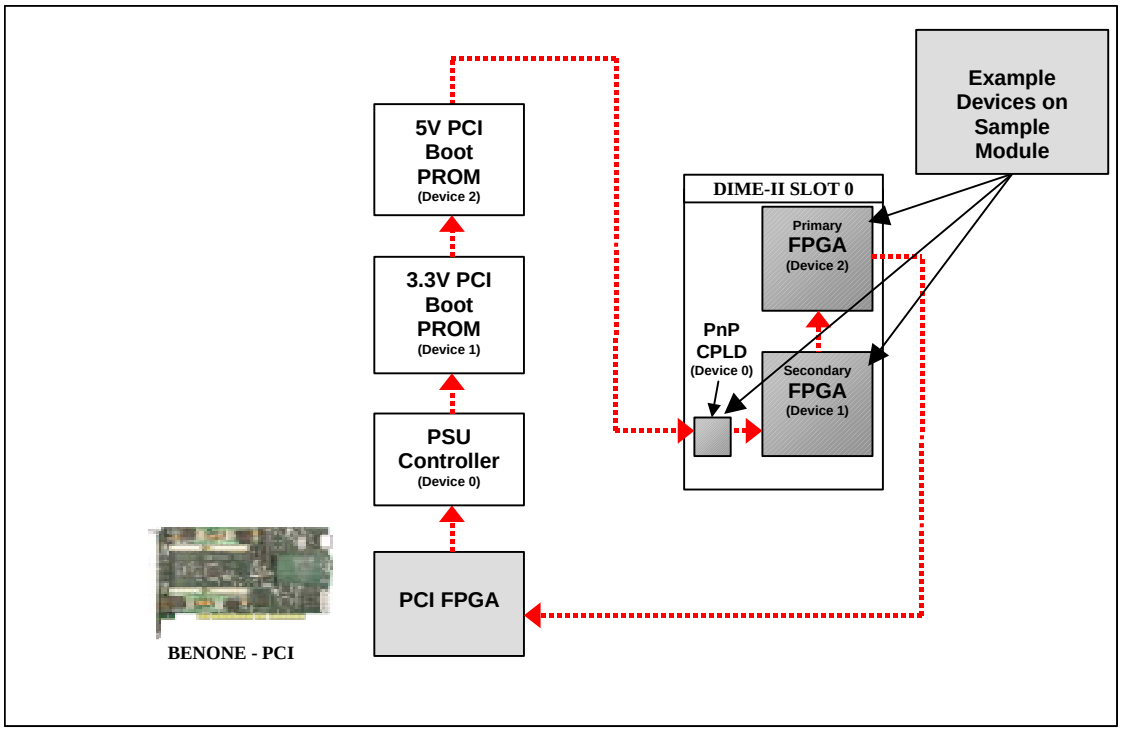



**Figure 13: BenONE - PCI JTAG Configuration within DIME Software**

## 5.2.2 FPGA Configuration Example 1

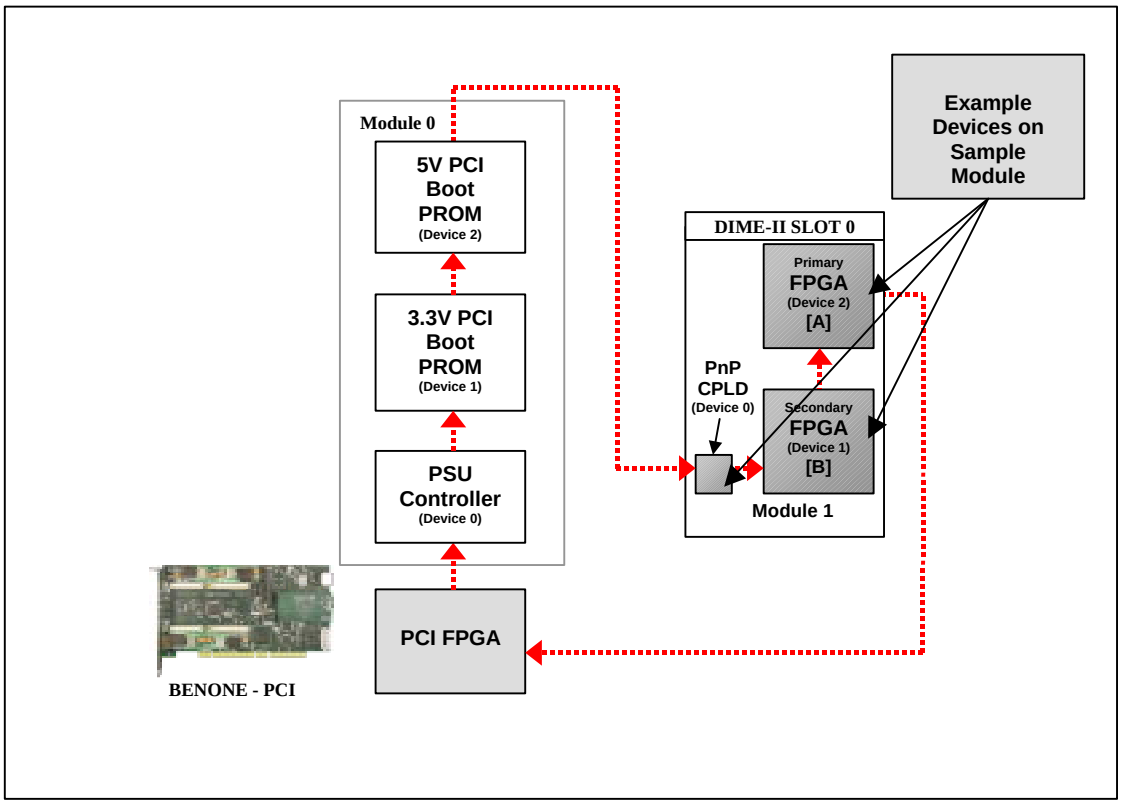



**Figure 14: Example configuration 1**

Configuration example 1 assumes the DIME-II slot to be populated and the module to have two FPGA devices. The **module ID** and device ID numbers for this configuration are identified in Figure 14 and listed for each device A – I in Table 2 below.

| FPGA Device | Module ID Number | Device ID Number |
|---|---|---|
| A | 1 | 2 |
| B | 1 | 1 |

**Table 2: Example configuration 1 module and device ID numbering**

## 5.2.3 Key Steps for FPGA Configuration

When developing applications incorporating FPGA configuration, using FUSE APIs or DIMEScript, there are a number of key steps that need to be performed. The key steps are shown here as an example. In the example, the FUSE C/C++ API is used, but the principle is the same for any of the FUSE APIs, or DIMEScript. Further and more detailed explanations are available in the API User Guides or DIMEScript User Guide.

To communicate with the BenONE - PCI, the software must find the card in the host system. This can be achieved by using the **DIME_LocateCard(int LocateType, DWORD MBType, void* LocateTypeArgs, DWORD DriverVersion, DWORD Flags)** function. Various arguments are required by the function to locate the card. For example, LocateType and MBType respectively determine which interface and motherboard type should be searched for.

Once the software has found the BenONE - PCI, the next step is to open the card using **DIME_OpenCard(LOCATE_HANDLE LocateHandle, int CardNumber, DWORD Flags)**. This is required to open the card and performs all the necessary set-up procedures in order to interface to the BenONE. This function requires several arguments to open the card: LocateHandle is the handle returned by the DIME_LocateCard function, CardNumber is the index of the card within the locate handle that the user wishes to open while Flags is a parameter which allows users to customise the card

opening process. The DIME_OpenCard handle is passed to later functions in order to allow these functions to communicate with the card.

Having opened the board, a number of software functions can then be called to perform a variety of different operations on the re-programmable devices in the chain.

This stage covers functions such as configuration of individual devices, setting bit filenames, resetting FPGA devices, and other functions listed in the above Section. This will be dependent on the particular application.

Once users have finished using the BenONE - PCI, the handle returned from DIME_OpenCard should be closed in order to free all the resources used to interface to the card. This can be accomplished by using the **DIME_CloseCard(DIME_HANDLE CardHandle)** function. Additionally, the handle returned from DIME_LocateCard should also be closed. This can be achieved using the **DIME_CloseLocate(LOCATE_HANDLE LocateHandle)** function.

Please consult the FUSE C-C ++ API Developers Guide for further details on all the available DIME software functions.

## Sample Configuration Example

A sample configuration sequence for "FPGA Configuration Example 1" from Section 5.2.2 is shown below:

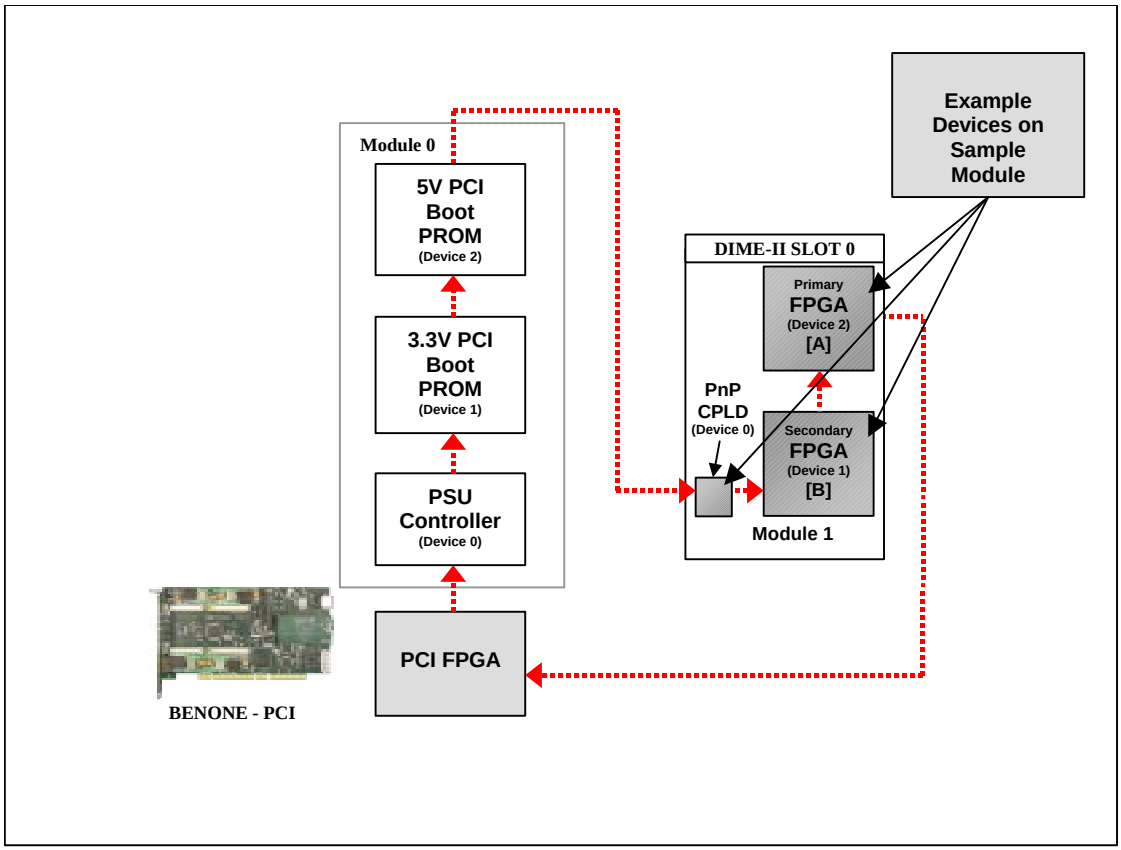



**Figure 15: Diagram for FPGA Configuration Example 1**

| FPGA Device | Module ID Number | Device ID Number |
|---|---|---|
| A | 1 | 2 |
| B | 1 | 1 |

**Table 3: Example configuration 1 module and device ID numbering**

The code, written in C, for the example configuration described above is illustrated as follows in Figure 16:

```
#include "dimesdl.h" //contains the API functions

//Declare variables
DWORD Status1, Status2, Status3, Status4;
DIME_HANDLE hBenONE;
LOCATE_HANDLE hLocate;

if ((hLocate =
DIME_LocateCard(dlPCI,mbtTHEBENONE,NULL,dldrDEFAULT,dlDEFAULT))==NULL)
{
        //Error: Could not locate the BenONE - PCI
        return (1);    //Exit the app
}

if ((hBenONE=DIME_OpenCard(hLocate,1,dccOPEN_DEFAULT))==NULL)
{
        //Error: Could not open the BenONE - PCI
        return (1);    //Exit the app
}

//Boot the secondary FPGA (device 1) on Module 1 with the bit file "BitfileC.bit"
Status1 = DIME_ConfigDevice(hBenONE,"BitfileC.bit",1,1,NULL,0);

//Boot the primary FPGA (device 2) on Module 1 with the bit file "Bitfile.bit"
Status2 = DIME_ConfigDevice(hBenONE,"Bitfile.bit",1,2,NULL,0);

DIME_CloseCard(hBenONE);
DIME_CloseLocate(hLocate);
```

**Figure 16: Example C code to configure FPGAs in 'Example Configuration 2'**

### 5.2.4 FPGA Configuration using FUSE Probe

The FUSE GUI Application is an easy to use software interface, which allows users to access a subset of the functionality provided by FUSE.

Full instructions on how to use the GUI are provided in the FUSE System Software User Guide.

### 5.2.5 FPGA Configuration using DIMEScript

DIMEScript is a high-level scripting language, which provides users with a simple and easy to use language for the configuration and control of DIME systems. DIMEScript uses a simple command set, eliminating the need for developers to use complicated programming interfaces to control and communicate with application designs running in FPGAs. DIMEScript also offers platform portability through ASCII based scripts, allowing users to use DIMEScript on both Windows and Linux installations.

DIMEScript can be used either to write script files, which can then be executed as a single process, or it can be used from a command line interface, with the user executing commands as required.

Full instructions on how to use the DIMEScript are provided in the DIMEScript User Guide, which is on the XtremeDSP Development Kit CD.

## 5.3 FPGA Configuration using the FUSE APIs

The FUSE Software development API enables users to call functions to control DIME hardware in their own programs. This allows users to develop software applications to complement the FPGA application designs running on DIME hardware.

FUSE APIs are available to support a number of development languages, including C, C++, Matlab and Java. Full instructions on how to use the APIs are provided in the relevant FUSE User Guide.

## 5.4 FPGA Configuration using DIME JTAG Chain

The DIME JTAG chain is the principal JTAG chain on the BenONE - PCI and facilitates the configuration of the FPGAs on the DIME-II module hosted on the BenONE - PCI. The DIME JTAG Chain is driven by the PCI FPGA, or can be driven from the ALT JTAG pin header on the BenONE - PCI, via a JTAG programmer, such as the Xilinx Parallel-III programmer. Therefore this is where you would connect the Parallel-III or IV cables in order to use products such as ChipscopeILA.

The DIME JTAG chain has built-in switches, which only switch a DIME-II module slot into the chain, if a module is populated, therefore if a module is not populated in a slot, the chain skips that slot.

If an external power supply is required for a programmer, a 3.3 V supply is available on the DIME JTAG Connector. A maximum supply of 3.3 V should be used to supply a configuration device, as higher voltages will irreparably damage the Virtex-E device populated as the User FPGA or on modules.

The DIME JTAG chain connected to the DIME JTAG Header is the same as that used by the FUSE software during FPGA configuration. If FUSE is being used, a JTAG controller cannot be used during FPGA configuration using the software.

The DIME JTAG chain configuration is shown in Figure 17, the header to access this chain is shown in Figure 18 and the header pinout is detailed in Table 4.

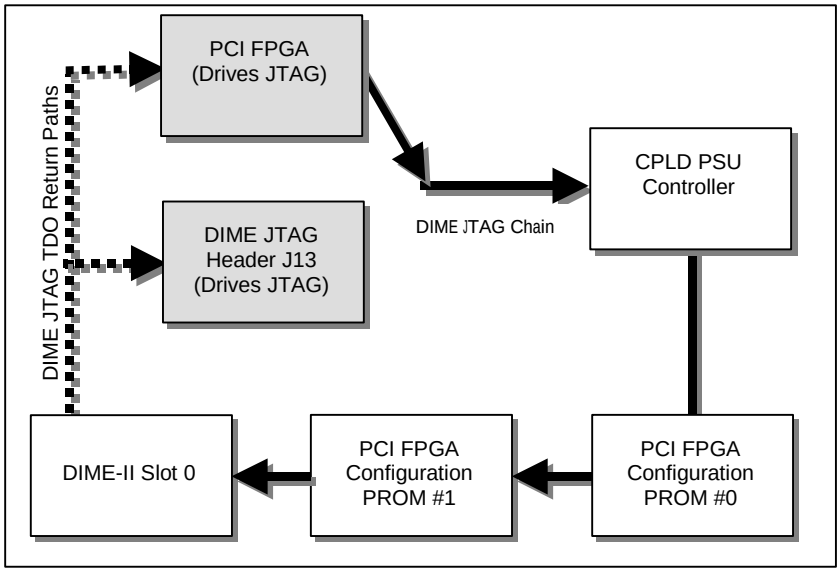



**Figure 17: DIME JTAG Chain**

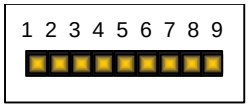



**Figure 18: ALT JTAG Connector J13**

| Pin # | Name | Description |
|---|---|---|
| 1 | 3.3V | 3.3 Volts Supply |
| 2 | GND | Signal Ground |
| 3 | N/C | Not connected – do not use |
| 4 | TCK | ALT JTAG TCK Signal |

| Pin # | Name | Description |
|---|---|---|
| 5 | N/C | Not connected – do not use |
| 6 | TDO | ALT JTAG TDO Signal |
| 7 | TDI | ALT JTAG TDI Signal |
| 8 | TRST# | ALT JTAG TRST# Signal |
| 9 | TMS | ALT JTAG TMS Signal |

**Table 4: ALT JTAG header pinouts**

Care must be taken when using this method of programming your module. Inadvertent programming of either the PSU Controller or PCI Boot proms on your BenONE - PCI could render it inoperable or in extreme cases damage to your module could occur.

## 5.5 FPGA Configuration using uP JTAG Chain

The uP JTAG chain facilitates the configuration of devices on DIME-II modules hosted on the BenONE - PCI. It should be noted that not all DIME-II modules will necessarily support the uP JTAG chain – please refer to the module User Guide for details. The uP JTAG Chain is driven by the uP JTAG pin header on the BenONE - PCI.

The uP JTAG chain has built-in switches, which only switches a DIME-II module slot into the chain if a module is populated. Therefore if a module is not populated in a slot, the chain skips that slot.

The uP JTAG Connector is configured to match the pinout of the ADSP-21160 EZ-ICE Emulator, which utilises the IEEE 1149.1 JTAG test access port of the ADSP-21160 to monitor and control the target board processor during emulation. The EZ-ICE probe requires the ADSP-21160's CLKIN, TMS, TCK, TRST, TDI, TDO, EMU, and GND signals be made accessible on the target system via a 14-pin connector (a 2 row × 7 pin strip header).

The uP JTAG chain configuration is shown in Figure 19, the frontplate connector to access this chain is shown in Figure 20 and the pinout for the connector is detailed in Table 5.

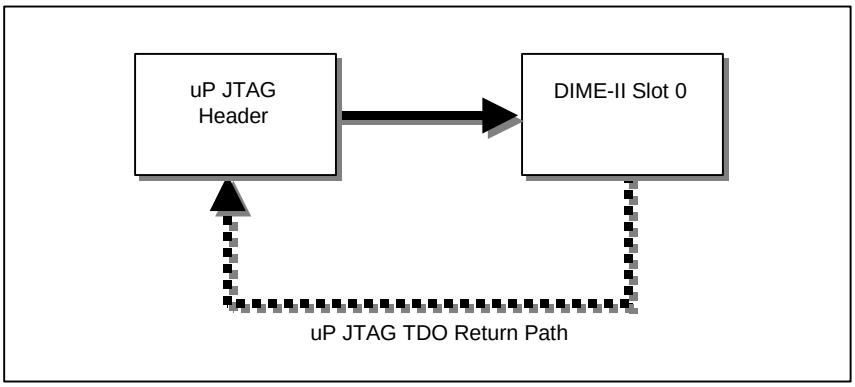



**Figure 19: DIME JTAG Chain**

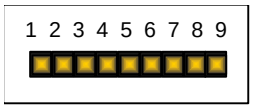



**Figure 20: uP JTAG Connector**

The reason for the set-up is so that the software cannot drive the JTAG chain at the same time as a separate program using a download cable connected to the external JTAG header.

| Pin # | Name | Description |
|---|---|---|
| 1 | 3.3V | 3.3 Volts Supply |
| 2 | GND | Signal Ground |
| 3 | N/C | Not connected – do not use |
| 4 | TCK | DIME JTAG TCK Signal |
| 5 | N/C | Not connected – do not use |
| 6 | TDO | DIME JTAG TDO Signal |
| 7 | TDI | DIME JTAG TDI Signal |
| 8 | TRST# | DIME JTAG TRST# Signal |
| 9 | TMS | DIME JTAG TMS Signal |

**Table 5: uP JTAG Connector pinouts**

# Clock Configuration

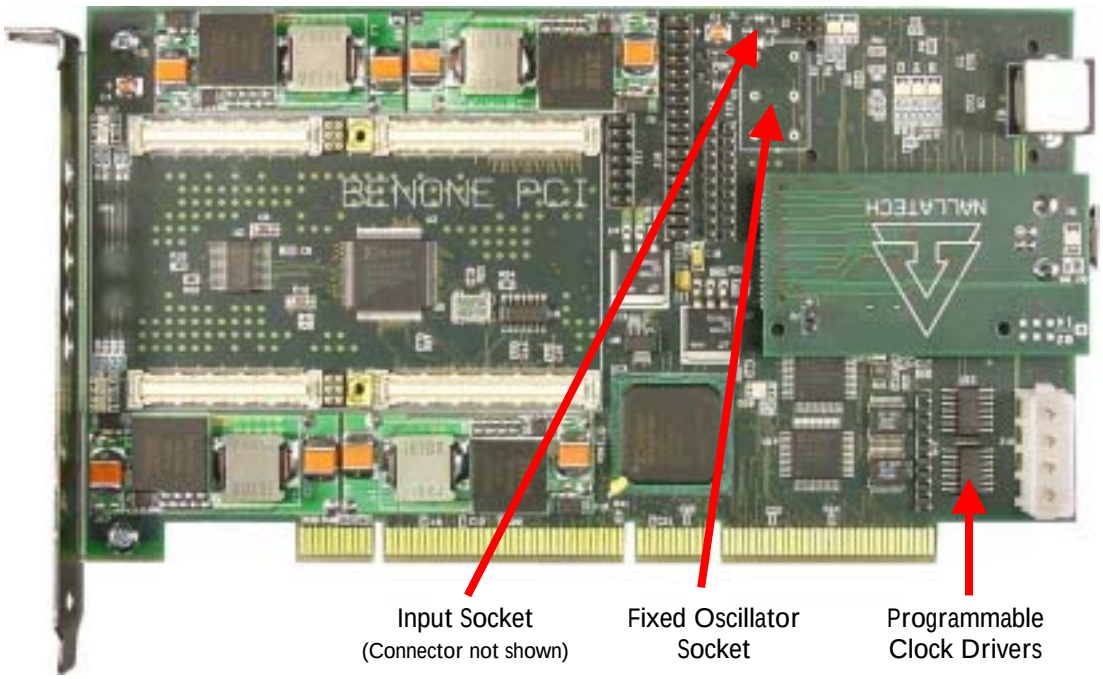
- BenONE PCI clocking description
- Configuring clock sources
- Using external clocks
- System level clocking

## 5.6 General Description

The BenONE - PCI has a comprehensive and flexible clock management system. The features available are as follows:

- Two on-board clocks for general use
- Two on-board programmable clock sources
- Single Fixed oscillator socket on-board clock source
- User clock input connector for on-board use
- The PCI clock

The clock nets on the BenONE - PCI have been designed to eliminate clock skew at the FPGA destination, using clock nets of the same length between the on-board clocks and the DIME-II module slot.




## 5.7 On-board clocks

The BenONE - PCI has two clocks (A and B), which can be used throughout the board. The fixed oscillator socket is fitted with a 50MHz unit.

## 5.8 Source Descriptions

### Programmable Oscillators

The Programmable Oscillators are controlled via FUSE Software, through any of the available interfaces/APIs. The available operating frequencies of the programmable oscillators is as follows:

20 MHz; 25 MHz; 30 MHz; 33.33 MHz; 40 MHz; 45 MHz; 50 MHz; 60 MHz; 66.66 MHz; 70 MHz; 75 MHz; 80 MHz; 90 MHz; 100 MHz; 120 MHz.

When a frequency is requested using FUSE, which does not exactly match one of the fifteen frequencies supported by the oscillators, the firmware will select the frequency from the available frequencies, which is numerically closest to the frequency requested.

### External Clock Source

External clock sources can be brought onto the BenONE - PCI for on-board use. Please refer to Section 5.9 Using external clocks on page 40for further details.

### Software Derived Clock

The Software Derived Clock can be used for debugging and is controlled from the FUSE APIs. Please refer to the FUSE System Software User Guide for more details**.** Note that this facility it not currently available in the firmware until a future firmware release.

### Fixed Oscillator

A 50MHz fixed oscillator is fitted to the XtremeDSP Development Kit, providing a high-accuracy clock source.

## 5.9 Using external clocks

The XtremeDSP Development Kit is provided with Clock Output functionality, enabling on-board clocks (A, B & C) to be outputted to a pin header, for use externally. Also provided is the Clock Input functionality, enabling external clocks to be inputted onto the XtremeDSP Development Kit from the clock input connector, for on-board use.

The XtremeDSP Development Kit will allow either a fixed frequency oscillator to be used or an external clock, as the external clock connector is located under the fixed frequency oscillator socket.

# System Level Design

- Design Partitioning
- DIME-II Communication Bus Speeds

## 5.10 Design Partitioning

The XtremeDSP Development Kit allows the user to partition the functionality of their application between software and hardware easily and effectively. Below, the design partitioning of the XtremeDSP Development Kit is shown in Figure 21:

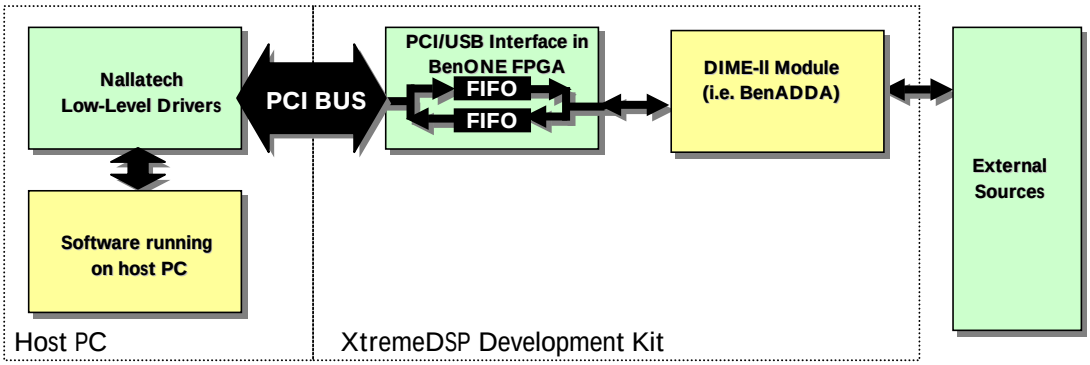



**Figure 21: XtremeDSP Development Kit Design Partitioning**

The green blocks do not require any further design from a user perspective. The PCI interface is pre-configured and the external sources are assumed to be in place. The user needs to design application designs for any FPGAs on the hosted DIME-II module. The software running on the host PC is available as a pre-designed GUI. For users who require additional functionality and wish to have their own software front-end, the DIME Software Library provides functions for use in application programs. These functions include FPGA configuration/reset, clock speed setting and data transfer.

| Interface COMM Signal | General Bus Signal Name | Dime-II Connector PIN No |
|---|---|---|
| ADIO<0> | LBUS<0> | PB1 |
| ADIO<1> | LBUS<1> | PB2 |
| ADIO<2> | LBUS<2> | PB3 |
| ADIO<3> | LBUS<3> | PB4 |
| ADIO<4> | LBUS<4> | PB6 |
| ADIO<5> | LBUS<5> | PB7 |
| ADIO<6> | LBUS<6> | PB8 |
| ADIO<7> | LBUS<7> | PB9 |
| ADIO<8> | LBUS<8> | PB10 |
| ADIO<9> | LBUS<9> | PB11 |
| ADIO<10> | LBUS<10> | PB12 |
| ADIO<11> | LBUS<11> | PB13 |

| Interface COMM Signal | General Bus Signal Name | Dime-II Connector PIN No |
|---|---|---|
| ADIO<12> | LBUS<12> | PB15 |
| ADIO<13> | LBUS<13> | PB16 |
| ADIO<14> | LBUS<14> | PB17 |
| ADIO<15> | LBUS<15> | PB18 |
| ADIO<16> | LBUS<16> | PB19 |
| ADIO<17> | LBUS<17> | PB20 |
| ADIO<18> | LBUS<18> | PB21 |
| ADIO<19> | LBUS<19> | PB22 |
| ADIO<20> | LBUS<20> | PB24 |
| ADIO<21> | LBUS<21> | PB25 |
| ADIO<22> | LBUS<22> | PB26 |
| ADIO<23> | LBUS<23> | PB27 |
| ADIO<24> | LBUS<24> | PB28 |
| ADIO<25> | LBUS<25> | PB29 |
| ADIO<26> | LBUS<26> | PB30 |
| ADIO<27> | LBUS<27> | PB31 |
| ADIO<28> | LBUS<28> | PB33 |
| ADIO<29> | LBUS<29> | PB34 |
| ADIO<30> | LBUS<30> | PB35 |
| ADIO<31> | LBUS<31> | PB36 |
| BUSY | ADJOUT<0> | PD29 |
| EMPTY | ADJOUT<1> | PD30 |
| RDI_WR | ADJOUT<2> | PD31 |
| AS_DSI | ADJOUT<3> | PD32 |
| RENI_WENI | ADJOUT<4> | PD33 |
| INTI | ADJOUT<5> | PD34 |
| RSTI | ADJOUT<6> | PD35 |

**Table 6 - Interface FPGA to DIME-II Slot**

Also DSP_CLK should be connected to CLK1(sometimes referred to as CLKB)

| Interface COMM Signal | General Bus Signal Name | Dime-II Connector PIN No |
|---|---|---|
| DSP_CLK | CLK1 | PC31 |

**Table 7 - Interface Clock to DIME-II**

## 5.11 Interface Communications Bus

The Interface Comms Bus is an important communications channel, as it provides a path for data communication between the User FPGA, the Interface (PCI or USB) FPGA and onto the host PC.

The Interface Comms bus has a pre-defined communications protocol to facilitate communications to the Interface FPGA. In order to communicate with the Interface FPGA from the User FPGA, the User FPGA application design must incorporate a mechanism to communicate over this bus. This communications mechanism can be implemented directly by the user in the design, or by using Nallatech's drop-in IP core - the PCI to User FPGA Interface Core. This core implements the Interface to User FPGA Comms communications mechanism and offers the user a simplified interface to which they can connect their own designs.

### 5.11.1 Interface to User FPGA Interface Core

The Interface to User FPGA Interface Core is a drop-in IP core, which can be incorporated into the User FPGA Application Design. This core implements a mechanism, which deals with the protocol to communicate over the Interface Comms Bus. This abstracts the complexities of the protocol and provides a simplified user interface, offering a memory-mapped address space for registers/peripherals and DMA channels for high-speed data transfer. A block diagram for the implementation of this core is shown below in Figure 22:

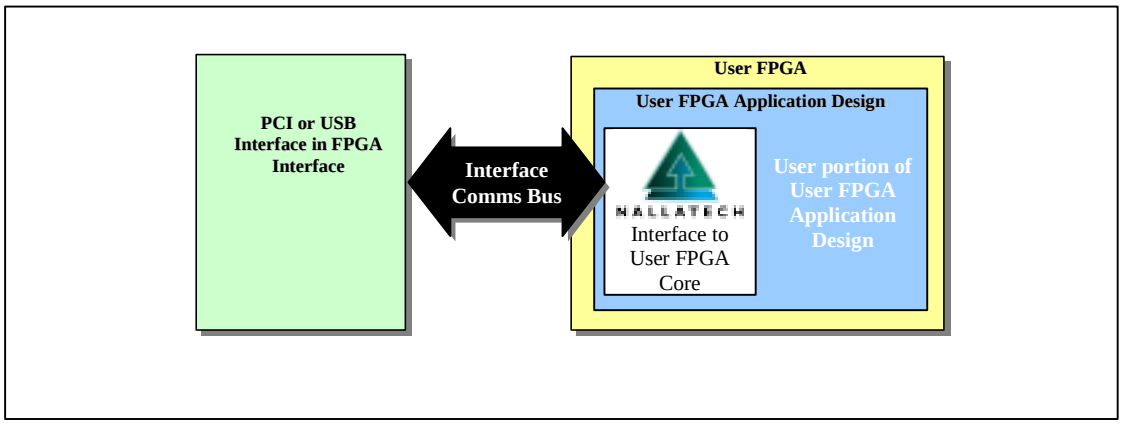



**Figure 22: Implementation with PCI or USB to User FPGA Interface Core**

For full details of the Interface to User FPGA Interface Core, please refer to Application Note NT302-0000, which is available on the supplied FUSE CD. The necessary VHDL code and an EDIF file for the core are also provided on the CD. Please note that this is a generic Application Note whether PCI or USB interfacing is used.

### 5.11.2 Implementing the Comms Communications Mechanism

Instead of communicating over the Interface Comms bus using the Interface to User FPGA Interface Core, a mechanism to communicate over this bus using the appropriate protocol can be implemented directly by the user in their design. A block diagram for the implementation of this core is shown in Figure 23:

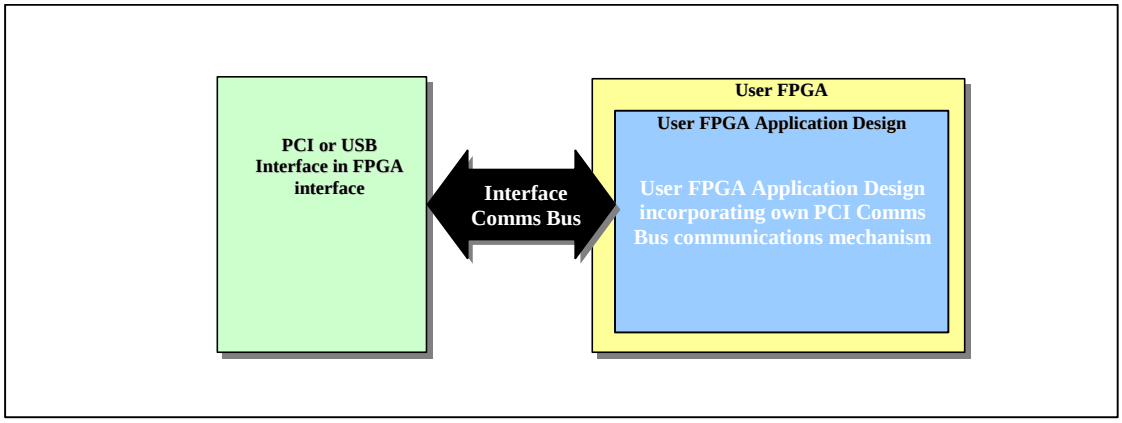



**Figure 23: Implementation with own communications mechanism**

### 5.11.3 Comms Bus Protocol

The information in this section only applies to users who wish to implement the Comms communications protocol directly in their own FPGA applications design, without using the Nallatech Interface to User FPGA Interface Core.

Data is transferred between the User FPGA and the Interface FPGA with the use of 5 control signals. Three of these signals are driven by the PCI or USB interface. These are AS/DS#, EMPTY and BUSY.

**AS/DS#**: This is an address strobe/data strobe signal. When this is HIGH, the data being transferred to the User FPGA is an address and when this signal is LOW, the data being transferred to the User FPGA is data from/to the last address given. Generally, addresses are only sent from the Interface FPGA to the User FPGA.

This signal is always in sync with the data passed through the internal FIFOs of the Interface FPGA so that the internal FIFO can have a mixture of actual data and addresses. This AS/DS# line will automatically indicate the true type of data.

**EMPTY**: This signal indicates that there is data waiting to be written from the PCI or USB FPGA interface to the User FPGA. This signal will go HIGH when there is no more data to be written to the User FPGA.

**BUSY**: This signal indicates that the PCI or USB FPGA interface can receive data from the User FPGA. When this signal goes HIGH, no more data should be written to the Interface FPGA.

The User FPGA drives the two remaining control signals. These are R#/W and REN#/WEN#.

**R#/W**: This signal determines the direction of the data transferred between the Interface FPGA and the User FPGA. If this signal is LOW, data is being read from the Interface FPGA and so the Interface FPGA drives the data bus. If the signal is HIGH, data is being written to the Interface FPGA and so the User FPGA drives the data bus.

**REN#/WEN#**: This signal is a read/write enable signal. When this signal is LOW, if the R#/W signal is HIGH, data is on the bus ready to be written to the Interface FPGA. If the signal is LOW, and R#/W is LOW, then data will be driven onto the data bus from the Interface FPGA on the next clock edge. When this signal is HIGH, there should be no data on the bus.

Lastly, there is the data bus that is used to transfer data between the Interface FPGA and the User FPGA. This bus is called **ADIO** and is a 32-bit bi-directional bus that can be driven by both the Interface FPGA and the User FPGA.

The general functionality is similar to that of FIFOs. The EMPTY and BUSY signals act similarly to FIFO_EMPTY and FIFO_FULL signals. The R#/W and REN#/WEN# signals combine to give the REN# and WEN# signals of a FIFO.

The clock used for the Interface FPGA to User FPGA communications is always DSPCLK.

#### Reading from Interface FPGA to User FPGA

Reading from the Interface FPGA is similar to reading from a FIFO. The EMPTY signal goes LOW to indicate that there's data to be read. The FIFO, whose data is read from on the Interface FPGA is a First-Word-Fall-Through with a latency of one clock cycle. When reading from the Interface FPGA the user must ensure that the read enable is not active until at least one clock after the EMPTY signal goes LOW. The read enable should go inactive immediately after the EMPTY signal goes HIGH, although no data will be read if EMPTY is HIGH and the read enable is active.

The diagram in Figure 24 shows a functional representation of data reads in operation. These are single word transfers.

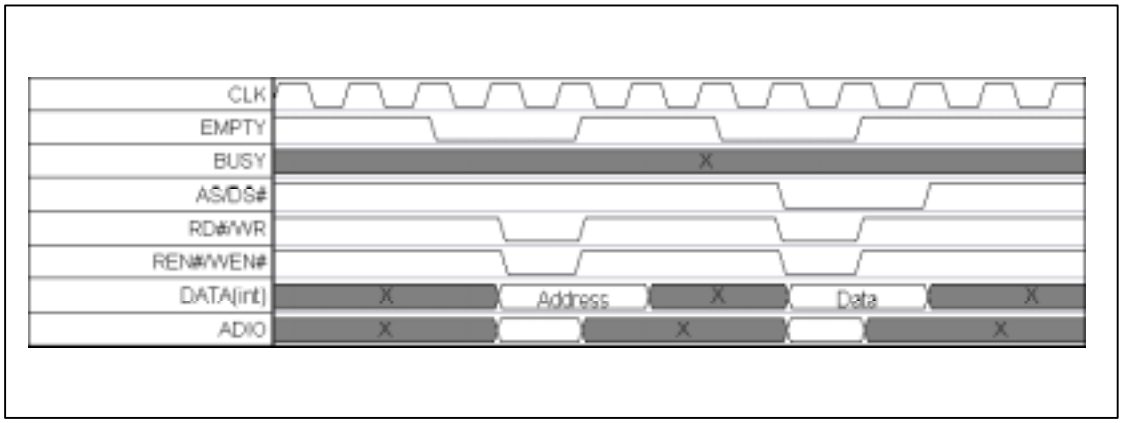



**Figure 24: Single Read Transfer**

The first read sends an address from the Interface FPGA to the User FPGA. The second read sends the data across. The DATA(int) bus shows the internal data waiting to be driven onto the ADIO bus.

The diagram in Figure 25 shows a burst-read in operation.

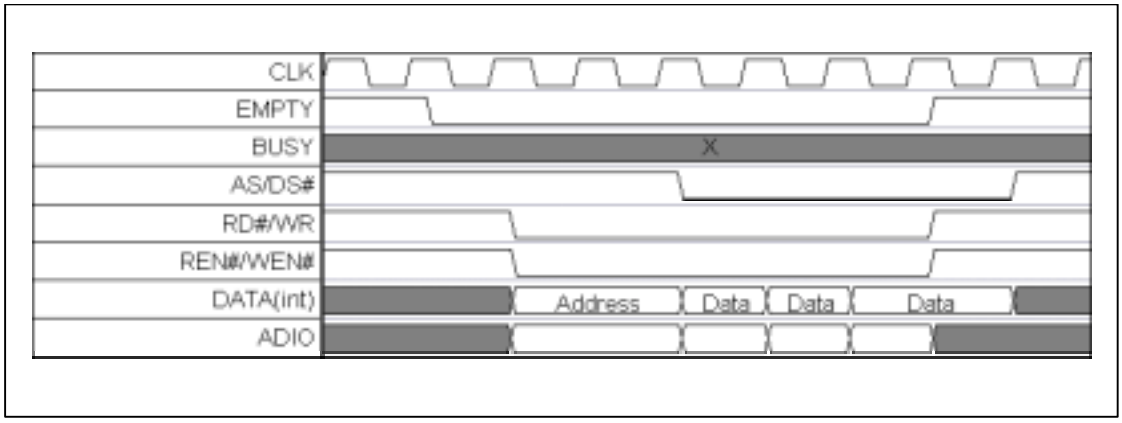



**Figure 25: Burst Read Transfer**

This example shows a single address being sent from the Interface FPGA to the User FPGA followed by a burst of data. Again the DATA(int) bus shows the internal data waiting to be driven onto the ADIO bus. It should be noted that after the first read, the next set of data will not be available until one clock cycle after the read. So long as the reads are continuous, data will then follow every clock cycle.

## Writing to Interface FPGA from User FPGA

Writing to the Interface FPGA is similar to writing to a FIFO. If the Interface FPGA can receive more data, the BUSY signal is LOW and when the Interface FPGA cannot receive any more data, the BUSY signal is HIGH. To help meet timing specifications the User FPGA application is allowed to over-run by two further data samples. In other words, after the BUSY signal is asserted two further data samples can be written to the Interface FPGA.

The diagram in Figure 26 shows a Burst-Write function in operation and also demonstrates the maximum data over-run of 2.

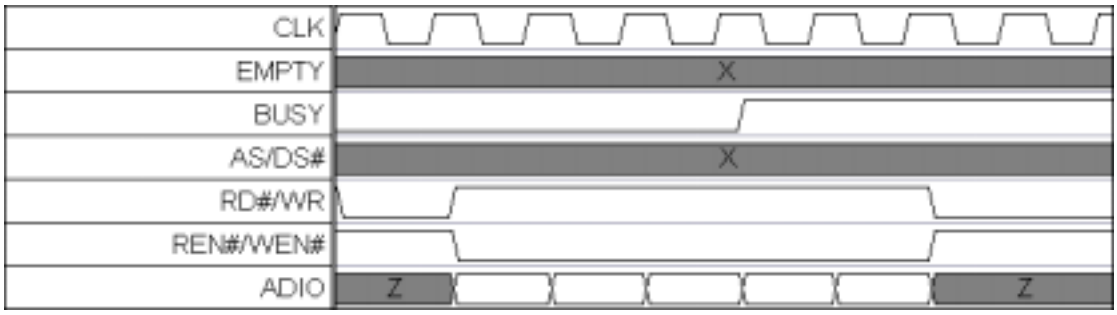



**Figure 26: Burst Write Transfer**

## Timing Information

The information in the table below provides the timing information required to write the User's Constraints File (UCF) for implementing a design into an FPGA.

| Signal | In/Out | Details |
|---|---|---|
| EMPTY | In | 9ns before clock |
| BUSY | In | 9ns before clock |
| AS/DS# | In | 19ns before clock |
| RD/WR# | Out | 11ns before clock |
| REN#/WEN# | Out | 11ns before clock |
| ADIO | In | 5ns before clock |
| ADIO | Out | 12.5ns before clock |

**Table 8 – Timing Information**

Timing specifications should be written into the UCF as a NET <name> OFFSET = <spec> specification ( in other words *NET EMPTY OFFSET = 9ns BEFORE DSPCLK;*).

## Other signals

There are two other signals between the User FPGA and the Interface FPGA. These are RST# and INT#.

RST#: This signal is driven by the PCI or USB FPGA Interface and can be used as a global reset within the User FPGA device.

INT#: This signal is driven by the User FPGA and can be used to signal an interrupt to the Interface FPGA and cause a PCI interrupt.

## Interface FPGA to User FPGA Communications From A Software Perspective

When writing software for the XtremeDSP Development Kit, some considerations should be made with regards to how this affects the Interface FPGA to User FPGA communications.

### DataRead/DataReadSingle

The DIME_DataRead and DIME_DataReadSingle commands are used to read data from the BenONE - PCI. These commands read data from the internal FIFOs of the Interface FPGA. For this reason, the User FPGA must send data to the Interface FPGA before it can be read.

### DataWrite/DataWriteSingle

The DIME_DataWrite and DIME_DataWriteSingle commands are used to write data to the BenONE - PCI. These commands write data to the internal FIFOs but they also write to the buffer that controls

the AS/DS# bit and ensures that this bit will be LOW. When data is written, the EMPTY signal will go LOW.

**AddressWriteSingle**

The DIME_AddressWriteSingle command writes a single address to the Interface FPGA. This is written into the internal buffers along with a HIGH bit for the AS/DS#. Again when the address is written, the EMPTY signal will go LOW to indicate that data is available.

**Interface Core Design**

Nallatech has developed an interface core design to be used to control the Interface FPGA to User FPGA communications. This application note (PCI to User FPGA Interface) can be found on the supplied FUSE CD.

**For further details on:**

- FUSE Development API, please refer to the FUSE C-C++ Developers Guide provided on the supplied FUSE CD.

## 5.12 DIME-II Communication Bus Speeds

The BenONE on the XtremeDSP Development Kit has a variety of different types of buses that allow the Interface (PCI/USB) FPGA and the DIME-II module slot to communicate with each other. There are three main communication bus types:

- Parallel Link
- Adjacent Bus
- Local Bus

These three buses permit data to be transferred to all the various devices being used in the DIME-II system. Below is a brief introduction to these buses.

The Parallel Link (P-Link) bus is a 12-bit bi-directional point-to-point bus, which will allow the DIME-II module to directly communicate with external data. Some of the bits of this bus are used to provide a software interface communication mechanism as described in Section 5.10.

There are two types of adjacent buses available on the XtremeDSP Development Kit: the Adjacent In Bus and Adjacent Out Bus. These buses are designed to facilitate the use of Pipelined architectures where the resultant data processed on one DIME-II module can be passed to the next module for further processing. The Adjacent In Bus is 28 bits wide and the Adjacent Out Bus is 40 bits wide.

The Local Bus is coupled from the Interface FPGA and the DIME-II module. The Local Bus is a 64-bit wide bus and may be used as either an overall system control or broadcast bus. Such a bus can typically be utilised to memory map internal registers and memory space in the FPGA into microprocessor memory space. Some of the bits of this bus are used to provide a software interface communication mechanism as described in Section 5.10.

| DIME-II Bus | Typical Max Frequency | Description |
|---|---|---|
| P-Link Bus | 125 MHz / 200MHz | A 12-bit point-to-point communications bus. When a Source Synchronous data transfer is employed a maximum frequency of 200 MHz may be achieved otherwise the maximum frequency possible is 125 MHz. |
| Adjacent Bus | 150 MHz | Due to the small transmission distance between adjacent buses, a maximum frequency of 200 MHz may be achieved. |
| Local Bus | 66 MHz | This bus is common to and distributes to all DIME-II modules and the User FPGA. Speeds of 66 MHz may be realised. |

**Table 9 – Bus Speeds**

# Section 6

# Reference Guide

In this Section:

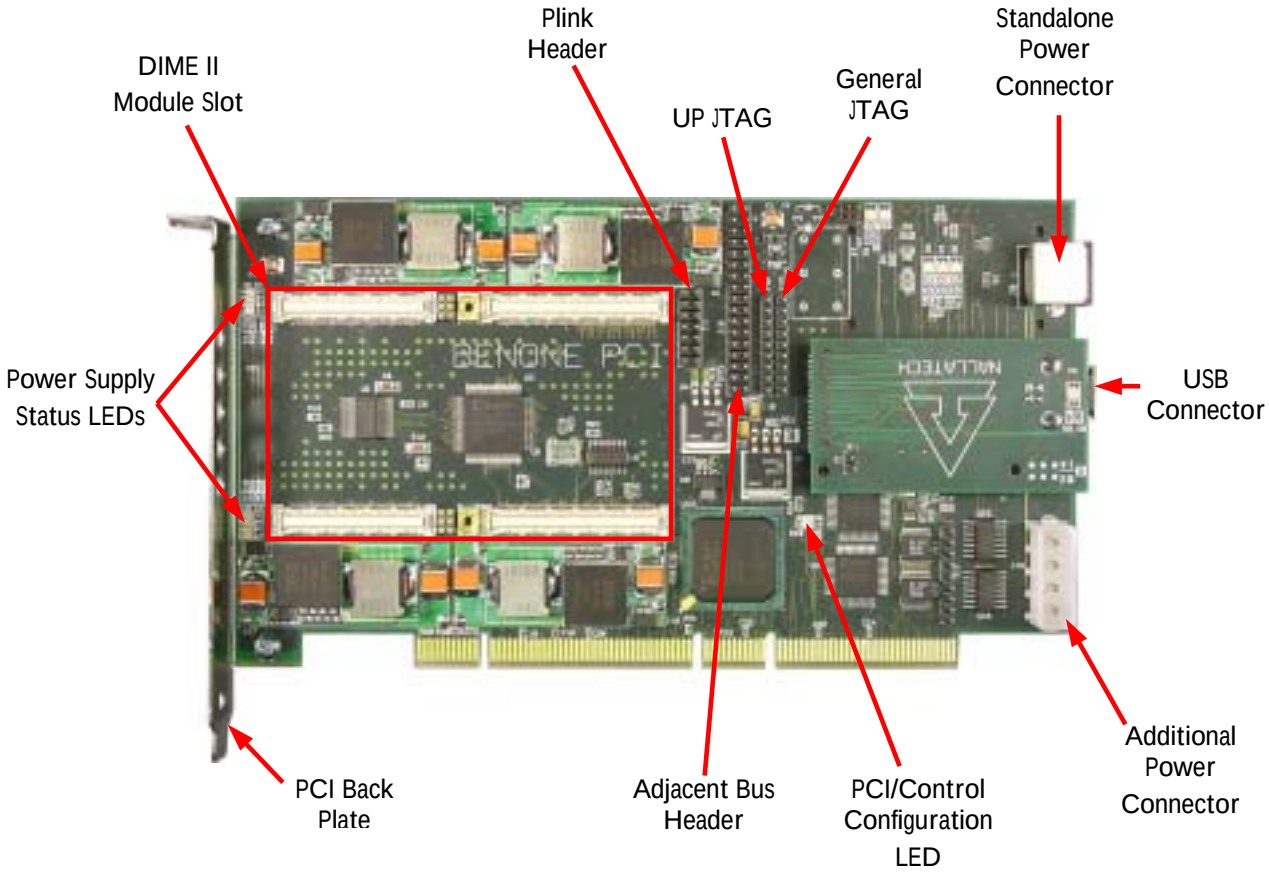
- Physical Layout
- Hardware Features
- Build Options and Jumper Settings
- Pin-out Information

# Physical Layout

In this Section:

- Physical features of the BenONE in the XtremeDSP Development Kit are identified

## 6.1 BenONE – PCI Physical Layout (front)



# Hardware Features

- BenONE – PCI Power Specifications
- BenONE – PCI Clock Circuit
- Resets and LEDs

## 6.2 BenONE – PCI Power Specifications

Power specifications on the BenONE – PCI will vary depending on the type of modular power supply that has been populated to power the DIME-II module slot, these options are discussed further in the following Sections. Several events happen when power is applied to the card. Firstly, each modular power supply is interrogated by the PCI FPGA to ascertain its output characteristics. Next the module is interrogated to ascertain its power requirements. All requirements have to be met before power is switched on to the module, this prevents inadvertent damage to any module populated on the BenONE - PCI.

### 6.2.1 Modular fixed power supply (FPS)

Nallatech offer the fixed power supply option, this has reduced output current capability and produces a fixed voltage output. This option reduces the flexibility of the BenONE – PCI card, as only DIME-II modules with the same power supply requirements will function. Power supplies are configured at the time of manufacture, voltages in the range of 1V to 3.3V are available with a maximum output current of 7A(abs. max). This is the power supply that is fitted as standard in the XtremeDSP Development kit.

Even though the unit is capable of supplying 7A(abs. max), the actual output voltage, and more importantly the drop from the input to the output, will limit to operating range of the unit due to its ability to dissipate the heat generated by the conversion. The following calculations can be used to calculate the max power that can be drawn.

There for a FPS that is supplying 1.5V to the system, under the conditions of an ambient temperature (TA) of 25°C and a thermal coefficient of 35°C/W:

$$P_{MAX} = \frac{T_J - T_A}{\theta_{JA}}$$

$$\Rightarrow T_J = T_A + (\theta_{JA} x P_{MAX}), where\, \theta_{JA} = 35^{\circ}C/W, T_J = 125^{\circ}C(\max), T_A = 25^{\circ}C$$

$$\Rightarrow 125 = 25 + (35 x P_{MAX})$$

$$\Rightarrow \underline{\underline{P_{MAX} = 2.857W}}$$

$$P_{MAX} = Iout_{MAX} x(V_{IN} - V_{OUT})$$

$$2.857 = Iout_{MAX} x(5 - 1.5)$$

$$\Rightarrow Iout_{MAX} = 816.33mA$$

For a FPS that is supplying 3.3V to the system:

$$P_{MAX} = \frac{T_J - T_A}{\theta_{JA}}$$

$$\Rightarrow T_J = T_A + (\theta_{JA} x P_{MAX}), where \theta_{JA} = 35^{\circ}C/W, T_J = 125^{\circ}C(\max), T_A = 25^{\circ}C$$

$$\Rightarrow 125 = 25 + (35 x P_{MAX})$$

$$\Rightarrow \underline{\underline{P_{MAX} = 2.857W}}$$

$$P_{MAX} = Iout_{MAX} x(V_{IN} - V_{OUT})$$

$$2.857 = Iout_{MAX} x(5 - 3.3)$$

$$\Rightarrow Iout_{MAX} = 1.681A$$

As part of the characterisation process it has been proved that a design targeted at an XC2V3000 using 80% of the chip, with a 15% toggle rate can be run 65MHz whilst remaining within the capability of the FPS units. For significantly higher power requirements, PPS units should be considered due to their higher power capabilities. Please contact Nallatech for details of the PPS units and upgrade options.

These power factors should be taken into account when designing the system using the BenONE with associated DIME-II modules. It may be necessary to make use of the PPS modules rather than the FPS option for the higher power situations.

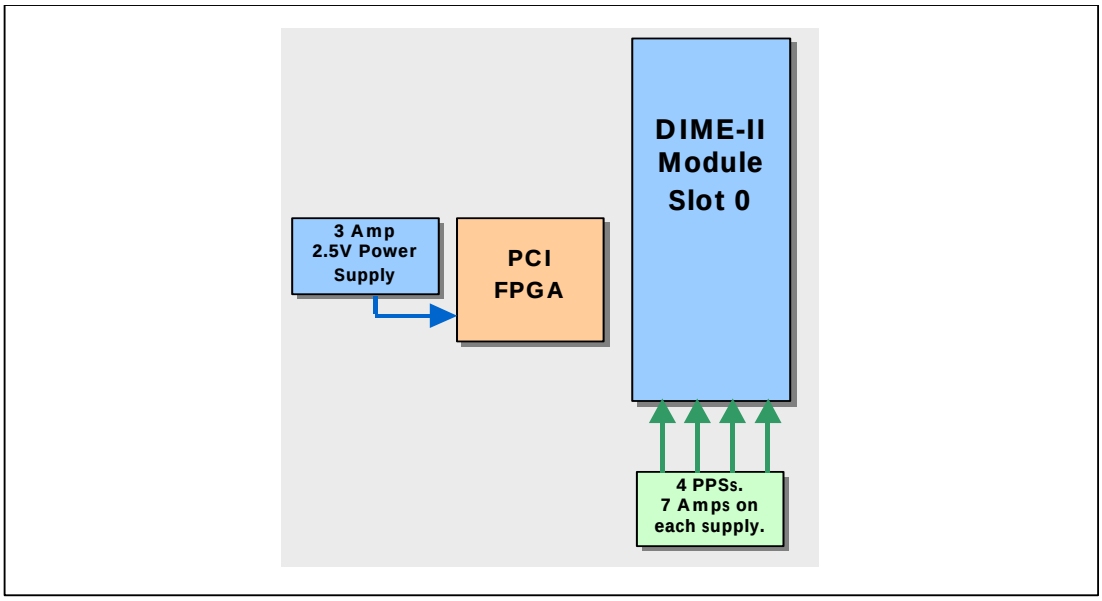



**Figure 27: Programmable Power Supplies on the BenONE - PCI**

4 LEDs on the BenONE – PCI indicate the status of each power supply unit. They will only illuminate when the slot is populated with a module, the BenONE – PCI card has been opened by the host system and power to the module slot is switched on. Please note that if a power supply is not used then the power indicator for that supply will remain RED indicating it is not switched on. For example, on the BenADDA only three supplies are required and so only three of the four LEDs will change from red to green

The power requirements of the FPGA devices on the attached module will depend upon the density and speed of the application design running in them. The Xilinx Power Estimator can be used to estimate the power requirements of application designs.

Xilinx Power Estimator: http://support.xilinx.com/support/techsup/powerest/index.htm...

### 6.2.2 Modular programmable power supply (PPS)

Using on-board DC-DC converters, dynamic adjustment of the voltages applied to the module slot is possible, reacting as required for different modules within the DIME-II range. Each supply is capable of supplying voltages in the range 1V to 3.3V at a maximum current of 15A(abs. max).

The PCI FPGA has a dedicated power supply fixed at 2.5V.

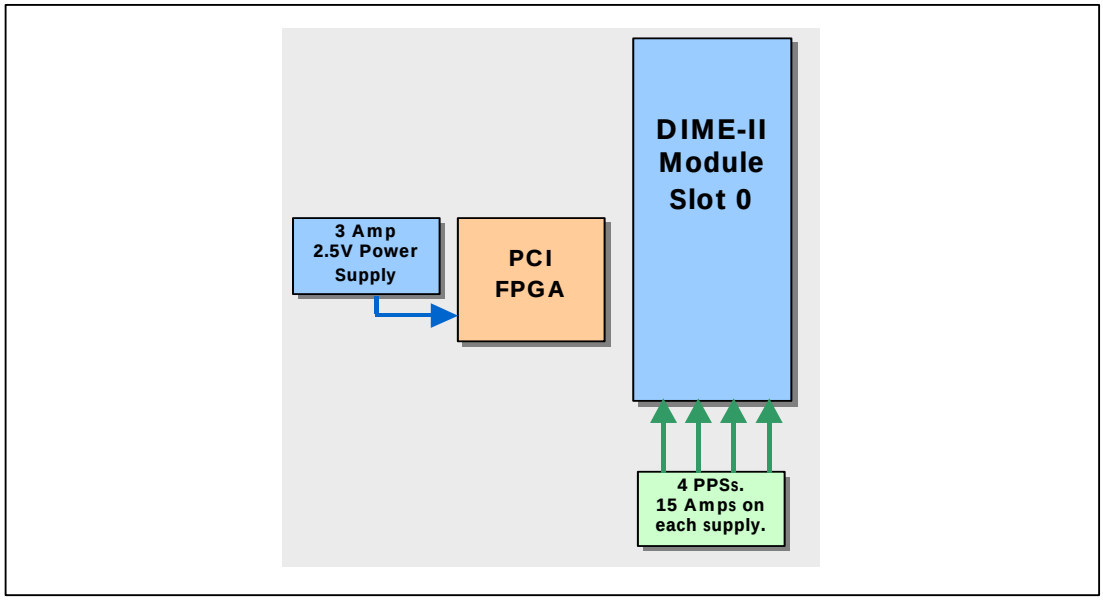



**Figure 28: Programmable Power Supplies on the BenONE - PCI**

It should be noted that no user intervention is required to set up the PPSs, these will be automatically configured by the system.

The PPS that supplies the core voltage to the FPGA can supply up to 15A on any of the core FPGA power supplies. In this situation consideration of system power supplying the PPP's should be made. Additionally the end-user should consider the thermal capability of the FPGA. Please refer to appropriate Xilinx resources.

## 6.3 Clock Circuit

The following diagram provides a basic overview of the BenONE clock circuit:

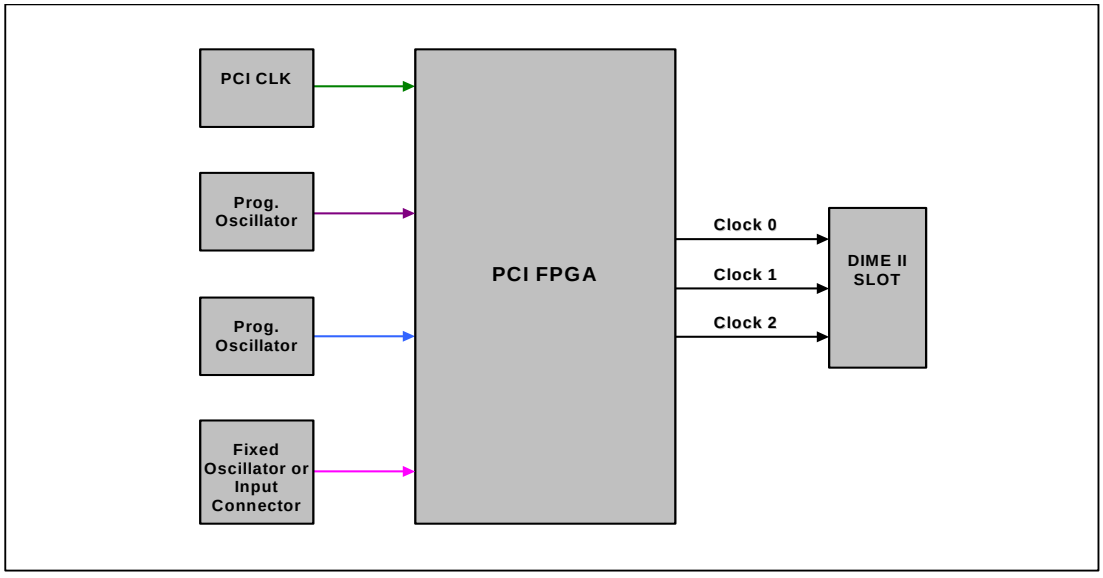



**Figure 29: BenONE – PCI Clock Circuit**

## 6.4 Reset and LEDs

### 6.4.1 Reset

The configuration of the reset on the BenONE - PCI is shown Figure 30:

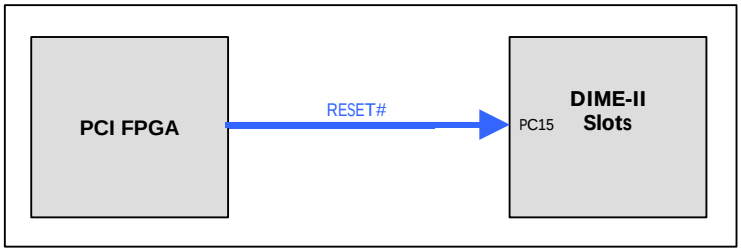



**Figure 30: BenONE - PCI Reset Circuit**

### 6.4.2 PCI FPGA Configuration LED

LED D10 will illuminate when the PCI FPGA is fully configured.

### 6.4.3 DIME-II Module Bulk Supply LEDs

Each of the bulk power supplies has a single tri-colour LED showing the status of the power supply.

| Silk Screen LED Identifier | Colour | Description |
|---|---|---|
| D7 | Red | PSUA Off |
| | Green | PSUA On |
| D8 | Red | PSUB Off |
| | Green | PSUB On |
| D12 | Red | PSUC Off |
| | Green | PSUC On |
| D13 | Red | PSUD Off |
| | Green | PSUD On |

Please note that individual modules do not use all power supplies. For example when a BenADDA is fitted to a BenONE it is normal for only three of the LEDs to change from RED to GREEN to indicate they have been turned on.

### 6.4.4 User LEDs

Please note that on Version 1 BenONE PCB there are only three available tri-colours (D3, D4, D5). On Version 2 of the BenONE PCB there are 4 tri-colours LEDs (D3, D4, D5 and D6).

# Build Options and Jumper Settings

- BenONE – PCI Build Options
- Jumper Settings

## 6.5 Build Options

### 6.5.1 I/O Modules.

This option allows the user to select an appropriate standalone IO module. There is currently only one option but this will increase to three.

| Option | Module | Nallatech Part Number |
|---|---|---|
| -M0 | No Module Populated | - |
| -M1 | USB 1.1 I/O Module | NT101-0134-1 |

**Table 10: I/O Module Options**

### 6.5.2 Compact Flash (-CF)

Allows the BenONE – PCI card to be booted from a compact flash module located on the rear of the card. The BenONE – PCI card utilises the Xilinx System ACE chipset to control the compact flash, however, a third party programmer is required to download designs to a compact flash card.

### 6.5.3 VAUX Battery (-B)

Located on the rear of the card this option is required if you wish to use the encryption bit stream format to protect your designs.

### 6.5.4 Power Supplies

The default option for DIME II module power supplies is the fixed voltage module. This offers the customer a cost saving by trading off the flexibility of a full programmable supply. Nallatech would recommend the full programmable supply option if a customer were to use the BenONE - PCI with several different modules. Special options can be offered with a mixture of fixed and programmable supplies, please contact Nallatech for more information on these options.

| Power supply Configuration | Power supply option |
|---|---|
| Fixed Voltage Power Supply | -FPS |
| Programmable Voltage Power Supply | -PPS |

**Table 11: Power Supply Options**

### 6.5.5 Standalone Power Connection(-SPC)

This option is required if you are to use the BenONE - PCI in its standalone configuration. See Section 6.8.1 'Standalone Power Connector', for the pin-out of this connector.

## 6.6 Jumper Settings

The BenONE - PCI requires no jumpers to be set by the user.

## 6.7 Peripherals

Apart from standard DIME-II modules such as the BenADDA, there are also additional accessories. The StrathLED is an add-on module that can be plugged into the adjacent bus header (J10) to provide an array of LEDs that can be used for display purposes.

# Pin-out Information

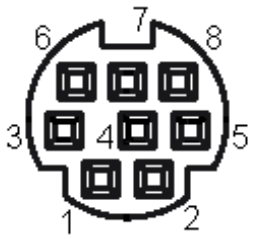
In this Section:

- External board connections

## 6.8 External Connectors.

### 6.8.1 Standalone Power Connector

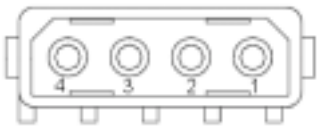
This is a standard 8 way mini DIN connector

| Supply Voltage | Pin# | Looking at connector |
|---|---|---|
| +5V | 1 |  |
| +5V | 2 | |
| +5V | 3 | |
| +12V | 4 | |
| -12V | 5 | |
| RETURN | 6 | |
| RETURN | 7 | |
| RETURN | 8 | |



### 6.8.2 Disk Drive Style Power Connector

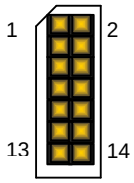
If you are developing very large designs that may require high current ratings Nallatech would advise the use of this connector to increase the power rating. You will see however that this connector is limited to supplying only the positive supplies.

| Supply Voltage | Pin# | Looking at connector |
|---|---|---|
| +12V | 1 |  |
| RETURN | 2 | |
| RETURN | 3 | |
| +5V | 4 | |

If the XtremeDSP Development Kit is located in a PCI slot then this connector if used must be connected to the same ATX power supply that is powering the slot. Alternatively if the XtremeDSP Development Kit is operating as a standalone unit this can be connected to any external source, however, ensure that if a power supply is also connected to the Mini DIN connector that it to is not attempting to power the +5V and +12V rails.

### 6.8.3 PLink Bus Header (J11)

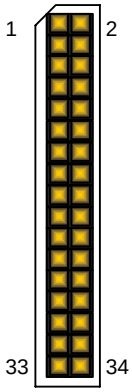
This header is connected to PLink0 on the DIME-II module slot.

| Header Pin Number | Name | DIME-II Module Connector | |
|---|---|---|---|
| 1 | PP0LK<0> | PA2 |  |
| 2 | PP0LK<1> | PA3 | |
| 3 | PP0LK<2> | PA4 | |
| 4 | PP0LK<3> | PA5 | |
| 5 | PP0LK<4> | PA6 | |
| 6 | PP0LK<5> | PA7 | |
| 7 | PP0LK<6> | PA8 | |
| 8 | PP0LK<7> | PA9 | |
| 9 | PP0LK<8> | PA11 | |
| 10 | PP0LK<9> | PA12 | |
| 11 | PP0LK<10> | PA13 | |
| 12 | PP0LK<11> | PA14 | |
| 13 | GND | N/A | |
| 14 | GND | N/A | |



### 6.8.4 Adjacent Bus Header (J10)

This header is a 28-bit general-purpose bus and is connected to the Adjacent IN Bus on the DIME-II module slot. The StrathLED is an optional module that can be plugged into this header to provide an array of LEDs that can be used for display purposes.

| Header Pin Number | Name | DIME-II Module Connector | |
|---|---|---|---|
| 1 | ADJIN<12> | PA42 | |
| 2 | ADJIN<13> | PA43 | |
| 3 | ADJIN<10> | PA40 |  |
| 4 | ADJIN<11> | PA41 | |
| 5 | ADJIN<8> | PA38 | |
| 6 | ADJIN<9> | PA39 | |
| 7 | ADJIN<6> | PA35 | |
| 8 | ADJIN<7> | PA36 | |
| 9 | ADJIN<4> | PA33 | |
| 10 | ADJIN<5> | PA34 | |
| 11 | ADJIN<2> | PA31 | |
| 12 | ADJIN<3> | PA32 | |
| 13 | ADJIN<0> | PA29 | |

| Header Pin Number | Name | DIME-II Module Connector | |
|---|---|---|---|
| 14 | ADJIN<1> | PA30 | |
| 15 | ADJIN<14> | PA44 | |
| 16 | ADJIN<15> | PA45 | |
| 17 | ADJIN<16> | PA47 | |
| 18 | ADJIN<17> | PA48 | |
| 19 | ADJIN<18> | PA49 | |
| 20 | ADJIN<19> | PA50 | |
| 21 | ADJIN<20> | PA51 | |
| 22 | ADJIN<21> | PA52 | |
| 23 | ADJIN<22> | PA53 | |
| 24 | ADJIN<23> | PA54 | |
| 25 | ADJIN<24> | PA56 | |
| 26 | ADJIN<25> | PA57 | |
| 27 | ADJIN<26> | PA58 | |
| 28 | ADJIN<27> | PA59 | |
| 29 | 3.3V | N/A | |
| 30 | GND | N/A | |
| 31 | NC | NC | |
| 32 | NC | NC | |
| 33 | NC | NC | |
| 34 | NC | NC | |

**Table 12: Adjacent Bus Header**

### 6.8.5 Fan Jumpers

Table 5 lists all the necessary jumper configurations for each fan. **Note:** Each fan jumper is a 2-pin header, one pin supplies 5 volts to the fan and the other provides a ground. The silkscreen on the PCB for each fan jumper is white box with a corner missing. Pin 1 is always nearest the missing corner.

| Fan Jumper Name | Description |
|---|---|
| J7  1 2 | Supplies power to a 5 Volt Cooling Fan<br>Pin 1: Ground<br>Pin 2: + 5 Volts |
| J8  1 2 | Supplies power to a 5 Volt Cooling Fan<br>Pin 1: Ground<br>Pin 2: + 5 Volts |

**Table 13: Fan Jumpers**

# Part 3

# BenADDA

This part of the User Guide provides you with information on installing and using the BenADDA DIME-II module. In the following Sections:

- Section 7: BenADDA Overview
- Section 8: BenADDA Installation Guide
- Section 9: BenADDA Implementation Guide
- Section 10: BenADDA Reference Guide

# Section 7

# BenADDA Overview

## 7.1 BenADDA

The BenADDA DIME-II module provides high-speed digital-to-analogue and analogue-to-digital conversion capability. As part of the scalable DIME-II family, the BenADDA can be easily integrated into systems, through the range of available DIME-II motherboards and associated software/firmware.

The module contains two high-speed ADC and two high-speed DAC channels, which allow for flexible, high-resolution data conversion for both baseband and IF applications. Key to the BenADDA's performance is the on-board Xilinx® Virtex™-II FPGA which provides you with a powerful data processing resource. Some of the main application areas for the BenADDA include mobile communications systems, infrared imaging, wideband cable systems and multi-channel, multi-mode receivers.

**Figure 31: BenADDA**

## 7.2 Key Features

The key features of the XtremeDSP Development Kit are:

- On-board Xilinx Virtex-II FPGA
- Various FPGA device packages, sizes and speed grade options available
- Compatible with Nallatech's FUSE™ reconfigurable computing operating system
- Two independent analogue capture channels
- 2x 14-Bit ADC Resolution, up to 105MSPS per channel sampling rate
- Two independent channels to extract analogue data
- 2x 14-Bit DAC Resolution, up to 160MSPS per channel sampling rate
- Up-to 8MB of ZBT SRAM memory, in two independent banks
- Nallatech ZBT SRAM interfacing IP Core available
- Multiple Clocking Options: Internal & External
- Example designs and source code included
- Status LEDs

## 7.3 BenADDA specification variations

The BenADDA module is available in the following specifications:

| Features | 2V250 FG456 | 2V1000 FG456 | 2V2000 FG676 | 2V3000 FG676 | 2V3000 FG1152 | 2V6000 FG1152 |
|---|---|---|---|---|---|---|
| ADC Up to 105MSPS (x2) | 14-Bit | 14-Bit | 14-Bit | 14-Bit | 14-Bit | 14-Bit |
| DAC 160MSPS (x2) | 14-Bit | 14-Bit | 14-Bit | 14-Bit | 14-Bit | 14-Bit |
| Total ZBT SRAM* | n/a | n/a | 4MB | 4MB | 8MB | 8MB |
| ZBT Memory Banks | n/a | n/a | 1 Bank | 1 Banks | 2 Bank | 2 Banks |
| On Board Programmable Clock s in User FPGA | 3 | 3 | 3 | 3 | 3 | 3 |
| No. of methods for Clocking ADCs/ DACs** | 7 | 7 | 7 | 7 | 7 | 7 |
| User FPGA Status LEDs | 2 | 2 | 2 | 2 | 2 | 2 |

**Table 14: BenADDA specifications**

*Figures quoted are in Mbytes and represent the maximum total size of memory on-board. Please note that t the actual populated memory size may differ from the max value and will depend upon the specific configuration. Figures here list the maximum available size at time of print.

**Clocking options include: external Clock source available through front panel, 4 user programmable clocks from User FPGA (can be used as single-ended or differential), either use of on-board Oscillator OR 2nd external clock source (this is a special build option). A total of 6 clocking methods will be available on the module at any one time.

# 7.4 BenADDA functional diagram

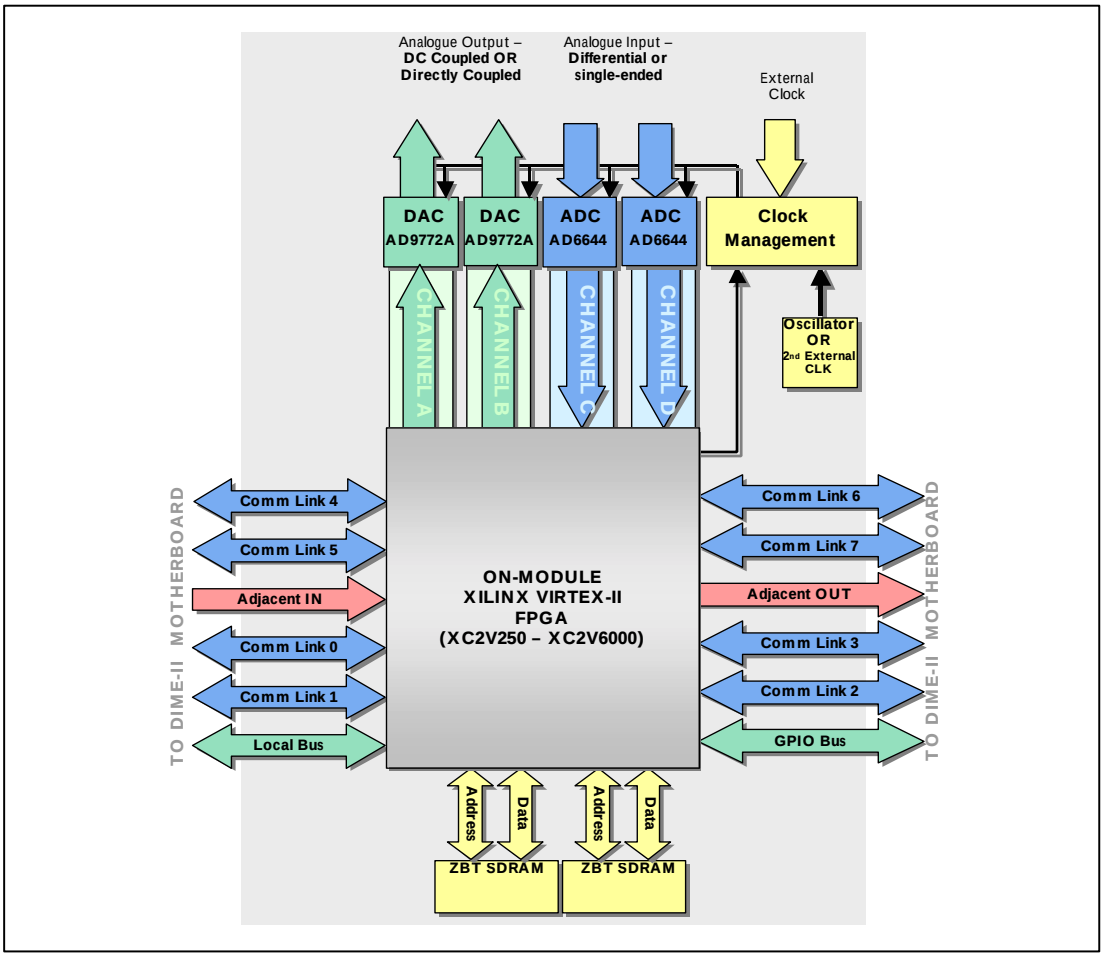



**Figure 32: BenADDA functional diagram**

# Section 8

# Installation Guide

This part of the User Guide provides you with installation information for the BenADDA hardware and software. In this Section:

- Hardware Installation
- Software Installation

# Hardware Installation

- Motherboard requirements
- BenADDA hardware installation features
- Hardware installation instructions

## 8.1 BenADDA motherboard requirements

The BenADDA should be hosted on a DIME-II motherboard. The BenADDA is a single slot module, so is compatible with any DIME-II motherboard, a range of which are available from Nallatech.

## 8.2 Hardware features applicable to installation

The BenADDA is an Electro-Static Discharge (ESD) sensitive device. ESD handling procedures must be observed during the handling and installation of the BenADDA.

If the BenADDA is supplied as a separate unit, mounting screws and a front panel will be supplied for fitting to the motherboard. The physical features of the DIME-II module referred to in the installation instructions are highlighted below:

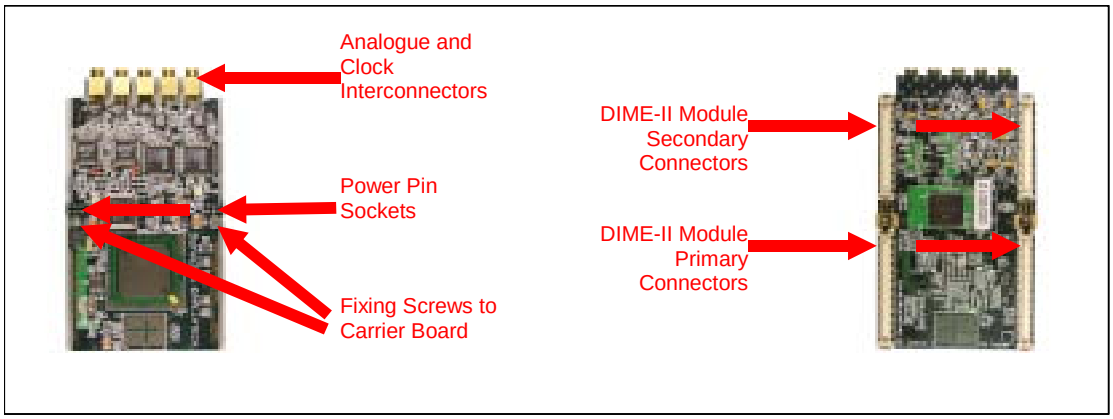



**Figure 33: Hardware features for installation**

## 8.3 Fitting the BenADDA

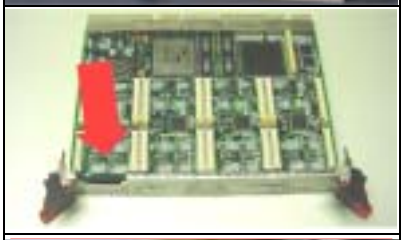
The XtremeDSP Development Kit fitting instructions are given below. In this example, the images show an unspecified DIME-II module being fitted to a Nallatech BenERA™ cPCI DIME-II motherboard, but the instructions are valid for any combination of DIME-II module or motherboard.

1. Ensure ESD handling procedures are observed during installation.

2. Ensure the power to the DIME-II host motherboard is switched off.

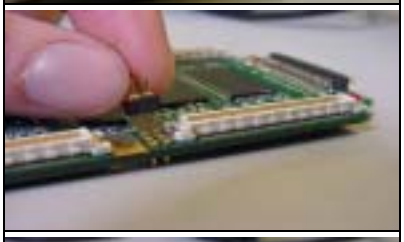

3. Remove the motherboard from the host PC or rack and place on a flat surface, with the module slots facing upwards.

4. Locate the empty DIME-II slot on the motherboard, onto which the module is to be installed.



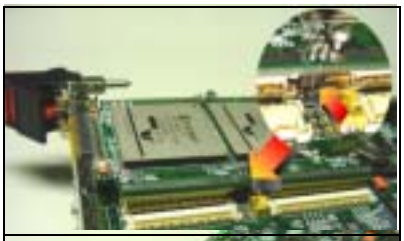
5. If the DIME-II power interconnects are not fitted to the module, these should be fitted to the sockets located on the underside of the module.

Insert the power interconnects as shown here.



Power interconnects are now properly fitted. Repeat step 5 again for the power interconnect sockets on the other side of the module.

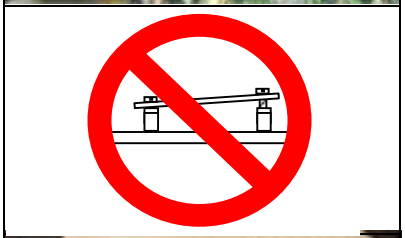
6. Orient the module so that it is facing the correct way on the module slot. Carefully line up the DIME-II Connectors and the power interconnects; paying special attention to ensure the power interconnect pins on both sides of the module are all lined up to the sockets on the motherboard.



7. Apply light downward pressure to the module. While still holding it in place, proceed to step 8.

8. Using a flat-bladed screwdriver, begin to tighten the DIME-II Fixing Screws. Tighten the screws alternately, tightening each screw one turn, before swapping and tightening the opposite screw by one turn, until both are fully tightened.

Avoid tightening either screw more than the other – this would result in the module being fixed to the motherboard at an angle, resulting in connector and module damage.



9. The DIME-II motherboard can then be re-fitted to the host PC or rack.

10. The PC/rack power can now be switched on.


This concludes the hardware installation instructions. You should now proceed to the next Section for software installation instructions.

# Software Installation

- Windows 95/98/2000/Me/XP Driver installation
- Linux driver installation
- Software tools installation

Remember you need to install the relevant software for the DIME-II carrier card being used. If you need to install the DIME-II carrier card software, make sure you have restarted your machine before installing the BenADDA module software, as certain environmental settings need to be updated.

## 8.4 Software Driver Installation

There is no hardware driver necessary to be installed. However, BenADDA data files need to be installed for the associated FUSE Software that is supplied with compatible DIME-II carrier cards.

Full software installation instructions can be found in the 'Getting Started' Section at the start of this User Guide.

### 8.4.1 Linux (Tested on redhat Standard installation 6.2)

For Linux installation procedures see the 'Linux installation' procedures for the BenONE motherboard Section on page 30, as this package contains files for the BenADDA in the DSP Development kit.

### 8.4.2 Software Tools Installation

There is no additional software that needs to be installed for the BenADDA module. However, in order to make full use of the module's potential, the following software components are recommended:

- DIME-II Carrier Card Software
- FPGA Synthesis Software
- Xilinx FPGA Implementation Software
- HDL Simulation Software

# Section 9

# Implementation Guide

This part of the User Guide contains information on how to use the BenADDA module. In this Section:

- FPGA Configuration
- FPGA Application Design
- Example Application 1
- Example Application 2

# FPGA Configuration

In this Section:

- FPGA Configuration using the FUSE Software GUI
- FPGA Configuration using the FUSE Software API

## 9.1 FPGA Configuration

The User FPGA on the BenADDA can be configured via the following methods:

- Using the FUSE Software GUI (Graphical User Interface)
- Using DIMEScript
- Using FUSE Software APIs

## 9.2 FPGA Configuration using FUSE

The FUSE Software provides the following functionality:

- Configuring FPGAs on the BenADDA
- Controlling reset signals
- Controlling programmable clock frequencies
- Sending data to designs running in the User FPGA
- Reading data from designs running in the User FPGA
- Resetting PCI FIFOs.

This functionality is provided in the following guises:

- FUSE GUI Application
- DIMEScript
- FUSE Software development APIs.

### 9.2.1 FPGA Configuration using FUSE GUI

The FUSE GUI Application is an easy to use software interface, which allows users to access a subset of the functionality provided by FUSE. Full instructions on how to use the GUI are provided in the FUSE System Software User Guide on the supplied FUSE CD.

### 9.2.2 FPGA Configuration using DIMEScript

DIMEScript is a high-level scripting language, which provides users with a simple and easy to use language for the configuration and control of DIME systems. DIMEScript uses a simple command set, eliminating the need for developers to use complicated programming interfaces to control and communicate with application designs running in FPGAs. DIMEScript also offers platform portability through ASCII based scripts, allowing users to use DIMEScript on both Windows and Linux installations.

DIMEScript can be used either to write script files, which can then be executed as a single process, or it can be used from a command line interface, with the user executing commands as required. Full instructions on how to use the DIMEScript are provided in the DIMEScript User Guide, which is on the supplied FUSE CD.

### 9.2.3 FPGA Configuration using the FUSE APIs

The FUSE Software development API is a software development API, enabling users to call functions to control DIME hardware in their own programs. This allows users to develop software applications to complement the FPGA application designs running on DIME hardware.

FUSE APIs are available to support a number of development languages, including C, C++, Matlab and Java. Full instructions on how to use the APIs are provided in the FUSE C-C++ API Developers Guide on the supplied FUSE CD.

# FPGA Application Design

- Interfacing to the User FPGA
- User FPGA design synthesis and implementation
- Communication between the User and PCI FPGAs

## 9.3 Interfacing to the FPGA

It is assumed that most users will be familiar with the principles of FPGA design, or have appropriate resources/support to complete FPGA designs. The following information is intended to detail any BenADDA-specific issues in the design process.

A User Constraints File (UCF) for the user-accessible connections to the FPGA is provided with the BenADDA, on the product CD. This allows you to quickly and easily link top-level HDL code to the FPGA I/O for implementation purposes.

Since the supplied UCF names are pre-defined, you can save nuisance and development times by writing your top-level HDL code using the same port naming convention as the Nallatech UCF. This saves changing the UCF retrospectively to match your code.

There are a number of user-accessible interfaces from the User FPGA to other devices on the BenADDA, which you can utilise for design purposes:

- User FPGA – DIME-II Signals
- User FPGA – ADCs
- User FPGA – DACs
- User FPGA – Clock FPGA
- User FPGA – ZBT Memory

These interfaces are shown and highlighted below in Figure 34:

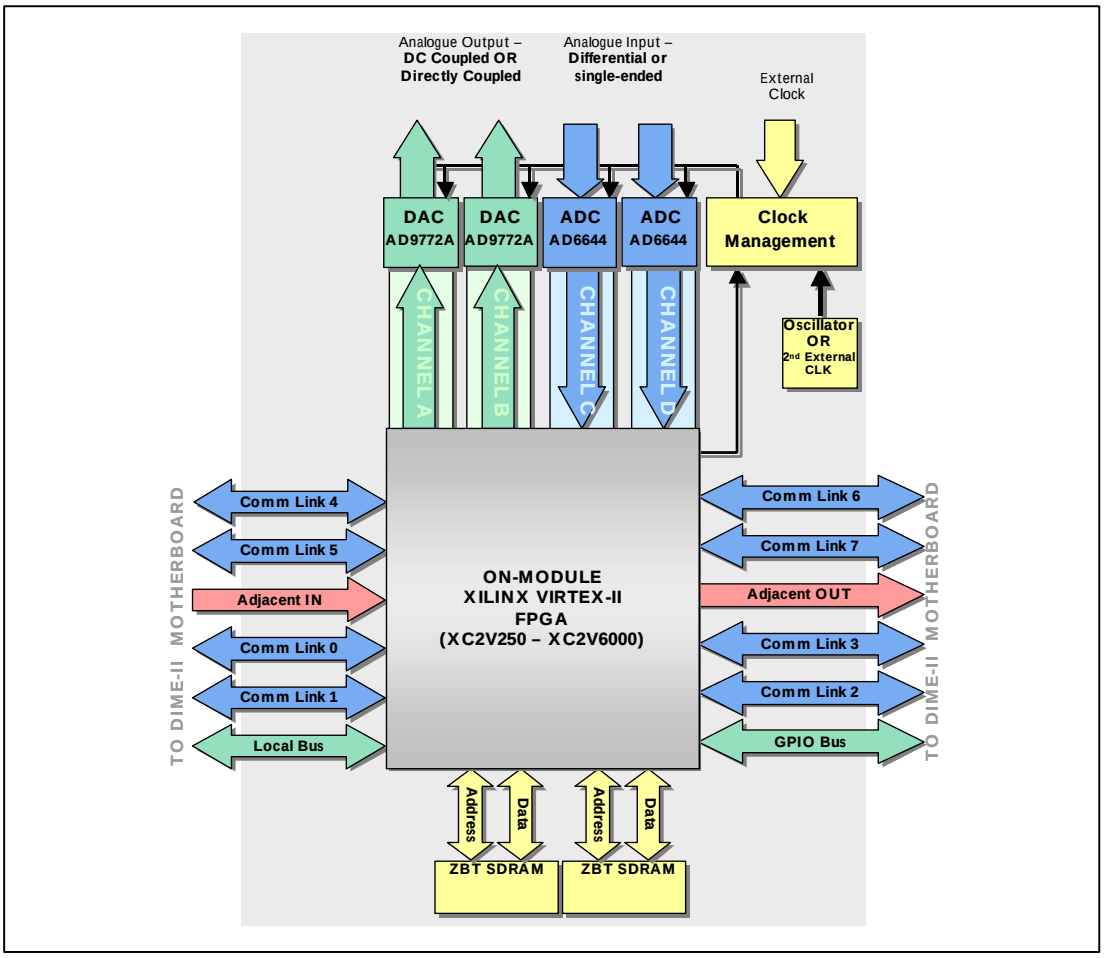



**Figure 34: Interfacing to User FPGA**

# 9.4 Synthesis and Implementation Settings

This Section details the synthesis and implementation settings, which should be used for the development of FPGA designs to run on Nallatech hardware.

## 9.4.1 Synthesis Options

When developing FPGA designs to run on Nallatech hardware, it is not necessary to select any specific settings for the synthesis of HDL code for FPGA designs.

## 9.4.2 Implementation Options

Developing FPGA designs to run on Nallatech hardware with on-board Xilinx FPGAs will ultimately require use of the Xilinx Implementation tools to take the synthesised design to the hardware device. These tools are available in a variety of formats, including Alliance, Foundation and ISE.

When performing the implementation stage of the design process, some settings are mandatory and need to be specified for the design to configure and run on Nallatech hardware. This Section details these settings. Also detailed are some common additional settings for the implementation options – it should be stressed however that the common settings are optional and that the developer should select the settings as appropriate for their design.

### Necessary Settings

(i). Enable Readback and Reconfiguration

(ii). Select the JTAG Start-up Clock (Selected configuration clock)

(iii). The pull down option should be enabled for unused pins.

## Optional Settings

If you are targeting a Virtex-II Engineering Silicon Part it is advised to set the following environmental variable:

- **XIL_BITGEN_VIRTEX2ES=YES**;

Engineering silicon parts are identified by the "ES" marking on the face of the chip. Please see the Xilinx answers database if you require further details.

It is worth also noting that when using Engineering Silicon parts some care should be taken when using DCM in that the DCM should be placed in the same quadrant as the BUFGMUX. Setting the above environmental variable will not explicitly carry out the location of these components for you but the switch ensures that when the DRC (Design Rule Check) is carried out a bitsfile generation, it will check that these have been placed in the same quadrant.

Note also that for the ES parts the Triple DES bit stream encryption and partial reconfiguration features are not compatible with standard bit stream generation.

See the following Xilinx answer database records:

Answer Record # 12521: 4.1i Virtex-II BitGen - "BitGen:218" warning message is issued when XIL_BITGEN_VIRTEX2ES is set

Answer Record # 13223: 4.1i SP3 - 4.1.03i Service Pack 3 update

Answer Record # 12719: 4.1i, BitGen - "ERROR: DesignRules:557 - Blockcheck: Invalid connection used between BUFGMUX and DCM...."

Answer Record # 12671: 4.1i SP2 - 4.1.02i Service Pack 2 update

Answer Record # 12518: 4.1i SP1 - 4.1.01i Service Pack 1 update

Answer Record # 12326: 4.1i Virtex-II BitGen - A patch is available to correct Virtex-II bit stream generation. Does it affect bit stream size?

Answer Record # 11756: 4.1i Virtex-II - Are there restrictions on IBUFG, DCM, and BUFG/BUFGMUX routing in Virtex-II devices?

- Common Settings

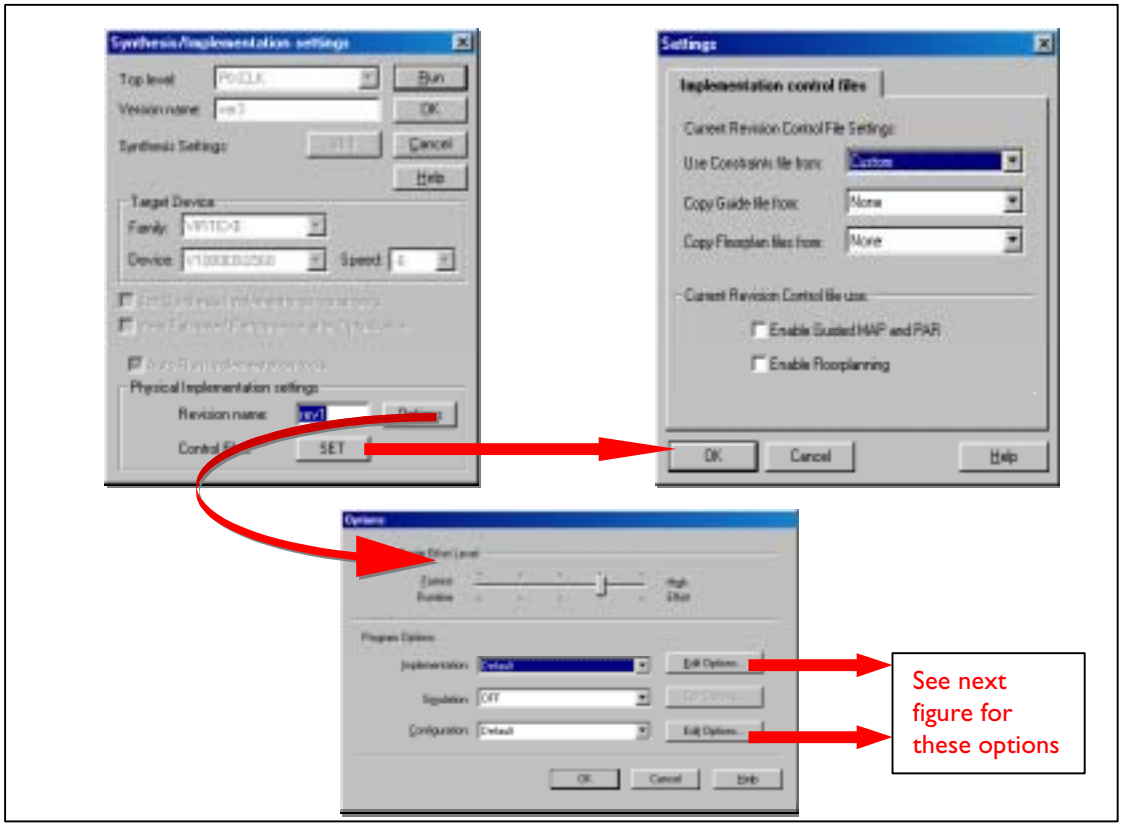



**Figure 35: Implementation Configuration Settings**

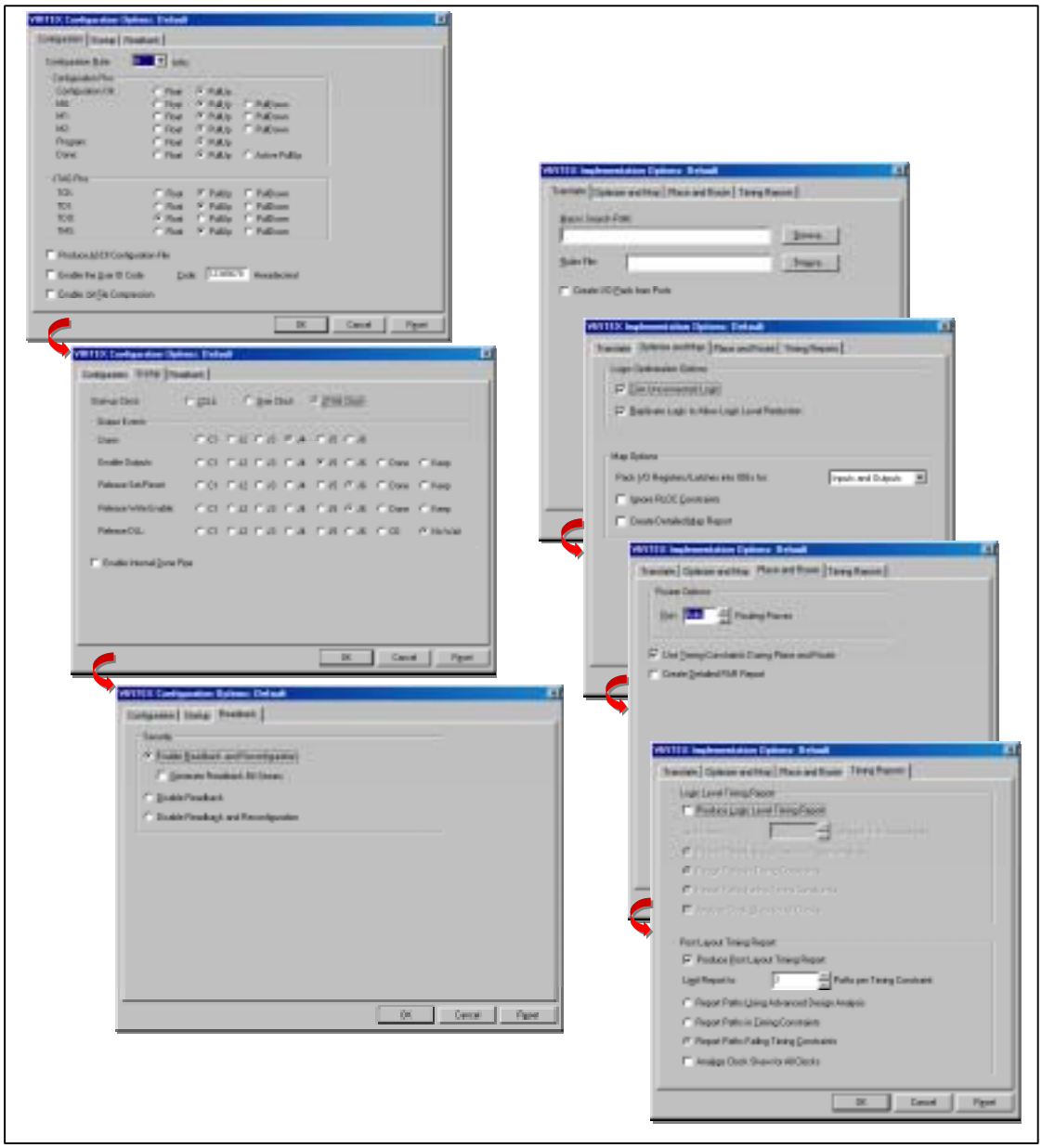


**Figure 36: Implementation Configuration Settings Cont.**

The 'common settings' shown in Figure 35 and Figure 36 are intended for illustrative purposes and are captures taken from Xilinx Foundation implementation tools. The figure shows the location of each of the options and shows a common selection of settings that have been used to create the example bit files provided with the product. They do not take into account any additional requirements of the design, which may apply in specific user applications.

Failure to enable readback will result in the value being returned from the status register of 0x0. This will manifest itself in the configuration software reporting an error of DONE LOW, INIT LOW, No CRC errors.

If reconfiguration has not been enabled and you configure once, it is necessary to cycle the power to the FPGAs in order to clear the security protection on the FPGA.

# Example Application 1

This Section covers:

- ADC to DAC feed-through design

## 9.5 Introduction

The screenshots in this Section display FPGA configuration procedures for a BenADDA module populated on a BenONE motherboard. However, these procedures can be applied to any DIME-II motherboard. This Section necessitates that all software required by the BenADDA has been installed as previously detailed and the operating system running on the host PC is either Windows 9x/NT/Me/2000/XP or Linux.

This example application describes a simple ADC to DAC feed-through design on the BenADDA. This design is included on the XtremeDSP Development Kit CD, to illustrate the capture of data from the ADC and the output of data to the DAC. The design also includes a led flash pattern.

In addition, this example can be used as a confidence test for you to ensure correct operation of the module.

## 9.6 Functional Description

The Clock FPGA is used to route the clock signals to the other devices on the BenADDA. The design takes the signal from the on-board crystal oscillator and distributes it to both ADCs, both DACs and the main FPGA as illustrated in Figure 37.

- The VHDL source code for this design is included on the XtremeDSP Development Kit CD at the path '<CDROM Drive>:\Examples\Adc_to_Dac\Source\AdctoDac.vhd'.
- For convenience, a Xilinx ISE Project Navigator File for each implementation option is included at the path '<CDROM Drive>:\Examples\Adc_to_Dac\Source\ISE'.

Please note that the BenADDA supports two variants that support different FPGA packages. One supports a FF1152 and one a FG676. Note that the XC2V3000 Virtex-II part is supported in both packages. Therefore care should be taken when selecting bitfiles for configuration that they are indeed for the correct target FPG package.

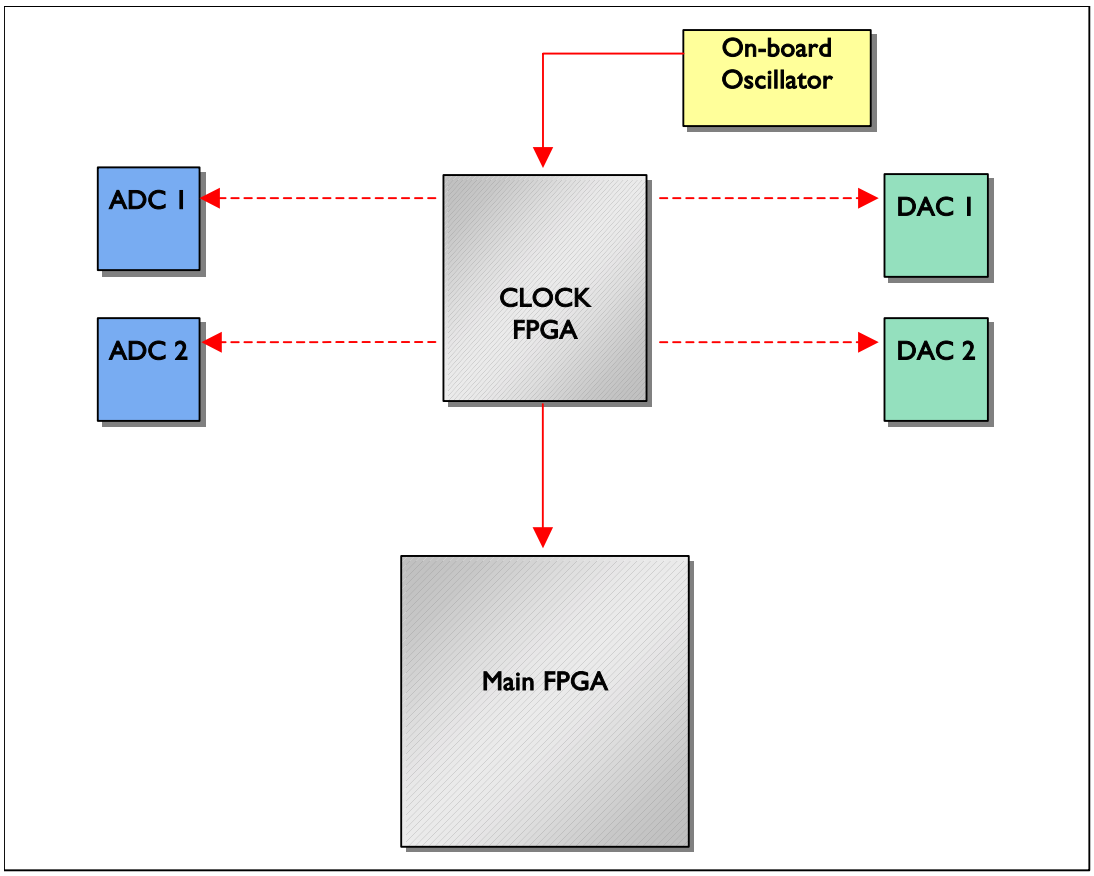



**Figure 37: Clock FPGA**

The design for the main FPGA takes the input data from both ADCs, converts it from 2s complement to offset binary and outputs it to both DACs as illustrated in Figure 38.

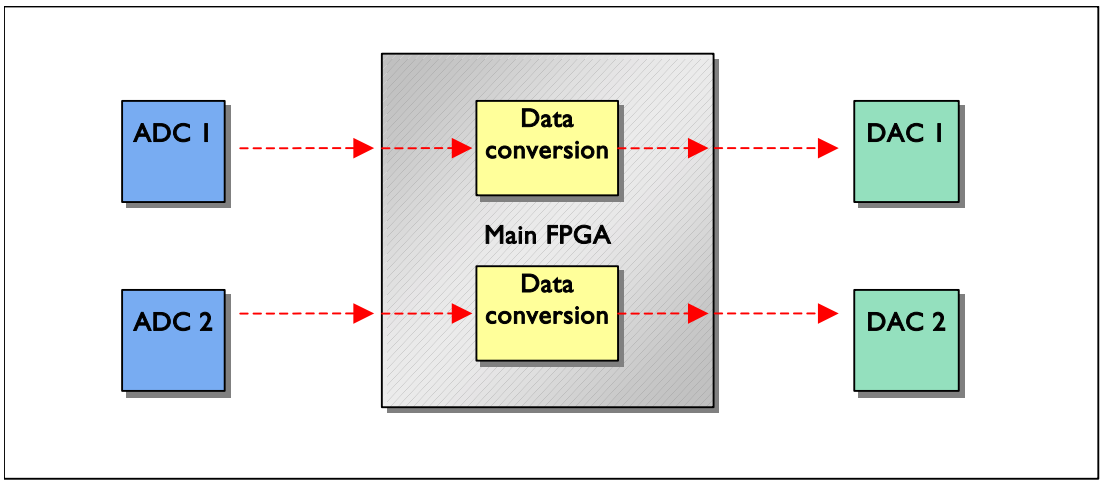



**Figure 38: ADC to DAC feed-through design**

## 9.7 Configuring the ADC to DAC feed-through example

To start the GUI:

1. On the Windows task bar, click:
   ***Start -> Programs -> FUSE -> Software -> FUSE Probe***

2. In the FUSE Probe program window shown below, select **Card Control\Open card** from the menu at the top of the window.

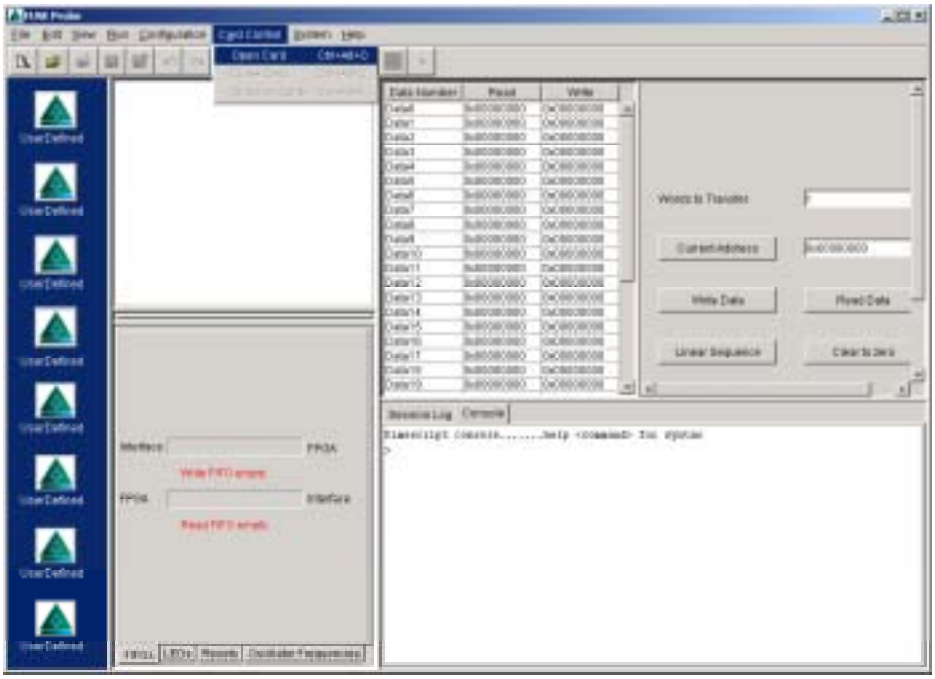


**Figure 39: Fuse System Software GUI**

3. The Locate Card window is then displayed. Select **PCI** in the Interface box, and **All Card Types** in the Card Type box. Then click on **Locate Cards**.

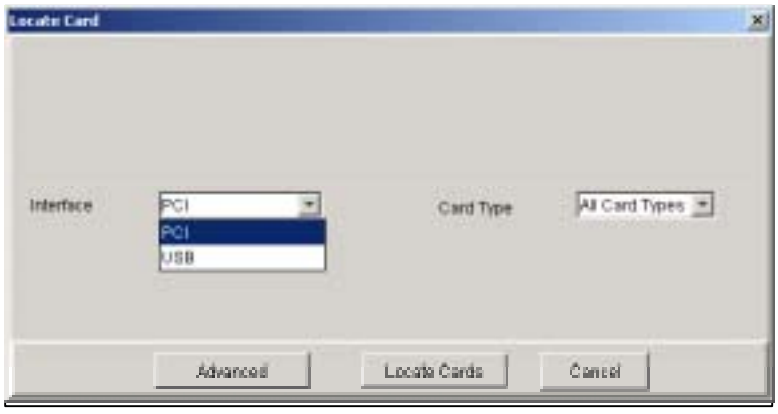



**Figure 40: Dialogue box to locate cards**

4. In the Selected Cards window, check the BenONE card box and click on **Open Cards**.

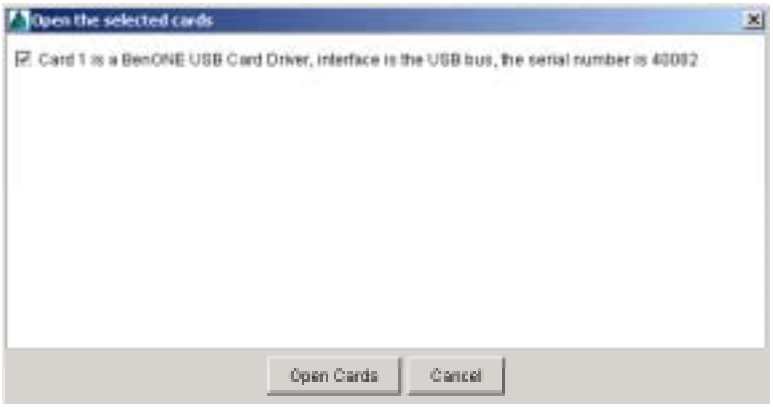



**Figure 41: Dialogue box to open cards**

5. Right click the Clock FPGA in the device tree under the BenADDA module and select '**Assign and Configure**'.

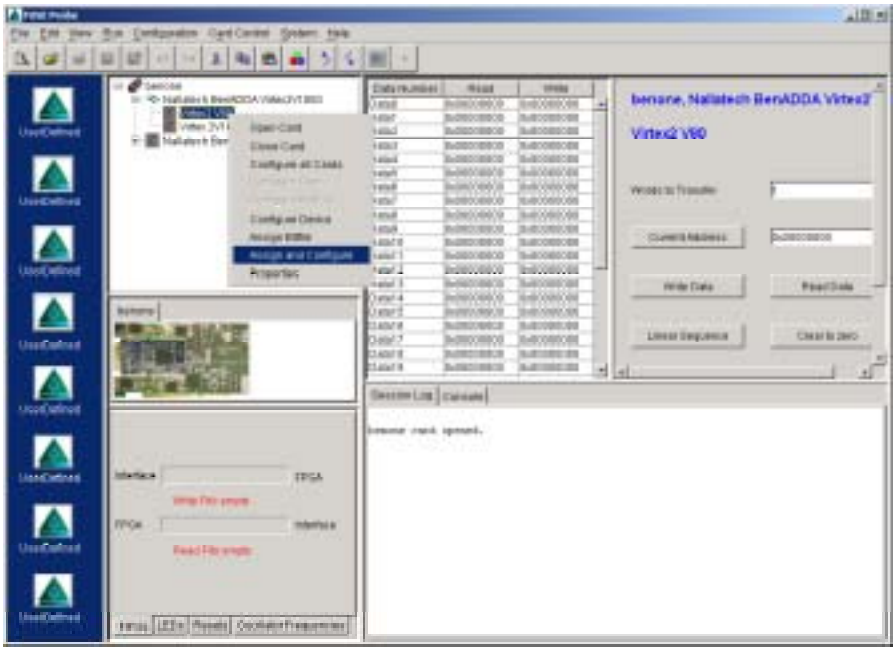



**Figure 42: GUI showing open card**

6. A dialogue box is displayed which allows you to browse and select the correct bitfile for the device. The bitfile for this device is named 'osc_clk_2v80.bit' and is located at the path '<CDROM Drive>:\Examples\Clock_Designs\PCB_V2\' on the XtremeDSP Development Kit CD.

Please note that the BenADDA supports two variants that support different FPGA packages. One supports a FF1152 and one a FG676. Note that the XC2V3000 Virtex-II part is supported in both packages. Therefore care should be taken when selecting bitfiles for configuration that they are indeed for the correct target FPG package.

7. In a similar way assign and configure the main FPGA. The bitfiles for this FPGA can be found at the path '<CDROM Drive>:\Examples\Adc_to_Dac\DSP_Kit\'. and have the naming convention 'adctodac_2v########.bit' where #### represents the device size and **** represents the package. Ensure the device is configured with the correct bitfile to avoid any damage to the FPGA.
8. A LED flash pattern should now be displayed on the LEDs and any data fed into the ADC should be seen at the output of the DAC. This data can also be viewed on an oscilloscope. After both devices have been configured select the reset tab in the bottom left of the main screen. Tick both reset check boxes and then un-tick both boxes to carry out a full reset on the system.

With Engineering Silicon XC2V3000 FPGAs, configuration procedures should be carried out twice in order to successfully start-up.

# Example Application 2

In this Section:

- Modulator Example

## 9.8 Introduction

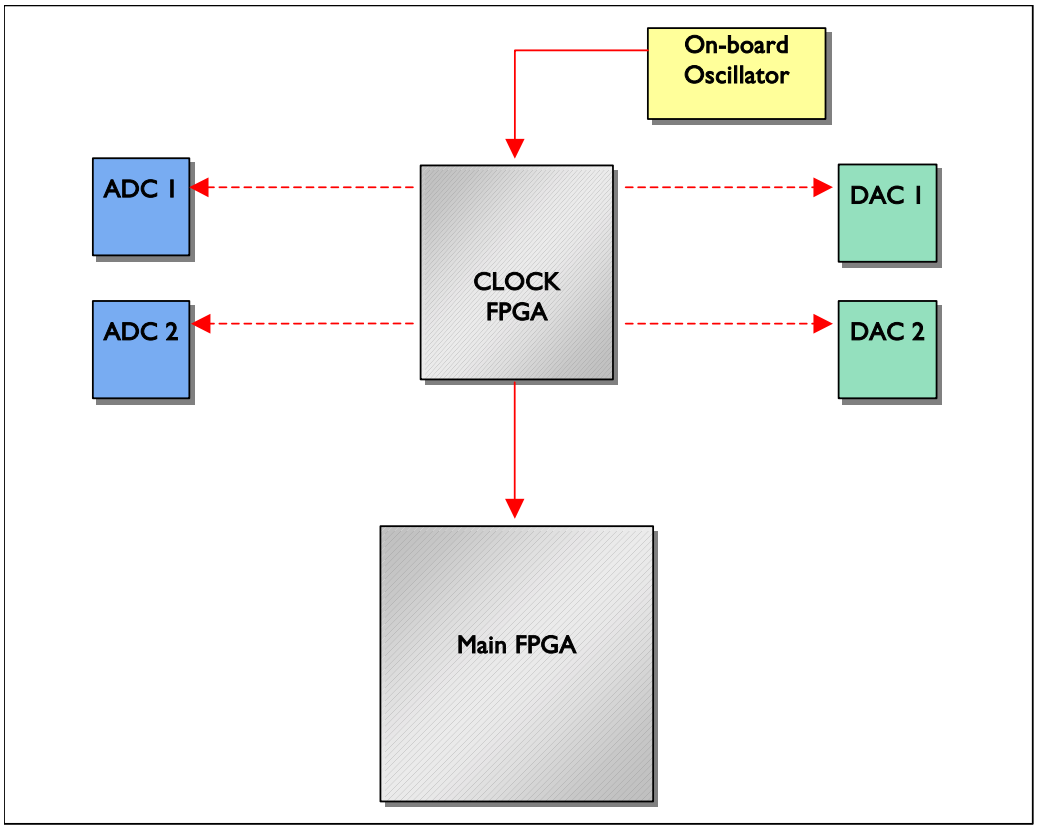
This Section necessitates that Xilinx Integrated Software Environment (ISE) 4.2i software has been installed and the operating system running on the host PC is either Windows 9x/NT/Me/2000/XP or Linux.

The second example application provides a basic introduction to using Coregen for the rapid creation of DSP designs. This example involves modulating the input waveform from both ADCs and outputting the modulated waves on the corresponding DAC. The 8 MHz sine wave that modulates the input waveform is internally generated in the FPGA. This design also includes an LED flash pattern.

## 9.9 Functional Description

The design for the Main FPGA takes the data from the ADCs and multiplies it by an internally generated sine wave. The outputs from both multipliers are converted from 2s complement to offset binary and then output to the DACs as illustrated in Figure 43.

- The VHDL source code for this design is included on the XtremeDSP Development Kit CD at the path: '<CDROMDrive>:\Examples\Modulator\Source\Modulator.vhd'
- For convenience, a Xilinx ISE Project Navigator File for each implementation option is included at the path '<CDROM Drive>:\Examples\Modulator\Source\ISE'.



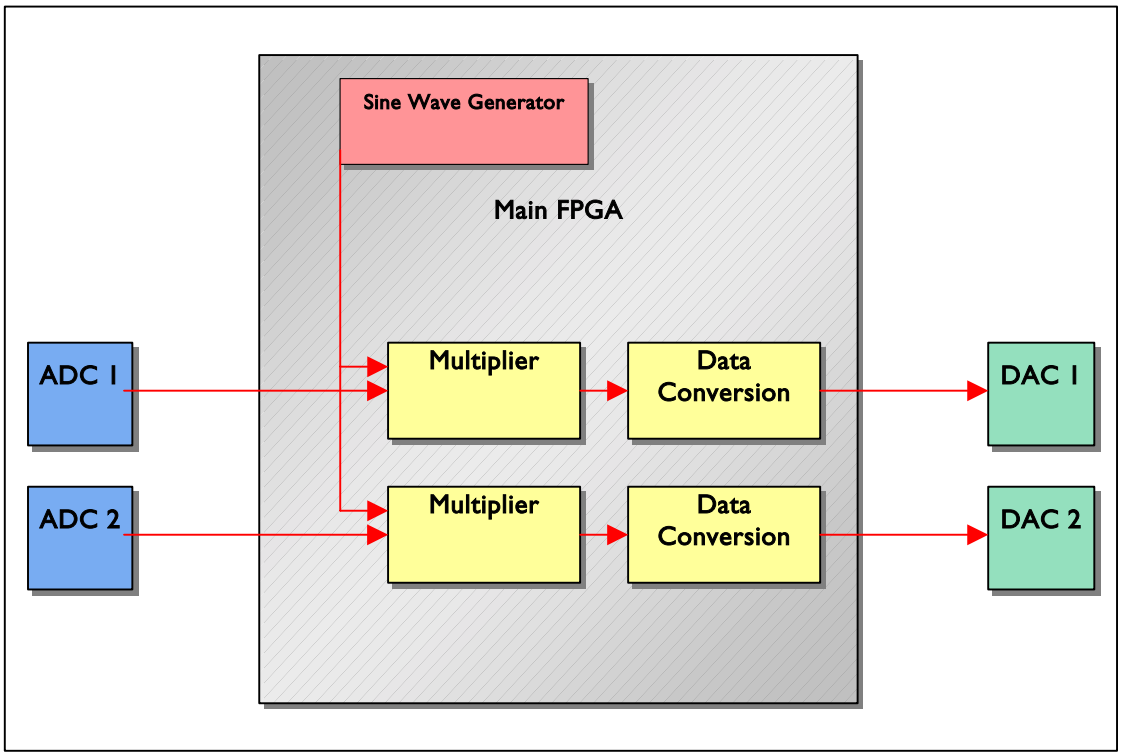



**Figure 43: Modulator Design**

### 9.9.1 Creation and Configuration of Sine Wave and Multiplier Components

The source code declares and instantiates the sine wave generator and multipliers created in Coregen. Coregen components are inserted into a design by right clicking on the top level of the design and selecting 'New source'. This brings up the dialogue box shown in Figure 44. Select Coregen IP, enter a file name and then click **Next** to proceed.

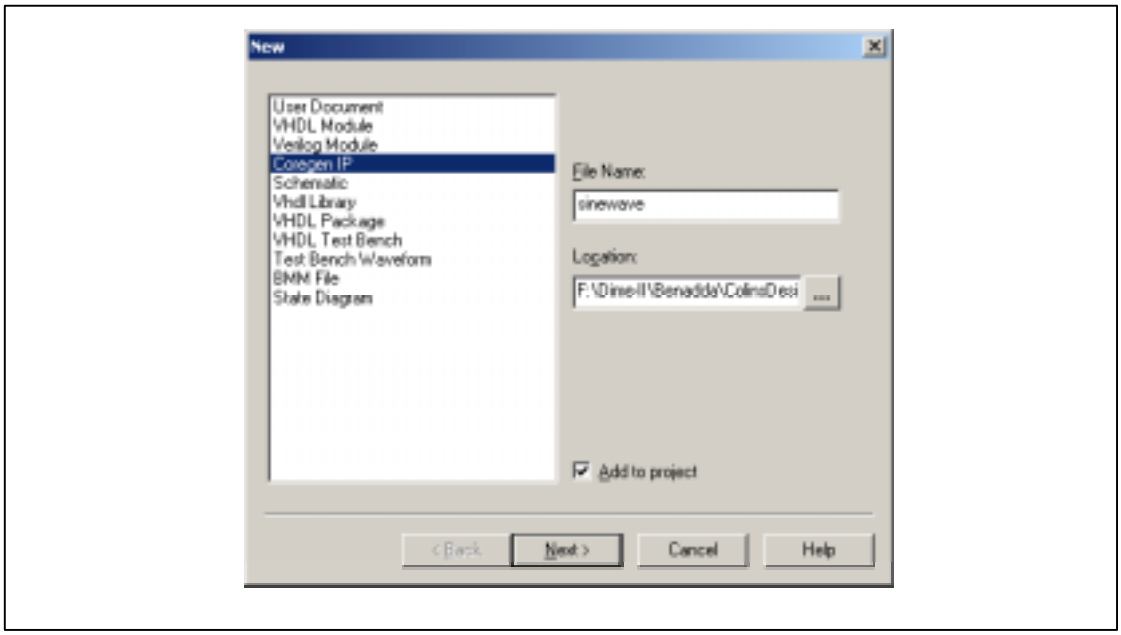



**Figure 44: New Source Dialogue Box**

The display box then shows the source type and source name. If these are correct, click **Finish** to display the Xilinx Core Generator window shown in Figure 45:

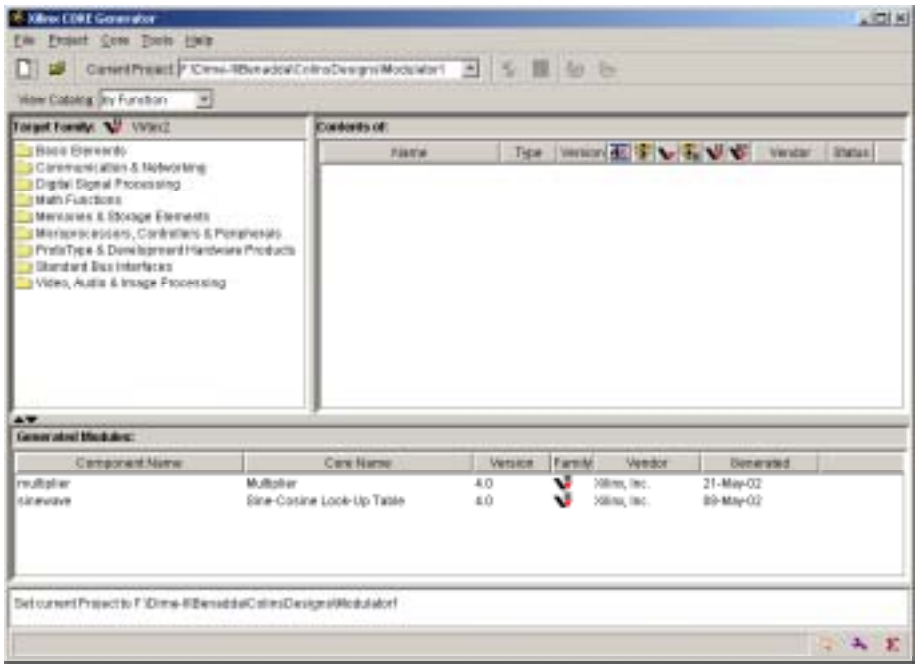


**Figure 45: Xilinx Core Generator Window**

## Sine wave component

To create the sine wave generator component, in the **Target Family** window double click on the 'Math Functions' folder. Click on the sub-folder 'Sine-Cosine Look-up Table' which displays the 'Sine-Cosine Look-upTable' component as shown in Figure 46:

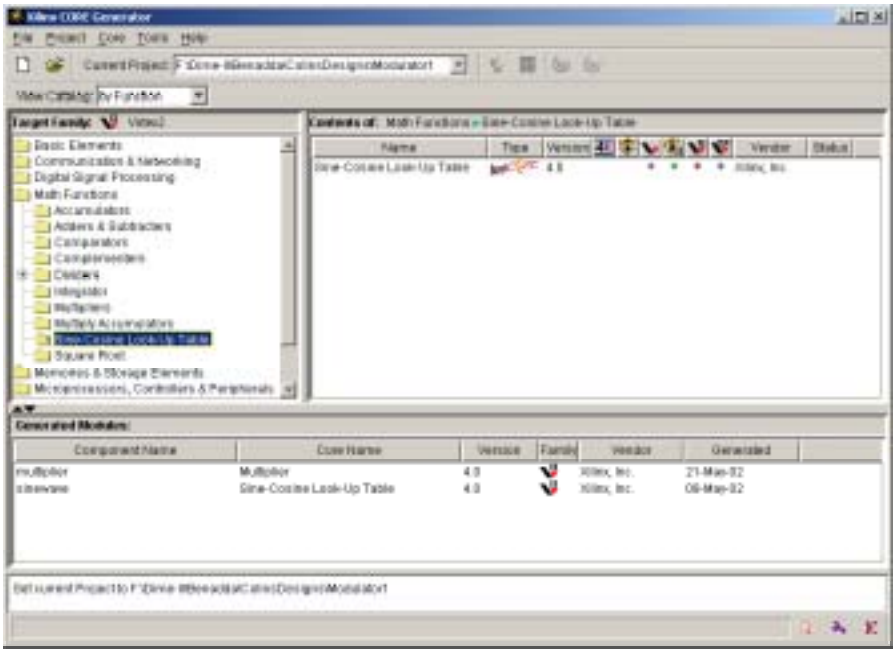


**Figure 46: Math Functions Folder**

To configure this component, double-click the name in the top right window to bring up a dialogue box. From here it is also possible to bring up the component's data sheet which explains general operation and customisable options. Figure 47 shows the options chosen for this design.

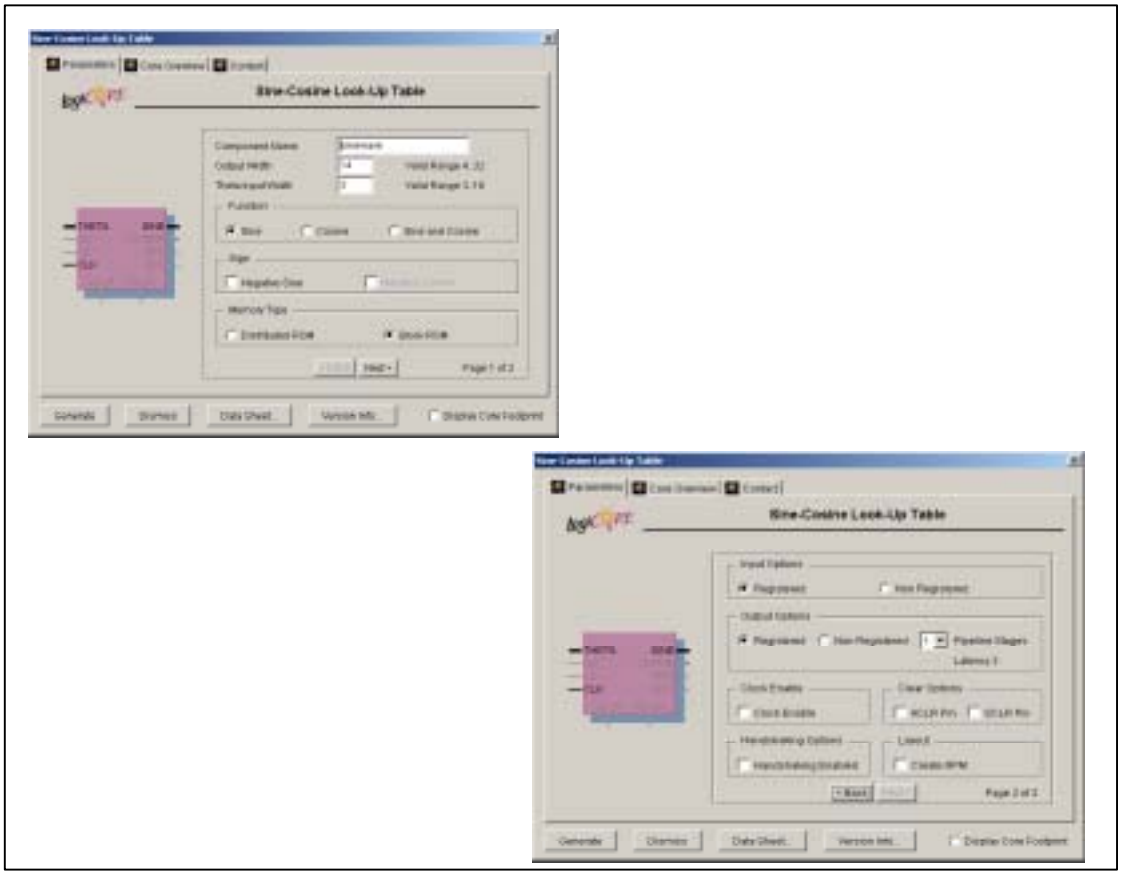



**Figure 47: Sine-Cosine Look-up Table**

Once the options have been selected, click **Generate** to save the net list into the design folder.

## 9.9.2 Multiplier Component

The multiplier component can be created in a similar way. In the **Target Family** window double click the 'Math Functions' folder and select the sub-folder 'Multipliers'. Then double click the component 'Multiplier' in the **Contents of** window, as shown in Figure 48.

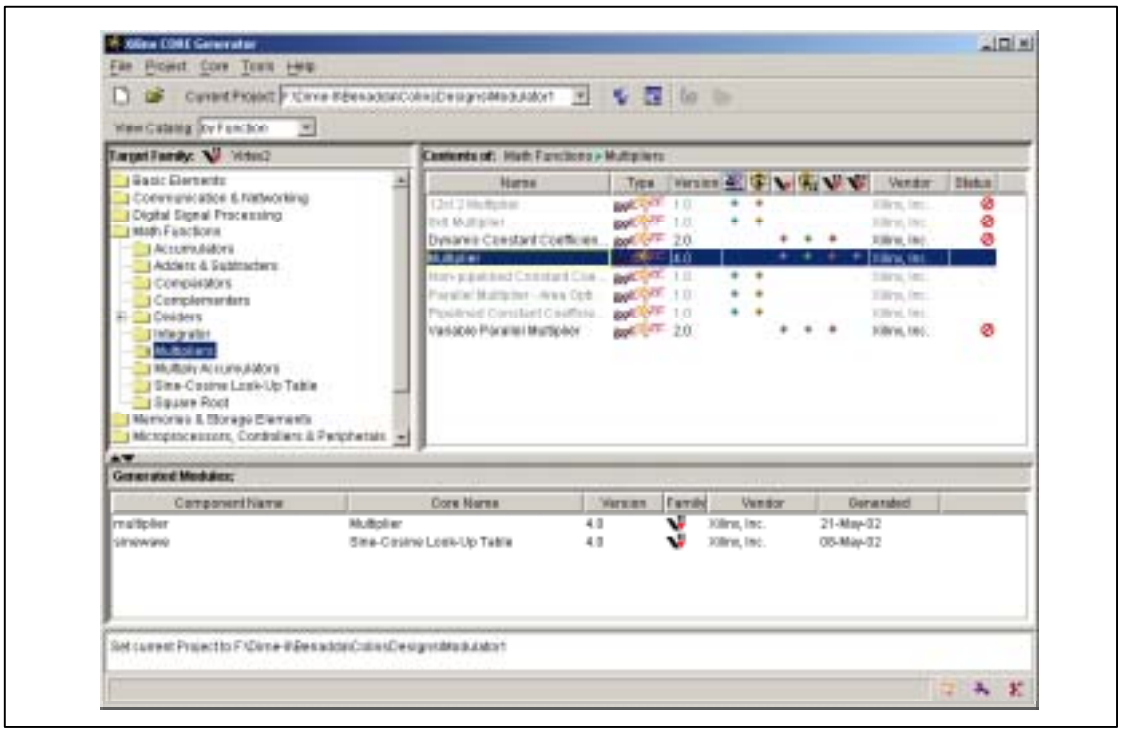


**Figure 48: Multiplier folder**

Configure this component as shown in Figure 49:

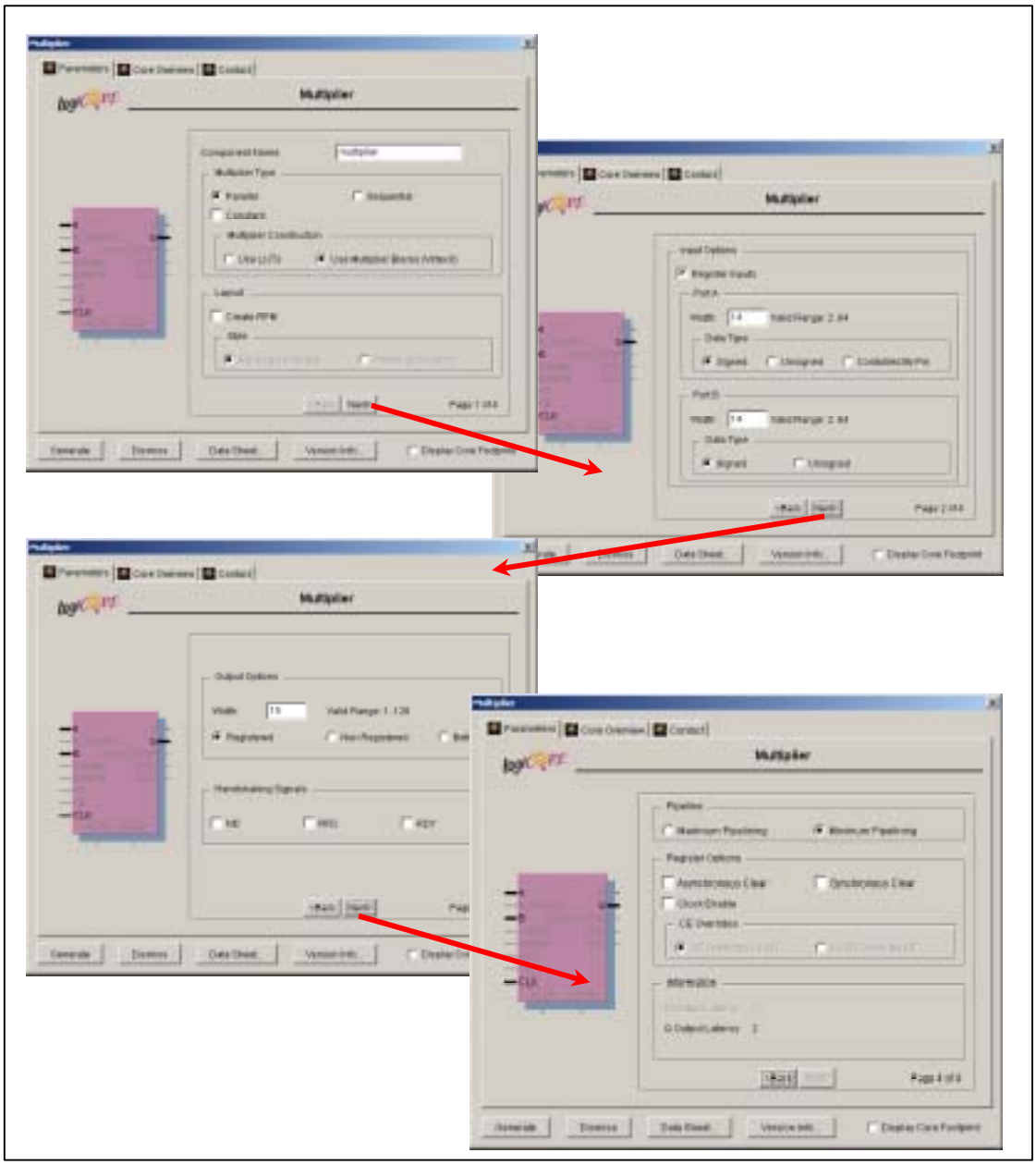



**Figure 49: Multiplier Configuration Settings**

After choosing the appropriate settings click **Generate** to save the net list into the design folder.

When the sine wave and multiplier components have been created, close Xilinx Core Generator and return to Xilinx ISE. To declare and instantiate these components in the code click the light bulb in the top right toolbar to bring up the language templates window. In the language templates window expand the Coregen folder and then expand the relevant language folder - VHDL or VERILOG. Within each of these folders there is a template for each component, which contains a declaration and an instantiation. These can simply be copied and pasted into your code.

## 9.10 Configuring the modulator example

The card can be opened and configured in the same way as described for the ADC to DAC feed-through design on page 82.

- The bit file for the Clock FPGA is located at the path '<CDROM Drive>:\Examples\Clock_Designs\PCB_V2\osc_clock_2v80.bit' on the XtremeDSP Development Kit CD.
- The bit file for the Main FPGA can be found at the path:'<CDROMDrive>:\Examples\Modulator\DSP_Kit\'

Please note that the BenADDA supports two variants that support different FPGA packages. One supports a FF1152 and one a FG676. Note that the XC2V3000 Virtex-II part is supported in both packages. Therefore care should be taken when selecting bitfiles for configuration that they are indeed for the correct target FPG package.

These bit files have the naming convention 'modulator_2v####****.bit' where #### represents the device size and **** represents the package. Ensure the device is configured with the correct bitfile to avoid any damage to the FPGA. After both devices have been configured select the reset tab in the bottom left of the main screen. You should tick both reset check boxes and then un-tick both boxes to carry out a full reset on the system.

With Engineering Silicon XC2V3000 FPGAs, configuration procedures should be carried out twice in order to successfully start-up.

A LED flash pattern should now be displayed on the LEDs and any data fed into the ADC should be modulated and seen at the output of the DAC.

# Example Application 3

In this Section:

- SysgenQAM16 Output to DACs

## 9.11 Introduction

The model is simply a variant on the 'sysgen_qam16' model that is included with the demos supplied with System Generator 2.2. The only modifications are the inclusion of a System Generator set of blocks to produce a sample QAM source for demonstration purposes and connecting the output from the carrier recovery stage to the DACs on the XtremeDSP kit. The example is stored in '<install_folder>\examples\sysgen\qam\full_model'.

Please note that the BenADDA supports two variants that support different FPGA packages. One supports a FF1152 and one a FG676. Note that the XC2V3000 Virtex-II part is supported in both packages. Therefore care should be taken when selecting bitfiles for configuration that they are indeed for the correct target FPG package.

## 9.12 Functional description

The model is shown below in Figure 50:

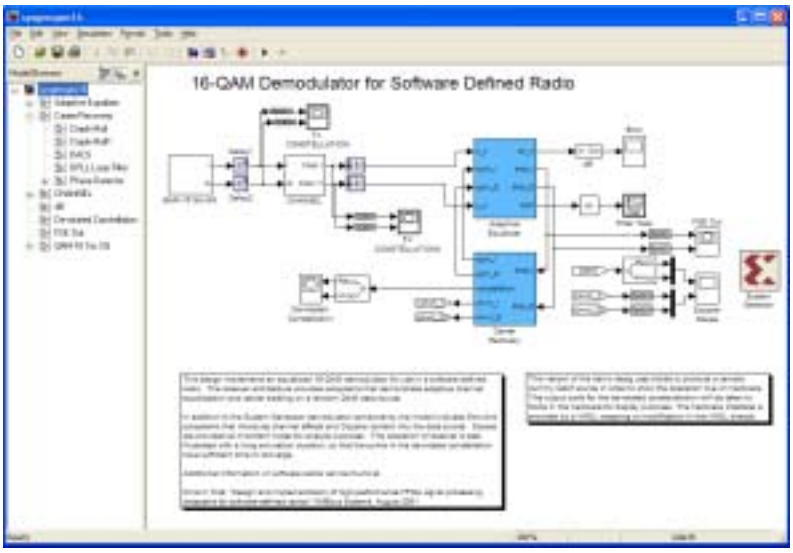



**Figure 50 : Modified sygen_qam16.mdl**

The DACs are connected as shown from the output of the carrier recovery stage:

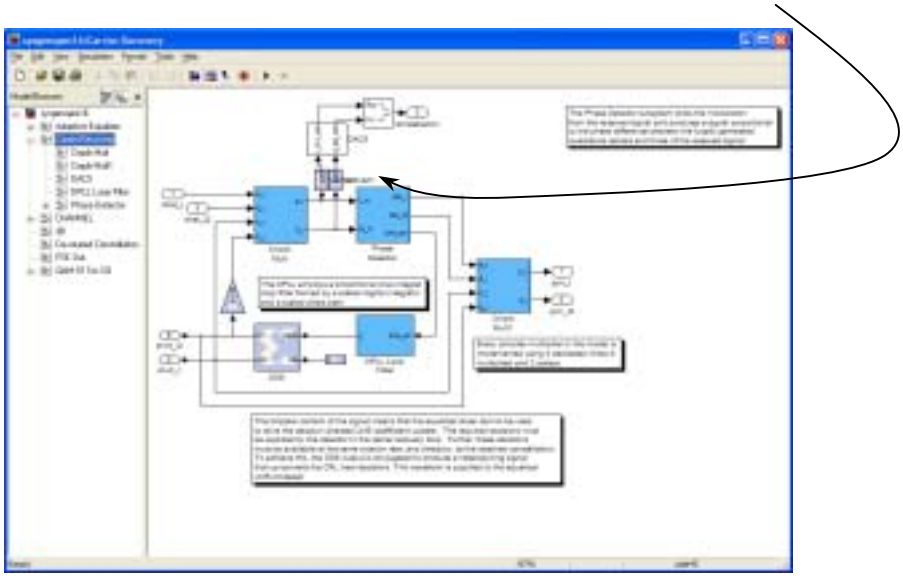


**Figure 51: Modified Carrier Recovery Subsystem**

The internal of the DACs subsystem is shown in Figure 52. Please note that the actual output from the System Generator design is signed 2's complement, but the DACs interpret an input value of 0 to be –1V and $2^{14}$ to be 1V. Therefore, the conversion is necessary.

Apart from this conversion there are other control signals which are used to set the DACs operation mode, such as the DIV and MOD control signals. Details of these options are given in Section 10.3.2, 'PLL Clock Multiplier' on page 101.

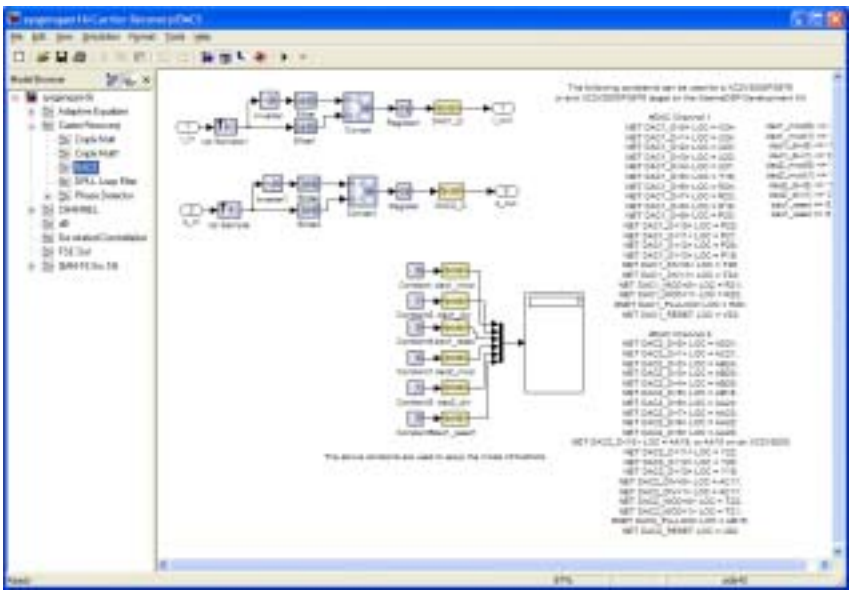


**Figure 52: DACs Subsystem**

## 9.13 Running the implementation

The design can be run on hardware by downloading the appropriate bitfiles to the FPGAs. This is done using the FUSE Probe utility. Open the card, assign bitfiles and then configure the module.

Please note that the BenADDA supports two variants that support different FPGA packages. One supports a FF1152 and one a FG676. Note that the XC2V3000 Virtex-II part is supported in both packages. Therefore care should be taken when selecting bitfiles for configuration that they are indeed for the correct target FPG package.

The bitfiles that need to be assigned are :

(1) osc_clk_2v80.bit or osc_clk_2v40.bit depending upon the clock FPGA fitted on the particular BenADDA.

(2) Sysgen_qam16_2v3000fg676.bit or sysgen_qam16_2v2000fg676.bit again depending upon the particular FPGA fitted.

For convenience these bitfiles are stored in the bitfiles folder in <install_folder>\examples\sysgen\QAM\qam_dplr\bitfiles.

Please note that this design is only currently supplied for the FG676 package devices but it can be ported or used as a basis for a design targeting a FF1152 package FPGA.

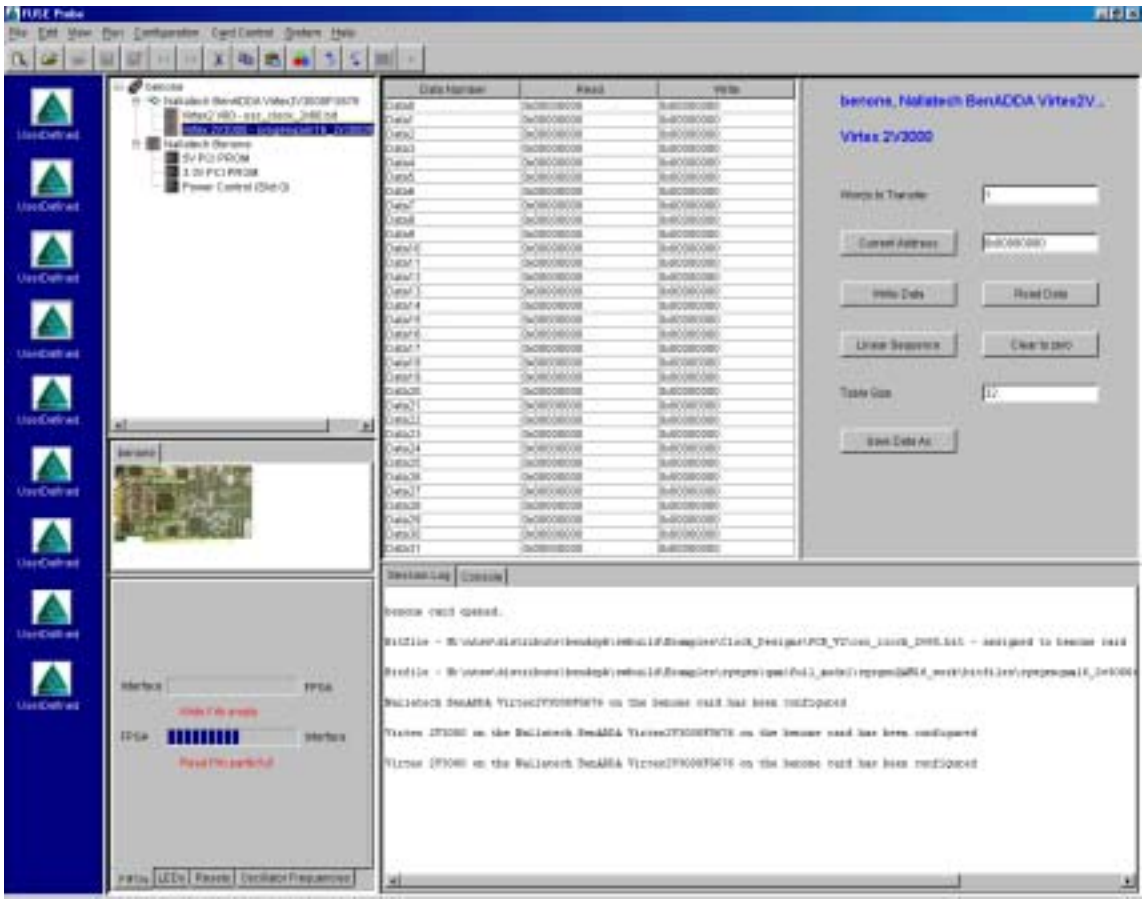


**Figure 53: Assigned Bitfiles in FUSE Probe**

You should see the locked constellation appear almost immediately.

## 9.14 Implementation Source

The full implementation and source files produced are stored in the <cdrom>:\examples\sysgen\QAM\qam_dplr\sysgenQAM16_work.

Please note, in System Generator projects a complete ISE Project Navigator file is available. Also there are two batch files included in the sysgenqam16_work folder namely xc2v3000fg676_qam16_batch.bat and xc2v2000fg676_qam16_batch.bat. These are setup for using xflow to create the implementation file and can be used if necessary to recreate the bitfiles or retarget the implementation as necessary.

## 9.15 UCF Modification

In this implementation there were some small modifications to the UCF produced from System Generator. Namely:

(1) NET "clk" LOC = AB14; #This maps to CLK1_FB
   a. This was added to pin lock the clock signal to the appropriate pin that was being driven by the osc_clk_2v80 design being run in the clock FPGA. This can be changed to another clock input as necessary.

(2) DAC2_D<10> LOC Constraint
   a. needs to be modified depending upon whether a XC2V3000FG676 or XC2V2000FG676 part is being targeted. Please refer to Section 11 'Pinout Information' on page 130 for further details of the specific pins depending upon the device / package.

# Section 10

# Reference Guide

In this Section:

- Physical Layout
- Hardware Features
- Interfacing
- Pinout Information

# Physical Layout

- Physical features of the BenADDA are identified

## 10.1 BenADDA physical layout (top)

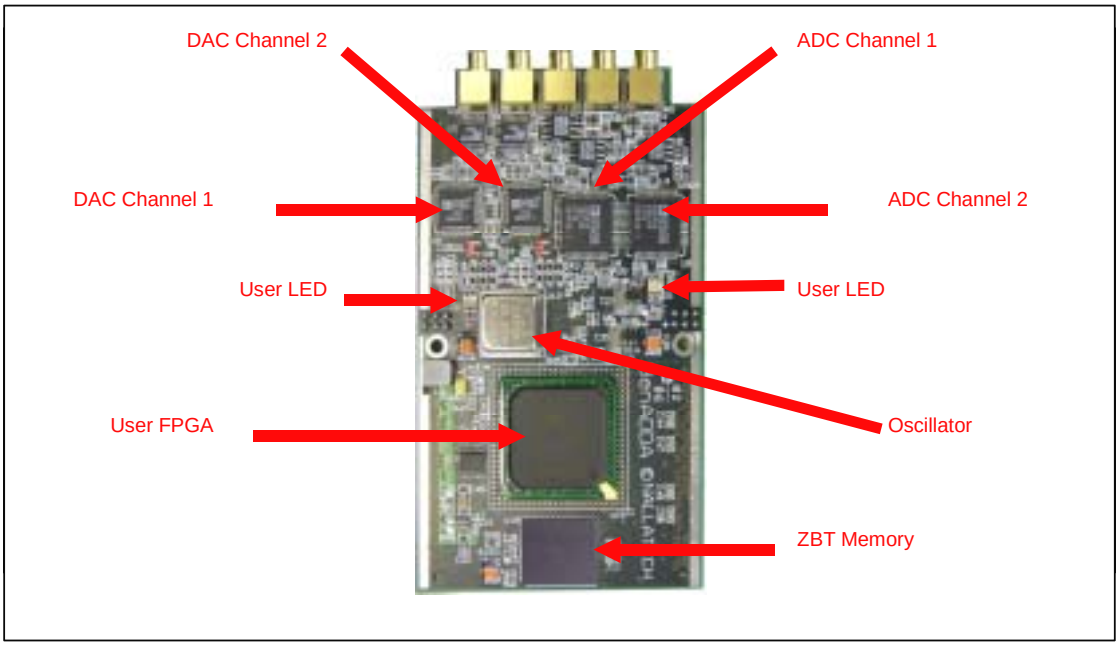



**Figure 54: BenADDA layout (top)**

## 10.2 BenADDA physical layout (bottom)

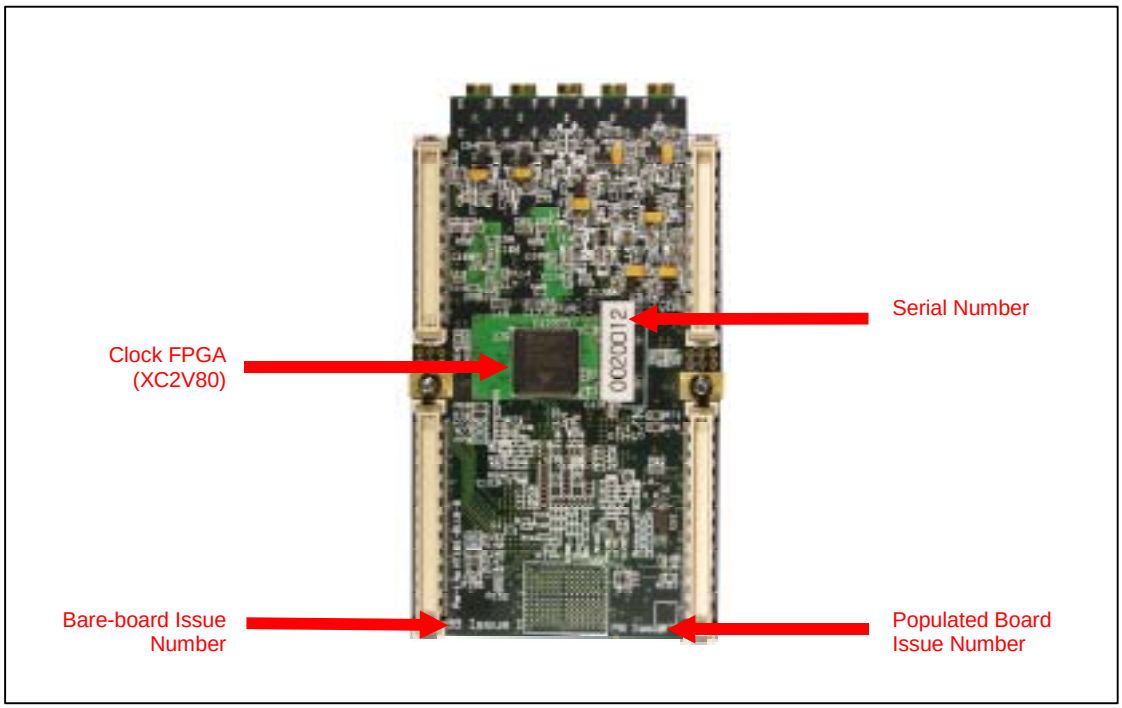



**Figure 55: BenADDA layout (bottom)**

# Hardware Features

- BenADDA Digital-to-Analogue Converter
- DAC Output Configurations
- BenADDA Analogue-to-Digital Converter
- Clocking the DACs and ADCs
- ZBT SRAM Memory
- User FPGA
- Control and Monitoring Signals
- Temperature Sensor

## 10.3 Digital-to-Analogue Converter

The BenADDA has two analogue output channels, each provided with independent data and control signals from the FPGA. Two sets of 14-bit wide data are fed to the two AD9772A devices, each of which has an isolated supply and ground plane. The diagram below illustrates the interfacing between one of the DACs and the FPGA:

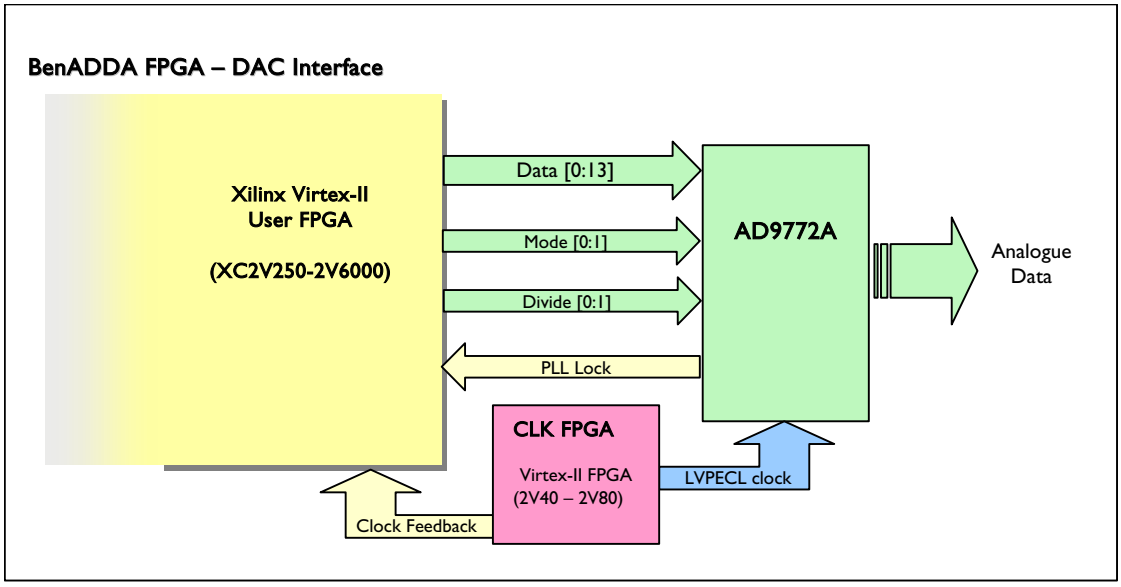



**Figure 56: Functional Diagram of DAC Interface**

The AD9772A device offers 14-bit resolution and a sampling rate of 160MSPS. Additional control signals exist between the DAC and the FPGA to enable full control of the DACs' functionality.

The main features of the AD9772A are:

- 14-bit DAC resolution, note that they are offset-binary input.
- 160MSPS input data rate

- LVPECL clock inputs
- Internal Phase-Locked Loop (PLL) clock multiplier
- Differential current outputs
- Choice of two output configurations (build option)

Please note that the input to the DACs is offset-binary format. This means that a 0 value input should give an approx –1Volt output and a $2^{14}$-1 value input should give an approx +1V output . Note that the input to the DACs is therefore not two's complement.

The BenADDA supports two output configurations – a single-ended DC coupled output, and a differential directly coupled output. These are both described later in this Section. The internal PLL clock multiplier of the AD9772A is also described to provide you with an insight into the internal operations of the DAC.

## 10.3.1 DAC Architecture

The AD9772A's architecture comprises four key areas as shown below:

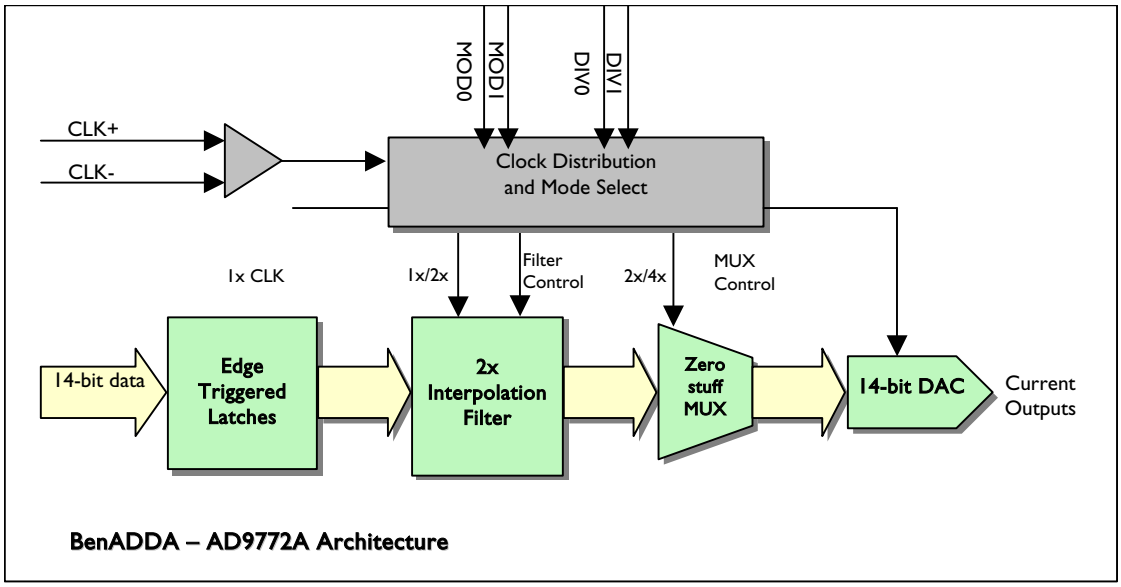



**Figure 57: AD9772 Architecture**

Figure 57 shows the internal architecture of the AD9772A. Initially, the user feeds the14-bits of data into the AD9772A. This data is latched into edge-triggered latches on the rising edge of the reference clock, interpolated by a factor of 2 by the digital filter, and then fed to the 14-bit DAC. The filter characteristic can be set to either low pass or high pass for baseband and IF applications respectively. The MOD0 input is used to control this function of the AD9772A.

The interpolated data can feed the DAC directly or undergo a "zero-stuffing" process, enabled using MOD1. This process involves inserting a mid-scale sample after every data sample originating from the digital filter, which improves the passband flatness of the DAC and also allows for the extraction of higher frequency images.

The AD9772A generates a variety of clock frequencies to operate its elements at the correct rates. To achieve these frequencies, it utilises an internal PLL whose VCO can generate clock rates of up to 400MSPS. The AD9772A can be operated with the PLL enabled or disabled (both operations are supported on the BenADDA).

## 10.3.2 PLL Clock Multiplier

Figure 58 illustrates how the BenADDA supports having the PLL enabled or disabled.

**NOTE -** PLL is enabled by default. Please contact Nallatech to disable the PLL function.

To supply the PLLVDD pin, you can populate a jumper to supply the pin with a 3.3v signal, or else tie the pin to ground. The supply for the PLLVDD pin is shared with the supply of the CLKVDD pin and both the CLK and PLL grounds on the chip share the same separate ground plane. When the PLL is disabled and the PLLVDD pin is tied to ground, the filter components for the LPF pin (internal loop filter for the PLL) on the DAC are not populated. This leaves the LPF with an open connection.

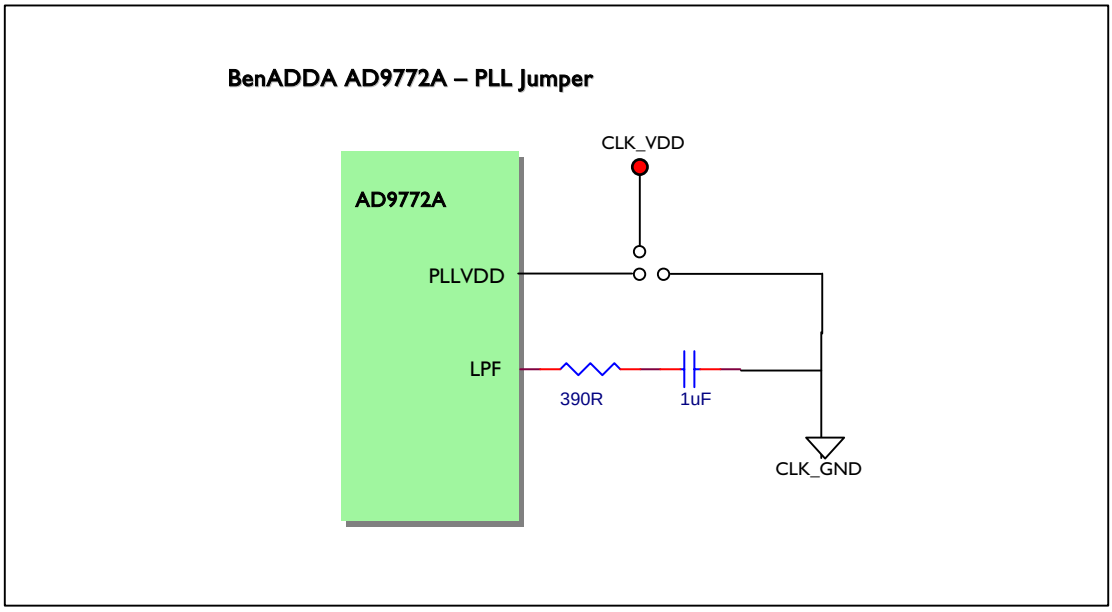



**Figure 58: PLL Jumper Option**

When the PLL is set to the default value *enabled,* the AD9772A will generate its own 2x clock from the reference clock. This allows you to transmit data at the same rate as the reference clock. The internal PLL can also generate another phase clock that allows the zero-stuffing option to be selected under the same circumstances.

If the PLL is *disabled*, then the input data rate must be half the reference clock frequency. This is due to the interpolation filter that adds extra samples every other clock cycle. Additionally, if the zero-stuffing option is selected, then the input data must be one quarter of the reference clock frequency. For example, the maximum reference clock of 160MSPS, with the PLL disabled and the zero-stuffing option selected, gives a maximum input data rate of 40MHz.

The internal PLL also deals with the phase relationship between the data and the reference clock. This means that when PLL is *enabled*, you do not need to use the RESET input to ensure correct alignment of clock and data.

If the PLL is *disabled*, consult the Analog Devices AD9772A datasheet provided on the BenADDA installation CD for more information on how the RESET input is used to ensure correct synchronisation. See Table 15 below for a summary of the DAC input data rates:

| | PLL Disabled | | PLL Enabled | |
|---|---|---|---|---|
| | Zero-stuffing OFF | Zero-stuffing ON | Zero-stuffing OFF | Zero-stuffing ON |
| **Input Data Rate** | ½ reference clock | ¼ reference clock | 1x reference clock | 1x reference clock |

**Table 15: DAC Input Data Rates**

As outlined earlier in this Section, the interpolation filter can be set to either a low or high pass characteristic, depending on whether you wish to capture baseband or IF signals. On the BenADDA, this feature of the AD9772A, and also the zero-stuffing option, is controlled by the FPGA (see **NOTE** below). The operation of the 'MOD' pins is summarised in Table 16:

| Digital Mode | MOD0 | MOD1 | Digital Filter | Zero-Stuffing |
|---|---|---|---|---|
| Baseband | 0 | 0 | LOW | NO |
| Baseband | 0 | 1 | LOW | YES |
| Direct IF | 1 | 0 | HIGH | NO |
| Direct IF | 1 | 1 | HIGH | YES |

**Table 16: Controlling Digital Modes of AD9772A**

The AD9772A contains an internal Voltage Controlled Oscillator (VCO), which can operate at up to 400MSPS. To ensure the optimum phase noise and successful "locking" of the PLL, a pre-scalar stage is incorporated to allow the sampling clock to be divided down as required for slower data rates. The divide-by-ratio is selected by the DIV0 and DIV1 inputs, as shown below in Table 17:

| Input Data Rate (MSPS) | MOD1 | DIV1 | DIV0 | Zero-stuffing | Divide-by-N-ratio |
|---|---|---|---|---|---|
| 48-160 | 0 | 0 | 0 | No | 1 |
| 24-100 | 0 | 0 | 1 | No | 2 |
| 12-50 | 0 | 1 | 0 | No | 4 |
| 6-25 | 0 | 1 | 1 | No | 8 |
| 24-100 | 1 | 0 | 0 | Yes | 1 |
| 12-50 | 1 | 0 | 1 | Yes | 2 |
| 6-25 | 1 | 1 | 0 | Yes | 4 |
| 3-12.5 | 1 | 1 | 1 | Yes | 8 |

**Table 17: Recommended Presale Ratio Settings**

The DIV0 and DIV1 signals are controlled by the on-board FPGA (see **NOTE**).

A PLL_LOCKED signal from each AD9772A is connected to the FPGA (see **NOTE**) on the BenADDA. This signal goes high to indicate that the PLL has "locked" to the input reference clock. If the PLL is not locked, due to the PLL being disabled or an unstable clock, PLL_LOCKED toggles between high and low in an asynchronous manner.

**NOTE**: When a XC2V250 FPGA is populated on the BenADDA, the signals MOD, DIV and PLLLOCK are not available for use via the FPGA. Please refer to 'XC2V250 FPGA DAC Control Signals' below, for details on how these control signals are connected for use.

For more information on the zero-stuffing process, interpolation filter and all aspects of the AD9772A device, refer to the XtremeDSP Development Kit CD.

### 10.3.3 XC2V250 FPGA DAC Control Signals

The following Section is only appropriate to BenADDA's that have a XC2V250 on-board User FPGA.

When an XC2V250 FPGA is populated on the BenADDA, there is a physical limit to the number of User IO available, which results in certain control and monitoring signals of the on-board DACs being unavailable for use through the FPGA. The signals no longer accessed through the FPGA are:

- MOD0, MOD1
- DIV0, DIV1
- PLL_LOCKED

However, the DACs can still be fully utilised by various jumpers. The location of these jumpers on the PCB is shown below in Figure 59:

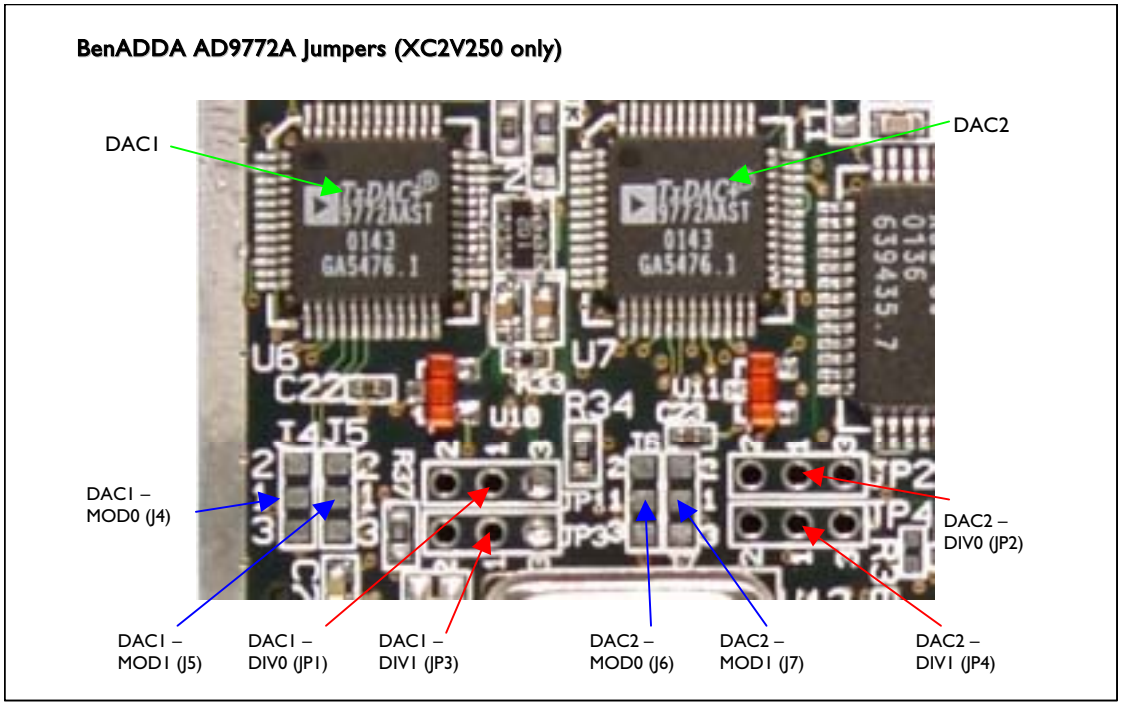



**Figure 59: Physical Location of Jumpers for DAC (XC2V250 only)**

#### MOD pins – XC2V250 only

When the on-board Virtex-II is a XC2V250 device, a fixed jumper controls the MOD pins. The MOD pins are fixed either HIGH or LOW at the time of production using a 0-ohm resistor connected to either GND (LOW) or VCCO_IO (HIGH).

DAC1 *(located at edge of the PCB)* has the MOD pins controlled by fixed jumpers:

- J4 for MOD0
- J5 for MOD1

DAC2 *(the inner DAC on the PCB)* has the MOD pins controlled by fixed jumpers:

- J6 for MOD0

- J7 for MOD1

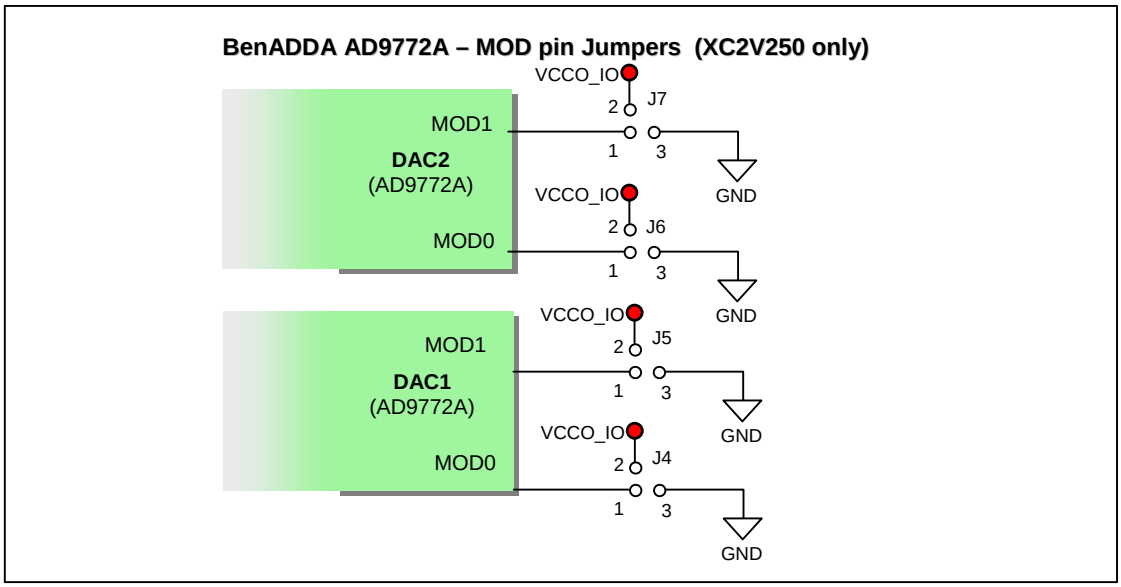



**Figure 60: DAC: Diagram of MOD fixed jumpers (2V250 only)**

As shown in Figure 60, all MOD jumpers have pin 2 connected HIGH and pin 3 connected LOW. So if a 0-ohm resistor is placed between pads 1 and 2 of a jumper, that particular MOD pin is set HIGH. Alternately, a 0-ohm resistor placed between pads 1 and 3 would set the MOD pin LOW.

## DIV pins – XC2V250 only

When the on-board Virtex-II is a XC2V250 device, a user-selectable jumper controls the DIV pins. The DIV pins are selected either HIGH or LOW by the use of a jumper connector.

The DIV pins are set either to logic high or low by connecting two pins of the 3-way jumper together.

DAC1 *(located at edge of the PCB)* has the DIV pins controlled by selectable jumpers:

- JP1 for DIV0
- JP3 for DIV1

DAC2 *(the inner DAC on the PCB)* has the DIV pins controlled by selectable jumpers:

- JP2 for DIV0
- JP4 for DIV1

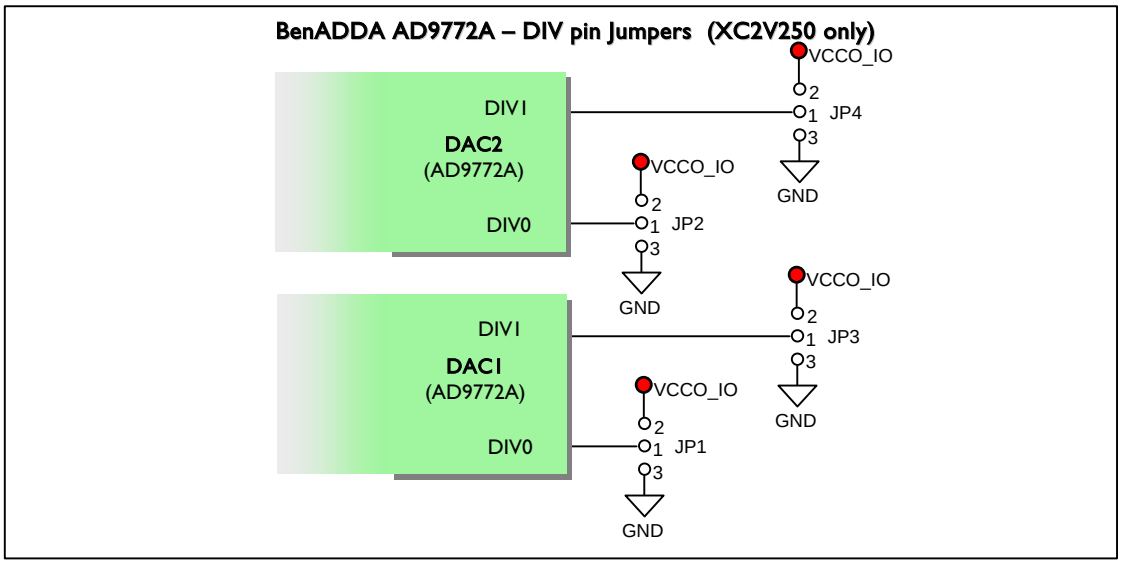



**Figure 61: DAC: Diagram of DIV selectable jumpers (2V250 only)**

Figure 61 shows that all DIV jumpers have pin 2 connected HIGH and pin 3 connected LOW. This allows you to manually choose between setting 1 and 2 (DIV High), or between 1 and 3 (DIV Low).

## PLL-LOCK pins – XC2V250 only

When the on-board Virtex-II is a XC2V250 device the PLLLOCK signal is unavailable at the User FPGA, therefore a LED is tied to the PLLLOCK signal to indicate its current operation. Figure 62 shows the set-up of the LED:

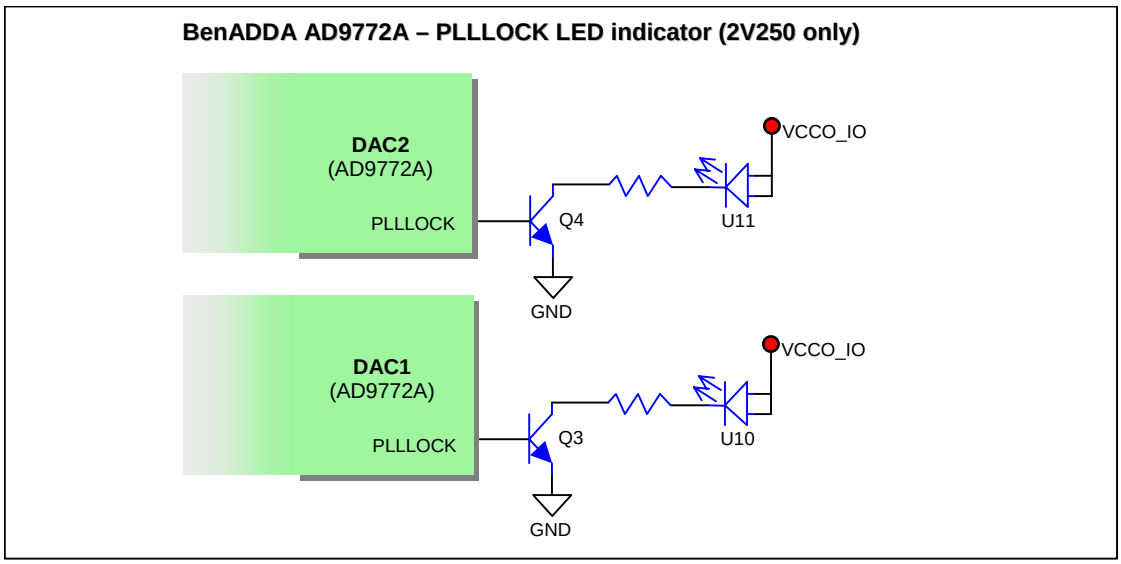



**Figure 62: PLLLOCK LED indicator (2V250 only)**

When a XC2V250 FPGA is populated the LED is switched on when PLLLOCK pin is HIGH. For example, when PLLLOCK is sending out a HIGH signal it will turn on the base of the transistor and the LED will light. So when PLLLOCK is sending out a LOW signal, the LED will be switched off.

For full details on PLLLOCK status for a specific design, please refer to the AD9772A Analog Devices datasheet provided on the BenADDA installation CD.

## 10.4 DAC Output Configurations

The AD9772A DAC supports two output configurations. The following are supported as build options on the BenADDA:

- Single-ended output, DC-coupling using an op-amp
- Differential outputs using termination resistors

### 10.4.1 Single-Ended DC-Coupling Using an Op-Amp

The op-amp configuration is useful for applications requiring DC coupling. Figure 63 illustrates the set-up adopted to use an op-amp configuration at the output of the AD9772A:

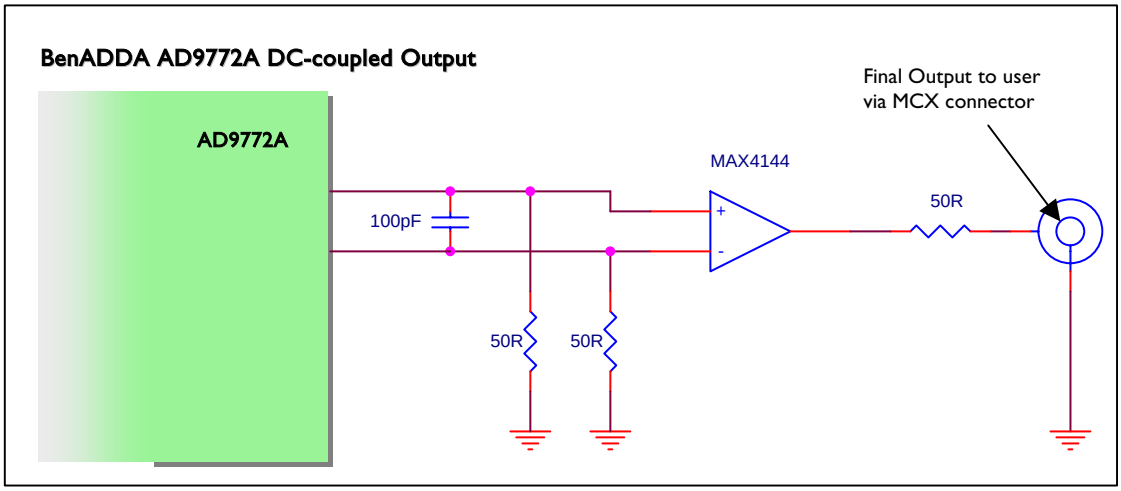



**Figure 63: AD9772A DC-coupled Output**

The op-amp used in the design is an instrumentation amplifier with a X2 gain from MAXIM. This means that the output configuration will drive an output voltage of ±1V into a 50Ω load.

### 10.4.2 Differential Outputs using Termination Resistors

It is also possible to drive differential outputs using a pair of termination resistors:

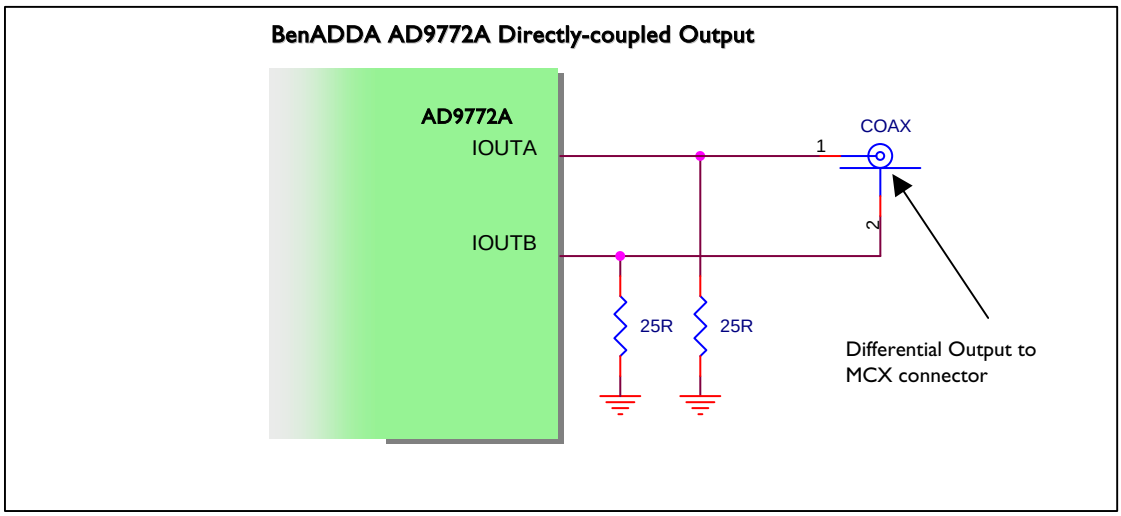


**Figure 64: AD9772 directly coupled option**

When terminated into a 50Ω load, this option will provide a fully differential, 0.5Vp-p output signal swing.

### 10.4.3 DAC Clocking

Each DAC device is clocked directly by an independent differential LVPECL signal. This LVPECL signal is driven from a second on-board FPGA (CLK FPGA). The second FPGA is solely dedicated to managing the clocking methods for each DAC and ADC device. The speeds at which the DACs are clocked depend on what bit file is assigned to this dedicated CLK FPGA.

Please refer to Section 10.6 – 'Clocking the DACs and ADCs (CLK FPGA), for a full description of the clocking options available.

## 10.5 Analogue-to-Digital Converter

Throughout this Section, all diagrams and text will refer to the AD6644. However this information is also applicable to the AD6645. The main difference between these devices is the sampling rate - AD6644 samples at 65MSPS, AD6645 samples at 80MSPS or 105MSPS.

The BenADDA has two analogue input channels, with each channel providing independent data and control signals to the FPGA. Two sets of 14-bit wide data are fed from two AD6644 devices, each of which has an isolated supply and ground plane. Figure 65 illustrates the interfacing between one of the ADCs and the FPGA:

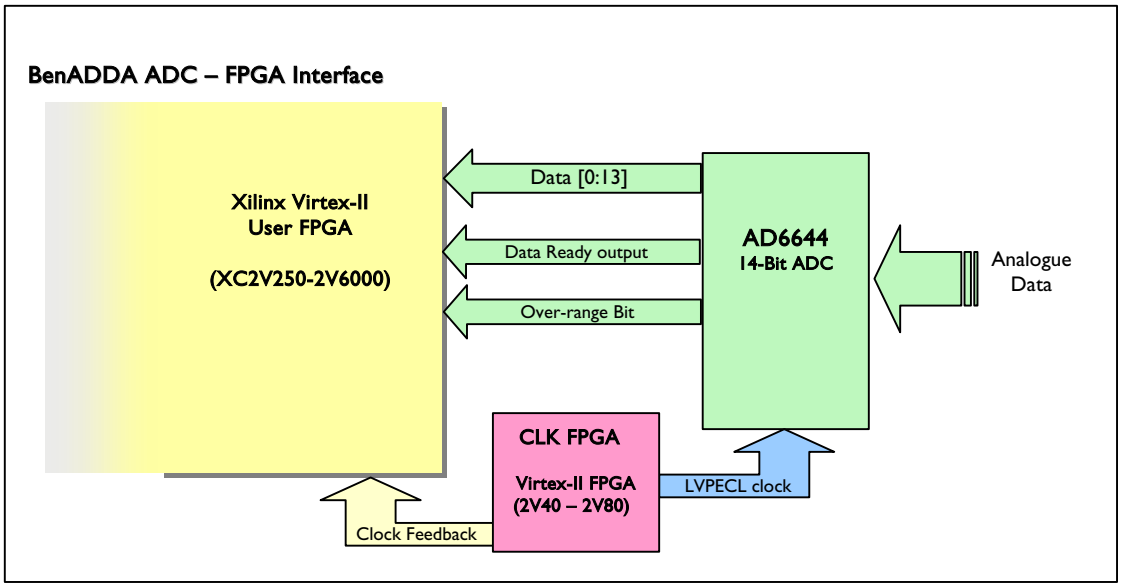



**Figure 65: Functional Diagram of DAC Interface**

The main features of the on-board ADC channels are:

- 14-bit ADC resolution, 2's complement format.
- 65MSPS sampling data rate (sampling rate up to 105MSPS if using AD6645)
- Differential OR Single-ended Analogue Inputs
- 3rd order filter on Analogue Inputs
- ADCs clocked differentially

The BenADDA uses ADCs from Analog Devices - the AD6644 or AD6645. The datasheets for these ADCs can be found on the BenADDA installation CD.

### 10.5.1 ADC Architecture

The AD6644 is straightforward to operate – the user is only required to apply data and a clock input. There are no set-up or control signals as with the DACs. Shown below is the functional diagram of the AD6644 architecture:

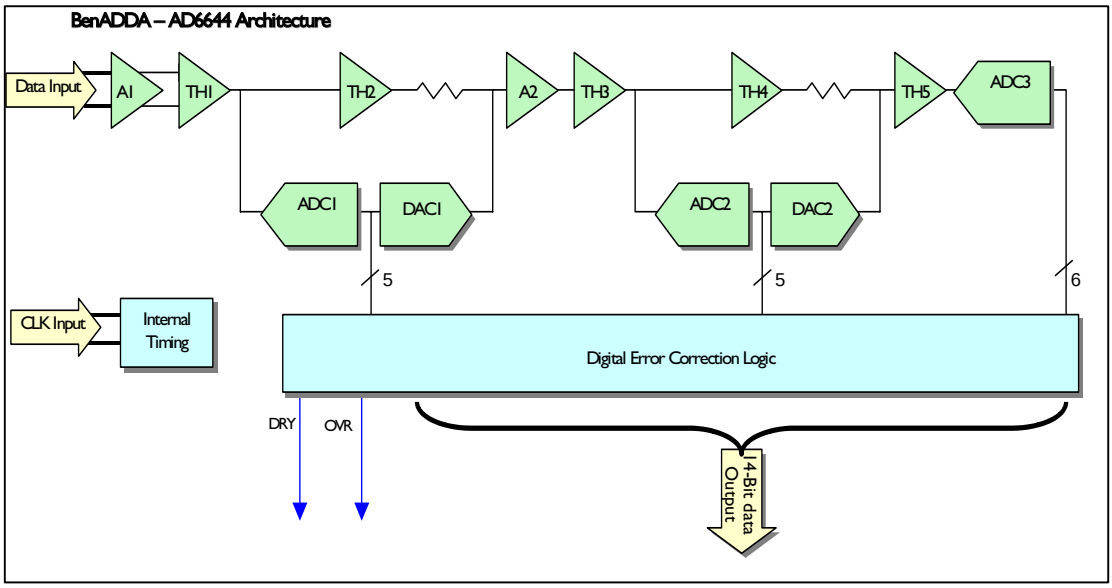



**Figure 66: AD6644 Architecture**

## Theory of AD6644 operation

The AD6644 has complementary analogue inputs; each input is centred at 2.4V and should swing +/ -0.55V around this 2.4V reference. This means that the differential analogue input signal will be $2.2V_{pp}$ as both input signals (AIN and AIN#) are 180 degrees out of phase with each other.

When data arrives at the AD6644, both analogue inputs are buffered prior to the first track-and-hold (TH1). The analogue signals are held in TH1 while the ENCODE (CLK) pulse is high and then data is applied to the input of a 5-bit coarse ADC (ADC1). The digital output of ADC1 is fed into the 5-bit DAC1. The output from the DAC1 is subtracted from the delayed analogue signal at the input of TH3 to generate a first residue signal. The purpose of TH2 is to provide a pipeline delay to compensate for the digital delay of ADC1.

This first residue signal is then applied to the second conversion stage. Again a similar process is achieved through this stage, which finally leads onto obtaining a second residue signal that is applied to a third 6-bit ADC. Finally the digital outputs of ADC1, ADC2 and ADC3 are added together and corrected in the digital error correction logic to generate the final output.

## 10.5.2 Analogue front-end Input

The BenADDA has been designed to take either single-ended or differential analogue inputs. This feature is a build option and is specified at the time of ordering.

The analogue input signal is dc-coupled through a differential op-amp, AD8138, which is fed into the AD6644. The op-amp has been configured to support either single-ended or differential inputs and will always output a differential signal. This means that all data will be input to the AD6644 differentially which helps to reduce noise induced on the input signal. The BenADDA has also been designed with a 3rd order filter on the front end of the AD6644. Again this filter helps to reduce the overall noise induced on the input signal, thereby improving the resolution at the output of the AD6644.

## 10.5.3 ADC Clocking

As with the DACs, each ADC device is clocked directly by an independent differential, LVPECL signal. This LVPECL signal is driven from a second on-board FPGA (CLK FPGA), solely dedicated to managing the various methods for clocking each ADC and DAC device. The speed at which the ADCs are clocked is dependent on what bit file is assigned to this dedicated CLK FPGA.

Please refer to Section 10.6 – 'Clocking the DAC and ADC devices', for a full description of the clocking options available.

# 10.6 Clocking the DACs and ADCs (CLK FPGA)

## 10.6.1 Overview of various Clocking Methods

The populated BenADDA has a total of 6 methods for clocking the DACs and ADCs:

- User external Clock source via MCX connector
- On-board Oscillator or 2nd External Clock source via MCX connector (This is a special build option and can be chosen instead of the on-board oscillator)
- 4 User programmable clocks from on-board User FPGA (Can either be used as 4 single-ended signals OR 2 differential signals).

All the above clock signals are fed directly into a second on-board FPGA (CLK FPGA). This CLK FPGA is a Virtex-II XC2V80 or XC2V40 and is solely dedicated to the use of clocks. Figure 67 gives an overview of the CLK FPGA:

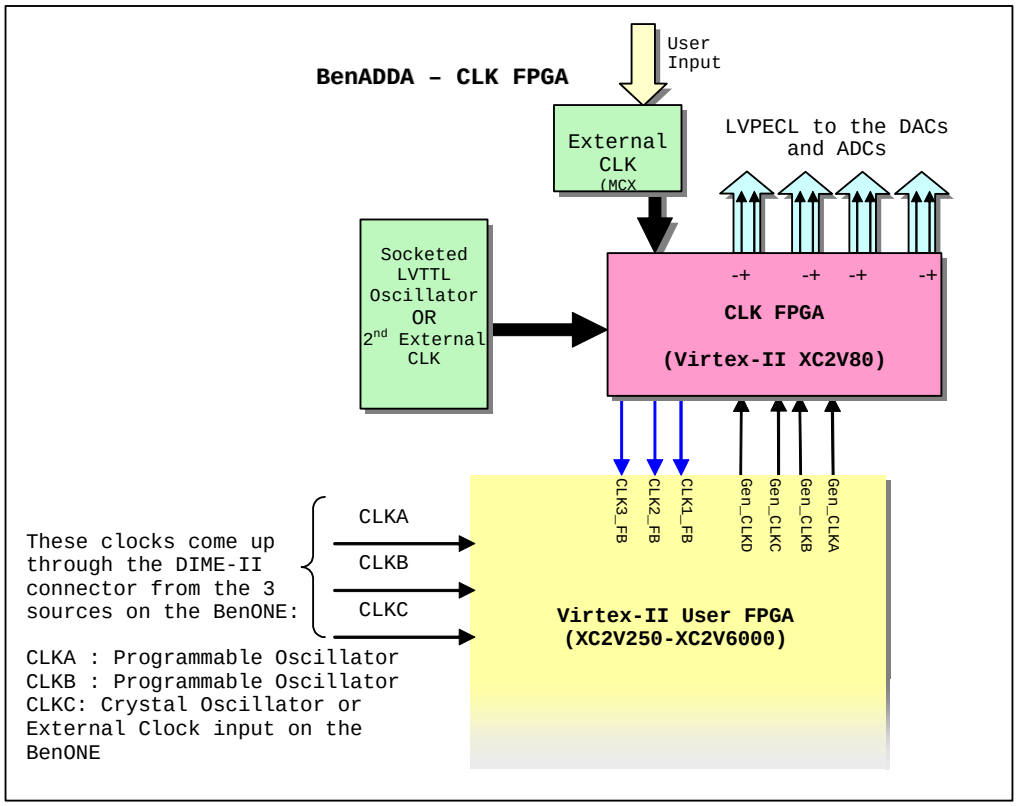



**Figure 67: Overview of CLK FPGA**

The are two example clock designs included on the XtremeDSP CD in the following folder <CDROM>:\Examples\Clock_Designs\PCB_V2\Source. One VHDL file shows how to use the crystal osicllator source on the BenADDA to clock the ADCs and DACs and the User FPGA. The second example shows how to use the external clock input to the BenADDA.

## 10.6.2 External Clock Source

### Using a clock source centred around 0V (Default Build)

The BenADDA provides a facility for an external clock source to drive the CLK_FPGA. In order to meet the signal input specifications for the CLK_FPGA, an op_amp is used to provide DC biasing to level shift the input signal above 0V.

At the heart of the external clock circuit is the AD8131 Differential Driver. This converts single-ended inputs into differential outputs suitable for the CLK_FPGA. The AD8131 has internal feedback with a fixed gain of 2, which allows for better thermal matching and tolerance levels. The common-mode level of the differential output is set by VOCM, thereby level shifting the input signal suitable for driving the CLK_FPGA.

VOCM is set at 1.25V to comply with the typical VICM value for the Virtex-II FPGA using LVDS voltage specifications. The Differential Driver Output Voltage for LVDS is specified as ±250-450mV (typ. ±350mV). It is therefore advised to limit the magnitude of the clock input signal to 125mVpp-225mVpp.

The maximum input signal the CLK_FPGA can accommodate is 3.3Vpp. In order to ensure that the CLK_FPGA cannot be accidentally blown, the supply of the AD8131 is limited between 0V to 3.3V.

### Using a DC bias Clock Source (Customer Specified Build)

The BenADDA has a second external clock source option that feeds straight to the CLK_FPGA. This is a special build option and must be specified at time of ordering.

This second option will supply an external clock signal straight to the CLK_FPGA. If this option is chosen a signal within the range of 0V to 3.3V MUST BE USED. If a signal lower than 0V or greater than 3.3V is supplied while in this configuration, the Clock FPGA will be damaged. It is recommended that the LVTTL signalling standard be used.

When using this set-up for the external clock circuit, it is still possible to use a single-ended source or a differential source, as long as the signal is within the operating ranges of the CLK_FPGA.

### Pinouts for External Clock Source

| Clock Signal Description | Signal Name | CLK FPGA Pin |
|---|---|---|
| External CLK source via Op_Amp | CLK_Op_Amp | B6 (GCLK6S) |
| Complement of External CLK source via Op_Amp | CLK_Op_Amp# | C6 (GCLK7P) |
| External CLK source straight to CLK FPGA | EXT_CLK | B8 (GCLK0S) |
| Complement of External CLK source straight to CLK FPGA | EXT_CLK# | A8 (GCLK1P) |

**Table 18: Pinout Information for External Clock source**

## 10.6.3 On-board Oscillator

An on-board crystal oscillator that generates an LVTTL clock signal is supplied with the standard build of the BenADDA. The LVTTL clock signal generated by this oscillator is driven directly into the CLK FPGA. This clock signal can then be used to derive the differential clock signals that can be used to clock both the DACs and the ADCs.

The Crystal Oscillator supplied with the BenADDA has a low jitter characteristic and its speed will be matched to the sampling frequency of the ADCs. For the ADC AD6644, a 65MHz crystal oscillator is supplied.

Although the BenADDA is supplied with a crystal oscillator that complements the speed of the ADCs, this part can easily be replaced with an alternative speed rating. The oscillator is fixed onto the BenADDA via sockets and can simply be lifted out. Any Oscillator with

similar characteristics can be brought and placed into the socket pins. The Oscillator supplied with the BenADDA is an 8-pin DIL package from Pletronics. This oscillator is powered from 3.3V and any replacement oscillator should follow the same criteria. Please contact Nallatech for advice on replacing the standard oscillator.

| Clock Signal Description | Signal Name | CLK FPGA Pin |
|---|---|---|
| LVTTL Clock Oscillator | Osc_CLK | M6 (GCLK4P) |

**Table 19: On-board Crystal Oscillator Pinout**

## 10.6.4 Generated Clock signals from User FPGA

Another method of clocking the DACs and ADCs is by using clock signals generated by the User FPGA. Within the User FPGA there are three DIME-II system clocks: CLKA. CLKB and CLKC. See Section 10.8.3, 'DIME-II system Clocks' for more information. Please note that in the XtremeDSP kit only CLKA and CLKB clock sources are programmable and CLKC is connected to a socket for a crystal oscillator.

These system clocks can be used to derive an appropriate clock frequency within the User FPGA and then driven into the CLK FPGA where they can be forwarded out to the appropriate DACs and/or ADCs.

These generated Clock signals are forwarded from the User FPGA to the CLK FPGA as four single-ended signals. From the CLK FPGA, the forwarded clock signals can then be sent out to the DACs/ADCs as differential signals. Table 20 shows the generated clock signals:

| Clock Signal Description | Signal Name | CLK FPGA Pin | User FPGA Pin |
|---|---|---|---|
| Generated Clock A | GEN_CLKA | K7 (GCLK0P) | Consult 'Pinout Information' Section * |
| Generated Clock C | GEN_CLKC | N8 (GCLK1S) | Consult 'Pinout Information' Section * |
| Generated Clock B | GEN_CLKB | M7 (GCLK6P) | Consult 'Pinout Information' Section * |
| Generated Clock D | GEN_CLKD | N7 (GCLK7S) | Consult 'Pinout Information' Section * |

**Table 20: Generated Clock Pinouts**

*The pin locations for the Generated Clock signals are listed for each FPGA option in the following places:

- For an XC2V250 FPGA, see 'Clock sources available at CLK FPGA' on page 134
- For an XC2V1000 FPGA, see 'Clock sources available at CLK FPGA' on page 140
- For an XC2V3000 or XC2V6000 FPGA, see 'Clock sources available at CLK FPGA' on page 168.

## 10.6.5 2nd External Clock source

The BenADDA is designed to provide maximum flexibility. With this in mind there is a special build option available that allows you to have two external clock input sources. However when the option of using both external clocks is chosen, the on-board oscillator is removed from the BenADDA.

This 2nd external clock source is an MCX connector that is populated on the BenADDA in place of the crystal oscillator. The MCX connector is directly connected to the CLK FPGA and either a single-ended or differential clock signal can be provided.

The Clock signal that is provided must be between 0V to 3.3V. Any deviation outside this range will damage the CLK FPGA.

| Clock Signal Description | Signal Name | CLK FPGA Pin |
|---|---|---|
| 2nd External Clock | EXT2_CLK | D7 (GCLK5P) |
| Complement of 2nd External Clock | EXT2_CLK# | A6 (GCLK4S) |

**Table 21: 2nd External CLK FPGA Pinouts**

## 10.6.6 Clock feedbacks for De-skewing

Feedback signals between the CLK FPGA and the User FPGA are necessary to allow all data going to and from the User FPGA to be clocked on the same clock edge as the data in the DACs and ADCs.

There are a total of three feedback pins from the CLK FPGA to the User FPGA. These feedback signals ensure that the clock to the DACs or ADCs and the feedback pins have coincident clock edges with minimum skew.

The feedback signals from the CLK FPGA to the User FPGA are matched in physical length with the clock signals sent to the DACs and ADCs. This design ensures minimum skew between the data clocked through the ADCs/DACs and the data being clocked in the User FPGA. Figure 68 outlines the set-up between the CLK FPGA and User FPGA for these feedback signals. There reason for there only being 3 clock feedback pins is simply sue to clock pin resource constraints.

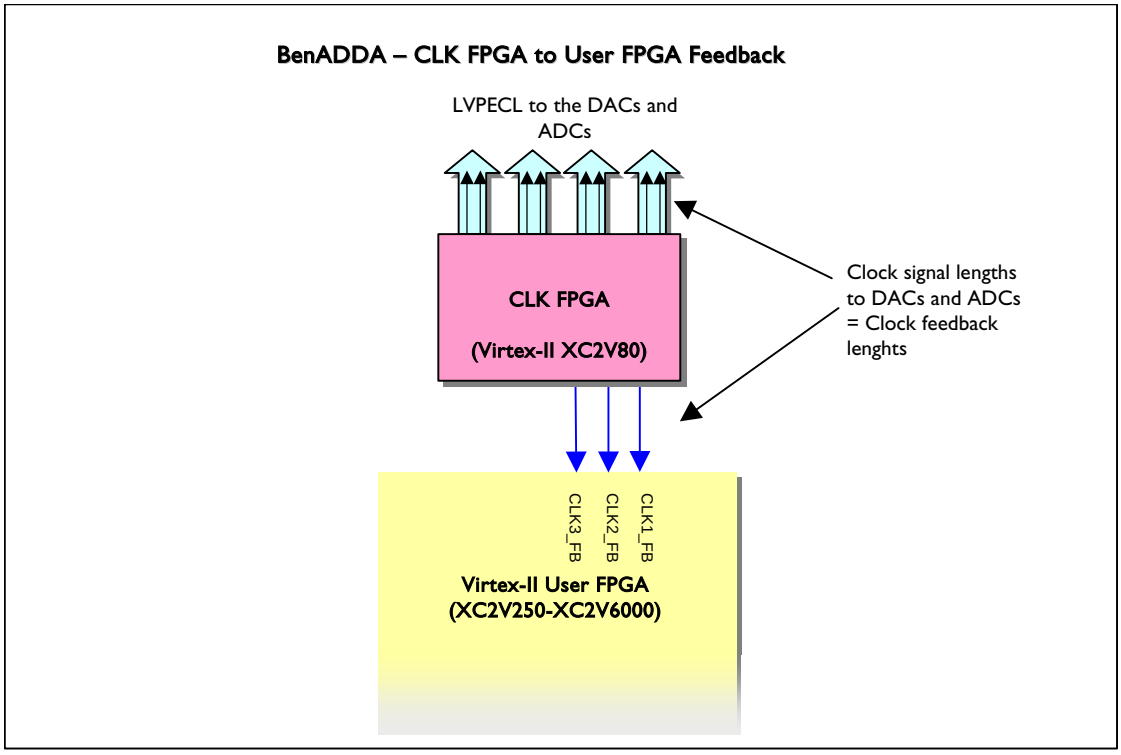



**Figure 68: Diagram of CLK FPGA feedback signals**

## 10.6.7 DAC and ADC clocking

The CLK FPGA is used to directly clock each ADC and DAC device independently. The ADCs and DACs are clocked differentially from the CLK FPGA and can be clocked at various speeds. The speed at which the ADCs and DACs are clocked depends on what bit file is downloaded into the CLK FPGA.

The BenADDA is supplied with various bit files. The user is simply required to choose the Clocking option most suited to the end application.

## 10.7 ZBT SRAM Memory

The BenADDA supports up to four ZBT SRAM devices when a XC2V3000 FPGA or higher is populated. This memory can provide on-board storage capabilities of 4 Mbytes (Higher Density Devices are supported) via a 32-bit data bus. The BenADDA supports various memory sizes, all of which are supplied from Micron in the 165-pin FBGA package. The memory chips are driven exclusively by the USER FPGA. Figure 69 illustrates the inter-connect between one Bank of ZBT SRAM and the USER FPGA.

### 10.7.1 Hardware Details

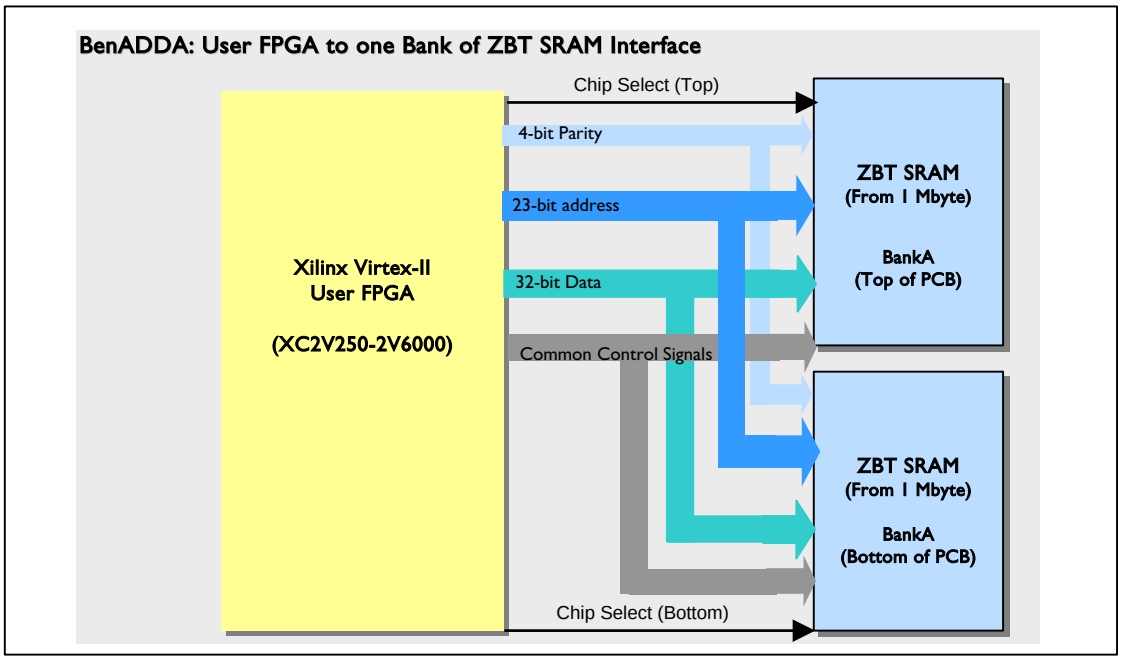



**Figure 69: Functional Block Diagram of ZBT SRAM (BANK A) Interface**

The four devices are split into two independent Banks (Bank A and B). The two devices that make up a single bank share all data, address and control signals, with the exception of the chip select signals. Using the chip select input allows access to each device individually but not simultaneously, which provides you with 32-bit access to 2Mbytes* of memory (twice the depth of one chip).

Each ZBT device integrates from 256K x 32 core with advanced synchronous periphery circuitry and a 2-bit burst counter. The SRAM is optimised for 100% bus utilization, eliminating any turnaround cycles for READ to WRITE, or WRITE to READ transitions. All synchronous inputs pass through registers controlled by a positive-edge-triggered single clock input. The synchronous inputs include all addresses, all data inputs, chip enable, synchronous clock enables, write enables and Read/Write. The asynchronous inputs include the output enable, clock, and snooze enable and a burst mode that can select between interleaved and linear modes.

Each Bank is clocked directly by an independent clock signal from the User FPGA. This means that Bank A has two ZBT devices that share the same clock signal, while Bank B has another separate clock signal shared between its two ZBT devices.

*Access to 2Mbytes of memory, if ZBT chips are 8Mbit devices. So one chip would give 1Mbyte of memory.

The main features of the ZBT Memory include:

- Fast cycle times: 6ns, 7.5ns and 10ns
- 100% bus utilisation
- Advanced control for minimum signal interface
- Single R/W (Read/Write) control pin
- Clock-controlled and registered addresses, data I/Os and control signals
- Common data inputs and data outputs
- Linear or Interleaved Burst modes
- Burst feature (optional)
- Pin/function compatibility with 2Mb, 4Mb, 8Mb, 16Mb, 32Mb, 64Mb & 128Mb ZBT SRAM

## 10.7.2 ZBT SRAM Clocking

The ZBT SRAM is clocked directly by an independent output from the FPGA. Each Bank has its own clock signal that has been de-skewed within the User FPGA. This ensures that the clock at the ZBT SRAM Banks and the feedback pin have coincident clock edges with minimum skew. Ultimately this process ensures that the internal logic is clocked in phase with the data entering the ZBT chips.

Driving the ZBT SRAM clock from the FPGA ensures maximum flexibility in the clocking mechanism during system design, as it can be derived from any of the DIME-II system clocks (CLKA, CLKB or CLKC), which enter the module through the DIME-II connectors. The pinouts for the various clock signals associated with the ZBTs are shown in Table 22:

| Signal Name | User FPGA Pin No. (2V3000 – 2V6000) |
|---|---|
| ZBTA_CLK | C18 |
| ZBTA_FB_OUT | C19 |
| ZBTA_FB_IN | E18 (GCLK4S) |
| | |
| ZBTB_CLK | D17 |
| ZBTB_FB_OUT | D16 |
| ZBTB_FB_IN | E16 (GCLK2S) |

**Table 22: Clock signals for ZBT Memory**

## 10.7.3 ZBT SRAM Clocking Example

An example of a typical clock arrangement for driving the ZBT SRAM with the input clock at the FPGA (e.g. CLKA, CLKB or CLKC) is illustrated in Figure 70:

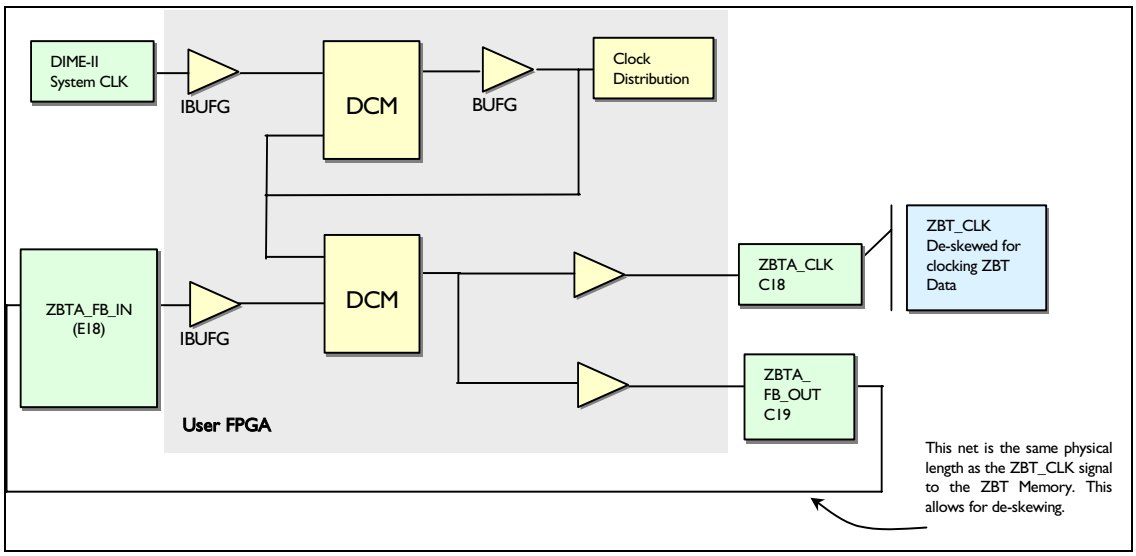



**Figure 70: ZBT SRAM Clocking Example: BANK A**

The above arrangement ensures that the device is triggered in phase. See Table 22 for pinout details for Bank A's Clocks.

# 10.8 User FPGA

## 10.8.1 Overview of User FPGA

The BenADDA module utilises the powerful processing capability of a Xilinx Virtex-II FPGA to support various FPGA sizes and speed grades. This provides the flexibility to choose the most appropriate FPGA to meet processing requirements. Figure 71 outlines all the possible DIME-II communications:

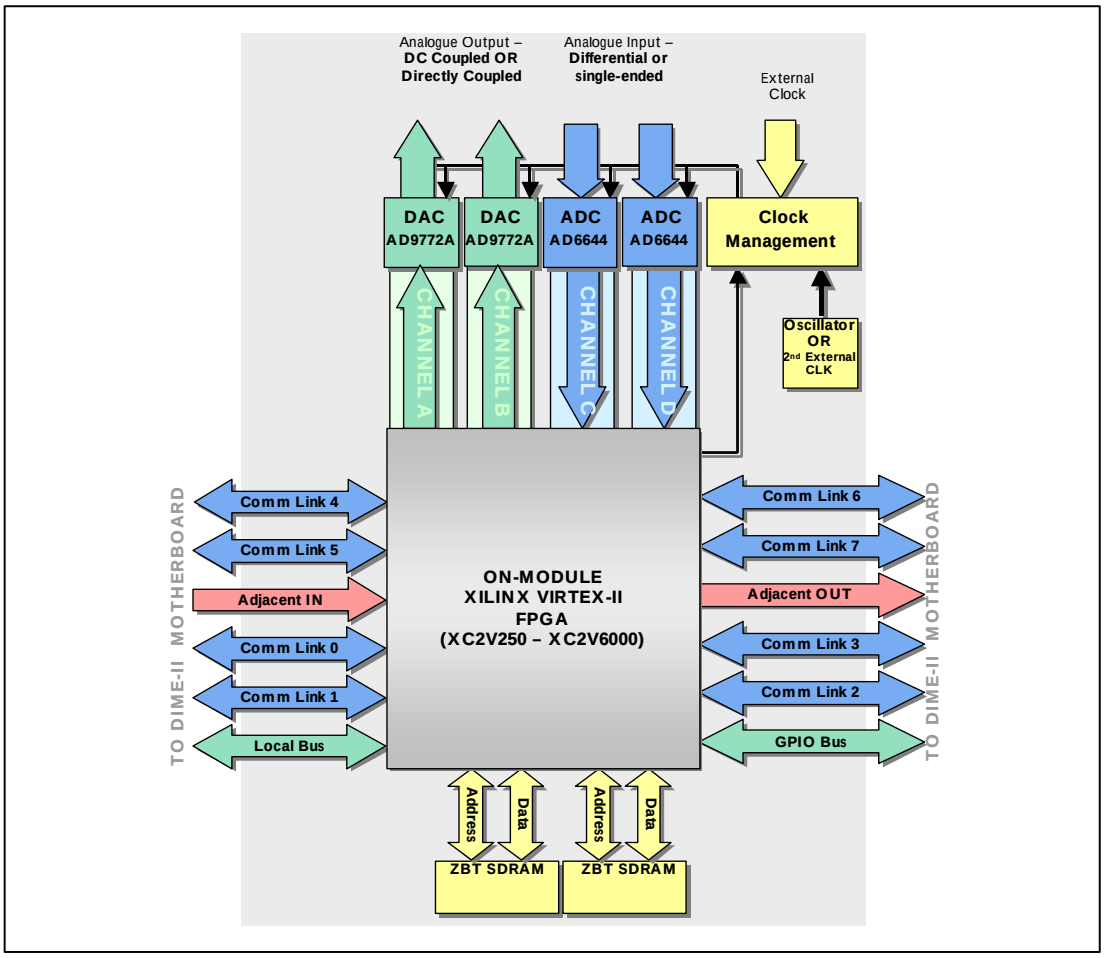



**Figure 71: User FPGA interfacing Overview**

Remember that not all DIME-II communication busses are available on all versions of the BenADDA. The following Section details the purpose of these communication busses.

## 10.8.2 DIME-II Communication busses

To support the large volume of communications between DIME-II modules and DIME-II motherboards, the following busses provide a robust infrastructure for these communications:

- Adjacent In Bus
- Adjacent Out Bus
- Comm Links 0 – 7 busses.
- Local Bus
- GP IO Bus

The infrastructure of these communication busses is described in the relevant Motherboard User Guide.

| Communication Bus | 2V250 (FG456) | 2V1000 (FG456) | 2V2000/ 2V3000 (FG676) | 2V3000 – 2V8000 (FF1152) |
|---|---|---|---|---|
| Adjacent IN | 16 bits | 26 bits | 40 bits | 64 bits |
| Adjacent OUT | 16 bits | 26 bits | 40 bits | 64 bits |
| Comm Link 0 | 12 bits | 12 bits | 12 bits | 12 bits |
| Comm Link 1 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 2 | 12 bits | 12 bits | 12 bits | 12 bits |
| Comm Link 3 | N/a | N/a | N/a bits | 12 bits |

| Communication Bus | 2V250 (FG456) | 2V1000 (FG456) | 2V2000/ 2V3000 (FG676) | 2V3000 – 2V8000 (FF1152) |
|---|---|---|---|---|
| Comm Link 4 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 5 | N/a | N/a | 12 bits | 12 bits |
| Comm Link 6 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 7 | N/a | N/a | 12 bits | 12 bits |
| Local Bus | 32 bits | 54 bits | 64 bits | 64 bits |
| GP IO Bus | N/a | N/a | N/a bits | 21 bits |

Table 23 outlines the size of each communication bus relative to FPGA size:

| Communication Bus | 2V250 (FG456) | 2V1000 (FG456) | 2V2000/ 2V3000 (FG676) | 2V3000 – 2V8000 (FF1152) |
|---|---|---|---|---|
| Adjacent IN | 16 bits | 26 bits | 40 bits | 64 bits |
| Adjacent OUT | 16 bits | 26 bits | 40 bits | 64 bits |
| Comm Link 0 | 12 bits | 12 bits | 12 bits | 12 bits |
| Comm Link 1 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 2 | 12 bits | 12 bits | 12 bits | 12 bits |
| Comm Link 3 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 4 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 5 | N/a | N/a | 12 bits | 12 bits |
| Comm Link 6 | N/a | N/a | N/a bits | 12 bits |
| Comm Link 7 | N/a | N/a | 12 bits | 12 bits |
| Local Bus | 32 bits | 54 bits | 64 bits | 64 bits |
| GP IO Bus | N/a | N/a | N/a bits | 21 bits |

**Table 23: Communication Bus Summary**

### 10.8.3 DIME-II System Clocks

The BenADDA has three system clocks available for use in the User FPGA - CLKA, CLKB and CLKC. All of these clock signals are generated on the DIME-II motherboard and are routed into the module site where the BenADDA is placed. Generally, these Clocks can be freely controlled by the user, are routed to Global Clock pins to provide maximum flexibility on the User FPGA. However, it should be noted that therefore that the available functionality of these DIME-II clocks is determined by carrier card. When the BenADDA is fitted to the BenONE carrier, as in the XtremeDSP kit, it should be noted that the available clocks are:

- CLKA - available programmable oscillator on the BenONE
- CLKB - available programmable oscillator on the BenONE
- CLKC - connected to a socket to support a crystal oscillator. Please note that no oscillator is supplied and this option on the BenONE is primarily intended to allow users fit a specific crystal if needed.

For full details of the generation of these clock signals prior to use on the module site, consult the appropriate motherboard User Guide.

## 10.9 Control and Monitoring Signals

The BenADDA produces a range of signals that allow the user to control and monitor the on-board module behaviour.

- 'Reset': The Reset signal is used to clear the memory of both on-board FPGAs
- 'JTAG': The JTAG chain is used for test and configuration purposes
- 'Config DONE': This signal is related to the configuration of the on-board FPGAs
- 'LEDs': There are two tri-colour LEDs on the BenADDA that are free to be used for the chosen application
- 'Temperature Sensor': Used to monitor the Temperature of the User FPGA.

### 10.9.1 BenADDA FPGA Reset

The BenADDA has a RESETl signal connected directly to both on-board FPGAs. RESETl is driven by the DIME-II motherboard and is available through the FUSE Software. This signal is active LOW and will reset the on-board FPGAs when a LOW is applied.

The pin locations for this RESET1 signal are listed for each FPGA option in the following places:

- For an XC2V250 FPGA, see Section 11.4.2 'DIME-II control and monitoring signals' on page 133
- For an XC2V250 FPGA, see Section 11.5.2 'DIME-II control and monitoring signals' on page 139
- For an XC2V3000 or XC2V6000 FPGA, see Section 11.8.2 'DIME-II control and monitoring signals' on page 165.

The pin locations are also listed in the UCF on the BenADDA installation CD.

### 10.9.2 JTAG Chain

DIME-II modules have a JTAG based Plug and Play, PnP, facility to enable auto-detection of the modules present in a system. Each DIME-II module has a unique ID number. The BenADDA IDs are listed in Table 24:

| ID Number (Hex) | Description |
|---|---|
| 30001033 | User FPGA: XC2V250 (FG456), CLK FPGA: XC2V80 |
| 30088033 | User FPGA: XC2V1000 (FG456), CLK FPGA: XC2V80 |
| 30044033 | User FPGA: XC2V3000 (FF1152), CLK FPGA: XC2V80 |
| 30555033 | User FPGA: XC2V6000 (FF1152), CLK FPGA: XC2V80 |

**Table 24: BenADDA Assigned MDF Code Listing**

The physical order of the devices in the JTAG chain, illustrated in Figure 72, is:

1. User FPGA
2. CLK FPGA

To establish the module order in the JTAG chain, Nallatech's FUSE System Software is deployed. The software initially scans the chain to identify and index the devices on the BenADDA module. The device nearest the TDO output of the module is identified as device 0 on that module. For each device upstream on the JTAG chain, the index is incremented. Figure 72 shows the device numbers that are assigned for this BenADDA module:

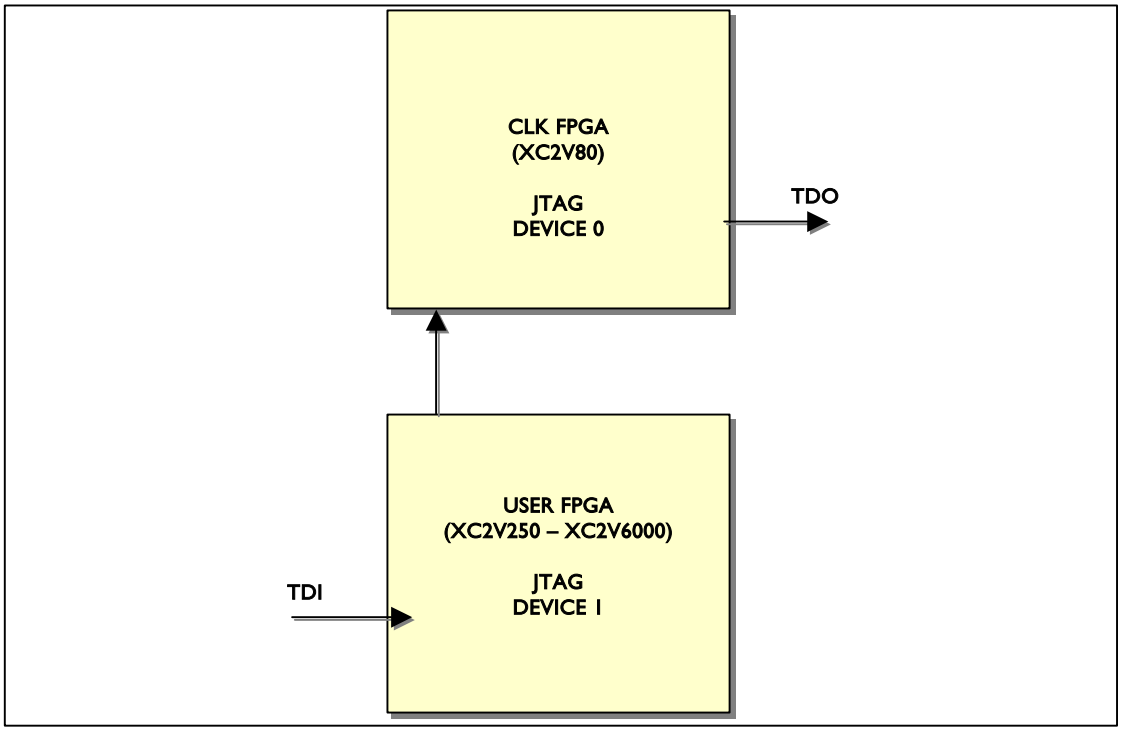



**Figure 72: JTAG Device Indexing**

## 10.9.3 Config DONE

The DIME-II standard allows a design to be implemented based on the status of the entire system. CONFIG DONE signal provides built-in control that can be used if the System Designer wishes. This feature allows a designer to synchronise all aspects of the complete system.

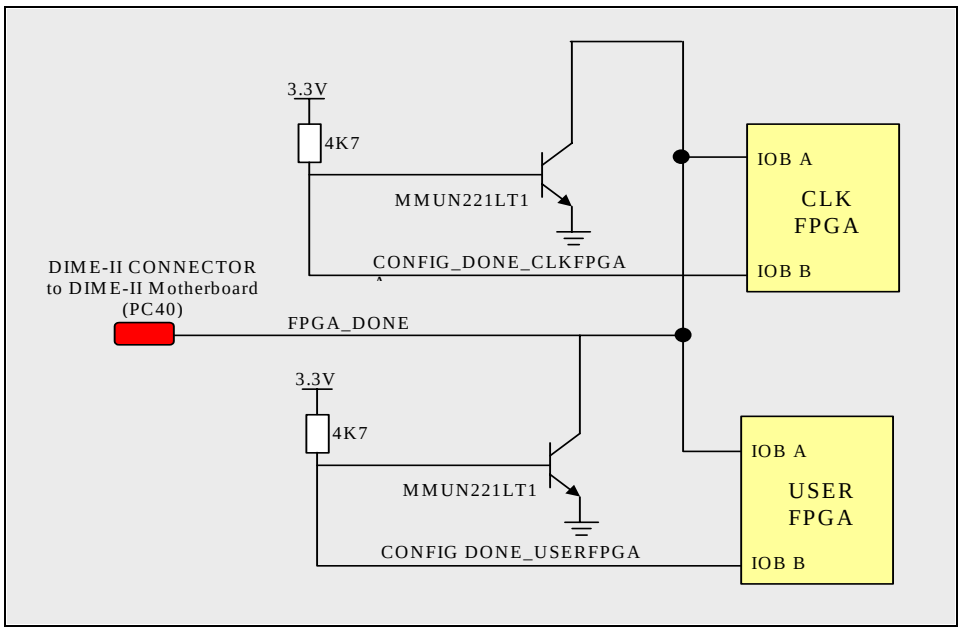



**Figure 73: CONFIG DONE Circuit**

Figure 73 shows the set-up of the 'CONFIGURATION DONE' operation on the BenADDA. The FPGA_DONE signal to the two on-board FPGAs determines the status of the FPGAs on the module. The FPGA_DONE signal is connected directly to the carrier motherboard where it is then connected to all other module sites and the PCI FPGA. This signal has a weak pull-up applied to it on the motherboard.

When the FPGAs on the BenADDA are NOT configured, the base of the two transistors (MMUN221LT1) will be switched on by the 3.3V pull-up. With the base of the transistors switched on, the FPGA_DONE signal (connected to the motherboard via the DIME-II connector) will be pulled LOW through the transistor.

Once the FPGA has been configured, the user should send out a LOW signal on the appropriate CONFIG_DONE pin. (i.e. a LOW would be driven out of 'IOB B' on the FPGA) This turns the base of the transistor off; and the FPGA_DONE value is now subject to the status of the complete system. Once all other FPGAs in the system are configured, the FPGA_DONE signal will be HIGH via the pull-up on the motherboard. However, if one FPGA is not configured, the FPGA_DONE signal from that device will still be pulled LOW, meaning that FPGA_DONE for the entire system would be LOW. The System designer will therefore be able to read the value of FPGA_DONE, via FUSE software, at the 'IOB A' input on the various FPGAs to determine the overall state of the system.

If the 'CONFIG DONE' signal is to be utilised in a system, the user should ensure that IOB B is driven low once the FPGA has been successfully configured. Alternatively, if a system initialisation sequence is required then IOB B can be driven low after this. The FPGA then polls IOB A to see that all other FPGAs in the system have been configured.

The pin locations for this Config DONE signal are listed for each FPGA option in the following places:

- For an XC2V250 FPGA, see Section 11.4.2 'DIME-II control and monitoring signals' on page 133
- For an XC2V1000 FPGA, see Section 11.5.2 'DIME-II control and monitoring signals' on page 139
- For an XC2V3000 or XC2V6000 FPGA, see Section 11.8.2 'DIME-II control and monitoring signals' on page 165.

## 10.9.4 User LEDs

The BenADDA has two LEDs, which can be used for specific design purposes, such as displaying status.

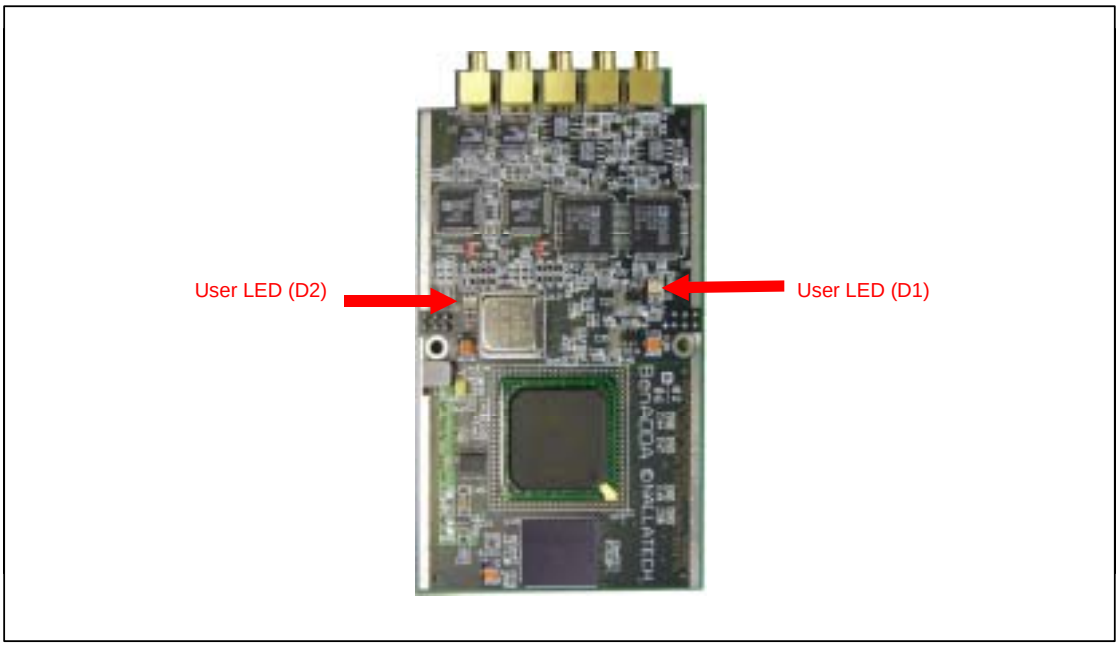



**Figure 74: User LEDs location (XC2V250 -2V1000 version shown)**

The LEDs are illuminated when the corresponding pin is logic 0, otherwise the LED is not illuminated (Active LOW). The LEDs used on the BenADDA are Tri-Colour (each LED displays a total of three different colours), meaning that each LED can act as three individual LEDs. Each LED has a RED and GREEN diode inside their chip. Applying logic 0 to either Cathode will cause that colour to illuminate and applying logic 0 to both Cathodes (green and red) will cause a third colour, yellow, to illuminate.

| Signal Description | User LED | Signal Name | User FPGA Pin |
|---|---|---|---|
| Green Diode for LED2 | D2* | LED2_Green | Consult 'Pinout Information' Section** |
| Red Diode for LED2 | D2* | LED2_Red | Consult 'Pinout Information' Section ** |
| Green Diode for LED1 | D1* | LED1_Green | Consult 'Pinout Information' Section ** |
| Red Diode for LED1 | D1* | LED1_Red | Consult 'Pinout Information' Section ** |

**Table 25: LED signals**

* This refers to the silkscreen marking on the PCB (see Figure 74).

**The pin locations for the User LED signals are listed for each FPGA option in the following places:

- For an XC2V250 FPGA, see 'User LEDs' on page 133
- For an XC2V1000 FPGA, see 'User LEDs' on page 139
- For an XC2V3000 or XC2V6000 FPGA, see 'User LEDs' on page 165.

## 10.10 Temperature Sensor

The BenADDA can be supplied with various sizes of User FPGAs which can become very hot when running at full potential. As a result, the BenADDA is fitted with a temperature device that monitors the heat levels within the User FPGA. Figure 75 shows the BenADDA temperature sensor interface:

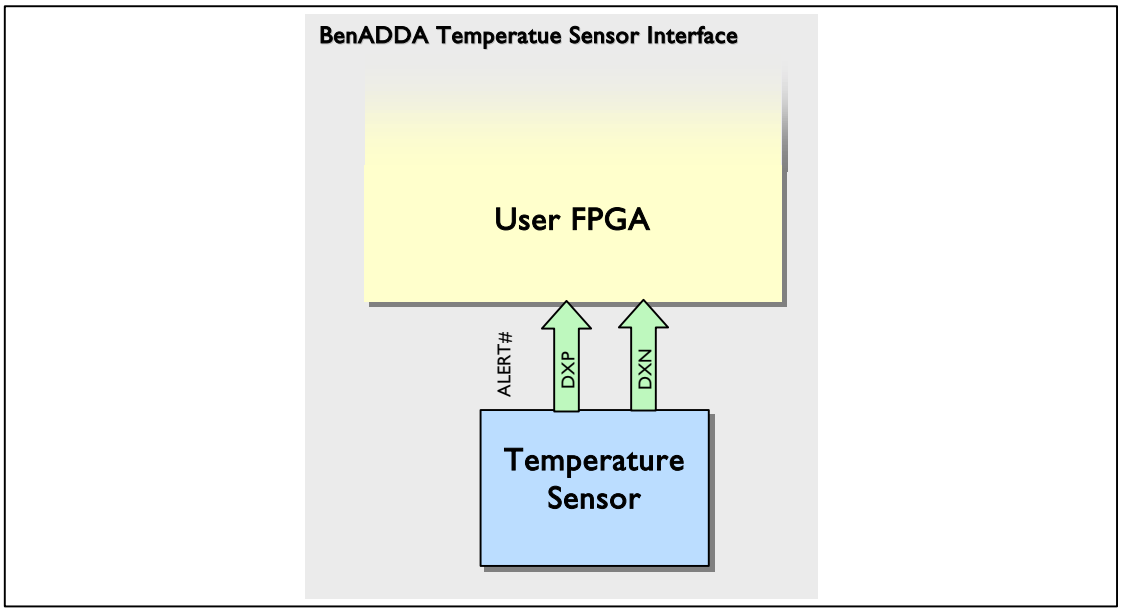



**Figure 75: Temperature Sensor Interface**

The Temperature Sensor used on the BenADDA is supplied from MAXIM (Part Number: MAX1617MEE).

For full details on the specification of this device please refer to its datasheet MAX1617, which is supplied on the XtremeDSP Development Kit CD.

Please note that this feature is not supplied as standard with the XtremeDSP Development Kit. For more information on the temperature sensor please contact Nallatech.

# Section 11

# Interfacing

In this Section:

- Interfacing via MCX connectors
- General Purpose I/O Bus J4
- General Purpose I/O Bus J5

## 11.1 Interfacing via MCX connectors

The BenADDA has five through-hole MCX connectors that allow interfacing to and from the module. All Analogue Input and Output signals to the BenADDA are conducted via four MCX connectors on the top of the module. The fifth MCX connector is available to provide an input source for an external clock. Figure 76 outlines the positioning of these connectors:

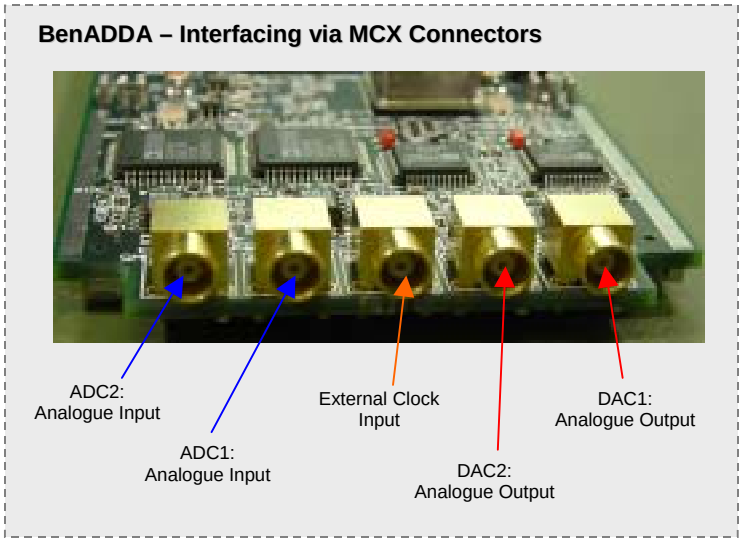



**Figure 76: MCX connectors**

Figure 76 show there are two Analogue Input channels, two Analogue Output channels and one external clock source input.

### 11.1.1 Interfacing to MCX connectors via supplied cable

The BenADDA is supplied with cables that are suitable for connecting between the on-board MCX connectors and a user input/output BNC connection.

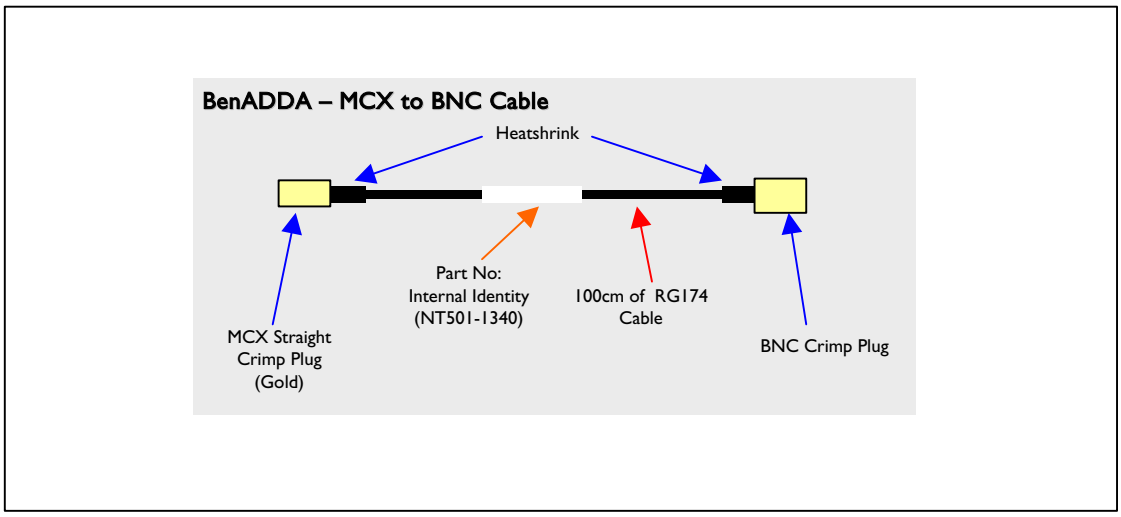



**Figure 77: Diagram of supplied Cable Assembly**

The MCX connectors are a "push" fit design; therefore the MCX crimp plug on the supplied cable pushes into the MCX connector.

## 11.2 User IO header - Interfacing

The BenADDA has a two-pin 0.1th pitch header that connects directly to the User FPGA. There is no assigned function for this two-pin header, which is therefore free to be used for your desired application.

The two pin header is connected directly to the Virtex-II User FPGA and therefore signals applied to this MUST be within the range of 0V – 3.3V. Virtex-II devices are NOT 5V tolerant.

Table 26 contains the pinout information for the User IO header:

| Signal Description | Signal Name | User FPGA Pin |
|---|---|---|
| Unassigned User IO connection to USER FPGA | User_IO_1 | Consult 'Pinout Information' Section * |
| Unassigned User IO connection to USER FPGA | User_IO_2 | Consult 'Pinout Information' Section * |

**Table 26: Pinouts of User IO header**

*The pin locations for the User IO signals are listed for each FPGA option in the following places:

- The XC2V250 FPGA has no available pin locations for the User IO Header
- For an XC2V1000 FPGA, see 'User IO header' on page 142
- For an XC2V3000 or XC2V6000 FPGA, see '

User IO Header' on page 171.

## 11.3 Design Partitioning

The XtremeDSP Development Kit allows the user to partition the functionality of their application between software and hardware easily and effectively. Below, the design partitioning of the XtremeDSP Development Kit is shown in Figure 21:

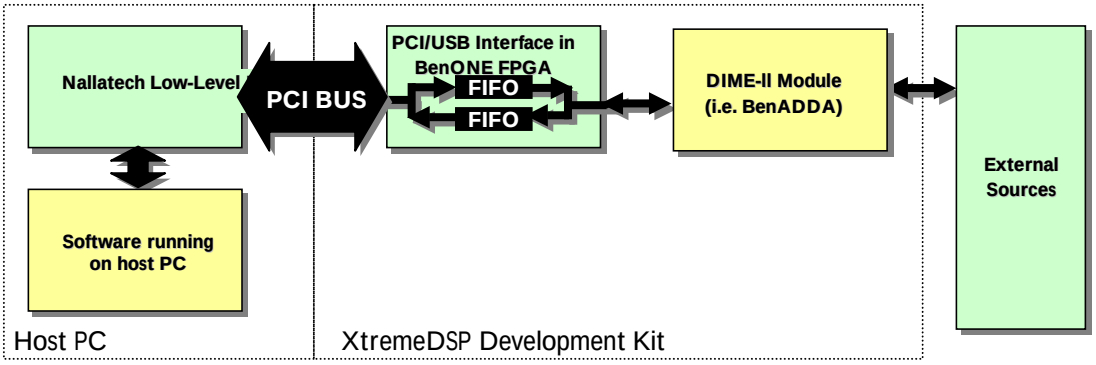



**Figure 78: XtremeDSP Development Kit Design Partitioning**

The Green blocks do not require any further design from a user perspective. The interface is pre-configured with either PCI or USB and the external sources are assumed to be in place. The user needs to design application designs for any FPGAs on the hosted DIME-II module, the BenADDA in this case. The Software running on the host PC is available as a pre-designed GUI. For users who require additional functionality and wish to have their own software front-end, the FUSE Software Library provides functions for use in application programs. These facilitate functionality such as FPGA configuration/reset, clock speed setting, data transfer.

In order to make use of these software-interfacing functions for specific data transfer to their designs, certain signals must be connected into the design. These signal are listed in the Interface COMM Signal column in Table 27 and Table 28. A core for use in the user FPGA is provided to aid the process of integration and communication with these signals. Please see 'PCI to User FPGA Interface' Application note that is provided on the FUSE Software CD for further details.

| Interface COMM Signal | General Bus Signal Name | Dime-II Connector PIN No | User FPGA (2V250 FG456) Pin No | User FPGA (2V1000 FG456) PIN No | User FPGA (2V2000 FG676) Pin No | User FPGA (2V3000 FG676) Pin No | User FPGA (2V3000/ 6000 FF1152) Pin No |
|---|---|---|---|---|---|---|---|
| ADIO<0> | LBUS<0> | PB1 | D6 | B8 | D10 | F8 | V4 |
| ADIO<1> | LBUS<1> | PB2 | E8 | E8 | G10 | G10 | U3 |
| ADIO<2> | LBUS<2> | PB3 | E7 | E7 | G9 | G9 | T3 |
| ADIO<3> | LBUS<3> | PB4 | A6 | A6 | C8 | C8 | P4 |
| ADIO<4> | LBUS<4> | PB6 | A4 | A4 | C6 | C6 | R4 |
| ADIO<5> | LBUS<5> | PB7 | B4 | B4 | D6 | D6 | M4 |
| ADIO<6> | LBUS<6> | PB8 | D2 | D2 | F4 | F4 | N4 |
| ADIO<7> | LBUS<7> | PB9 | E3 | E3 | G5 | G5 | K4 |
| ADIO<8> | LBUS<8> | PB10 | D1 | D1 | F3 | F3 | L4 |
| ADIO<9> | LBUS<9> | PB11 | E2 | E2 | G4 | G4 | L3 |
| ADIO<10> | LBUS<10> | PB12 | E1 | E1 | G3 | G3 | M3 |
| ADIO<11> | LBUS<11> | PB13 | H5 | H5 | K7 | K7 | H3 |
| ADIO<12> | LBUS<12> | PB15 | G2 | G2 | J4 | J4 | J3 |
| ADIO<13> | LBUS<13> | PB16 | G1 | G1 | J3 | J3 | H4 |
| ADIO<14> | LBUS<14> | PB17 | J6 | J6 | L8 | L8 | J4 |

| Interface COMM Signal | General Bus Signal Name | Dime-II Connector PIN No | User FPGA (2V250 FG456) Pin No | User FPGA (2V1000 FG456) PIN No | User FPGA (2V2000 FG676) Pin No | User FPGA (2V3000 FG676) Pin No | User FPGA (2V3000/ 6000 FF1152) Pin No |
|---|---|---|---|---|---|---|---|
| ADIO<15> | LBUS<15> | PB18 | H4 | H4 | K6 | K6 | F3 |
| ADIO<16> | LBUS<16> | PB19 | H3 | H3 | K5 | K5 | G3 |
| ADIO<17> | LBUS<17> | PB20 | H2 | H2 | K4 | K4 | E4 |
| ADIO<18> | LBUS<18> | PB21 | H1 | H1 | K3 | K3 | F4 |
| ADIO<19> | LBUS<19> | PB22 | K2 | K2 | M4 | M4 | D3 |
| ADIO<20> | LBUS<20> | PB24 | K1 | K1 | M3 | M3 | E3 |
| ADIO<21> | LBUS<21> | PB25 | K4 | K4 | M6 | M6 | AA4 |
| ADIO<22> | LBUS<22> | PB26 | K3 | K3 | M5 | M5 | AB4 |
| ADIO<23> | LBUS<23> | PB27 | L5 | L5 | N7 | N7 | AC3 |
| ADIO<24> | LBUS<24> | PB28 | L4 | L4 | N6 | N6 | AC4 |
| ADIO<25> | LBUS<25> | PB29 | L3 | L3 | N5 | N5 | D2 |
| ADIO<26> | LBUS<26> | PB30 | L2 | L2 | N4 | N4 | D1 |
| ADIO<27> | LBUS<27> | PB31 | M5 | M5 | P7 | P7 | E2 |
| ADIO<28> | LBUS<28> | PB33 | N1 | N1 | R3 | R3 | E1 |
| ADIO<29> | LBUS<29> | PB34 | N2 | N2 | R4 | R4 | F2 |
| ADIO<30> | LBUS<30> | PB35 | P6 | P6 | T8 | T8 | F1 |
| ADIO<31> | LBUS<31> | PB36 | P5 | P5 | T7 | T7 | G2 |
| BUSY | ADJOUT<0> | PD29 | B19 | B19 | D21 | D21 | N32 |
| EMPTY | ADJOUT<1> | PD30 | A19 | A19 | C21 | C21 | P32 |
| RDI_WR | ADJOUT<2> | PD31 | C22 | C22 | E24 | E24 | T30 |
| AS_DSI | ADJOUT<3> | PD32 | C21 | C21 | E23 | E23 | U30 |
| RENI_WENI | ADJOUT<4> | PD33 | E20 | E20 | G22 | G22 | U27 |
| INTI | ADJOUT<5> | PD34 | E19 | E19 | G21 | G21 | U26 |
| RSTI | ADJOUT<6> | PD35 | D22 | D22 | F24 | F24 | U29 |

**Table 27 - Interface to User FPGA Comms Signals**

Also DSP_CLK should be connected to CLK1(sometimes referred to as CLKB)

| Interface COMM Signal | General Bus Signal Name | Dime-II Connector PIN No | User FPGA (2V250 FG456) Pin No | User FPGA (2V1000 FG456) PIN No | User FPGA (2V2000 FG676) Pin No | User FPGA (2V3000 FG676) Pin No | User FPGA (2V3000/ 6000 FF1152) Pin No |
|---|---|---|---|---|---|---|---|
| DSP_CLK | CLK1 | PC31 | V11 | V11 | Y13 | Y13 | AG17 |

**Table 28 - Interface to User FPGA Clock Requirements**

# Pinout Information

This Section provides you with the following pinout information for each User FPGA option (XC2V250FG456, XC2V1000FG456, XC2V2000FG676, XC2V3000FG676, XC2V3000FF1152 XC2V6000FF1152):

- User FPGA to DIME-II motherboard communication
- DIME-II control and monitoring signals
- ZBT SRAM interface to user FPGA
- Clock signals relating to DACs and ADCs
- DAC Signal Pinouts
- ADC Signal Pinouts

## 11.4 XC2V250 FG456

### 11.4.1 User FPGA to DIME-II motherboard communication

#### Local Bus Pinouts

| Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No |
|---|---|---|---|---|---|
| LBUS<0> | PB1 | D6 | LBUS<16> | PB19 | H3 |
| LBUS<1> | PB2 | E8 | LBUS<17> | PB20 | H2 |
| LBUS<2> | PB3 | E7 | LBUS<18> | PB21 | H1 |
| LBUS<3> | PB4 | A6 | LBUS<19> | PB22 | K2 |
| LBUS<4> | PB6 | A4 | LBUS<20> | PB24 | K1 |
| LBUS<5> | PB7 | B4 | LBUS<21> | PB25 | K4 |
| LBUS<6> | PB8 | D2 | LBUS<22> | PB26 | K3 |
| LBUS<7> | PB9 | E3 | LBUS<23> | PB27 | L5 |
| LBUS<8> | PB10 | D1 | LBUS<24> | PB28 | L4 |
| LBUS<9> | PB11 | E2 | LBUS<25> | PB29 | L3 |
| LBUS<10> | PB12 | E1 | LBUS<26> | PB30 | L2 |
| LBUS<11> | PB13 | H5 | LBUS<27> | PB31 | M5 |
| LBUS<12> | PB15 | G2 | LBUS<28> | PB33 | N1 |
| LBUS<13> | PB16 | G1 | LBUS<29> | PB34 | N2 |
| LBUS<14> | PB17 | J6 | LBUS<30> | PB35 | P6 |
| LBUS<15> | PB18 | H4 | LBUS<31> | PB36 | P5 |

**Table 29: Local Bus Pinouts (2V250)**

## Adjacent IN Bus – User FPGA communications

| Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No |
|---|---|---|---|---|---|
| ADJIN<0>* | PA29 | D12 | ADJIN<8> | PA38 | R2 |
| ADJIN<1>* | PA30 | E12 | ADJIN<9> | PA39 | R1 |
| ADJIN<2> | PA31 | M2 | ADJIN<10> | PA40 | U4 |
| ADJIN<3> | PA32 | M1 | ADJIN<11> | PA41 | U5 |
| ADJIN<4> | PA33 | M4 | ADJIN<12> | PA42 | V4 |
| ADJIN<5> | PA34 | M3 | ADJIN<13> | PA43 | V3 |
| ADJIN<6> | PA35 | R4 | ADJIN<14> | PA44 | W2 |
| ADJIN<7> | PA36 | R3 | ADJIN<15> | PA45 | W1 |

**Table 30: Adjacent IN BUS Pinouts (2V250)**

* ADJIN<0> is connected to a clock pin (GCLK0S) on the Virtex-II.

* ADJIN<1> is connected to a clock pin (GCLK1P) on the Virtex-II.

## Adjacent OUT Bus - User FPGA connections

| Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No |
|---|---|---|---|---|---|
| ADJOUT<0> | PD29 | B19 | ADJOUT<8> | PD38 | H20 |
| ADJOUT<1> | PD30 | A19 | ADJOUT<9> | PD39 | H19 |
| ADJOUT<2> | PD31 | C22 | ADJOUT<10> | PD40 | H22 |
| ADJOUT<3> | PD32 | C21 | ADJOUT<11> | PD41 | H21 |
| ADJOUT<4> | PD33 | E20 | ADJOUT<12> | PD42 | K22 |
| ADJOUT<5> | PD34 | E19 | ADJOUT<13> | PD43 | K21 |
| ADJOUT<6> | PD35 | D22 | ADJOUT<14> | PD44 | L19 |
| ADJOUT<7> | PD36 | D21 | ADJOUT<15> | PD45 | L20 |

**Table 31: Adjacent OUT BUS Pinouts (2V250)**

## PLINKS connected to the User FPGA: PLINKS 0, 2, 5 and 7.

| Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No |
|---|---|---|---|---|---|
| PP0LK<0>* | PA2 | C11 | PP2LK<0>* | PD2 | F13 |
| PP0LK<1>* | PA3 | D11 | PP2LK<1>* | PD3 | F12 |
| PP0LK<2> | PA4 | F11 | PP2LK<2> | PD4 | B12 |
| PP0LK<3> | PA5 | E11 | PP2LK<3> | PD5 | C12 |
| PP0LK<4> | PA6 | A10 | PP2LK<4> | PD6 | C13 |
| PP0LK<5> | PA7 | B10 | PP2LK<5> | PD7 | B13 |
| PP0LK<6> | PA8 | C10 | PP2LK<6> | PD8 | E13 |
| PP0LK<7> | PA9 | D10 | PP2LK<7> | PD9 | D13 |
| PP0LK<8> | PA11 | F10 | PP2LK<8> | PD11 | C17 |
| PP0LK<9> | PA12 | E10 | PP2LK<9> | PD12 | B17 |
| PP0LK<10> | PA13 | C5 | PP2LK<10> | PD13 | D18 |
| PP0LK<11> | PA14 | C4 | PP2LK<11> | PD14 | C18 |

**Table 32: PLINK Pinouts (2V250)**

* PP0LK<0> is connected to a clock pin (GCLK6S) on the Virtex-II.

* PP0LK<1> is connected to a clock pin (GCLK7P) on the Virtex-II.

* PP2LK<0> is connected to a clock pin (GCLK2S) on the Virtex-II.

* PP2LK<1> is connected to a clock pin (GCLK3P) on the Virtex-II.

## 11.4.2 DIME-II control and monitoring signals

### DIME-II Specific Pins

| Dime-II Connector PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V250) PIN No |
|---|---|---|---|
| CONFIG_DONE | FPGA_DONE | PC40 | L22 |
| N/c | CONFIG_DONE* | N/a | L21 |
| CLK0 | CLKA | PC24 | B11 |
| CLK1 | CLKB | PC31 | V11 |
| CLK2 | CLKC | PC42 | Y11 |
| RESETI | RESETI | PC15 | E14 |
| SLOT_ID0 | SLOT_ID0 | PC51 | J17 |
| SLOT_ID1 | SLOT_ID1 | PC52 | G22 |

**Table 33: User FPGA Specific Pinouts (2V250)**

*CONFIG_DONE is driven from the IO of the Virtex-II into the base of the transistor to signal that the on-board User FPGA has been configured successfully. User to drive this pin LOW once the FPGA is configured.

### User LEDs

| Signal Name | User FPGA (2V250) PIN No |
|---|---|
| LED_Green1 | V9 |
| LED_Red1 | N/A |
| | |
| LED_Green2 | V21 |
| LED_Red2 | V22 |

**Table 34: User LED Pinouts (2V250)**

### On-board Temperature Sensor

| Signal Name | User FPGA (2V250) PIN No |
|---|---|
| DXN | D5 |
| DXP | A3 |
| ALERTI | T1 |

**Table 35: Temperature Sensor Pinouts (2V250)**

## 11.4.3 Clock signals relating to DACs and ADCs

### Clock sources available at CLK FPGA

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V250) PIN No | Signal Description |
|---|---|---|---|
| CLK_Op_Amp | B6 (GCLK6S) | N/a | External CLK source via Op_Amp |
| CLK_Op_Ampl | C6 (GCLK7P) | N/a | Complement of External CLK source via Op_Amp |
| EXT_CLK | B8 (GCLK0S) | N/a | External CLK source straight to CLK FPGA |
| EXT_CLKl | A8 (GCLK1P) | N/a | Complement of External CLK source straight to CLK FPGA |
| Osc_CLK | M6 (GCLK4P) | N/a | LVTTL Clock Oscillator |
| GEN_CLKA | K7 (GCLK0P) | Y10 | Generated Clock A |
| GEN_CLKC | N8 (GCLK1S) | AA10 | Generated Clock C |
| GEN_CLKB | M7 (GCLK6P) | V10 | Generated Clock B |
| GEN_CLKD | N7 (GCLK7S) | W10 | Generated Clock D |
| EXT2_CLK | D7 (GCLK5P) | N/a | 2nd External Clock |
| EXT2_CLKl | A6 (GCLK4S) | N/a | Complement of 2nd External Clock |

**Table 36: Clock Signals at CLK FPGA (2V250)**

### Clock Feedback signals

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V250) PIN No | Signal Description |
|---|---|---|---|
| CLK1_FB | J2 | Y12 | Feedback to User FPGA |
| CLK3_FB | H4 | W12 | Feedback to User FPGA |
| CLK2_FB | H12 | AA11 | Feedback to User FPGA |

**Table 37: Clock Feedback Signals (2V250)**

### Clocking Pinouts for DACs and ADCs

| Signal Name | CLK FPGA (2V80) Pin No |
|---|---|
| ADC_CLKA | E4 |
| ADC_CLKAl | D1 |
| ADC_CLKB | G1 |
| ADC_CLKBl | F1 |
| DAC_CLKA | D13 |
| DAC_CLKAl | D12 |
| DAC_CLKB | G10 |
| DAC_CLKBl | F12 |

**Table 38: Clocking Pinouts for DACs and ADCs**

### 11.4.4 DAC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V250) PIN No | Signal Name (DAC 2) | User FPGA (2V250) PIN No |
|---|---|---|---|
| DAC1_D<0> | T22 | DAC2_D<0> | AB19 |
| DAC1_D<1> | R22 | DAC2_D<1> | AA19 |
| DAC1_D<2> | R21 | DAC2_D<2> | Y22 |
| DAC1_D<3> | R20 | DAC2_D<3> | Y21 |
| DAC1_D<4> | R19 | DAC2_D<4> | Y18 |
| DAC1_D<5> | P17 | DAC2_D<5> | Y17 |
| DAC1_D<6> | N22 | DAC2_D<6> | Y16 |
| DAC1_D<7> | N21 | DAC2_D<7> | W22 |
| DAC1_D<8> | N17 | DAC2_D<8> | W21 |
| DAC1_D<9> | M21 | DAC2_D<9> | W20 |
| DAC1_D<10> | M20 | DAC2_D<10> | W18 |
| DAC1_D<11> | M19 | DAC2_D<11> | W16 |
| DAC1_D<12> | M18 | DAC2_D<12> | W17 |
| DAC1_D<13> | M17 | DAC2_D<13> | V20 |
| DAC1_RESET | T21 | DAC2_RESET | R18 |

**Table 39: DACs Signal Pinouts (2V250)**

### 11.4.5 ADC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V250) PIN No | Signal Name (DAC 2) | User FPGA (2V250) PIN No |
|---|---|---|---|
| ADC1_D<0> | AB18 | ADC2_D<0> | AB10 |
| ADC1_D<1> | AB16 | ADC2_D<1> | AB4 |
| ADC1_D<2> | AB17 | ADC2_D<2> | AB5 |
| ADC1_D<3> | AA18 | ADC2_D<3> | AA3 |
| ADC1_D<4> | AA16 | ADC2_D<4> | AA4 |
| ADC1_D<5> | AA17 | ADC2_D<5> | AA5 |
| ADC1_D<6> | AB13 | ADC2_D<6> | Y4 |
| ADC1_D<7> | AA13 | ADC2_D<7> | Y5 |
| ADC1_D<8> | Y13 | ADC2_D<8> | Y6 |
| ADC1_D<9> | W13 | ADC2_D<9> | W5 |
| ADC1_D<10> | W11 | ADC2_D<10> | W6 |
| ADC1_D<11> | V12 | ADC2_D<11> | V6 |
| ADC1_D<12> | V13 | ADC2_D<12> | V7 |
| ADC1_D<13> | U11 | ADC2_D<13> | V8 |

**Table 40: ADC Signal Pinouts (2V250)**

### 11.4.6 User IO header

| Signal Name | User FPGA (2V250) PIN No |
|---|---|
| User_IO_1 | N/a |
| User_IO_2 | N/a |

**Table 41: User IO Header Pinouts (2V250)**

# 11.5 XC2V1000 FG456

## 11.5.1 User FPGA to DIME-II motherboard communication

### Local Bus Pinouts

| Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|---|---|
| LBUS<0> | PB1 | B8 | LBUS<27> | PB31 | M5 |
| LBUS<1> | PB2 | E8 | LBUS<28> | PB33 | N1 |
| LBUS<2> | PB3 | E7 | LBUS<29> | PB34 | N2 |
| LBUS<3> | PB4 | A6 | LBUS<30> | PB35 | P6 |
| LBUS<4> | PB6 | A4 | LBUS<31> | PB36 | P5 |
| LBUS<5> | PB7 | B4 | LBUS<32> | PB37 | C8 |
| LBUS<6> | PB8 | D2 | LBUS<33> | PB38 | D8 |
| LBUS<7> | PB9 | E3 | LBUS<34> | PB39 | D7 |
| LBUS<8> | PB10 | D1 | LBUS<35> | PB40 | F5 |
| LBUS<9> | PB11 | E2 | LBUS<36> | PB42 | F4 |
| LBUS<10> | PB12 | E1 | LBUS<37> | PB43 | F2 |
| LBUS<11> | PB13 | H5 | LBUS<38> | PB44 | F3 |
| LBUS<12> | PB15 | G2 | LBUS<39> | PB45 | G4 |
| LBUS<13> | PB16 | G1 | LBUS<40> | PB46 | G3 |
| LBUS<14> | PB17 | J6 | LBUS<41> | PB47 | J5 |
| LBUS<15> | PB18 | H4 | LBUS<42> | PB48 | J2 |
| LBUS<16> | PB19 | H3 | LBUS<43> | PB49 | J1 |
| LBUS<17> | PB20 | H2 | LBUS<44> | PB51 | J4 |
| LBUS<18> | PB21 | H1 | LBUS<45> | PB52 | J3 |
| LBUS<19> | PB22 | K2 | LBUS<46> | PB53 | K6 |
| LBUS<20> | PB24 | K1 | LBUS<47> | PB54 | K5 |
| LBUS<21> | PB25 | K4 | LBUS<48> | PB55 | L6 |
| LBUS<22> | PB26 | K3 | LBUS<49> | PB56 | N5 |
| LBUS<23> | PB27 | L5 | LBUS<50> | PB57 | N3 |
| LBUS<24> | PB28 | L4 | LBUS<51> | PB58 | N4 |
| LBUS<25> | PB29 | L3 | LBUS<52> | PB60 | N6 |
| LBUS<26> | PB30 | L2 | LBUS<53> | PB61 | R5 |

**Table 42: Local Bus Pinouts (2V1000)**

### Adjacent IN Bus – User FPGA communications

| Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|---|---|
| ADJIN<0>* | PA29 | D12 | ADJIN<13> | PA43 | V3 |
| ADJIN<1>* | PA30 | E12 | ADJIN<14> | PA44 | W2 |
| ADJIN<2> | PA31 | M2 | ADJIN<15> | PA45 | W1 |
| ADJIN<3> | PA32 | M1 | ADJIN<16> | PA47 | P2 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|---|---|
| ADJIN<4> | PA33 | M4 | ADJIN<17> | PA48 | P1 |
| ADJIN<5> | PA34 | M3 | ADJIN<18> | PA49 | P3 |
| ADJIN<6> | PA35 | R4 | ADJIN<19> | PA50 | P4 |
| ADJIN<7> | PA36 | R3 | ADJIN<20> | PA51 | T3 |
| ADJIN<8> | PA38 | R2 | ADJIN<21> | PA52 | T2 |
| ADJIN<9> | PA39 | R1 | ADJIN<22> | PA53 | T4 |
| ADJIN<10> | PA40 | U4 | ADJIN<23> | PA54 | T5 |
| ADJIN<11> | PA41 | U5 | ADJIN<24> | PA56 | U3 |
| ADJIN<12> | PA42 | V4 | ADJIN<25> | PA57 | U2 |

**Table 43: Adjacent IN BUS Pinouts – (2V1000)**

* ADJIN<0> is connected to a clock pin (GCLK0S) on the Virtex-II.

* ADJIN<1> is connected to a clock pin (GCLK1P) on the Virtex-II.

## Adjacent OUT Bus - User FPGA connections

| Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|---|---|
| ADJOUT<0> | PD29 | B19 | ADJOUT<13> | PD43 | K21 |
| ADJOUT<1> | PD30 | A19 | ADJOUT<14> | PD44 | L19 |
| ADJOUT<2> | PD31 | C22 | ADJOUT<15> | PD45 | L20 |
| ADJOUT<3> | PD32 | C21 | ADJOUT<16> | PD47 | F19 |
| ADJOUT<4> | PD33 | E20 | ADJOUT<17> | PD48 | F20 |
| ADJOUT<5> | PD34 | E19 | ADJOUT<18> | PD49 | F22 |
| ADJOUT<6> | PD35 | D22 | ADJOUT<19> | PD50 | F21 |
| ADJOUT<7> | PD36 | D21 | ADJOUT<20> | PD51 | G18 |
| ADJOUT<8> | PD38 | H20 | ADJOUT<21> | PD52 | G19 |
| ADJOUT<9> | PD39 | H19 | ADJOUT<22> | PD53 | G21 |
| ADJOUT<10> | PD40 | H22 | ADJOUT<23> | PD54 | G20 |
| ADJOUT<11> | PD41 | H21 | ADJOUT<24> | PD56 | J18 |
| ADJOUT<12> | PD42 | K22 | ADJOUT<25> | PD57 | H18 |

**Table 44: Adjacent OUT BUS Pinouts – (2V1000)**

## PLINKS connected to the User FPGA: PLINKS 0, 2, 5 and 7.

| Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|---|---|
| PP0LK<0>* | PA2 | C11 | PP2LK<0>* | PD2 | F13 |
| PP0LK<1>* | PA3 | D11 | PP2LK<1>* | PD3 | F12 |
| PP0LK<2> | PA4 | F11 | PP2LK<2> | PD4 | B12 |
| PP0LK<3> | PA5 | E11 | PP2LK<3> | PD5 | C12 |
| PP0LK<4> | PA6 | A10 | PP2LK<4> | PD6 | C13 |
| PP0LK<5> | PA7 | B10 | PP2LK<5> | PD7 | B13 |
| PP0LK<6> | PA8 | C10 | PP2LK<6> | PD8 | E13 |
| PP0LK<7> | PA9 | D10 | PP2LK<7> | PD9 | D13 |
| PP0LK<8> | PA11 | F10 | PP2LK<8> | PD11 | C17 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|---|---|
| PP0LK<9> | PA12 | E10 | PP2LK<9> | PD12 | B17 |
| PP0LK<10> | PA13 | C5 | PP2LK<10> | PD13 | D18 |
| PP0LK<11> | PA14 | C4 | PP2LK<11> | PD14 | C18 |

**Table 45: PLINK Pinouts – (2V1000)**

* PP0LK<0> is connected to a clock pin (GCLK6S) on the Virtex-II.

* PP0LK<1> is connected to a clock pin (GCLK7P) on the Virtex-II.

* PP2LK<0> is connected to a clock pin (GCLK2S) on the Virtex-II.

* PP2LK<1> is connected to a clock pin (GCLK3P) on the Virtex-II.

## 11.5.2 DIME-II control and monitoring signals

### DIME-II Specific Pins

| Dime-II Connector PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V1000) PIN No |
|---|---|---|---|
| CONFIG_DONE | FPGA_DONE | PC40 | L22 |
| N/c | CONFIG_DONE* | N/a | L21 |
| CLK0 | CLKA | PC24 | B11 |
| CLK1 | CLKB | PC31 | V11 |
| CLK2 | CLKC | PC42 | Y11 |
| RESETI | RESETI | PC15 | E14 |
| SLOT_ID0 | SLOT_ID0 | PC51 | J17 |
| SLOT_ID1 | SLOT_ID1 | PC52 | G22 |

**Table 46: User FPGA Specific Pinouts (2V1000)**

*CONFIG_DONE is driven from the IO of the Virtex-II into the base of the transistor to signal that the on-board User FPGA has been configured successfully. User to drive this pin LOW once the FPGA is configured.

### User LEDs

| Signal Name | User FPGA (2V1000) PIN No |
|---|---|
| LED_Green1 | V9 |
| LED_Red1 | N/A |
| | |
| LED_Green2 | V21 |
| LED_Red2 | V22 |

**Table 47: User LED Pinouts – (2V1000)**

### On-board Temperature Sensor

| Signal Name | User FPGA (2V1000) PIN No |
|---|---|
| DXN | D5 |
| DXP | A3 |
| ALERTI | T1 |

**Table 48: Temperature Sensor Pinouts**

## 11.5.3 Clock signals relating to DACs and ADCs

### Clock sources available at CLK FPGA

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V1000) PIN No | Signal Description |
|---|---|---|---|
| CLK_Op_Amp | B6 (GCLK6S) | N/a | External CLK source via Op_Amp |
| CLK_Op_Ampl | C6 (GCLK7P) | N/a | Complement of External CLK source via Op_Amp |
| EXT_CLK | B8 (GCLK0S) | N/a | External CLK source straight to CLK FPGA |
| EXT_CLKl | A8 (GCLK1P) | N/a | Complement of External CLK source straight to CLK FPGA |
| Osc_CLK | M6 (GCLK4P) | N/a | LVTTL Clock Oscillator |
| GEN_CLKA | K7 (GCLK0P) | Y10 | Generated Clock A |
| GEN_CLKC | N8 (GCLK1S) | AA10 | Generated Clock C |
| GEN_CLKB | M7 (GCLK6P) | V10 | Generated Clock B |
| GEN_CLKD | N7 (GCLK7S) | W10 | Generated Clock D |
| EXT2_CLK | D7 (GCLK5P) | N/a | 2nd External Clock |
| EXT2_CLKl | A6 (GCLK4S) | N/a | Complement of 2nd External Clock |

**Table 49: Clock Signals at CLK FPGA (2V1000)**

### Clock Feedback signals

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V1000) PIN No | Signal Description |
|---|---|---|---|
| CLK1_FB | J2 | Y12 | Feedback to User FPGA |
| CLK3_FB | H4 | W12 | Feedback to User FPGA |
| CLK2_FB | H12 | AA11 | Feedback to User FPGA |

**Table 50: Clock Feedback Signals (2V1000)**

### Clocking Pinouts for DACs and ADCs

| Signal Name | CLK FPGA (2V80) Pin No |
|---|---|
| ADC_CLKA | E4 |
| ADC_CLKAl | D1 |
| ADC_CLKB | G1 |
| ADC_CLKBl | F1 |
| DAC_CLKA | D13 |
| DAC_CLKAl | D12 |
| DAC_CLKB | G10 |
| DAC_CLKBl | F12 |

**Table 51: Clocking Pinouts for DACs and ADCs**

### 11.5.4 DAC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V1000) PIN No | Signal Name (DAC 2) | User FPGA (2V1000) PIN No |
|---|---|---|---|
| DAC1_D<0> | T22 | DAC2_D<0> | AB19 |
| DAC1_D<1> | R22 | DAC2_D<1> | AA19 |
| DAC1_D<2> | R21 | DAC2_D<2> | Y22 |
| DAC1_D<3> | R20 | DAC2_D<3> | Y21 |
| DAC1_D<4> | R19 | DAC2_D<4> | Y18 |
| DAC1_D<5> | P17 | DAC2_D<5> | Y17 |
| DAC1_D<6> | N22 | DAC2_D<6> | Y16 |
| DAC1_D<7> | N21 | DAC2_D<7> | W22 |
| DAC1_D<8> | N17 | DAC2_D<8> | W21 |
| DAC1_D<9> | M21 | DAC2_D<9> | W20 |
| DAC1_D<10> | M20 | DAC2_D<10> | W18 |
| DAC1_D<11> | M19 | DAC2_D<11> | W16 |
| DAC1_D<12> | M18 | DAC2_D<12> | W17 |
| DAC1_D<13> | M17 | DAC2_D<13> | V20 |
| DAC1_DIV0 | P18 | DAC2_DIV0 | V18 |
| DAC1_DIV1 | P22 | DAC2_DIV1 | V17 |
| DAC1_MOD0 | N19 | DAC2_MOD0 | AA15 |
| DAC1_MOD1 | N20 | DAC2_MOD1 | AB15 |
| DAC1_PLLLOCK | N18 | DAC2_PLLLOCK | P20 |
| DAC1_RESET | T21 | DAC2_RESET | P19 |

**Table 52: DACs Signal Pinouts (2V1000)**

### 11.5.5 ADC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V1000) PIN No | Signal Name (DAC 2) | User FPGA (2V1000) PIN No |
|---|---|---|---|
| ADC1_D<0> | AB18 | ADC2_D<0> | AB10 |
| ADC1_D<1> | AB16 | ADC2_D<1> | AB4 |
| ADC1_D<2> | AB17 | ADC2_D<2> | AB5 |
| ADC1_D<3> | AA18 | ADC2_D<3> | AA3 |
| ADC1_D<4> | AA16 | ADC2_D<4> | AA4 |
| ADC1_D<5> | AA17 | ADC2_D<5> | AA5 |
| ADC1_D<6> | AB13 | ADC2_D<6> | Y4 |
| ADC1_D<7> | AA13 | ADC2_D<7> | Y5 |
| ADC1_D<8> | Y13 | ADC2_D<8> | Y6 |
| ADC1_D<9> | W13 | ADC2_D<9> | W5 |
| ADC1_D<10> | W11 | ADC2_D<10> | W6 |
| ADC1_D<11> | V12 | ADC2_D<11> | V6 |
| ADC1_D<12> | V13 | ADC2_D<12> | V7 |
| ADC1_D<13> | U11 | ADC2_D<13> | V8 |
| ADC1_DRY | U12 | ADC2_DRY | U10 |
| ADC1_OVR | U13 | ADC2_OVR | W8 |

**Table 53: ADC Signal Pinouts (2V1000)**

### 11.5.6 User IO header

| Signal Name | User FPGA (2V1000) PIN No |
|---|---|
| User_IO_1 | AA8 |
| User_IO_2 | AB8 |

**Table 54: User IO Header Pinouts (2V1000)**

## 11.6 XC2V2000 FG676

### 11.6.1 User FPGA to DIME-II motherboard communication

#### Local Bus Pinouts

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| LBUS<0> | PB1 | D10 | LBUS<32> | PB37 | E10 |
| LBUS<1> | PB2 | G10 | LBUS<33> | PB38 | F10 |
| LBUS<2> | PB3 | G9 | LBUS<34> | PB39 | F9 |
| LBUS<3> | PB4 | C8 | LBUS<35> | PB40 | H7 |
| LBUS<4> | PB6 | C6 | LBUS<36> | PB42 | H6 |
| LBUS<5> | PB7 | D6 | LBUS<37> | PB43 | H4 |
| LBUS<6> | PB8 | F4 | LBUS<38> | PB44 | H5 |
| LBUS<7> | PB9 | G5 | LBUS<39> | PB45 | J6 |
| LBUS<8> | PB10 | F3 | LBUS<40> | PB46 | J5 |
| LBUS<9> | PB11 | G4 | LBUS<41> | PB47 | L7 |
| LBUS<10> | PB12 | G3 | LBUS<42> | PB48 | L4 |
| LBUS<11> | PB13 | K7 | LBUS<43> | PB49 | L3 |
| LBUS<12> | PB15 | J4 | LBUS<44> | PB51 | L6 |
| LBUS<13> | PB16 | J3 | LBUS<45> | PB52 | L5 |
| LBUS<14> | PB17 | L8 | LBUS<46> | PB53 | M8 |
| LBUS<15> | PB18 | K6 | LBUS<47> | PB54 | M7 |
| LBUS<16> | PB19 | K5 | LBUS<48> | PB55 | N8 |
| LBUS<17> | PB20 | K4 | LBUS<49> | PB56 | R7 |
| LBUS<18> | PB21 | K3 | LBUS<50> | PB57 | R5 |
| LBUS<19> | PB22 | M4 | LBUS<51> | PB58 | R6 |
| LBUS<20> | PB24 | M3 | LBUS<52> | PB60 | R8 |
| LBUS<21> | PB25 | M6 | LBUS<53> | PB61 | U7 |
| LBUS<22> | PB26 | M5 | LBUS<54> | PB62 | D2 |
| LBUS<23> | PB27 | N7 | LBUS<55> | PB63 | D1 |
| LBUS<24> | PB28 | N6 | LBUS<56> | PB64 | E2 |
| LBUS<25> | PB29 | N5 | LBUS<57> | PB65 | E1 |
| LBUS<26> | PB30 | N4 | LBUS<58> | PB66 | F2 |
| LBUS<27> | PB31 | P7 | LBUS<59> | PB67 | F1 |
| LBUS<28> | PB33 | R3 | LBUS<60> | PB69 | G1 |
| LBUS<29> | PB34 | R4 | LBUS<61> | PB70 | J7 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| LBUS<30> | PB35 | T8 | LBUS<62> | PB71 | H1 |
| LBUS<31> | PB36 | T7 | LBUS<63> | PB72 | J2 |

**Table 55: Local Bus Pinouts (2V2000)**

## Adjacent IN Bus – User FPGA communications

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| ADJIN<0> | PA29 | F14 | ADJIN<32> | PA65 | AB2 |
| ADJIN<1> | PA30 | G14 | ADJIN<33> | PA66 | AB1 |
| ADJIN<2> | PA31 | P4 | ADJIN<34> | PA67 | AC2 |
| ADJIN<3> | PA32 | P3 | ADJIN<35> | PA68 | AC1 |
| ADJIN<4> | PA33 | P6 | ADJIN<36> | PA69 | AD2 |
| ADJIN<5> | PA34 | P5 | ADJIN<37> | PA70 | AD1 |
| ADJIN<6> | PA35 | U6 | ADJIN<38> | PA71 | AF2 |
| ADJIN<7> | PA36 | U5 | ADJIN<39> | PA72 | AE1 |
| ADJIN<8> | PA38 | U4 | ADJIN<40> | SA2 | NC |
| ADJIN<9> | PA39 | U3 | ADJIN<41> | SB1 | NC |
| ADJIN<10> | PA40 | W6 | ADJIN<42> | SA3 | NC |
| ADJIN<11> | PA41 | W7 | ADJIN<43> | SB2 | NC |
| ADJIN<12> | PA42 | Y6 | ADJIN<44> | SA4 | NC |
| ADJIN<13> | PA43 | Y5 | ADJIN<45> | SB3 | NC |
| ADJIN<14> | PA44 | AA4 | ADJIN<46> | SA5 | NC |
| ADJIN<15> | PA45 | AA3 | ADJIN<47> | SB4 | NC |
| ADJIN<16> | PA47 | T4 | ADJIN<48> | SA6 | NC |
| ADJIN<17> | PA48 | T3 | ADJIN<49> | SB6 | NC |
| ADJIN<18> | PA49 | T5 | ADJIN<50> | SA7 | NC |
| ADJIN<19> | PA50 | T6 | ADJIN<51> | SB7 | NC |
| ADJIN<20> | PA51 | V5 | ADJIN<52> | SA8 | NC |
| ADJIN<21> | PA52 | V4 | ADJIN<53> | SB8 | NC |
| ADJIN<22> | PA53 | V6 | ADJIN<54> | SA9 | NC |
| ADJIN<23> | PA54 | V7 | ADJIN<55> | SB9 | NC |
| ADJIN<24> | PA56 | W5 | ADJIN<56> | SA11 | NC |
| ADJIN<25> | PA57 | W4 | ADJIN<57> | SB10 | NC |
| ADJIN<26> | PA58 | Y1 | ADJIN<58> | SA12 | NC |
| ADJIN<27> | PA59 | W1 | ADJIN<59> | SB11 | NC |
| ADJIN<28> | PA60 | W2 | ADJIN<60> | SA13 | NC |
| ADJIN<29> | PA61 | W3 | ADJIN<61> | SB12 | NC |
| ADJIN<30> | PA62 | AA2 | ADJIN<62> | SA14 | NC |
| ADJIN<31> | PA63 | AA1 | ADJIN<63> | SB13 | NC |

**Table 56: Adjacent IN BUS Pinouts - User FPGA (2V2000)**

## Adjacent OUT Bus - User FPGA connections

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| ADJOUT<0> | PD29 | D21 | ADJOUT<32> | PD65 | AD14 |
| ADJOUT<1> | PD30 | C21 | ADJOUT<33> | PD66 | AC14 |
| ADJOUT<2> | PD31 | E24 | ADJOUT<34> | PD67 | M21 |
| ADJOUT<3> | PD32 | E23 | ADJOUT<35> | PD68 | M22 |
| ADJOUT<4> | PD33 | G22 | ADJOUT<36> | PD69 | M26 |
| ADJOUT<5> | PD34 | G21 | ADJOUT<37> | PD70 | M25 |
| ADJOUT<6> | PD35 | F24 | ADJOUT<38> | PD71 | N20 |
| ADJOUT<7> | PD36 | F23 | ADJOUT<39> | PD72 | M20 |
| ADJOUT<8> | PD38 | K22 | ADJOUT<40> | SD2 | NC |
| ADJOUT<9> | PD39 | K21 | ADJOUT<41> | SC1 | NC |
| ADJOUT<10> | PD40 | K24 | ADJOUT<42> | SD3 | NC |
| ADJOUT<11> | PD41 | K23 | ADJOUT<43> | SC2 | NC |
| ADJOUT<12> | PD42 | M24 | ADJOUT<44> | SD4 | NC |
| ADJOUT<13> | PD43 | M23 | ADJOUT<45> | SC3 | NC |
| ADJOUT<14> | PD44 | N21 | ADJOUT<46> | SD5 | NC |
| ADJOUT<15> | PD45 | N22 | ADJOUT<47> | SC4 | NC |
| ADJOUT<16> | PD47 | H21 | ADJOUT<48> | SD6 | NC |
| ADJOUT<17> | PD48 | H22 | ADJOUT<49> | SC6 | NC |
| ADJOUT<18> | PD49 | H24 | ADJOUT<50> | SD7 | NC |
| ADJOUT<19> | PD50 | H23 | ADJOUT<51> | SC7 | NC |
| ADJOUT<20> | PD51 | J20 | ADJOUT<52> | SD8 | NC |
| ADJOUT<21> | PD52 | J21 | ADJOUT<53> | SC8 | NC |
| ADJOUT<22> | PD53 | J23 | ADJOUT<54> | SD9 | NC |
| ADJOUT<23> | PD54 | J22 | ADJOUT<55> | SC9 | NC |
| ADJOUT<24> | PD56 | L20 | ADJOUT<56> | SD11 | NC |
| ADJOUT<25> | PD57 | K20 | ADJOUT<57> | SC10 | NC |
| ADJOUT<26> | PD58 | K26 | ADJOUT<58> | SD12 | NC |
| ADJOUT<27> | PD59 | J26 | ADJOUT<59> | SC11 | NC |
| ADJOUT<28> | PD60 | L21 | ADJOUT<60> | SD13 | NC |
| ADJOUT<29> | PD61 | L22 | ADJOUT<61> | SC12 | NC |
| ADJOUT<30> | PD62 | L26 | ADJOUT<62> | SD14 | NC |
| ADJOUT<31> | PD63 | L25 | ADJOUT<63> | SC13 | NC |

**Table 57: Adjacent OUT BUS Pinouts - User FPGA (2V2000)**

## PLINKS connected to the User FPGA: PLINKS 0, 1, 2 and 3

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| PP0LK<0> | PA2 | E13 | PP2LK<0> | PD2 | H15 |
| PP0LK<1> | PA3 | F13 | PP2LK<1> | PD3 | H14 |
| PP0LK<2> | PA4 | H13 | PP2LK<2> | PD4 | D14 |
| PP0LK<3> | PA5 | G13 | PP2LK<3> | PD5 | E14 |
| PP0LK<4> | PA6 | C12 | PP2LK<4> | PD6 | E15 |
| PP0LK<5> | PA7 | D12 | PP2LK<5> | PD7 | D15 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| PP0LK<6> | PA8 | E12 | PP2LK<6> | PD8 | G15 |
| PP0LK<7> | PA9 | F12 | PP2LK<7> | PD9 | F15 |
| PP0LK<8> | PA11 | H12 | PP2LK<8> | PD11 | E19 |
| PP0LK<9> | PA12 | G12 | PP2LK<9> | PD12 | D19 |
| PP0LK<10> | PA13 | E7 | PP2LK<10> | PD13 | F20 |
| PP0LK<11> | PA14 | E6 | PP2LK<11> | PD14 | E20 |
| | | | | | |
| PP1LK<0> | PA15 | NC | PP3LK<0> | PD15 | NC |
| PP1LK<1> | PA16 | NC | PP3LK<1> | PD16 | NC |
| PP1LK<2> | PA17 | NC | PP3LK<2> | PD17 | NC |
| PP1LK<3> | PA18 | NC | PP3LK<3> | PD18 | NC |
| PP1LK<4> | PA20 | NC | PP3LK<4> | PD20 | NC |
| PP1LK<5> | PA21 | NC | PP3LK<5> | PD21 | NC |
| PP1LK<6> | PA22 | NC | PP3LK<6> | PD22 | NC |
| PP1LK<7> | PA23 | NC | PP3LK<7> | PD23 | NC |
| PP1LK<8> | PA24 | NC | PP3LK<8> | PD24 | NC |
| PP1LK<9> | PA25 | NC | PP3LK<9> | PD25 | NC |
| PP1LK<10> | PA26 | NC | PP3LK<10> | PD26 | NC |
| PP1LK<11> | PA27 | NC | PP3LK<11> | PD27 | NC |

**Table 58: PLINK Pinouts - User FPGA (2V2000)**

## PLINKS connected to the User FPGA: PLINKS 4, 5, 6 and 7

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| PP4LK<0> | SA48 | NC | PP6LK<0> | SD48 | NC |
| PP4LK<1> | SA49 | NC | PP6LK<1> | SD49 | NC |
| PP4LK<2> | SA50 | NC | PP6LK<2> | SD50 | NC |
| PP4LK<3> | SA51 | NC | PP6LK<3> | SD51 | NC |
| PP4LK<4> | SA52 | NC | PP6LK<4> | SD52 | NC |
| PP4LK<5> | SA53 | NC | PP6LK<5> | SD53 | NC |
| PP4LK<6> | SA54 | NC | PP6LK<6> | SD54 | NC |
| PP4LK<7> | SA56 | NC | PP6LK<7> | SD56 | NC |
| PP4LK<8> | SA57 | NC | PP6LK<8> | SD57 | NC |
| PP4LK<9> | SA58 | NC | PP6LK<9> | SD58 | NC |
| PP4LK<10> | SA59 | NC | PP6LK<10> | SD59 | NC |
| PP4LK<11> | SA60 | NC | PP6LK<11> | SD60 | NC |
| | | | | | |
| PP5LK<0> | SB47 | B12 | PP7LK<0> | SC47 | A17 |
| PP5LK<1> | SB48 | B11 | PP7LK<1> | SC48 | A16 |
| PP5LK<2> | SB49 | C11 | PP7LK<2> | SC49 | C16 |
| PP5LK<3> | SB51 | D11 | PP7LK<3> | SC51 | B16 |
| PP5LK<4> | SB52 | H11 | PP7LK<4> | SC52 | E16 |
| PP5LK<5> | SB53 | G11 | PP7LK<5> | SC53 | D16 |
| PP5LK<6> | SB54 | E11 | PP7LK<6> | SC54 | D17 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|---|---|
| PP5LK<7> | SB55 | F11 | PP7LK<7> | SC55 | C17 |
| PP5LK<8> | SB56 | A11 | PP7LK<8> | SC56 | G17 |
| PP5LK<9> | SB57 | A10 | PP7LK<9> | SC57 | H16 |
| PP5LK<10> | SB58 | C10 | PP7LK<10> | SC58 | F17 |
| PP5LK<11> | SB60 | C9 | PP7LK<11> | SC60 | E17 |

**Table 59: PLINK Pinouts - User FPGA (2V2000)**

## 11.6.2 DIME-II control and monitoring signals

### DIME-II Specific Pins

| Dime-II Connector PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| CONFIG_DONE | FPGA_DONE | PC40 | N24 |
| N/C | CONFIG_DONE* | N/a | N23 |
| CLK0 | CLKA | PC24 | D13 |
| CLK1 | CLKB | PC31 | Y13 |
| CLK2 | CLKC | PC42 | AB13 |
| RESETI | RESETI | PC15 | G16 |
| SLOT_ID0 | SLOT_ID0 | PC51 | L19 |
| SLOT_ID1 | SLOT_ID1 | PC52 | J24 |

**Table 60: User FPGA Specific Pinouts (2V2000)**

*CONFIG_DONE is driven from the IO of the Virtex-II into the base of the transistor to signal that the on-board User FPGA has been configured successfully. User to drive this pin LOW once the FPGA is configured.

### User LEDs

| Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|
| LED_Green1 | Y11 |
| LED_Red1 | AA11 |
| | |
| LED_Green2 | Y23 |
| LED_Red2 | Y24 |

**Table 61: User LED Pinouts - User FPGA (2V2000)**

### On-board Temperature Sensor

| Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|
| DXN | NC |
| DXP | NC |
| ALERTI | V3 |

**Table 62: User LED Pinouts - User FPGA (2V2000)**

## 11.6.3 ZBT SRAM BANK

### Clock and Control Signals for ZBT Bank

| Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|
| ZBT_CLK | A13 |
| ZBT_FB_OUT | A12 |
| ZBT_FB_IN | C13 |
| ZBT_ADV | B15 |
| ZBT_CKEl | C15 |
| ZBT_CSl<0> | A18 |
| ZBT_CSl<1> | A19 |
| ZBT_Oel | A15 |
| ZBT_Wel | A14 |

**Table 63: ZBT Clock and Control Signals Pinouts (2V2000)**

### ZBT Address Signals – Bank

| Signal Name | User FPGA (2V2000FG676) PIN No | Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| ZBT_A<0> | C2 | ZBT_A<11> | E9 |
| ZBT_A<1> | B6 | ZBT_A<12> | G6 |
| ZBT_A<2> | B4 | ZBT_A<13> | A3 |
| ZBT_A<3> | B5 | ZBT_A<14> | C1 |
| ZBT_A<4> | D9 | ZBT_A<15> | A9 |
| ZBT_A<5> | B3 | ZBT_A<16> | B9 |
| ZBT_A<6> | B1 | ZBT_A<17> | A5 |
| ZBT_A<7> | B8 | ZBT_A<18>* | A4 |
| ZBT_A<8> | A8 | ZBT_A<19>* | A2 |
| ZBT_A<9> | A7 | ZBT_A<20>* | G7 |
| ZBT_A<10> | A6 | ZBT_A<21>* | NC |

**Table 64: ZBT Address Signals Pinouts – Ban (2V2000)**

*These are Expansion PINs and may not be supported by the ZBT memory. Only use the above address pins if the onboard ZBT memory is in the following list:

- ZBT_A<18> use if ZBT Memory is 16Mb
- ZBT_A<19> use if ZBT Memory is 32Mb
- ZBT_A<20> use if ZBT Memory is 64Mb
- ZBT_A<20> use if ZBT Memory is 128Mb

### ZBT Data Signals – Bank

| Signal Name | User FPGA (2V2000FG676) PIN No | Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| ZBT_D<0> | B18 | ZBT_D<16> | NC |
| ZBT_D<1> | C18 | ZBT_D<17> | NC |
| ZBT_D<2> | D18 | ZBT_D<18> | NC |
| ZBT_D<3> | E18 | ZBT_D<19> | NC |

| Signal Name | User FPGA (2V2000FG676) PIN No | Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| ZBT_D<4> | A25 | ZBT_D<20> | NC |
| ZBT_D<5> | A24 | ZBT_D<21> | NC |
| ZBT_D<6> | B24 | ZBT_D<22> | NC |
| ZBT_D<7> | A23 | ZBT_D<23> | NC |
| ZBT_D<8> | A21 | ZBT_D<24> | NC |
| ZBT_D<9> | A20 | ZBT_D<25> | NC |
| ZBT_D<10> | C19 | ZBT_D<26> | NC |
| ZBT_D<11> | B19 | ZBT_D<27> | NC |
| ZBT_D<12> | B23 | ZBT_D<28> | NC |
| ZBT_D<13> | A22 | ZBT_D<29> | NC |
| ZBT_D<14> | B22 | ZBT_D<30> | NC |
| ZBT_D<15> | B21 | ZBT_D<31> | NC |

**Table 65: ZBT Data Signals Pinouts – Bank (2V2000)**

### ZBT Parity Bits – Bank

| Signal Name | User FPGA (2V2000FG676) PIN No | Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| ZBT_P<0> | NC | ZBT_P<2> | NC |
| ZBT_P<1> | NC | ZBT_P<3> | NC |

**Table 66: ZBT Parity Bits Pinouts – Bank (2V2000)**

## 11.6.4 Clock signals relating to DACs and ADCs

### Clock sources arriving at CLK FPGA

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V2000FG676) PIN No | Signal Description |
|---|---|---|---|
| CLK_Op_Amp | B6 (GCLK6S) | N/a | External CLK source via Op_Amp |
| CLK_Op_Ampl | C6 (GCLK7P) | N/a | Complement of External CLK source via Op_Amp |
| EXT_CLK | B8 (GCLK0S) | N/a | External CLK source straight to CLK FPGA |
| EXT_CLKl | A8 (GCLK1P) | N/a | Complement of External CLK source straight to CLK FPGA |
| Osc_CLK | M6 (GCLK4P) | N/a | LVTTL Clock Oscillator |
| GEN_CLKA | K7 (GCLK0P) | AB12 | Generated Clock A |
| GEN_CLKC | N8 (GCLK1S) | AC12 | Generated Clock C |
| GEN_CLKB | M7 (GCLK6P) | Y12 | Generated Clock B |
| GEN_CLKD | N7 (GCLK7S) | AA12 | Generated Clock D |
| EXT2_CLK | D7 (GCLK5P) | N/a | 2nd External Clock |
| EXT2_CLKl | A6 (GCLK4S) | N/a | Complement of 2nd External Clock |

**Table 67: Clock Signals at CLK FPGA (2V2000)**

### Clock Feedback signals

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V2000FG676) PIN No | Signal Description |
|---|---|---|---|
| CLK1_FB | J2 | AB14 | Feedback to User FPGA |
| CLK3_FB | H4 | AA14 | Feedback to User FPGA |
| CLK2_FB | H12 | AC13 | Feedback to User FPGA |

**Table 68: Clock Feedback Signals (2V2000)**

### Clocking Pinouts for DACs and ADCs

| Signal Name | CLK FPGA (2V80) Pin No |
|---|---|
| ADC_CLKA | E4 |
| ADC_CLKAI | D1 |
| ADC_CLKB | G1 |
| ADC_CLKBI | F1 |
| DAC_CLKA | D13 |
| DAC_CLKAI | D12 |
| DAC_CLKB | G10 |
| DAC_CLKBI | F12 |

**Table 69: Clocking Pinouts for DACs and ADCs**

## 11.6.5 DAC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V2000FG676) PIN No | Signal Name (DAC 2) | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| DAC1_D<0> | V24 | DAC2_D<0> | AD21 |
| DAC1_D<1> | U24 | DAC2_D<1> | AC21 |
| DAC1_D<2> | U23 | DAC2_D<2> | AB24 |
| DAC1_D<3> | U22 | DAC2_D<3> | AB23 |
| DAC1_D<4> | U21 | DAC2_D<4> | AB20 |
| DAC1_D<5> | T19 | DAC2_D<5> | AB18 |
| DAC1_D<6> | R24 | DAC2_D<6> | AA24 |
| DAC1_D<7> | R23 | DAC2_D<7> | AA23 |
| DAC1_D<8> | R19 | DAC2_D<8> | AA22 |
| DAC1_D<9> | P23 | DAC2_D<9> | AA20 |
| DAC1_D<10> | P22 | DAC2_D<10> | AA18 |
| DAC1_D<11> | P21 | DAC2_D<11> | Y22 |
| DAC1_D<12> | P20 | DAC2_D<12> | Y20 |
| DAC1_D<13> | P19 | DAC2_D<13> | Y19 |
| DAC1_DIV0 | T20 | DAC2_DIV0 | AC17 |
| DAC1_DIV1 | T24 | DAC2_DIV1 | AD17 |
| DAC1_MOD0 | R21 | DAC2_MOD0 | T22 |
| DAC1_MOD1 | R22 | DAC2_MOD1 | T21 |
| DAC1_PLLLOCK | R20 | DAC2_PLLLOCK | AB16 |
| DAC1_RESET | V23 | DAC2_RESET | U20 |

**Table 70: DACs Signal Pinouts (2V2000)**

### 11.6.6 ADC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V2000FG676) PIN No | Signal Name (DAC 2) | User FPGA (2V2000FG676) PIN No |
|---|---|---|---|
| ADC1_D<0> | AF20 | ADC2_D<0> | AD12 |
| ADC1_D<1> | AD18 | ADC2_D<1> | AD6 |
| ADC1_D<2> | AC22 | ADC2_D<2> | AD7 |
| ADC1_D<3> | AC18 | ADC2_D<3> | AC5 |
| ADC1_D<4> | AD15 | ADC2_D<4> | AC6 |
| ADC1_D<5> | AC15 | ADC2_D<5> | AC7 |
| ADC1_D<6> | AB15 | ADC2_D<6> | AB6 |
| ADC1_D<7> | AA15 | ADC2_D<7> | AB7 |
| ADC1_D<8> | AA13 | ADC2_D<8> | AB8 |
| ADC1_D<9> | Y14 | ADC2_D<9> | AA7 |
| ADC1_D<10> | Y15 | ADC2_D<10> | AA8 |
| ADC1_D<11> | W13 | ADC2_D<11> | Y8 |
| ADC1_D<12> | W14 | ADC2_D<12> | Y10 |
| ADC1_D<13> | W15 | ADC2_D<13> | W12 |
| ADC1_DRY | Y16 | ADC2_DRY | AA10 |
| ADC1_OVR | AA16 | ADC2_OVR | AB10 |

**Table 71: ADC Signal Pinouts (2V2000)**

### 11.6.7 User IO Header

| Signal Name | User FPGA (2V2000FG676) PIN No |
|---|---|
| User_IO_1 | AC10 |
| User_IO_2 | AD10 |

**Table 72: User IO Header Pinouts (2V2000)**

## 11.7 XC2V3000 FG676

### 11.7.1 User FPGA to DIME-II motherboard communication

#### Local Bus Pinouts

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| LBUS<0> | PB1 | F8 | LBUS<32> | PB37 | E10 |
| LBUS<1> | PB2 | G10 | LBUS<33> | PB38 | F10 |
| LBUS<2> | PB3 | G9 | LBUS<34> | PB39 | F9 |
| LBUS<3> | PB4 | C8 | LBUS<35> | PB40 | H7 |
| LBUS<4> | PB6 | C6 | LBUS<36> | PB42 | H6 |
| LBUS<5> | PB7 | D6 | LBUS<37> | PB43 | H4 |
| LBUS<6> | PB8 | F4 | LBUS<38> | PB44 | H5 |
| LBUS<7> | PB9 | G5 | LBUS<39> | PB45 | J6 |
| LBUS<8> | PB10 | F3 | LBUS<40> | PB46 | J5 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| LBUS<9> | PB11 | G4 | LBUS<41> | PB47 | L7 |
| LBUS<10> | PB12 | G3 | LBUS<42> | PB48 | L4 |
| LBUS<11> | PB13 | K7 | LBUS<43> | PB49 | L3 |
| LBUS<12> | PB15 | J4 | LBUS<44> | PB51 | L6 |
| LBUS<13> | PB16 | J3 | LBUS<45> | PB52 | L5 |
| LBUS<14> | PB17 | L8 | LBUS<46> | PB53 | M8 |
| LBUS<15> | PB18 | K6 | LBUS<47> | PB54 | M7 |
| LBUS<16> | PB19 | K5 | LBUS<48> | PB55 | N8 |
| LBUS<17> | PB20 | K4 | LBUS<49> | PB56 | R7 |
| LBUS<18> | PB21 | K3 | LBUS<50> | PB57 | R5 |
| LBUS<19> | PB22 | M4 | LBUS<51> | PB58 | R6 |
| LBUS<20> | PB24 | M3 | LBUS<52> | PB60 | R8 |
| LBUS<21> | PB25 | M6 | LBUS<53> | PB61 | U7 |
| LBUS<22> | PB26 | M5 | LBUS<54> | PB62 | D2 |
| LBUS<23> | PB27 | N7 | LBUS<55> | PB63 | D1 |
| LBUS<24> | PB28 | N6 | LBUS<56> | PB64 | E2 |
| LBUS<25> | PB29 | N5 | LBUS<57> | PB65 | E1 |
| LBUS<26> | PB30 | N4 | LBUS<58> | PB66 | F2 |
| LBUS<27> | PB31 | P7 | LBUS<59> | PB67 | F1 |
| LBUS<28> | PB33 | R3 | LBUS<60> | PB69 | G1 |
| LBUS<29> | PB34 | R4 | LBUS<61> | PB70 | J7 |
| LBUS<30> | PB35 | T8 | LBUS<62> | PB71 | H1 |
| LBUS<31> | PB36 | T7 | LBUS<63> | PB72 | J2 |

**Table 73: Local Bus Pinouts (2V3000)**

## Adjacent IN Bus – User FPGA communications

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| ADJIN<0> | PA29 | F14 | ADJIN<32> | PA65 | AB2 |
| ADJIN<1> | PA30 | G14 | ADJIN<33> | PA66 | AB1 |
| ADJIN<2> | PA31 | P4 | ADJIN<34> | PA67 | AC2 |
| ADJIN<3> | PA32 | P3 | ADJIN<35> | PA68 | AC1 |
| ADJIN<4> | PA33 | P6 | ADJIN<36> | PA69 | AD2 |
| ADJIN<5> | PA34 | P5 | ADJIN<37> | PA70 | AD1 |
| ADJIN<6> | PA35 | U6 | ADJIN<38> | PA71 | AF2 |
| ADJIN<7> | PA36 | U5 | ADJIN<39> | PA72 | AE1 |
| ADJIN<8> | PA38 | U4 | ADJIN<40> | SA2 | NC |
| ADJIN<9> | PA39 | U3 | ADJIN<41> | SB1 | NC |
| ADJIN<10> | PA40 | W6 | ADJIN<42> | SA3 | NC |
| ADJIN<11> | PA41 | W7 | ADJIN<43> | SB2 | NC |
| ADJIN<12> | PA42 | Y6 | ADJIN<44> | SA4 | NC |
| ADJIN<13> | PA43 | Y5 | ADJIN<45> | SB3 | NC |
| ADJIN<14> | PA44 | AA4 | ADJIN<46> | SA5 | NC |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| ADJIN<15> | PA45 | AA3 | ADJIN<47> | SB4 | NC |
| ADJIN<16> | PA47 | T4 | ADJIN<48> | SA6 | NC |
| ADJIN<17> | PA48 | T3 | ADJIN<49> | SB6 | NC |
| ADJIN<18> | PA49 | T5 | ADJIN<50> | SA7 | NC |
| ADJIN<19> | PA50 | T6 | ADJIN<51> | SB7 | NC |
| ADJIN<20> | PA51 | V5 | ADJIN<52> | SA8 | NC |
| ADJIN<21> | PA52 | V4 | ADJIN<53> | SB8 | NC |
| ADJIN<22> | PA53 | V6 | ADJIN<54> | SA9 | NC |
| ADJIN<23> | PA54 | V7 | ADJIN<55> | SB9 | NC |
| ADJIN<24> | PA56 | W5 | ADJIN<56> | SA11 | NC |
| ADJIN<25> | PA57 | W4 | ADJIN<57> | SB10 | NC |
| ADJIN<26> | PA58 | Y1 | ADJIN<58> | SA12 | NC |
| ADJIN<27> | PA59 | W1 | ADJIN<59> | SB11 | NC |
| ADJIN<28> | PA60 | W2 | ADJIN<60> | SA13 | NC |
| ADJIN<29> | PA61 | W3 | ADJIN<61> | SB12 | NC |
| ADJIN<30> | PA62 | AA2 | ADJIN<62> | SA14 | NC |
| ADJIN<31> | PA63 | AA1 | ADJIN<63> | SB13 | NC |

**Table 74: Adjacent IN BUS Pinouts - User FPGA (2V3000)**

## Adjacent OUT Bus - User FPGA connections

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| ADJOUT<0> | PD29 | D21 | ADJOUT<32> | PD65 | AD14 |
| ADJOUT<1> | PD30 | C21 | ADJOUT<33> | PD66 | AC14 |
| ADJOUT<2> | PD31 | E24 | ADJOUT<34> | PD67 | M21 |
| ADJOUT<3> | PD32 | E23 | ADJOUT<35> | PD68 | M22 |
| ADJOUT<4> | PD33 | G22 | ADJOUT<36> | PD69 | M26 |
| ADJOUT<5> | PD34 | G21 | ADJOUT<37> | PD70 | M25 |
| ADJOUT<6> | PD35 | F24 | ADJOUT<38> | PD71 | N20 |
| ADJOUT<7> | PD36 | F23 | ADJOUT<39> | PD72 | M20 |
| ADJOUT<8> | PD38 | K22 | ADJOUT<40> | SD2 | NC |
| ADJOUT<9> | PD39 | K21 | ADJOUT<41> | SC1 | NC |
| ADJOUT<10> | PD40 | K24 | ADJOUT<42> | SD3 | NC |
| ADJOUT<11> | PD41 | K23 | ADJOUT<43> | SC2 | NC |
| ADJOUT<12> | PD42 | M24 | ADJOUT<44> | SD4 | NC |
| ADJOUT<13> | PD43 | M23 | ADJOUT<45> | SC3 | NC |
| ADJOUT<14> | PD44 | N21 | ADJOUT<46> | SD5 | NC |
| ADJOUT<15> | PD45 | N22 | ADJOUT<47> | SC4 | NC |
| ADJOUT<16> | PD47 | H21 | ADJOUT<48> | SD6 | NC |
| ADJOUT<17> | PD48 | H22 | ADJOUT<49> | SC6 | NC |
| ADJOUT<18> | PD49 | H24 | ADJOUT<50> | SD7 | NC |
| ADJOUT<19> | PD50 | H23 | ADJOUT<51> | SC7 | NC |
| ADJOUT<20> | PD51 | J20 | ADJOUT<52> | SD8 | NC |
| ADJOUT<21> | PD52 | J21 | ADJOUT<53> | SC8 | NC |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| ADJOUT<22> | PD53 | J23 | ADJOUT<54> | SD9 | NC |
| ADJOUT<23> | PD54 | J22 | ADJOUT<55> | SC9 | NC |
| ADJOUT<24> | PD56 | L20 | ADJOUT<56> | SD11 | NC |
| ADJOUT<25> | PD57 | K20 | ADJOUT<57> | SC10 | NC |
| ADJOUT<26> | PD58 | K26 | ADJOUT<58> | SD12 | NC |
| ADJOUT<27> | PD59 | J26 | ADJOUT<59> | SC11 | NC |
| ADJOUT<28> | PD60 | L21 | ADJOUT<60> | SD13 | NC |
| ADJOUT<29> | PD61 | L22 | ADJOUT<61> | SC12 | NC |
| ADJOUT<30> | PD62 | L26 | ADJOUT<62> | SD14 | NC |
| ADJOUT<31> | PD63 | L25 | ADJOUT<63> | SC13 | NC |

**Table 75: Adjacent OUT BUS Pinouts - User FPGA (2V3000)**

## PLINKS connected to the User FPGA: PLINKS 0, 1, 2 and 3

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| PP0LK<0> | PA2 | E13 | PP2LK<0> | PD2 | H15 |
| PP0LK<1> | PA3 | F13 | PP2LK<1> | PD3 | H14 |
| PP0LK<2> | PA4 | H13 | PP2LK<2> | PD4 | D14 |
| PP0LK<3> | PA5 | G13 | PP2LK<3> | PD5 | E14 |
| PP0LK<4> | PA6 | C12 | PP2LK<4> | PD6 | E15 |
| PP0LK<5> | PA7 | D12 | PP2LK<5> | PD7 | D15 |
| PP0LK<6> | PA8 | E12 | PP2LK<6> | PD8 | G15 |
| PP0LK<7> | PA9 | F12 | PP2LK<7> | PD9 | F15 |
| PP0LK<8> | PA11 | H12 | PP2LK<8> | PD11 | E19 |
| PP0LK<9> | PA12 | G12 | PP2LK<9> | PD12 | D19 |
| PP0LK<10> | PA13 | E7 | PP2LK<10> | PD13 | F20 |
| PP0LK<11> | PA14 | E6 | PP2LK<11> | PD14 | E20 |
| | | | | | |
| PP1LK<0> | PA15 | NC | PP3LK<0> | PD15 | NC |
| PP1LK<1> | PA16 | NC | PP3LK<1> | PD16 | NC |
| PP1LK<2> | PA17 | NC | PP3LK<2> | PD17 | NC |
| PP1LK<3> | PA18 | NC | PP3LK<3> | PD18 | NC |
| PP1LK<4> | PA20 | NC | PP3LK<4> | PD20 | NC |
| PP1LK<5> | PA21 | NC | PP3LK<5> | PD21 | NC |
| PP1LK<6> | PA22 | NC | PP3LK<6> | PD22 | NC |
| PP1LK<7> | PA23 | NC | PP3LK<7> | PD23 | NC |
| PP1LK<8> | PA24 | NC | PP3LK<8> | PD24 | NC |
| PP1LK<9> | PA25 | NC | PP3LK<9> | PD25 | NC |
| PP1LK<10> | PA26 | NC | PP3LK<10> | PD26 | NC |
| PP1LK<11> | PA27 | NC | PP3LK<11> | PD27 | NC |

**Table 76: PLINK Pinouts - User FPGA (2V3000)**

## PLINKS connected to the User FPGA: PLINKS 4, 5, 6 and 7

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|---|---|
| PP4LK<0> | SA48 | NC | PP6LK<0> | SD48 | NC |
| PP4LK<1> | SA49 | NC | PP6LK<1> | SD49 | NC |
| PP4LK<2> | SA50 | NC | PP6LK<2> | SD50 | NC |
| PP4LK<3> | SA51 | NC | PP6LK<3> | SD51 | NC |
| PP4LK<4> | SA52 | NC | PP6LK<4> | SD52 | NC |
| PP4LK<5> | SA53 | NC | PP6LK<5> | SD53 | NC |
| PP4LK<6> | SA54 | NC | PP6LK<6> | SD54 | NC |
| PP4LK<7> | SA56 | NC | PP6LK<7> | SD56 | NC |
| PP4LK<8> | SA57 | NC | PP6LK<8> | SD57 | NC |
| PP4LK<9> | SA58 | NC | PP6LK<9> | SD58 | NC |
| PP4LK<10> | SA59 | NC | PP6LK<10> | SD59 | NC |
| PP4LK<11> | SA60 | NC | PP6LK<11> | SD60 | NC |
| | | | | | |
| PP5LK<0> | SB47 | B12 | PP7LK<0> | SC47 | A17 |
| PP5LK<1> | SB48 | B11 | PP7LK<1> | SC48 | A16 |
| PP5LK<2> | SB49 | C11 | PP7LK<2> | SC49 | C16 |
| PP5LK<3> | SB51 | D11 | PP7LK<3> | SC51 | B16 |
| PP5LK<4> | SB52 | H11 | PP7LK<4> | SC52 | E16 |
| PP5LK<5> | SB53 | G11 | PP7LK<5> | SC53 | D16 |
| PP5LK<6> | SB54 | E11 | PP7LK<6> | SC54 | D17 |
| PP5LK<7> | SB55 | F11 | PP7LK<7> | SC55 | C17 |
| PP5LK<8> | SB56 | A11 | PP7LK<8> | SC56 | G17 |
| PP5LK<9> | SB57 | A10 | PP7LK<9> | SC57 | H16 |
| PP5LK<10> | SB58 | C10 | PP7LK<10> | SC58 | F17 |
| PP5LK<11> | SB60 | C9 | PP7LK<11> | SC60 | E17 |

**Table 77: PLINK Pinouts - User FPGA (2V3000)**

## 11.7.2 DIME-II control and monitoring signals

### DIME-II Specific Pins

| Dime-II Connector PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| CONFIG_DONE | FPGA_DONE | PC40 | N24 |
| N/C | CONFIG_DONE* | N/a | N23 |
| CLK0 | CLKA | PC24 | D13 |
| CLK1 | CLKB | PC31 | Y13 |
| CLK2 | CLKC | PC42 | AB13 |
| RESETI | RESETI | PC15 | G16 |
| SLOT_ID0 | SLOT_ID0 | PC51 | L19 |
| SLOT_ID1 | SLOT_ID1 | PC52 | J24 |

**Table 78: User FPGA Specific Pinouts (2V3000)**

*CONFIG_DONE is driven from the IO of the Virtex-II into the base of the transistor to signal that the on-board User FPGA has been configured successfully. User to drive this pin LOW once the FPGA is configured.

### User LEDs

| Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|
| LED_Green1 | Y11 |
| LED_Red1 | AA11 |
| | |
| LED_Green2 | Y23 |
| LED_Red2 | Y24 |

**Table 79: User LED Pinouts - User FPGA (2V3000)**

### On-board Temperature Sensor

| Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|
| DXN | NC |
| DXP | NC |
| ALERTI | V3 |

**Table 80: Temperature Sensor Pinouts (2V3000)**

## 11.7.3 ZBT SRAM BANK

### Clock and Control Signals for ZBT Bank

| Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|
| ZBT_CLK | A13 |
| ZBT_FB_OUT | A12 |
| ZBT_FB_IN | C13 |

| Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|
| ZBT_ADV | B15 |
| ZBT_CKEI | C15 |
| ZBT_CSI<0> | A18 |
| ZBT_CSI<1> | A19 |
| ZBT_OEI | A15 |
| ZBT_WEI | A14 |

**Table 81: ZBT Clock and Control Signals Pinouts – Bank**

## ZBT Address Signals – Bank

| Signal Name | User FPGA (2V3000FG676) PIN No | Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| ZBT_A<0> | C2 | ZBT_A<11> | E9 |
| ZBT_A<1> | B6 | ZBT_A<12> | G6 |
| ZBT_A<2> | B4 | ZBT_A<13> | A3 |
| ZBT_A<3> | B5 | ZBT_A<14> | C1 |
| ZBT_A<4> | D9 | ZBT_A<15> | A9 |
| ZBT_A<5> | B3 | ZBT_A<16> | B9 |
| ZBT_A<6> | B1 | ZBT_A<17> | A5 |
| ZBT_A<7> | B8 | ZBT_A<18>* | A4 |
| ZBT_A<8> | A8 | ZBT_A<19>* | A2 |
| ZBT_A<9> | A7 | ZBT_A<20>* | G7 |
| ZBT_A<10> | A6 | ZBT_A<21>* | NC |

**Table 82: ZBT Address Signals Pinouts – Bank**

*These are Expansion PINs and may not be supported by the ZBT memory. Only use the above address pins if the onboard ZBT memory is in the following list:

- ZBT_A<18> use if ZBT Memory is 16Mb
- ZBT_A<19> use if ZBT Memory is 32Mb
- ZBT_A<20> use if ZBT Memory is 64Mb
- ZBT_A<20> use if ZBT Memory is 128Mb

## ZBT Data Signals – Bank

| Signal Name | User FPGA (2V3000FG676) PIN No | Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| ZBT_D<0> | B18 | ZBTA_D<16> | NC |
| ZBT_D<1> | C18 | ZBT_D<17> | NC |
| ZBT_D<2> | D18 | ZBT_D<18> | NC |
| ZBT_D<3> | E18 | ZBT_D<19> | NC |
| ZBT_D<4> | A25 | ZBT_D<20> | NC |
| ZBT_D<5> | A24 | ZBT_D<21> | NC |
| ZBT_D<6> | B24 | ZBT_D<22> | NC |
| ZBT_D<7> | A23 | ZBT_D<23> | NC |
| ZBT_D<8> | A21 | ZBT_D<24> | NC |

| Signal Name | User FPGA (2V3000FG676) PIN No | Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| ZBT_D<9> | A20 | ZBT_D<25> | NC |
| ZBT_D<10> | C19 | ZBT_D<26> | NC |
| ZBT_D<11> | B19 | ZBT_D<27> | NC |
| ZBT_D<12> | B23 | ZBT_D<28> | NC |
| ZBT_D<13> | A22 | ZBT_D<29> | NC |
| ZBT_D<14> | B22 | ZBT_D<30> | NC |
| ZBT_D<15> | B21 | ZBT_D<31> | NC |

**Table 83: ZBT Data Signals Pinouts – Bank**

## ZBT Parity Bits – Bank

| Signal Name | User FPGA (2V3000FG676) PIN No | Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| ZBT_P<0> | NC | ZBT_P<2> | NC |
| ZBT_P<1> | NC | ZBT_P<3> | NC |

**Table 84: ZBT Parity Bits Pinouts – Bank**

## 11.7.4 Clock signals relating to DACs and ADCs

### Clock sources arriving at CLK FPGA

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V3000FG676) PIN No | Signal Description |
|---|---|---|---|
| CLK_Op_Amp | B6 (GCLK6S) | N/a | External CLK source via Op_Amp |
| CLK_Op_Ampl | C6 (GCLK7P) | N/a | Complement of External CLK source via Op_Amp |
| EXT_CLK | B8 (GCLK0S) | N/a | External CLK source straight to CLK FPGA |
| EXT_CLKl | A8 (GCLK1P) | N/a | Complement of External CLK source straight to CLK FPGA |
| Osc_CLK | M6 (GCLK4P) | N/a | LVTTL Clock Oscillator |
| GEN_CLKA | K7 (GCLK0P) | AB12 | Generated Clock A |
| GEN_CLKC | N8 (GCLK1S) | AC12 | Generated Clock C |
| GEN_CLKB | M7 (GCLK6P) | Y12 | Generated Clock B |
| GEN_CLKD | N7 (GCLK7S) | AA12 | Generated Clock D |
| EXT2_CLK | D7 (GCLK5P) | N/a | 2nd External Clock |
| EXT2_CLKl | A6 (GCLK4S) | N/a | Complement of 2nd External Clock |

**Table 85: Clock Signals at CLK FPGA**

### Clock Feedback signals

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V3000FG676) PIN No | Signal Description |
|---|---|---|---|
| CLK1_FB | J2 | AB14 | Feedback to User FPGA |
| CLK3_FB | H4 | AA14 | Feedback to User FPGA |
| CLK2_FB | H12 | AC13 | Feedback to User FPGA |

**Table 86: Clock Feedback Signals**

### Clocking Pinouts for DACs and ADCs

| Signal Name | CLK FPGA (2V80) Pin No |
|---|---|
| ADC_CLKA | E4 |
| ADC_CLKAl | D1 |
| ADC_CLKB | G1 |
| ADC_CLKBl | F1 |
| DAC_CLKA | D13 |
| DAC_CLKAl | D12 |
| DAC_CLKB | G10 |
| DAC_CLKBl | F12 |

**Table 87: Clocking Pinouts for DACs and ADCs**

## 11.7.5 DAC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V3000FG676) PIN No | Signal Name (DAC 2) | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|

| Signal Name (DAC 1) | User FPGA (2V3000FG676) PIN No | Signal Name (DAC 2) | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| DAC1_D<0> | V24 | DAC2_D<0> | AD21 |
| DAC1_D<1> | U24 | DAC2_D<1> | AC21 |
| DAC1_D<2> | U23 | DAC2_D<2> | AB24 |
| DAC1_D<3> | U22 | DAC2_D<3> | AB23 |
| DAC1_D<4> | U21 | DAC2_D<4> | AB20 |
| DAC1_D<5> | T19 | DAC2_D<5> | AB19 |
| DAC1_D<6> | R24 | DAC2_D<6> | AA24 |
| DAC1_D<7> | R23 | DAC2_D<7> | AA23 |
| DAC1_D<8> | R19 | DAC2_D<8> | AA22 |
| DAC1_D<9> | P23 | DAC2_D<9> | AA20 |
| DAC1_D<10> | P22 | DAC2_D<10> | AA19 |
| DAC1_D<11> | P21 | DAC2_D<11> | Y22 |
| DAC1_D<12> | P20 | DAC2_D<12> | Y20 |
| DAC1_D<13> | P19 | DAC2_D<13> | Y19 |
| DAC1_DIV0 | T20 | DAC2_DIV0 | AC17 |
| DAC1_DIV1 | T24 | DAC2_DIV1 | AD17 |
| DAC1_MOD0 | R21 | DAC2_MOD0 | T22 |
| DAC1_MOD1 | R22 | DAC2_MOD1 | T21 |
| DAC1_PLLLOCK | R20 | DAC2_PLLLOCK | AB16 |
| DAC1_RESET | V23 | DAC2_RESET | U20 |

**Table 88: DACs Signal Pinouts**

### 11.7.6 ADC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V3000FG676) PIN No | Signal Name (DAC 2) | User FPGA (2V3000FG676) PIN No |
|---|---|---|---|
| ADC1_D<0> | AF20 | ADC2_D<0> | AD12 |
| ADC1_D<1> | AD19 | ADC2_D<1> | AD6 |
| ADC1_D<2> | AC22 | ADC2_D<2> | AD7 |
| ADC1_D<3> | AC19 | ADC2_D<3> | AC5 |
| ADC1_D<4> | AD15 | ADC2_D<4> | AC6 |
| ADC1_D<5> | AC15 | ADC2_D<5> | AC7 |
| ADC1_D<6> | AB15 | ADC2_D<6> | AB6 |
| ADC1_D<7> | AA15 | ADC2_D<7> | AB7 |
| ADC1_D<8> | AA13 | ADC2_D<8> | AB8 |
| ADC1_D<9> | Y14 | ADC2_D<9> | AA7 |
| ADC1_D<10> | Y15 | ADC2_D<10> | AA8 |
| ADC1_D<11> | W13 | ADC2_D<11> | Y8 |
| ADC1_D<12> | W14 | ADC2_D<12> | Y10 |
| ADC1_D<13> | W15 | ADC2_D<13> | W12 |
| ADC1_DRY | Y16 | ADC2_DRY | AA10 |
| ADC1_OVR | AA16 | ADC2_OVR | AB10 |

**Table 89: ADC Signal Pinouts**

### 11.7.7 User IO Header

| Signal Name | User FPGA (2V3000FG676) PIN No |
|---|---|
| User_IO_1 | AC10 |
| User_IO_2 | AD10 |

**Table 90: User IO Header Pinouts**

## 11.8 XC2V3000/XC2V6000 FF1152

### 11.8.1 User FPGA to DIME-II motherboard communication

#### Local Bus Pinouts

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| LBUS<0> | PB1 | V4 | LBUS<32> | PB37 | G1 |
| LBUS<1> | PB2 | U3 | LBUS<33> | PB38 | H2 |
| LBUS<2> | PB3 | T3 | LBUS<34> | PB39 | J2 |
| LBUS<3> | PB4 | P4 | LBUS<35> | PB40 | J1 |
| LBUS<4> | PB6 | R4 | LBUS<36> | PB42 | K2 |
| LBUS<5> | PB7 | M4 | LBUS<37> | PB43 | L2 |
| LBUS<6> | PB8 | N4 | LBUS<38> | PB44 | L1 |
| LBUS<7> | PB9 | K4 | LBUS<39> | PB45 | M2 |
| LBUS<8> | PB10 | L4 | LBUS<40> | PB46 | M1 |
| LBUS<9> | PB11 | L3 | LBUS<41> | PB47 | N3 |
| LBUS<10> | PB12 | M3 | LBUS<42> | PB48 | N2 |
| LBUS<11> | PB13 | H3 | LBUS<43> | PB49 | P3 |
| LBUS<12> | PB15 | J3 | LBUS<44> | PB51 | P2 |
| LBUS<13> | PB16 | H4 | LBUS<45> | PB52 | R3 |
| LBUS<14> | PB17 | J4 | LBUS<46> | PB53 | R1 |
| LBUS<15> | PB18 | F3 | LBUS<47> | PB54 | T2 |
| LBUS<16> | PB19 | G3 | LBUS<48> | PB55 | U2 |
| LBUS<17> | PB20 | E4 | LBUS<49> | PB56 | U1 |
| LBUS<18> | PB21 | F4 | LBUS<50> | PB57 | V2 |
| LBUS<19> | PB22 | D3 | LBUS<51> | PB58 | V1 |
| LBUS<20> | PB24 | E3 | LBUS<52> | PB60 | W3 |
| LBUS<21> | PB25 | AA4 | LBUS<53> | PB61 | Y3 |
| LBUS<22> | PB26 | AB4 | LBUS<54> | PB62 | AD3 |
| LBUS<23> | PB27 | AC3 | LBUS<55> | PB63 | AA2 |
| LBUS<24> | PB28 | AC4 | LBUS<56> | PB64 | AA1 |
| LBUS<25> | PB29 | D2 | LBUS<57> | PB65 | AB2 |
| LBUS<26> | PB30 | D1 | LBUS<58> | PB66 | AB1 |
| LBUS<27> | PB31 | E2 | LBUS<59> | PB67 | AC2 |
| LBUS<28> | PB33 | E1 | LBUS<60> | PB69 | AC1 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| LBUS<29> | PB34 | F2 | LBUS<61> | PB70 | AD2 |
| LBUS<30> | PB35 | F1 | LBUS<62> | PB71 | AD1 |
| LBUS<31> | PB36 | G2 | LBUS<63> | PB72 | AD4 |

**Table 91: Local Bus Pinouts (2V3000/6000)**

## Adjacent IN Bus – User FPGA communications

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| ADJIN<0> | PA29 | AK17 (GCLK0P) | ADJIN<32> | PA65 | W10 |
| ADJIN<1> | PA30 | AK16 (GCLK1S) | ADJIN<33> | PA66 | V10 |
| ADJIN<2> | PA31 | N10 | ADJIN<34> | PA67 | AA6 |
| ADJIN<3> | PA32 | M10 | ADJIN<35> | PA68 | Y6 |
| ADJIN<4> | PA33 | M6 | ADJIN<36> | PA69 | AA8 |
| ADJIN<5> | PA34 | L6 | ADJIN<37> | PA70 | Y7 |
| ADJIN<6> | PA35 | M7 | ADJIN<38> | PA71 | AA10 |
| ADJIN<7> | PA36 | N7 | ADJIN<39> | PA72 | Y10 |
| ADJIN<8> | PA38 | P8 | ADJIN<40> | SA2 | AB5 |
| ADJIN<9> | PA39 | N8 | ADJIN<41> | SB1 | AA5 |
| ADJIN<10> | PA40 | P6 | ADJIN<42> | SA3 | Y9 |
| ADJIN<11> | PA41 | N6 | ADJIN<43> | SB2 | AA9 |
| ADJIN<12> | PA42 | N5 | ADJIN<44> | SA4 | AC6 |
| ADJIN<13> | PA43 | P5 | ADJIN<45> | SB3 | AB6 |
| ADJIN<14> | PA44 | R10 | ADJIN<46> | SA5 | AC7 |
| ADJIN<15> | PA45 | P10 | ADJIN<47> | SB4 | AB7 |
| ADJIN<16> | PA47 | R9 | ADJIN<48> | SA6 | AB8 |
| ADJIN<17> | PA48 | P9 | ADJIN<49> | SB6 | AC8 |
| ADJIN<18> | PA49 | T8 | ADJIN<50> | SA7 | AC10 |
| ADJIN<19> | PA50 | R8 | ADJIN<51> | SB7 | AB10 |
| ADJIN<20> | PA51 | T7 | ADJIN<52> | SA8 | AE5 |
| ADJIN<21> | PA52 | U7 | ADJIN<53> | SB8 | AD5 |
| ADJIN<22> | PA53 | U6 | ADJIN<54> | SA9 | AF4 |
| ADJIN<23> | PA54 | T6 | ADJIN<55> | SB9 | AE4 |
| ADJIN<24> | PA56 | U8 | ADJIN<56> | SA11 | AC9 |
| ADJIN<25> | PA57 | U9 | ADJIN<57> | SB10 | AB9 |
| ADJIN<26> | PA58 | W6 | ADJIN<58> | SA12 | AF1 |
| ADJIN<27> | PA59 | V6 | ADJIN<59> | SB11 | AE2 |
| ADJIN<28> | PA60 | W5 | ADJIN<60> | SA13 | AE6 |
| ADJIN<29> | PA61 | V5 | ADJIN<61> | SB12 | AD6 |
| ADJIN<30> | PA62 | W7 | ADJIN<62> | SA14 | AE9 |
| ADJIN<31> | PA63 | V7 | ADJIN<63> | SB13 | AD9 |

**Table 92: Adjacent IN BUS Pinouts - User FPGA (2V3000/6000)**

## Adjacent OUT Bus - User FPGA connections

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| ADJOUT<0> | PD29 | N32 | ADJOUT<32> | PD65 | AK19 (GCLK6P) |
| ADJOUT<1> | PD30 | P32 | ADJOUT<33> | PD66 | AK18 (GCLK7S) |
| ADJOUT<2> | PD31 | T30 | ADJOUT<34> | PD67 | AD33 |
| ADJOUT<3> | PD32 | U30 | ADJOUT<35> | PD68 | AC33 |
| ADJOUT<4> | PD33 | U27 | ADJOUT<36> | PD69 | AF34 |
| ADJOUT<5> | PD34 | U26 | ADJOUT<37> | PD70 | AE33 |
| ADJOUT<6> | PD35 | U29 | ADJOUT<38> | PD71 | AF33 |
| ADJOUT<7> | PD36 | T29 | ADJOUT<39> | PD72 | AG33 |
| ADJOUT<8> | PD38 | U33 | ADJOUT<40> | SD2 | W28 |
| ADJOUT<9> | PD39 | U34 | ADJOUT<41> | SC1 | V28 |
| ADJOUT<10> | PD40 | T28 | ADJOUT<42> | SD3 | AB29 |
| ADJOUT<11> | PD41 | U28 | ADJOUT<43> | SC2 | AA29 |
| ADJOUT<12> | PD42 | U31 | ADJOUT<44> | SD4 | Y28 |
| ADJOUT<13> | PD43 | V31 | ADJOUT<45> | SC3 | Y29 |
| ADJOUT<14> | PD44 | V26 | ADJOUT<46> | SD5 | Y27 |
| ADJOUT<15> | PD45 | V27 | ADJOUT<47> | SC4 | W27 |
| ADJOUT<16> | PD47 | W30 | ADJOUT<48> | SD6 | AB27 |
| ADJOUT<17> | PD48 | V30 | ADJOUT<49> | SC6 | AA27 |
| ADJOUT<18> | PD49 | V32 | ADJOUT<50> | SD7 | AA26 |
| ADJOUT<19> | PD50 | W32 | ADJOUT<51> | SC7 | Y26 |
| ADJOUT<20> | PD51 | Y31 | ADJOUT<52> | SD8 | AC26 |
| ADJOUT<21> | PD52 | W31 | ADJOUT<53> | SC8 | AB26 |
| ADJOUT<22> | PD53 | V33 | ADJOUT<54> | SD9 | AE26 |
| ADJOUT<23> | PD54 | V34 | ADJOUT<55> | SC9 | AD26 |
| ADJOUT<24> | PD56 | AB33 | ADJOUT<56> | SD11 | AD27 |
| ADJOUT<25> | PD57 | AA33 | ADJOUT<57> | SC10 | AC27 |
| ADJOUT<26> | PD58 | AB31 | ADJOUT<58> | SD12 | AC28 |
| ADJOUT<27> | PD59 | AA31 | ADJOUT<59> | SC11 | AB28 |
| ADJOUT<28> | PD60 | AC31 | ADJOUT<60> | SD13 | AD29 |
| ADJOUT<29> | PD61 | AD31 | ADJOUT<61> | SC12 | AC29 |
| ADJOUT<30> | PD62 | AC32 | ADJOUT<62> | SD14 | AE30 |
| ADJOUT<31> | PD63 | AB32 | ADJOUT<63> | SC13 | AD30 |

**Table 93: Adjacent OUT BUS Pinouts - User FPGA (2V3000/6000)**

## PLINKS connected to the User FPGA: PLINKS 0, 1, 2 and 3

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| PP0LK<0> | PA2 | G5 | PP2LK<0> | PD2 | H32 |
| PP0LK<1> | PA3 | F5 | PP2LK<1> | PD3 | J32 |
| PP0LK<2> | PA4 | G6 | PP2LK<2> | PD4 | M29 |
| PP0LK<3> | PA5 | H6 | PP2LK<3> | PD5 | L29 |
| PP0LK<4> | PA6 | H5 | PP2LK<4> | PD6 | L33 |
| PP0LK<5> | PA7 | J5 | PP2LK<5> | PD7 | M33 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| PP0LK<6> | PA8 | J7 | PP2LK<6> | PD8 | K29 |
| PP0LK<7> | PA9 | H7 | PP2LK<7> | PD9 | L30 |
| PP0LK<8> | PA11 | K8 | PP2LK<8> | PD11 | L31 |
| PP0LK<9> | PA12 | J8 | PP2LK<9> | PD12 | M31 |
| PP0LK<10> | PA13 | K9 | PP2LK<10> | PD13 | P27 |
| PP0LK<11> | PA14 | J9 | PP2LK<11> | PD14 | N27 |
| | | | | | |
| PP1LK<0> | PA15 | K6 | PP3LK<0> | PD15 | R26 |
| PP1LK<1> | PA16 | J6 | PP3LK<1> | PD16 | P26 |
| PP1LK<2> | PA17 | K5 | PP3LK<2> | PD17 | R29 |
| PP1LK<3> | PA18 | L5 | PP3LK<3> | PD18 | R28 |
| PP1LK<4> | PA20 | K7 | PP3LK<4> | PD20 | P29 |
| PP1LK<5> | PA21 | L7 | PP3LK<5> | PD21 | N29 |
| PP1LK<6> | PA22 | M8 | PP3LK<6> | PD22 | L34 |
| PP1LK<7> | PA23 | L8 | PP3LK<7> | PD23 | M34 |
| PP1LK<8> | PA24 | L9 | PP3LK<8> | PD24 | N33 |
| PP1LK<9> | PA25 | L10 | PP3LK<9> | PD25 | P33 |
| PP1LK<10> | PA26 | N9 | PP3LK<10> | PD26 | N31 |
| PP1LK<11> | PA27 | M9 | PP3LK<11> | PD27 | P31 |

**Table 94: PLINK Pinouts - User FPGA (2V3000/6000)**

## PLINKS connected to the User FPGA: PLINKS 4, 5, 6 and 7

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| PP4LK<0> | SA48 | AG2 | PP6LK<0> | SD48 | AE28 |
| PP4LK<1> | SA49 | AF2 | PP6LK<1> | SD49 | AD28 |
| PP4LK<2> | SA50 | AG3 | PP6LK<2> | SD50 | AF27 |
| PP4LK<3> | SA51 | AF3 | PP6LK<3> | SD51 | AE27 |
| PP4LK<4> | SA52 | AE7 | PP6LK<4> | SD52 | AF29 |
| PP4LK<5> | SA53 | AD7 | PP6LK<5> | SD53 | AE29 |
| PP4LK<6> | SA54 | AG5 | PP6LK<6> | SD54 | AG30 |
| PP4LK<7> | SA56 | AF5 | PP6LK<7> | SD56 | AF30 |
| PP4LK<8> | SA57 | AD8 | PP6LK<8> | SD57 | AG28 |
| PP4LK<9> | SA58 | AE8 | PP6LK<9> | SD58 | AF28 |
| PP4LK<10> | SA59 | AH5 | PP6LK<10> | SD59 | AH29 |
| PP4LK<11> | SA60 | AG4 | PP6LK<11> | SD60 | AG29 |
| | | | | | |
| PP5LK<0> | SB47 | H16 (GCLK0S) | PP7LK<0> | SC47 | J18 (GCLK6S) |
| PP5LK<1> | SB48 | H17 (GCLK1P) | PP7LK<1> | SC48 | K18 (GCLK7P) |
| PP5LK<2> | SB49 | AL2 | PP7LK<2> | SC49 | AJ31 |
| PP5LK<3> | SB51 | AK2 | PP7LK<3> | SC51 | AK31 |
| PP5LK<4> | SB52 | AJ1 | PP7LK<4> | SC52 | AJ32 |
| PP5LK<5> | SB53 | AH1 | PP7LK<5> | SC53 | AH32 |
| PP5LK<6> | SB54 | AJ2 | PP7LK<6> | SC54 | AL32 |

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| PP5LK<7> | SB55 | AH2 | PP7LK<7> | SC55 | AK32 |
| PP5LK<8> | SB56 | AJ3 | PP7LK<8> | SC56 | AL33 |
| PP5LK<9> | SB57 | AH3 | PP7LK<9> | SC57 | AK33 |
| PP5LK<10> | SB58 | AK4 | PP7LK<10> | SC58 | AL34 |
| PP5LK<11> | SB60 | AJ4 | PP7LK<11> | SC60 | AK34 |

**Table 95: PLINK Pinouts - User FPGA (2V3000/6000)**

## General Purpose IO

| Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|---|---|
| GP_IO<0> | PC72 | M27 | GP_IO<11> | PC28 | L26 |
| GP_IO<1> | PC70 | L28 | GP_IO<12> | PC17 | H33 |
| GP_IO<2> | PC10 | G33 | GP_IO<13> | PC16 | G34 |
| GP_IO<3> | PC9 | F34 | GP_IO<14> | PC8 | F33 |
| GP_IO<4> | PC69 | K28 | GP_IO<15> | PC7 | E34 |
| GP_IO<5> | PC67 | K30 | GP_IO<16> | PC6 | E33 |
| GP_IO<6> | PC66 | K33 | GP_IO<17> | PC4 | D34 |
| GP_IO<7> | PC65 | J34 | GP_IO<18> | PC3 | D33 |
| GP_IO<8> | PC55 | G29 | GP_IO<19> | PC2 | K24 |
| GP_IO<9> | PC54 | N26 | GP_IO<20> | PC1 | L25 |
| GP_IO<10> | PC48 | N28 | | | |

**Table 96: GP_IO Pinouts - User FPGA (2V3000/6000)**

## 11.8.2 DIME-II control and monitoring signals

### DIME-II Specific Pins

| Dime-II Connector PIN No | Signal Name | Dime-II Connector PIN No | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| CONFIG_DONE | FPGA_DONE | PC40 | J31 |
| N/C | CONFIG_DONE* | N/a | F32 |
| CLK0 | CLKA | PC24 | E17 (GCLK3P) |
| CLK1 | CLKB | PC31 | AG17 (GCLK2P) |
| CLK2 | CLKC | PC42 | E19 (GCLK5P) |
| RESETI | RESETI | PC15 | AC25 |
| SLOT_ID0 | SLOT_ID0 | PC51 | R27 |
| SLOT_ID1 | SLOT_ID1 | PC52 | T27 |

**Table 97: User FPGA Specific Pinouts (2V3000/6000)**

*CONFIG_DONE is driven from the IO of the Virtex-II into the base of the transistor to signal that the on-board User FPGA has been configured successfully. User to drive this pin LOW once the FPGA is configured.

### User LEDs

| Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|
| LED_Green1 | AM8 |
| LED_Red1 | AN3 |
| | |
| LED_Green2 | AN32 |
| LED_Red2 | AJ26 |

**Table 98: User LED Pinouts - User FPGA (2V3000/6000)**

### On-board Temperature Sensor

| Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|
| DXN | F28 |
| DXP | G27 |
| ALERTI | AJ33 |

**Table 99: Temperature Sensor Pinouts (2V3000/6000)**

## 11.8.3 ZBT SRAM BANK A

### Clock and Control Signals for ZBT Bank A

| Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|
| ZBTA_CLK | C18 |
| ZBTA_FB_OUT | C19 |
| ZBTA_FB_IN | E18 (GCLK4S) |
| ZBTA_ADV | G18 |
| ZBTA_CKEI | F19 |

| Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|
| ZBTA_CSl<0> | D19 |
| ZBTA_CSl<1> | D18 |
| ZBTA_OEl | G19 |
| ZBTA_WEl | F18 |

**Table 100: ZBT Clock and Control Signals Pinouts – Bank A (2V3000/6000)**

## ZBT Address Signals – Bank A

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTA_A<0> | C33 | ZBTA_A<11> | K19 |
| ZBTA_A<1> | E22 | ZBTA_A<12> | E26 |
| ZBTA_A<2> | J20 | ZBTA_A<13> | H26 |
| ZBTA_A<3> | D29 | ZBTA_A<14> | G26 |
| ZBTA_A<4> | K21 | ZBTA_A<15> | E21 |
| ZBTA_A<5> | H21 | ZBTA_A<16> | F26 |
| ZBTA_A<6> | C29 | ZBTA_A<17> | H20 |
| ZBTA_A<7> | G23 | ZBTA_A<18>* | H22 |
| ZBTA_A<8> | H24 | ZBTA_A<19>* | H25 |
| ZBTA_A<9> | J23 | ZBTA_A<20>* | J24 |
| ZBTA_A<10> | G20 | ZBTA_A<21>* | F22 |

**Table 101: ZBT Address Signals Pinouts – Bank A (2V3000/6000)**

*These are Expansion PINs and may not be supported by the ZBT memory. Only use the above address pins if the onboard ZBT memory is in the following list:

- ZBTA_A<18> use if ZBT Memory is 16Mb
- ZBTA_A<19> use if ZBT Memory is 32Mb
- ZBTA_A<20> use if ZBT Memory is 64Mb
- ZBTA_A<20> use if ZBT Memory is 128Mb

## ZBT Data Signals – Bank A

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTA_D<0> | C24 | ZBTA_D<16> | A23 |
| ZBTA_D<1> | D25 | ZBTA_D<17> | A24 |
| ZBTA_D<2> | C26 | ZBTA_D<18> | A26 |
| ZBTA_D<3> | C27 | ZBTA_D<19> | A28 |
| ZBTA_D<4> | E25 | ZBTA_D<20> | B22 |
| ZBTA_D<5> | D26 | ZBTA_D<21> | B23 |
| ZBTA_D<6> | D27 | ZBTA_D<22> | B24 |
| ZBTA_D<7> | E27 | ZBTA_D<23> | B27 |
| ZBTA_D<8> | D20 | ZBTA_D<24> | B28 |
| ZBTA_D<9> | D21 | ZBTA_D<25> | B29 |
| ZBTA_D<10> | C22 | ZBTA_D<26> | B30 |
| ZBTA_D<11> | C23 | ZBTA_D<27> | B31 |

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTA_D<12> | F21 | ZBTA_D<28> | A29 |
| ZBTA_D<13> | D22 | ZBTA_D<29> | A30 |
| ZBTA_D<14> | D23 | ZBTA_D<30> | A31 |
| ZBTA_D<15> | D24 | ZBTA_D<31> | B32 |

**Table 102: ZBT Data Signals Pinouts – Bank A (2V3000/6000)**

### ZBT Parity Bits – Bank A

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTA_P<0> | E28 | ZBTA_P<2> | B21 |
| ZBTA_P<1> | F20 | ZBTA_P<3> | F25 |

**Table 103: ZBT Parity Bits Pinouts – Bank A (2V3000/6000)**

## 11.8.4 ZBT SRAM BANK B

### Clock and Control Signals for ZBT Bank B

| Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|
| ZBTB_CLK | D17 |
| ZBTB_FB_OUT | D16 |
| ZBTB_FB_IN | E16 (GCLK2S) |
| ZBTB_ADV | G17 |
| ZBTB_CKEl | F16 |
| ZBTB_CSl<0> | C15 |
| ZBTB_CSl<1> | C16 |
| ZBTB_Oel | G16 |
| ZBTB_Wel | F17 |

**Table 104: ZBT Clock and Control Signals Pinouts – Bank B (2V3000/6000)**

### ZBT Address Signals – Bank B

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTB_A<0> | F10 | ZBTB_A<11> | J11 |
| ZBTB_A<1> | J15 | ZBTB_A<12> | G11 |
| ZBTB_A<2> | K15 | ZBTB_A<13> | C2 |
| ZBTB_A<3> | F8 | ZBTB_A<14> | F9 |
| ZBTB_A<4> | F15 | ZBTB_A<15> | G10 |
| ZBTB_A<5> | C13 | ZBTB_A<16> | G9 |
| ZBTB_A<6> | H10 | ZBTB_A<17> | K16 |
| ZBTB_A<7> | F14 | ZBTB_A<18>* | H14 |
| ZBTB_A<8> | J12 | ZBTB_A<19>* | H9 |
| ZBTB_A<9> | J10 | ZBTB_A<20>* | H11 |
| ZBTB_A<10> | J14 | ZBTB_A<21>* | C14 |

Table 105: ZBT Address Signals Pinouts – Bank B (2V3000/6000)

*These are Expansion PINs and may not be supported by the ZBT memory. Only use the above address pins if the onboard ZBT memory is in the following list:

- ZBTB_A<18> use if ZBT Memory is 16Mb
- ZBTB_A<19> use if ZBT Memory is 32Mb
- ZBTB_A<20> use if ZBT Memory is 64Mb
- ZBTB_A<20> use if ZBT Memory is 128Mb

### ZBT Data Signals – Bank B

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTB_D<0> | B7 | ZBTB_D<16> | F13 |
| ZBTB_D<1> | B6 | ZBTB_D<17> | D13 |
| ZBTB_D<2> | B5 | ZBTB_D<18> | D12 |
| ZBTB_D<3> | B4 | ZBTB_D<19> | D11 |
| ZBTB_D<4> | A7 | ZBTB_D<20> | E13 |
| ZBTB_D<5> | A6 | ZBTB_D<21> | C12 |
| ZBTB_D<6> | A5 | ZBTB_D<22> | C11 |
| ZBTB_D<7> | A4 | ZBTB_D<23> | C9 |
| ZBTB_D<8> | A12 | ZBTB_D<24> | D10 |
| ZBTB_D<9> | A11 | ZBTB_D<25> | D9 |
| ZBTB_D<10> | B10 | ZBTB_D<26> | D8 |
| ZBTB_D<11> | B9 | ZBTB_D<27> | E8 |
| ZBTB_D<12> | B12 | ZBTB_D<28> | C8 |
| ZBTB_D<13> | B11 | ZBTB_D<29> | C7 |
| ZBTB_D<14> | A9 | ZBTB_D<30> | C6 |
| ZBTB_D<15> | B8 | ZBTB_D<31> | D6 |

Table 106: ZBT Data Signals Pinouts – Bank B (2V3000/6000)

### ZBT Parity Bits – Bank B

| Signal Name | User FPGA (2V3000/2V6000) PIN No | Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ZBTB_P<0> | B3 | ZBTB_P<2> | E14 |
| ZBTB_P<1> | B13 | ZBTB_P<3> | E7 |

Table 107: ZBT Parity Bits Pinouts – Bank B (2V3000/6000)

## 11.8.5 Clock signals relating to DACs and ADCs

### Clock sources available at CLK FPGA

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V3000/2V6000) PIN No | Signal Description |
|---|---|---|---|
| CLK_Op_Amp | B6 (GCLK6S) | N/a | External CLK source via Op_Amp |
| CLK_Op_Ampl | C6 (GCLK7P) | N/a | Complement of External |

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V3000/2V6000) PIN No | Signal Description |
|---|---|---|---|
| | | | CLK source via Op_Amp |
| EXT_CLK | B8 (GCLK0S) | N/a | External CLK source straight to CLK FPGA |
| EXT_CLKI | A8 (GCLK1P) | N/a | Complement of External CLK source straight to CLK FPGA |
| Osc_CLK | M6 (GCLK4P) | N/a | LVTTL Clock Oscillator |
| GEN_CLKA | K7 (GCLK0P) | AH18 | Generated Clock A |
| GEN_CLKC | N8 (GCLK1S) | AH19 | Generated Clock C |
| GEN_CLKB | M7 (GCLK6P) | AM20 | Generated Clock B |
| GEN_CLKD | N7 (GCLK7S) | AM19 | Generated Clock D |
| EXT2_CLK | D7 (GCLK5P) | N/a | 2nd External Clock |
| EXT2_CLKI | A6 (GCLK4S) | N/a | Complement of 2nd External Clock |

**Table 108: Clock Signals at CLK FPGA (2V3000/6000)**

### Clock Feedback signals

| Signal Name | CLK FPGA (2V80) Pin No | User FPGA (2V3000/2V6000) PIN No | Signal Description |
|---|---|---|---|
| CLK1_FB | J2 | AF18 | Feedback to User FPGA |
| CLK3_FB | H4 | AG18 | Feedback to User FPGA |
| CLK2_FB | H12 | AF17 | Feedback to User FPGA |

**Table 109: Clock Feedback Signals (2V3000/6000)**

### Clocking Pinouts for DACs and ADCs

| Signal Name | CLK FPGA (2V80) Pin No |
|---|---|
| ADC_CLKA | E4 |
| ADC_CLKAI | D1 |
| ADC_CLKB | G1 |
| ADC_CLKBI | F1 |
| DAC_CLKA | D13 |
| DAC_CLKAI | D12 |
| DAC_CLKB | G10 |
| DAC_CLKBI | F12 |

**Table 110: Clocking Pinouts for DACs and ADCs**

## 11.8.6 DAC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V3000/2V6000) PIN No | Signal Name (DAC 2) | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| DAC1_D<0> | AM26 | DAC2_D<0> | AL22 |
| DAC1_D<1> | AL26 | DAC2_D<1> | AJ22 |
| DAC1_D<2> | AK26 | DAC2_D<2> | AH22 |
| DAC1_D<3> | AH26 | DAC2_D<3> | AP23 |
| DAC1_D<4> | AN27 | DAC2_D<4> | AN23 |
| DAC1_D<5> | AM27 | DAC2_D<5> | AM23 |

| Signal Name (DAC 1) | User FPGA (2V3000/2V6000) PIN No | Signal Name (DAC 2) | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| DAC1_D<6> | AL27 | DAC2_D<6> | AN24 |
| DAC1_D<7> | AK27 | DAC2_D<7> | AM24 |
| DAC1_D<8> | AP28 | DAC2_D<8> | AL24 |
| DAC1_D<9> | AN28 | DAC2_D<9> | AL23 |
| DAC1_D<10> | AP29 | DAC2_D<10> | AJ23 |
| DAC1_D<11> | AN30 | DAC2_D<11> | AH23 |
| DAC1_D<12> | AP31 | DAC2_D<12> | AK24 |
| DAC1_D<13> | AN31 | DAC2_D<13> | AJ24 |
| DAC1_DIV0 | AJ25 | DAC2_DIV0 | AF21 |
| DAC1_DIV1 | AK25 | DAC2_DIV1 | AF20 |
| DAC1_MOD0 | AP26 | DAC2_MOD0 | AF23 |
| DAC1_MOD1 | AH25 | DAC2_MOD1 | AF22 |
| DAC1_PLLLOCK | AH24 | DAC2_PLLLOCK | AE20 |
| DAC1_RESET | AL25 | DAC2_RESET | AE21 |

**Table 111: DACs Signal Pinouts (2V3000/6000)**

## 11.8.7 ADC Signal Pinouts

| Signal Name (DAC 1) | User FPGA (2V3000/2V6000) PIN No | Signal Name (DAC 2) | User FPGA (2V3000/2V6000) PIN No |
|---|---|---|---|
| ADC1_D<0> | AJ12 | ADC2_D<0> | AM9 |
| ADC1_D<1> | AL11 | ADC2_D<1> | AK10 |
| ADC1_D<2> | AH12 | ADC2_D<2> | AH10 |
| ADC1_D<3> | AH13 | ADC2_D<3> | AH9 |
| ADC1_D<4> | AM11 | ADC2_D<4> | AJ9 |
| ADC1_D<5> | AN11 | ADC2_D<5> | AN4 |
| ADC1_D<6> | AJ11 | ADC2_D<6> | AP4 |
| ADC1_D<7> | AE13 | ADC2_D<7> | AN5 |
| ADC1_D<8> | AE12 | ADC2_D<8> | AP5 |
| ADC1_D<9> | AG11 | ADC2_D<9> | AN6 |
| ADC1_D<10> | AG12 | ADC2_D<10> | AP6 |
| ADC1_D<11> | AF12 | ADC2_D<11> | AN7 |
| ADC1_D<12> | AF13 | ADC2_D<12> | AL8 |
| ADC1_D<13> | AE14 | ADC2_D<13> | AN8 |
| ADC1_DRY | AE15 | ADC2_DRY | AL9 |
| ADC1_OVR | AL13 | ADC2_OVR | AP9 |

**Table 112: ADC Signal Pinouts (2V3000/6000)**

### 11.8.8 User IO Header

| Signal Name | User FPGA (2V3000/2V6000) PIN No |
|---|---|
| User_IO_1 | AJ5 |
| User_IO_2 | AH6 |

**Table 113: User IO Header Pinouts (2V3000/6000)**

## Standard Terms and Conditions

### GENERAL

These Terms and Conditions shall apply to all contracts for goods sold or work done by Nallatech Limited. (hereinafter referred to as the "company" or Nallatech) and purchased by any customer (hereinafter referred to as the customer).

Nallatech Limited trading in the style Nallatech (the company), submits all quotations and price lists and accepts all orders subject to the following conditions of contract which apply to all contracts for goods supplied or work done by them or their employees to the exclusion of all other representations, conditions or warranties, express or implied.

The buyer agrees to execute and return any license agreements as may be required by the company in order to authorise the use of those licensable items. If the licensable item is to be resold this condition shall be enforced by the re-seller on the end customer.

Each order received by the company will be deemed to form a separate contract to which these conditions apply and any waiver or any act of non-enforcement or variation of these terms or part thereof shall not bind or prejudice the company in relation to any other contract.

The company reserves the right to re-issue its price list at any time and to refuse to accept orders at a price other than at the price stated on the price list in force at the time of order.

The company reserves the right to vary the specification or withdraw from the offer any of its products without prior warning.

The company reserves the right to refuse to accept any contract that is deemed to be contrary to the company's policies in force at the time.

### PRICING

All prices shown on the company's price list, or on quotations offered by them, are based upon the acceptance of these conditions. Any variation of these conditions requested by the buyer could result in changes in the offered pricing or refusal to supply.

All quoted pricing is in Pounds Sterling and is exclusive of Value Added Tax (VAT) and delivery. In addition to the invoiced value the buyer is liable for all import duty as may be applicable in the buyer's location. If there is any documentation required for import formalities, whether or not for the purposes of duty assessment, the buyer shall make this clear at the time of order.

Quotations are made by Nallatech upon the customer's request but there is no obligation for either party until Nallatech accepts the customer's order.

Nallatech reserves the right to increase the price of goods agreed to be sold in proportion to any increase of costs to Nallatech between the date of acceptance of the order and the date of delivery or where the increase is due to any act or default of the customer, including the cancellation or rescheduling by the customer of part of any order.

Nallatech reserves the right (without prejudice to any other remedy) to cancel any uncompleted order or to suspend delivery in the event of any of the customer's commitment with Nallatech not being met.

### DELIVERY

All delivery times offered by the company are to be treated as best estimates and no penalty can be accepted for non-compliance with them.

Delivery shall be made by the company using a courier service of its choice. The cost of the delivery plus a nominal fee for administration will be added to the invoice issued. Payment of all inward customs duties and fees are the sole responsibility of the buyer. If multiple shipments are requested by the buyer, multiple delivery charges will be made. In the case of multiple deliveries separate invoices will be raised.

If requested at the time of ordering an alternative delivery service can be used, but only if account details are supplied to the company so that the delivery can be invoiced directly to the buyer by the delivery service.

The buyer accepts that any 'to be advised' scheduled orders not completed within twelve months from the date of acceptance of the original order, or orders held up by the buyers lack of action regarding delivery, can be shipped and invoiced by the company and paid in full by the buyer, immediately after completion of that twelve month period.

### INSURANCE

All shipments from the company are insured by them. If any goods received by the buyer are in an unsatisfactory condition, the following courses of action shall be taken.

If the outer packaging is visibly damaged, then the goods should not be accepted from the courier, or they should be signed for only after noting that the packaging has sustained damage.

If the goods are found to be damaged after unpacking, the company must be informed immediately.

Under no circumstances should the damaged goods be returned, unless expressly authorised by the company.

If the damage is not reported within 48 hours of receipt, the insurers of the company shall bear no liability.

Any returns made to the company for any reason, at any time shall be packaged in the original packaging, or its direct equivalent and must be adequately insured by the buyer.

Any equipment sent to the company for any purpose, including but not limited to equipment originally supplied by the company must be adequately insured by the buyer while on the premises of the company.

### PAYMENT

Nallatech Ltd. terms of payment are 30 days net.

Any charges incurred in making the payment, either currency conversion or otherwise shall be paid by the buyer.

The company reserves the right to charge interest at a rate of 2% above the base rate of the Bank of Scotland PLC on any overdue accounts. The interest will be charged on any outstanding amount from said due date of payment, until payment is made in full, such interest will accrue on a daily basis.

### TECHNICAL SUPPORT

The company offers a dedicated technical support via telephone and an email address. It will also accept faxed support queries.

Technical support will be given free of charge for 90 days from the date of invoice, for queries regarding the use of the products in the system configuration for which they were sold. Features not documented in the user manual or a written offer of the company will not be supported. Interfacing with other products other than those that are pre-approved by the company as compatible will not be supported. If the development tools and system hardware is demonstrably working, no support can be given with application level problems.

### WARRANTY

The company offers as part of a purchase contract 12 months warranty against parts and defective workmanship of hardware elements of a system. The basis of this warranty is that the fault be discussed with the companies technical support staff before any return is made. If it is agreed that a return for repair is necessary then the faulty item and any other component of the system as requested by those staff shall be returned carriage paid to the company. Insurance terms as discussed in the INSURANCE Section will apply.

The company will not accept returned goods unless this has been expressly authorised.

After warranty repair, goods will be returned to the buyer carriage paid by the company using their preferred method.

Faults incurred by abuse of the product (as defined by the company) are not covered by the warranty.

Attempted repair or alteration of the goods as supplied by the company, by another party immediately invalidates the warranty offered.

The said warranty is contingent upon the proper use of the goods by the customer and does not cover any part of the goods which has been modified without Nallatech's prior written consent or which has been subjected to unusual physical or electrical stress or on which the original identification marks have been removed or altered. Nor will such warranty apply if repair or parts required as a result of causes other than ordinary authorised use including without limitation accident, air conditioning, humidity control or other environmental conditions.

Under no circumstances will the company be liable for any incidental or consequential damage or expense of any kind, including, but not limited to, personal injuries and loss of profits arising in connection with any contract or with the use, abuse, unsafe use or inability to use the companies goods. The company's maximum liability shall not exceed and the customer's remedy is limited to, either:

(i) repair or replacement of the defective part or product or at the companies option;

(ii) return of the product and refund of the purchase price and such remedy shall be the customer's entire and exclusive remedy.

Warranty of the software written by the company shall be limited to 90 days warranty that the media is free from defects and no warranty express or implied is given that the computer software will be free from error or will meet the specification requirements of the buyer.

The terms of any warranty offered by a third party whose software is supplied by the company will be honoured by the company exactly. No other warranty is offered by the company on these products.

Return of faulty equipment after the warranty period has expired, the company may at its discretion make a quotation for repair of the equipment or declare that the equipment is beyond repair.

**PASSING OF RISK AND TITLE**

The passing of risk for any supply made by the company shall occur at the time of delivery. The title however shall not pass to the buyer until payment has been received in full by the company. And no other sums whatever shall be due from the customer to Nallatech.

If the customer (who shall in such case act on his own account and not as agent for Nallatech) shall sell the goods prior to making payment in full for them, the beneficial entitlement of Nallatech therein shall attach to the proceeds of such sale or to the claim for such proceeds.

The customer shall store any goods owned by Nallatech in such a way that they are clearly identifiable as Nallatech's property and shall maintain records of them identifying them as Nallatech's property. The customer will allow Nallatech to inspect these records and the goods themselves upon request.

In the event of failure by the customer to pay any part of the price of the goods, in addition to any other remedies available to Nallatech under these terms and conditions or otherwise, Nallatech shall be entitled to repossess the goods. The customer will assist and allow Nallatech to repossess the goods as aforesaid and for this purpose admit or procure the admission of Nallatech or its employees and agents to the premises in which the goods are situated.

**INTELLECTUAL PROPERTY**

The buyer agrees to preserve the Intellectual Property Rights (IPR) of the company at all times and that no contract for supply of goods involves loss of IPR by the company unless expressly offered as part of the contract by the company.

**GOVERNING LAW**

This agreement and performance of both parties shall be governed by Scottish law.

Any disputes under any contract entered into by the company shall be settled in a court if the company's choice operating under Scottish law and the buyer agrees to attend any such proceedings. No action can be brought arising out of any contract more than 12 months after the completion of the contract.

**INDEMNITY**

The buyer shall indemnify the company against all claims made against the company by a third party in respect of the goods supplied by the company.

**SEVERABILITY**

If any part of these terms and conditions is found to be illegal, void or unenforceable for any reason, then such clause or Section shall be severable from the remaining clauses and Sections of these terms and conditions which shall remain in force.

**NOTICES**

Any notice to be given hereunder shall be in writing and shall be deemed to have been duly given if sent or delivered to the party concerned at its address specified on the invoice or such other addresses as that party may from time to time notify in writing and shall be deemed to have been served, if sent by post, 48 hours after posting.

## Software Licensing Agreement

Nallatech Ltd software is licensed for use by end users under the following conditions. By installing the software you agree to be bound by the terms of this license. If you do not agree with the terms of this license, do not install the Software and promptly return it to the place where you obtained it.:

**1. Licence:** Nallatech Ltd grants you a licence to use the software programs and documentation in this package("Licensed materials") if you have a single license, on only one computer at a time *or* by only one user at a time;
if you have acquired multiple licenses, the Software may be used on either stand alone computers or on computer networks, by a number of simultaneous users equal to or less than the number of licenses that you have acquired; and, if you maintain the confidentiality of the Software and documentation at all times.

**2. Restrictions:** This software contains trade secrets in its human perceivable form and, to protect them, except as permitted by applicable law, you may not reverse engineer, disassemble or otherwise reduce the software to any human perceivable form. You may not modify, translate, rent, lease, loan or create derivative works based upon the software or part thereof without a specific run-time licence from Nallatech Ltd.

**3. Copyright:** The Licensed Materials are Copyrighted. Accordingly, you may either make one copy of the Licensed Materials for backup and/or archival purposes or copy the Licensed Materials to another medium and keep the original Licensed Materials for backup and/or archival purposes. Additionally, if the package contains multiple versions of the Licensed Materials, then you may only use the Licensed Materials in one version on a single computer. In no event may you use two copies of the Licensed Materials at the same time.

**4. Warranty:** Nallatech Ltd warrants the media to be free from defects in material and workmanship and that the software will substantially conform to the related documentation for a period of ninety (90) days after the date of your purchase. Nallatech Ltd does not warrant that the Licensed Materials will be free from error or will meet your specific requirements.

**5. Limitations:** Nallatech Ltd makes no warranty or condition, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding the Licensed Materials.

Neither Nallatech Ltd nor any applicable Licenser will be liable for any incidental or consequential damages, including but not limited to lost profits.

**6. Export Control:** The Software is subject to the export control laws of the United States and of the United Kingdom. The Software may not be shipped, transferred, or re-exported directly or indirectly into any country prohibited by the United States Export Administration Act 1969 as amended, and the regulations there under, or be used for any purpose prohibited by the Act.

## User Guide Conditions

Information in this User Guide is subject to change without notice. Any changes will be included in future versions of this document. Information within this manual may include technical, typing or printing inaccuracies or errors and no liability will arise there from.

This User Guide is supplied without warranty or condition, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding the information provided herein.

Under no circumstances will Nallatech Limited be liable for any incidental or consequential damage or expense of any kind, including, but not limited to, loss of profits, arising in connection with the use of the information provided herein.

# Index

## X



## Z

# Remarks form

We welcome any comments you may have on our product and its documentation. Your remarks will be examined thoroughly and taken into account for future versions of this product.

| | |
|---|---|
| **XtremeDSP Development Kit User Guide<br>NT107-0132 9** | **21/05/2003** |

**Errors detected:**

**Suggested Improvements:**

Please send this completed form to:

Nallatech
Boolean House
One Napier Park
Cumbernauld
Glasgow G68 0BH
United Kingdom

If you prefer you may send your remarks via E-mail to support@nallatech.com or by fax to +44 (0) 1236 789599.

If you want Nallatech to reply to your comments, please include your name, address and telephone number.