BEE2 System Platform

Hardware Reference Manual

UG001 (v1.0) January 7, 2006

Contents

Chapter 1

BEE2 Hardware Development System

Features

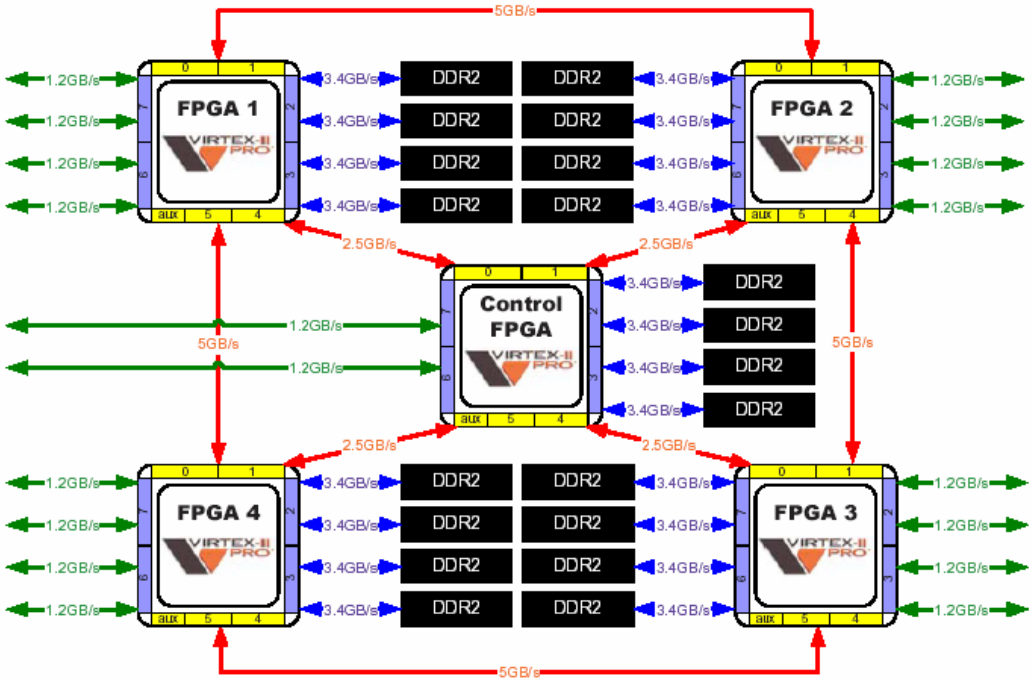
The BEE2 System Platform provides an advanced development system suitable for high-end reconfigurable computing, task-specific acceleration, and ASIC emulation and simulation. The fundamental unit is a module, which is a single printed circuit board containing five large field programmable gate arrays (FPGAs) arranged within a very high-speed inter-chip and inter-board communication structure. Additional peripherals are present to enhance usability and permit a fully functional free-standing computing system. Each FPGA has direct access to four DDR2 memory modules for program and intermediate result storage.

- 5 Xilinx Virtex-II Pro 70 FPGAs, Speed Grade 6
- Up to 40 GB DDR2 SDRAM
- SystemACE controller and Type II CompactFlash™ for configuration and data storage
- USB port
- DVI port
- SelectMAP configuration from control FPGA
- 10/100 Ethernet PHY
- RS-232 serial port (control FPGA)
- Status LEDs
- Serial ATA ports
- 18 AUI links (8 or 10 Gbps)
- Voltage and temperature monitoring
- Real Time Clock
- On-board power supplies
- Panel display
- Internal or external system clock and user programmable clock

General Description

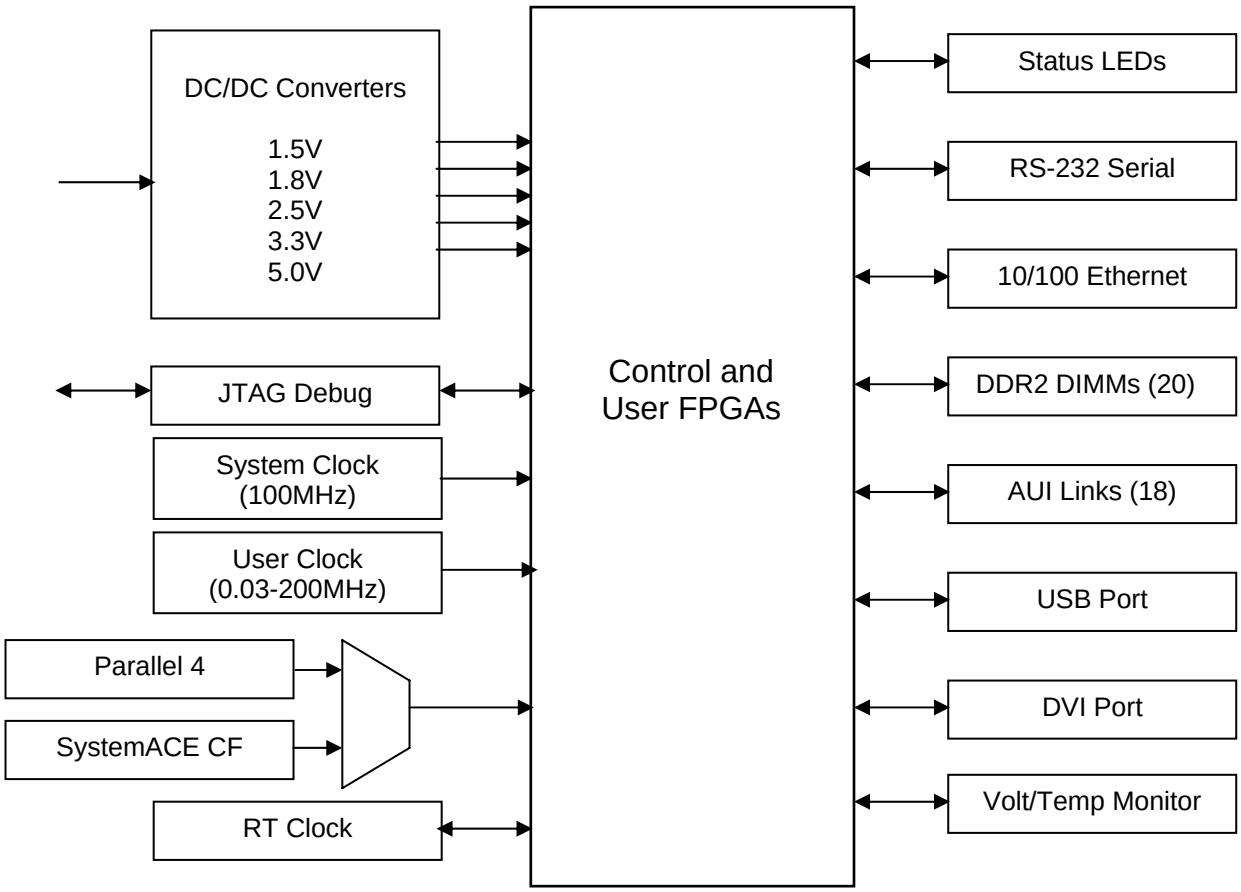
Block Diagram

The module general architecture is shown on Figure 1. The basic compute elements on the module are five Xilinx Virtex 2 Pro 70K FPGAs. The center FPGA, termed the *control FPGA*, has primary responsibility for system management. Each of the remaining four *user FPGAs* are tasked with computional workloads and can communicate with the control FPGA at a rate of 2.5GB/s. The user FPGAs can also inter-communicate at a speed of 5GB/s via a ring configuration. These Intra-module links are implemented using parallel buses utilizing general purpose I/O signals of the Xilinx FPGA's high-speed serial links. The selected product family has an additional 20 high speed serial RocketIO blocks implemented directly on chip which can be used of up to 3.125Gb/s, and in this instance, specificed to meet the IB4X description common to both by the Infiniband and the 10Gb-Ethernet-CX4 standards. It groups four 2.5Gb/s (Infiniband) or 3.125Gb/s (10Gb-CX4) differential pairs in a single unit to reach an effective maximum bit-rate of either 8Gb/s or 10Gb/s per port. Each user FPGA has four IB4X ports, whereas the control FPGA has only two. Each individual port requires four RocketIOs, which leaves four transceivers unused on the user FPGAs, and twelve on the control FPGA.



Compute elements block diagram 1

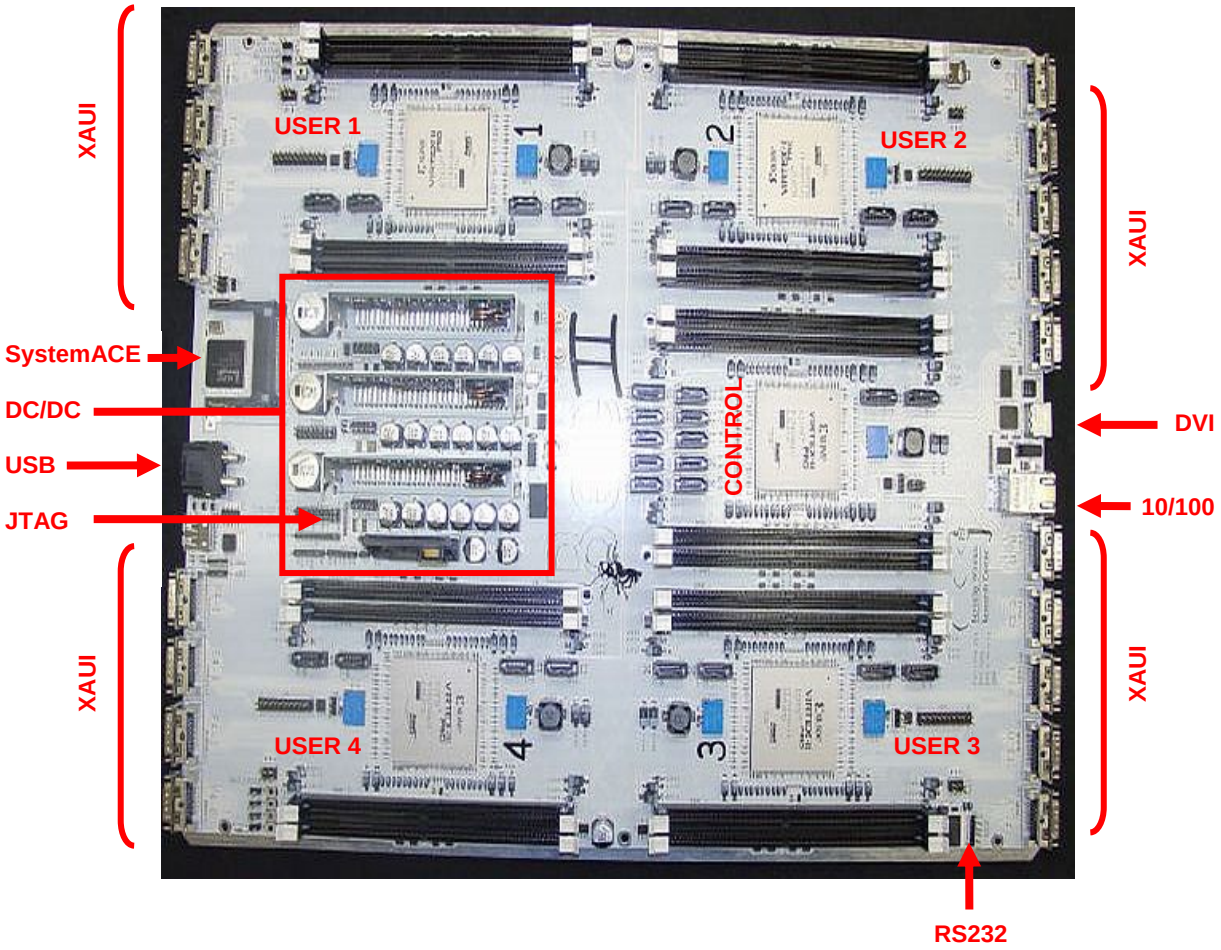
Up to four DDR2 DIMMs can be attached to each FPGA. Individual 72-bit wide DRAM modules can be accessed at speeds of up to 400Mb/s/wire, corresponding to an aggregate throughput of 3.4GB/s per DIMM.



System block diagram 2

Module Subsystems

This section contains an overview of each subsystem of a BEE2 module. Additional information is available from individual data sheets, in the case of an external component supplier, or from reference designs and documentation in the Appendix or website.


XAUI
USER 1
USER 2
XAUI
SystemACE
DC/DC
USB
JTAG
CONTROL
DVI
10/100
XAUI
USER 4
USER 3
XAUI
RS232

Power

Power is supplied to the module from the main system a 5V DC. Each module further reduces, regulates, and distributes the voltage using on-board regulators. From the supplied voltage, the module derives 1.5V, 1.8V, 2.5V, 3.3V and 5.0V via independent regulators.

Clocks

On-board are oscillators to produce stable clocks for XAUI (2) system, user…

User FPGAs

User FPGAS…

Control FPGA

Control FGPA…

Configuration

Configuration of the module is accomplished in one of two ways: a basic pre-built configuration stored in the CompactFlash attached to the SystemACE chip…

Also, serial downloading via JTAG

SelectMAP for individual chips under control of central control FPGA, and from OS

Temperature Monitor

Temperature monitoring from sensors incorporated into each chip as well as a separate sensor withing the ADC…

Serial Port

A control serial port supporting RS232…

Real Time Clock

Provisions for setting an maintaining precise RTC system time (specs and chip), trimming, battery backup and retention (time)

Ethernet

An Ethernet PHY 10/100 (revisit XUP and 310)

DDR2 Memory

Single channel controller, bus attached or not, DDR1, approved DIMMs

Multi-Gigabit Transceivers (Control FPGA)

Extra MGT channels in addition to …

Multi-Gigabit Transceivers (Userl FPGA)

AUI links (XAUI), bandwidth, etc.

LVCMOS Links

Interchip communication, bandwidth

Display Panel

Bus (poss change)

Voltage Monitor

Various.

USB

USB 2.0 or not? chip

DVI

What particular type, chip

Chapter 2

Using the System

Linux

Distributions

Distributions, default shipped, compiling, drivers, etc.

Shipped

On CF

Compiling

Appendix material

Accessing User FPGAs

Communication example

BEE2 Functional Test

1. Prepare Board
   a. Insert 20 DIMMs
   b. Insert compact flash labeled *BEE2 Test Suite*
   c. Connect RS232 to serial port on beehive and to board at J44 observing correct polarity
   <Optional: Connect JTAG to Parallel 4 cable>
   d. Connect J76 DVI to LCD panel monitor
   <Optional: If not present, connect five FPGA fans and locate above heatsinks (loosely), noting the black ground signal connects to pin 1 of J32-36.
   e. Orient external cooler to provide airflow to the DC/DC converters and power the unit on
   f. Initiate terminal session on a host computer (usually *beehive*) for port COM1 or COM2 (115200, 8, N, 1, N)
   g. Turn on Lambda lab power supply, disable output (select OUT to be *off*) confirm voltage setting of 5V, and connect power cable to board at J46 (Omit step for frame-type supply). *Note that the board requires a minmum of 7A for startup, and a maximum of 60A during full testing.*

2. Initiate Testing
    a. Apply power to board by selecting OUT to be *on* at the Lambda; confirm current **~7.0A** and verify five green LEDs (located near the compact flash), one for each 1.5V, 1.8V, 2.5V, 3.3V and 5.0V. Visually confirm all five FPGA fans are operating.
    b. Observe TinySH prompt in Hyperterm: **BEE2 %**
    c. Select DVI input on LCD panel by pressing switch for 2 seconds (beeps twice) and confirm blue background with above prompt
    d. Configure FPGAs by typing **`conf`** on the command line
    e. Confirm current increases by increments of about 4 A every 12 sec until total supply output reaches **~26.1A** and stabilizes. The monitor will display:

```
BEE2 % conf
Opening bitstream
January 7, 2006Starting configuration for FPGA1
Closing bitstream
Opening bitstream
Starting configuration for FPGA2
Closing bitstream
Opening bitstream
Starting configuration for FPGA3
Closing bitstream
Opening bitstream
Starting configuration for FPGA4
Closing bitstream
```

Green configuration done LEDs (num) at each FPGA will be illuminated. At this point, all FPGAs are configured and operational, including basic DDR refresh, although the data is random and un-initialized.

3. Perform Basic Suite

*Note: Available commands can be viewed by typing **`help`** on the command line; a more complete listing can be had by typing **`?`**. For multipart commands, <tab> will list further options, e.g. **`ddrreset <tab>`***

    a. Check for presence of all serial devices by typing **`listiic`** and confirm eight devices in chain:

```
BEE2 % listiic
Voltage monitor                                    : found
at address 0x28
Temperature sensor/fan controller (Control FPGA): found at
address 0x30
```

```
Temperature sensor/fan controller (FPGA 1)      : found at
address 0x32
Temperature sensor/fan controller (FPGA 2)      : found at
address 0x34
Temperature sensor/fan controller (FPGA 3)      : found at
address 0x52
Temperature sensor/fan controller (FPGA 4)      : found at
address 0x54
DVI interface                                   : found at
address 0x70
Real Time Clock (Clock control)                 : found at
address 0xAE
Real Time Clock (EEPROM control)                : found at
address 0xDE
```

b. Test the real-time clock by typing **`testrtc`** (note that no battery is present, therefore time-of-day is not retained on power down)

```
BEE2 % testrtc
Status register : 0x01
Device operating from VCC
Alarm 1 time is not reached
Alarm 0 time is not reached
RWEL (Register Write Enable Latch) : 0
WEL (Write Enable Latch) : 0
A power failure happened
RTC date is (MM/DD/YYYY)  25/45/19165
RTC time is (HH:MM:SS) 45:85:85
Control registers :
DTR (Digital Trimming Register) is: (DTR2 DTR1 DTR0) 000
ATR (Analog Trimming Register) is: 0
Pulse interrupt bit is: 0
Alarm 1 enable bit is: 0
Alarm 0 enable bit is: 0
Frequency output bits are (FO1 FO0): 00
EEPROM byte protection bits are (BP2 BP1 BP0): 000
```

c. Test the network connection by typing **`testmac`** (note that when network is unconnected, no packet traffic is expected; a preferred test is booting Linux with a nfs-mounted root file system.)

```
BEE2 % testmac
MAC Device initialization...OK
MAC Device self test...OK
Sending a test packet to the network (BROADCAST recipient)
Waiting for a packet...
No packet received
```

d. Test the board clock oscillators by typing **`testclks`**, resulting in correct measured values of 100, 120, 25, 25, and 32 MHz.

```
BEE2 % testclks
Set up the RTC to get a 1Hz output on the PHZ output
Unlock the counters...done
Wait for the counters to be locked...done
System Clock has a frequency of 100006109 Hz
User Clock has a frequency of 119997841 Hz
Ethernet Receive Clock has a frequency of 2500103 Hz
Ethernet Transmit Clock has a frequency of 2500201 Hz
SystemACE Clock has a frequency of 32002427 Hz
Re-unlock the counters...done
```

e. Test the DVI output by typing **`testdvi`**, which will cycle through the four available resolutions in about 20 seconds. Note the monitor will correctly display only 1280x1024, but the Hyperterm output will report test parameters:

```
BEE2 % testdvi
DVI chip vendor ID   : 0x014C
DVI chip device ID   : 0x0410
DVI chip revision ID : 0x00
Device Address       : 0x70
A screen is plugged, and it's turned on

Testing modes: 640x480, 800x600, 1024x768, 1280x1024
```

f. View the current board and FPGA temperatures with **`gettemp`**, which acquires data from each FPGA and the board ADC U15

```
BEE2 % gettemp
Temperature on board......28 celsius
Temperature on CtrlFPGA...33 celsius
Temperature on FPGA1......34 celsius
Temperature on FPGA2......36 celsius
Temperature on FPGA3......35 celsius
Temperature on FPGA4......35 celsius
```

g. Test the systemace chip by reading the internal ID directly from the device

(need data)

The next group of tests exercises multiple devices and substantially increases the current consumption. The current noted is for the listed sequence only; testing in a different order may result in slightly different values. As a general rule, in order to ensure a know state, one typically starts the device, then cycles the device off then on again prior to performing the tests. The test procedures are usually initiated

*with <**device**>**reset** <**param1**> <**param2**>. Also, tests run continuously until explicitly ended, but the displayed data is only re-sampled with <**device**>**refresh**.*

h. Test the AUI links with this sequence **`xauistart`** (initializes the process)**, `xauireset <param1> <param2>`** (begins the testing)**, `xauirefresh`** (updates the display with current results)**, `xauistop`** (ends the process).

(need data)

i. Test the DDR memory with the following sequence **`ddrstart`** to initialize the process, **`ddroff, ddron`** to restore to a known state, then **`ddrreset <rand|altern|sburst|lburst|xtalk|const|stop> [<fpga>][<module>], ddrrefresh, ddrstop.`** Note that the current decreases as the refresh is halted by ddroff, then resumes as non-random data is applied.

```
BEE2 % ddrstart
BEE2 % ddroff                    ~21A
BEE2 % ddron                     ~32A
BEE2 % ddrreset altern           ~50A
(supplying no fpga/module param tests all)
BEE2 % ddrstop                   ~21A
```

(ddr still running?)

j. Test the inter-chip communication with the sequence **`icstart, icoff, icon, icreset [<fpga>] [<module>], icrefresh, icstop.`**

```
BEE2 % icstart
BEE2 % icoff                     ~xxA
BEE2 % icon                      ~xxA
BEE2 % icreset                   ~xxA
BEE2 % icrefresh                 ~xxA
BEE2 % icstop                    ~xxA
```

k. End all testing and un-configure the user FGPAs with the **`unconf`** command (**~21A**).