SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

FP7-ICT-2011-7

DEEP

Dynamical Exascale Entry Platform

Grant Agreement Number: 287530

D3.5
DEEP Booster

Approved

Version: 2.0
Author(s): P. Arts (Eurotech)
Date: 16.01.2015
Project and Deliverable Information Sheet

<table>
<thead>
<tr>
<th>DEEP Project</th>
<th>Project Ref. №:</th>
<th>287530</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Title:</td>
<td>Dynamical Exascale Entry Platform</td>
<td></td>
</tr>
<tr>
<td>Project Web Site:</td>
<td><a href="http://www.deep-project.eu">http://www.deep-project.eu</a></td>
<td></td>
</tr>
<tr>
<td>Deliverable ID:</td>
<td>D3.5</td>
<td></td>
</tr>
<tr>
<td>Deliverable Nature:</td>
<td>Report</td>
<td></td>
</tr>
<tr>
<td>Deliverable Level:</td>
<td>PU*</td>
<td></td>
</tr>
<tr>
<td>Contractual Date of Delivery:</td>
<td>30 / November / 2014</td>
<td></td>
</tr>
<tr>
<td>Actual Date of Delivery:</td>
<td>5 / December / 2014</td>
<td></td>
</tr>
</tbody>
</table>

**EC Project Officer:** Luis Carlos Busquets Pérez

* - The dissemination level are indicated as follows: **PU** – Public, **PP** – Restricted to other participants (including the Commission Services), **RE** – Restricted to a group specified by the consortium (including the Commission Services). **CO** – Confidential, only for members of the consortium (including the Commission Services).

Document Control Sheet

<table>
<thead>
<tr>
<th>Document</th>
<th>Title:</th>
<th>DEEP Booster</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID:</td>
<td>D3.5</td>
<td></td>
</tr>
<tr>
<td>Version:</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>Status:</td>
<td>Approved</td>
<td></td>
</tr>
<tr>
<td>Available at:</td>
<td><a href="http://www.deep-project.eu">http://www.deep-project.eu</a></td>
<td></td>
</tr>
<tr>
<td>Software Tool:</td>
<td>Microsoft Word</td>
<td></td>
</tr>
<tr>
<td>File(s):</td>
<td>DEEP_D3.5_Booster_v2.0_ECapproved</td>
<td></td>
</tr>
</tbody>
</table>

**Authorship**

| Written by: | P. Arts (Eurotech) |
| Reviewed by: | Th. Moschny (ParTec), N. Eicker (JUELICH) |
| Approved by: | PMT/BoP |

Document Status Sheet

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>05/December/2014</td>
<td>Final</td>
<td>EC Submission</td>
</tr>
<tr>
<td>2.0</td>
<td>16/February/2015</td>
<td>Final</td>
<td>EC approved</td>
</tr>
</tbody>
</table>
## Document Keywords

| Keywords                  | DEEP, HPC, Exascale, Booster, Booster Node Card, Booster Interface Card, Backplane |

---

## Copyright notices

© 2011-2015 DEEP Consortium Partners. All rights reserved. This document is a project document of the DEEP project. All contents are reserved by default and may not be disclosed to third parties without the written consent of the DEEP partners, except as mandated by the European Commission contract 287530 for reviewing and dissemination purposes. All trademarks and other rights on third party products mentioned in this document are acknowledged as own by the respective holders.
Table of Contents

Project and Deliverable Information Sheet ................................................................. i
Document Control Sheet ............................................................................................... i
Document Status Sheet ............................................................................................... i
Document Keywords .................................................................................................... ii
Table of Contents .......................................................................................................... iii
List of Figures .............................................................................................................. v
List of Tables ................................................................................................................ v

Executive Summary .................................................................................................... 6

1 Introduction .............................................................................................................. 6

2 BNC Design and Implementation ........................................................................... 8
   2.1 BNC Architecture .............................................................................................. 8
   2.2 BNC Electric Design ......................................................................................... 10
   2.3 BNC Signal Integrity Testing ............................................................................ 13
   2.4 BNC Firmware Development ............................................................................ 15
      2.4.1 Atmel AVR μ-controller ............................................................................ 16
      2.4.2 CPLD ......................................................................................................... 16
      2.4.3 BMC ......................................................................................................... 17
   2.5 BNC Assembly .................................................................................................... 20

3 BIC Design and Implementation ........................................................................... 21
   3.1 BIC Architecture .............................................................................................. 21
   3.2 BIC Electric Design ......................................................................................... 22
      3.2.1 BIC Ethernet switch ................................................................................ 25
      3.2.2 BIC UART Mux ....................................................................................... 25
      3.2.3 BIC ARM7 μ-controller ......................................................................... 26
   3.3 BIC Signal Integrity Testing ............................................................................ 26
   3.4 BIC Firmware Development ............................................................................ 27
      3.4.1 Atmel ARM7 μ-controller ...................................................................... 27
   3.5 BIC Assembly .................................................................................................... 28

4 Pseudo-BIC .............................................................................................................. 29
   4.1 Pseudo-BIC design ........................................................................................... 29
      4.1.1 HDI adaptor card ..................................................................................... 30
      4.1.2 Galibier card ........................................................................................... 30
      4.1.3 Server ....................................................................................................... 32
   4.2 Pseudo-BIC tests .............................................................................................. 33
   4.3 Pseudo-BIC installation ................................................................................... 33

5 Backplane Design and Implementation ................................................................. 34
   5.1 Backplane Architecture ................................................................................... 34
   5.2 Backplane Electric Design .............................................................................. 35
   5.3 Backplane Signal integrity Testing .................................................................. 36
   5.4 Backplane Assembly ....................................................................................... 37

6 EXTOLL Torus Cables .............................................................................................. 39
   6.1 EXTOLL Torus Cables Design .......................................................................... 39
   6.2 EXTOLL Torus Cables Signal Integrity Testing ................................................ 41
7 Integration of the Booster Rack ........................................................................................................ 42
  7.1 Mounting Boards in the Chassis ................................................................................................. 42
  7.2 Hardware and Bring-Up Status at Juelich ................................................................................... 42
References and Applicable Documents ............................................................................................. 44
List of Acronyms and Abbreviations .................................................................................................. 45
List of Figures

Figure 1: Diagram of the DEEP architecture ................................................................. 7
Figure 2: BNC block diagram ...................................................................................... 8
Figure 3: BNC sensors and control block diagram ...................................................... 9
Figure 4: BNC management network ....................................................................... 10
Figure 5: BNC PCB design ......................................................................................... 12
Figure 6: BNC SI test setup for X direction measurements ....................................... 13
Figure 7: BNC SI test setup for Y direction measurements ....................................... 13
Figure 8: BNC SI test setup for Z direction measurements ....................................... 14
Figure 9: BNC SI measurement example of Eye diagram ........................................... 14
Figure 10: BNC and BIC SI measurement results matrix ............................................ 14
Figure 11: BNC Xeon Phi control diagram ................................................................ 15
Figure 12: BNC power state diagram ....................................................................... 17
Figure 13: BNC board management control functional block diagram .................... 18
Figure 14: BNC Riser card A, Riser card B and front panel & cable ......................... 20
Figure 15: BNC with 2 Xeon Phi cards mounted ....................................................... 20
Figure 16: BNC assembled with coldplate and front panel ....................................... 21
Figure 17: BIC block diagram ................................................................................. 22
Figure 18: BIC dimensions ....................................................................................... 23
Figure 19: BIC PCB design ...................................................................................... 24
Figure 20: BIC Ethernet switch block diagram .......................................................... 25
Figure 21: BIC serial console block diagram ............................................................. 25
Figure 22: BIC ARM7 block diagram ..................................................................... 26
Figure 23: BIC SI measurement of EXTOLL 7th link ................................................ 27
Figure 24: BIC card with SSD, InfiniBand and COM Express, without coldplate .... 28
Figure 25: BIC with coldplate ................................................................................. 28
Figure 26: Pseudo-BIC and its components connected to a Booster Chassis .......... 29
Figure 27: HDI6 connector card and its schematics .................................................... 30
Figure 28: HDI6 adaptor card plugged into the BIC slot of the half-Backplane ........ 30
Figure 29: EXTOLL Galibier card ............................................................................ 31
Figure 30: Servers for the Pseudo-BICs installed on top of rack, with one Galibier card each ................................................................. 32
Figure 31: EXTOLL network connections on the backplane .................................... 34
Figure 32: Backplane DIP switches to set the BIC position address ......................... 34
Figure 33: Backplane PCB design ........................................................................... 36
Figure 34: Testing X direction over the Backplane between slots 3 and 4 using Molex cables ................................................................. 36
Figure 35: Backplane top view .............................................................................. 37
Figure 36: Backplane bottom view ....................................................................... 37
Figure 37: EXTOLL cable connections .................................................................. 37
Figure 38: Backplane power connections on the bottom side .................................. 38
Figure 39: Liquid connection and power connection on the backplane ............... 38
Figure 40: EXTOLL cabling options ..................................................................... 39
Figure 41: Molex cable design specification ............................................................ 39
Figure 42: EXTOLL Y direction cables inserted in the backplane ....................... 41
Figure 43: EXTOLL cables in three different lengths .......................................... 41
Figure 44: Booster rack chassis with half a backplane mounted .......................... 42

List of Tables

Table 1: BNC firmware board management controller functions ......................... 20
Table 2: Characteristics of the EXTOLL Galibier card ........................................... 31
Table 3: Characteristics of the Pseudo-BIC servers .............................................. 32
Executive Summary

This deliverable describes the final design of the DEEP Booster prototype system and discusses the status of HW installed in Juelich. It focuses on integration of the Booster Nodes with the backplane into a chassis, including power supply, cooling and firmware.

Based on input from WP6, WP7, and tasks 3.1 and 3.2, the Booster Node Cards (“BNC”), Booster Interface Cards (“BIC”), backplane, chassis and firmware have been designed and implemented. The BNCs provide the highly scalable compute capacity for DEEP, while the BIC, Backplane, cabling and the EXTOLL network controllers implement the connectivity between Booster nodes and to the Cluster. Furthermore, the BIC bridges between the EXTOLL fabric used on the Booster and the InfiniBand™ interconnect of the Cluster.

The Intel® Xeon Phi™ co-processors are used as Booster Node CPUs. Two processors have been integrated within one BNC blade in combination with two EXTOLL NICs, forming two logical nodes. The BNC has been designed for this highly compact and energy efficient system. Both BNC as well as BIC are designed to be liquid-cooled, using the Eurotech Aurora technology. The intra-Booster network is based on the existing EXTOLL technology as provided by UniHD, ported to Altera Stratix FPGAs on the BNCs and BICs.

The FPGA running EXTOLL connects with the Intel Xeon Phi card via PCI Express (PCIe); to make this possible, a PCI root port mode was added to the EXTOLL implementation. A custom-designed backplane carries the EXTOLL BNC↔BNC and BNC↔BIC connections. Simulations and tests have been performed to verify the signal integrity of the different components and perform network validation. The backplane also distributes power to the BNCs and BICs, and the cooling liquid distribution is attached to it. The backplane also interconnects the management signals between the BIC and the BNC: 100 Mb Ethernet, SMBus, CAN, I2C and digital lines.

At the time of writing, 16 BNCs, two 8-slot backplanes and the rack for the DEEP Booster have been shipped to Juelich. The bring-up of the first chassis is underway, using a standard server platform to stand in for the BIC which is not yet ready.

The Deliverable presents details on the design and implementation of all HW components, and of the BNC firmware. It also reports on the mechanical, power and thermal design, signal integrity simulations and validation and memory subsystem implementation. At the end, it gives a succinct summary of the bring-up status.

1 Introduction

Work Package 3 covers the architecture, design and implementation of the Booster for the DEEP System. The prototype itself will serve to evaluate hardware and software on the way to Exascale computing.

While the Cluster subsystem of the DEEP Architecture is based on production-level commodity hardware, the Booster is a custom design that was developed and implemented in this Work Package. The overall DEEP Architecture consisting of a Cluster, a Booster and the bridging Booster Interface is shown in Figure 1.

The final DEEP System will consist of one cabinet holding an Intel® Xeon®-based Cluster (128 nodes) and one cabinet for the Intel® Xeon Phi™-based Booster (up to 512 co-processors of the “Knights Corner” product line). An additional rack that is placed between the two cabinets holds power supplies and interconnect equipment.
Successful integration of the prototype hardware with the environmental and operational infrastructure at the installation site (Juelich) is critical to be able to conduct a meaningful evaluation of the DEEP Architecture as a possible blueprint for building future energy-efficient HPC systems.

Among others, the DEEP System will be used for further experiments in hot-water cooling and energy-efficient computing above and beyond the results that have been obtained with existing hot-water cooling experimentation platforms at BADW-LRZ and UniReg\textsuperscript{[5],[6]}

The BNCs each hold two Intel Xeon Phi cards and two EXTOLL network interface controllers (NIC), the latter implemented on Altera Stratix V FPGAs, forming logically two Booster Nodes (BNs). The BNCs are interconnected via a custom designed backplane, which also carries management networks and power connections. Components on the BNCs are cooled by custom-made, high-precision coldplates, with liquid coolant connections passing through the backplane. Thermal validation shows that a fully loaded DEEP Booster can be cooled with treated water of up to 50 °C, enabling the use of free cooling for $\geq 98\%$ of the days at the installation site at Juelich, Germany.

The BICs use a COM Express board with a standard x86 CPU, up to two InfiniBand adapters and a single EXTOLL NIC. A PCI Express switch connects these components and allows avoiding load onto the CPU for EXTOLL$\leftrightarrow$InfiniBand communication.

Figure 1: Diagram of the DEEP architecture

As part of WP6 the DEEP infrastructure has been designed and implemented at Juelich. The Cluster subsystem has been integrated with the infrastructure and put into production use as reported in Deliverable D6.3. The lessons learned from this integration and the initial operation of the DEEP Cluster have been included in the design of the Booster subsystem rack. With the results of Tasks 3.1 and 3.2, the rack dimensions for a Booster holding up to 256 BNCs, 32 BICs, and 32 half backplanes in 16 chassis were defined.

Sections 2 through 5 describe the final design and the implementation of the relevant hardware and firmware components of the Booster: BNC, BIC, backplane, and cabling. Section 6 details the progress in integrating the HW components with the Booster rack (which was described in Deliverable D3.4) and the bring-up status of the system at the end of November 2014.


2 BNC Design and Implementation

2.1 BNC Architecture

The BNC is designed to implement the network, control and power distribution and cooling of the Xeon Phi cards. Figure 2 shows a block diagram of the BNC. The network interfaces are on the hardware-level realised by two Altera Stratix V FPGAs. Deliverables D3.1 and D3.2 contain a more detailed discussion on the BNC architecture and design.

![BNC block diagram](image)

Figure 2: BNC block diagram

In order to interconnect the BNC with the Xeon Phi coprocessor cards, corresponding riser cards were designed. A board and a cable connecting the front panel with the BNC board have been designed – this allows Ethernet access via the front panel. For the power connection of the Xeon Phi cards two auxiliary power cables were designed, which connect the 12 V DC power supply of the BNC board to each of the two auxiliary power connectors on the Xeon Phi cards. See Figure 3 for a schematic of the power distribution, sensors and control components on the BNC.
Figure 3: BNC sensors and control block diagram

The power supply per BNC card can deliver up to 700W at 54 V DC of which 300W are attributed to each Xeon Phi and up to 30 W to each FPGA.

Current sensors and voltage sensors are installed (HS in the block diagram in Figure 3) to measure the power consumption of each Xeon Phi card and the BNC card itself. The power is switched and converted to 12 V DC. An additional 3.3 V DC power line supplies standby voltage to an AVR µ-controller, to allow remote power-on/off of the whole node.

For programming the firmware stored in the flash memory on the BNC, the Serial Peripheral Interface (SPI) is used. The SPI is connected to the BMC for remote upgrading of the firmware of the board. Remote firmware upgrade is only possible for the BMC firmware and the FPGA firmware.

As described in Deliverable D3.2 and shown in Figure 4, the BNC is controlled via

- digital lines
- an I2C / SMBus and
- Ethernet.

The digital lines and the I2C bus are active in standby and are supplied by the 3.3 V DC standby voltage. The Ethernet connection is controlled by the BMC which is only active when the 12 V DC is activated.

The digital lines and the I2C bus are controlled by an Atmel AVR low-power RISC 8-bit µ-controller. This minimises the standby losses. This processor is connected to the Altera MAXII Complex Programmable Logic Device (“CPLD”) which manages power-on/off and other control functions.

The Board Management Controller (BMC) is an Emulex Pilot 3 with the AMI software development kit. It supervises and manages the board during operation and is accessible via Ethernet over the backplane and via the front panel.
The Atmel AVR μ-controller and CPLD have their logic programmed in the on-chip flash memory of 64K Bytes for the AVR μ-controller and 8 K bit for the CPLD. The AVR μ-controller and CPLD firmware are determined during production by loading a binary file into the on-chip memory and cannot be upgraded remotely. It is, however, possible to do field upgrades using JTAG, yet this will require disassembly of the system. The firmware of the BMC is stored in the separate “boot flash” device. Also the EXTOLL firmware on the FPGA is contained in the separate external “EPCQ flash” flash device. Both will be programmed during production, but are also field-upgradable after installation via the Ethernet connections. The boot flash can be updated by the BMC itself. The EPCQ flash can be updated by the BMC via the CPLD, using the SPI connection between the CPLD and the EPCQ flash devices.

2.2 BNC Electric Design

The PCB of the BNC is a 5U node card that is connected on one side to the two Xeon Phi cards and the front panel, and on the other side to the backplane.

The PCB dimensions are 210mm (length) × 224mm (width) × 2.4mm (height). The PCB has 18 layers and is made of high speed FR-4 material (NELCO N4000-13 SI). The maximum height of the components including the cold plate is 24 mm. The board pitch (slot width in the chassis) is 25 mm. The overall dimension of the BNC including cold plate, Xeon Phi cards and front panel is 540 mm (length) × 224 mm (width) × 24 mm (height).

The riser cards have 8 layers.

For the design the 10 GBase-KR (10G (speed), BASE (BASEband), K (bacKplane) and R (Random signalling)) design rules have been used and the high-frequency signals on the board were designed for 100 Ohm and 10 Gbit/s.

The BNC has about 1500 electric components mounted on the PCB. The main components are:

- FPGA Stratix V as network controller for the EXTOLL network
- BMC Emulex (ex-ServerEngines) Pilot3 SE-SM4310-S01P for board management
- Backplane Column Connector Molex, Impact TM 25 Gbit/s 100 Ohm Connector Family
- ALLEGRO ACS709LLFTR-20BB-T current sensors
• VICOR VI_CHIP and 1/8_BRICK Isolated 48 V DC to 12 V DC/3.3 V DC power supply
• Altera MAX II CPLD for power on/off control
• Atmel AVR µ-controller for I2C and digital line (standby) control

The connectors are:
  • On the front Panel
    o On/off button
    o Reset button
    o Gigabit Ethernet RJ45 connector with LEDs
    o Power-on LED
    o Reset/Standby LED
  • Backplane
    o FPGA0 X-,Y+,Y-,Z+,Z- 4 Gbit/s ×8 lanes bi-directional
    o FPGA1 X+,Y+,Y-,Z+,Z- 4 Gbit/s ×8 lanes bi-directional
    o FPGA0 7th link 4 Gbit/s ×4 lanes bi-directional
    o 100 Mb Ethernet connection to BMC
    o I2C connection to the AVR µ-controller
    o 4 digital lines to the AVR µ-controller for on/off and reset
    o 48 V DC Power connection
    o 3.3 V DC Standby voltage connection
    o UART connection
    o JTAG connection
    o CAN connection
  • Debug connectors
    o SPI header for reprogramming the flash memory of the BMC
    o JTAG connector for reprogramming the flash memory of FPGA0, FPGA1, AVR µ-controller and CPLD
    o (Mini-SAS, bottom side not mounted, 7th link EXTOLL connector)

The layout and more details of the BNC printed circuit board are shown in Figure 5.
Figure 5: BNC PCB design
2.3 BNC Signal Integrity Testing

The BNC’s signal integrity (SI) has been tested using the SI-toolkit of Altera, loaded on the FPGAs.

Having an ASIC implementation of EXTOLL in mind, the PCB was designed for speeds up to 10 Gbit/s, but the Altera Stratix V implementation of EXTOLL is limiting its speed to 4 Gbit/s per lane. This is due to the limitations of the FPGA in combination with the complex logic of EXTOLL. Therefore the SI has been measured at the speed of the EXTOLL network: 4 Gbit/s.

For the SI tests two BNC cards in combination with an 8-slot backplane were used.

By looping back first the links internally in the FPGA, the quality and stability of the FPGA design has been tested. After that, the external loopback over the backplane and the network cables has been tested and signal quality was verified (see Figure 6, Figure 7 and Figure 8).

Figure 6: BNC SI test setup for X direction measurements

Figure 7: BNC SI test setup for Y direction measurements
The eye diagram has been measured for the sending (Tx) and receiving (Rx) lines for each of the 8 lines in the Z+, Z-, Y+, Y- and X+ (FPGA0) / X- (FPGA1) direction. The X- (FPGA0) / X+ (FPGA1) is formed by LVDS links directly on the BNC PCB between the two FPGA’s. The data quality of these lines can be easily analysed and showed not to be critical.

All eye diagrams show good signal quality in all direction on all eight sending and receiving links with the maximum (worst-case) cable length of 60 cm.

The SI measurements for the 7th link are explained in chapter 3.3.
2.4 BNC Firmware Development

The BNC firmware implements the start-up sequence and reset and control functions. Booting of a DEEP Booster node consists of 4 stages. First the AVR µ-controller is powered up via standby power supplied over the backplane. With the AVR µ-controller powered, the Booster Node is still in standby and the power consumption is low (< 1W). When the “power on” signal is given, the AVR µ-controller activates the 12V DC on the nodes and the BMC and FPGA’s are powered up. The BMC checks “powergood” of the system and releases the Xeon Phi from reset. After that the BIC can start the OS on the Xeon Phi cards.

The Xeon Phi cards are controlled by (see Figure 11):

- Powering on/off the PCI slot: the power supplied to the Xeon Phi cards can be controlled by controlling the 48 V DC to 12 V DC converter.
- Controlling the PERST line of the Xeon Phi: the PCIe reset line (PERST) allows the Xeon Phi to be kept in reset or to give a reset when in operation.
- In-band over the EXTOLL network: EXTOLL has the possibility to manage the Xeon Phi over the high speed network.
- Out-of-band over the SMBus: the SMC on the Xeon Phi is connected by SMBus to the board management controller.

![BNC Xeon Phi control diagram](image)

Figure 11: BNC Xeon Phi control diagram

The firmware of the Atmel AVR µ-controller, CPLD and BMC is described in the following subsections.
2.4.1 *Atmel AVR μ-controller*

The basic functionality of the AVR μ-controller firmware covers controlling power on/off, reset and calibration of the current sensors. The AVR has general purpose IO pins (GPIO), analogue-to-digital converters, and CAN as well as I2C low-speed communication busses.

The AVR μ-controller controls the power on and off of the BMC and the FPGAs. Furthermore it controls the reset of the Xeon Phis by controlling the PERST reset line, and the reset of the BMC.

Since the AVR μ-controller is the first controller to be alive, it checks the standby voltage levels before powering up the BMC and CPLD. It also monitors the temperature sensors to avoid overheating of the BNC. In case the temperature is too high, the BNC will be put into standby.

The AVR μ-controller controls the BMC power on/off and reset over CAN. The power status of the CPLD and BMC can be checked via CAN as well. Also temperature sensors, voltage sensors and the current sensors can be queried via CAN.

Since the AVR μ-controller is controlling the board during standby, the current sensors are calibrated via the AVR μ-controller by a “0-current” measurement of the Hall sensors. The command for this calibration is given to the AVR μ-controller over the I2C bus over the backplane. There are also I2C commands for reading temperatures, voltages and currents during operation of the BNC.

The position of the BNC in the rack can be determined by the BIC as it is programmed in the AVR μ-controller on each BIC. The system administrator can get the position by requesting the BNC address with an I2C command over the backplane.

To control the firmware revision there is an I2C command to request the firmware versions from the AVR μ-controller.

2.4.2 *CPLD*

The main function of the complex programmable logic device (CPLD) is the power on/off and reset sequence. By programming the CPLD the power-up sequence, timing of the sequence and conditions for power-up can be programmed.

The CPLD is programmed as a state machine with 16 states. Those states include:

- On/off of each of the Xeon Phi cards
- Reset of each of the Xeon Phi cards
- Reprogramming of the flash image of FPGA0 and 1
- Configuring FPGA0 and 1

The state machine defines a number of stable conditions depending on its previous condition and on the present values of its inputs. By defining the states in which the BNC can be during power up, the different steps and the timing during the power on/off and reset can be controlled.

Since errors in the CPLD logic can damage the board, the CPLD cannot be reprogrammed by the user. The CPLD is programmed during manufacturing by loading a binary into the CPLD chip. The firmware can be updated with a JTAG programmer in the lab.

Figure 12 gives an overview of the different power states of the different components of the BNC and the transitions between them.
2.4.3 BMC

The Board Management Controller (BMC) manages the BNC. The management functions include reading sensors, controlling power, signalling errors, programming firmware, and providing information to the management network.

To allow high speed sensor data collection from all sensors in the DEEP Booster rack, the BMC uses MQTT machine-to-machine connectivity protocol to push the data to the sensor data collector on the BIC via Ethernet. The data pusher daemon can be started remotely. When active it will keep on pushing all sensor data every 10 msec to the BIC until the data pusher daemon is stopped.

The BMC controls the LEDs on the front panel to provide information about the status of the BNC. The LEDs are controlled by the general purpose IO (GPIO) pins of the BMC.

The BMC interacts with the AVR over I2C and uses digital lines for on/off and reset control.

The FPGA firmware can be programmed and updated by writing a bitfile into the EPCQ flash memory of the FPGAs. Also the BMC firmware can be reprogrammed by writing a bitfile...
into the flash memory of the BMC. The flash memory chips are connected over SPI to the BMC.

A graphic representation of the BMC firmware can be found in Figure 13.

![Figure 13: BNC board management control functional block diagram](image)

The BMC gives the system administrator management control of the BNC over Ethernet. The standard interface to control equipment in a data centre is the Intelligent Platform Management Interface (IPMI). Not all commands needed for the control of the BNC are defined in the IPMI standard. For these control signals custom commands are defined.

The functions of the BMC can be divided in 7 groups:

1. Node on/off: Each BNC and each Xeon Phi can be: turned on or off, soft reset and a hard reset (power cycle) can be initiated, and the status can be requested
2. Sensors: the BNC has temperature sensors, current sensors, voltage sensors and the sensors on each Xeon Phi card.
3. Set network: set the configuration of the Ethernet connection to the BNC
4. Update: update of the firmware of the BMC and FPGA 0 and 1
5. Connect: the Ethernet connection to the node
6. Interface: definition of the interface and commands
7. RAS features: UBoot loader of the BMC for remote booting and the control of general purpose IO signals over Ethernet for remote control of the Booster Node (more information will be worked out in D7.5)

Table 1 shows the main functionality of the BMC.
<table>
<thead>
<tr>
<th>feature #</th>
<th>group</th>
<th>feature name</th>
<th>custom command or IPMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Front Panel</td>
<td>Control of on/off button</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Front Panel</td>
<td>Control of reset button for hard reset</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Control of LED’s</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>node on/off</td>
<td>on / off via digital signal over backplane</td>
<td>command</td>
</tr>
<tr>
<td>5</td>
<td>node on/off</td>
<td>hard reset node via digital signal over backplane</td>
<td>command</td>
</tr>
<tr>
<td>6</td>
<td>node on/off</td>
<td>switch on Xeon Phi 1</td>
<td>command</td>
</tr>
<tr>
<td>7</td>
<td>node on/off</td>
<td>switch off Xeon Phi 1</td>
<td>command</td>
</tr>
<tr>
<td>8</td>
<td>node on/off</td>
<td>power cycle Xeon Phi 1</td>
<td>command</td>
</tr>
<tr>
<td>9</td>
<td>node on/off</td>
<td>soft reset Xeon Phi 1</td>
<td>command</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>show Xeon Phi 1 status</td>
<td>command</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>switch on Xeon Phi 2</td>
<td>command</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>switch off Xeon Phi 2</td>
<td>command</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>power cycle Xeon Phi 2</td>
<td>command</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>soft reset Xeon Phi 2</td>
<td>command</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>show Xeon Phi 2 status</td>
<td>command</td>
</tr>
<tr>
<td>16</td>
<td>sensors</td>
<td>get the entire sensor list (temperature, current) (by means of ADC, I2C and SMBUS drivers)</td>
<td>command</td>
</tr>
<tr>
<td>17</td>
<td>sensors</td>
<td>get Xeon Phi 1 &amp; 2 information over SMBus interface</td>
<td>command</td>
</tr>
<tr>
<td>18</td>
<td>sensors</td>
<td>MQTT client service pulling real time sensor readings to selectable broker</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>calibrating current sensor using I2C</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>GPIO driver</td>
<td>command</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>Time stamp updated with NTP server in the local network</td>
<td>IPMI</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>Boot loader supporting kernel boot from TFTP (BMC)</td>
<td>command</td>
</tr>
<tr>
<td>26</td>
<td>set network</td>
<td>set static IP</td>
<td>IPMI</td>
</tr>
<tr>
<td>27</td>
<td>set network</td>
<td>set static broadcast address</td>
<td>IPMI</td>
</tr>
<tr>
<td>28</td>
<td>set network</td>
<td>set static network mask</td>
<td>IPMI</td>
</tr>
<tr>
<td>29</td>
<td>set network</td>
<td>set static default gateway</td>
<td>IPMI</td>
</tr>
<tr>
<td>30</td>
<td>set network</td>
<td>set DHCP IP (only)</td>
<td>IPMI</td>
</tr>
<tr>
<td>31</td>
<td>set network</td>
<td>set DHCP IP and default gateway</td>
<td>IPMI</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>update boot loader via eth</td>
<td>IPMI</td>
</tr>
<tr>
<td>36</td>
<td></td>
<td>update BMC via Ethernet</td>
<td>IPMI</td>
</tr>
<tr>
<td>38</td>
<td></td>
<td>update FPGA 1 via Ethernet</td>
<td>command</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>update FPGA 2 via Ethernet</td>
<td>command</td>
</tr>
<tr>
<td>42</td>
<td></td>
<td>I2C communication with slave micro-controller (including commands for power status information)</td>
<td>command</td>
</tr>
<tr>
<td>43</td>
<td>connect</td>
<td>ssh to BMC</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>connect</td>
<td>set ssh authorized_keys</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>connect</td>
<td>set password of BMC admin</td>
<td>IPMI</td>
</tr>
<tr>
<td>46</td>
<td></td>
<td>change boot order of boot devices in boot loader (remotely)</td>
<td>IPMI</td>
</tr>
<tr>
<td>47</td>
<td></td>
<td>inject boot media</td>
<td>IPMI</td>
</tr>
<tr>
<td>48</td>
<td></td>
<td>web server</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>interface</td>
<td>get BMC firmware version (via network)</td>
<td>command</td>
</tr>
<tr>
<td>53</td>
<td>interface</td>
<td>custom command via network</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>-------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>54</td>
<td>IPMI stack that supports features (see column commands=yes)</td>
<td>Web server for having a graphical perspective of Booster's telemetry and power status</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>RAS features</td>
<td>UBoot bootloader save in flash environmental variables (mac address and Boosters serial number)</td>
<td>UBoot bootloader to support TFTP image load of Linux kernel</td>
</tr>
<tr>
<td>56</td>
<td></td>
<td>GPIO interrupts for managing soft shutdown and reset of the BMC itself</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td></td>
<td></td>
<td>command</td>
</tr>
<tr>
<td>58</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: BNC firmware board management controller functions

2.5 BNC Assembly

The BNC connects two Intel Xeon Phi 7120X cards by two riser cards (riser card “A” and “B”) to the BNC, using 8 lanes of PCI Express generation 2. The front panel buttons, LEDs and RJ45 socket are mounted on a small printed circuit board that is connected with a high speed flat cable to the BNC (see Figure 14).

![Figure 14: BNC Riser card A, Riser card B and front panel & cable](image)

The power of the Xeon Phi cards is supplied by in total four auxiliary power cables. When the BNC, the Xeon Phis, riser cards, front panel and cables are connected, a cold plate is attached for the cooling (see Figure 15).

![Figure 15: BNC with 2 Xeon Phi cards mounted](image)

The coldplate is bolted to the boards and the thermal contact between the electronics and the aluminium plate is made by thermal interface material.

After mounting the coldplate, the front panel cover plate with latches is mounted (see Figure 16).
3 BIC Design and Implementation

3.1 BIC Architecture

The BIC is designed to connect the Booster with the DEEP Cluster. It is also responsible for remotely booting and managing the Xeon Phi cards, and for controlling all 8 BNCs on a half backplane. The BIC bridges between the InfiniBand network of the Cluster and the EXTOLL network of the Booster. The bridging is performed by connecting InfiniBand host channel adaptors, an EXTOLL network FPGA and a control CPU to a PCI Express switch. To make the bridging efficient, the whole memory of the Xeon Phi cards controlled by a BIC is mapped into the PCI Express address space using EXTOLL’s shared memory functional unit (SMFU) engine. Since the Xeon Phi card has 16 GByte of memory, and up to 16 Intel Xeon Phi (KNC) co-processors are controlled by a single BIC, a total address space of more than 256 GB has to be provided on the BIC (see Figure 17).

The boot function of the Xeon Phi cards is controlled by a x86 CPU on a Computer-On-Module Express card (COM Express). All Xeon Phi cards that are assigned to a specific BIC are mapped to the MMIO space of the COM Express CPU. The EXTOLL network is transparent for the PCI Express commands to boot and control the Xeon Phi as PCI Express end points.

The BIC collects the high frequency sensor data from all 8 BMC MQTT daemons. The BIC CPU in the COM Express type 6 card is required to be a low power version to make the Booster as energy-efficient as possible. The memory of the COM Express card should be sufficient to run an in-memory database for the real-time high frequency data collection implemented by WP 7. Currently, an Intel Core i7 CPU and an AMD R-series CPU are considered.

The BNCs are connected to the BIC by Ethernet over the backplane. The BIC has an Ethernet switch with 8 downlinks and 3 uplinks. This switch has fixed ports and by reading out the port number, the position of the BNC on the half backplane can be determined.

The BIC controls and manages the nodes fault conditions. It can raise alarms to the external management server and protect the system by switching off in case of faults.
Figure 17: BIC block diagram

3.2 BIC Electric Design

The PCB of the BIC is a 230 mm wide card that is connected on one side to the front panel and on the other side to the backplane.

The size of the PCB is 500 mm (length) × 230 mm (width) × 2.4 mm (height). The PCB has 12 layers and is made of the same high speed FR-4 material as the BNC (NELCO N4000-13 SI). The maximum height of the components including cold plate is 30 mm. The overall dimension including coldplate, COM Express card, InfiniBand cards, and front panel is 540 mm (length) × 234 mm (width) × 30 mm (height), see Figure 18. The slot width of the BIC (board pitch) is 31 mm.

For the design the 10 GBase-KR (10G (speed), BASE (BASEband), K (bacKplane) and R (Random signalling)) design rules have been used. The high-frequency signals on the board are designed for 100 Ohm and 10 Gbit/s.
The BIC has about 1200 electric components mounted on the PCB. The main components are:

- FPGA Altera Stratix V as network element for the EXTOLL network
- Backplane Column Connector Molex, Impact TM 25Gbit/s 100 Ohm Connector Family
- High Speed Signal connector, SamTec HDI-6
- VICOR 1/8_BRICK Isolated 48 V DC to 12 V DC power supply
- PLX PEX8747 PCI Express switch
- Atmel ARM7-32 bit CPU for I2C and digital line control
- 2 pcs Marvell 88E6175 7 port Ethernet switch
- 5x Halo S010NZ Gigabit Ethernet transformers

The following plugin cards/components are mounted on the board connectors of the BIC:

- Advanet ADBC8037 type 6 COM Express; 8GB DDR3 and Intel Core-i7-3615QE, 2.3GHz, 45W
- Intel AXX1FDRBIOM single port FDR InfiniBand ConnectX-3 I/O module
- Intel 240Gbyte SSD 1,8” SSDSC1NB240G401

The connectors are:

- On the front Panel
  - On/off button
- Reset button for InfiniBand, PCI Express switch, FPGA, Ethernet switch
- 1x Gigabit Ethernet RJ45 connector with LEDs
- 2x USB
- VGA connector + power plug for external display
- 8x power good LED
- 8x port good at PLX
- Reset/Standby LED

- Backplane
  - 2x 7th link 4 Gbit/s x8 lanes bi-directional
  - 8x 100 Mb Ethernet connection to BMC
  - I2C connection to BNC
  - 8x 4 digital lines to the AVR μ-controller for on/off and reset
  - 48 V DC Power connection
  - 3.3 V DC Standby voltage connection
  - 8x UART connection
  - CAN connection

- Side Panel
  - 4x High Speed data connection Samtec HDI-6 for external EXTOLL connections
  - 1x (optional 2x) QSFP+ FDR InfiniBand connector
  - UART connector for debugging

- Debug connectors
  - SPI header for reprogramming of flash for BMC
  - JTAG connector for reprogramming of flash FPGA, Atmel ARM7 μ-controller
  - I2C and CAN debug connector

- HDD
  - Micro SATA connector for 1,8” SSD

The layout and more details of the BNC printed circuit board are shown in Figure 19.

![PCB layout](image1)
![PCB top view](image2)
![PCB bottom view](image3)

Figure 19: BIC PCB design
3.2.1 **BIC Ethernet switch**

The BIC has a Gigabit Ethernet switch with eight down links and three up links. The eight 100 Mbit/s down links are connected to the Ethernet port of the BNCs over the backplane. The COM Express card is connected to one of the two 1 Gbit/s uplinks. The other uplink is connected to the RJ45 plug on the front panel of the BIC. The ARM7 \( \mu \)-controller is connected to a RMII 100 Mbit up link.

As sketched in Figure 20 the switch is build up by two Marvel 88E6350 with 7 ports (5 with PHY).

![Figure 20: BIC Ethernet switch block diagram](image)

The switch is a level 2 switch with limited possibilities for VLAN. This will allow the system administrator to configure the switch for different users with different permissions. The BIC is delivered with the switch not configured and the switch behaves as a basic level 3 switch.

3.2.2 **BIC UART Mux**

The BIC connects the UART ports of all the BNC nodes. These UART port can be used as a serial console for debugging. The UART ports of the BNCs are connected over the Backplane to a multiplexer on the BIC. This allows the BIC to have a serial console connection to each of the BNCs one at a time. The BNC node is selected by addressing the multiplexer via the ARM7 \( \mu \)-controller. The UART serial console can be accessed from the COM Express board. See Figure 21.

![Figure 21: BIC serial console block diagram](image)
3.2.3  *BIC ARM7 μ-controller*

The basic functionality of the ARM7 μ-controller on the BIC is the power enable/disable (on/off) and reset of the 8 BNC nodes connected via the backplane to the BIC, to control the CAN bus towards the AVR μ-controller on the BNC nodes and to calibrate the current sensors in the system during standby (before switching on the system). The ARM7 also selects the BNC card for serial console. Only one serial console to one BNC node at the time can be used via the single COM Express UART connection.

The BIC’s position in the rack is detected by the ARM7, reading out dipswitch positions on the backplane. See Figure 22.

![Figure 22: BIC ARM7 block diagram](image)

**3.3 BIC Signal Integrity Testing**

The BIC’s signal integrity (SI) has been tested using the SI-toolkit of Altera, loaded on the FPGA of the BIC. The SI has been measured at 4 Gbit/s per lane, since this is the speed for which the EXTOLL network has been designed on the Stratix V FPGA. The two 7th link EXTOLL links to backplane have 4 active lanes per link each.

For the SI tests one BNC card in combination with one BIC on an 8 slot Backplane has been used. One 7th link is connected to the BNC in slot 0 and the other 7th link is connected to slot 7. (The 7th links of the BNC’s in slot 1 to 6 are not connected).

To improve the SI of the 7th link, the transceivers of the unused lanes are deactivated in the FPGA and the lanes tied to ground. Especially on lane 3 this has a positive effect on the eye diagram.
With this improvement the eye diagram for the 4 lanes between the BIC and the BNC in slot0 and slot7 of the backplane show good results.

See Figure 10 for the results of the 7th link measurements between BNC and BIC.

### 3.4 BIC Firmware Development

The BIC does not have a CPLD for booting. During power up the (self-bootable) COM Express board boots autonomously. After that it controls the reset lines via a buffer/repeater (1 reset line in, 8 reset lines out) and controls the reset line of the PCI Express switch, InfiniBand HCAs and FPGA at the same time.

Booting of the DEEP Booster interface node consists of 6 stages. When 12 VDC power is applied to the BIC, the ARM7 μ-controller and the COM Express board are powered up independently. After booting the OS, the COM Express card controls the reset line of the Ethernet switches, PCI switch, FPGA and InfiniBand cards. By releasing the reset lines, the BIC is powered up. After starting the OS on the ARM7 the BNC nodes can be switched on and the BNCs are powered up. When the BNCs are powered up the COM Express boards boots the Xeon Phi cards over EXTOLL.

The only firmware programmed to control and manage the BIC is that of the ARM7 μ-controller.

The software running on the COM Express board for collecting high frequency sensor data and resiliency is a deliverable of WP 7. The Cluster-Booster protocol and the software allowing for the remote-boot of the Xeon Phi via the EXTOLL fabric are deliverables of WP 4.

#### 3.4.1 Atmel ARM7 μ-controller

The ARM7 is programmed in its own dedicated programming language. The μ-controller’s GPIOs are programmed to manage the digital lines that control the on/off and reset of the BNC nodes.

The CAN bus is connected over the backplane with all the BNCs AVR μ-controllers. Over CAN the BNCs can be managed when the BNCs are in standby. The management includes reading voltages and current, power on/off and reset.

The ARM7 has a 10 bits analogue-to-digital converter used to measure the voltages and input current of the BIC.
The ARM7 is always powered on, even when the BIC is in standby. The power consumption is less than 1 W during standby.

### 3.5 BIC Assembly

During assembly the COM Express card, the ConnectX3 InfiniBand module and the hard disk are mounted on the BIC node card. Then the cold plate is attached to the assembly and the front panel cover is mounted.

The InfiniBand HCA has a custom heat spreader to cool the card from both sides, Figure 24.

![Figure 24: BIC card with SSD, InfiniBand and COM Express, without coldplate](image)

A SSD disk is attached to the Micro SATA connector. The COM Express card is plugged into its type 6 board-to-board connector and the InfiniBand HCA is connected to its PCIe gen3×8 slot.

After that the coldplate and front panel are mounted (see Figure 25).

![Figure 25: BIC with coldplate](image)
4 Pseudo-BIC

The “Pseudo-BIC” is a stop-gap solution to enable the bring-up of the 16 BNCs of the two half-Booster-Chassis while a working BIC is not yet available. This system boots and manages the KNCs in the attached BNCs, and connects them to the DEEP Cluster, providing therefore all the BIC functionality needed by the developers from WP4, WP5 and WP8 to install and test their software layers. Compared to the BIC being developed by Eurotech, the Pseudo-BIC has a number of disadvantages:

- It cannot connect to the Ethernet links carried over the Backplane; thus, Ethernet connections to all BNCs required for WP7 and for system management have to be implemented by running cables from an external Ethernet level 2 switch to the BNC front panels.

- It cannot be fully integrated with the Booster chassis (because of form factor), and uses air cooling, which complicates integration with the Booster rack.

- It cannot control BNC on the lowest level, i.e. the digital lines enabling the BIC’s ARM controller to pilot the BNCs’ board management controllers are not accessible. This makes system management less flexible and disables some of the fine-grained security measures. Thus, e.g. in case of the detection of a BN that’s too hot the whole chassis has to be shutdown instead of just powering down a single BNC.

Like a BIC, a single Pseudo-BIC controls eight BNCs. Therefore, two Pseudo-BICs are needed to bring-up the two half-Chassis of the Booster.

4.1 Pseudo-BIC design

The Pseudo-BIC is built from three off-the-shelf, and one custom-designed components: an Intel Xeon based server, an EXTOLL Galibier card utilising a Xilinx Virtex-6 FPGA, EXTOLL cables and a custom-built HDI6 connector which plugs into the Backplane. These components are described in the following subsections. Figure 26 shows how the Pseudo-BIC is currently integrated with the Booster chassis.

Figure 26: Pseudo-BIC and its components connected to a Booster Chassis.
4.1.1 **HDI adaptor card**
The HDI6 adaptor card (see Figure 27) is a PCB specifically designed and built to make the two EXTOLL 7th links of each Booster 8-slot Backplane accessible. The EXTOLL signals passed through the BIC Molex connector on the Backplane are routed through the HDI6 adaptor PCB into a Samtec HDI6 connector. A copper or optical EXTOLL cable can then be plugged in to take the signals to an EXTOLL Galibier card installed in the Pseudo-BIC server (see Figure 26).

![Figure 27: HDI6 connector card and its schematics.](image)

The HDI6 adaptor is plugged into the BIC slot of the half Backplane, which is situated on the upper part of the Booster Chassis (see Figure 26 and Figure 28).

![Figure 28: HDI6 adaptor card plugged into the BIC slot of the half-Backplane.](image)

4.1.2 **Galibier card**
The EXTOLL Galibier PCI Express card implements the EXTOLL protocol on a Virtex-6 FPGA and provides four EXTOLL ports. The card (Figure 29) is air cooled and can be integrated in any standard server supporting PCI Express.
The main characteristics of the EXTOLL Galibier card are enumerated in Table 2.

<table>
<thead>
<tr>
<th>EXTOLL Galibier card</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• VEL02 Small Message Engine</td>
</tr>
<tr>
<td></td>
<td>• RMA2 Engine</td>
</tr>
<tr>
<td></td>
<td>• Shared Memory Engine</td>
</tr>
<tr>
<td></td>
<td>• Virtualised Device</td>
</tr>
<tr>
<td></td>
<td>• Collective Operation Offload</td>
</tr>
<tr>
<td></td>
<td>• Lossless Fabric</td>
</tr>
<tr>
<td></td>
<td>• Switchless Design</td>
</tr>
<tr>
<td></td>
<td>• Arbitrary Topologies (with up to 4 links)</td>
</tr>
<tr>
<td>Software Support</td>
<td>• MPI Support (Open MPI 1.4.x, ParaStationMPI)</td>
</tr>
<tr>
<td></td>
<td>• Low-Level API access,</td>
</tr>
<tr>
<td></td>
<td>• Open Source</td>
</tr>
<tr>
<td>Operating System Support</td>
<td>• Linux: Kernel &gt; 2.6.30</td>
</tr>
<tr>
<td>Link Ports</td>
<td>• 4 Ports with 16Gbit/s per direction</td>
</tr>
<tr>
<td></td>
<td>• Link Connector: Samtec HDI6</td>
</tr>
<tr>
<td>FPGA</td>
<td>• Xilinx Virtex-6 LXT Series (XC6VLX240T)</td>
</tr>
<tr>
<td>Host Interface</td>
<td>8-lane PCI Express 2.0 5 GT/s → 4 GByte/s</td>
</tr>
</tbody>
</table>

Table 2: Characteristics of the EXTOLL Galibier card.

Within the Pseudo-BIC, the Galibier card is used to boot and control the 16 KNCs of its attached eight BNCs. It is also used to carry the Cluster Booster Protocol, implemented in WP4 to realise the communication between the DEEP Booster and Cluster.
4.1.3 **Server**

A two-socket Intel Xeon based server (see Figure 30) constitutes the “brain” of the Pseudo-BIC. It contains an InfiniBand host adapter and one Galibier card connected to its PCI Express slot. The combination of server & Galibier is responsible for performing the sequence to boot the attached 16 KNCs, and starting all other actions required to setup the system and its network within the half-Chassis.

![Figure 30: Servers for the Pseudo-BICs installed on top of rack, with one Galibier card each.](image)

The characteristics of the specific server model used to bring up the first Booster Chassis are described in Table 3:

<table>
<thead>
<tr>
<th>Model</th>
<th>Megware SlideSX® Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>0.5 U design</td>
</tr>
<tr>
<td>Nodes</td>
<td>Mainboard Dual-Socket Intel LGA 2011</td>
</tr>
<tr>
<td></td>
<td>2×CPU Intel Xeon E5-2609v2, 4 cores each, 2.5 GHz (1333 MHz)</td>
</tr>
<tr>
<td>Memory</td>
<td>64 GB main memory</td>
</tr>
<tr>
<td></td>
<td>8×8 GB DDR3 ECC registered 1333/1600 MHz</td>
</tr>
<tr>
<td></td>
<td>128 GB SSD MLC 2.5” SATA3 6 Gbit/s</td>
</tr>
<tr>
<td>Network</td>
<td>Dual Gigabit Ethernet RJ-45 onboard</td>
</tr>
<tr>
<td></td>
<td>Single FDR InfiniBand, Mellanox ConnectX-3 QSFP onboard</td>
</tr>
<tr>
<td>Air cooling</td>
<td>2×CPU Coolers SlideSX® design</td>
</tr>
<tr>
<td>Operating system</td>
<td>CentOS Linux</td>
</tr>
</tbody>
</table>

**Table 3: Characteristics of the Pseudo-BIC servers.**

After powering up the rack, the eight Board Management Controllers (BMCs) of the attached BNCs switch on the power of their KNCs and release all BNC level reset signals preventing them to start. In this state the EXTOLL Network across the BNCs is ready to be configured, and the KNCs are waiting for their boot image to be loaded. The management software on the Pseudo-BIC sets up the routing between the nodes inside the EXTOLL network. The management software uses the EXTOLL “Remote Register file Access” (RRA) feature of the Galibier card to setup the routing tables of all nodes in order to enable the torus topology.
implemented by the cabling. After this step every node has its unique EXTOLL network ID. This ID is used for the boot loader to download the KNC boot image to the 16 KNCs. After completion of the boot image download and the acknowledgment from each KNC that its OS is running, the nodes are ready for operation.

The Pseudo-BIC servers are also responsible to enable the Cluster-Booster Protocol between the nodes of the DEEP Cluster and Booster. They perform the tasks needed for the messages to be transferred between the EXTOLL network of the Booster and the InfiniBand fabric of the Cluster. For this purpose, the servers contain a built-in InfiniBand FDR HCA. Furthermore each Pseudo-BIC server is connected via Ethernet to the login nodes for remote access and management.

4.2 Pseudo-BIC tests

The HDI6 adaptor was tested by UniHD before the installation in the Booster Rack at Juelich. The board has a DC-DC converter to provide the 3.3 V DC power required for Active Optical Cables (AOC). Tests have confirmed that this voltage is stable and can supply optical transceivers. For the signal integrity tests of the HDI6 board a connection similar to the final Booster was used. A Galibier card in a Megware SlideSX server was connected to the HDI6 board plugged into a 8-slot half backplane with a single BNC mounted. PRBS patterns were exchanged between the Galibier card and the Stratix FPGA on the BNC. Measurements with an oscilloscope confirmed good signal quality on both ends. Further tests with this setup included low level network initialisation tests between the Altera Stratix FPGA and the EXTOLL Galibier card, enumeration of the two EXTOLL nodes on the BNC and booting a KNC attached to the BNC. All tests were successful.

4.3 Pseudo-BIC installation

Two HDI6 adaptor boards are installed within the Booster Rack in Juelich and copper-based connections are made between the boards and the Pseudo-BICs. Both Pseudo-BICs recognise their Galibier cards. The management software on the Pseudo-BICs is able to enumerate the first two nodes on the BNC next to the HDI6 board. The connection is stable. Further tests, such as enumeration of more nodes, firmware trouble shooting and corrections are ongoing to bring up all the 32 Booster Nodes.
5 Backplane Design and Implementation

5.1 Backplane Architecture

The backplane is designed to connect the BNCs to each other and to the BIC, and to distribute power.

The position of torus cable connectors was carefully chosen in such a way that all cables have a maximum length of 70 cm. On top of that the torus cables have to be mountable when the backplane is installed in the rack. Figure 31 indicates the position of the X, Y and Z direction connectors.

The maximum current to each backplane is 120 A (8×700 Watt per BNC + 200 Watt per BIC = 5.8 kW at 48 V DC = 120A). The power will be supplied to the backplane by mounting three pairs of distribution cables to the power taps on the bottom side of the backplane.

To determine the position of each board in the Booster the Backplane position is set by DIP switches on the PCB, see Figure 32. Each of the 32 BICs in the DEEP System will have its own unique “address”. For this, the DIP switches are read by the ARM7 μ-controller on the BIC.
5.2 Backplane Electric Design

The PCB of the backplane is one half of 23” wide card that is connected on the front side to eight BNCs, one BIC and the EXTOLL network cables and on the backside to the 54 V DC power distribution cables and additional EXTOLL network cables.

The PCB size is 265 mm (height) × 445 mm (width) × 5.0 mm (thick). The PCB has 24 layers and is made of the same high-speed FR-4 material as BNC and BIC (NELCO N4000-13 SI).

For the design the 10 GBase-KR (10G (speed), BASE (BASEband), K (backplane) and R (Random signalling)) design rules have been used and the high-frequency signals on the board are designed for 100 Ohm and 10 Gbit/s.

The backplane has about 160 electric components mounted on the PCB. The main components are:

- Backplane Column Connector by Molex, Impact TM 25Gbit/s 100ohm Connector Family
- POWER TAP M4 PRESSFIT
- A6H-0101 Omron Electronics DIP Switches for setting position in Booster

The main connections are:

- Power
  - 4 pairs of + - for 54 V DC/40 A(max)
- DIP Switches
  - 10 positions to set the position of the Backplane/BIC in the rack
- 8x BNC slot
  - FPGA0 X-, Y+, Y-, Z+, Z- 4 Gbit/s ×8 lanes bi-directional
  - FPGA1 X+, Y+, Y-, Z+, Z- 4 Gbit/s ×8 lanes bi-directional
  - 100 Mbit/s Ethernet connection to BMC
  - I2C connection to the AVR µ-controllers
  - 4 digital lines to the AVR µ-controllers for on/off and reset
  - 48 V DC Power connection
  - 3.3 V DC Standby voltage connection
  - UART connection
  - JTAG connection
  - CAN connection
- 1x BIC slot
  - 2× 7th link 4 Gbit/s ×8 lanes bi-directional
  - 8× 100 Mb Ethernet connection to BMC
  - I2C connection to BNC
  - 8× 4 digital lines to the AVR µ-controller for on/off and reset
  - 48 V DC Power connection
  - 3.3 V DC Standby voltage connection
  - 8× UART connection
  - CAN connection
- TORUS cable connectors in X, Y and Z direction for FPGA0 and FPGA1

The layout and more details of the backplane printed circuit board are shown in Figure 33.
5.3 Backplane Signal integrity Testing

The Backplane has been tested by measuring the SI between 2 BNCs and between a BIC and one BNC for all different slots.

Figure 34: Testing X direction over the Backplane between slots 3 and 4 using Molex cables

The SI quality of the signals over the backplane is good. The measured eye diagrams for the different slots on the backplane show good signal integrity for all the slots for both BNC as well as BIC.
5.4 Backplane Assembly
The Backplane is mounted in the rack after the mounting of the liquid distribution system and the open chassis. See Figure 35 and Figure 36 of the assembled backplanes.

Figure 35: Backplane top view

Figure 36: Backplane bottom view

After the backplane is mounted, the torus cables and the power cables can be mounted. See Figure 37 and Figure 38.

Figure 37: EXTOLL cable connections
When inserting the boards in the system, the liquid quick disconnect couplings are sticking through the backplane as shown in Figure 39. Currently the electric power connectors on the backplane are close to the liquid distribution. To avoid contact between the liquid distribution and power distribution the connectors will be covered.

In this way any contact between the cooling liquid and the electronics is avoided, also in case of a fault condition, like a leaking connector.
6 EXTOLL Torus Cables

6.1 EXTOLL Torus Cables Design

For the Backplane connections the Molex Impact TM 25 Gbit/s 100 Ohm Connector Family was chosen. The main reason for that is that this family of high-speed connectors offers both cable connectors as well as backplane connectors with different options (see Figure 40).

![Figure 40: EXTOLL cabling options](image)

Based on the DEEP requirements Molex has designed torus cables in 4 configurations with 0.2m, 0.4m, 0.6m and 0.7m length, respectively (see Figure 41).

![Figure 41: Molex cable design specification](image)

The signal pattern on the connector is the same for every length. The cables are “pass-through” female to female: pin1 on connector 1 has the same signal as pin1 on connector 2. This means that no cabling faults can be made by mounting the wrong cable in the wrong position or in the wrong direction.
For the cables 30AWG, 85 Ohm TwinAx is used. AWG stands for American Wire Gage. 30AWG means a high quality cable, with low losses at high frequencies.

Although the BNC, BIC and Backplane have all been designed using the 10 GBase-KR design rules and for 100 Ohm and 10 Gbit/s, the cables have 85 Ohm, which is a (telecom) standard and widely available. Using 85 Ohm instead of 100 Ohm is a price and availability optimisation.

This choice could in theory cause some reflections and communication problems. In-depth measurements were conducted to compare the use of 100 Ohm vs. 85 Ohm cables, and it turns out that at the speed needed by DEEP, reflections passing from 100 Ohm to 85 Ohm are marginal and of negligible influence on the performance.
6.2 EXTOLL Torus Cables Signal Integrity Testing

The torus cables have been tested by using one BNC plugged into a backplane (see Figure 42). The cable length has been varied from 0.2 m to 0.7 m (see Figure 43) and tested at 4 Gbit/s and 8 Gbit/s speed.

Figure 42: EXTOLL Y direction cables inserted in the backplane

Figure 43: EXTOLL cables in three different lengths

The cable length has no influence on the eye diagram at 4 Gbit/s. Only at 8 Gbit/s some influence on the eye diagram as function of the cable length can be noticed. Since the EXTOLL design for the Altera Stratix V does not support more than 4 Gbit/s, these tests show that the cables and the rest of the design have sufficient margins.
7 Integration of the Booster Rack

7.1 Mounting Boards in the Chassis

After the installation of the rack with liquid distribution and open frame chassis (as previously described in Deliverable D3.4), the backplanes were mounted. Figure 44 shows a chassis with an 8-slot backplane mounted.

When the backplanes are mounted, the power cables are attached from the AC/DC rectifiers via a fuse to the power taps on the bottom side (back side) of the backplane (see Figure 38).

The 3D torus cables can be attached when the backplanes are mounted in the rack. The 3D torus cables can also be mounted when the BNC’s and BIC’s are installed, yet it will be more difficult to reach the connectors (see Deliverable D6.5 for more details).

![Figure 44: Booster rack chassis with half a backplane mounted](image)

7.2 Hardware and Bring-Up Status at Juelich

At the end of November 2014, the following components had been shipped to Juelich:

- 16× BNC boards of version B1 (final design), with coldplates, front panels and 2 Intel Xeon Phi 7120X cards each. The boards were tested by Eurotech and have the initial release version of the BNC firmware installed.
- 2× 8-slot “left” backplanes of the final design, with material for connecting the 54 V DC power.
- Equipment needed to connect the front-panel Ethernet ports of the BNCs.
- Two chassis that each can hold 16 BNCs, 2 BICs and two 8-slot Backplanes (a left and a right half, respectively).

The EXTOLL cables required for connecting these 16 BNCs into a torus were expected to arrive shortly.
Using these components and the Booster rack described in D3.4, the bring-up of two half chassis with in total 16 BNCs has started:

- The two “left” 8-slot backplanes were installed (one per chassis).
- Firmware on three early B1 BNC boards was brought up to the first release level.
- The BNC boards were inserted into the chassis, and the cooling system was checked for leaks. Some leaks were identified and fixed.
- Power was applied to the backplanes, and one by one, the status of the BNCs was validated; then, the power-on steps up to and including step 3 in section 2.4 were performed.
- The HDI6 connector is used to connect an external server with an EXTOLL network card and a HDI6 cable to the backplane for power-on (see Section 4).
- Results are that power-on works up to and including step 3. Tests for step 4 (KNCs booting an OS) are currently ongoing.

No BIC hardware is available at this point of time. Eurotech has completed a redesign that fixes all bugs identified with the first prototype:

- Power supply stability and quality of supplied voltages.
- Correct configuration of PCI Express switch (partly done in HW).
- EXTOLL high-speed lane improvements.
- Provision of 8 lanes of PCI Express generation 2 or better between PCI Express switch and Altera Stratix V FPGA

Availability of first samples of this B0 version is expected before Christmas 2014, and after validation of the full BIC functionality, tests with the two chassis in Jülich are planned.

For the considered Intel Core i7-based COM Express boards, Eurotech has ordered a firmware (BIOS) extension that should provide the required PCI Express address space (sufficient to map the complete memory of all attached Xeon Phis). A first version of this extended BIOS is available and tested. The MMIO space cannot be set yet above 64 GB. BIOS developers and experts in Intel are looking into the problem to set the MMIO space to 512 GB for this CPU.
References and Applicable Documents

List of Acronyms and Abbreviations

A
- **AC/DC**: Alternating Current / Direct Current
- **ASHRAE**: American Society of Heating, Refrigerating and Air-Conditioning Engineers
- **Aurora**: The name of Eurotech’s cluster systems

B
- **BADW-LRZ**: Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften. Computing Centre, Garching, Germany
- **BIC**: Booster Interface Card: Interface card to connect the Booster to the Cluster InfiniBand® network
- **BMC**: Baseboard Management Controller
- **BN**: Booster Node (functional entity)
- **BNC**: Booster Node Card: A physical instantiation of the BN
- **Booster System**: Hardware subsystem of DEEP comprising of BNC, BIC and Intra-Booster network
- **BoP**: Board of Partners for the DEEP project

C
- **CAN bus**: Controller area network bus. Vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer.
- **CN**: Cluster Node (functional entity)
- **CooLMUC**: Prototype at BADW-LRZ with direct warm water cooling
- **CPU**: Central Processing Unit
- **COMe**: Computer-on-module (COM) form factor: highly integrated and compact PC that can be used like an integrated circuit component.
- **COM Express**: same as COMe.
- **CPLD**: Complex Programmable Logic Device
- **CPU**: Central Processing Unit
- **CRAC**: Computer room air conditioning

D
- **DC**: Direct Current (electricity)
- **DEEP**: Dynamical Exascale Entry Platform: EU-FP7 Exascale Project led by Forschungszentrum Juelich
- **DEEP Architecture**: Functional architecture of DEEP (e.g. concept of an integrated Cluster Booster Architecture)
- **DEEP Booster**: Booster part of the DEEP System
- **DEEP System**: The production machine based on the DEEP Architecture developed and installed by the DEEP project

E
- **EC**: European Commission
- **Energy Efficiency evaluator**: Platform used for the investigations of the energy-aware functionality of DEEP, used only in the DEEP project
EU: European Union
Eurotech: Eurotech S.p.A., Amaro, Italy
Exaflop: \(10^{18}\) floating point operations per second
Exascale: Computer systems or applications, which are able to run with a performance above \(10^{18}\) floating point operations per second
EXTOLL: High speed interconnect technology for cluster computers developed by University of Heidelberg

F
FLOP: Floating point Operation
FPGA: Field-Programmable Gate Array: Integrated circuit to be configured by the customer or designer after manufacturing

G
GPIO: General Purpose I/O

H
HCA: Host Channel Adapter
HDI6: High Density Interconnect
HPC: High Performance Computing
HW: Hardware

I
I2C: Inter-Integrated Circuit. Serial computer bus used for attaching low-speed peripherals to computer motherboards and embedded systems.
IB: InfiniBand
InfiniBand: Computer network communications link used in high-performance computing
Intel: Intel GmbH Braunschweig, Germany
Intel Xeon\textsuperscript{®} Phi\textsuperscript{TM}: official product name of the Intel Many Core (MIC) architecture processors. The first available Intel Xeon\textsuperscript{®} Phi\textsuperscript{TM} product is code-named Knights Corner (KNC).
IPMI: Intelligent Platform Management Interface
IT: Information Technology

J
JSC: Juelich Supercomputing Center
JTAG: Joint Test Action Group. Standard test access port and boundary-scan architecture
JUELICH: Forschungszentrum Jülich GmbH, Jülich, Germany

K
KNC: Knights Corner: Code name of a processor based on the MIC architecture

L
LDAP: Lightweight Directory Access Protocol
LED: Light-emitting diode.
LVDS: Low-Voltage Differential Signalling.

M

MMIO: Memory-mapped I/O.
MPI: Message Passing Interface: API specification typically used in parallel programs that allows processes to communicate with one another by sending and receiving messages
MQTT: machine-to-machine connectivity protocol
MTBF: Mean Time Between Failures

N

NIC: Network Interface Card: Hardware component that connects a computer to a computer network

O

OS: Operating System

P

ParaStation Consortium: Involved in research and development of solutions for high performance computing, especially for cluster computing
ParaStationMPI: Software for cluster management and control developed by ParTec
ParTec: ParTec Cluster Competence Center GmbH, Munich, Germany
PC: Normally Personal Computer, but in the context of the proposal also Project Coordinator
PCB: Printed Circuit Board
PCI: Peripheral Component Interconnect. Standard for attaching components to a computer system. It covers mechanical, electrical and logical aspects.
PCI-Express: An implementation option of PCI using high speed serial links as physical interconnect layer.
PCIe: Same as PCI-Express
PERST: PCI express fundamental reset signal
PHY: PHYsical layer, a circuit that en/decodes between a digital and an modulated analogue signal
PLX switch: switch to connect a single PCI Express port to multiple end-points, produced by the company PLX Technology
PMT: Project Management Team of the DEEP project

Q

QDR: Quad Data Rate: Communication signalling technique of InfiniBand®
QSFP: Quad (4-channel) Small Form-factor Pluggable. Compact, hot-pluggable transceiver used for data communications applications

R

DEEP - 287530 47 16.01.2015
RAS: Reliability, Availability and Serviceability
RDMA: Remote Direct Memory Access
RISC: Reduced Instruction Set Computing. CPU design strategy.
RJ: Registered Jack. Standardised physical network interface. Followed by a number is an specific connector model.
SAS: Serial Attached SCSI (Small Computer System Interface). Point-to-point serial protocol that moves data to and from computer storage devices
SATA: Serial ATA (AT Attachment). Computer bus interface to connect host bus adapters to mass storage devices.
SCIF: Symmetric Communication Interface from Intel
SI: Signal Integrity.
SIMD: Single Instruction, Multiple Data. Describes computers with multiple processing elements that perform the same operation on multiple data points simultaneously
SMBus: System Management Bus
SMFU: Shared Memory Functional Unit
SPI: Serial Peripheral Interface
SSD: Solid State Disk
SW: Software

T

U
UART: Universal Asynchronous Receiver/Transmitter
UniReg: University of Regensburg, Germany
UPS: Uninterruptible Power Supply
USB: Universal Serial Bus

V
VELO: Virtualised Engine for Low Overhead: An EXTOLL communications channel

W
WP: Work Package

X

Y

Z