HMC Extension Module
HW Detailed Design
(draft)

	Organisatie / Organization	Datum / Date
Auteur(s) / Author(s): Gijs Schoonderbeek	ASTRON	17-03-2015
Controle / Checked: A.W. Gunst	ASTRON	17-02-2015
Goedkeuring / Approval: A.W. Gunst	ASTRON	
Autorisatie / Authorisation: **Handtekening / Signature**	ASTRON	

Distribution list:

Group:	Others:
Andre Gunst Eric Kooistra Sjouke Zwier Daniël van der Schuur Jonathan Hargreaves	UniBoard2 Project Group

Document history:

Revision	Date	Chapter / Page	Modification / Change
0.1	13-01-2014	Schoonderbeek	Creation
0.2	08-01-2014	Schoonderbeek	Update, add architecture + background information.
0.3	11-02-2014	Schoonderbeek	
0.4	16-02-2014	Schoonderbeek	Genaral Update
0.5	27-02-2014	Schoonderbeek	Update Section 7

Table of contents:

List of figures:

List of tables:

Terminology:

BCK	Backplane interface
bps	Bits per second
BW	BandWidth
DC	Direct Current
DC/DC	DC tot DC converter
DDR4	Double Data Rate type 4 Memory
ETH	Ethernet
FPGA	Field Programmable Gate Array
FLIT	Flow control digit (header, body and tail flits
Gbps	Giga Bit Per Second
Hardware	Boards, subracks and COTS equipment
HEM	HMC Extension Module
HMC	Hybrid Memory Cube
LVDS	Low Voltage Differential Signaling
IO	Input Output
JTAG	Joint Test Action Group, Interface for Boundary Scan and programming devices like FPGA's
Link	Bundle of 8 or 16 lanes
Lane	Single transceiver pair (Tx and Rx) within a link
PCB	Printed Circuit Board

PIM	Power Input Module
POL	Point of Load
PHY	physical interface (layer 1 of OSI model)
OBT	On-Board Transciever, Parallel optics from FCI
QSFP	Quad Small Form factor Pluggable
RoHS	Reduction of Hazardous Substances
SODIMM	Small Outline Dual Inline Memory Module
TAP	Test Access Point
TSV	Through-Silicon Via
Xo	Crystal Oscillator

References:

[1] Deliverable 8.2 UniBoard^2 Hardware Design Document, Astron, Gijs Schoonderbeek, 07-10-2013, ASTRON-TN-040 1.0 3
[2] UniBoard2 Hardware design document, Astron, Gijs Schoonderbeek, ASTRON-TN-040 1.1
[3] UniBoard2 HW Detailed Design, Gijs Schoonderbeek, ASTRON-TN-042 1.0
[4] Hybrid Memory Cube with HMC-15G-SR PHY Datasheet, Micron Technology Inc. Rev B 1/14 EN
[5] Altera Generation 10 Design Seminar chapter 6, Interfacing to the Hybid Memory Cube using Arria 10 devices, Altera 2014
[6] MiniPOD™ AFBR-81uVxyZ Twelve-Channel Transmitter, AFBR-82uVxyZ - Twelve-Channel Receiver datasheet, Avago Technologies, July 12, 2012
[7] TN4308 Hybrid Memory Cube Schematic Review Checklist and Pin Connection Guidelines, Micron Technology Inc., Rev. A/14
[8] UG-01152 Hybrid Memory Cube Controller IP Core, User Guide, Altera Corporation, 19-9-2014
[9] Datasheet AFBR-79EQDZ 40 Gigabit Ethernet & InfiniBand QSFP+ Pluggable, Parallel Fiber-Optics Module, Avago Technologies, January 28,2013, AV02-2924EN

1 Introduction

1.1 Scope

In this document the detailed hardware design for the Hybrid Memory Cube (HMC) Extension Module (HEM) is described. This board will be used in combination with UniBoard^2 [1]. The goal of this document is to make all design choices before the schematic and PCB design is started. HEM is part of the UniBoard^2 project which is a Joint Research Activity (JRA) in the RadioNet3 project, funded by the EC through the FP7 programme, under grant agreement no. 283393. The partners in this JRA are the Universities of Bordeaux and Orleans, INAF, MPG Bonn, the University of Manchester, ASTRON and JIVE.

1.2 Board Overview

In Figure 1 an overview of HEM (HMC Extension Module) in combination with a UniBoard^2 is shown. In this block diagram a first concept of HEM is used. The final block diagram will be described at the end of this document.

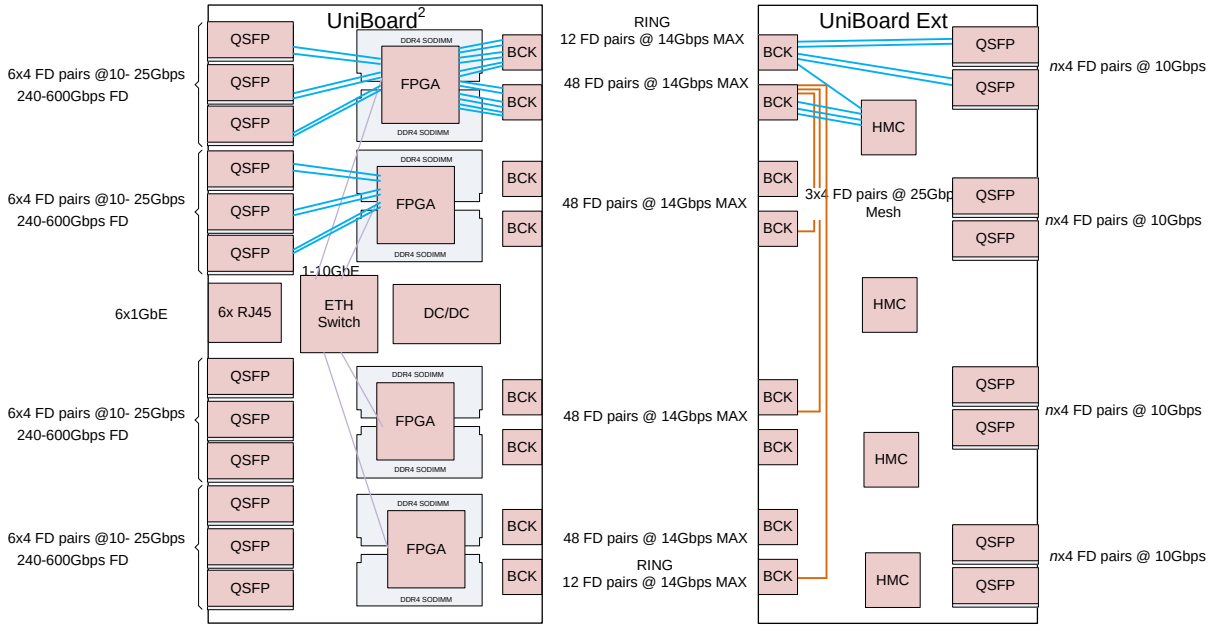

Figure 1 Overview of UniBoard^2 in combination with HEM

In the following section an overview of the technologies available for HEM is discussed. Thereafter an overview of the board options is described.

1.3 Purpose of HEM

HEM, as part of the UniBoard2 project, will be a useful technology demonstrator. Both the evaluation of technologies and the practical use of the boards are key. With HEM we want to evaluate:

- An alternative memory for DDR4 (higher memory bandwidth)
- Memory in the mesh / corner turning (write from one node, read from another node)
- Write through a HMC into another HMC (increase depth of memory)

2 Technology

2.1 HMC

2.1.1 Background

Hybrid Memory Cube is a memory device with a four links where each lane consists of sixteen serial interfaces. A simplified block diagram of a HMC device is shown in Figure 2.

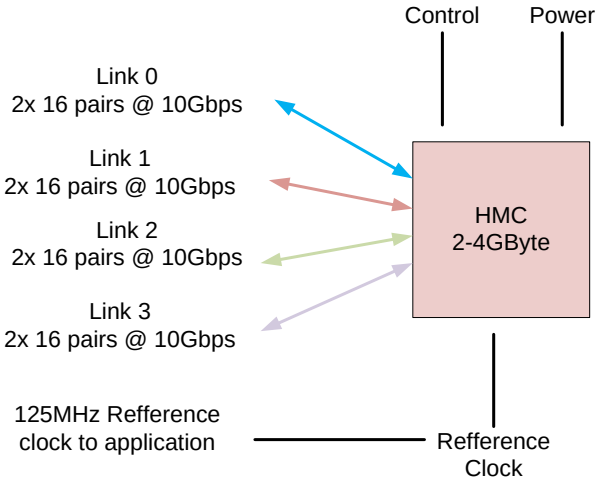

Figure 2 Overview of HMC device

The serial data arrives at the HMC on the IO layer of the device. On top of this physical layer, multiple layers of RAM chips are placed each interconnected with through-silicon via (TSV) technology. Data transfer to and from the device are packed in FLITs, a packet with a header, a tail and optionally a data field. By using this protocol, flow control including data checking and packet ordering is achieved. In Figure 3 two example options of connecting to a HMC are shown.

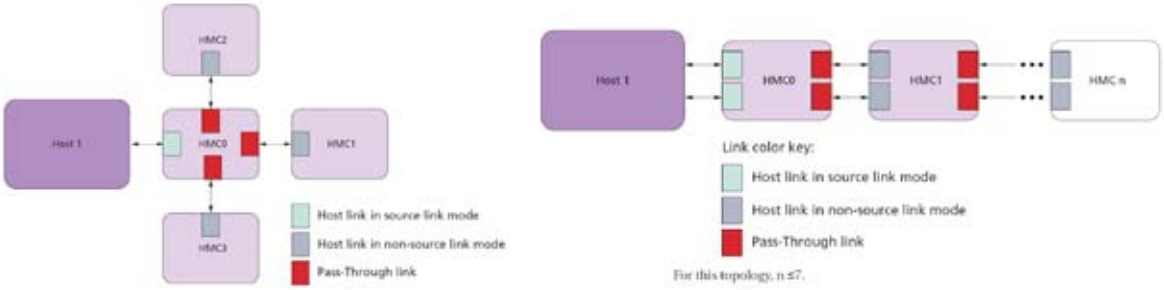

Figure 3 chaining multiple HMC devices

More information about HMC can be found in [4] and [5].

2.1.2 Device

For the first proto the HMC-15G-SR MT43A4G80100NFH-S15 is selected. The selection at the time of writing this document is limited. The MT43A4G80100NFH-S15 is 4GBytes (largest device) with four links (max. number of links).

2.1.3 Glue logic

For the control lines between the FPGA and the HMC a level converter will be placed to make sure that the I^2C levels comply with the specification.

2.1.4 Transceiver connections

Each HMC device has four lanes, where each lane has 16 Transceivers. AC coupling in the Rx line will be used (like is used on UniBoard^2). According to [7], the polarity of a lane can be inverted within an Rx pair, and lane order can be reversed within a link if needed for routing requirements. Both lane polarity and lane order are detected and corrected during initialization.
The order of the transceivers (in each link) can be swapped even further by the used of the Non-Default RX/TX Mapping Parameter Value, see [8].

2.1.5 Configuration

An overview of the control lines is shown in Table 1.

Table 1 HMC control lines

Signal	Description	Default state / connection
LxRXPS / LxTXPS	Power reduction pins	Fixed enable
FERR_N	Fatal Error indication	To FPGA
TDI, TDO, TMS, TCK, TRST_N	JTAG signals	To JTAG bridge
SCL / SDA	I2C interface	To FPGA
CUB[2:0]	I2C address	Fixed to address
REFCLK_BOOT[1:0]	PLL Configuration	Fixed
REFCLKSEL	Clock coupling AC/DC	Fixed

2.1.6 Power

2.1.6.1 Requirements

An overview of the power consumption is shown in Table 2.

Table 2 HMC power supplies

Signal	Description	Voltage	Accuracy	Current	Power
VDD	Core power source	0.9V	±0.027V	6.4A	5.8W
V_{TT}	Link TX Termination source	1.2V	±0.06V	3A	3.6W
V_{TR}	Link RX Termination source	1.2V	±0.06V	3A	3.6W
VDDPLLA	Link A source	1.2V	±0.06V		0.5W
VDDPPLb	Link B source	1.2V	±0.06V		0.5W
VDDPLLR	Intermed. Freq. PLL Source	1.2V	±0.06V		0.05W
VDDM	DRAM Source	1.2V	±0.06V	4.8A	5.8W
VCCP	DRAM Wordline boost Source	2.5V	±0.125V		0.1W
VDDK	JTAG I2C	1.5V	-0.05/+0.2V		0.05W
				TOTAL	20W

2.1.6.2 Power sequencing

Power Sequencing is not needed for HEM.

2.1.6.3 *Implementation*

In [3] the power tree for the FPGAs in UniBoard^2 is shown. On HEM a comparable tree will be used. In Table 3 an overview of the power supplies of HEM is shown.

Table 3 Implemented Power supplies

Name	Voltage	Current	Number POLs	Used for
HMC_VDD	0.9V	≥ 10A	Each HMC	V_{DD}
HMC_1V5	1.5V	≥ 5A	Central	V_{DDK}
HMC_1V2	1.2V	≥ 15A	Each HMC	V_{TT}, V_{TR}, V_{DDM}, V_{DDPPLx}
HMC_2V5	2.5V	≥ 5A	Central	V_{CCP}

On each PLL power supply a filter will be implemented. All power supplies will have 100nF decoupling capacitors.

2.2 Board IO

Not all transceivers available on UniBoard^2 have to be used for the HMC memory devices. Some IO (e.g. the ring transceivers) can be used to couple multiple UniBoard^2 boards. For this IO the standard QSFP, as used on UniBoard, can be used. Or by the use of parallel optics, the number of cables and board space can be saved. In the following subsections two parallel optic solutions are described.

2.2.1 MicroPod

Instead of links with four transceivers, cable and cost can be reduced by using parallel optic modules. An example of an optical module is the Avago Micro modules which can transmit or receive 12 transceiver channels. The maximal data rate is module dependent and can range from 10 to 15 Gbps. For a full duplex connection, both transmit and receive modules are necessary.

Figure 4 Avago MicroPOD

2.2.2 FCI OBT

FCI has developed the Leap™ On-Board-Transceiver (OBT) system, see Figure 5. The one inch square board mounted optical module features 12-transmit and 12-receive channels, each working at maximal 25Gb/s over distances up to 100m with a total of 300Gb/s throughput. The OBT is scheduled to be available in the second half of 2015.

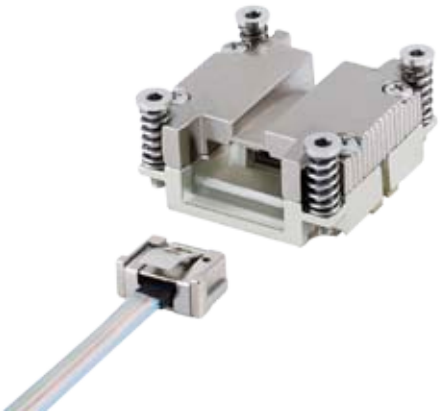

Figure 5 FCI Parallel optical module

2.2.3 Overview board IO

In Table 4 an overview of the board IO options are shown.

Table 4 Overview of Board IO options

	QSFP	MicroPOD	FCI OBT
Board space	-	+	
Cost for 12 tranceivers	€600 (3x€200)	€478 (€333+€145)	
Nof. Cables	3	2	
Connection to others	+++	-	
Max. Speed	25Gbps	15Gbps	
Pro	Used on UniBoard^2	Astron Unkown	New device
Con	More cables / board space	reduction of 1 cable	

Although the maximal speed of QSFP is 25Gbps, the data rate of the ring is limed to the transceivers speed of the ring which is limit to 14Gbps. For HEM the QSFP is chosen, this is known and standard technology which increases the use of HEM.

3 Scenarios

3.1 Corner turning

The block diagram of a typical beamforming/correlator application is shown in Figure 6.

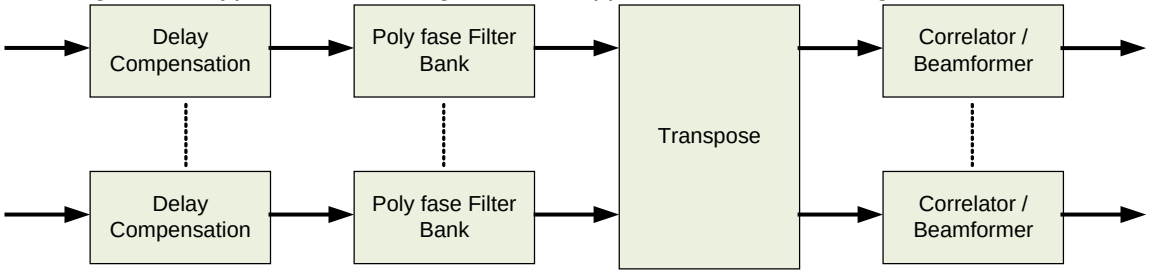

Figure 6 Block diagram of a typical application

For this application a continuous data stream from multiple signal inputs (e.g. antennas) arrive at multiple FPGAs. Commonly the first processing step is antenna based, e.g. delay compensation or a filterbank operation. The next processing step needs to be done for all the antennas, but with a sub-set of the bandwidth. Hence in between those processing steps a transpose operation is required, which re-orders the

data. When applying HMC modules, the outputs of the filterbank can be stored in the HMC for all signal inputs, all frequency channels and a number of time slices. Other FPGA's (or other parts of the FPGA) read data from the HMC in a different format, e.g. from all signal inputs for only a limited number of frequency channels for a number of time slices. This rotated dataset is read and processed at a single FPGA node, see Figure 7.

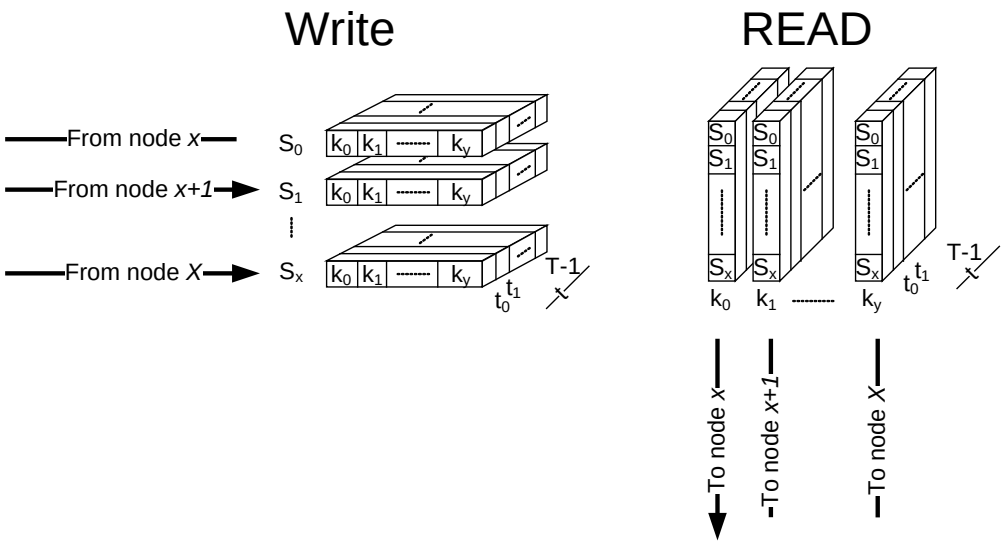

Figure 7 Transpose operation (also referred to as corner turning)

The typical memory size for these applications is two seconds of data (read one second, write next second of data). With data arriving at 10Gbps at the input (1GSample 10 bits) for sixteen inputs per FPGA, the total write and read data rate is 160Gbps, with a depth of 40GByte. At the moment these sizes are not possible. In this case the 1 second has to be reduced to e.g. a 1/10th of a second. This result in a higher output data rate and more post processing.

3.2 Store and process

When data arrives from different antennas, the data has to be aligned on a point in the sky before it can be processed. This can be accomplished by writing the data in a circular buffer and read out a number of samples later. The data rate for these applications is in the range of 10Gbps for sixteen inputs. The size depends of the application as is in the range of 500ms.

4 Board Options

4.1 Triple HMC (HMC in the Mesh)

Each HMC module has four links of 16 SERDES interfaces (lanes). By connecting each HMC interface to a different FPGA interface, data can be written by one FPGA to the HMC and read out by another FPGA, see Figure 8. For this architecture all backplane transceivers are used. The ring interface can be closed on HEM, but a better solution will be an optical connection (see Section 2.2). In this case multiple UniBoard^2 boards can be connected in a ring.

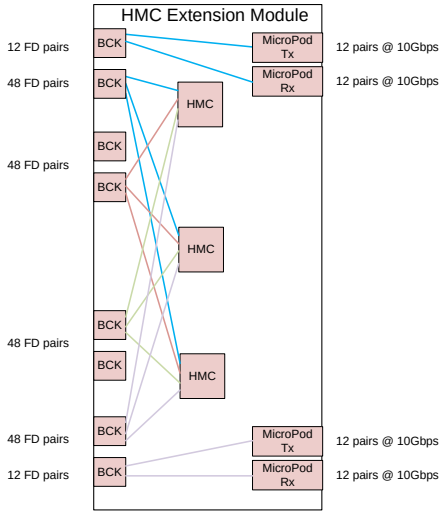

Figure 8 Three HMC Extension Module

The specification for this architecture is shown in Table 5

Table 5 Triple HMC module specification

Speed to Memory	1.92Tbps (4x3x160Gbps)
Memory depth	12GByte (3x4GByte)
IO speed	2x120Gbps FD
Mesh Speed Mesh	-
Usage for corner turning	- direct write and read from HMCs - 480Gbps data transfers - 3Gbyte memory depth per FPGA
Usage for store and process	- Data located at multiple HMCs - 480Gbps data transfers - 3Gbyte memory depth per FPGA
Layout	Crossings in layout
Firmware	Only Memory interfaces

An example of a typical application.
With a signal input connected to FPGA node 0, the data is split in frequency channels by a filterbank. A third of the channels will be stored in each HMC. All FPGA's read a quarter of the dataset to process it further (e.g. correlation or beamforming).

4.2 Quad HMC

For the quad HMC architecture option, each FPGA has its own HMC, see Figure 9. In this case maximal 3 links per HMC are used. This will limit the bandwidth to the HMC but increases the memory size (with respect to 3x HMC). Optionally only one or two interfaces could be used to the HMC as well. In that case the other interfaces can be used for the mesh or IO.

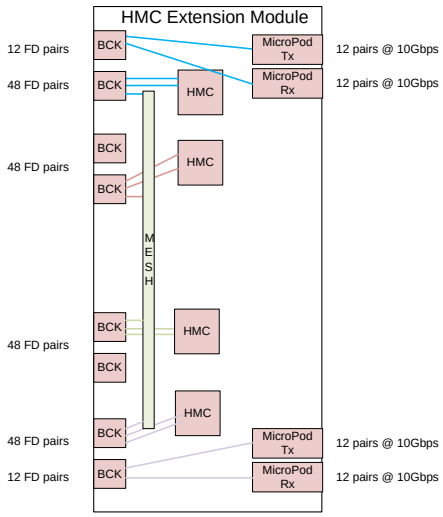

Figure 9 HEM with four HMC modules

The mesh can be made out of 16 transceivers, resulting in a mesh with 5 transceivers (floor(16 transceivers / 3 interfaces to other FPGAs) = 5 lanes per interface) to each FPGA. The ring interface could be used as well to create a full mesh (Figure 10). In this case the ring can be 12 transceivers wide resulting in a maximal data throughput of 168Gbps FD.

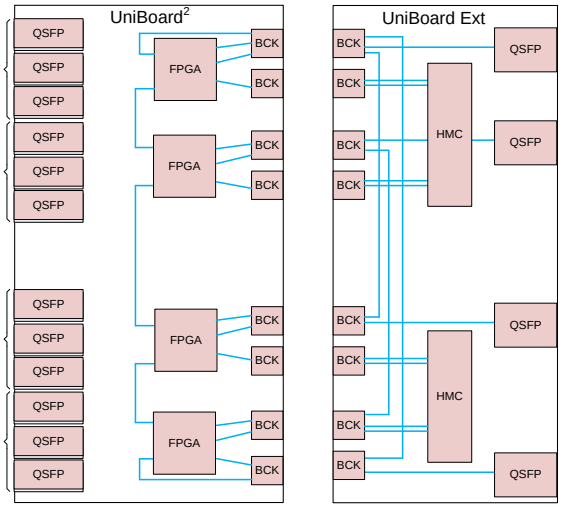

Figure 10 Using the ring interface for the mesh

The specification for this architecture is shown in Table 6

Table 6 Quad HMC module specification

Speed to Memory	1.28Tbps (4x2x160Gbps)
Memory depth	16GByte (4x4GByte)
IO speed	
Mesh Speed Mesh	-
Usage for corner turning	Mesh used transpose
Usage for store and process	- Data single HMCs - 320Gbps data transfers - 4Gbyte memory depth per FPGA
Options	Add HMC in parallel to increase depth
Layout	Crossing due to mesh, long traces
Firmware	Mesh and memory interfaces

4.3 Dual HMC

A combination of the three and four HMC's is the dual HMC architecture, see Figure 11. For this architecture 16 transceivers per FPGA are used for the mesh (or create a full mesh in combination with the ring interface) and two times 16 transceivers for two HMC modules.

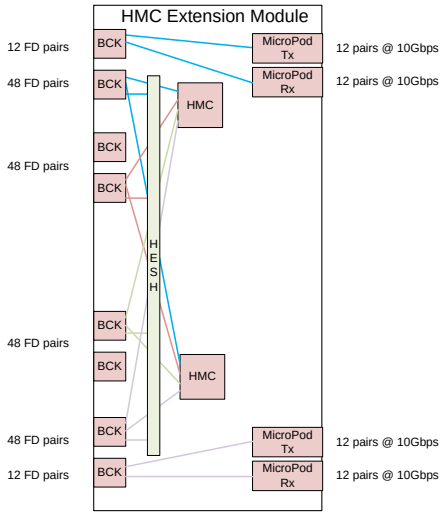

Figure 11 HEM with two HMC modules

The specification for this architecture is shown in Table 7.

Table 7 Quad HMC module specification

Speed to Memory single FPGA	320Gbps
Total memory speed	1.28Tbps (2x4x160Gbps)
Memory depth	8 GByte (2x4GByte)
IO speed	
Mesh Speed Mesh	-
Usage for corner turning	- direct write and read from HMCs - 320Gbps data transfers - 2 Gbyte memory depth per FPGA
Usage for store and process	- Data multiple HMCs - 320Gbps data transfers - 2 Gbyte memory depth per FPGA
Layout	Crossings in layout, long traces
Firmware	Mesh and Memory interfaces

4.4 HMCs in a Ring

Another option of the four HMC's is the HMCs in a ring architecture, see Figure 12. For this architecture 16 transceivers per FPGA are used to create a mesh (a full mesh can be created when the ring interface is used as well) and two times 16 transceivers for two HMC modules. The HMC devices are connected in a ring structure. In this way data can be written in a neighbour HMC.

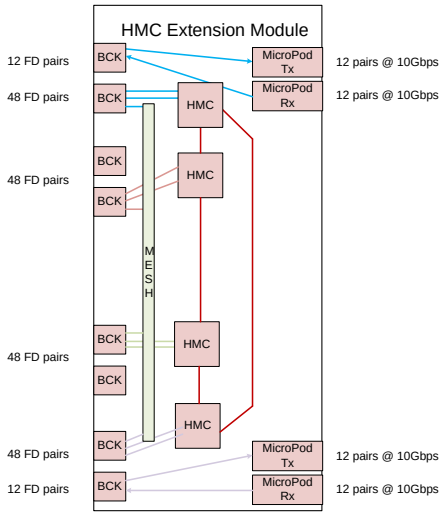

Figure 12 Architecture where the HMC are place in a ring

The specification for this architecture is shown in Table 8

Table 8 Ring HMC module specification

Speed to Memory single FPGA	320Gbps
Total memory speed	1.28Tbps (4x2x160Gbps)
Memory depth	16 GByte (4x4GByte)
IO speed	
Mesh Speed Mesh	-
Usage for corner turning	- pass data through HMC's for corner turning - 160Gbps - 4 Gbyte memory depth per FPGA
Usage for store and process	- Data multiple HMCs - 320Gbps data transfers - 4 Gbyte memory depth per FPGA
Layout	Few crossing due to mesh. Short traces
Firmware	Mesh and memory interfaces

4.5 Ring or FPGA/HMC

The four HMC's can also be used in a ring FPGA/HMC architecture, by using a ring of FPGA-HMC-FPGA-etc. (Figure 13). For this architecture all 48 transceivers are used to connect the FPGA's to the HMC. Each FPGA has a two links to its "own" HMC and one link to its neighbour HMC, see Figure 13.

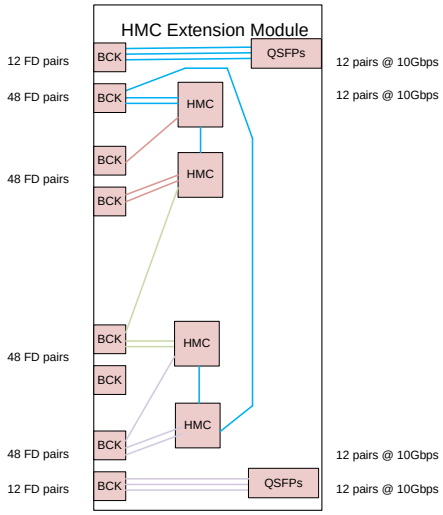

Figure 13 Architecture where the HMCs and FPGAs are placed in a ring

The specification for this architecture is shown in Table 9.

Table 9 Ring FPGA/HMC module specification

Speed to Memory single FPGA	320Gbps
Total memory speed	1.28Tbps (4x2x160Gbps)
Memory depth	16 GByte (4x4GByte)
IO speed	
Mesh Speed Mesh	-
Usage for corner turning	Use FPGA mesh for corner turning
Usage for store and process	- Data multiple HMCs - 480Gbps data transfers - 4 Gbyte memory depth per FPGA
Options	Optimal for distributed processing, store between the processing steps.
Layout	Simple layout, minimal number of crossings, small lines
Firmware	Single interface

An example of a typical application.
Data can arrive in the first FPGA where processing is done. When the processing is finished the data is written into memory. The next FPGA reads the data, optionally in a different format, for processing and dumps the results in the next HMC. This “process → store → process→...” is typical done when programming in OpenCL.

5 Glue Logic

5.1 Backplane Interface

For the backplane interface the Molex/TE Impact range connectors are used (Figure 14).

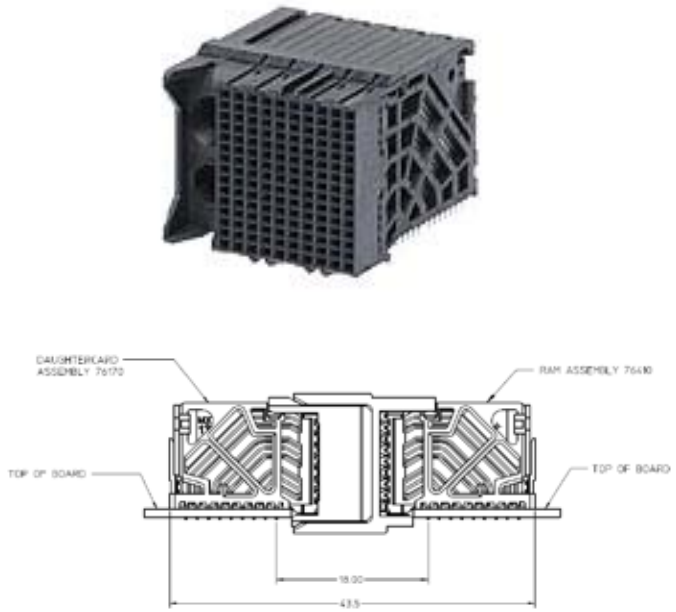

Figure 14 Picture of backplane connectors

The connectors are 100Ω differential and designed to meet 25 Gbps.

5.2 Clock

The interface to the HMC is a synchronous interface. This means that both sides of the interface have the same clock source. For this purpose the FPGA's on UniBoard2 have a reference clock from the black plane side. By using a central clock source for HEM, all FPGA's can access one HMC device and all HMC devices can access one FPGA. With a reference clock of 125MHz (table 38 HMC-15G-SR datasheet) the links of the HMC can reach data rates up to 10Gbps, 12.5Gbps and 15Gbps. The specification for the reference clock is shown in Table 10.

Table 10 Reference clock specifications

Frequency	125MHz
rise/hall time	< 640ps
phase noise (1MHz-1GHz)	<-126dBc/Hz
duty cycle	45 and 55%
drift	<100ppm
Type	Differential (LVDS)
Coupling	AC (100nF)
Amplitude	200-800mV
Nof clocks	4 (FPGA's) 4 MHC's

For the Xo a Silicon Lab 530FA device (same as used on UniBoard2) with a frequency of 125MHz will be used (530FA125M000DG).

Options for the buffers are shown in Table 11.

Table 11 Clock buffer selection

Part	Manf	Outputs	Output level	Remark
854S006I	IDT	6	LVDS	Used on UniBoard^2
8SLVD1208I	IDT	8	LVDS	

5.3 Power Entry

To optimize design reuse the same input power structure with a PIM and 12V brick as used on UniBoard^2 will be used on HEM.

5.3.1 Power consumption

An estimation of the power consumption of the different parts of HEM is shown in Table 12.

Table 12 HEM Power estimation

Power	Power single item	#	Power total	effeciency	Power
HMC	20W	4	80W	85%	94W
QSFP	3.5W	6	21W	85%	24W
Clock	10W	1	10W	85%	12W
Other					10W
Total					140W

5.4 Test strategy

5.4.1.1 Boundary scan

The board will be equipped with a boundary scan interface. Via this interface access will be provided to all boundary scan able devices (HMC devices). This will enable testing the HEM in combination with UniBoard^2. A Lattice FPGA on HEM will be used to implement a scanbridge, just like UniBoard^2 [3].

A 10 pins JTAG connector (with Altera pinning) will be placed on the edge of the HEM, this will ease debugging of firmware. The first four TAPs of the bridge on HEM will be connected to the HMC devices. The JTAG connection for UniBoard will be placed in series with the bridge on HEM. In this way a boundary scan test can be done with the combination HEM / UniBoard2.

5.5 PCB

For HEM the same PCB build-up as used for UniBoard^2 will be used, see Figure 15.

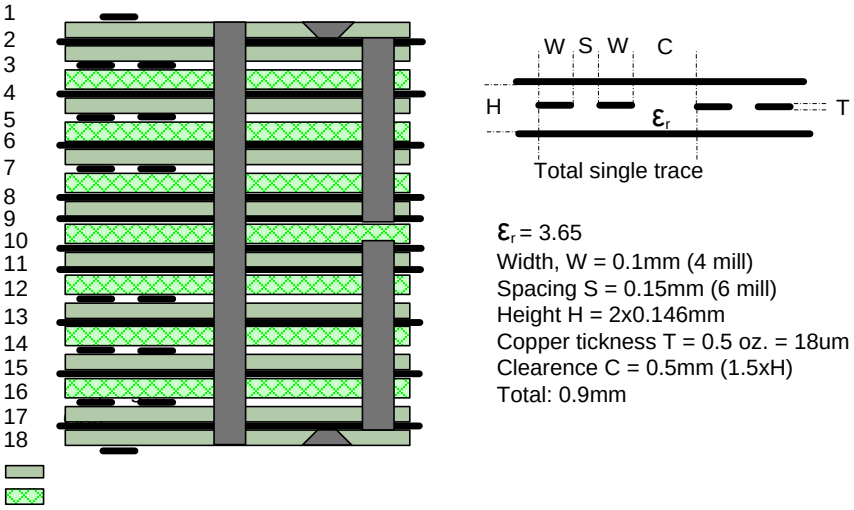

Figure 15 Layer stack

With this build-up all high-speed traces are embed between ground planes. The material used for the PCB is Megtron-6

5.5.1 Hyperlynx Linesim

To get an estimation of the transceiver specification, a high-speed simulation with HyperLynx has been done. In Figure 16 the schematic of the simulation is shown.

Design File: FPGA-HMC.ffs
HyperLynx LineSim V8.1

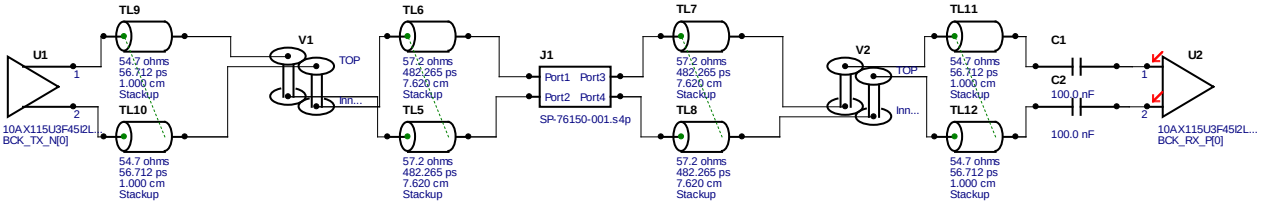

Figure 16 Schematic for line simulation

As there is no model currently available for the high-speed interface of the HMC, on both sides of the link the Altera Transceiver module (10AX115U3F45I2LG_unb2_incl_TRC.ibis) is used. The simulation results are shown in Figure 17.

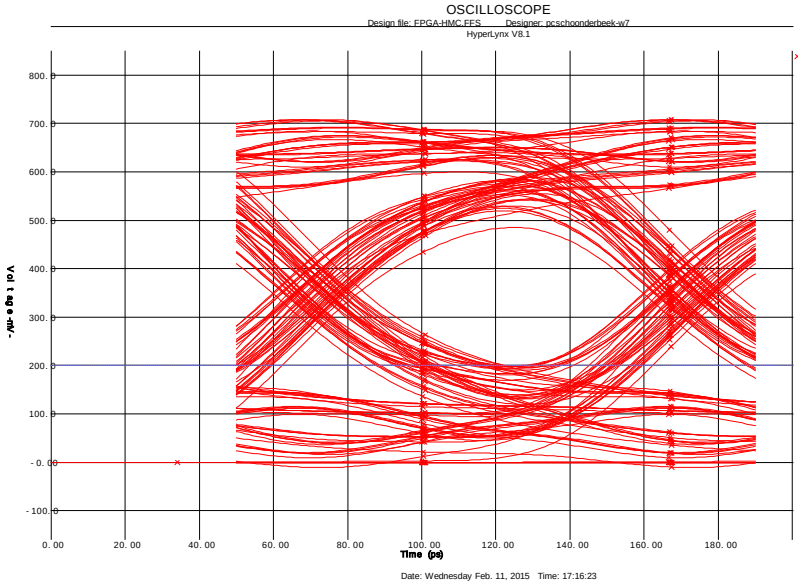

Figure 17 Simulation results

Conclusion:
From this simulation it can be seen that an open eye is open. It should be possible to make an reliable interface from the FPGA, via the backplane connector to the HMC.

6 Green

To make steps towards green processing the following precautions will be taken:

- RoSH compliant
- Reduce the use of tantalum components
- Use efficient power supplies (this reduces heat and less power required for cooling)
- Enable shutdown of power supplies (if memory is not used, switch off the power)
- Liquid cooling
- Remove functionality what is not really needed (this also reduces debug time)

- Use low RDSon FET's for switching

7 Final Design

In Figure 18 the final block diagram of the HEM is shown.

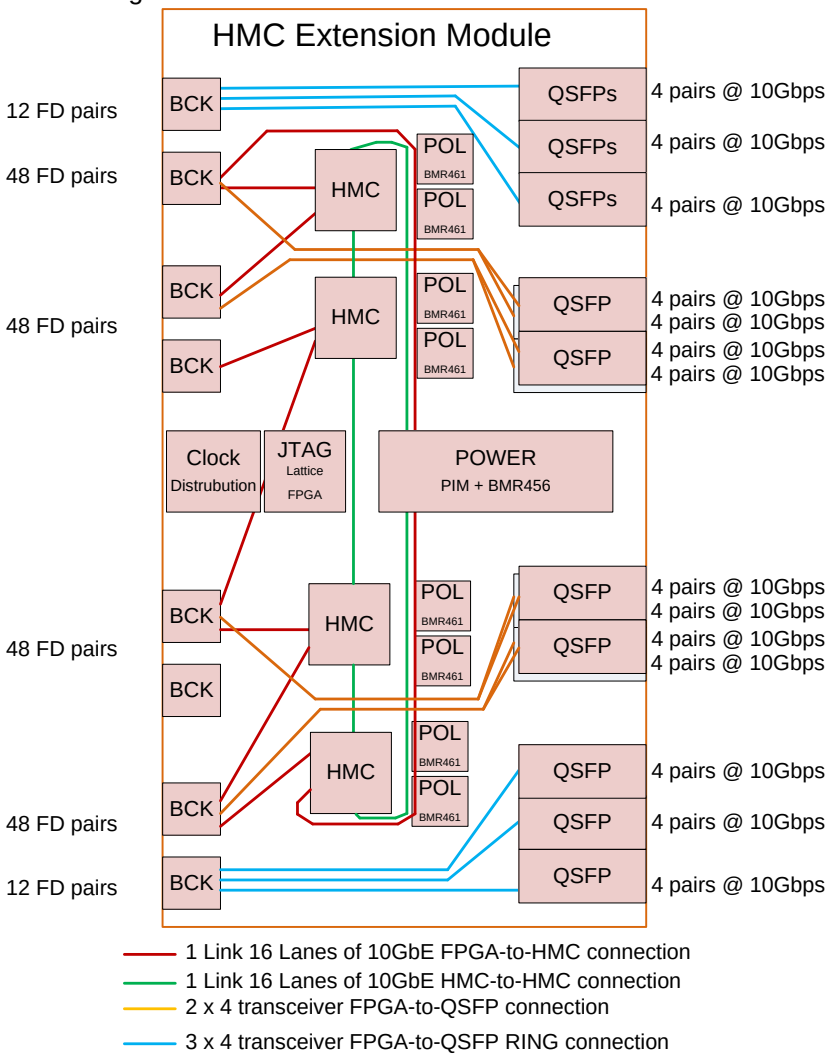

Figure 18 Final Block Diagram

With the HMC connections as shown in Figure 18 we can test:

- Write one FPGA read other FPGA
- Read and write though HMC's (by using HMC-HMC interconnections. Up-to 3 HMC deep.
- Process-store-process

7.1 Link Power

The power of a HMC link can be controller via the use an I2C expenders. On address 0x20h of I2C interface 0 (BCK_x_SDA0 and BCK_x_SCL0) and SX1508 is placed. In Table 13 the function of the 8 GPIO pins is shown.

Table 13 Link Power IO-expanders

GPIO	FPGA 0	FPGA 1	FPGA 2	FPGA 3
0	RXPS HMC 0 Link 0	RXPS HMC 0 Link 2	RXPS HMC 1 Link 2	RXPS HMC 2 Link 2
1	Not used	Not used	Not used	RXPS HMC 3 Link 1
2	RXPS HMC 0 Link 1	RXPS HMC 1 Link 1	RXPS HMC 2 Link 1	RXPS HMC 3 Link 2
3	Not used	Not used	Not used	Not used
4	TXPS HMC 0 Link 0	TXPS HMC 0 Link 2	TXPS HMC 1 Link 2	TXPS HMC 2 Link 2
5	Not used	Not used	Not used	TXPS HMC 3 Link 1
6	TXPS HMC 0 Link 1	TXPS HMC 1 Link 1	TXPS HMC 2 Link 1	TXPS HMC 3 Link 2
7	Not used	Not used	Not used	Not used

Default the RXPS are set high to enable the links to the FPGA's. To disable the link, the GPIO register has to be set in the correct mode by writing to register 0x0Fh to 0x07h and set the corresponding GPIO in register 0x08h

7.2 Power Information

For the local power of the HMC (core and transceiver powers) the Ericsson POL are used. These digital POLs can have readout and some adjust option. The POLs are connected to interface 0 (BCK_x_SDA0 and BCK_x_SCL0). At address 0x01h the Core supply is connected, and on 0x0Dh the Transceiver power supply.

7.3 Ring LEDs

To control the LEDs on the cage an I2C IO Expander is placed on HEM. The expander is placed on 0x21h.

Table 14 IO Expander Cage LEDs

GPIO	Function	GPIO	Function
0	Red Cage 0	4	Ref Cage 2
1	Red Cage 1	5	Not used
2	Green Cage 0	6	Green Cage 2
3	Green Cage 1	7	Not used

The green LEDs are placed on blinking capable IO pins of the expander.

7.4 I2C overview

A number of devices are connected to the first I2C interface of the FPGA. In Table 15 an overview of the connections are shown.

Table 15 Connection to I2C interface 0

Address	Part	Use	See
0x01h	POL HEM Core power	Readout and adjust core power supply	Section 7.2
0x0Dh	POL HEM Transceiver	Readout and adjust tranceiver power supply	Section 7.2
0x20h	IO expander Links Power	Control ON/OFF of HMC links	Section 0
0x08h-0x0Ah	HMC control address	Access HMC configuration	Reff. [4]
0xA0h	QSFP cage	Control and readout link status	[9]
0x21h	IO expander link LEDs	Set the LED on the front panel	Section 7.3