Bordeaux 12-10-2010

Uniboard face to face meeting

Hardware tests: Schoonderbeek

Since the last meeting we built a board, and tested most of it

PCB errors and problems in the PCB design.
Due to wrong shapes used in PCB design there were some possibilities of opens, or "tombstone" components (resistors that stand upright).

The design is based on a block for each FPGA that is replicated and rotated between the front node and back nodes.

Question: Why FPGAS are not connectorized?
FPGAS are not connectorized because of signal integrity and also for space consideration

Boundary scan tests performed. There is a JTAG boundary scan connector on one side of the board that can access all FPGAS and transceivers pins. This has been used for pin-to-pin interconnection integrity.

DDR3 were tested up to 1033 MT/s
Inner interconnect mesh tested up to 6.5 Gb/s
IO Interface tested up to 10 G with optical interface

Problems:
The DDR3 power supply cannot deliver the max current, problems when all DDR3 modules are active.

Output power of the transceiver marginally too low for copper connection. Timing is very good, but power may cause problems with long cables

One swapped signal (Tx <-> Rx) in the third 10G link of each front node chip.

The Ethernet switch has problems with memory, only a flat switch mode was possible, more tests needed

Memory:

Problems with DDR interconnect, some lines are interrupted.

Eye pattern for data is good, soldering is good (X-ray scan)
Boundary scan testing was performed using a dummy "loop-back" memory chip, showing missing connections

Removing one FPGA reveals no solder problems but open lines in the PCB
This happened even if designs rules were met, no critical design process used, and the board tested at production. We must understand why this happened to prevent it in future design.

Assembler and PCB manufacturer are different companies, it is not clear if it is a production or assembly problem. We have only one card produced now.

An XGB daughter board has been designed. It interfaces with the backplane, carrying out CX4 connectors to a box backplane

A container box has also been designed, for single Uniboard applications.
It has a backplane on the backside, 48V power input, fans on the right side.

Documents are available on the wiki page

Future projects:
For APERTIF
A basic system with 4 UB and 8 ADC boards (ADU), 1 power/clock board in a single box with 1 backplane.
Each ADC board has 4 dual ADCs, total 8 input channels per Uniboard

Change the transceivers (Vitesse transceiver used now) for better copper connections.

J. Hargreaves.
Uniboard testing

For testing the FGA we used a system with NIOSII uP and a JTAG UART,
built with the Altera SOPC builder software.

The SOPC configuration contains a peripheral used to reprogram the
configuration FLASH, in order to download a new personality. This will
be the normal way to reprogram the board, accessing the NIOS system via the
1G Ethernet control link.

DDR
The test reads and writes pseudo-random data in sequential and data mask modes,
exercising all addresses. 1 and 4G DDR modules were tested at 800
and 1066 MT/s (supported standard DDR3 frequencies in the Altera DDR IP).
We were able to increase the clock speed of this latter up to 20%, on good
memory modules.

Problem with the 1.5V supply. The current sense resistor was not set
correctly due to stray resistances.

One DDR module failed, due to 4 bad connections on the board (see previous
presentation).

Front node 10G ports
Port 3 of each chip has TX and RX lines swapped, other lines tested
connecting one port of a chip to the same port of a second chip, with
1m and 10m optical cables. Tests OK with 10m cables, some errors with
1m cables on port 2.
Tested also 2m and 5m passive copper cables. 2m OK on ports 0-1,
not on port 2.

Need to tune the analog parameters for the link to achieve better results

Fast serial interconnect mesh.

Tested only the 12 full-featured transceivers (3 links between each
FN-BN pair). The maximum nominal speed for these links is 8.5Gbps but
tests gave good results up to 6.25, and marginal at 7Gbps.
We used the link hard block for 8/10B coding and alignment.
The fourth link in each interconnection could in principle be used
with general logic for coding/alignment, but this would lower the
max frequency for all links in the interconnection.

There are compiling problems using all the transceivers in a
design. Discussion with Altera going on to understand the problem

The back node 10G links were tested with the backplane and CX4 cables,
interconnecting different chips.
Again 3 links per chip are good, the fourth is "no hard IP".

ADC:
We tested using 4 LOFAR receiver units per BN, at 200 MHz, 8 bit.
Sent counter data, no real data. Higher speed possible when APERTIF
units will be available

1Gb Ethernet
We tested it using Altera 10-100-1000MB Ethernet IP
Tested only pseudo data between each FPGA, using the internal switch,
no test done using the 4 RJ45 connectors.

Watchdog interrupt (WDI): external component that resets the FPGA (one
for each FPGA) if it is not reset within a predefined interval. Used to
prevent software infinite loops or other "deadlock" situations.

I2C sensors: FPGA temperature sensor works as expected, 1GbE and voltage
controller temperature sensors still have problems

Development scripts
Each design has a set of generation scripts that generate all files
unb_sopc<design> -> generates SOPC files
unb_app<design><app-main>

In unb_common there are some useful files used for all designs
TCL common files for FPGA pinout
unb_node_ctrl.vhd for basic FPGA clock, reset, watchdog
unb_system_info.vhd for FPGA ID version

Test are performed in parallel on different portions of the board.
We use a Uniboard reservation page to share the board

Future plans:
We must prepare a general test design with all tests inside.
This must also contain some DSP dummy work, to provide realistic power consumption and heat the FPGA.

Most of the test-bench is written in VHDL, with some software to make it easy to control.

Comments:
We use both Linux and Windows. Take care the scripts do not contain system specific dependencies

Problems with IP licensing?
10G license is shared in the project (3 licenses available).
DDR design should be included with NIOS.

Schoonderbeek

Uniboard communication software

You need registers to control the board.
Ethernet connection cannot directly address registers, so a NIOSII CPU is added to the system.
A small server program on the NIOS CPU allows an external client to access the registers using a custom protocol.

The server processes UDP packets. UDP is simple, stateless, fast.
But is unreliable, can skip packets without noticing.

UDP command packet format is composed of a number of 32 bit words with the general structure:
- sequence number
- opcode
- Number of arguments,
- arguments
Few simple commands are used. Examples:
0xffffffff is a "wait for next PPS" command
0x6 download personality
0x1 + n + start address: Write N words to N consecutive address

The IP/UDP interface must support ARP, for standard IPv4 addressing

The server must fit in 64K memory. This is the main reason why the standard TCP/IP protocol has not been used.

For client development, a generic low level library has been written.
The library must:
- support multi-cast
- do not freeze if some FPGAs do not respond properly, but control correctly the remaining ones
- support command check/retransmission
- support odd size fields (not byte or words), and fields in part of a register

Which language to use?
C requires a lot of effort to be safe
C++ with template is good, safe, but do not bind well with other languages
Python probably is the best, both safe and easy to interface with other languages

Erlang is quite good for implementation of soft real time systems. It is
- Exceptional for binary data - designed for that
- Very safe and expressive.
- Very bad for text processing

Links easily with MySQL
Used together with JSON (lightweight data interchange format)

An interesting concept used in the library is represented by ports: mechanism to start an external process and communicating with it with one of the available protocols

Erlang client library can support all this, managing packets,

assembling packets to be sent and disassembling replies, or
assembling commands in a single packet and disassembling the associated
replies

Can be used interactively with Erlang interpreter, for debug

Useful links:
www.erlang.org
www.json.org
erlport.org (Python base class for "port")

The library check for lost commands and/or replies(up to 3 retries)
Client waits for a reply and sends again the command after 0.5s.
NIOS caches the commands and the responses, and if a command is
duplicated (same seq. number, thus already processed) resends the reply.

The system will provide the "fpga" library, whit these basic functionalities.
Every application should provide a specific application library, with
medium level functions (start an integration, specify parameters, etc.)

Questions:
Why UDP and not TCP/IP?
UDP, being stateless, is less prone to hang up. Many TCP/IP small
clients can accept only one connection at a time, and if this hangs up
you cannot break in and return to a safe state.

System latency: how fast is the system? Erlang is soft real time,
quite fast but without strict time limits.

Module design and reuse
Kooistra

We need plug-and-play VHDL code, both to be reused in different
designs, and to be able to port it to future technologies.
For example Stratix V is already here and we do not want to rewrite everything

Casual reuse (copy and paste form files) is good, but still requires a lot
of work and good inside knowledge of the modules

Reuse does not come for free, you have to
-design with reuse in time,
-Provide a reference design together with the module,
-Provide source files including C code, project, scripts
-Code using consistent coding styles
-Document everything

Terminology:
A design is an entity that runs on a FPGA (top level and everything below)
A module is a complex function or library of functions usable in
different situations
A component is a low level design entity.

Uniboard code is organized in several groups of modules, organized in the
directory tree under the "modules" hierarchy. Some directories are:
common: simple generic components like counters, memories, FIFO...
unb_common: modules for Uniboard general functionalities like clock/reset
generation, Watchdog timer,
LOFAR: several components for on-board peripherals (I2C, MDIO, Ethernet...)
udp_packetizier
dp: streaming components, mux, latency adapters

Example: the Ethernet module

It is available in the modules/tse directory, and interfaces with
the Altera Triple Speed Ethernet IP.
It has two Avalon (standard Altera) interfaces, using the two
available basic protocols.
A streaming (point-to-point, with minimum control signals) ST interface
provides a channel to feed UDP packets from/to the module.
A memory mapped (microprocessor like) MM interface is used for control
and as a second mechanism to read/write packets

VHDL "record" types are used to represent signals in the interfaces
Constants are used for data bus widths

VHDL wrapper are used to encapsulate in a consistent way each
IP from a specific vendor/version. Everyone sees the IP in a consistent way
and only the wrapper needs to be changed when the vendor/chip is changed.

Test-bench. I use another triple speed Ethernet IP to generate the stimulus
in the simulator test-bench, connecting it to the modulus under test with
signals that include a VHDL "AFTER" delay

The software provides some public functions to get/send packets
(files avs_eth.h, avs_eth.c avs_eth_regs.h) plus a main() function that
can be used either in the simulation test-bench or as an example.

A test-bench has been provided with a ring of 8 eth instances, simulated
in Modelsim and in the real hardware.

Documentation includes:
Introduction, interface, design, implementation, application, verification

In general if a module has a MM interface, it also needs software function
modules

Modules must have the SOPC builder interface: a TCL file specifying
parameters and port mapping. In this way it is possible to build a system
using the modules in the graphic Sopc-builder environment, parameterizing
each module and connecting them together using the ST Avalon interfaces

DSP functions can be modularized in SOPC builder

Questions:

Q: How to synchronize notations and styles?
A: Try to keep a consistent notation within each module, but synchronizing
the notation and style across a big project is really difficult.
Keep the interface as standard as possible, the inside coding style
is not so important.

Q: Doing things really general can bloat the code. How this works here?
A: I need a lot of generality for proper testing. Adding generality may
be complex, e.g. the ST port is not really needed in the Triple Speed
Ethernet module, but it may comes out useful e.g. if one recycles the
same module for 10G Ethernet, where the ST port can be used to feed in/out
the astronomic data being processed.

Q: It is necessary to define protocols for efficient reuse. Which protocols?
A: Use Avalon ST definitions. They are quite general and are the standard
used here.

Hargreaves: Correlator design

General architecture:
Station data comes in through 10G links
8 stations per FN, 2 pol, 8 bit, 64 MHz Bandwidth
8 stations per FN, 2 pol, 8(4)bits, 128 MHz Bandwidth (complex)
BN process 32 stations, 2 pol, 32 MHz bandwidth, total of 2112 cross
products/chip

To prove the concept we designed an autocorrelation processing system
4x16 MHz, 2 pol., 2 bits, + validity bit

VDIF data frames of up to 8K bytes
Polyphase Filter Bank with 4k data points
Back-node uses 4 x 1k data points

The MAC processes all the stations in 4 passes, with 132 MAC cells

Correlation products are exported from the BN using the 10G ports.

Correlation is done with 9 bits
No aux data is carried with the data stream. Control system knows
which aux data belong to each data frame

Question: Processing is done on real or complex data samples?
Correlator expects real data, that is transformed to complex in the
station processor. Complex data could be used, but this has
to be rearranged. VDIF supports complex data but this possibility has not been
explored.

Uniboard EXBOX - Des Small

RFI mitigation

Work done on RFI detectors. Data affected is then flagged/deleted.

Detectors implemented:
Robust radar detector

Basically a pulse detector
Can be done after the ADC, for narrow pulses,
after channelization or at the dedispersion level

Blind Giant pulsar detector
Remove signals with zero Dispersion Measure or with narrow spectral lines

Cyclostationarity

Telecommunication signals are often present in data, and mimic real
astronomic signals.

RFI classical mitigation:
Identify stronger eigenvectors and remove them from the signal.
But if RFI is comparable to source, you remove both (or none)

If RFI has some cyclostationary property, you can remove it.

To identify cyclostationary signals, compute Frobius norm of the
correlation matrix then FFT over time

Uniboard applications at KVN

At east 3 boards needed

---
multi-Uniboard applications

To connect boards together one can use either switches on the 10G links,
or backplanes, interconnecting in a rigid way the back-node fast links.

APERTIF (beamformer/correlator) for WSRT and LOFAR applications are good
applications for a multi-UB system.

One can exploit the independence of sub-bands and beams, so they may be
processed at different boards

On the input side, typical applications are filterbanks and digital receivers

In the Uniboard the input and output bandwidths are comparable.
Depending from applications, this may be true or not.

Architecture is "almost fully connected". FN and BN are connected in
such a way that each FN is connected to each BN, but not FN-FN or
BN-BN interconnect exists.

Connecting more boards, a similar architecture apply. For example one
can connect all Front Node FPGA O in different boards together via a switch,
or all Back Node FPGA 0 using the 4 BN fast serial interconnect in a
backplane, but typically without interconnects between different rows
of FN or BN chips.

With ADC present in the system, ADC parallel inputs can be used to connect
up to 4 ADCs to each BN FPGA, leaving the BN serial interconnects for
UB-UB communication.

This architecture is planned for APERTIF. The antenna system at each antenna
is composed of two sub-racks (one or each polarization), with 64 inputs
each (64 ADC), 4 processing Uniboards. Output of the rack are beams,
and each beam is then correlated.

11K visibilities (37*(24*25/2)) processed in racks of 3 Uniboards (1
empty slot)

In total we need 96 Uniboards for the beamformer and 24 for the correlator.

LOFAR stations.
576 dual polarization antennas, 17.5 MHz bandwidth each
A total of 12 UB are needed. More powerful UB can allow for wider bandwidth.

So the idea is to have single UB boxes, for single apps, and sub-racks with typically 4 boards. To build larger systems, these sub-racks are further interconnected using switches.

Q. When a production run could be arranged?
Szmoru: End of next year could be a good date. Caution in respinning the board with Stratix5, it would be a never ending race.

Q: Good to have a total number of required boards for all applications and institutes, in order to have realistic numbers and quotations.

Q: How much does it cost?
A: Now in production is about 50K Euros. Plus the cost of the IP involved. Altera IP costs if you develop applications, not if you simply use them. 10 Gb Ethernet costs some 15K, Triple speed Ethernet also costs.

Q: What is the power consumption?
Rough estimate is 200-250W but we have never really used all chip resources.

Q: Speed for ADC port is enough?
A: Altera specs are 800 MHz for LVDS. We plan to use the whole band of one LVDS port for one ADC at 800 MS/s, 8 bit.

New fast ADC at Bordeaux Observatory

Bordeaux has experience with the ALMA digitizer: 4 GS/s, 3 bit, 1:16 output demultiplexer. Output is 48 bits (3x16) at 250 MS/s.

Current work on two designs :
- Commercial components (from I2V)
Board with 2x10 bit, 5 GS/s 3 GHz bandwidth. Final goal is to interleave the two ADCs to increase the sampling rate and use the full input bandwidth.
- Collaboration with UdB to design a 65 nm technology ADC (3 bit initially, 6 bit in the final version), plus a Track&Hold (8 GHz -3 dB), and an internal 1:4 demux

To further demux the ADC outputs, we plan to use a small FPGA with minimum functionality, basically just demux.
Communication from ASIC to FPGA uses a fast serial link, 1.6 GHz.
Fast parallel or serial protocol is used between FPGA and DSP
Available links are 4x32 LVDS; 16 x fast serial, 10G Ethernet

First board (2 commercial ADCs): we have a total of 2x4x10 bit @1.25 GHz
A FPGA maps this to 24 TX lines, @100 Gb/s

The ASIC ADC provides 4 serial links x 3 bits @ 2GHz. It is possible to lower the rate using a commercial 1:2 demux.

Road map:
Run of the ASIC design for end 2010
Q1-2011 test with prototype 10 bit 5 GS/s
Q2-2011 test of the ASIC chip
End 2011 - run for Track & Hold
End 2012 - run for 6 bit 8 GS/s ADC

Q: Are there already 30 GS/s ADCs on the market? A spinoff of a German University claims to be able to sell them
A: There are high rate ADCs but not with high enough bandwidth. You can also get samples of some components, but not real production

Cais: our final goal is to achieve 16 GS 6 bits, and there are no such component on the market

Q: Which threshold spacing do you use?
A: Equispaced thresholds symmetric around zero

Potential interest of Uniboard Daugterboard for ADC

Kenfack - Univ de Paris - Nancay

Nancay ADC chip: 6 bit 3 GS/s flash, bandwidth 0.1-1.5 GHz

Technology: Bipolar Track/Hold, ECL D-latch/comparator and encoder.
Encoding using a Bubble corrector and a Wallace-tree encoder
A random code generator is used to embed the clock in the code.
Noise has been modelled. Equivalent N. of bits is 5.2, with SNR
34.7-33.2 dB at the low-high bandwidth extremes.

Total estimated power is 2.6W. 4mm2 chip area.

Total of 65 comparators (63 + overflow/underflow)
A second chip provides 1:8 demux

Test bench using Virtex6 with 6 GB transceivers
Using the demux, we can feed a Virtex6 using 48 LVDS signals.

Roadmap: Ready in Dec 2010

Now received the chips. Managed to receive data at 2.8 GHz, work on the
GTX receivers at 3 GHz in progress

--
Why a daughter board
Need to bring data from the receiver to the Uniboard platform
A small board can be screened near to the receiver and can communicate to
the Uniboard using a fiber cable.

System is composed of 2 boards interconnected using
Samtec FMC connectors. Connection of 10x40 rows 400 pins.
VITA 57.1 standard

Example application: receive 16 500 MHz inputs and send them using 16
10G links to the UB FN

Q: What is the max frequency of the FMC interface?
A: (Up to 9 GHz) It is an industrial standard, and is basically a BGA array.

Q: Why do not enter by the backplane XGI connectors?
A: We need an optical link. Better to use copper on the output side.

Recommendations and Actions

General comments:

We need more testing

People have not played with the card yet. Difficult to imagine the
problems will arise.

How to write common reusable components? Establish common guidelines.

Useful to have a "base" design with almost empty DSP and all
infrastructures.

Finalize a test-bench design. This could be also useful as a style
example.

More interchange should be useful in order to homogenize styles.

Do we assume that only documentation is code itself? (of course NO)
Agree on a documentation style.

At least what is absolutely needed is port description and port diagram
for every reusable module. No need to go to describe each individual
component, but each reusable module should be documented properly.

Have a sort of "peer review" for the modules in the SVN? Organize a
procedure so that every module in the SVN repository is checked by
other potential users, to verify that documentation is good enough.

Plans to use Uniboard in existing environments. What is the environment
in which the board will be used? What is needed?
What kind of software control system is needed?

13-oct-2010
Shanghai Uniboard

Applications:
- correlator (CVN)
- back-end for 65 meter telescope

CVN: will be a 16 station 16 channels per station, 2-4 GHz BW

65 m telescope: active surface up to 7mm (0.45 efficiency)

Digital back-end
for VLBI: DBBC, 2-4 GHz BW, 4 IF, 32 sub-channels with up to 64 MHz each
Spectroscopy: goal is to have a 32 MHz BW, 50 Hz/ch
Continuum and pulsar observations

Goal is to have everything working for 2012
2015 for the high bandwidth system

Questions from SHAO:
Possibility to use more Uniboards in Shanghai
Status of UB-1
Possibility to share IP

Arpad:
Status of the hardware: minor problems to be solved before going to production. Probably they will be solved in the next month, production of prototypes before January. Then we will have one board per institute.

Applications work is in progress both for the system and for the individual applications.

At the end of next year a new version Uniboard will be started

There are three licenses for Altera IP shared among institutes.
Our IP is open source.

At the end of the project we will consider a real production run both for boards and for cabinets for multi-board systems.

Baudry: At the end of 2011 there will be a call for production and a production run can be organized.

How many boards required?
SHAO: 30-40 boards near the end of next year

Status of the project:
7 engineers working on this project since next month
Started to buy Mentor Graphics software for modelling

Future Uniboard

Current Uniboard:
general board, that can be connected together using network switches
Infiniband switches with up to 128 ports exist and can perform as corner turners

Uniboard2
Follow-up project, currently in selection phase, supported by Radionet
Start in 2012, end in 2015/16
Using 28 nm process, or even next one (depending on start date)
But note that even Stratix5 is not currently available beyond engineering samples. Probably available in Q1, 2011
Or we can go to Xilinx chips, even if we gained a lot of experience with Altera

Green computing, reduce power consumption:
- effects of Hardcopy and partial Hardcopy, algorithm tuning for power reduction

A lot of design work to report, simulate, estimate power consumption of Uniboard 1

Timing:
JRAs last 3 years. UB1 ends at end of 2011. So it would be good to have the project starting at beginning 2012, not to loose people. But it would be good to start one year later to better evaluate chip-design options.

Better to change name. Using Uniboard 2, 3 etc is not politically sound.

The UB1 project will anyway continue after 2012, adding partners and collaborations outside EC money.

Q: Open model? The repository will be completely open?
Arpad: Open to anyone wishing to collaborate. Will not be open worldwide, like Casper

Q: Other people in other Radionet JRAs are studying ADCs, but information is not shared

Q: UB2 will be a complete redesign or based on UB1?
A: Probably UB2 will be somewhat different. But it will be ready around 2016.

Casper roadmap. Now they are going to Virtex 6, but always one chip per board. Future plans are always one chip per board, with optical link.