Internal Technical Report
CPU-GPU based DIGITAL Backend

S. Harshavardhan Reddy & Irappa M. Halagali

Ver. 1.0 , 05/03/2014.

Index

1. INTRODUCTION
2. BLOCK DIAGRAM
3. SPECIFICATIONS
   a. uGMRT
   b. GWB-II
   c. iADC
   d. ROACH
   e. GPU Board
   f. HOST Machines
   g. Infiniband Switch
4. DESIGN DETAILS
   a. Digitisation and Packetising in FPGA
   b. Acquisition and Shared Memory buffering
   c. Time slicing and data sharing using Infiniband
   d. Correlation in GPU
   e. Beamformer
   f. Software Model
   g. Online control, visibility and beam data recording
   h. Computation and IO requirements
5. TESTS and RESULTS
   a. LAB :
   b. Antenna :
6. CONCLUSIONS
   Appendix - 1 SOP
   Appendix - 2 SOFTWARE PROGRAMS

1. Introduction :

The GMRT consists of an array of 30 antennas, each of 45 m diameter, spread over a region of 25 km diameter, and operating at 5 different wave bands from 150 MHz to 1450 MHz. The maximum instantaneous operating bandwidth at any frequency band is 32 MHz. Each antenna provides signals in two orthogonal polarizations, which are processed through a heterodyne receiver chain and brought to the central receiver building, where they are converted to baseband signals and fed to the digital back-end consisting of correlator and pulsar receiver. The existing GMRT Software back-end (GSB) is built on software based approach designed from off-the-shelf components, PCI based ADC cards and a Linux cluster of 48 nodes with gigabit inter-node connectivity for real-time data transfer requirements.

GMRT is upgrading to uGMRT and the back-end systems are undergoing major changes to achieve the upgrade system specifications like increased bandwidth of 400MHz, direct processing of RF signals, increased dynamic range, improved channel resolution. The digital backend part of this upgrade was named GWB(GMRT WIDEBAND BACKEND). As part of this upgrade, a version of GWB(GWB-II) has been built and released for users. This is a 4-antenna dual polarization or 8-antenna single polarization correlator developed on CPU-GPUs and FPGA boards.

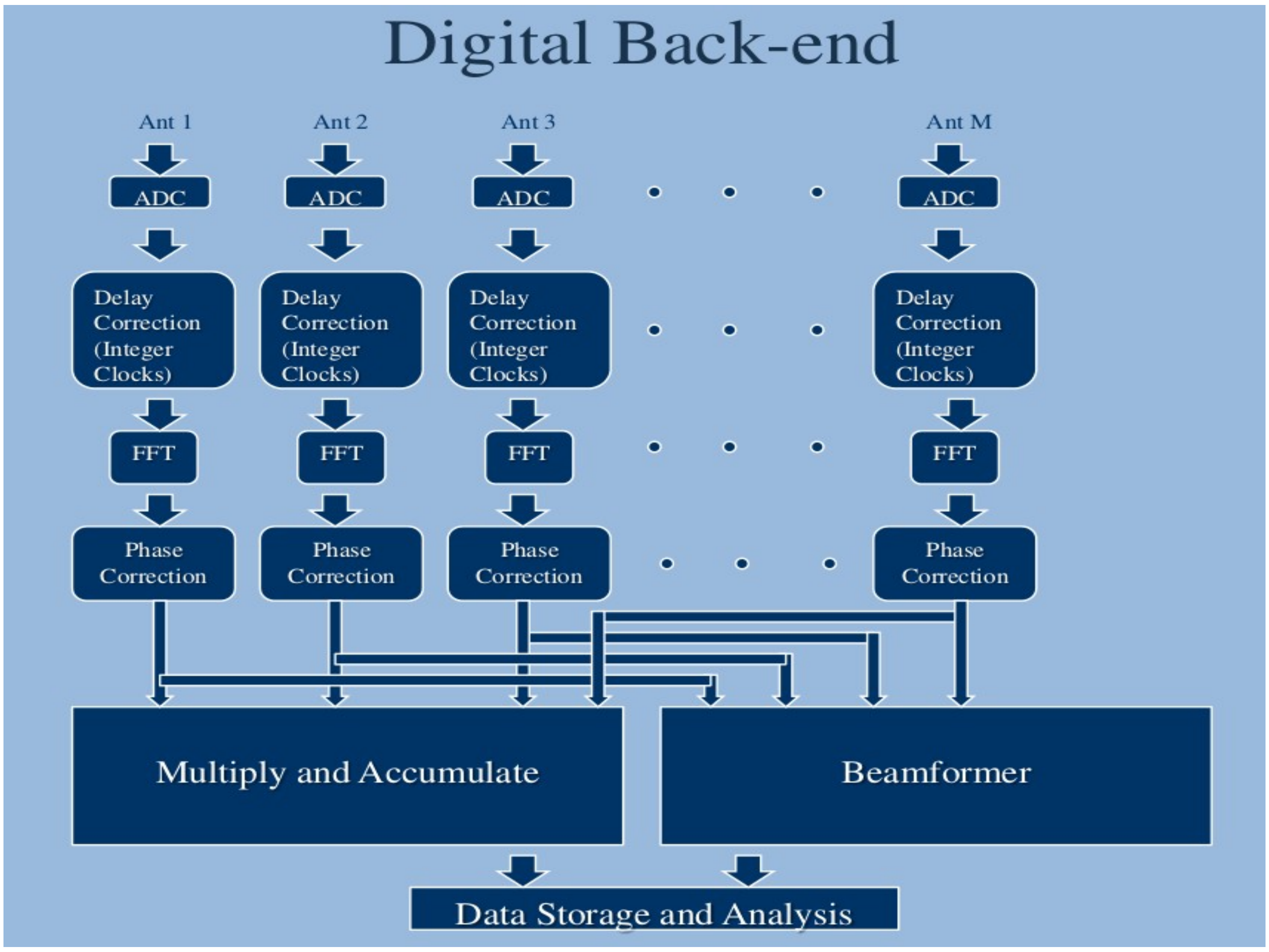
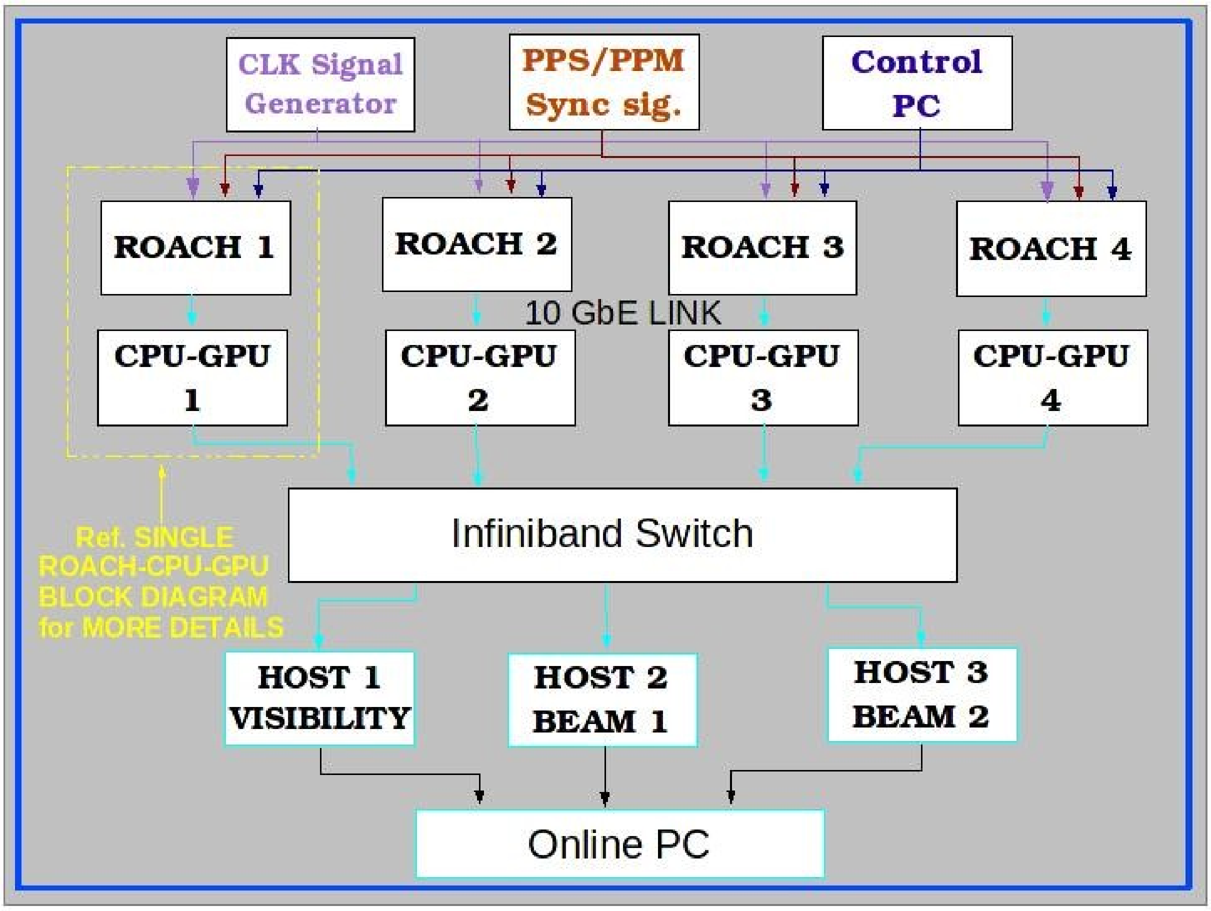


Fig : Digital Back-end

The design is a hybrid one using FPGAs and CPU-GPUs for various processes in the digital back-end chain. FPGAs connected with ADCs perform the digitisation and packetising the data while CPU-GPUs acquire the data, perform correlation and record the visibilities onto a disk for post-processing and analysis.

2. BLOCK DIAGRAM :



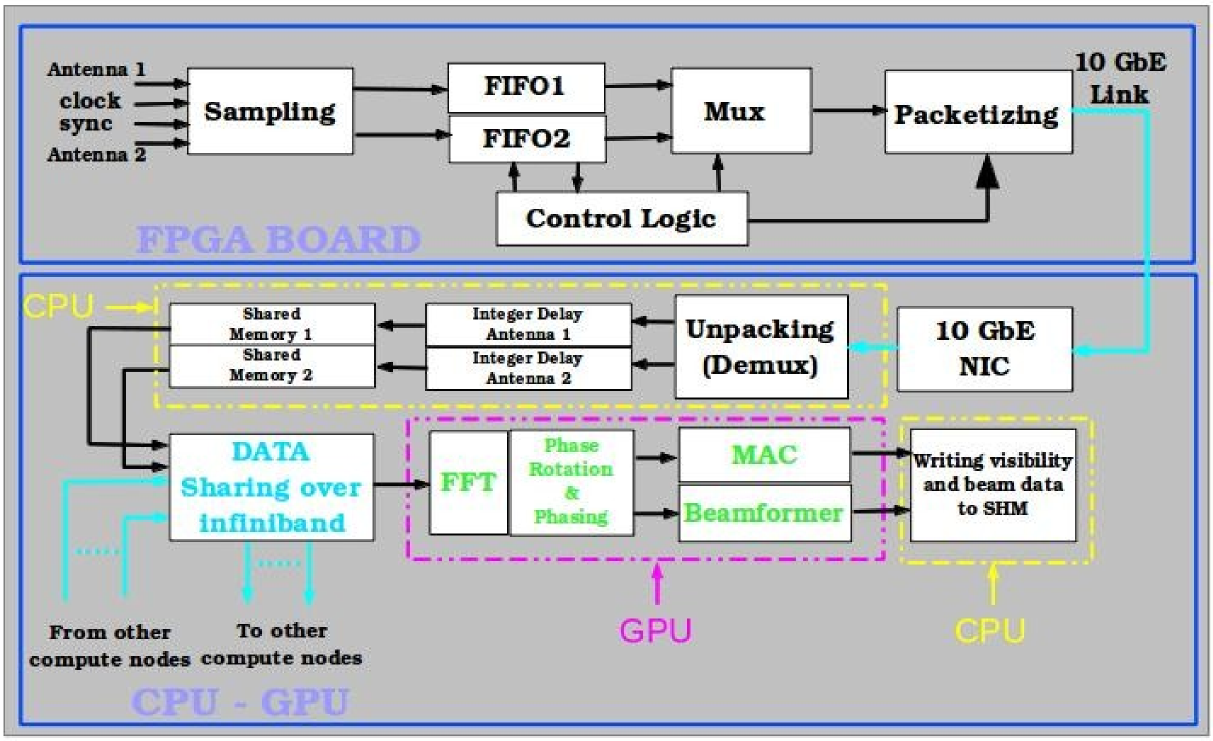


Fig : Data flow on single ROACH and CPU-GPU

Hardware :

1. iADC boards :- 1x Atmel/e2V AT84AD001B 8-bit dual 1Gsps, with clock 0dbm 50Ω 10MHz – 1GHz.
2. ROACH boards:- ROACH stands for Reconfigurable Open Architecture Computing Hardware. The board is a standalone FPGA processing board built around Xilinx Virtex-5.
3. CPU-GPUs with Nvidia GPU (Tesla C2050 or above) with Myricom 10GbE NIC card and Infiniband NIC.
4. Mellanox QDR 18-port Infiniband switch
5. Control PC for ROACH boards configuring and programming
6. PCs with Infiniband NIC as host, IA recording, PA recording nodes
7. Signal generator as clock source for the ADCs.

Software :

1. CASPER MSSGE Toolflow - Matlab-Simulink + System Generator +Xilinx 11.5 EDK
2. Centos Linux 5.6 (Kernel 2.6.18-194.32.1.el5 or higher on X86_64)
3. Python 2.6 on Control PC for ROACH boards
4. Nvidia GPU device drivers + CUDA toolkit on CPU-GPUs
5. OpenMPI 1.4.5 or higher.

Implementation :

The 4-antenna dual polarization correlator is implemented using two ROACH boards each with two ADCs and four CPU-GPUs and an 8-port infiniband switch(40 gbps).

3. SPECIFICATIONS :

a. uGMRT

Number of Stations	:	32
Maximum instantaneous Bandwidth	:	400MHz
Number of spectral channels	:	2048 – 8192
Number of input polarizations	:	2
Full Stokes capability	:	Yes
Dump time	:	min. 128ms
Coarse and fine delay tracking	:	+/- 128 us
Fringe rotation	:	up to 5 Hz
Subarray support	:	Yes

Incoherent and Phased array beams(for pulsar work)

b. GWB : The specifications achieved in the current GWB system

Number of Stations	:	4/8
Maximum instantaneous Bandwidth	:	200MHz
Number of spectral channels	:	2048

Number of input polarizations	:	1 for 8 antennas, 2 for 4 antennas
Full Stokes capability	:	Yes
Dump time	:	min. 671ms
Coarse and fine delay tracking	:	+/- 128 us
Fringe rotation	:	up to 5 Hz
Subarray support	:	No
Incoherent and Phased array beams	:	Yes

c. iADC :

ADC with 1x Atmel/e2V AT84AD001B 8-bit Dual 1Gsps. It needs

- Inputs
  - Clock: 10MHz-1GHz 50Ohm 0dBm
  - Signal: 2 analog inputs.
  - Sync: LVTTL (5V tolerant, (SN65LVDS1)
- Outputs
  - 1x Tyco Z-DOK+ 40 differential pair connector
    - 8x (1x8 or 2x4) 8-bit offset binary data
    - 2x digital clocks to CASPER standard Z-DOK clock pins

d. ROACH (FPGA) Boards :

ROACH (Reconfigurable Open Architecture Computing Hardware) is Vertex 5 FPGA based reconfigurable hardware. The centrepiece of ROACH is a Xilinx Virtex 5 FPGA (either LX110T for logic-intensive applications, or SX95T for DSP-slice-intensive applications). A separate PowerPC runs Linux and is used to control the board (program the FPGA and allow interfacing between the FPGA "software registers/BRAMs/FIFOs" and external devices using Ethernet).

Two quad data rate (QDR) SRAMs provide high-speed, medium-capacity memory (specifically for doing corner-turns), and one DDR2 DIMM provides slower-speed, high-capacity buffer memory for the FPGA. The PowerPC has an independent DDR2 DIMM in order to boot Linux/BORPH.

The two Z-DOK connectors allow ADC, DAC and other interface cards to be attached to the FPGA, in the same manner as the IBOB allowed (with backwards compatibility for the ADC boards used with the IBOB).

Four CX4 connectors provide a total of 40Gbits/sec bandwidth for connecting ROACH boards together, or connecting them to other XAUI/10GbE-capable devices.

100Mbps ethernet port is used for both control and acquisition of the data.

e. GPU Boards : C2050

Architecture	:	Tesla 20-series GPU
Number of cores	:	448
Caches	:	64 KB L1 cache + Shared Memory / 32 cores, 768 KB L2 cache
PCI-e DMA Engines	:	2
Floating Point Peak Performance	:	1030 Gflops (single) 515 Gflops (double)
GPU Memory	:	3 GB
Memory Bandwith	:	144 GB/s (GDDR5)
System I/O	:	PCIe x16 Gen2
Power	:	247 W (max)

f. Host Machines : Dell T7500

Processor	:	Intel Xeon X5550
cpu MHz	:	2.67GHz
cache size	:	8192 KB
No. of cores	:	4 per CPU
No. of CPUs	:	2
Memory	:	12 GB (scalable upto 192 GB)
No. of PCI-e slots	:	5 (one X4, two X8, two X16)
Operating System	:	Redhat Linux Centos 5.6
SMPS	:	1100 watt 85PLUS Power Factor Correcting(PFC) power supply

g. Infiniband switch : Mellanox 18-port switch

–18-port QSFP non-blocking ports

–Aggregate data throughput up to 1.44 Tb/s

–Port-to-port latency < 100ns

–9 virtual lanes: 8 data + 1 management

–Adaptive routing

–Congestion control

–Port mirroring

–48K entry linear forwarding data base

–Supports Passive/Active copper or fiber cables

4. Design details :

a. Digitisation and Packetising in FPGA :

The design is made of Matlab Simulink blocks, CASPER BEE-XPS block set. The ADC is an 8-bit sampler and is connected to the ROACH board through a 40-pin Z-DOK connector. Each ADC can sample two input signals. The sampled data from each channel is stored in the FIFOs. Each FIFO is 8192 bytes in size. The data from FIFOs is then packetised in the 10GbE block using control logic and is sent over 10GbE network. The packet size is 8242 bytes. The packet structure is 42 bytes of UDP packet header, 8 bytes of packet counter, 4096 bytes from channel 1 and 4096 bytes from channel 2. The sync signal to the ADC is a PPS/PPM signal to synchronize the data from many ROACH boards. For 200MHz BW, 8 bits per sample and two input channels, the data is 6.4 gbps.

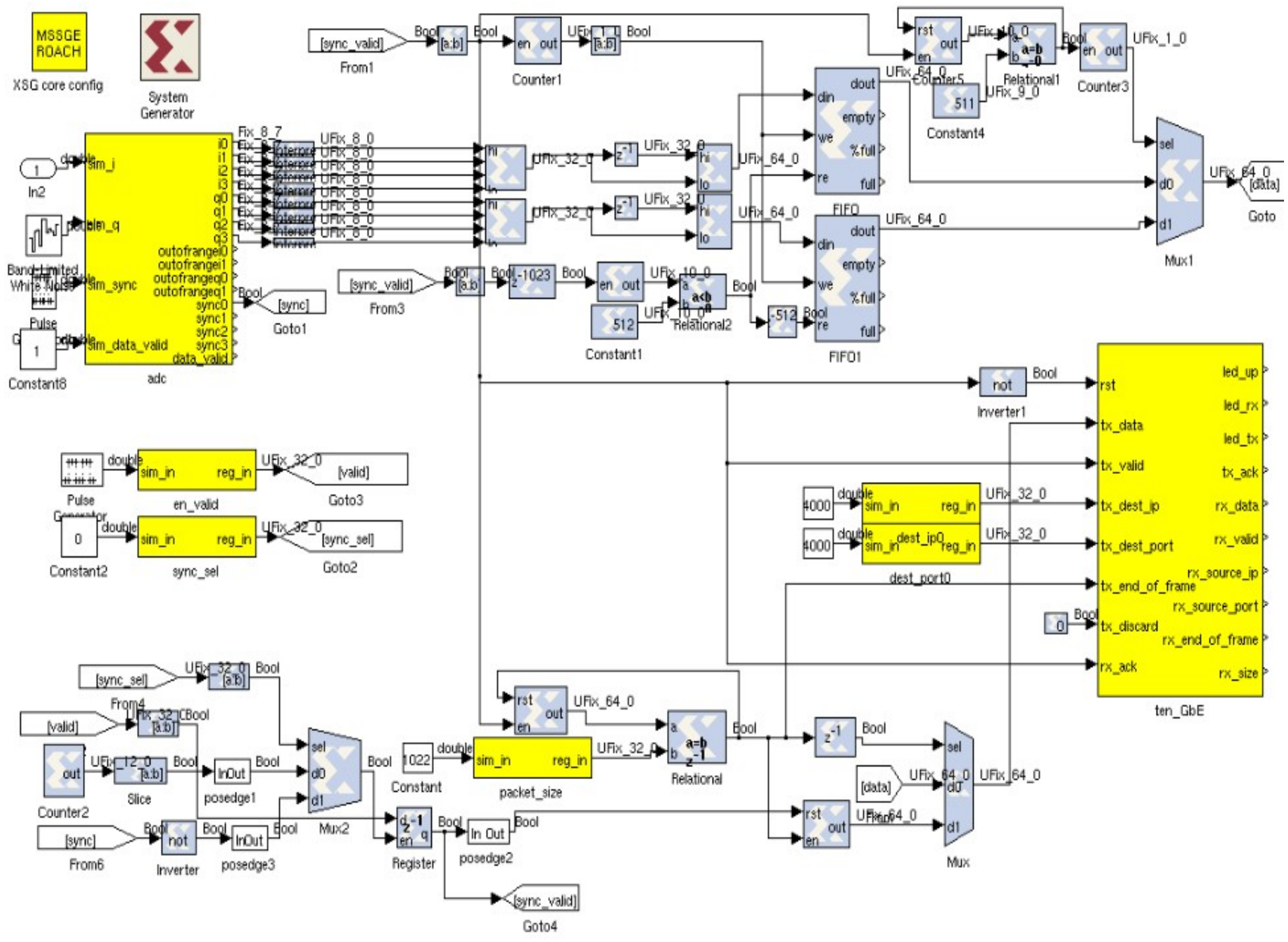


Fig : FPGA design

b. Acquisition and Shared Memory buffering :

The packets are received on the CPU side through the Myricom 10GbE NIC. The data thus received is written into shared memories. Two shared memories are created on each node for each input channel. Each shared memory is of four buffers, each buffer size is 256 MB. The data before writing into the shared memory is corrected for coarse delay. Also the time when the first packet is received in the first acquisition node is noted and shared with other nodes for time synchronization. The nodes are NTP synchronized with the GPS time server. The coarse delay is corrected for every buffer of 256 MB.

c. Time slicing and data sharing using Infiniband :

The data from the shared memories is read and is shared over infiniband network with the other nodes. The shared memory buffer is read and sliced into number of nodes times slices. Node 0 keeps the first slice and sends the second slice to node1, third slice to node2 and so on. At the same time, node 0 receives slice 0 from node1, node 2 and so on. Similarly, node 1 keeps slice 1, sends the remaining slices to corresponding nodes and receives slice 1 from all the other nodes. This happens over all the nodes at the same time. To keep the nodes busy all the time, non-blocking MPI calls are used for this communication.

Algorithm :

```
    step 1 : initialise variable i = 0
    step 2 : Repeat steps 3 to 4 till i = log2(number of compute nodes) - 1
    step 3 : variables wr = exp2(i), cr = exp2(i+1), new_rank = rank%cr, idx = rank/cr, j = 0
    step 4 : Repeat steps 5 to 15 till j = wr – 1
    step 5 : if new_rank < wr perform steps 6 to 9 else perform steps 10 to 13
    step 6 :  dest = new_rank + wr + j
    step 7 : if dest > cr -1 perform step 8 else perform step 9
    step 8 : dest = dest - wr + (cr*idx)
    step 9 : dest = dest + (cr*idx)
    step 10 : dest = new_rank - (wr + j)
    step 11 : if dest < 0 perform step 12 else perfrom step 13
    step 12 : dest = dest + wr + (cr*idx)
    step 13: dest = dest + (cr*idx)
    step 14 : send the data to node ranked dest
    step 15 : receive the data from node ranked dest
```

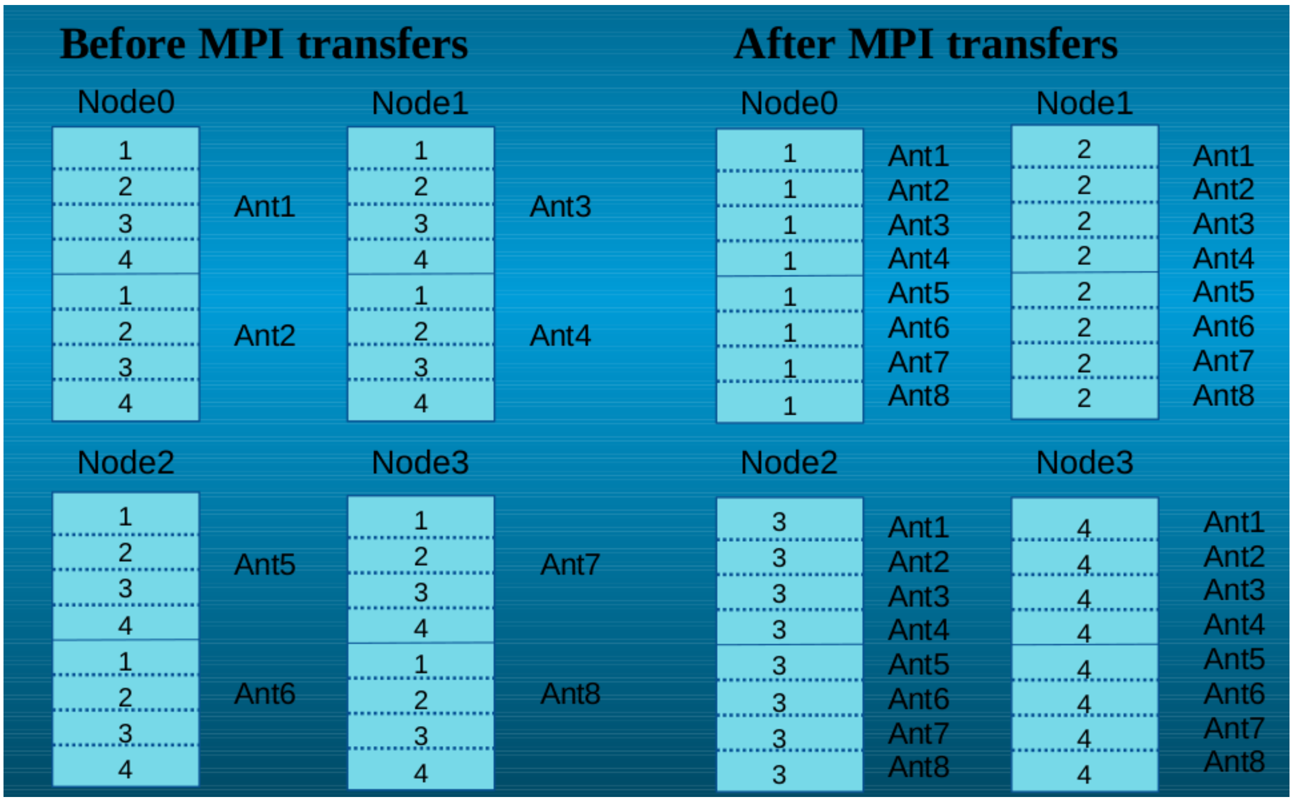


Fig : Data sharing

d. Correlation in GPU :

A slice of data of all antennas in each node is copied to GPU for correlation process. FFT is performed using CUFFT library. The fourier transformed data is fine delay corrected and compensated for fringe rotation. The delay cal function calculates the fstc and fringe values from the starting time which was noted when the first packet was received.

FFT : Real to Complex FFT is performed efficiently by performing Complex to Complex CUFFT and reconstructing the output taking advantage of the symmetry.

For N-point FFT form a complex N/2 sample sequence(even samples are real and odd samples are imaginary).

$$z(n') = fe(n') + jfo(n') \qquad n' = 0..N/2 - 1$$

Take the FFT of z using CUFFT to get

$$Z(k) = FFT_{N/2}(k, z) \qquad k = 0..N/2 - 1$$

Launch a GPU kernel to perform

$$F(k) = \frac{1}{2}(\{Z(k) + Z(N/2 - k)^*\} - je^{-j2\pi k/N}\{Z(k) - Z(N/2 - k)^*\})$$

for k = 0, 1, ----, N/2 - 1

$$F(N/2) = \frac{1}{2}(\{Z(0) + Z(0)^*\} + j\{Z(0) - Z(0)^*\})$$ for k = N/2

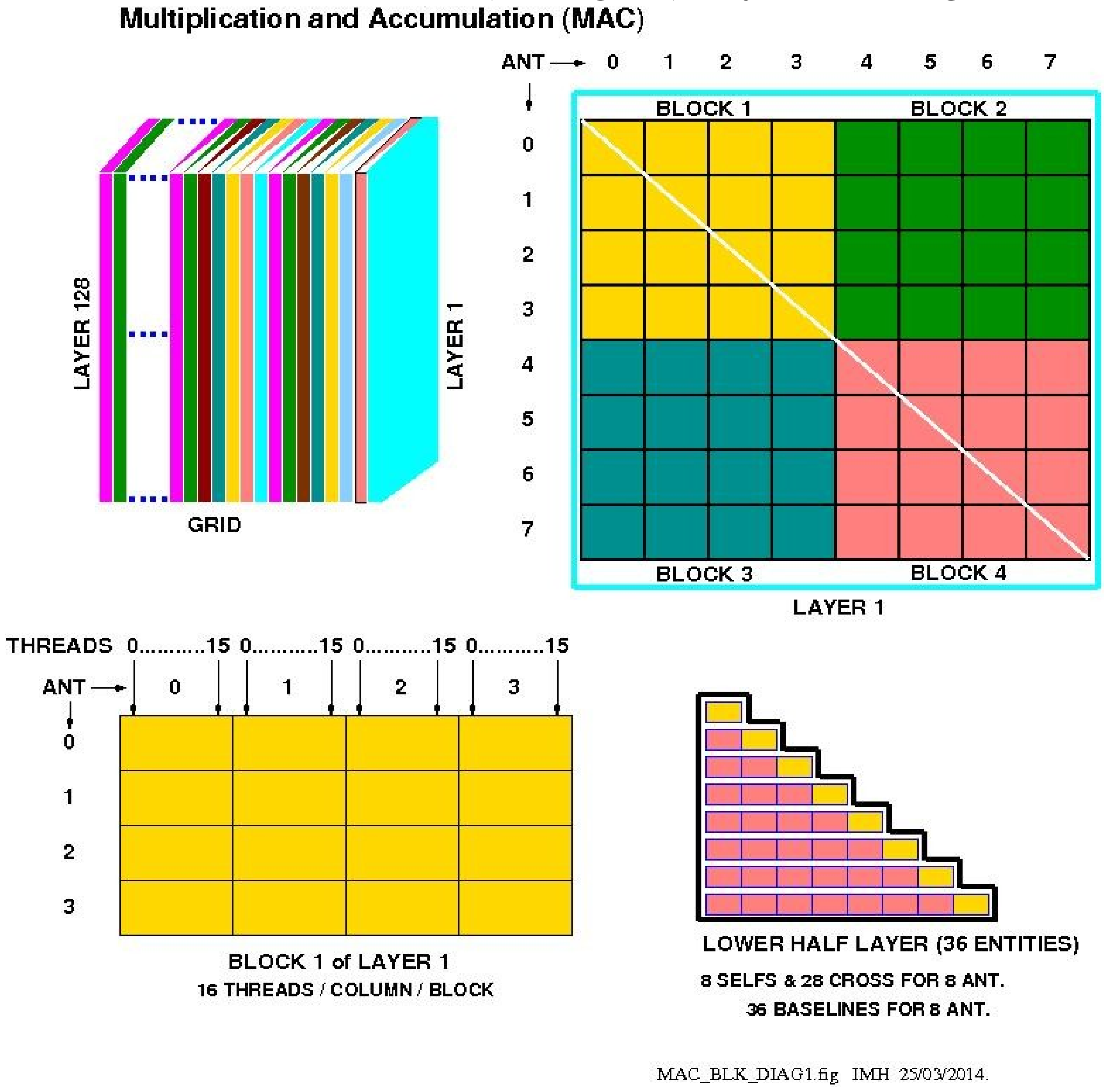
Parallelism is achieved over number of channels at the thread level and over the no. of

FFTs at the black level.

Phase rotation : This is done to compensate the fractional delay and fringe rotation. The delay cal function provides the values of delay in seconds, rate of change of delay, initial phase and fringe. From the values, total amount of phase that is to be applied is calculated and applied to every channel for all the antennas. Parallelism is achieved over the channels at the thread level and over the no. of FFTs per integration at the block level.

MAC : Multiply and Accumulation part is the most compute intensive one of all the other processes (see section 4.h Computations & IO requirements). For an array of N antennas, there will be N*(N+1)/2 baselines and correspondingly N*(N+1)/2 visibilities (including selfs of all N antennas). In the current system, the number of antennas should be in the integer powers of 2. While computing, the parallelism is achieved at both the thread level and block level in no. of frequency channels in X-dimension and no. of baselines in Y-dimension.

Let us say there are N antennas and K frequency channels. The block dimensions are selected as (bd.x,bd.y). Then grid dimensions (gd.x,gd.y) must be (K/bd.x , (N/bd.y)^2). The grid dimesions are selected in such a way that the product of (square of bd.y) and gd.y must be equal to (square of N). Each block does multiplication and accumulation of (bd.y)^2 baselines for bd.x channels. Here is an example for 8 antennas and 2048 channels. For 8 antennas, 36 baselines are formed (including selfs). Only the lower triangle elements



Multiplication and Accumulation (MAC)
LAYER 128
LAYER 1
GRID
ANT
0 1 2 3 4 5 6 7
BLOCK 1
BLOCK 2
BLOCK 3
BLOCK 4
LAYER 1
THREADS 0...........15 0...........15 0...........15 0...........15
BLOCK 1 of LAYER 1
16 THREADS / COLUMN / BLOCK
LOWER HALF LAYER (36 ENTITIES)
8 SELFS & 28 CROSS FOR 8 ANT.
36 BASELINES FOR 8 ANT.
MAC_BLK_DIAG1.fig IMH 25/03/2014.

are needed to be calculated from the baseline matrix(see fig). When we select block dimensions as (16,4) then grid dimensions become (128,4), which means each block calculates products of one 4x4 matrix for 16 channels. If any block is completely out of the lower triangle it simply quits without performing any operations.To reduce accessing the global memory for every baseline, shared memory is used.

e. Beamformer :

The beamformer specification is to provide two modes of operations : (i) an incoherent array mode, where the voltage samples from the selected antennae are added after converting to intensities, (ii) a coherent or phased array mode, where the voltage signals from the selected antennae are first added, and then converted to intensity samples. Both these modes require antenna based time delays to be corrected prior to the addition of the signals. Furthermore, for the coherent beamformer mode, it is also required to calibrate out antenna based phase offsets.

Currently, the sampling period is set to 1.3ms which is equal to averaging of 128 FFTs at 2048 spectral channels. The design provides two beams at the same time in two different beam host nodes. In each beam, either Incoherent array beam(IA) or Coherent beam beam(PA) can be selected. Further, if 4 stokes of PA are needed, then Interferometry needs to be run in 4 stokes mode. In the design, the parallelism is achieved over no. of spectral channels at thread level and no. of sampling period blocks in each integration of visibilty data(which is 671ms) at the block level. Each thread hence calculates average of one spectral channel for all the 128 FFTs in a sampling period block. The design was optimised to read the global memory only one for the processing of both beams. Also, array masking has been implemented at the lowest level to improve the performance and remove computations for antennas masked out of the array. Phasing for Coherent array(PA) mode has been implemented in the phase rotation stage.

f. Software Model :

The various processes mentioned above in the CPU-GPU are performed parallelly using nested openMP sections. The main thread creates two threads. One for data acquisition and shared memory buffering. The other thread creates four more threads for (1)data reading from shared memories, (2)MPI sharing of data, (3)correlation in GPU and (4)writing the visibilities to shared memory to be written by a host node. Double buffering ping-pong scheme has been implemented to enable the processing and data transfer in parallel.

g. Online control and visibility recording :

PROGRAMING Used :

Control and Monitoring programs for ONLINE system and GWB Systems are implemented using various programing and scripting languages, Such as C, C++, QT3, Bash, Perl, Perl-Tk, Tcl-Tk etc.

ONLINE System :

GMRT Wide-band Back-end (GWB) can be controlled from Online Machines (both *'lenyadri'* as well as *'shivneri'*). For this, a socket is created, through which online sends the commands to GWB and receives their acknowledgments. This Socket program has been written in C, C++. In this way, data control and monitoring information is shared between the two systems (ONLINE and GWB).

GWB System :

GWB data acquisition, processing and recording programs are written in C and C++. Where as Graphical user interface such as GWB DasConsole, GWB configuration, GWB Pulsar configuration, etc. are designed using C, C++ and QT3. Power Equalization, GAC-Config are written using C, Perl-Tk. Also, various other scripting languages like Bash, Perl, Perl-Tk, Tcl-Tk etc. are used for most online as well as offline applications.

USER INTERFACE :

After starting the data acquisition through GWB, we can control and Monitor it using both GUI and Command line tools as per the user's/observer's requirement. For this, DasConsole, GWB-Config, DasMon, Pulsar-DasConsole, Pulsar-Config, PowerEq, Phasing, GAC-Config etc. are Interface tools available.

OUTPUT DATA :

Data acquired from GWB is stored in Standard GMRT LTA format. And Pulsar Data is stored in RAW formats. This data can be monitored using online/offline tools viz. DasMon. ltahdr, tax, xtract, rantsol, pmon, etc.

h. Computation and IO requirements :

For uGMRT, FFT(16k point) computation requires 2.9 teraflops, phase rotation takes 0.1 teraflops and MAC takes 6.6 teraflops. The IO requirement at 400 MHz bandwidth and 4-bits per sample is nearly 25 GB/s.

The same computation requirements for GWB-II are : 153 gigaflops for FFT(4k point), 6.4 gigaflops for phase rotation and 230 gigaflops for MAC. IO requirement is 3.2 GB/s. The performance achieved on C2050s is 345 gigaflops for FFT and 490 gigaflops for MAC. For host to device transfer the bandwidth achieved is nearly 4.3 GB/s. On infiniband network the bandwidth achieved is 5 GB/s.

5. TESTS and RESULTS :

Interferometer :

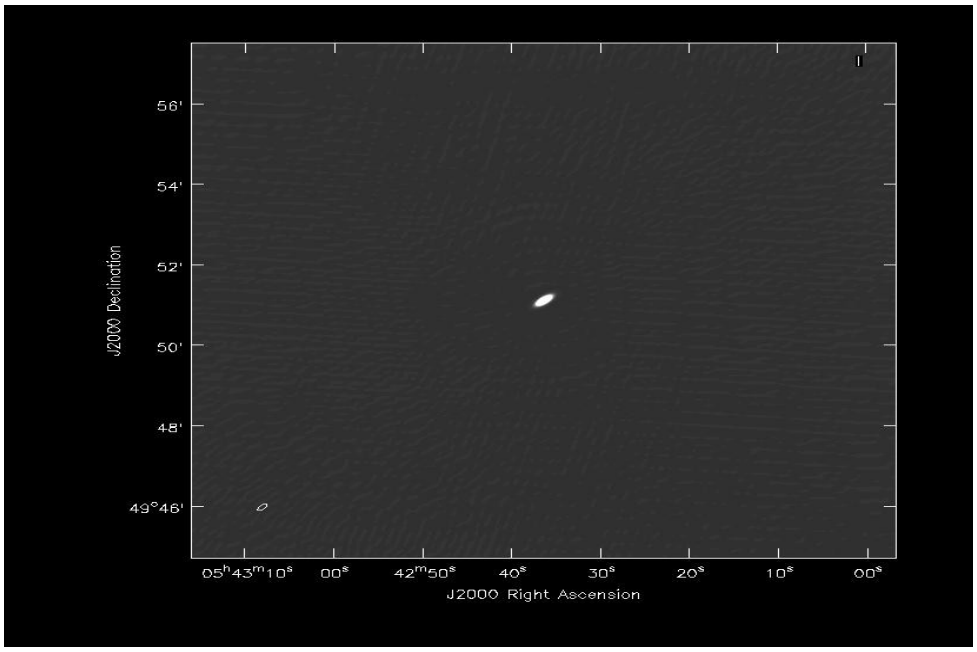


Image of 3C147 : First image of GWB correlator taken with 8 antennas at 1280MHz RF and 32 MHz signals. Data recorded on 07 June 2011

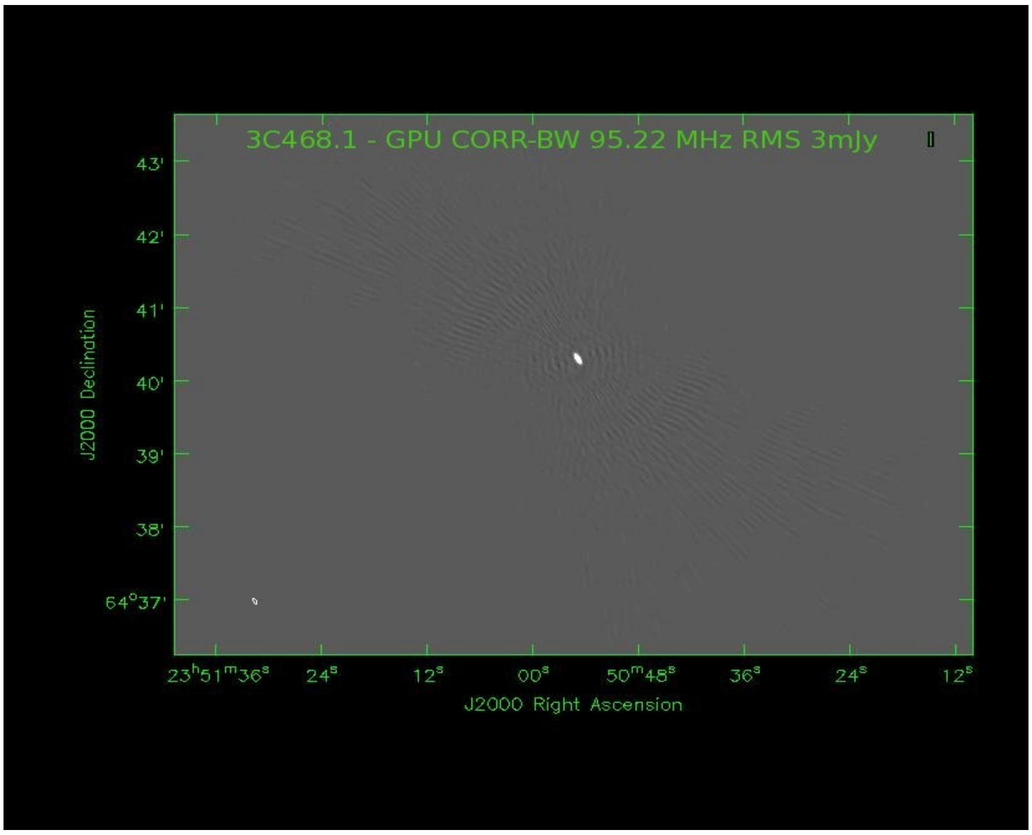
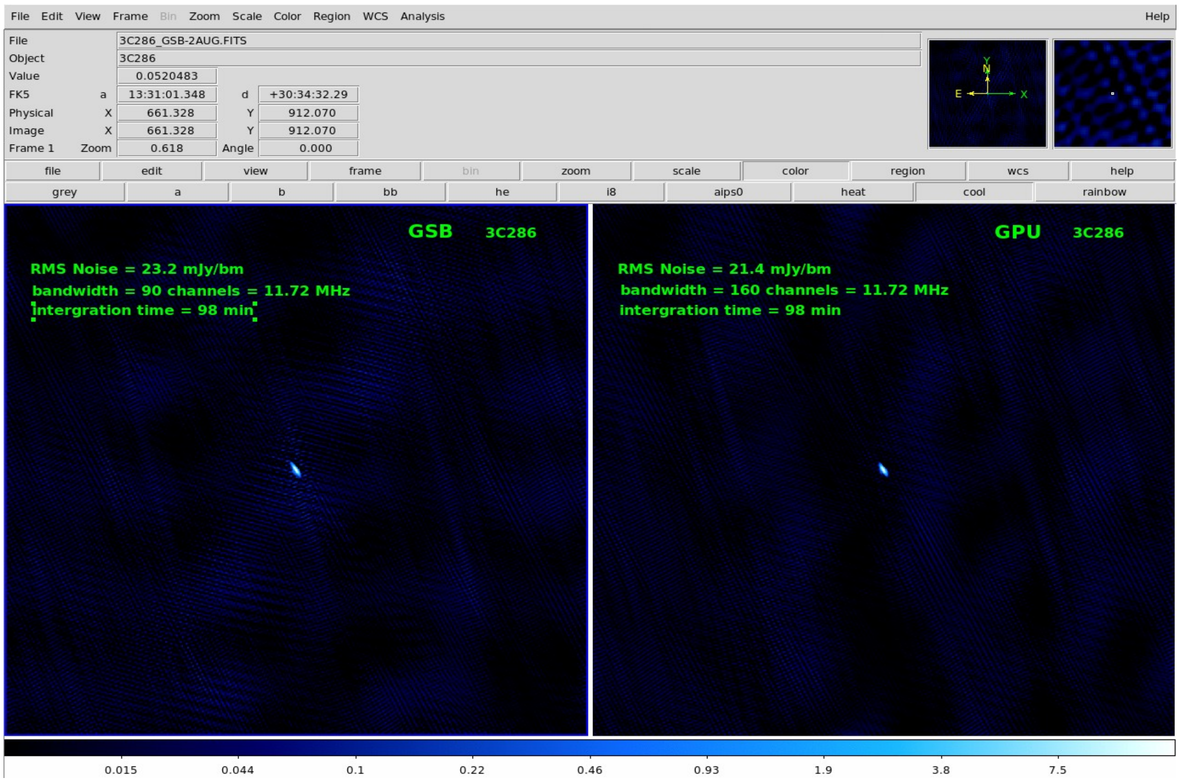


Image of 3C468.1 : First image using broadband signals


File Edit View Frame Bin Zoom Scale Color Region WCS Analysis Help
file edit view frame bin zoom scale color region wcs help
grey a b bb he i8 aips0 heat cool rainbow
E X Y N
0.015 0.044 0.1 0.22 0.46 0.93 1.9 3.8 7.5

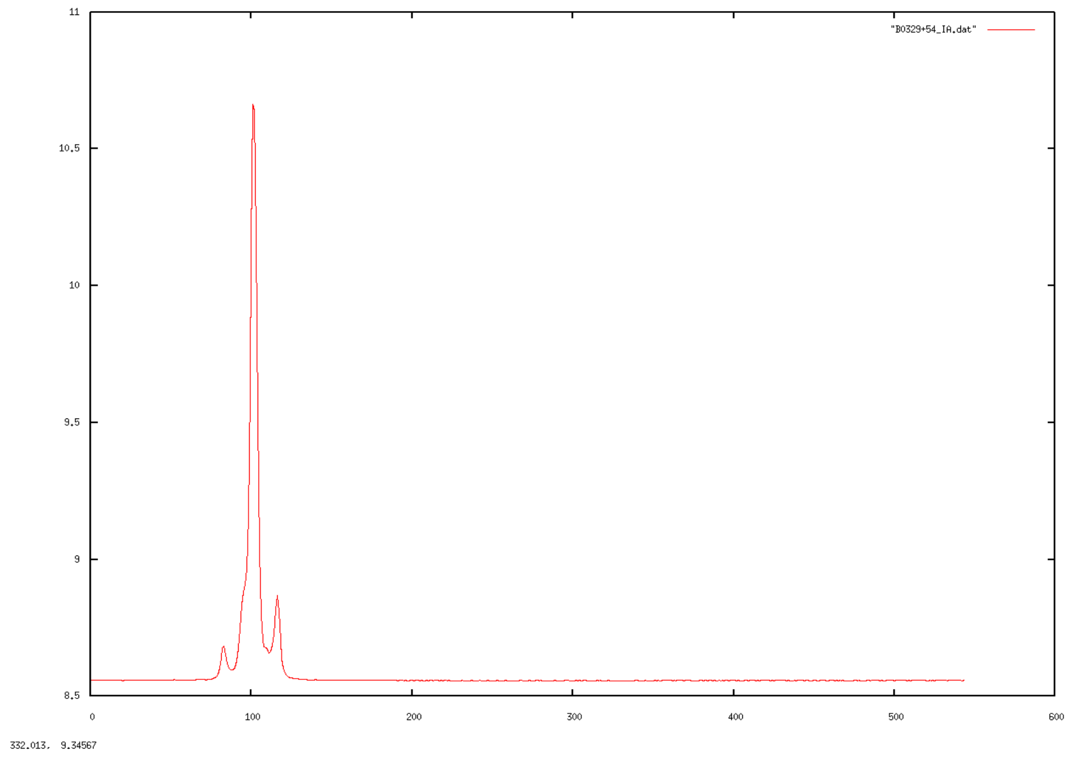
File 3C286_GSB-2AUG.FITS
Object 3C286
Value 0.0520483
FK5 a 13:31:01.348 d +30:34:32.29
Physical X 661.328 Y 912.070
Image X 661.328 Y 912.070
Frame 1 Zoom 0.618 Angle 0.000

GSB 3C286
RMS Noise = 23.2 mJy/bm
bandwidth = 90 channels = 11.72 MHz
intergration time = 98 min

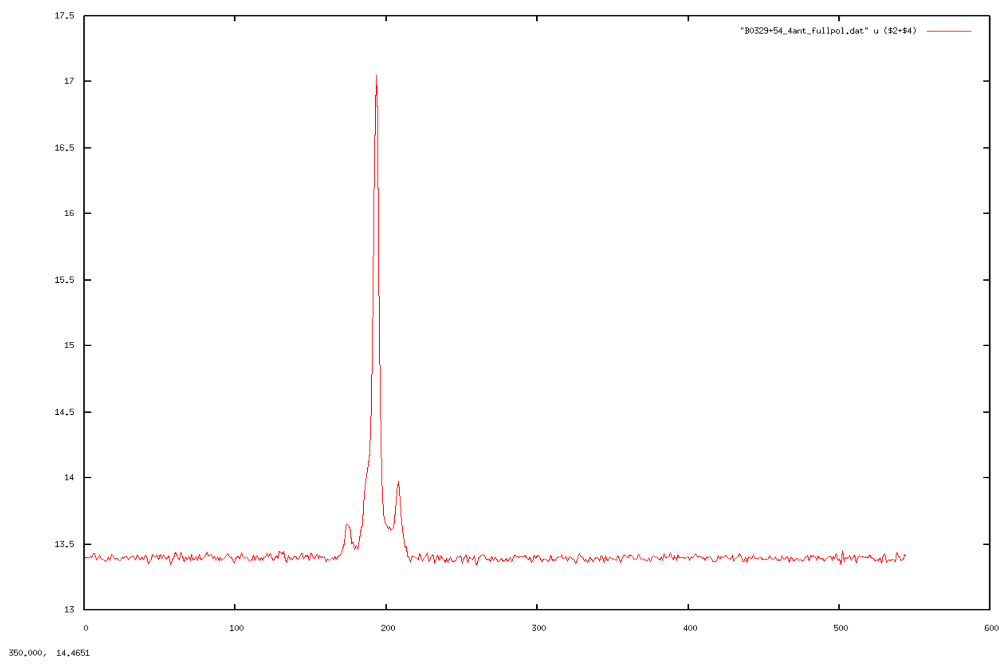
GPU 3C286
RMS Noise = 21.4 mJy/bm
bandwidth = 160 channels = 11.72 MHz
intergration time = 98 min

Image of 3C286 : A comparision between GSB and GWB

Beamformer :



Profile of B0329+54 : Incoherent array beamformer at 610MHz RF and sampling period of 1.3ms



Profile of B0329+54 : Phased array beamformer at 610Mhz RF and sampling period of 1.3ms

9. Conclusions :

An 8-antenn single pol/4-antenna dual pol backed has been released for testing by users. Scaling this prototype to 32-antennas completes the digital backend for uGMRT.

Appendix - 1 SOP

8-Antenna 8-bit / 4-bit GPU based CORRELATOR TESTING PROCEDURE

Version 3 , Released Date : 01/11/2013.

Written By : Sanjay S. Kudale / S. Harshavardhan Reddy/ I. M. Halagali.

1. HARDWARE SETUP

2. TERMINALs SETUP

3. PROCEDURE

  a. EDIT THE FILES

  b. CONFIGURE THE GPU

  c. ONLINE

4. ANALYSING THE RESULT USING TAX PROGRAM

5. POWER ON-OFF PROCEDURE

1. HARDWARE SETUP :

[CTRL_PC]	: 192.168.4.68 (hostname : ctrlpoco)
[USER_CTRL_PC]	: gmrt
[CTRL_PC_PASS]	: gmrttifr

CLUSTER II

[GPU_CTRL_PC]	: 192.168.11.15 (hostname : node51)
	# This is one of the PC in “cluster II. This m/c can be accessed from gsbm3”
[USER_GPU_CTRL_PC]	: gpuuser
[GPU_CTRL_PC_PASS]	: gmrt123
[ROACH0_IP]	: 192.168.100.19
[ROACH1_IP]	: 192.168.100.71

NOTE :

USING BROADBAND SIGNAL FOR GPU : BEFORE STARTING H/W SETUP, MAKE SURE RF POWER AT ADC INPUT BETWEEN -14dbm to -17dbm, Or equalize to 100-150 counts using GWB power equalization.

In this setup, two ROACH boards each with two iADC cards are used. Each iADC card has a clock, sync and two inputs for signal . While the clock input is a must for all iADC cards, sync inputs are needed only for iADC 0s of the two ROACH boards. Each GPU PC (a total of 4 GPU PCs) is equipped with two 10 GbE cards apart from the GPU card. One 10 GbE card is directly connected to one of the two ROACH boards while the other 10GbE card is connected to neighbouring GPU PCs via a 10 GBE switch.

The following connections needs to be done for antenna test.

1. Connect the ROACH boards to the controlling PC using 100MBPS ethernet.

2. Feed 400MHz, 0dbm clock signal to the clk_i inputs of the iADC cards from the signal generator. Also feed the PPS signal to the Sync input of the iADC 0 cards of both ROACH boards.

3. Connect the antenna signals to the iADCs and make note of the sequence as it is needed for “fstc” and “fringe” corrections. The sequence is as follows:

ROACH 0 iADC 0 (I+ & Q+)	--------	Antennas 0 and 1
ROACH 0 iADC 1 (I+ & Q+)	--------	Antennas 2 and 3
ROACH 1 iADC 0 (I+ & Q+)	--------	Antennas 4 and 5
ROACH 1 iADC 1 (I+ & Q+)	--------	Antennas 6 and 7

The integer delay correction is different from fringe and fstc corrections. For fstc and fringe corrections, every GPU node corrects corresponding values of all antennas for different time slices. While for integer delay corrections, every node corrects corresponding values of only two antennas. The mapping for integer delay correction is as follows.

Node 0	---------------	Antennas 0 and 1
Node 1	---------------	Antennas 2 and 3
Node 2	---------------	Antennas 4 and 5
Node 3	---------------	Antennas 6 and 7

4. Connect the ROACH boards to the GPU PCs using 10gbE cables. There are four 10 GbE connectors on each ROACH board. The design has been made to allot one 10 GbE connector per iADC board, which means only two 10 GbE connectors are used. Also, each 10 GbE connector sends out data at the rate of 800 MiB/s.

The mapping of 10 GbE connections is as follows:

10 GbE port on ROACH		GPU PC
ROACH 0 PORT 0	------------	Node 0
ROACH 0 PORT 1	------------	Node 1
ROACH 1 PORT 0	------------	Node 2
ROACH 1 PORT 1	------------	Node 3

5. Turn ON the ROACH boards.

2. TERMINALs SETUP :

Open the terminals and set into the directories :

CONTROL TERMINAL (192.168.4.68) :
```
ssh -X gmrt@ctrlpoco
cd harsha
```
==
GPU TERMINAL (192.168.11.15) :
```
ssh -X gpuuser@node51
cd ~/delay_cal/
```
==
SOCKCMD TERMINAL (192.168.11.15): (GPU CORR- sockcmd, connection to ONLINE)
```
ssh -X gpuuser@node51
cd ~/bin/released/
```
==
COLLECT TERMINAL (192.168.11.15) : (collect, acq-record connection)
```
 ssh -X gpuuser@node51
cd ~/bin/released/
```
==
DASSRV TERMINAL : (dassrv, communication between ONLINE-GPU/GSB CORR)
```
ssh -X observer@lenyadri/shivneri
```
(lenyadri or shivneri depend on whether GPU corr

is running from standby ONLINE or default ONLINE along with GSB)
cd /odisk/online1/gsbe/dassrv-gpu/
===
RECORD TERMINAL (192.168.11.15) :
ssh -X gpuuser@node51
cd /data1/gpuuser/
===
ONLINE USER TERMINALs :
ssh -X observer@lenyadri/shivneri
It is assumed, ONLINE is running, master, user0, user4
are started and running in terminals MASTER/USER0/USER4 respectively.
===

3. PROCEDURE :

a. EDIT THE FILES :

===
GPU TERMINAL (ssh -X gpuuser@node51) :
ONLINE m/c:
(lenyadri or shivneri depend on whether GPU corr
is running from standby ONLINE or default ONLINE)

shivneri	192.168.1.12
lenyadri	192.168.1.13

For change ONLINE m/c to other, edit host file /home/gpuuser/bin/released/SYS_FILES/hosts.dat, comment/uncomment relevant m/c.
===

GPU TERMINAL (ssh -X gpuuser@node51) :
edit /home/gpuuser/bin/released/SYS_FILES/sampler.hdr for antenna connections.
For 4 node cluster, keep no. of antennas = 8
For 2 node cluster, keep no. of antennas = 4
Follow the antenna connection seq. in sampler.hdr
file as below. (4 antennas per ROACH board)

```
ANTE1-ROACH0
ANTE2-ROACH0
ANTE3-ROACH0
ANTE4-ROACH0
ANTE1-ROACH1
ANTE2-ROACH1
ANTE3-ROACH1
ANTE4-ROACH1
```

GPU TERMINAL (ssh -X gpuuser@node51) :
edit /home/gpuuser/bin/released/gpu.hdr for required setup
GSB_LTA=1 is equal to 671088.64 micro sec.
GSB_LTA=4 is equal to 2684354.56 micro sec= ~2.6 sec
GSB_ACQ_BW is acquisition bandwidth (generally 200 MHz)
GSB_CHAN_MAX= 2048 for 2k channels,

GSB_CHAN_NUM= 0:2047:1 for its range.
GSB_STOKES = 1, currently only '1' is possible.
currently, GSB_RF, GSB_SIDEBAND & GSB_LO-1 are not implemented, so ignore.
==

b. CONFIGURE THE GPU :

==
CONTROL TERMINAL (ssh -X gmrt@ctrlpoco) :
./dual_adc_pps_8bit_cluster_das2.py
==
GPU TERMINAL :
./gpu_corr_released.sh
===

c. ONLINE :

Warning : Pl. ensure that sampler.hdr & gpu.hdr files are updated.

===
SOCKCMD TERMINAL :
./sockcmd.sh
===
COLLECT TERMINAL :
./collect
==
DASSRV TERMINAL :
./dassrv_released
===
USER0 TERMINAL :

cp 9;allant;cmode 1;tpa(11)=15;initndas '/temp2/data/gsb.hdr' # use "cmode 1" for GPU ONLY.

OR

cp 9;cmode 9;tpa(11)=15;initndas '/temp2/data/gsb.hdr' # use "cmode 9" for BOTH GSB & GPU.

OR

cp 9;cmode 8;tpa(11)=15;initndas '/temp2/data/gsb.hdr' # use "cmode 8" for GSB ONLY.

===
CONTROL TERMINAL (ssh -X gmrt@ctrlpoco) :
./set_cluster_das2.py
===
USER0 TERMINAL :
NOTE : Wait till minute boundry appears in GSB acq.
ante 8 5 3 9 10 29 30 8 25 #edit the antennas used and makesure sampler.hdr has same set of ant.

```
cp 0;defs 4;suba 4

cp 9;suba 4;prjtit'GPU';prjobs'GPU';initprj(15,'PRJCODE')
```
# use “initprj(15,'PRJCODE')” for BOTH GSB & GPU for default online.

OR

```
cp 9;suba 4;prjtit'GPU';prjobs'GPU';initprj(1,'PRJCODE')
```
# use “initprj(1,'PRJCODE') ” for GPU only for standby or only one backend.

===

USER4 TERMINAL :

```
>tpa 1299 1299 1350 1350 51 51
>prjfreq
>lnkndasq
>gts'3c48'
>strtndas
```
===

RECORD TERMINAL :

```
~/bin/released/record PRJCODE filename integration
        eg. ~/bin/released/record PRJCODE GPU_TEST.lta 1
```
===

4. ANALYSING THE RESULT USING TAX PROGRAM :

The results are analysed using the software tax (built at GMRT). Follow the steps given below :

scp the data to any NIS m/c's /scratch area.

edit fmt to 31.4f

# Offline analysis: (currenly in node51 ”192.168.11.15” m/c)

```
~/bin/released/ltahdr
~/bin/released/listscan
~/bin/released/gvfits
```

5. POWER ON-OFF PROCEDURE

a. POWER OFF :

1. Shutdown the nodes using script “ “ on node51.

2. Switch OFF Roach units in GPU rack : GPU ROACH 1 & 2 by holding down the black switch on the front panel for 10sec.

3. Switch off signal generator at bottom of the GPU rack.

a. POWER ON :

1. Turn On the GPU nodes 51 to 54.

2. Switch ON Roach units in GPU rack : GPU ROACH 1 & 2 by pressing the black switch on the front panel.

3. Switch ON the signal generator in the GPU rack.
4. Settings on the signal generator to feed clock.
    1. Set Frequency :

    2. Set Amplitude :

    3. RF ON and MOD off :

    4. Reference :

Appendix - 2 SOFTWARE PROGRAMS

To be added