

SOP for testing PACKETIZED CORRELATOR.

Sandeep C. Chaudhari & Irappa M. Halagali : 31/08/2012.

VERSION : 1

Packetized correlator is an general purpose re-configurable digital backend for radio astronomy applications on CASPER hardware ROACH-Vertex5 boards using XAUI or 10Gbe designs. This SOP is for XAUI designs.

I : Power ON and instrument settings.

- (a) Power on Distribution boards , AC distribution boards in the RACK
(verify : PC , ethernet swithces & ROACH Units are powered ON).
- (b) Now Switch ON the instruments and do the following settings -
- (c) Signal generator settings as CLOCK to F engine ROACH boards : Freq = 800MHz , Power = 0dbm, Set "Mod to OFF and RF to ON"
- (d) Connect the PPS signal available in the rack to all Sync input of F engine ROACH boards.
- (e) Noise Source as input signal through 200 MHz LPF , Total Power over BW = -12dbm @ iADC's inputs.
- (f) Power "ON" the ROACH boards by pressing the switch at front panel. All Roach boards will boot through NFS (Network File System) on control PC {192.168.4.71}.

II : Interconnections.

Connect the clock & pps signals available at top of the rack to all ROACH units clk_i and Sync inputs of iADC respectively.

(A) Testing by using Noise Source.

Feed the Noise Source input signal to I+ & Q+ inputs of all iADC-ROACH units.

(B) Testing by using Antenna signals.

Connect the antenna signals to I+ & Q+ inputs of all iADC-ROACH units.

III : XAUI BASED 4/8/16 Antenna packetized correlator setup.

GMRT Digital Backend specifications :

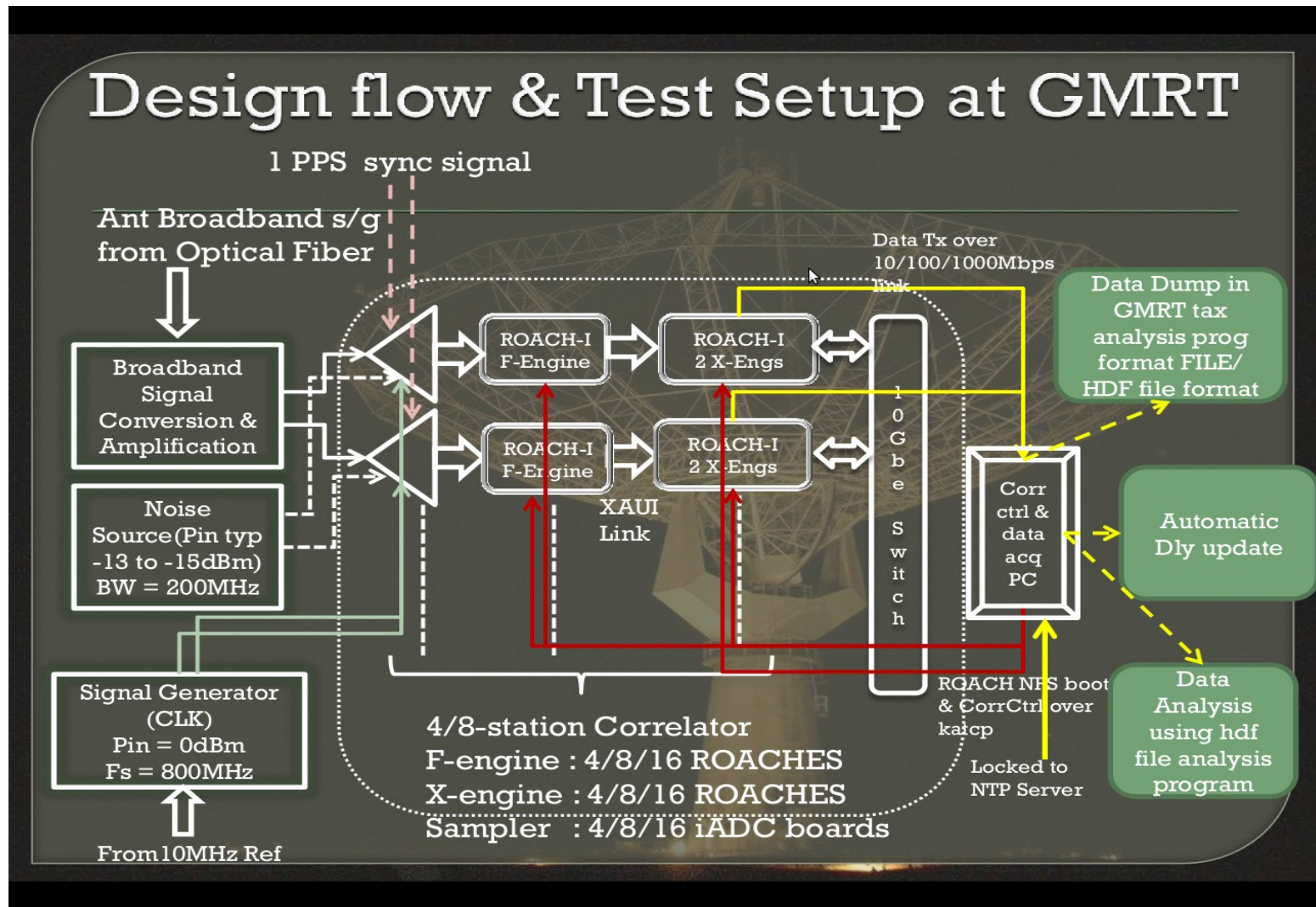
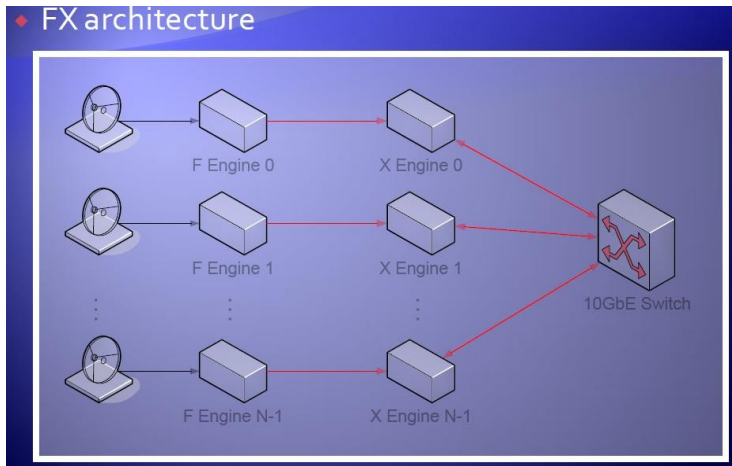
Number of stations	:	4/8/16 dual pol.
Maximum instantaneous Bandwidth	:	400 MHz
Number of spectral channels	:	512
Number of input polarizations	:	2
Full stokes capability	:	yes
Dump time	:	minimum 128 ms
Coarse and Fine delay correction range	:	+/- 128 us.

	4 antenna	8 antenna	16 antenna
Total ROACH boards required	8	16	32
F engine Roach boards	4	8	16
X engine Roach boards	4	8	16

Note : For complete packetized correlator setup refer the following diagrams.

- XAUI Packetized Correlator Connections :-**
1. nth F-eng CX-4 port-0 => nth X-eng CX-4 port-0.
 2. All X-eng CX-4 port-3 => 10GbE Switch.
 3. All ROACH board's eth-0 port connected to 1Gbps switch.
 4. 1Gbps to the control PC's eth-1 port in a private network.

Control PC eth-0 IP : 192.168.4.71
Control PC eth-1 IP : 192.168.100.1 (Private Network)
ROACH Board eth-0 IP : 192.168.100.<Last 2 MAC No.>



IV : Initialization & Data acquisition.

Login to control PC : `ssh -X gmrt@192.168.4.71`

Password : `gmrttifr`

> `sudo konsole`

This opens new console. Now start working in newly opened console. Press `shift+ctrl+N` to open more konsoles.

Note : 1. For all scripts -h commandline option will show options associated with that script.

2. Go to working Directory in Each KONSOLE TAB

Working Directories :

1. for 4 antenna

`/home/gmrt/Packetized_Corr/XAUI_Corr/4Ant_Corr/Ant_Tests/` or

`/home/gmrt/Packetized_Corr/XAUI_Corr/4Ant_Corr/Basic_Test/` (for Noise Source tests)

2. for 8 antenna

`/home/gmrt/Packetized_Corr/XAUI_Corr/8Ant_Corr/Ant_Tests/` or

`/home/gmrt/Packetized_Corr/XAUI_Corr/8Ant_Corr/Basic_Test/` (for Noise Source tests)

KONSOLE TAB-1 : { To issue some basic commands from ipython }

> `ipython`

```
In [1]: import corr,struct,time,sys,spread,numpy,pylab,katsdisp,h5py,signal
```

```
In [2]: dh=katsdisp.KATData()
```

```
In [3]: dh.start_spread_receiver()
```

```
In [4]: c=corr.corr_functions.Correlator(connect=True,
config_file='config_8ant', log_handler=None, log_level=20)
```

```
In [5]: c.deprog_all() (This Step is required if we want to test 4-antenna correlator in a 8-antenna setup)
```

KONSOLE TAB-2 : { For Configuring the system }

```
> corr_rx.py -a config_4ant #for 4 antenna
```

```
> corr_rx.py -a config_8ant #for 8 antenna
```

```
Parsing config file...INFO:corrsys:Configuration file config_4ant parsed ok.
done.
```

```
Initialising SPEAD transports for inter data...
```

```
Data reception on port 7148
```

```
Sending Signal Display data to 127.0.0.1:7149.
```

```
Storing to file 1346318311.91.corr.h5
```

```
starting target with kwargs {'acc_scale': False, 'sd_port': 7149, 'sd_ip': '127.0.0.1',
'filename': '1346318311.91.corr.h5'}
```

```
WARNING: This function is not yet tested. YMMV.
```

```
INFO:rx:Data reception on port 7148.
```

```
INFO:rx:Sending Signal Display data to 127.0.0.1:7149.
```

```
INFO:rx:Starting file 1346318311.91.corr.h5.
```

KONSOLE TAB-3 : { For initializing the system }

```
> corr_init.py config_4ant          #for 4 antenna
> corr_init.py config_8ant          #for 8 antenna
Connecting... done                  # messages for 4 antenna initialization.
=====
Initial configuration:
=====
Clearing the FPGAs... done.
Programming the F engines with r_128w_512_11_r370_mod3_16_2012_Jul_04_2019.bof and the X engines
with r_1f_2x_4a_r340c_2011_Feb_11_1454.bof... done.

Pausing 10GbE data exchange... Pausing Xengs... done.
Syncing the F engines... Armed. Expect trigg at 14:55:40 local (09:25:40 UTC). SPEAD packet
sent.
Checking F engine clocks... ok
Setting the board indices... done
Setting the FFT shift schedule to 0x7FF... done
Configuring EQ... done
Configuring the 10GbE cores... done
Waiting 13.6 seconds for ARP to complete... done
Starting 10GbE data exchange... X engines re-enabled.
Flushing loopback muxs... done.
=====
Verifying correct data exchange...
=====
Wait 2 seconds for system to stabalise... done
Resetting error counters... done
Checking that all XAUI links are working... ok
Checking that the same timestamp F engine data is arriving at all X boards within a sync
period... ok
Checking that FPGAs are sending 10GbE packets... ok
Checking that all X engine FPGAs are receiving 10GbE packets... ok
Waiting for loopback muxes to sync... ok
Checking that all X engines are receiving all their packets... ok
Setting the number of accumulations to 1562368 (2.000 seconds) and syncing VACCs... done
Checking vector accumulators... Waiting for an integration to finish... done. Checking... ok
Sending SPEAD metatdata and data descriptors to 127.0.0.1:7148... done
Configuring output to 192.168.100.1:7148... skipped.
Starting transmission of data... done
Resetting error counters... done
Enabling KITT... done
```

KONSOLE TAB-4 : { For running the data/packet transmitting script on X-engines to dump data in h5 format in the current directory }

A spead packet transmitting script invoked on all x-eng ROACH boards by this shell script.

Note : Dump file will be in the current directory with extention .h5 <corr_rx_script_start_time_since_epoch.h5>

```
#for 4 antenna
> cd /home/gmrt/Packetized_Corr/XAUI_Corr/4Ant_Corr/
> ./roach_TXallstart_lab.sh
#for 8 antenna
> cd /home/gmrt/Packetized_Corr/XAUI_Corr/8Ant_Corr/
> ./roach_TXallstart_lab.sh
```

```

1346319332.159112223
ssh root@roach030167 corr_tx_spead_inter.py -l 1024 -i 192.168.100.1 -x 2 -s
1346319332.159112223 315
ssh root@roach030116 corr_tx_spead_inter.py -l 1024 -i 192.168.100.1 -x 2 -s
1346319332.159112223 315
ssh root@roach030174 corr_tx_spead_inter.py -l 1024 -i 192.168.100.1 -x 2 -s
1346319332.159112223 325
ssh root@roach040235 corr_tx_spead_inter.py -l 1024 -i 192.168.100.1 -x 2 -s
1346319332.159112223 315

```

For Stopping the data transmission at any instant of time then run following shell script, which will kill spead packet tx script on all x-eng ROACH boards.

#for 4 antenna

```
>/home/gmrt/Packetized_Corr/XAUI_Corr/4Ant_Corr/roach_TXallstop_lab.sh
```

#for 8 antenna

```
>/home/gmrt/Packetized_Corr/XAUI_Corr/8Ant_Corr/roach_TXallstop_lab.sh
```

```

ssh root@roach030167 pkill -f corr_tx_spead_inter.py
ssh root@roach030116 pkill -f corr_tx_spead_inter.py
ssh root@roach030174 pkill -f corr_tx_spead_inter.py
ssh root@roach040235 pkill -f corr_tx_spead_inter.py

```

V : Offline data analysis.

For 4 antenna , h5 data file analysis script “h5file_analysis_4Ant_mod4.py” in the directory ~/Packetized_Corr/XAUI_Corr/4Ant_Corr/Ant_Tests/ is used.

For 8 antenna , h5 data file analysis script “h5file_analysis_8Ant_mod4.py” in the directory ~/Packetized_Corr/XAUI_Corr/8Ant_Corr/Ant_Tests/ is used.

1)Plot spectrum for a given time and given baseline with self and cross :-

```
>./h5file_analysis_4Ant_mod4.py -f <h5 file> -s -n --timerange=1 10 -b 2x 2y
```

This plots normalized cross spectrum and phase plot on another y-axis in linear scale. (Log scale not working)

```
>./h5file_analysis_4Ant_mod4.py -f <h5 file> -s --timerange=1 10 -b 2x 2y -p -l
```

This plots selfs & cross amplitude for given baseline in logarithmic scale and cross phase on another y-axis.

2)Plot given channel number over time for a given baseline / reference antenna :-

```
> ./h5file_analysis_4Ant_mod4.py -f <h5 file> -c 128 --timerange=1 20 -b 0x 1x
```

This plots normalized cross amplitude and phase on another y-axis over time for a given channel baseline.

```
> ./h5file_analysis_4Ant_mod4.py -f <h5 file> -c 128 --timerange=1 20 -b 0x 1x -p
```

This plots cross phase over time for a given channel and baseline

```
> ./h5file_analysis_4Ant_mod4.py -f <h5 file> -c 128 --timerange=1 20 -r 0x -p
```

This plots cross phases over time for a given channel and all baselines with reference antenna.

```
> ./h5file_analysis_4Ant_mod4.py -f <h5 file> -c 128 --timerange=1 20 -r 0x -n
```

This plots normalized cross amplitude over time for a given channel and all baselines with reference antenna.

```
> ./h5file_analysis_4Ant_mod4.py -f <h5 file> -c 128 --timerange=1 20 -r 0x
```

This plots normalized cross amplitude and phase on another y-axis over time for a given channel and all baselines with reference antenna.