SKA Project Series
LFAA Signal Processing structure

G. Comoretto[1]

[1]INAF - Osservatorio Astrofisico di Arcetri

Arcetri Technical Report N° 2/2014
13-mar-2014

Abstract

The signal processing chain for the LFAA correlator, from ADC to the multiplier elementary cell, is described. The correlator adopts a FX architecture, with the frequency channelization implemented as a two stage polyphase filterbank. Delay compensation is performed after the first channelization, both in the time and in the frequency domain.

Contents

List of Tables

List of Figures

List of acronyms

ADC: Analog to Digital Converter

ADD: Architecture Design Document

ASIC: Application Specific Integrated Circuit

COTS: Commercial Off-The-Shelf

CSP: Central Signal Processor

DDR3/DDR4: Double Data Rate 3(4) memory standard

DSP: Digital Signal Processing

EMC: Electromagnetic Compatibility

EMI: Electromagnetic Interface

FFT: Fast Fourier Transformation

FPGA: Field Programmable Gate Array

Gb: Giga bit

GB: Giga byte

GPU: General Processing Unit

HDL: High Level Design Language

HMC: Hyper Memory Cube memory standard

ICD: Interface Control Document

IICD: Internal Interface Control Document

INAF: National Institute for Astrophysics

I/O: Input/Output

IP: Intellectual Property

LFAA: Low Frequency Aperture Array Element or Consortium

LFA: Low Frequency Array

LTA: Long Term Accumulator

MATLAB: MATLAB simulation language and application

M&C: Monitor and Control

PDR: Preliminary Design Review

PFB: Polyphase Filter Bank

PIP: Physical Implementation Proposal

PMX: PowerMX

PPS: Peak per Second

RFI: Radio Frequency Interference

RS: Requirement Specification

SAD: System Baseline Design

SDP: Science Data Processing

SKA: Square Kilometre Array

SKAO: SKA Organisation (or office)

SOW: Statement of Work

SW: Software

SYSML: System Engineering Simulation Language and application

Tb: Tera bit

TB: Tera byte

TBC: To be confirmed

TBD: To be decided

TDT: Time Domain Team

UDP: User Datagram Protocol

UML: Unified Modelling Language

WBS: Work Breakdown Structure

WE: Work Element

WP: Work Package

1 Introduction

The LFAA processing structure is distributed across the beamformer, that is part of the LFAA antenna subsystem, and the LFA CSP correlator. Data processing is however almost continuous, without a ”natural” boundary, and it must be considered at a system level as a whole. Some element can be arbitrarily assigned to one or the other subsystem, and a clear interface must be established.

In this report we try to described the processing as a whole, with some proposals for the boundary that are intended as a guideline.

The general system structure is shown in fig. 1.

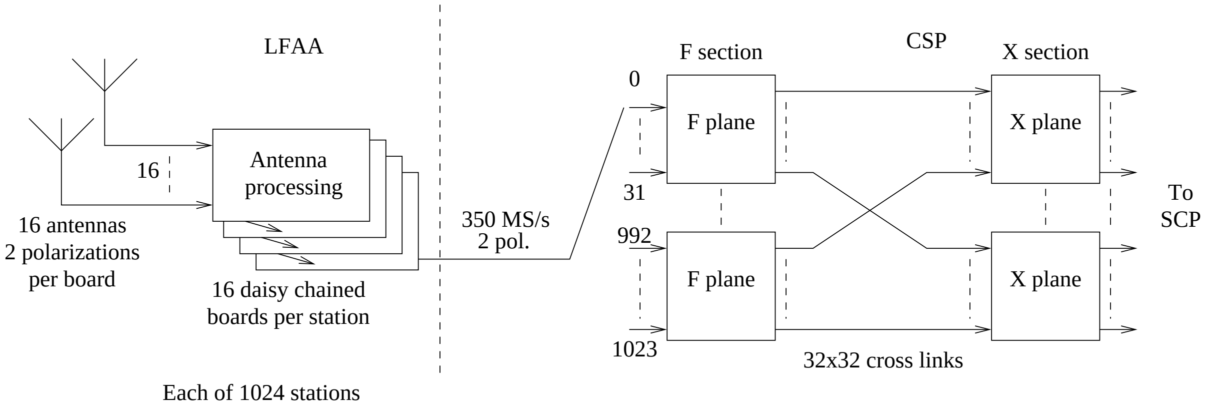

Figure 1: Structure of the LFAA instrument

The antenna processor and beamformer (section 3) performs an initial frequency channelization, amplitude and phase correction for each antenna, and the beamforming in the frequency domain. The output data is phase corrected for a reference point in the subarray (station), but not for geometric delay due to the subarray position in the array. A possibility to trade bandwidth for number of beams is provided. The antenna processor and beamformer are shown in figure 1 and 2as separate entities, but the beamformer can be implemented physically in a distributed structure among the antenna processors.

The CSP F section (section 4) processes data from several stations, correcting for geometric delays in the time domain, and for fine delay in the frequency domain. A second (fine) channelization is performed on each coarse frequency channel. Data for all antennas in a group of channels is sent to the X section (section 5) of the correlator.

Some utility modules are also used across the design (section 7), like small test signal generators/checkers to test signal integrity across links, a test vector generator to test the system without a real input signal, and the infrastructure to proper paketizing the data with the correct timing and data valid informations.

2 General description

This document is based and details the general physical architecture document presented at TIM1 [1]. Characteristics are based on the level 1 SKA specifications[7] and derived level 2 and 3 specifications.

Some assumptions have been made:

- The oversampling factor $O_f = 8/7$ has been adopted. This increases the complexity of the first polyphase filter, but reduces by $\approx 4\%$ the requirements on bandwidth for the links and memory blocks
- An attenuation of 60 dB has been assumed in the filters. This is consistent with the specifications, but may prove insufficient for effective RFI rejection
- The integration period is a multiple of the minimum period in the level 1 requirements,

2.1 Level 1 specifications

The level 1 specifications for the correlator are listed in table 1[7].

Specification for adjacent channel leakage are inconsistent. It is not possible to have a flat response across spectral channels and a leakage of 1/1000 between adjacent channels with a reasonable filter length.

Parameter	Specification	Notes
Station diameter	35 m	
Antennas per station	256	
N. of stations	1024	
Antenna distribution	50%	< 600 m
	25%	$600 < d < 1000$ m
	25%	$1 < d < 100$ km
Subarrays	Multiple	Number unspecified, likely 16
		All parameters independently configurable
Spectral channels	> 256000	Over observed band. More required
Spectral leakage	< -30 dB	For adjacent chans, see note
	< -60 dB	Probably smaller, considering RFI
Attenuation at chan. edge	-3 dB	to obtain a flat frequency response
Spectral ripple	< 0.01 dB	After calibration
SNR degradation	$< 2\%$	relative to analog processing
Integration time	0.6–6 s	

Table 1: LFAA correlator specifications

2.2 General LFAA processing description

The low processing chain is shown in figure 2.

The processing pipeline includes:

- The ADC interface. Current idea is to use a serial JESD204B protocol. Probably the best solution is to use a COTS IP block
- Framing logic. Data is processed in frames of 1024 samples. Due to oversampling, frames overlap by 1/7 (oversampling factor of 32/28, see [2]) and thus ADC samples are framed in blocks of 892 samples. Frame alignment occurs relative to a specific timing pulse, and then proceeds in free run.
- (Optional) Wideband RFI detection. A simple total power measurement over a FFT frame (896 points). If a threshold is exceeded, frame is marked bad. Problem: blanking a frame would require special processing in the polyphase filter.
- First stage polyphase filter. Coverts the sequence of real ADC samples into 512 frequency channels, equispaced in the Nyquist ADC band ($0 < f < f_s/2$). Output data rate is $7/8192 f_s$
- Frequency and beam selection. Only a subset of the frequency channels, i.e. those within the observed band, is selected, for a maximum of 300 MHz. It is possible to select a fraction of the band, processing it multiple times to implement multiple beams
- Amplitude and phase calibration. The filtered channels are multiplied by a complex (fixed) calibration curve, specific for each antenna. and by a (variable) amplitude-phase beamforming coefficient. The coefficient is beam specific.
- Beamforming. It is basically an adder.
- Delay model. To be determined how much of the model is computed locally and how much processed in the control computer and downloaded
- Total power. For each station, a coarse autocorrelation spectrum is provided, with one point per frequency channel. A cross spectrum can also be provided, on a subset of the frequency channels, by selecting appropriate coefficients in the beamforming (to place it in a sot of bypass mode)
- Coarse RFI block. It is a placeholder for any RFI scheme that operates on the station data at the coarse frequency resolution. RFI information (flagged data, etc) is embedded in the data frames, in format TBD.
- Coarse re-quantization. Data must be re-quantized to a number of bits TBD (8+8?). A fixed re-quantization can be used, as the amplitude calibration includes a frequency equalization. and transmitted over the fiber link in appropriate format (UDP frames with header)
- First corner turning. Data from several antennas, both polarization, all frequencies and a complete integration frame (200 ms?) is stored and retrieved as a complete time sequence for a small number

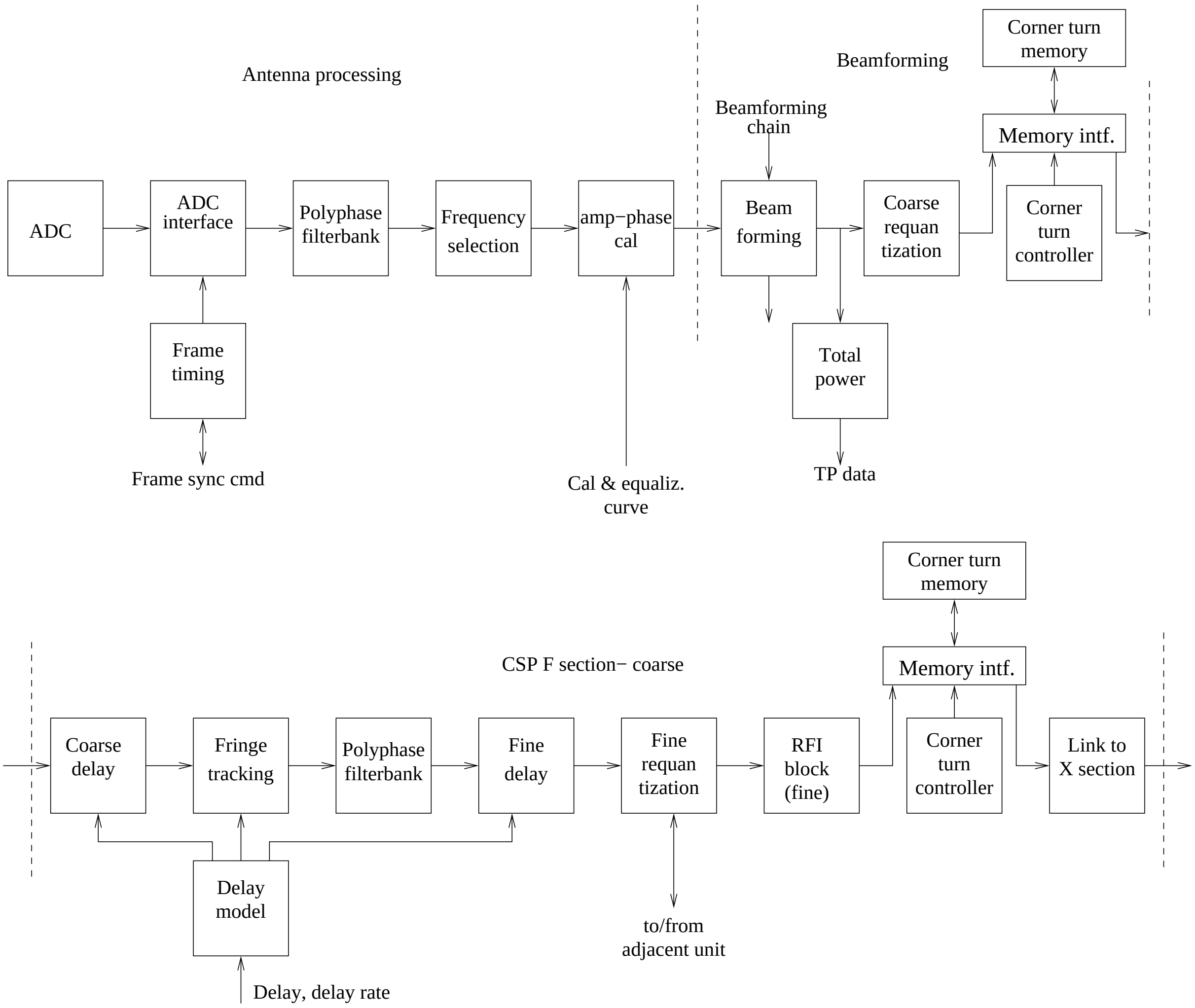

Figure 2: General processing in the LFAA, from the ADC to the correlation chip

of frequency channels, both polarizations, one antenna. It is implemented ad fast wide external memory (DDR4/HMC). The memory interface deals with the appropriate addressing scheme for the fast memory.

- Delay model. Computes varying geometric delay. Delay is fixed for the duration of the integration frame.
- Fine polyphase filterbank. Divides each coarse frequency channel into 1024 fine channels. Only 892 (7/8 of the total) channels are output, i.e. the overlapping region of each coarse channel is discarded.
- Fine delay. Subsample delay computed as a frequency slope. Adjusted during the integration frame, as needed.
- Fine RFI block. Analogue to the coarse RFI block.
- Fine re-quantization. To the data width used in the correlator
- Link to X section. Each output link transmits data for 1/32 of the band, 32 antennas, both polarizations, to a correlator plane (32 planes in total). To achieve this data is exchanged between adjacent FPGAs and slightly reformatted.
- Second corner turning. Data from all antennas, one integration frame, 1/32 of the band arrive to each correlator plane one coarse channel (892 fine channels) at a time. These data must be distributed to the correlation units one fine channel at a time. Is implemented in large external memory

- Correlator unit. Processes all antennas/polarizations for a correlation time frame and one fine frequency channel. Outputs correlation coefficients to the long term accumulation. Actual correlation and LTA processing is out of the scope of this memo
- Long term accumulator. Accumulates multiple integration frames. It is implemented in large external memory.

Signal processing occurs in several physical locations. An antenna processing board contains ADCs for a number (tentatively 16) of antennas, both polarizations. These boards perform all processing up to beamforming. Beamforming is performed in a hierarchic structure: all antennas in a board are combined in a single signal, and signals for all boards in a station are then summed together. The sum and subsequent processing can be performed either in the antenna boards themselves, or in a separate platform.

Signal for each station is then transferred over a fiber to the CSP. A first FPGA receives the signals for a number of stations (e.g. 16), and performs all the processing that is specific for each station (F part). Two (or more) F part FPGAs share their processed data, and connect with a set of 32 links to 32 X boards.

2.3 Data rates and internal clock frequencies

The sample rate for the LFAA correlator is too high for direct processing in a FPGA. Time multiplexing is thus necessary. Practical clock frequency for current or near future FPGAs families imposes a minimum time multiplexing factor of 4 for the input samples, with a corresponding clock frequency of 214 MHz. Subsequent clock and sample rates are bound to this initial choice, and are summarized in figure 3. At each stage the clock speed, multiplexing factor, and data rate are listed.

The coarse channelizer operates at fixed clock rate, but already at the channelizer output the channelized frames are shorter than the number of clock cycles needed to compute them, because of discarded channels, and therefore the sample rate is less than the clock rate.

The approximate number of multipliers required in each stage is also shown, This does not include multipliers used by the RFI detection and mitigation stages.

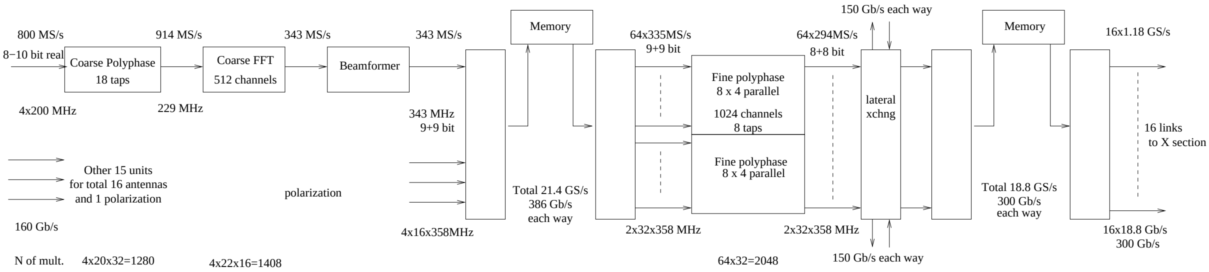

Figure 3: Details on sample and clock frequencies in the LFAA correlator F section

3 Antenna processor

This section is part of the LFAA work-package. It includes the ADC interface, the first stage channelizer, a calibration correction block, the beamformer and the processing section for the beamformed signal (RFI mitigation, re-quantization, link formatting).

3.1 ADC input section

This section includes the ADC interface, and the frame synchronization logic.

Each ADC produces real samples with 8(or 9, 10, TBD) bit resolution, that are transmitted to the FPGA using a JESD204B serial link interface. The interface is an industry standard, for which FPGA vendors provide COTS IP blocks. The interface block provides synchronization, i.e. it is possible to precisely tag each sample to a specific clock cycle.

Samples are framed in fixed length frames. Each frame contains a number of samples equal to the FFT length of the coarse polyphase block (PFB) divided by the oversample factor O_f used in the PFB. In our case $O_f = 8/7$ and the frame length is 892 samples.

The frame is synchronized to the system PPS pulse. The synchronization unit is programmed specifying a specific second and an arbitrary offset, in samples, from the PPS pulse. Any pending frame is not transmitted, and a new frame is started at the specified clock cycle. A frame counter keeps track of how many frames have been produced since the synchronization.

The output of the ADC section consists of n_s streams of real sample data. Each stream is multiplexed $\times 2$, i.e. two consecutive samples are presented at each clock cycle. An example of the frame format is shown in table 2, for a 16 ADC antenna FPGA.

```
<-                  448 clock cycles                     ->
[(S00T0, S00T1), (S00T2,S00T3), ... , (S00T894, S00T895)]
[(S01T0, S01T1), (S01T2,S01T3), ... , (S01T894, S01T895)]
[(S02T0, S02T1), (S02T2,S02T3), ... , (S02T894, S02T895)]
[(S03T0, S03T1), (S03T2,S03T3), ... , (S03T894, S03T895)]
...
[(S15T0, S15T1), (S15T2,S15T3), ... , (S15T894, S15T895)]
```

Table 2: Format of the frame from the ADC input block

In this table each sample is labelled by antenna (`S<nn>`) and time (`T<nnn>`).

3.2 First channelizer

The first channelizer (or coarse PFB) is implemented as an overlapping polyphase filterbank. It converts a real input stream, at data rate f_s, with samples of 8-10 bits, to a complex output, in records of n_c channels (512) complex, with a data ate of $O_f/2f_s$.

The channelizer is implemented using complex radix-4 FFT. As described in appendix A, the input samples are first packed into complex values $y_k = x_{2k} + i\,x_{2k+1}$, and then processed in a serial/parallel FFT architecture. As a FFT serial processor can process 4 complex signals in parallel, two antennas are processed by each FFT engine. The two parallel samples for each antenna are combined in the last stage of the FFT, implemented as a parallel butterfly, and the FFT for the original, real valued, sample sequence is recovered in a dedicated stage (see A.2). 8 FFT engines are required for a 16 input antenna FPGA.

Output is synchronized among all the (256k) antennas in the LFAA. Synchronization is usually wrong in an absolute sense, as the local time in each antenna is off by the cable delay. Local time-keeping must account for this, not to duplicate cable delay error.

Output of polyphase filterbank is composed of frames of 1024 coarse channels, over 4 links per PFB, with the format shown in table 3, with each sample denoted by antenna number and frequency index (`F<n>`, with $0 \leq n \leq 511$. Each output frame corresponds to one input frame, with the difference in frame length compensated by the difference in clock rate.

```
<-                  512 clock cycles                       ->
[S00F000, S00F001, ... , S00F127, S01F000, .... , S03F127]
[S00F128, S00F129, ... , S00F255, S01F128, .... , S03F255]
[S00F256, S00F383, ... , S00F257, S01F256, .... , S03F257]
[S00F384, S00F511, ... , S00F385, S01F384, .... , S03F385]
```

Table 3: Format of the frame at the output of the coarse FFT unit

The frame is rearranged in order to output data from all antennas for a limited number of coarse frequency channels. The channelizer output (table 4 is usually a frame of all the channels in the passband, e.g. 384 channels for $f_s = 800$ MHz and a 300 MHz bandwidth (50-350 MHz). The output average data rate per antenna is 300 MHz times O_f, i.e. 342 MS/s, irrespective of f_s.

The channelizer output can be composed of 2-4 sequences of less channels to implement 2-4 beams. In this way it is possible to implement multiple beams, with reduced bandwidth, for a constant total bandwidth. Although the mechanism can in principle produce an arbitrary structure of beams each with an independent bandwidth and center frequency, a general structure with complete arbitrariness can be difficult to manage. We will assume that all beams will have the same characteristics, and that the number of beams is restricted to a power of 2.

```
<-      384 clock cycles      ->
[S00F064, S00F065, ..., S00F447]
[S01F064, S01F065, ..., S01F447]
[S02F064, S02F065, ..., S02F447]
...
[S15F064, S15F065, ..., S15F447]
```

Table 4: Format of the frame from the coarse channelizer, 1 beam

An example of a frame with two beams is shown in table 5. Samples are now labelled also by beam, as B<n>. Both beams cover the frequency range from 78 to 228 MHz.

```
<-                384 clock cycles                       ->
[S00B0F100, S00B0F101, ...S00B0F291, S00B1F100, ... S00B1F291]
[S01B0F100, S01B0F101, ...S01B0F291, S01B1F100, ... S01B1F291]
[S02B0F100, S02B0F101, ...S02B0F291, S02B1F100, ... S02B1F291]
...
[S15B0F100, S15B0F101, ...S15B0F291, S15B1F100, ... S15B1F291]
```

Table 5: Format of the frame from the coarse channelizer, 2 beams

3.3 Beamformer

The beamforming process includes

- Calibration: Each antenna has an individual frequency response, that must be calibrated and corrected, by applying a complex calibration correction. An equalization curve can be also included, to flatten the spectral signal content.
- Delay/amplitude coefficient, i.e. the (frequency dependent) weight in the beamforming process.
- The actual sum of signals from all antennas

Both multiplicative coefficients can be combined together in a single multiplicative table, stored in memory. The task of computing and updating 384 complex coefficients for each of the 256K antennas can be not trivial, and some sort of internal intelligence could be provided to simplify it. For example the local control computer could update only the delay value, and a phase slope can be computed on the fly and added to a calibration/amplitude curve that is updated less frequently. A different delay must be specified for each antenna and beam. This can however prevent some advanced beamforming technique, e.g adaptive RFI nulling.

The maximum delay with respect to the phase center across a station is 15m = 50ns = 40 samples at 800 MS/s. For 512 channels, this corresponds to 1 turn every 25.6 channels, or a max decorrelation of 0.48% in one channel The same phase error occurs for Earth rotation in about 20 s. Coefficients must be updated at most every 10 s.

The beamformer then is just an adder. All signals in one antenna FPGA are combined together to produce a partial sum. Sums from all antenna FPGAs in a station are combined together in a second stage adder or, more effciently, using a ring (daisy chain) structure. This would minimize the link resources, as the signal has to travel only between adjacent boards, but require an internal syncronization buffer to compensate the delay across the ring. using a 10G link it would be possible to propagate a partial sum with 14+14 bits per sample.

The last board in the chain will combine the two polarizations in a single high speed link.

3.4 Station beam post-processing

The output of the beamformer is further processed before it is sent to the CSP. This includes:

- A total power spectrum is computed for diagnostics.
- A RFI block: This is essentially a placeholder for the RFI mitigation system designed as part of the RFI WE.

- Final re-quantization: to 8+8 bit (likely, TBD)

The total power meter basically builds a vector of 2 (polarizations) times 384 (frequency channels, beams) squares of the data samples, that are integrated over a programmable time interval (number of frames). The same circuit can be used for calibration. By implementing a large number of "beams", and adopting a set of beamforming coefficients corresponding to a single antenna per beam, it is possible to cross correlate one antenna with all the others. A narrow-band correlator, with just one broad channel at a time, can then be implemented.

Output signal should be quantized to (at least) 8+8 bit. The output data rate for 8+8 bit, 2 pol, 300MHz, $O_f = 8/7$ would be 11.0 Gbps, exceeding one 10Gb Ethernet link. 7+7 bit would fit in 10Gbps (barely). 6+6 bit would give 8.532 Gbps

Dynamic signal quantization (i.e. automatic adjustment of signal level) would be problematic for calibration, a fixed quantization would be better. But in this case the quantization should be able to cope with variation in signal level due to variation in the astronomic signal in the beam. A quantization scheme of 6+6 bit would marginal. 7+7 bit gives a headroom of 6-7 dB for astronomical signal increase in each channel. 8+8 bit gives 14 dB of headroom. (see discussion in [4])

The output format in the data link is shown in table 6. Data is divided in frames of 768 samples (1536 bytes for 8+8 bit format), with a frame every coarse channel time sample (893 kHz for $f_s = 800$ MHz and $O_f = 8/7$).

```
<-                     768 samples                    ->
[S00F064,S01F064, S00F065,S01F065, ... , S00F447,S01F447]
```

Table 6: Format of the link between the LFAA stations and the CSP

4 Central Signal Processor - F section

The CSP receives one link of data from each station. Each link carries 300 MHz of bandwidth for one station, both polarizations. The bandwidth can be split among different beams. The F section of the correlator is responsible of channelizer the received data streams to the final resolution, to compensate for the geometric delay and to present a subset of the frequency channels to the X section.

The F section is organized in F nodes. A node processes the data for a subset of the stations, and is implemented in a single FPGA.

4.1 F board corner turner

The first element of each F node is a corner turner. The corner turner stores the coarse FFT frames sequentially in a large memory, in the order they arrive, and reads back a limited number of coarse frequency channels, to be processed in parallel by the second stage channelizer, as consecutive time samples. It performs also delay compensation, in units of one coarse channelizer time step.

Antennas are stored sequentially, i.e. one frame for all antennas is stored before next frame for all antennas. Polarizations are the fastest running index. Blocks of consecutive frequency channels are stored sequentially for all antennas, with block length chosen in oder to be read back in a streaming fashion, with no addressing overhead. This depends on the memory technology, but a good assumption is to have a block length of 64 bytes. With 8+8 (or 9+9) bit per sample, 2 polarization, each block could then contain 16 adjacent frequency channels. Format for writing data to memory is shown in table. 7. Station index includes polarization, with even indexes for the first polarization and odd indexes for the second. For example `S06` and `S07` denote the two polarizations of the fourth station (station 3).

Each memory block stores a number of antennas limited by the available memory bandwidth. For DDR4 at 3200 MT/s (25.6 TB/s)[1] 8 antennas can be stored per memory bank (86% memory bus efficiency, full duplex). The memory controller provides local store to efficiently interleaved read and write of sufficiently large memory block. Multiple memory banks are used to store all available antennas received by a single F node.

Alternatively HMC can be used for the corner turner. Standard HMC links carry 10 Gb/s of data, so sligtly more than 1 link (each way) per antenna is required. For a 16 antenna node 20 links are required each way, with a reasonable overhead.

[1] The currently available Xilinx and FPGAs can support nly 2400 MT/s DDR4[?]. No 3200 DDR4 components will be available in the short time frame. 3200 MT/s capability thus requires both new generation (e.g. Stratix10) FPGAs and DDR4 chips.

```
<-     8 clock cycles       ->
<-    192 clock (1536 memory transfer) cycles       ->
[S00F064,S02F064, ... , S14F064, S00F080, ... S14F432]
[S01F064,S03F064, ... , S15F064, S01F080, ... S15F432]
[S00F065,S02F065, ... , S14F065, S00F081, ... S14F433]
[S01F065,S03F065, ... , S15F065, S01F081, ... S15F433]
...
[S01F079,S03F079, ... , S15F079, S01F095, ... S15F447]
```

Table 7: Format of the input frame to the first corner turner

Data is written and read back in units of a whole elementary integration period (integration frame). The integration frame length can be anything much larger than the fine channelizer filter length, *simeq*10 ms, but a short frame length would result in a bandwidth bottleneck between the correlator and the LTA. Assuming a frame length of 500 ms, the memory required to store one integration frame for 16 antennas is 11 GB. A 8 GB memory bank would store 0.39 ms of data for 16 antennas.

The corner turner reads back a whole time segment, for 16 coarse frequency channels, one station, two polarizations. Then it reads the same frequency channels for the second antenna, and in turn all antennas for the given set of 16 frequency channels. Then a new set of coarse frequency channels are processed. In this way the X correlator will receive only a single coarse frequency channel at a time, limiting the size of its internal corner turner memory.

```
<-           (n) clock cycles         - ><-    (n) clock cycles      ->
[S00F064T0, S00F064T1, ... S00F064T<n-1>][S02F064T0, ...  S02F064T<n-1>][S04...
[S01F064T0, S01F064T1, ... S01F064T<n-1>][S03F064T0, ...  S03F064T<n-1>][S05...
[S00F065T0, S00F065T1, ... S00F065T<n-1>][S02F065T0, ...  S02F065T<n-1>][S04...
...
[S01F079T0, S01F079T1, ... S01F079T<n-1>][S03F079T0, ...  S03F079T<n-1>][S05...
```

Table 8: Format of the output frame to the first corner turner

Starting time in each time segment is adjusted at a per antenna base in order to compensate for the gross delay. Delay step is 1 broad channel step, i.e. (sample time)/1024/ovs. It must also include the time required to fill the filter, i.e. successive integration frames slightly overlap in time.

4.2 Fine channelizer

The fine channelizer further divides each coarse node to the final frequency resolution. Minimum requested frequency resolution is given by the number of channels (256K) in the observed band (300 MHz), for a maximum channel width of 1150 Hz. For $f_s = 800$ MHz this corresponds to a minimum number of 800 channels in each coarse frequency channel. Using 1024 channels result in a total number of 344 k-channels, or a frequency resolution of 872 Hz

If, as proposed by Selex[**?**], $f_s = 1$ GS/s, the system would produce 275 k-channels.

Each node will have 32 parallel channelizer, processing in parallel 16 coarse frequency channels for 2 polarizations for one station. These are physically implemented as 8 radix-4 iserial FFT processors, each processing simultaneously 4 signals.

Output of the channelizer array is a set of 32 streams, representing 4 groups of broad frequency channels and 2 polarizations.

Each stream presents,(in order of index, from fastest to slowest):

- polarizations (2 in parallel)
- frequencies(broad) (16 in parallel)
- frequencies(fine) (1024/ovs = 896). Only the non overlapped portion of each broad channel is kept, and the overlapping fine channels are discarded
- time (length of time frame)
- antennas (as many as stored in the corner turner)

- frequencies (broad, in step of 16 broad channels)

An example of the first 3 parallel streams is shown in table 9. The frequency is denoted as `F<x>.<y>`, where x is the broad channel index, in the range $[64 \ldots 447]$, and y is the fine channel index, in the range $[6 \ldots : 959]$.

```
[S00F064.064T0,...S00F064.959T0][S00F064.064T1,...S00F064.959T1]...[...S00F064.959T<x>][S02...
[S01F064.064T0,...S01F064.959T0][S01F064.064T1,...S01F064.959T1]...[...S01F064.959T<x>][S03...
[S00F065.064T0,...S00F065.959T0][S00F065.064T1,...S00F065.959T1]...[...S00F065.959T<x>][S02...
...
```

Table 9: Format of the output frame from the second channelizer

Each stream is composed of individual sub-frames containing all fine channels for one time, antenna, polarization and broad frequency channel.

A fine delay correction is applied in the frequency domain to each frame. Max delay rate, for 100 km baseline, is 24 ns/s. Assuming a time segment of 0.5 s the delay variation in one segment is 12 ns or 1/100 of a broad channel sample time, and resulting phase variation is marginally important. The fringe rotation rate varies from 1.2 to 8.4 Hz across the bandwidth, and is more important. Frame-to-frame phase variation is up to 0.01 turns per fine time step, with a resulting decorrelation of $\simeq 3\,10^{-4}$.

Although the decorrelation is acceptable, a much better performance can be achieved by correcting the phase, with a finer time step, before the polyphase filter. The delay correction would then be composed by three parts (see fig. 4). Assuming the total (time varying) geometric delay is $\tau(t)$ we have:

- The gross delay, τ_0, in units of gross channelizer time step $1024/(f_s O_f)$ (1.12 ms in the baseline case). This delay is kept fixed during one integration period (hundreds of ms)
- The fine delay across the bandwidth. This is corrected in the frequency/time domain of the first channelization, with frequency and time step equal to those of the coarse channelized samples. The correction is $\exp(2\pi f_0 \tau_1)$, with f_0 to the nominal sky frequency of the channel, and $\tau_1(t) = \tau(t) - \tau_0$, and automatically includes the fringe frequency correction.
- The residual phase across each fine channel, $\exp(2\pi f_1 \tau_1)$, with f_1 the frequency of the fine channel with respect to f_0. As $f_1 \ll f_0$, this term can be updated ad the much longer time step of the fine channelizer.

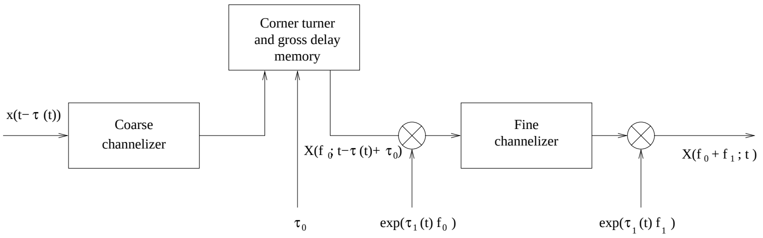

Figure 4: Delay correction

4.3 Post-processing

After the fine channelizer the signal incurs in a simplified version of the processing after the coarse channelizer, described in section 3.4. Total power measurement is not desirable, as it would require a large amount of memory and as it already occurs in the correlator (it is the autocorrelation spectrum). Also a calibration in phase/amplitude is best performed on the correlated data. What remains is:

- RFI detection/excision over the fine frequency resolution data. Again, this is basically a placeholder for the RFI processing designed in the RFI work-package
- Re-quantization: RFI immunity is no more necessary, as any channel still affected by strong RFI will produce invalid data. 8+8 bit quantization is required to leave room for strong spectral features ([4]).

4.4 F to X interconnect

Quantized data must be grouped to be sent to the X processor. The X processor is divided into n_x *correlation planes* (*X-planes*), with each plane processing all antennas for a subset ($1/n_x$ of the total) of the frequency channels. Each node must then send this subset of the frequency channels to all n_x X nodes. To reduce the number of links, it is convenient to group the nodes into n_F *frequency planes*, or *F-planes*, with n_n nodes per plane. Each link carries n_a/n_F antennas and $1/n_x$ frequency channels. A reasonable number of links occur for $n_F = n_X = \sqrt{n_a}$=32, with $n_a = 1024$ is the number of antennas.

With this architecture (fig. 5) each node produces n_X/n_n streams of data, containing samples from all antennas and $1/n_x$ of the total number of frequency channels. The channelizer corner turner already splits the coarse frequency channels into 16 separate streams, so each node produces 16 streams with 8 antennas.

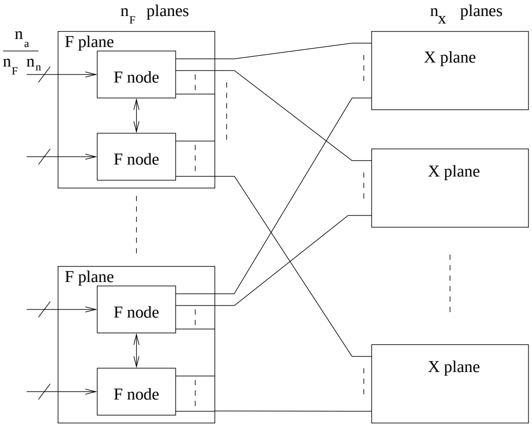

Figure 5: Interconnection between F and X sections of the correlator

Grouping 4 nodes will produce the required number of antennas per link. This is performed in two steps (fig. 6). Assuming 2 nodes will fit in a single FPGA, corresponding outputs from each node are first merged together. Each frame is divided in 2, and corresponding halves are grouped in a frame with data from two antennas, half the fine frequency channels, one coarse channel, one polarization and one time sample.

Then one of the two links is transferred to the adjacent FPGA, and merged in a link with the corresponding link from the other 2 antennas. Frames to the X plane then contain data for both polarizations, 4 antennas, half (448) fine channels (table 10). Each frame is 7168 bytes long. 8 consecutive frames contain data for all antennas, one time sample and 1/16 of the coarse channels (24 coarse channels).

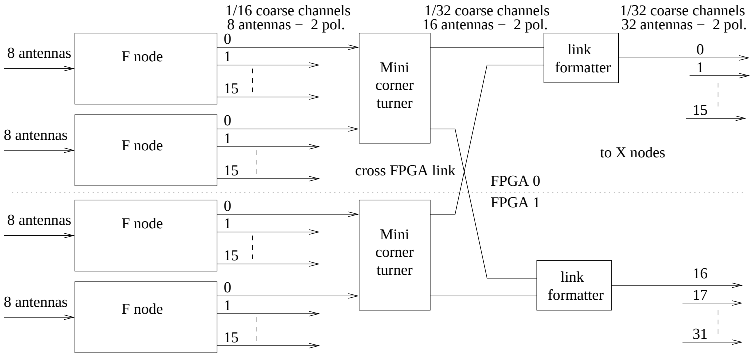

Figure 6: Grouping of signals from 4 nodes in a F-plane

Total bandwidth for each link is 300 MHz times sample size times 64 signals, i.e. 8.2 Gb/s for 8+8 bit samples.

```
link 0
<--       896x4 samples, 7168 bytes                 -->
[S00F064.064,...S00F064.511,S08F064.064,...S08F064.511]
[S01F064.064,...S01F064.511,S09F064.064,...S09F064.511]
[S16F064.064,...S16F064.511,S24F064.064,...S24F064.511]
[S17F064.064,...S17F064.511,S25F064.064,...S25F064.511]
link 1
[S00F065.065,...S00F065.511,S08F065.065,...S08F065.511]
[S01F065.065,...S01F065.511,S09F065.065,...S09F065.511]
[S16F065.065,...S16F065.511,S24F065.065,...S24F065.511]
[S17F065.065,...S17F065.511,S25F065.065,...S25F065.511]
...
link 16
[S00F064.512,...S00F064.959,S08F064.512,...S08F064.959]
[S01F064.512,...S01F064.959,S09F064.512,...S09F064.959]
[S16F064.512,...S16F064.959,S24F064.512,...S24F064.959]
[S17F064.512,...S17F064.959,S25F064.512,...S25F064.959]
...
```

Table 10: Format of the links from the F nodes to the X nodes

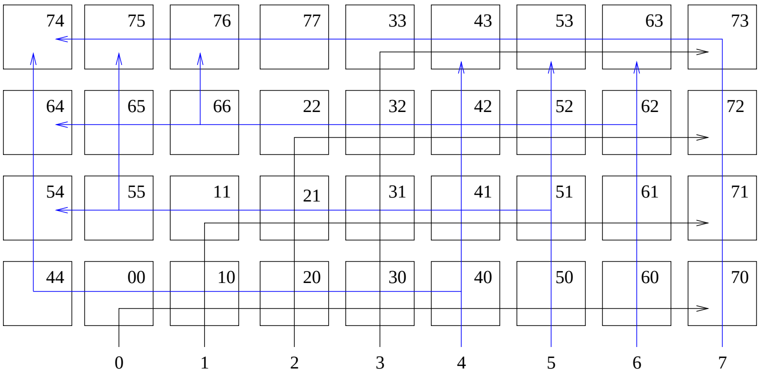

Figure 7: Structure of a systolic array correlator, 8 antennas

5 Central Signal processor - X section

The X section is only sketched here, as it is not part of the F processing. Two complete different architectures have been proposed:

- A systolic array. An array of $(N + 1) \times (N/2)$ of multiplier cells, that process incoming data and pass them on to the next (front or side) cell. It is conceptually simple, but requires lots of on-board accumulation memory, and a dense interconnection grid (fig. 7).
- A multiplexed array of multipliers, that are fed with a subset of all the antennas (Larry D'Addario chip architecture [5]). This has the advantage of not requiring any interconnection between different correlation cells.

Both architecture require data for all antennas to be presented simultaneously, and then require a corner turning stage.

5.1 Corner turning

A first level of corner turning is performed in the F node, in particular in the corner turner memory. Data received is 1/32 of the band, 32 antennas, 2 polarizations, 1 time sequence. Index order is:

- 32 antenna groups (on 32 links)

- polarization
- antenna index LS bit
- fine frequencies
- antenna index next bit
- time
- 8 antennas in group
- coarse frequencies

Output should be (fastest to slowest)

- polarization
- antennas
- time
- fine frequencies
- coarse frequencies

As data is already grouped in 24 coarse frequency (the slowest varying index is the coarse frequency), the corner turner must store only a single coarse frequency at a time.

This is a total of up to $\simeq$ 440 times, 1024 antennas, 2 polarizations, and 448 fine frequency channels, i.e. 800 MB of memory, i.e. less than 2 GB including double buffering. Memory bandwidth is 307 Gb/s, or 4 memory banks at 2.4 GT/s (read + write). Each bank can process 8 links in parallel, with records containing the corresponding sample for 16 antennas, both polarizations. All records for all 32 antennas are then read back in sequence.

5.2 Correlator

Using the multiplexed array structure proposed by D'Addario in [5] and revised in [6], the correlator cell is composed of a memory and a set of $n_m \times n_m$ multipliers. At the beginning of the integration the memory is loaded with a complete set of samples for all antennas and a single frequency channel. In our case the memory required is 2048 (antennas, polarizations) times 436 2 byte samples (in 0.5 s), i.e. 2 MB for double polarization. This may fit in the on-chip memory of a medium size FPGA, or in an external fast static memory chip.

The square matrix of multipliers is fed with samples from different subsets of the antennas (polarizations are seen as different antennas, 2048 "antennas" total), until all possible combinations have been correlated. For n_a antennas and n_m^2 multipliers, the number of passages is $(n_a/n_m+1)(n_a/n_m)/2$, e.g. 2080 for $n_m = 32$.

Each antenna/polarization produces a stream of 300 MS/s divided into 32 correlator planes, i.e. 9.375 MS/s per plane. Each sample must be processed 2080 times. Assuming a clock frequency of 410 MHz, 48 cells per plane are necessary. Using 64 cells would require 305 MHz multiplier clock speed.

Link speed from correlator to LTA is a bottleneck for large n_a. Each cell produces n_m^2 visibilities every integration period. For 436 samples per integration period (0.5 s integration time) and assuming $16 + 16$ bit visibilities, one gets $\simeq$ 10 bytes of output per clock cycle.

5.3 Long term accumulation

The baseline distribution for the LFAA is dominated by very short baselines, with 75% of the antennas within a diameter of 1 km, and only few stations at the maximum baseline of 100 km. This implies that half the visibilities can have a much longer (20 s) integration time, and only a small fraction actually need the minimum dump rate. Using a set of even just a few different dump times would reduce the data rate to the SDP by an order of magnitude.

The LTA can be implemented within the correlator FPGA. The total memory required is 8 times that of the corner turner for 1024 antennas, 0.5 s integration time, and visibility size 4 times the sample size, or 128 MB per correlation cell in a 64 cell plane, i.e. within the range of large static memories.

Spectral averaging can be implemented, if requested, by simply not resetting the accumulator in each multiplier between processing of adjacent frequency bands. Up to 488 frequency channels can be summed together. This processing can be performed in the SDP.

6 Pulsar binning

Unless used only for very slow pulsars, it must be implemented at the coarse channelization level. This requires multiple copies of a coarse channel to be switched across multiple polyphase filters. Polyphase filtering causes time broadening of the signal, so a detailed analysis is required to assess the effect on the timing. If resources are to be kept constant, this implies that only a fraction of the broad channels can be processed, for a constant product of bandwidth times bins.

7 Other utility modules needed in the system

- Internal test vector generators
- Link integrity tests
- Timing accuracy. Determine procedures and formats to have accurate timing at per sample level

A Polyphase filter for real valued time multiplexed signal

Samples from the ADC are real valued, and time multiplexed in groups of 32 parallel samples at frequency of $f_s/32$. Although conceptually the polyphase filtering algorithm is identical to that for a complex valued, non time multiplexed signal, several implementation details are needed to handle the particular case.

The filter-bank is composed of three elements:

- A polyphase filter for a sequence that is partly parallel (the time multiplexed samples) and partly serial. Basically the filter handles a serial/parallel matrix of $32 \times N$ samples, multiplies it by a $32 \times k \times 1024$ tap coefficients (with $k \approx 20$ is the length of the filter for each sample), and produces a matrix of 32×32 convolved samples to be interpreted as a time sequence of 1024 convolved samples.
- A parallel/serial FFT engine, with 16 complex parallel streams and 32 serial samples per stream, that performs the FFT of the 1024 real samples interpreted as 512 complex samples
- A final recombination stage, that transforms the 512 FFT outputs into the equivalent positive frequencies of a 1024 point FFT of the original real samples

These three components are described in the following chapters.

A.1 Polyphase filter

The polyphase filter computes the sum $x_j = \sum_k s_{j0+j+Nk} h_{j+Nk}$, where $j = 0 \ldots (N-1)$ is the index of the input of the FFT (of size N), j_0 is the sample corresponding to the beginning of the filter (different for subsequent FFTs) and k is the index of each individual filter. h_j represent filter taps for the prototype channel pow pass filter.

For time multiplexed samples, with time multiplexing factor of n_m, we have n_m parallel filters, with the structure described in figure 8 (with 5 taps per filter). In the limiting case of $n_m = N$ the structure is the one described e.g. in Harris[8]. For $n_m = 1$ (no time multiplexing) this is just a single serial structure.

The delay blocks in fig. 8 are usually implemented as FIFO memory blocks. The filter coefficients are indicated for the first sample in each block of N/n_m. Subsequent samples are multiplied by the subsequent filter taps. In our case, with $n_m = 4$ and $N = 1024$ each FIFO delays the signal by 256 clock cycles.

If the polyphase filter-bank implements an overlap between adjacent frequency channels, two modifications are necessary

- Each delay block *rewinds* by the appropriate number of samples between FFT frames ($1 - 1/O_f$ its length).
- A dual port memory is present at each output stage, to rotate the FFT frame as in fig. 34 of Harris[8].

For example, in our case with $O_f = 8/7$, each FIFO rewinds by 32 samples between successive frames. The output frame is rotated by successive multiple of 32 samples at each frame.

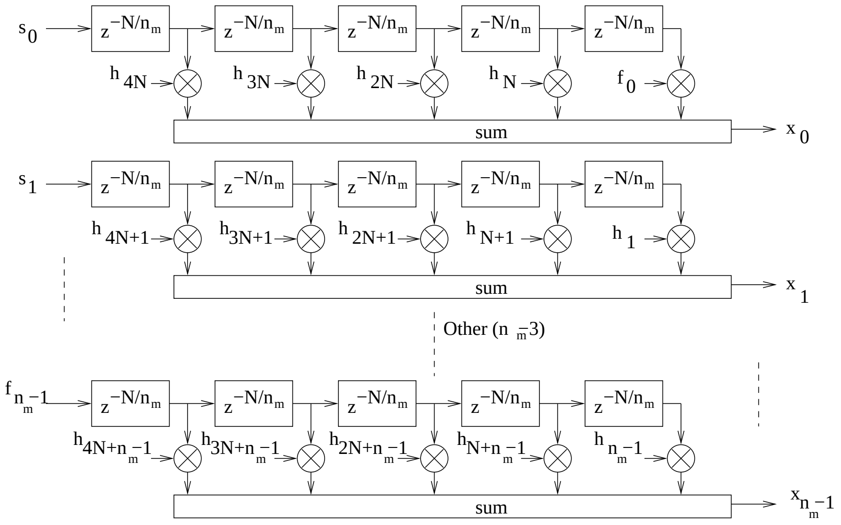

Figure 8: Polyphase filter for a time multiplexed signal

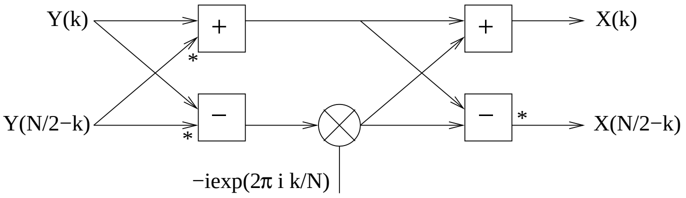

Figure 9: Complex FFT to real FFT stage

A.2 FFT of a real sequence

The ADC produces real samples. Several algorithms exist for producing a FFT of real valued data, that reduce the number of required operations by about a factor of 2. One can see for example the `rfft()` routine in Numerical Recipes[9].

The basic idea is to compute the FFT of a sequence $y_j, j = 0 \dots N/2 - 1$ derived from the original time series $x_j, j = 0 \dots N - 1$, with $y_j = x_{2j} + ix_{2j+1}$. If Y_k is the transform of y_j, with

$$Y_k = \sum_{j=0}^{N/2-1} y_j \exp\left(2\pi i \frac{2jk}{N}\right) = \sum_{j=0}^{N/2-1} (x_{2j} + ix_{2j+1}) \exp\left(2\pi i \frac{2jk}{N}\right) \tag{1}$$

Considering that $Y_{-k} = Y_{N/2-k}$, one can easily show[9] that

$$X_k = \frac{1}{2}\left(\left(Y_k + Y^*_{N/2-k}\right) - iexp\left(2\pi i \frac{k}{N}\right)\left(Y_k - Y^*_{N/2-k}\right)\right) \tag{2}$$

$$X_{N/2-k} = \frac{1}{2}\left(\left(Y_k + Y^*_{N/2-k}\right) + iexp\left(2\pi i \frac{k}{N}\right)\left(Y_k - Y^*_{N/2-k}\right)\right)^* \tag{3}$$

$$\tag{4}$$

The overall processing can be then divided into two butterfly (sum/difference) operations, and a complex multiplication by the twiddle factor $-iexp\left(2\pi i \frac{k}{N}\right)$. This is shown in figure 9.

A.3 FFT for a time multiplexed signal

The FFT of a time multiplexed sequence can be decomposed into a set of serial and parallel radix-2 and radix-4 stages. We adopted a decimation-in-frequency structure, as it is more easily adapted to a long sequence of successive samples. The overall structure of the FFT block is shown in fig. 10.

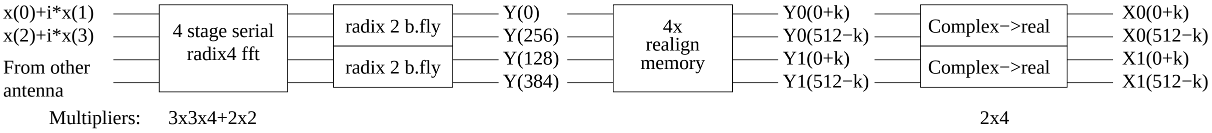

Figure 10: Structure of a 1024 real points FFT for a ×4 multiplexed signal

Each parallel signal is first processed in a serial 256 point FFT. The bank of FFT is implemented with 4 successive stages of serial radix-4 FFT followed by one stage of parallelradix-2 FFT. As a radix-4 FFT analyzes simultaneously 4 different signals, each FFT block performs the channelization for 2 independent antennas. Channel and sample ordering after these three stages are scrambled, due to bit reversal with different bases, and need a memory transpose block before the next parallel FFT (not shown). Twiddle factors are different for each of the 2 FFTs, to reflect the fact that these are parts of a larger 512-point FFT. The last radix-4 stage has twiddle factors that are always multiple of $\exp(i\pi/8)$, with many factors just multiple of $\exp(i\pi/4)$. A total of 4 real multipliers are required for this stage. Other stages, with more generic twiddle factors, require 3 complex multipliers per stage.

The parallel FFT takes 2 complex values at each clock cycle and produce 2 frequency channels. It is a simple butterfly stage, with no multipliers.

References

[1] K.Z. Adami: ”Low correlator system architecture”, SKA CSP TIM1 meeting, Auckland, February 2014

[2] G. Comoretto: ”Structure and processing of the F section for SKA”, SKA CSP Report xxx, 2014

[3] G. Comoretto: ”SKA Project Series - Effects of the bandpass slope on the SKA Low quantization”, Arcetri internal report 7/2013
`http://www.arcetri.astro.it/ricerca/rapporti-tecnici/205-reports/692-13-7`

[4] G. Comoretto: ”Quantization of Gaussian noise and RFI immunity”, SKA CSP report 9, 2014

[5] L. D’Addario: ”LowPower Correlator Architecture for the MidFrequency SKA”, SKA Memo 133, March 2011. // http://www.skatelescope.org/publications/

[6] B. Carlson: ”A scalable correlator chip architecture for the SKA, based on D’Addario’s correlator building block”, SKA CSP Memo 0003, December 19, 2013.

[7] T. Cornwell: “SKA Level 1 Requirements - rev. 3”, Feb. 24, 2014

[8] Fredric J. Harris, Chris Dick and Michael Rice: “Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications”, *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4 (2003)

[9] ”FFT of real functions”, section 12.3 in *Numerical Recipes - The Art of Scientific Computing*, Cambridge University Press, 3rd edition (2007)