An FPGA Based Phased Array Processor for the Sub-Millimeter Array

Vinayak Nagpal *

advised by Jonathan Weintroub †

September 5, 2006

*Chalmers University of Technology, Gothenburg, Sweden

†Harvard Smithsonian Center for Astrophysics, Cambridge, MA

Contents

1 Introduction **6**
- 1.1 Science Goals . . . 6
- 1.2 Project Objectives . . . 6
- 1.3 Project Partners . . . 6
- 1.4 The iBOB and iADC Hardware Platform . . . 7
- 1.5 Development Platform . . . 11
  - 1.5.1 DSP Logic Design . . . 13
  - 1.5.2 Interface Logic Design . . . 13
  - 1.5.3 Embedded Software . . . 13
  - 1.5.4 Design Synthesis . . . 14
- 1.6 The Submillimter Array and IF/LO subsystem . . . 14

2 The SMA Phased Array Processor **14**
- 2.1 Generic Phased Array . . . 14
  - 2.1.1 Time Domain Phased Array Processing . . . 15
  - 2.1.2 Frequency Domain Phased Array Processing . . . 17
- 2.2 System Architecture . . . 18
- 2.3 Analog Subsystem . . . 18
  - 2.3.1 IF/LO Subsystem . . . 18
  - 2.3.2 Phase Switching . . . 19
  - 2.3.3 Fringe Rate Correction . . . 21
- 2.4 Mark *Vb* Digital Backend (DBE) . . . 22
- 2.5 Digital Delay Lines . . . 23
  - 2.5.1 Coarse Delay . . . 25
  - 2.5.2 Fine Delay . . . 25
  - 2.5.3 Super Fine Delay . . . 26
- 2.6 Delay Line Results . . . 31

3 Beamformer Calibration **37**
- 3.1 Design of Calibration Correlator . . . 38
- 3.2 Calibration Correlator Results . . . 39
- 3.3 Final Architecture . . . 44

4 Discussion and Future Work **44**

List of Figures

1 Picture showing iBOB. 7
2 Picture showing iADC. 8
3 Picture showing iBOB with 2 iADCs and 2 Infiniband cables. . . . 8
4 Symbolic Diagram for iBOB + 2 iADCs + 2 Infiniband links 9
5 Interface Diagram for iBOB + 2 iADCs 10
6 Summary of BEE2 development platform 12
7 Simulink Screenshot. 12
8 Simple two element phased array 15
9 Simple Heterodyne two element phased array 16
10 Actual Heterodyne two element phased array 16
11 Time Domain Beamformer 17
12 Frequency Domain Beamformer 18
13 Symbolic System Architecture 19
14 SMA IF/LO Subsystem 20
15 Mark Vb DBE Symbolic System Diagram 22
16 Mark Vb DBE Block Diagram 23
17 Architecture of 1 Delay Line iBOB Design 24
18 Coarse Delay 25
19 Fine Delay 26
20 Digital 10 tap filter for $D = 3$ 27
21 Digital 10 tap filter for $D = 3.3$ 27
22 Simple Finite Impulse Response Digital Filter Implementation . . . 28
23 Demux-by-4 FIR filter tap implementation 29
24 Demux-by-4 Finite Impulse Response Digital Filter Implementation 29
25 Double Buffering Scheme shown for a 3 tap filter. 30
26 Snapshot of two data channels and their average when they are phase aligned 32
27 Output of MATLAB function xcorr with 8000 samples of 2 aligned channels 33
28 Snapshot of two data channels and their average when one is delayed digitally by 55.5 ns 34
29 Output of MATLAB function xcorr with 8000 samples of 2 channels when one is delayed by 55.5 ns 35
30 Zoom into peak of Fig.29 36
31 Signal flow to SMA correlator and phased array processor 37
32 Architecture of *Calibration Correlator* 38
33 Autocorrelation Spectra (∞ Signal to Noise) 40
34 Cross correlation Spectra with -2 dB Signal to Noise 41
35 Cross correlation Spectra with -9 dB Signal to Noise 41

36 Cross correlation Spectra with −9 dB Signal to Noise 42
37 Cross correlation Spectra with −9 dB Signal to Noise 42
38 Cross correlation Spectra with no cable delay 43
39 Overall System Architecture 45
40 Final Architecture 46

Abstract

It has been widely acknowledged that Very Long Baseline Interferometry (VLBI) in the submillimeter wavelengths can make imaging observations of super massive black holes possible. The Sub-Millimeter Array (SMA) along with the James Clerk Maxwell Telescope (JCMT) and Caltech Submillimeter Observatory (CSO) on the Mauna Kea summit in Hawaii can together provide a large collecting area as one or more stations for VLBI observations aimed at studying an event horizon. To work as a VLBI station with full collecting area the SMA (or a combination SMA, JCMT, CSO antennas) would need a processor to enable phased array operation. This masters project focusses on building such a processor.

Back end processing for high bandwidth radio telescopes has traditionally been done using custom designed application specific integrated circuits (ASIC). Recent advances in Field Programmable Gate Array (FPGA) technology have made FPGAs both powerful and economically viable for radio astronomy back ends. We have attempted to take advantage of these advances and built a proof-of-concept 500 MHz phased array processor for the SMA using FPGAs. The phased array processing is done in the time domain using high speed sampling and digital delay lines. The design is capable of spooling the phased sum to a Mark Vb VLBI data recorder. It is based on hardware built by the Berkeley Wireless Research Center and the Berkeley Space Science Laboratory.

We digitize signals after the 1^{st} SMA downconvertor using 1024 MHz sampling and have demonstrated the capability to sum signals from 8 antennas through programmable digital delay lines up to a precision of $\approx 1/10$ the sampling rate i.e. 0.1 ns. To calibrate geometric, atmospheric and instrument delays for accurate phasing, a single baseline 512 MHz 32 channel FX correlator has also been designed to fit on a single FPGA chip.

1 Introduction

1.1 Science Goals

The primary scientific goal of recent work to extend Very Long Baseline Interferometry (VLBI) into the sub-millimeter regime is an imaging observation of the event horizon of a black hole [2]. In this context, the sources most likely to be studied are $SgrA*$ and $M87$. VLBI at 0.8 mm wavelength has the potential to image up to 20 micro-arc second angular resolution. There is also a radiative transfer advantage obtained due to reduced electron scattering. Therefore we have a strong case to retrofit the Sub Millimeter Array (SMA) with a phased array processor and VLBI recording interface, thereby enabling it to participate in such VLBI observations with its full collecting area.

1.2 Project Objectives

To accommodate the development of a more or less complete system within the purview of a masters thesis it was decided to build a proof-of-concept system which would take IF signals from 8 antennas. In principle these could include any combination of SMA/JCMT/CSO antennas. We decided to limit ourselves to single polarization data and only 500 MHz (of the available 2 GHz) bandwidth. The objective was to compute in real time the phased sum of these antennas and spool the result to a Mark Vb VLBI recording unit and take care of the various calibrations involved. To cut short the design time we decided to use the iBOB FPGA boards and iADC sampling boards built by the Center for Astronomy Signal Processing and Electronics Research (CASPER) [1] group at Univ. of California Berkeley. The time schedule for the project was fixed at 10 months.

1.3 Project Partners

The CASPER [1] team at UC Berkeley is working extensively towards developing FPGA based technology for accelerating and standardizing the development

[1] CASPER works in collaboration with the Berkeley Wireless Research Center, Space Science Laboratory and Radio Astronomy Laboratory at UC Berkeley

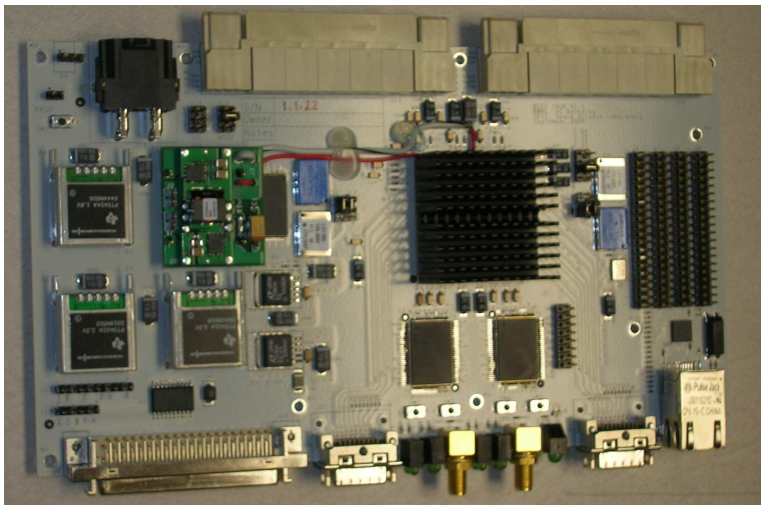

Figure 1: Picture showing iBOB.

of back ends for radio telescopes. The CASPER paradigm focusses on building general purpose FPGA based hardware boards and provide an extensive library of pre designed blocks which can be used to quickly and efficiently design digital subsystems commonly required by radio telescopes.
This paradigm was being deployed by the Massachusetts Institute of Technology Haystack Observatory in next generation digital back end for their Mark Vb data VLBI storage equipment. The proof-of-concept SMA phased array processor also fitted very well into the capabilities of CASPER boards. In addition it was found that interfacing the processor with VLBI storage equipment would also become very easy if both sub systems used the same hardware platforms.
This masters project thus involved extensive collaboration with CASPER in terms of acquiring training/technical support also with MIT/Haystack for interfacing with VLBI storage equipment.

1.4 The iBOB and iADC Hardware Platform

The iBOB and iADC boards are a part the BEE2 (Berkeley Emulation Engine) FPGA platform. The BEE2 platform was developed at the Berkeley Wireless Research Center(BWRC) primarily for applications requiring multi tera-flops of processing power and for emulating multi-processor computer architectures. For

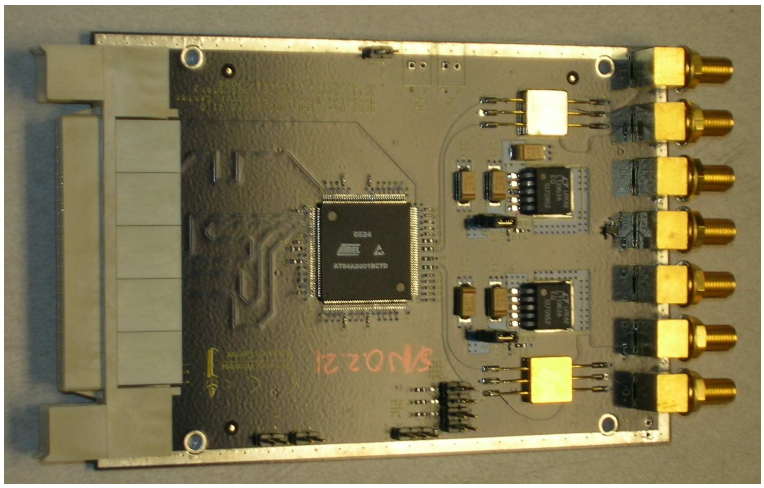

Figure 2: Picture showing iADC.

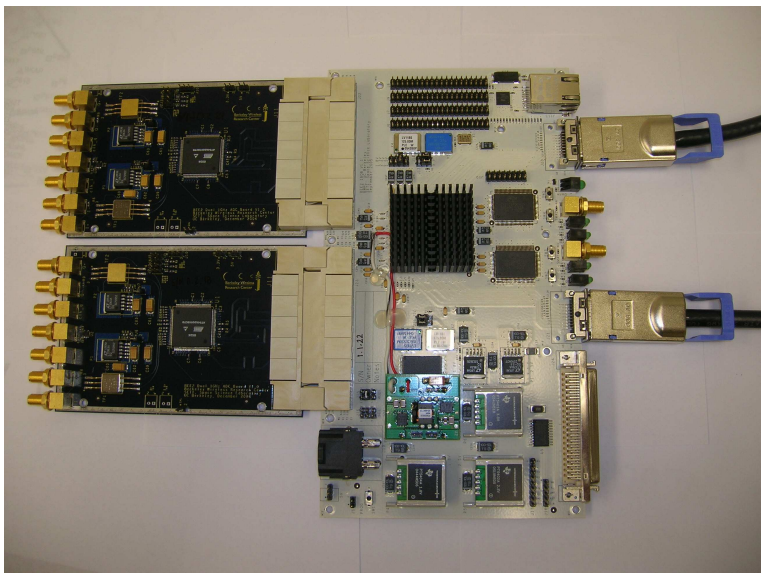

Figure 3: Picture showing iBOB with 2 iADCs and 2 Infiniband cables.

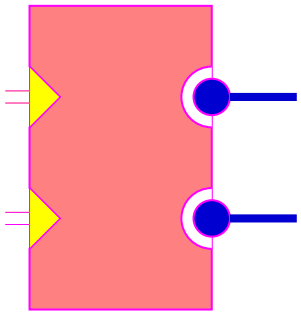

Figure 4: Symbolic Diagram for iBOB + 2 iADCs + 2 Infiniband links

the purpose of this project we have used the iBOB and iADC boards which are add-ons to the BEE2 suite. The iBOB is equipped with a Xilinx Virtex II Pro (vp50) FPGA and high speed data interfaces (Infiniband connectors). The iADCs are smaller boards which plug directly onto the iBOB and provide fast sampling using an Atmel analog to digital conversion chip. A single iADC can provide 2 GHz sampling for one channel or 1 GHz sampling for 2 data channels. One iBOB can mount 2 such iADC boards. Figure 1 shows a photograph of one iBOB, Figure 2 shows an iADC and Figure 3 shows a iBOB +2 iADC setup. The various interfaces of this setup are shown in Fig. 5. Brief descriptions of various components and interfaces shown in the diagram are listed below and Figure 4 shows symbolic representation of a iBOB+2 iADC+2 Infiniband setup.

1. RS-232 Serial Port: This is used to communicate control instructions to FPGA design from an external computer.

2. JTAG Connector: (Joint Test Action Group) JTAG interface is used for loading designs into the FPGA or for burning the EPROM which stores a default design loaded into the FPGA at power-on. In this project we have cascaded 3 iBOBs into a single JTAG chain to access all FPGAs using one programming cable.

3. Infiniband Connectors: Each iBOB is equipped with 2 Infiniband connectors. Infiniband is a high speed bi-directional serial bus. The BEE2 platform uses these in a 10 Gbps data rate configuration (Infiniband supports maximum of

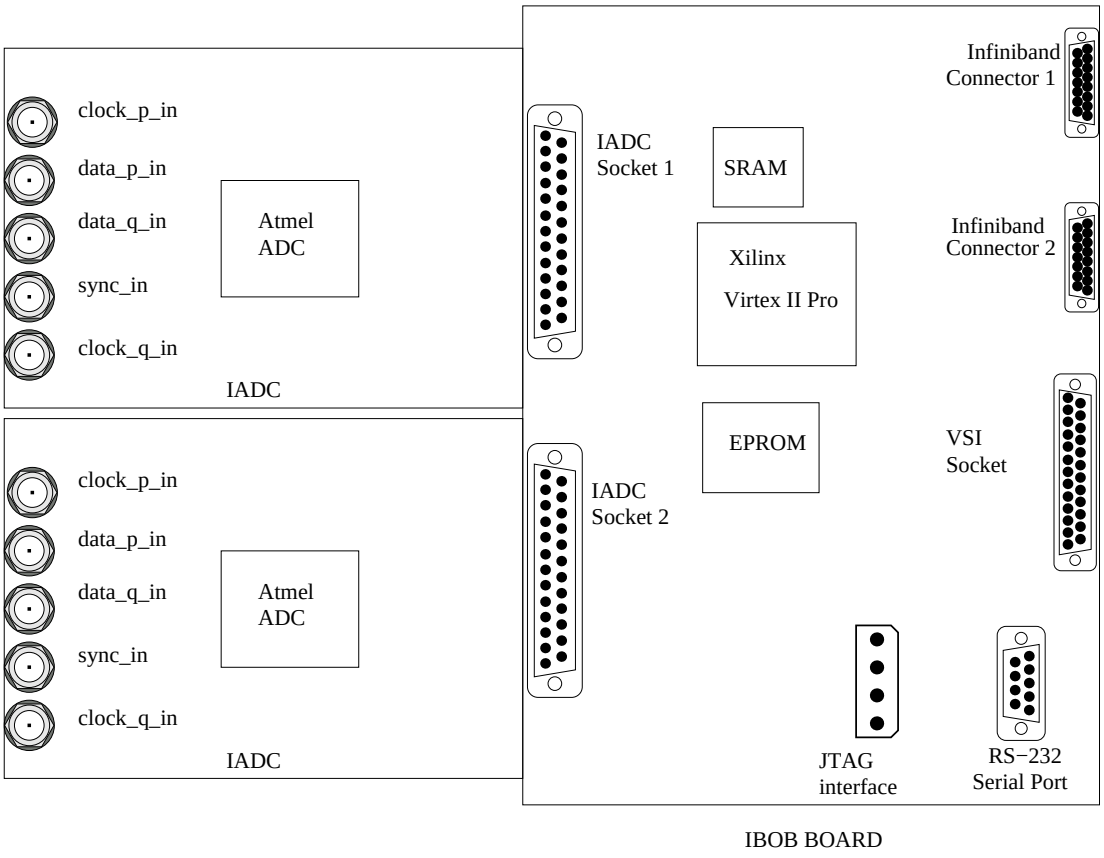

Figure 5: Interface Diagram for iBOB + 2 iADCs

120 Gbps). The interface derives its clock from a 156 MHz crystal oscillator provided on-board. *Rocket IO* components available within the Virtex II Pro provide the physical layer to drive these links. The transport protocol deployed is based on the IEEE 802.3ae 10 Gb Ethernet specification also called *X* (Roman Numeral 10) *Attachment Unit Interface* or *XAUI*. The BEE2 platform uses a proprietary XAUI core licensed from Xilinx.

4. VSI Connector: The Versatile Scientific Interface (VSI) bus is the standard interface adopted for VLBI and directly plugs into the Mark Vb data storage modules. The VSI interface logic is designed into the FPGA as a BWRC library component.

5. iADC: The Atmel (AT84AD001) sampling chip can be configured from the FPGA. The `clock-in` ports on the iADC are driven with a -6 dbm sine wave clock of 1024 MHz. This derives the FPGA clock using a divide-by-4. The FPGA processes 4 data samples per clock (demux-by-4) at a clock rate of $1024/4 = 256$ MHz. The `sync-in` inputs allow a synchronization pulse to provide alignment markers for multiple iBOB designs. The circuitry to interface with and operate the ADC from the FPGA is provided by BWRC as a standard library component.

6. SRAM: The FPGA can access a SRAM chip in addition to its own memory for off-chip storage.

1.5 Development Platform

Complementing the BEE2 hardware platform is an improved FPGA development environment. The board specific details are masked from the logic designer by providing highly parameterized library components for all input/output interfaces. The BEE2 development platform is summarized in Figure 6.

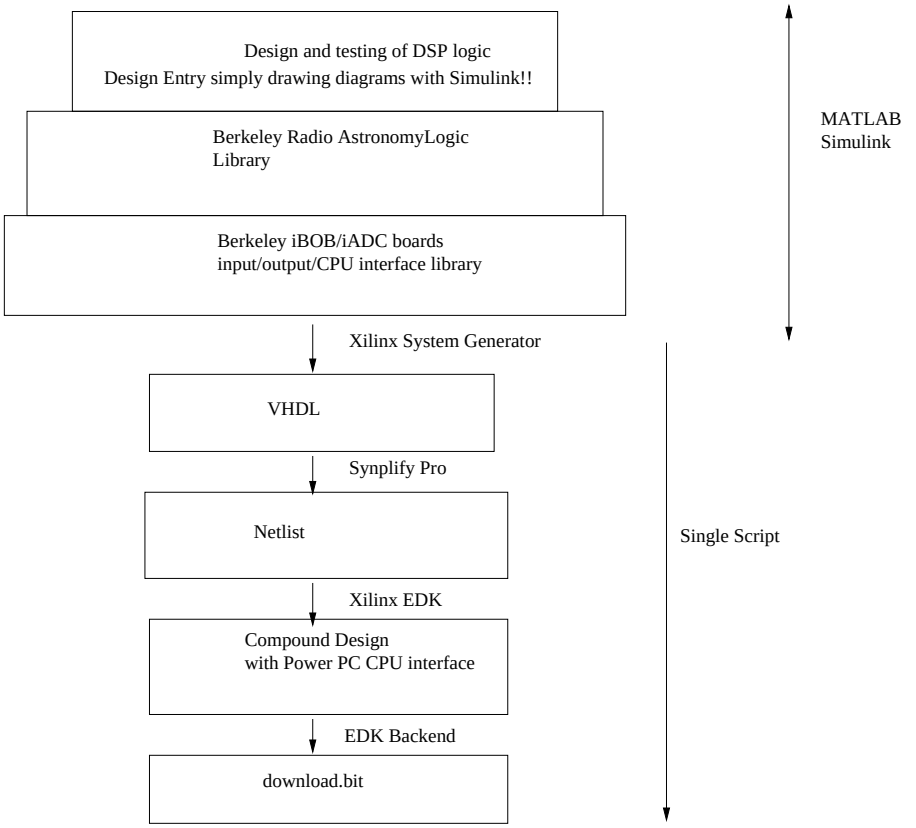

Figure 6: Summary of BEE2 development platform

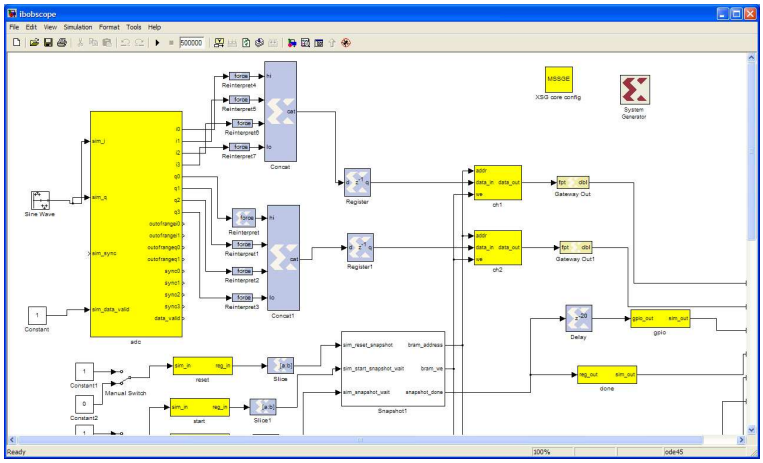

Figure 7: Simulink Screenshot.

1.5.1 DSP Logic Design

The DSP logic design is done using the Xilinx System Generator Block set for MATLAB/Simulink. This reduces the logic design task to that of drawing diagrams using basic hardware building blocks which can be simulated and tested in the Simulink environment. The designs can be later compiled into synthesize able VHDL using Xilinx System Generator. A screenshot of Simulink screen with Xilinx components (blue blocks) is shown in Figure 7.

1.5.2 Interface Logic Design

The FPGA interfaces like iADC, SRAM, PPC (refer item 1.5.3 below) registers, PPC Shared RAM, XAUI connectors, VSI bus and general purpose I/O (GPIO) units like LEDs, switches are done using BWRC library components which are in the form of Simulink blocks. These blocks serve a dual purpose. They provide a Simulink model of the interface for the logic design and testing phase and also provide a logic circuit to take care of the actual interfacing details during the compilation/synthesis phase. The logic circuit replaces the Simulink model automatically if the design compilation is invoked using BWRC scripts. Logic interface blocks can be seen as yellow blocks in Figure 7.

1.5.3 Embedded Software

Virtex II Pro offers two on chip 32-bit Power PC (PPC) processors in addition to the reconfigure-able fabric. The FPGA can be configured to bridge the memory and data busses of these processors to the clock domain of the designer's custom DSP logic. This enables the PPC to interact with DSP logic by read/write operations on certain registers/RAMs. BWRC has designed a small operating system to run on these PPCs called *Tiny Shell*. Tiny Shell provides a basic command interpreter and serial port driver which is used to communicate with an external computer for sending control information.

1.5.4 Design Synthesis

Compilation scripts (provided by BWRC) take care of the underlying details of replacing Simulink models with interface circuitry. They also generate the required constraint files, embedded software configurations and invoke various back-end tools automatically with the required parameters. In effect the compilation of Simulink designs into VHDL, then to net-lists, mapping of busses in Xilinx Embedded Design Kit (EDK) and back-end synthesis are all reduced to a single click operation.

1.6 The Submillimeter Array and IF/LO subsystem

The Sub-MM Array (SMA) [5] is an 8 element interferometer operating in the range of $200 - 900$ GHz on the Mauna Kea summit in Hawaii. Each element is a steerable smooth parabolic reflector antenna having a diameter of $6m$. The antennas can be moved between pads to provide different size array configurations. The longest baselines obtainable are about $0.5km$. The total bandwidth available with SMA receivers is 2 GHz. Our phased array processor is proposed to tap into the SMA signal chain after the first down converter which presents 1 GHz signal bandwidth centered at about 1 GHz. We sample this data at a rate of 1024 MHz allowing a Nyquist bandwidth of 512 MHz.

2 The SMA Phased Array Processor

2.1 Generic Phased Array

A phased array is a group of antennas in which the relative phases of the respective signals received at the antennas is varied in such a way that the voltage sum of these signals causes the effective radiation pattern of the array to be reinforced in a desired direction. For a single baseline interferometer a simple phased array can be seen in Figure 8. The phase plot shows the phase difference between the two received signals as a function of frequency. A delay τ corresponds to the slope $\frac{\phi}{f}$ in the phase plot. In a heterodyne receiver the sky signal is down converted to an intermediate frequency as shown in Figure 10. This introduces an effect if the

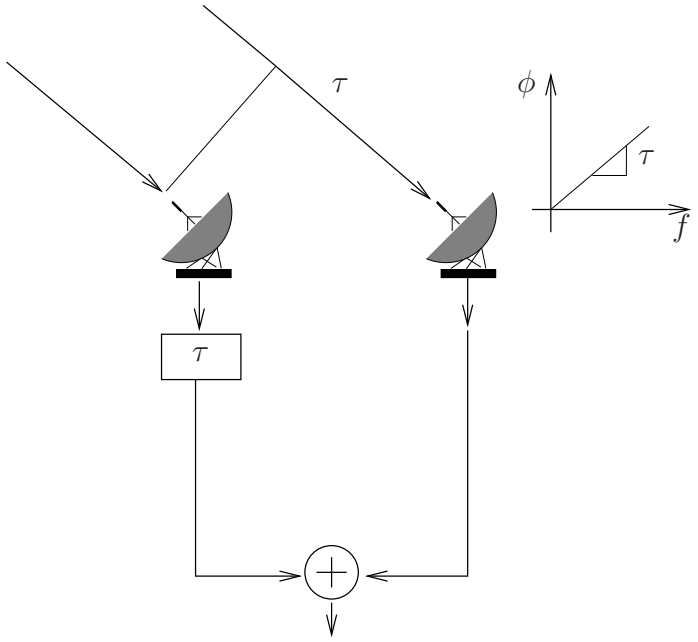

Figure 8: Simple two element phased array

local oscillator phases at two antennas is slightly different. As seen in the phase plot this causes a flat phase shift over the entire frequency band corresponding to $\phi_1 - \phi_2$. This effect cannot be corrected using delay elements alone. If this effect is not corrected in the analog domain by carefully adjusting LO phases the backend needs to use frequency domain techniques to correct for this phase shift. The SMA first LO's are phase programmable and the correlator software can adjust the LO phases such that $\phi_1 - \phi_2 = 0$. The τ delay compensation must be done after down conversion and is not exact because the compensation is done at a higher wavelength λ_{if} instead of the actual λ_{sky}. This can be corrected during fringe rotation explained in section 2.3.3. The delay line compensation can be achieved using two equivalent approaches.

2.1.1 Time Domain Phased Array Processing

Figure 11 shows the time domain approach which utilizes a variable delay line per antenna. This simple approach can compensate delays only (not phase offset effects). The accuracy or phase coherence performance of the time domain

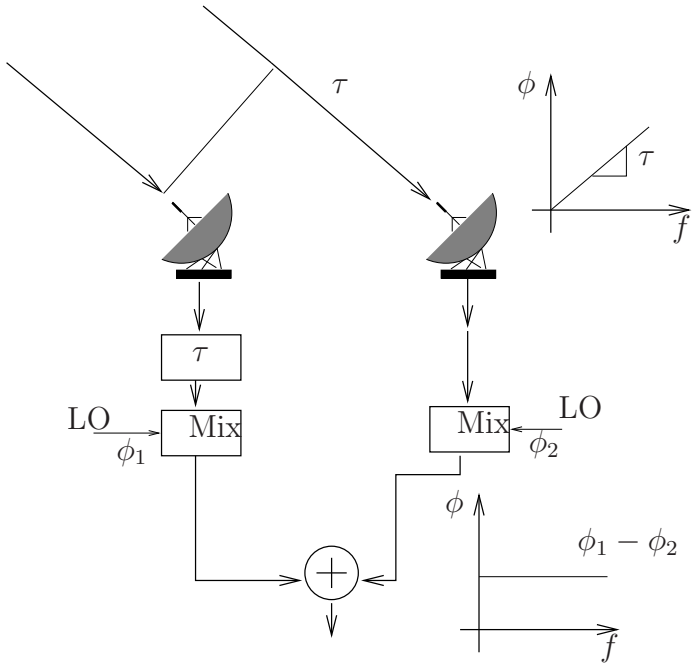

Figure 9: Simple Heterodyne two element phased array

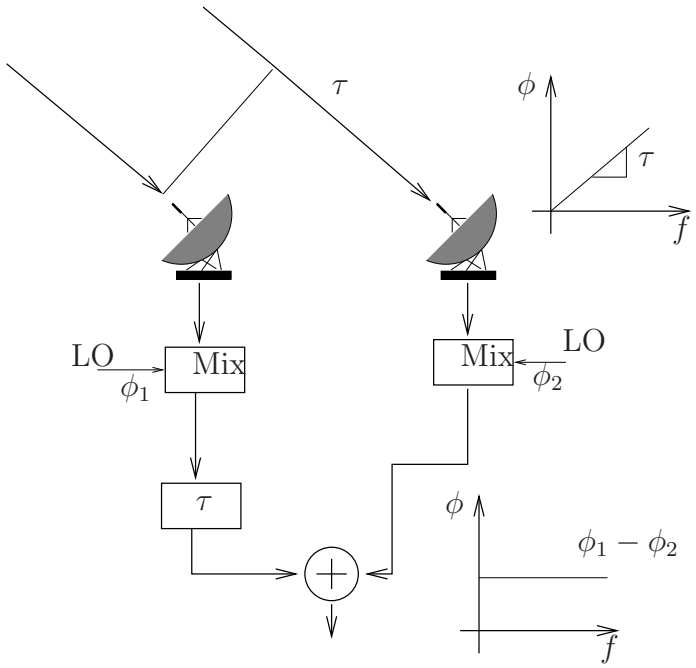

Figure 10: Actual Heterodyne two element phased array

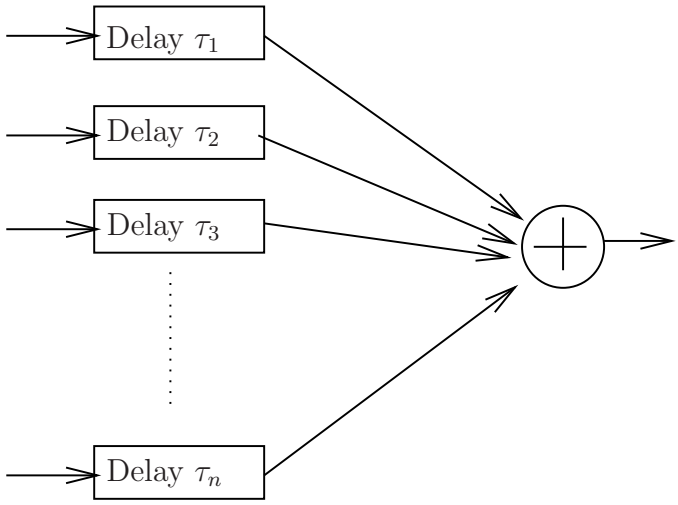

Figure 11: Time Domain Beamformer

beamformer depends on the delay step size. The smaller the delay step τ_{min} is compared to the signal bandwidth B the better will be the phase coherence of the beam formed signal.

$$\tau_{min} << \frac{1}{2B} \tag{1}$$

2.1.2 Frequency Domain Phased Array Processing

Figure 12 shows frequency domain delay lines. The ϕ adjust blocks adjust the channel phase such that the phase response will have the appropriate delay slope. In addition to delays this approach can compensate for fixed and variable phase offsets in the system like the LO phase difference described earlier. The phase coherence of the frequency domain beamformer improves by increasing the number of frequency channels N. A big disadvantage of this approach is the need to have multiple (1 per antenna) FFT and Inverse FFT blocks which are computationally expensive. For the SMA phased array processor we have chosen to use the time domain approach because the phase offset effects can be corrected using existing SMA analog subsystems.

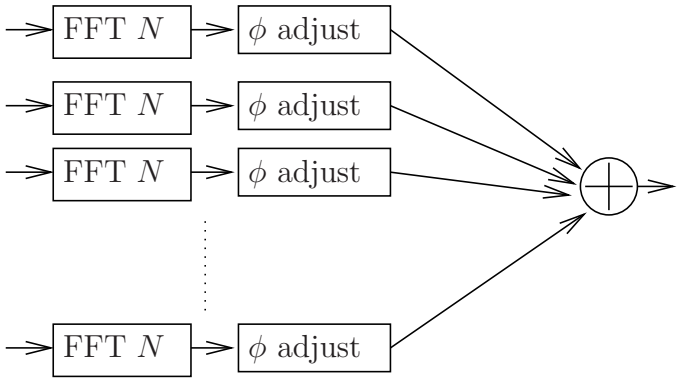

Figure 12: Frequency Domain Beamformer

2.2 System Architecture

The SMA phased array processor is built upon building blocks described in the previous sections. A system architecture diagram showing various components and interconnects is shown in Figure 13. The entire system can be implemented on 4 iBOBs and 4 iADC boards. The first 2 iBOBs implement identical designs that implement digital delay lines and addition logic. The 3^{rd} iBOB implements the digital back-end *DBE* for Mark Vb storage unit and the 4th iBOB is used for system calibration. The symbolic representation for the system (without calibration) is seen in Figure 13.

2.3 Analog Subsystem

2.3.1 IF/LO Subsystem

Figure 14 shows the distribution of bandwidth at the IF input to ADCs. The SMA 1^{st} downconverter output presents 1 GHz of bandwidth centered at ≈ 1 GHz. We can choose either of two 500 MHz bands from this using any one of two block filters which were custom ordered for this purpose. The available Nyquist bandwidth with 1024 MHz sampling is 512 MHz, we utilize most of this by deploying anti-aliasing filters having half-power bandwidth of 480 MHz centered at 760 MHz or 1280 MHz at the IF output. The filter choice determines which band we select as shown in Figure 14. It can be seen that the sampling clock appears on opposite

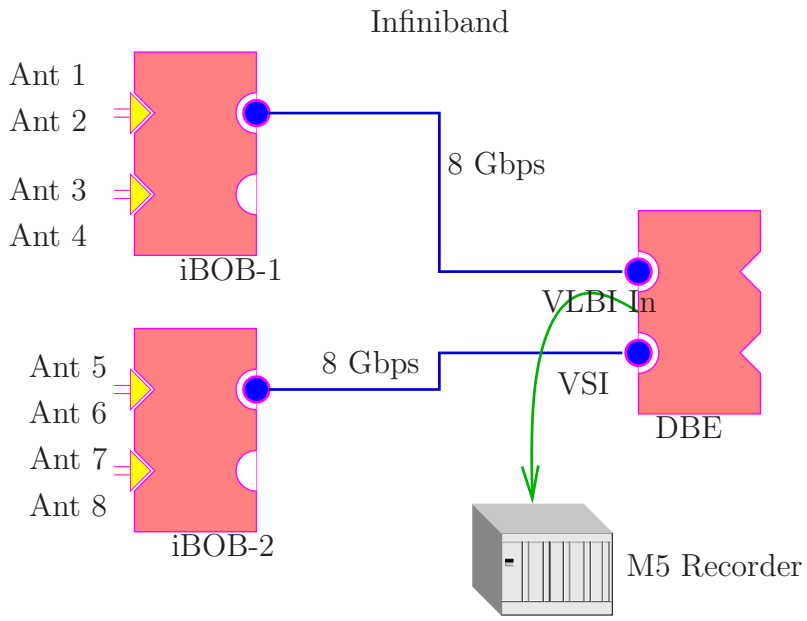

Figure 13: Symbolic System Architecture

edges of the two available bands causing the band to show a frequency flip in one configuration. The choice of 1024 MHz for sampling was dictated by the fact that the MIT/Haystack Mark *Vb* recording subsystem is designed to work at this rate.

2.3.2 Phase Switching

Periodic switching of LO phase is used routinely in most interferometers to ensure that the interferometer output is zero when the input signals have no correlation. The SMA has two levels of phase switching and atleast the 180° switching is essential for proper operation of the SMA correlator [4]. Submillimeter receivers cannot have low noise amplifiers and side band separating filters working at the sky frequency in their front end, their first stage is usually a mixer. In a regular heterodyne receiver front end LNA's and filters (at RF) allow easy separation of the lower and upper sidebands, however as submillimeter receivers have a mixer in their first stage alternate methods are used for sideband separation. In the SMA this is achieved using 90° phase switching and appropriate addition/subtraction to cancel out one of the sidebands. In addition to this 180° phase (Dicke) switching is used for cancelling offset and leakage effects. These two switching functions

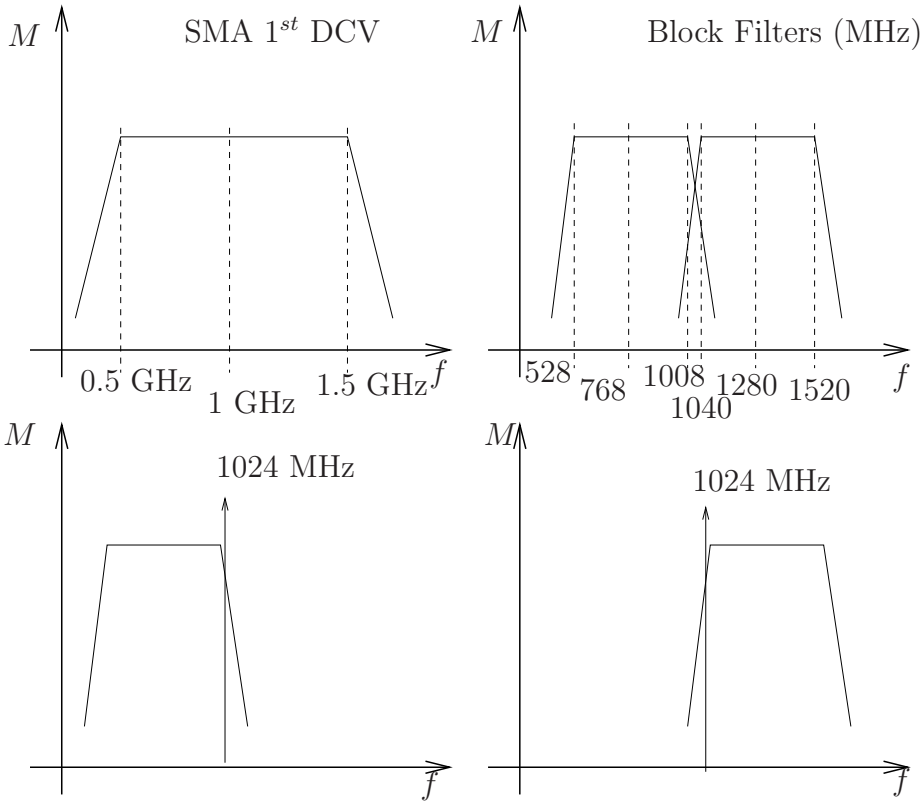

Figure 14: SMA IF/LO Subsystem

are achieved using a fast 90° Walsh sequence based phase switch superimposed by an independent slow 180° Walsh sequence based phase switch. Their combined effect causes the LO phase to switch between 90°, 180°, 270° and 360° based on two superimposing Walsh sequences. In the SMA correlator these phase switches are corrected for and sidebands are separated post correlation. However for the purpose of phased array operation, 90° phase switching correction in time domain prior to correlation would require digital real time implementation of 500 MHz Hilbert Transformers. These are complex to design and computationally expensive. If 90° phase switching is turned off, both sidebands would appear overlapped in the signal. For VLBI purposes the unwanted sideband will act like noise and eventually be eliminated in the VLBI correlation processing. This causes a significant Signal-to-Noise Ratio (SNR) disadvantage. In VLBI experiments many single dish telescopes are expected to participate and single dish submillimeter telescopes can have no mechanism to separate the sidebands. This is because a second antenna is needed to provide a phase reference to separate the sidebands if a mixer is the first receiver stage. Thus in submillimeter VLBI the SNR disadvantage of overlapped sidebands is more or less unavoidable. In this project we plan to run the SMA with 90° phase switching turned off for phased array VLBI operation. The 180° phase switching can be left running and compensated for easily in the phased array processor by switching between addition and subtraction on respective Walsh ticks.

2.3.3 Fringe Rate Correction

Over time, when observing a point source the output of a interferometer should remain constant. However as the earth rotates, the source appears to move across the sky. This causes the phase difference between the signal received by two antennas to change slowly. This change can be analyzed as the relative doppler shift caused at the two antennas by earths rotation. This effect needs to be compensated for by introducing appropriate phase shifts per antenna before correlation or beamforming. This is called fringe rotation [3] and can be achieved either digitally by programming correct phase shifts or in analog by shifting the local oscillator

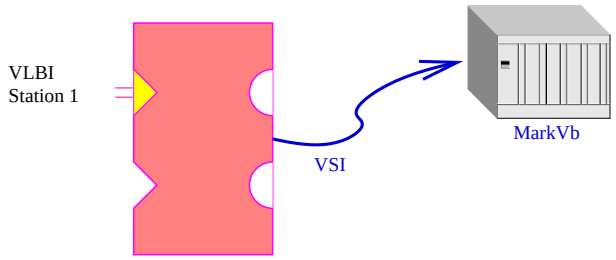

Figure 15: Mark Vb DBE Symbolic System Diagram

(LO) phase with the tracking rate given by equation 2.

$$\frac{d\Delta\phi}{dt} = \frac{2\pi d\cos(\theta)}{\lambda_{sky}}\frac{d\theta}{dt} \tag{2}$$

where $\Delta\phi$ is the relative phase difference to be introduced in a pair of antennas, $\theta = 90 - e$ where e is the source elevation. The fringe rate can be modified to also correct for the fact that delay compensation is done at the IF rate instead of the RF. (Using a tracking rate adjustment for λ_{if}). [3] In the SMA these effect are corrected for in analog by the correlator software. It is therefore not needed to include special provision in the phased array processor to correct for fringe rotation.

2.4 Mark Vb Digital Backend (DBE)

The DBE is designed by the CASPER group using iBOB boards for MIT/Haystack. The DBE is designed to accept a single channel of 512 MHz bandwidth at the iADC input. It channels this signal into 32 frequency bins using poly phase filter banks and fast fourier transforms, it then takes care of compensating for filter bandshapes performs bin weighting and spools the data over the Versatile Scientific Interface (VSI) bus to a Mark V data recorder. Figures 15 and 16 show the DBE design system diagram and blocks diagram respectively. We modified this DBE design (Simulink based) to receive data over XAUI links and sum together the partial phased sums from 2 iBOBS as shown earlier in Figures 39 and 13. This results in a 8 channel phased sum passing through the signal processing chain required for Mark Vb data storage.

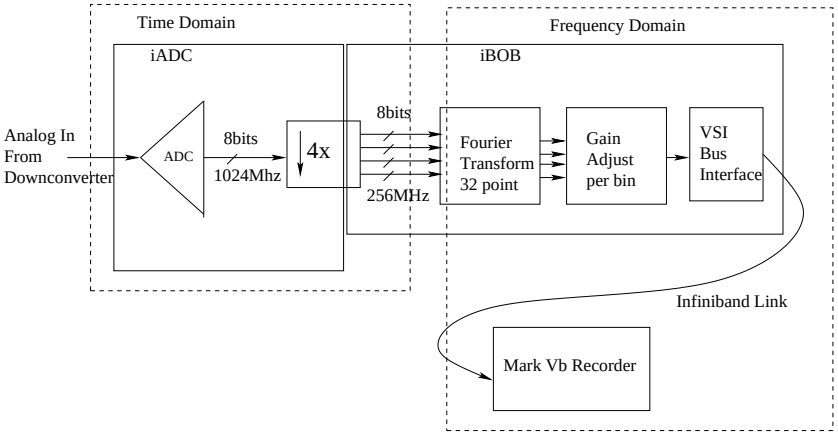

Figure 16: Mark Vb DBE Block Diagram

2.5 Digital Delay Lines

The digital delay lines are implemented across 2 iBOB boards. They read sampled data from 8 antennas using 2 iADC boards per iBOB, i.e. they receive data at the rate of 1024 MHz $\times$ 8 bits= 8 Gbps $\times$ 4 channels $\times$2 boards = 64 Gbps. Each iBOB implements 4 digital delay lines each of which is capable of introducing accurate programmable time delays. The maximum supported delay is 4000 ns and the precision step is 0.1 ns, which is 1/10 times the sampling period T_s.

$$T_s = \frac{1}{1024MHz} = 0.976 \times 10^{-9} \approx 1ns \tag{3}$$

$$\tau_{min} = \frac{1}{10} \times \frac{1}{2B} = \frac{1}{10} \times \frac{1}{2 \times 512 \times 10^6} = 0.0976 \times 10^{-9} \approx 0.1ns \tag{4}$$

Figure 17 shows a detailed design of one of the iBOBs. Since the FPGA cannot be clocked at rates as high as 1024 MHz the iADC and iBOBs perform a demux-by-4 which presents 4 samples per clock at a rate of 256 Mhz. The design expects to be provided with delay values from the control computer. If delays are correctly programmed the 4 channels are phase aligned. This result is transmitted over XAUI links to the DBE at 8 Gbps, i.e. 1024Msamples/sec and 8bits/sample. All components of this design have been developed under the purview of this masters

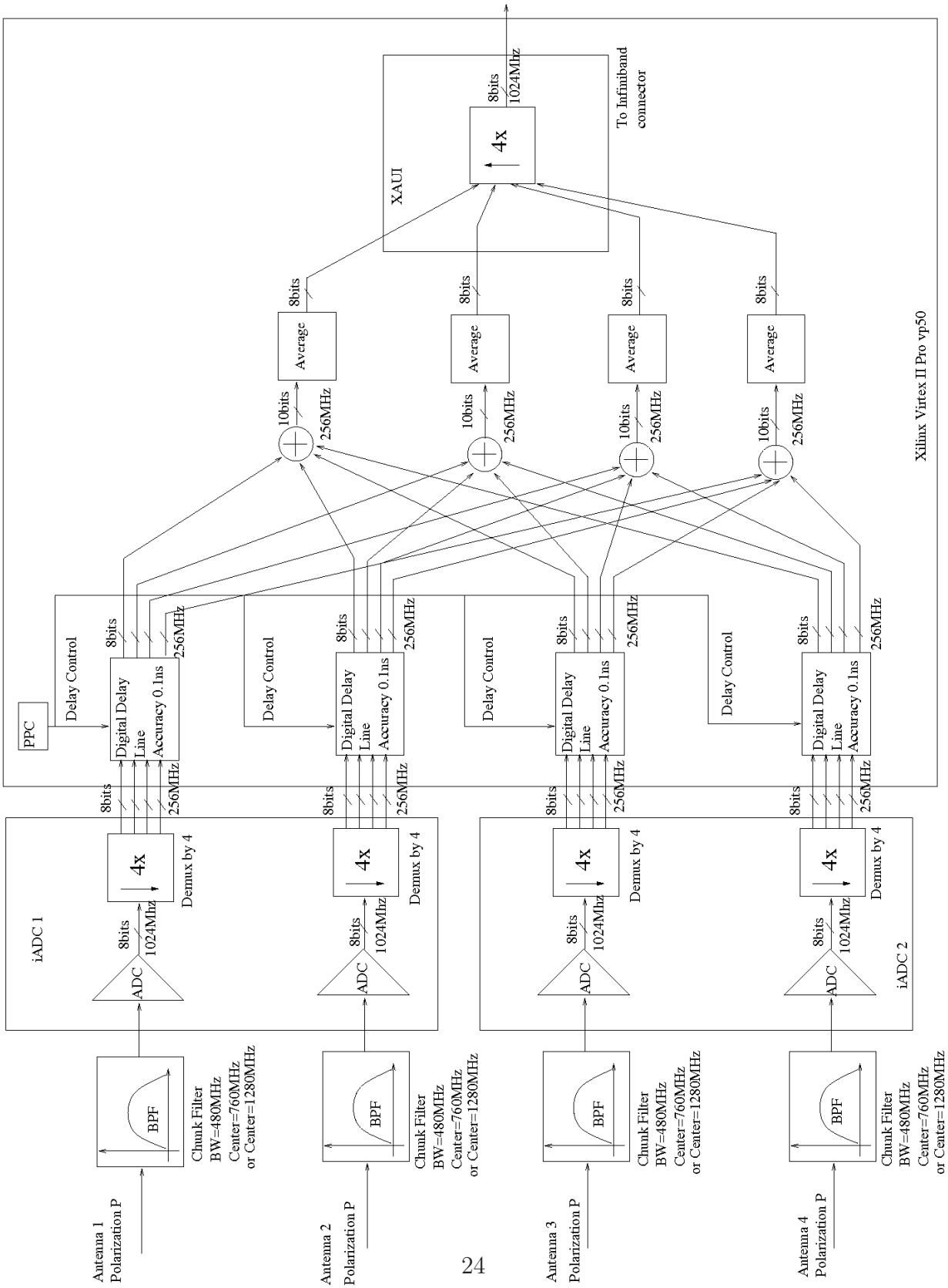

Figure 17: Architecture of 1 Delay Line iBOB Design

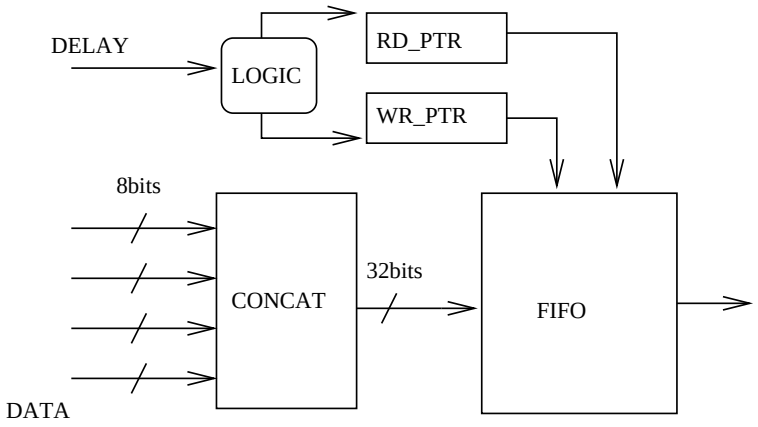

Figure 18: Coarse Delay

project. The delay precision of 0.1 ns is achieved in 3 steps.

2.5.1 Coarse Delay

The coarse delay achieves a delay step size of 4 ns and a maximum delay of 4000 ns. The demux-by-4 is disregarded in this stage. The 4 8-bit samples appearing per clock are concatenated into one 32 bit number per clock. These are fed into a RAM based FIFO (First In First Out) structure with read and write pointers maintained in flip flops. The FIFO can accommodate 1000 32-bit values i.e. 4000 time samples. We have designed control logic that controls the read and write pointers based upon the programmed coarse delay value C. The control logic is unable to handle coarse delay values smaller than 3, thus that is used as a base value in all channels. When the programed value is reduced by 1 the control logic skips a read for one clock by outputting the same word twice. When the programmed value is increased by 1, the control logic causes a jump in the write pointer causing a word to be skipped. Thus the control logic maintains the FIFO occupancy at the programmed delay value. (C). Figure 18 shows the block diagram.

2.5.2 Fine Delay

The fine delay achieves a delay step size of 1 ns. This stage simply performs a re-alignment of the quadruplet of samples arriving at one clock. This is done using

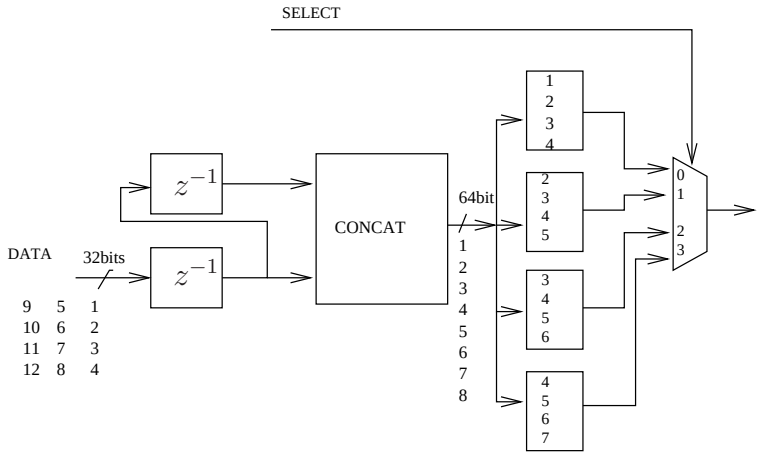

Figure 19: Fine Delay

a barrel selector arrangement shown in Figure 19. It can be seen how we can select to realign the samples to sample number $2-3-4-5$ from the input of sample sequence $1-2-3-4$ by setting the select line to 1. This shows that the select line value represents a phase advance of one sample rather than a delay. The fine delay block can adjust delays between 0 and 3 samples by setting the select line S to $3-d$ where d is the desired fine delay.

2.5.3 Super Fine Delay

In the fine delay stage we could achieve the delay step equal to the sampling interval i.e. ≈ 1 ns. To achieve a delay step finer than that would require to split the sampling interval and hence interpolate between samples. A delay line can be simply analyzed like a filtering operation. If the total desired delay is D the digital filter output $y(n)$ can be written as

$$y(n) = x(n - D) \tag{5}$$

where $x(n)$ is the input sample stream. [6] In the z transform domain the transfer function of this filter can be written as

$$H(z) = z^{-D} \tag{6}$$

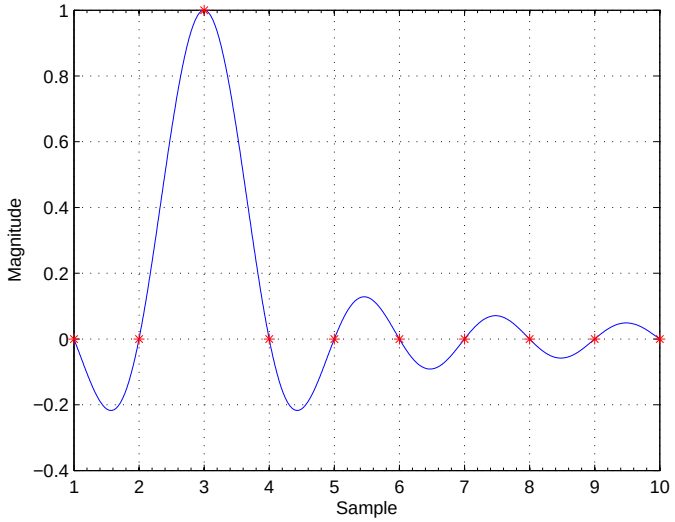

Figure 20: Digital 10 tap filter for $D = 3$

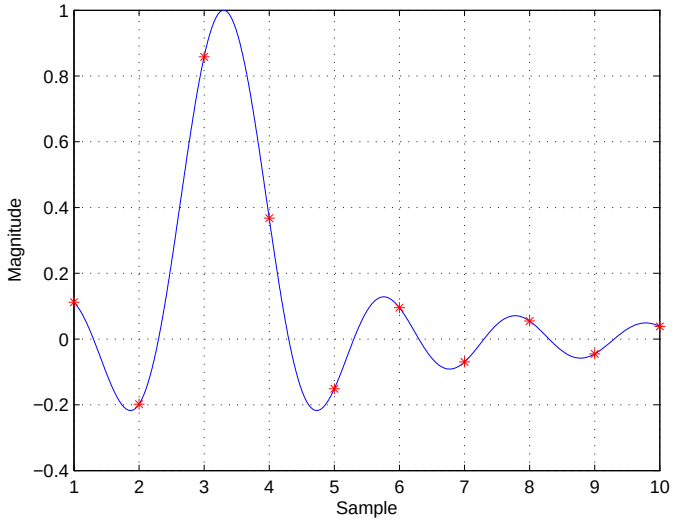

Figure 21: Digital 10 tap filter for $D = 3.3$

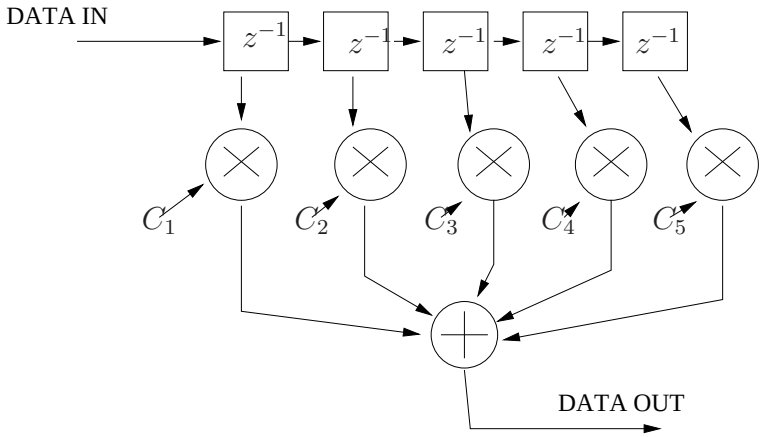

Figure 22: Simple Finite Impulse Response Digital Filter Implementation

and the impulse response of this filter $h_D(n)$ can be shown to be,

$$h_D(n) = \frac{\sin \pi(n - D)}{\pi(n - D)} \tag{7}$$

If D is an integer number of sample periods, the impulse response of a 10 tap filter simply corresponds to 3 delay flip flops as can be seen in Figure 20 where $D = 3$ samples. If D is not an integer number of samples, say $D = 3.3$ samples then the impulse response will be a shifted sinc pulse that is resampled as shown in Figure 21. It is clear from these diagrams that a fractional delay filter cannot have symmetrical coefficients unless $D = 0.5$. To implement a variable fractional delay line we have pre-computed coefficients for 10 such filters corresponding to delays of $D = 0.1$ to $D = 0.9$ in steps of 0.1 sample period and stored these coefficients in a RAM. We have implemented a 10 tap digital FIR (Finite impulse response) filter in real time hardware and we can load any of these coefficient sets on demand using control logic.

Fig. 22 shows the digital implementation of a simple 5 tap using a tapped delay line FIR filter where $C_1 ... C_5$ are the filter coefficients. However our system is demux-by-4 and we get 4 consecutive samples per clock. This requires a complex demux-by-4 pipelined FIR filter which we have designed. Such a filter is implemented by performing 4 multiplications and 4 partial sums per stage. Each stage corresponds

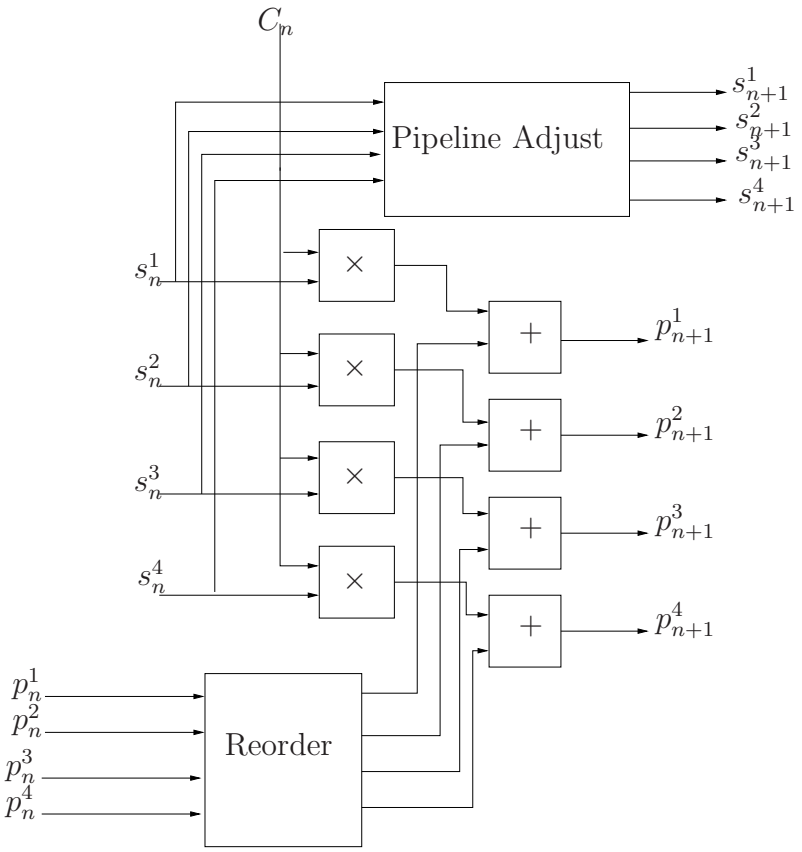

Figure 23: Demux-by-4 FIR filter tap implementation

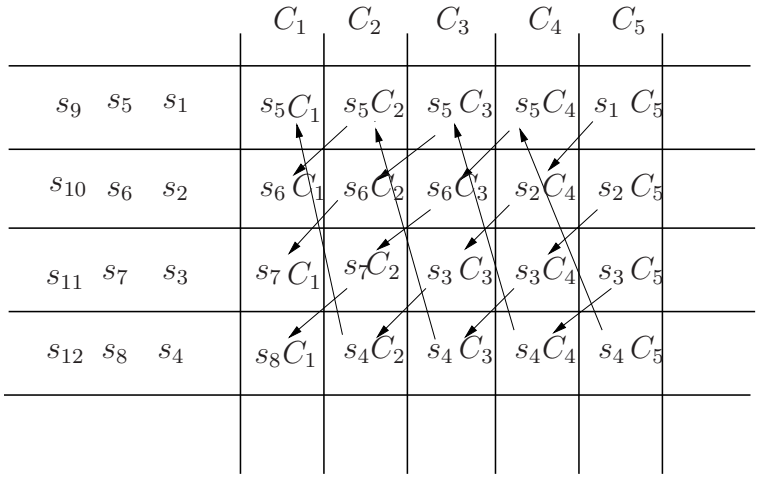

Figure 24: Demux-by-4 Finite Impulse Response Digital Filter Implementation

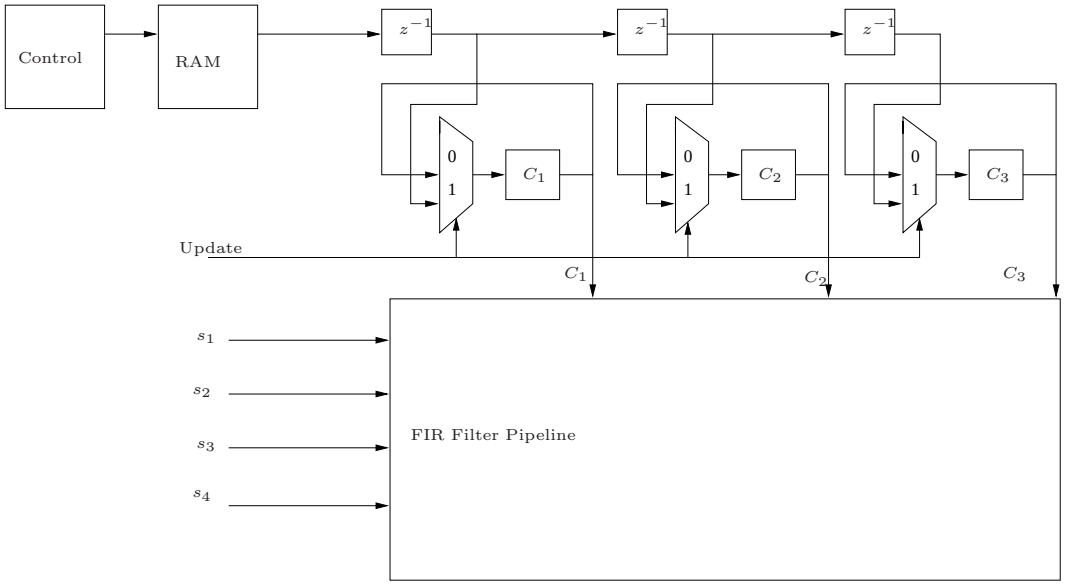

Figure 25: Double Buffering Scheme shown for a 3 tap filter.

to one filter tap. Partial sums are computed such that at any stage the output partial sums provide the correct FIR filtered output. Fig. 23 shows one tap of the filter chain. $p_n^1 ... p_n^4$ are the partial sums and $s_n^1 ... s_n^4$ the data samples passed on from the previous tap. The partial sums are reordered in everystage and the samples are delayed for adjusting the pipeline before the multiplication and addition operations are performed. Fig 24 shows the 5 tap case and how the first filtered output o_1 is computed in the pipeline.

$$o_1 = s_1 \times C_5 + s_2 \times C_4 + s_3 \times C_3 + s_4 \times C_2 + s_5 \times C_1 \tag{8}$$

The filter coefficients can be changed and loaded from RAM to correspond to the desired fractional delay. A double buffering scheme is used such that the change of fractional delay can be put into effect instantaneously. The new set of coefficients is read out of RAM into a shift register which is used to parallel load the actual buffer which supplies the coefficients to the FIR pipeline. Fig. 25 shows this arrangement for a 3 tap case. When the delay needs to be changed the control logic reads out the corresponding set of new coefficients and when they have been read out into a serial load shift register an update pulse is sent out which loads

the new set into the FIR pipeline.

2.6 Delay Line Results

The digital delay line can be tested by programming arbitrary delays into data streams and monitoring the output. Using the Power PC microprocessor we can record snapshots of data into onchip block RAMs. For this purpose we have designed a special circuit called *iBOBscope*. The design simply uses a block RAM that is mapped to the Power PC bus with control logic to drive it address bus and write enable. We can store ≈ 8000 time samples i.e. 8 ns of data per channel using this scheme. The stored samples can be read into an external PC using RS-232 serial interface and then analyzed using MATLAB or similar tools. In our test setup we fed all the ADC inputs with the same band limited noise. The band limiting filter used was 100 MHz wide thus highly oversampling the noise (1024 MHz) to see coherence in the various channels visually. Fig. 26 shows the snapshot of 2 data channels and their sum (actually average) when both ADC inputs were fed with the same signal through equal lengths of cable. It can be seen that the channels are phase aligned and their average is having the same total power as each of the channels. When the snapshots of these two channels are correlated with each other using MATLAB function `xcorr` the result shows a peak at 0 indicating perfect phase alignment shown in Fig. 27. We can now program our digital delay lines and then observe post-delay-line snapshots with the same test setup to check whether the delay lines are working correctly.

Fig. 28 shows the 2 channels and their average when one of the delay lines are programmed to a value of 55.5 ns. We can observe that the channels are not phase aligned and the average signal power is reduced. We can repeat the MATLAB `xcorr` function on this set and see the correlation peak. As can be seen in Fig. 29 and Fig. 30 the correlation peak is between 55 and 56 ns which assures us that the digital delay lines have correctly interpolated the samples for a delay of 55.5 ns.

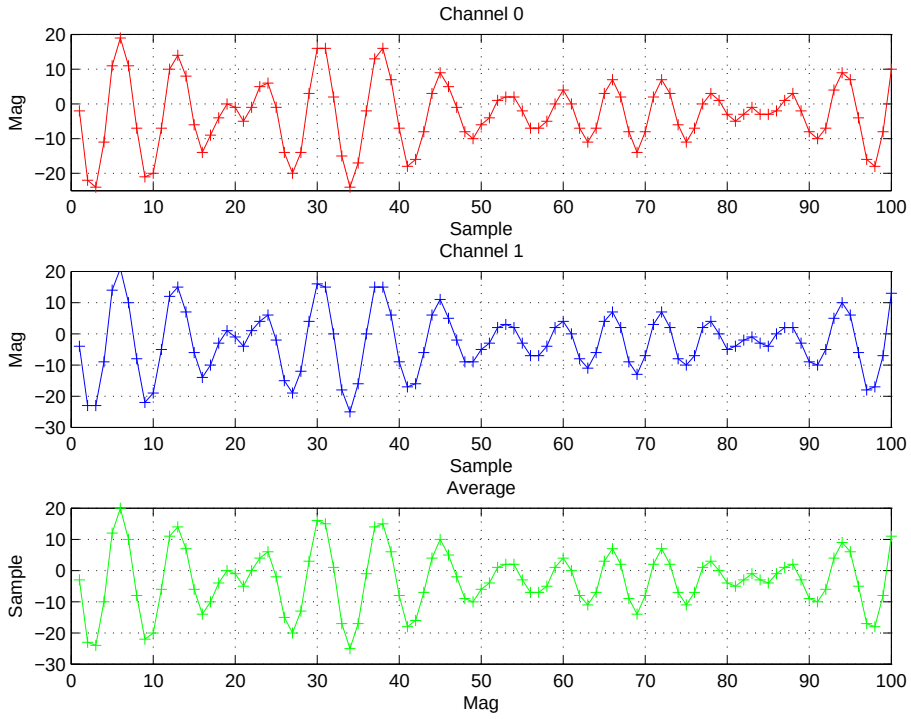

Figure 26: Snapshot of two data channels and their average when they are phase aligned

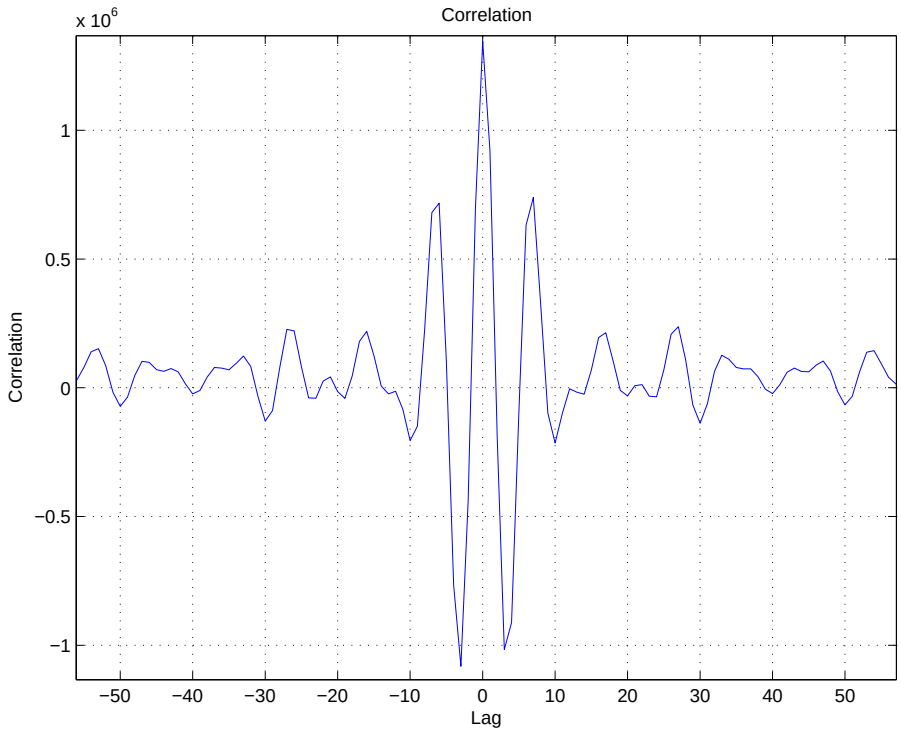

Figure 27: Output of MATLAB function xcorr with 8000 samples of 2 aligned channels

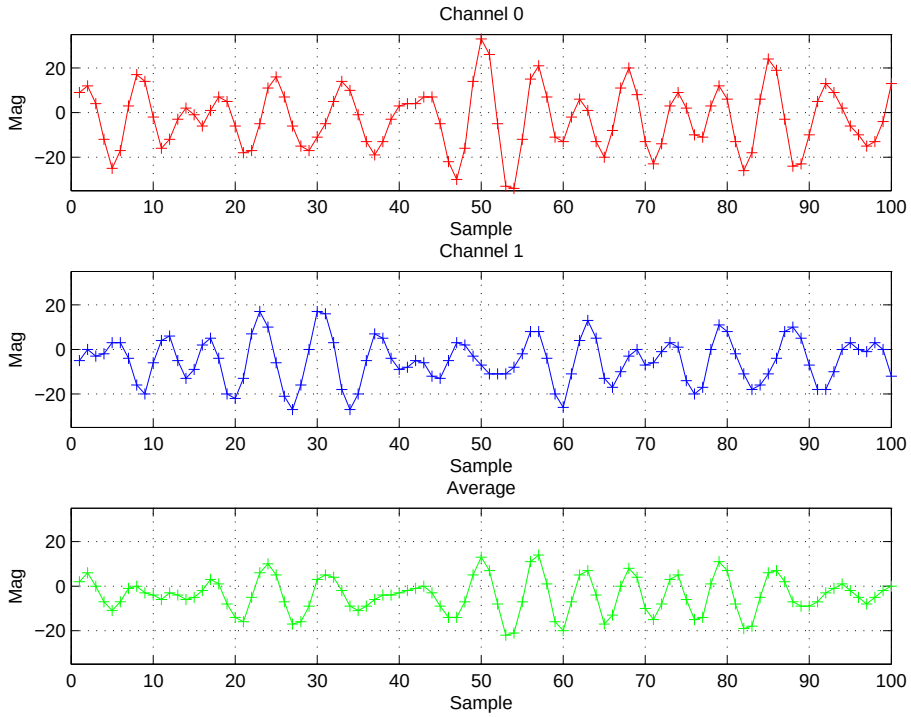

Figure 28: Snapshot of two data channels and their average when one is delayed digitally by 55.5 ns

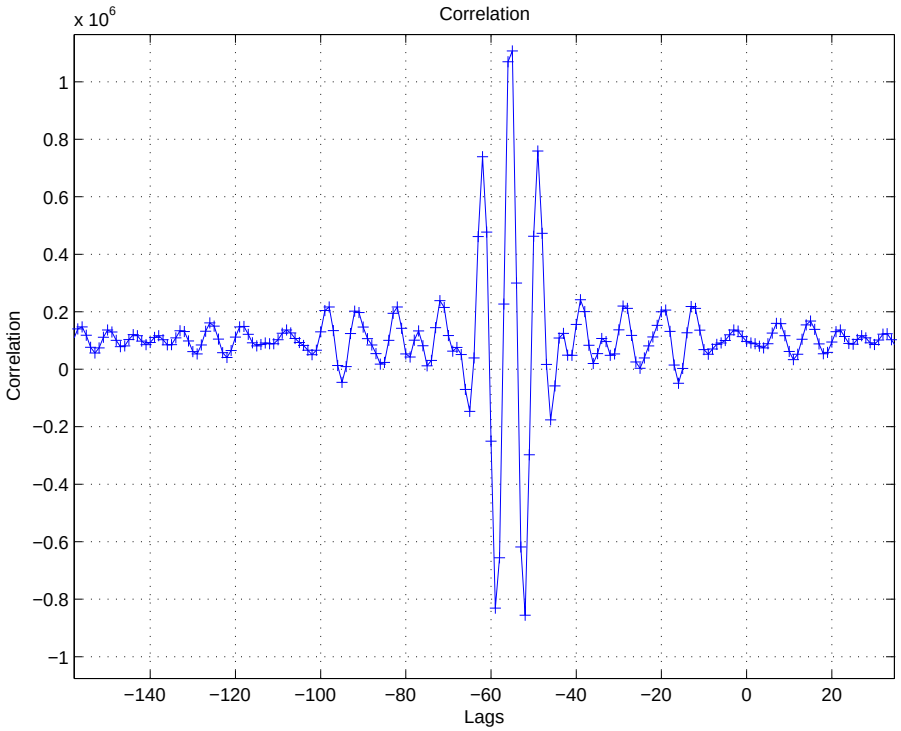

Figure 29: Output of MATLAB function xcorr with 8000 samples of 2 channels when one is delayed by 55.5 ns

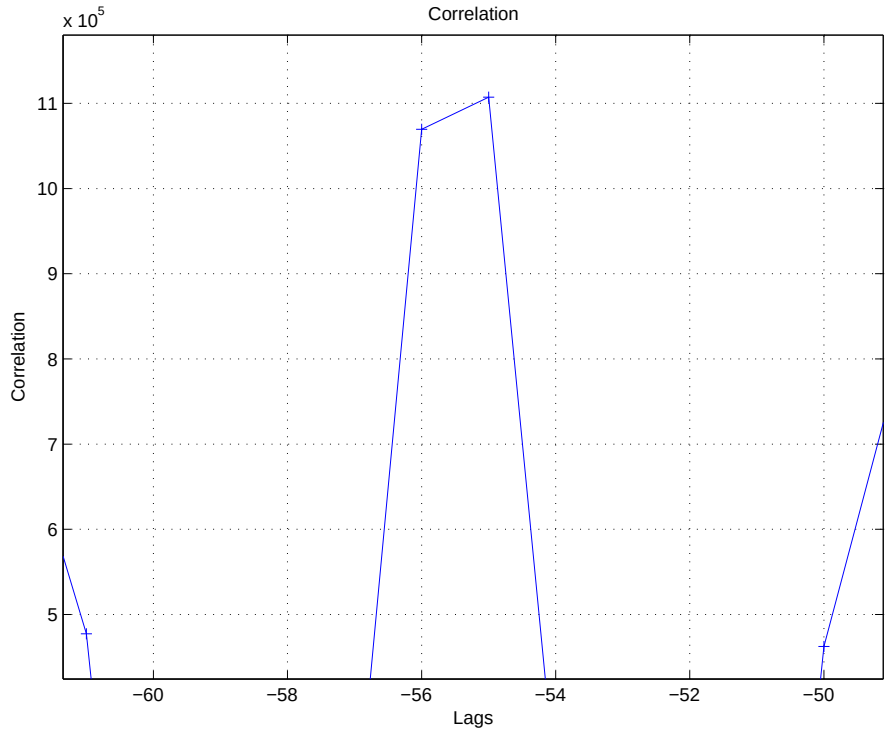

Figure 30: Zoom into peak of Fig.29

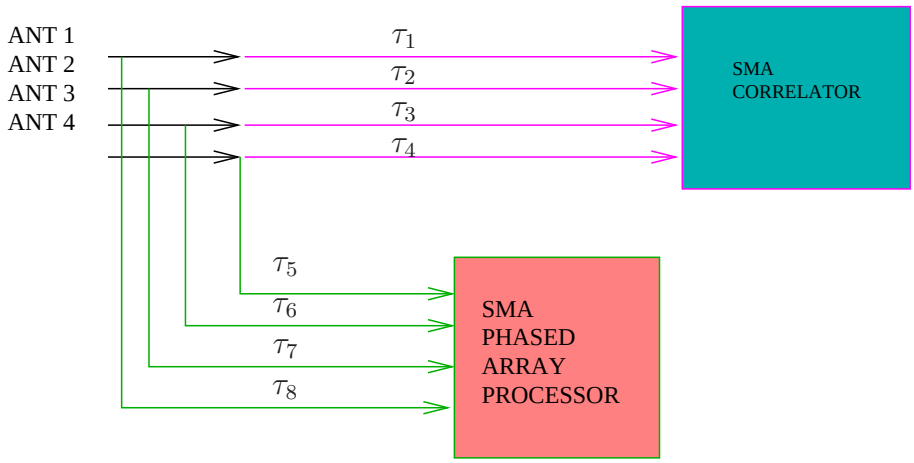

Figure 31: Signal flow to SMA correlator and phased array processor

3 Beamformer Calibration

Calibration implies the problem of finding the correct delay values to program into delay lines such that we get accurate phasing. The system delay can be classified into *geometric, instrumental* and *atmospheric* delays. The geometric delays can be calculated *a priori* and the instrumental effects can be calibrated out by careful measurements along with some form of phase switching (180°) arrangement. The atmospheric delays however change with time and need to be calibrated in real time for achieving good phase coherence. One way to calibrate out the atmospheric effects is to steer towards a strong astronomical source and use that as reference to find the atmospheric delay per antenna. However the astronomical signal is buried deep under system noise and we would need a correlator to measure the delays between channels. The SMA correlator can be run in parallel with the beamformer to extract this information. The signal path to the SMA correlator follows a different analog path (and analog processing chain) than the SMA beamformer as seen in fig. 31. The differences in analog paths in various antennas τ_1, τ_2.. etc. must be calibrated before the SMA correlator can be used to continuously track the atmosphere. To solve this problem it was decided to reuse a correlator being built by CASPER on the iBOB board. It was initially decided to just use this correlator for calibrating the SMA but it eventually turned out that we had to redesign some

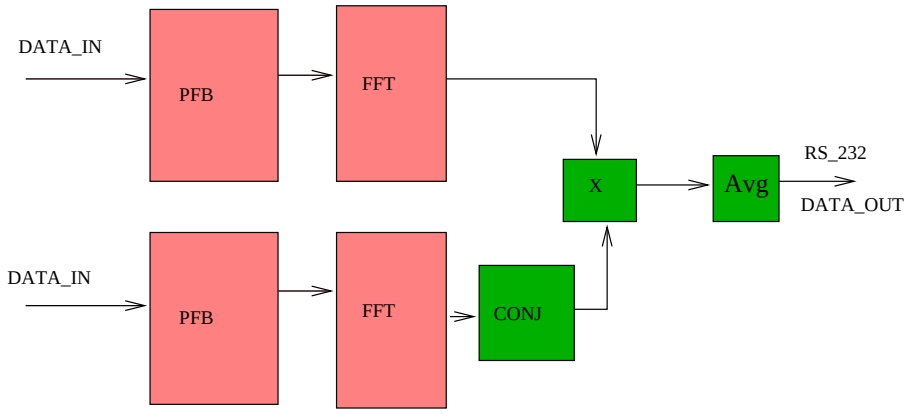

Figure 32: Architecture of *Calibration Correlator*

parts of the CASPER correlator to suit the requirements of our project. It was then decided that we can use this custom designed *Calibration Correlator* (using CASPER library components) for calibrating the SMA beamformer independent of the SMA correlator.

3.1 Design of Calibration Correlator

To accurately phase up 8 antennas we need to know 7 delays, i.e. delay of each antenna w.r.t. a reference antenna. If we can measure the cross-correlations of each of 7 antennas w.r.t the reference we can derive these delays from the phase slope in correlation functions. If we observe a fairly strong astronomical source and measure these 7 correlation functions for a few seconds of integration we would be able to track these calibration delays. Since atmospheric and instrumental effects (even geometric effects) do not change significantly over the order of a few minutes we do not need to measure these 7 correlations simultaneously. We should be able to use an easy-to-design single baseline correlator and time multiplex these 7 measurements on it. The CASPER astronomy library provides us with pre-designed blocks for FIR based implementation of poly-phase filter banks (PFB) and Fast Fourier transform (FFT) circuits. The PFB performs a pre-conditioning of signal before computing FFT to get a sharper frequency resolution. The PFB can be shown to have better performance than windowing and overlap methods

commonly used with FFT. Using these blocks we added circuitry to multiply two FFT streams to get the cross correlation function and integrate the result for a few seconds. Fig. 32 shows a system diagram of the designed correlator. It is an FX correlator with 64 complex/32 real frequency channels and can do onchip integration of about 16 seconds. The correlation spectra can be read out using RS-232 serial interface to a control PC. The design heavily relies on the Polyphase Filter Bank (PFB) and Fast Fourier Transform (PFB-FFT) library developed by CASPER. The PFB-FFT provide a highly optimized implementation of the FFT algorithm with preceding decimating FIR filters to give sharp spectral response. [1] After the PFB-FFT stage we have implemented a complex multiply (with complex conjugate) operation to give correlation result. A Vector Accumulator (Avg) has been designed to integrate the spectra for a maximum of 16 seconds in Block RAMs. The integration result is stored in CPU mapped memory to read out via serial port. Fig. 32 shows the block diagram of the correlator with green blocks representing ones designed by us and pink ones supplied by the CASPER library. The design of this correlator consumed a very large portion of the project time line because of the challenges involved and the fact that the CASPER libraries were undergoing constant development and upgrades during our project. The various design challenges we faced included dynamic range, sensitivity and fitting the logic within the FPGA with the specified timing. (256 MHz clock rate) We have tested the operation of the correlator using a test setup wherein a correlated component (simulated 480 MHz wideband noise) was added to two independent simulated receiver noise components (480 MHz uncorrelated noise sources).

3.2 Calibration Correlator Results

Fig. 33 shows auto correlation spectra measured with the designed correlator. As we can notice for real data we get a two sided spectrum. The phase plot shows a slope corresponding to the difference in cable lengths feeding the 2 ADC inputs. Fig. 34, 35, 36, 37 show the cross correlation measured using progressively weaker correlated component signals. Fig. 38 shows the cross correlation when the cable lengths are equalized thereby giving a flat phase response.

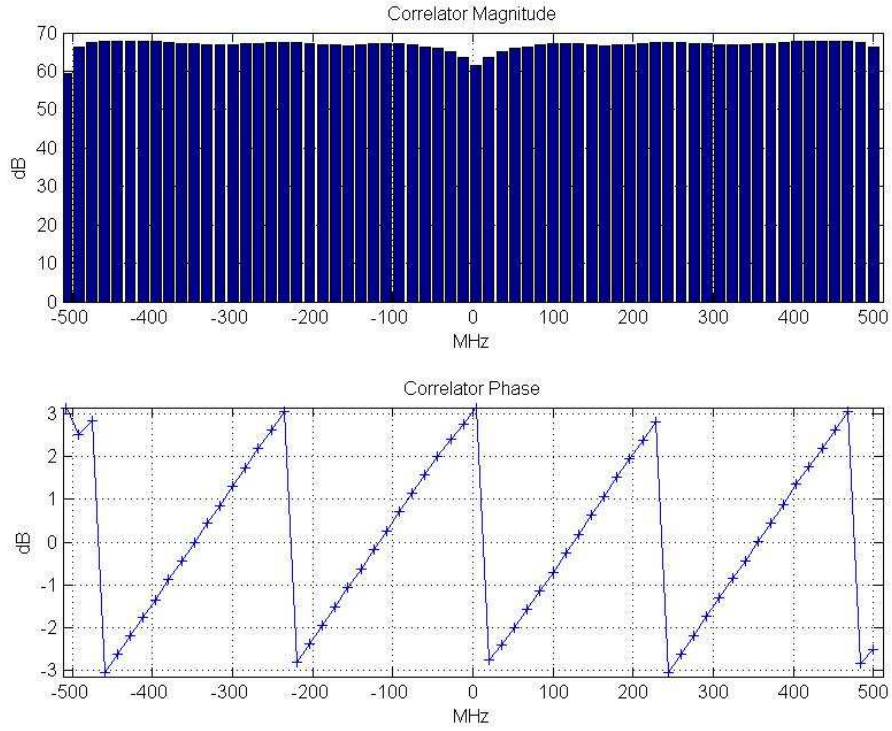

Figure 33: Autocorrelation Spectra (∞ Signal to Noise)

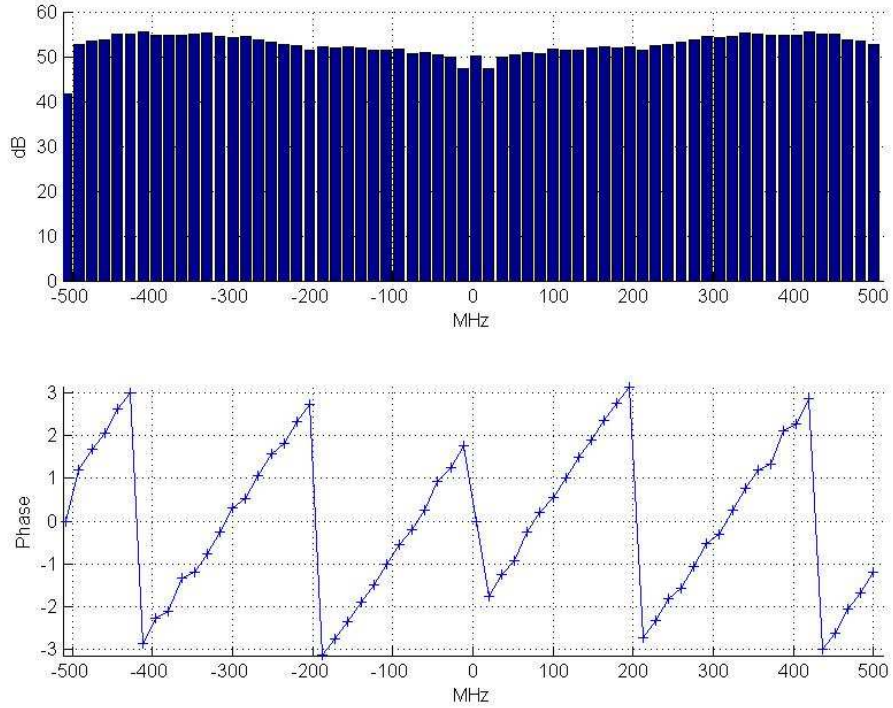

Figure 34: Cross correlation Spectra with −2 dB Signal to Noise

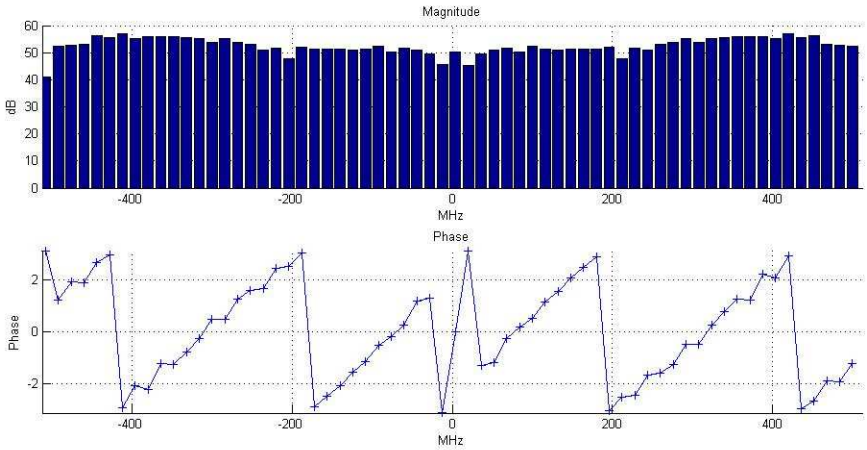

Figure 35: Cross correlation Spectra with −9 dB Signal to Noise

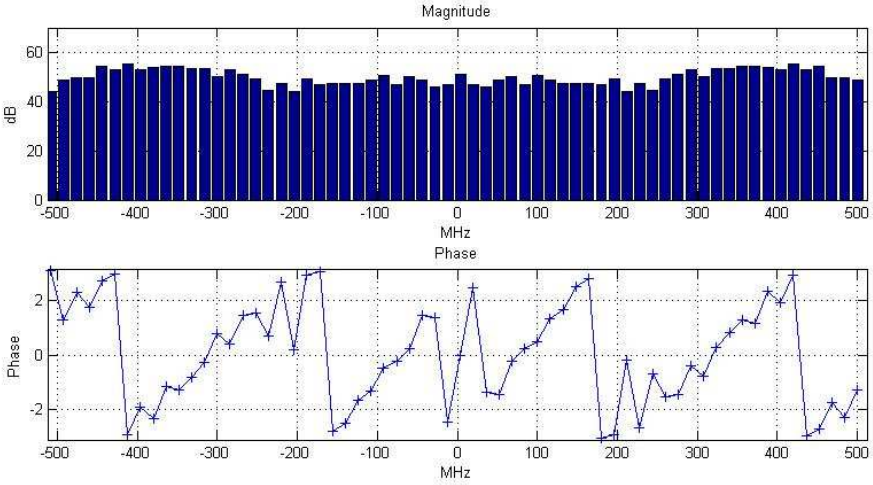

Figure 36: Cross correlation Spectra with −9 dB Signal to Noise

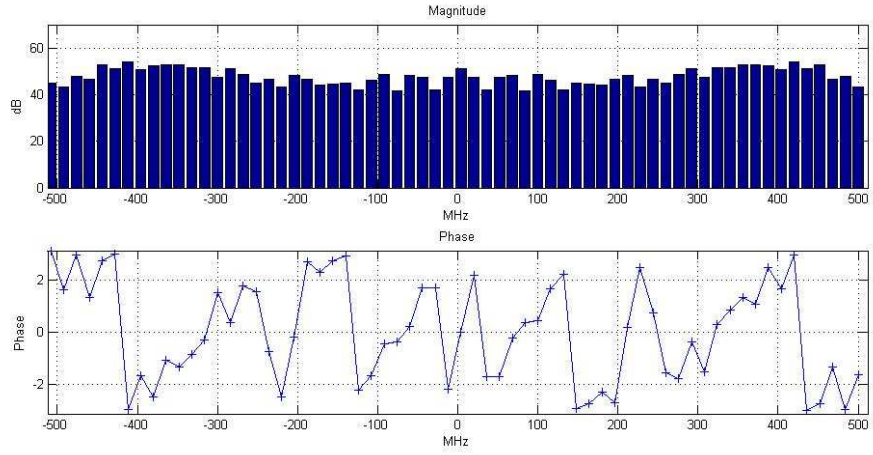

Figure 37: Cross correlation Spectra with −9 dB Signal to Noise

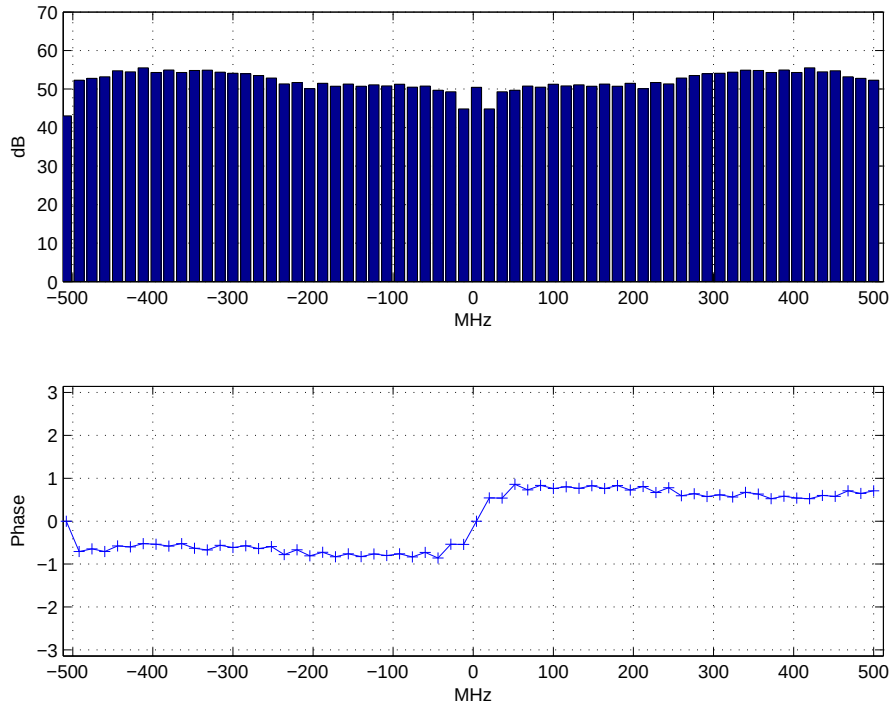

Figure 38: Cross correlation Spectra with no cable delay

3.3 Final Architecture

Finally the correlator with time multiplexed correlation calibrations can be seen in Fig. 39 and Fig. 40. The spare Infiniband connector on *Beamform* boards described in (1) is used to spool the sampled and delayed data channels (before sum) through the spare Infiniband in a time multiplexed fashion. The *Correlator* receives 2 channels from the *Beamform* boards over XAUI and calculates correlation spectra which are read out over RS-232 to the control computer which can extract delays from it and appropriately program the delay lines in *Beamform* boards.

4 Discussion and Future Work

The system built during this masters project has cleared away most of the difficult problems in the path of a working beam former on the Mauna Kea summit. However some components need some more work before the system can be ready for a sky test.

1. **Correlator Sensitivity** As seen in the previous sections the calibrating correlator cannot extract delays with reasonable confidence if the correlated signal goes more than 15 dB below noise. With an astronomical signal the expected SNR is about -30 to -40 dB. Some work is needed to improve the sensitivity of the correlator.

2. **XAUI Links** Though we have tested XAUI links and their synchronization, we have not integrated these links with the above designs. It might be needed to add additional circuitry to designs to take care of infrequent link failures in the XAUI.

3. **Mark *V* DBE** Integration blocks for the Mark V DBE have been designed but have not yet been tested with actual Mark Vb recorders because of logistic problems with acquiring them.

4. **Delay Extraction and Automation** Software modules to extract delays automatically from the correlator result and program delays lines need to be built.

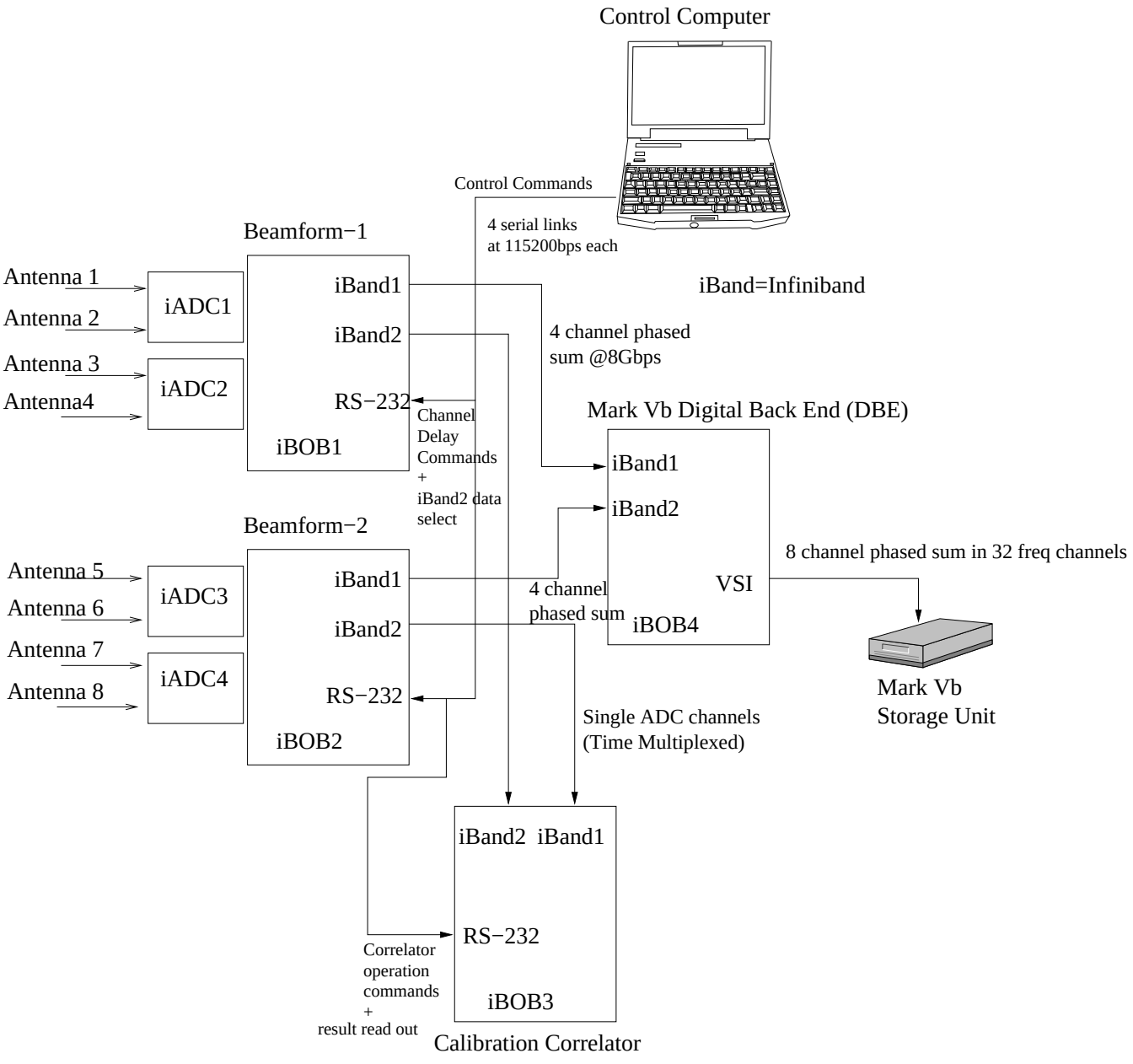

Figure 39: Overall System Architecture

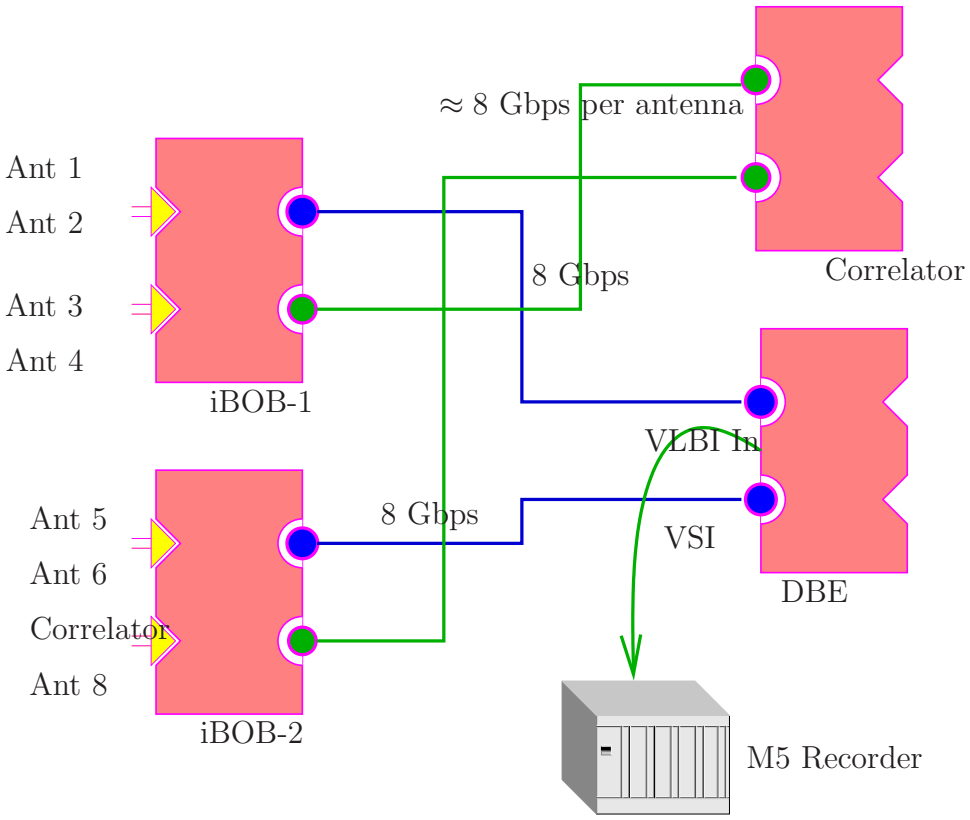

Figure 40: Final Architecture

In conclusion we have demonstrated the power of FPGA based back end design in this masters project. With a very short development cycle FPGA's with supporting infrastructure can help us build telescope backends. We have also come a long way with the design of a phased array processor for the telescopes on the Mauna Kea summit.

References

[1] Chen Chang et. al Aaron Parsons, Don Backer. A new approach to radio astronomy signal processing: Packet switched, fpga-based, upgradeable, modular hardware and reusable, platform-independent signal processing libraries. URSI, 2005.

[2] Fulvia Melia Heino Falcke and Eric Agol. Viewing the shadow of the black hole at the galactic center. *The Astrophysical Journal*, 528(2):L13–L16, 2000.

[3] Oliver Peter Lay. Astronomical interferometry at submillimter wavelengths, September 1994.

[4] Colin Masson. Fringe rotation, delay stepping and phase switching. Technical Report 56, Submillimeter Array Technical Memorandum, 1991.

[5] Paul T. P. Ho James M. Moran and Kwok Yung Lo. The submillimeter array. *The Astrophysical Journal*, 616(2):L1–L6, 2004.

[6] V. Välimäki and T.I. Laakso. Principles of fractional delay filters. volume IEEE International Conference on Acoustics, Speech and Signal Processing, June 2000.