

EVN Correlator Design

JUC Memo #1, Version 3.3

June 27th 2014

**Jonathan Hargreaves (JIVE)
Harro Verkouter (JIVE)**

Table of Contents

Table of Contents	2
List of Abbreviations	3
1 Control Software	4
MAC and IP address assignment	5
Resets	6
FPGA Configuration	6
Time Synchronization	6
2 Algorithmic Overview	7
Specifications	7
Data Input	7
Overview of Signal Flow	9
Packet Reception	10
Synchronization of Data and Delay Model	11
Time Multiplexed Station Processing	11
Delay and Phase Correction - Introduction	11
Delay and Phase Models - Implementation	12
Polyphase Filterbank	17
Corner Turner	20
Correlation Engine	20
Validity	22
Bit Truncation	23
Correlator Product Output	24
Appendix A FPGA Resources	26
Front Node	26
Back Node	26
References	27

List of Abbreviations

1GbE	One gigabit per second Ethernet
10GbE	Ten gigabit per second Ethernet
BN	Back Node. One of the four FPGAs on the right half of the UniBoard
DDR	Double Data Rate. Data transferred on both edges of the clock
DDR3	A memory module conforming to JEDEC standard JESD79-3B
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
EEPROM	Electrically Erasable Programmable Read Only Memory
EPCS	Altera programmable configuration device
EVN	European VLBI Network
FFT	Fast Fourier Transform. An efficient implementation of a DFT
FN	Front Node. One of four FPGAs on the left side of the UniBoard
FPGA	Field Programmable Gate Array
GMAC/s	One billion multiply-accumulate operations per second
INTA,INTB	General purpose signals connected to all FPGAs on the UniBoard
JUC	JIVE UniBoard Correlator
MAC	(1) In Networking: Media Access Controller (2) In DSP: Multiply-Accumulate
MTU	Maximum Transmission Unit. The largest packet allowed on a network
MT/s	Million transfers per second
PFB	Polyphase Filter Bank
PPS	Pulse per second
PSN	Packet Sequence Number
SOPC	System on a Programmable Chip
SRAM	Static Random Access Memory
UDP	User Datagram Protocol
VDIF	VLBI Data Interchange Format
VLBI	Very Long Baseline Interferometry
WDI	Watchdog Interrupt

1 Control Software

This document describes the control system from the point of view of the UniBoard itself. A high level overview of the whole JIVE UniBoard Correlator system is given in Verkouter [1].

Each of the eight Altera GX230 FPGAs on the UniBoard includes, as part of its power-up configuration, a 1 Gigabit Ethernet port and embedded Nios II processor to send and receive control and status information. The embedded processor together with its peripherals is referred to as an SOPC system, and can be assembled using Altera's SOPC builder tool.

The FPGA Ethernet ports connect to the outside world via a Vitesse Semiconductors VSC7389 single chip switch that provides a non-blocking wire speed connection to four RJ-45 connectors. The scheme is shown in Figure 1.1

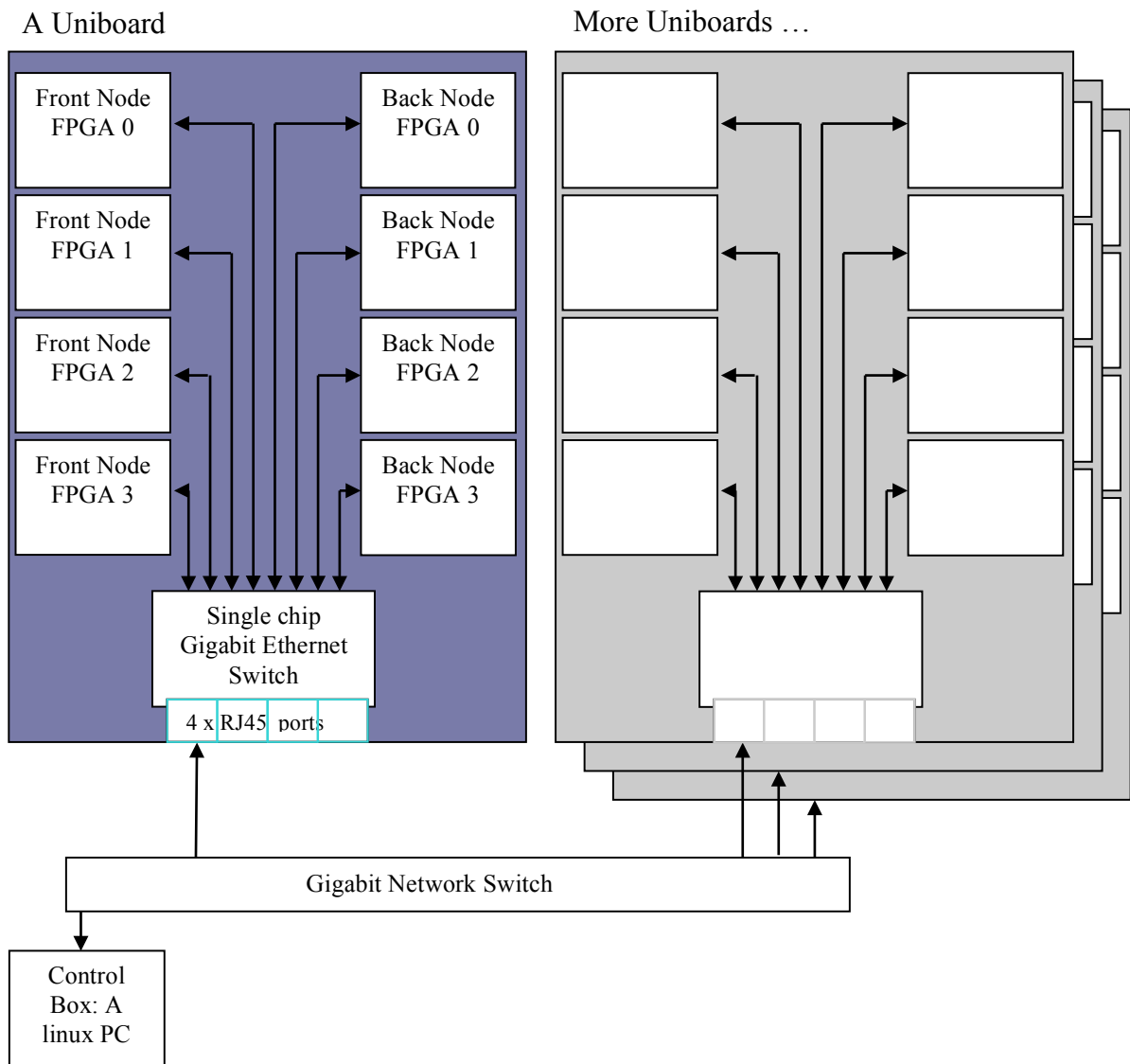


Figure 1.1: Gigabit Ethernet control network for UniBoard

Figure 1.1 shows a single Gigabit connection to several UniBoards via a switch to an external

computer, an arrangement that provides sufficient bandwidth for control, including updating delay models, of at least 64 UniBoards.

A set of registers within each FPGA provides a control/monitor interface to the firmware. Some registers are provided as part of ready-made blocks of IP, such as ten Gigabit MACs and DDR memory interfaces, while others are created using parallel IO ports (PIOs) to link the Nios software to the fabric of the FPGA. Within the SOPC system these registers have symbolic addresses – software running on the Nios processor translates these to present a uniform memory mapped register set to the outside world.

The control computer communicates with the FPGAs by sending non-jumbo (1400 byte MTU) UDP control packets containing a combination of the read/write instruction fields detailed below. Packets from the control computer may include requests to read, write or read-and-write-back data to or from specific registers, or blocks of registers. A read-and-write-back performs a masked AND, OR or XOR operation to reset, set or toggle individual bits within a register. In addition, configuration data can be streamed to an FPGA for storage in the flash memory.

Each control packet includes a unique packet sequence number (PSN). Once an FPGA receives a control packet it starts executing the commands embedded therein, accumulating the return values in a reply having the PSN of the currently executing command packet. It is vital to recognize that *every* command yields a return value. Since almost all commands take a 32-bit address as operand they will return at least that address if the action was successful and the bitwise NOT of the address if a fault occurred (eg writing to a read-only location). This may be followed with the results of the action, which is highly command dependent. The accumulated reply packet is not sent until all commands have been processed. Further details of the command protocol are given in Verkouter [2].

The FPGA may not transmit to the control computer without first receiving a control packet from it. Correlation data output should be directed to a back end processor at a different IP address from the control computer.

After a power-up or reset the control computer reads a version code from each FPGA to determine the configuration of each chip. The FN chip contains a different set of registers to the BN, and there may be alternative versions of each to accommodate different operating modes.

MAC and IP address assignment

At power-up each 1GbE port must be assigned a unique MAC address before it can be used. By default a MAC address is assembled by the Nios processor using a base address hard coded into UNB_OS, a 5-bit board address and 3-bit chip address hardwired to each FPGA. The board address can be set using a toggle switch on the backplane so that 32 UniBoards can be connected in a subnet. Similarly UNB_OS assigns an IP address to its control port in the form a.a.b.n where a.a is the base address (10.99 by default), b is the board address (0-31) and n is the node number (1 to 8).

After the control port is configured, the control system can initialize any 10GbE ports in the design by setting their MAC address, IP address, subnet and gateway. See Hargreaves [3] for details on setting the port parameters

Resets

Soft Resets

A number of soft resets are available to independently reset individual components to their power on state. Both FNs and BNs have application resets to clear address counters, state machines and FIFOs prior to the start of a new scan. For the FNs this includes the delay model coefficient FIFOs. The application resets must be applied during board configuration before each scan is processed. The other soft resets are:

FN

- Soft reset: resets the Vitesse PHY chips during ten gigabit port initialization

BN

- DDR reset: recover from a DDR error.
- Transceiver reset: recover from a synchronization error on the FN-BN transceivers.
- TenGbE reset: recover from incorrect startup of the ten gigabit Ethernet port.

Hard Reset and Watchdog Timer

On a hard reset the FPGA is reconfigured from its configuration EEPROM as it would be after a power cycle. This can be used to load a new configuration or, during debugging, to recover from error. A STM6823 watchdog timer accompanies each FPGA. This generates a hard reset automatically if the WDI output from the FPGA does not change state for approximately 1.6 seconds.

During normal operation the Nios processor toggles the WDI line periodically on receipt of an interrupt from an internal timer. This interrupt is set at a lower priority than signals from the control Ethernet port, so that a hard reset occurs if a bug causes the processor to stop responding. It is also possible for the control computer to force a hard reset by telling the Nios to set a bit in the control register.

FPGA Configuration

Each FPGA loads its configuration from a Numonyx M25P128 128Mbit serial flash memory. The SOPC system interfaces to this device using Altera's EPCS Device Controller Core which allows block and fine-grained read/write access to the flash. An uncompressed configuration image for the Stratix IV GX230 FPGA is approximately 104Mbits. Compressed images can be much smaller depending on how much of the FPGA logic is used. It is possible to hold a safe 'fall back' image along with one or two application images in the flash. The remaining memory will be available as non-volatile general-purpose storage and may be used for example to store default MAC and IP addresses.

Time Synchronization

The UniBoards in the EVN Correlator use internal clocks that are asynchronous to the incoming data and to those of other UniBoards. Correlator time is derived from the timestamps on the incoming data. There is no PPS or reference clock. Details of the control system used to synchronize data and delay models are given in Verkouter [4].

2 Algorithmic Overview

Specifications

Specifications for the EVN correlator are the following¹

	Supported June 2014	Planned
Stations	32	32
Polarizations	2	2
Bandwidth (real time) ²	64MHz (1 UNB)	64n MHz (n UNBs)
Sub-bands (real time)	16MHz, 32MHz under test	1, 2, 4, 8, 16, 32, 64MHz
Input resolution (max)	2 bits	1, 2, 4, 8 bits
Integration time (all products) ³	0.022s – 1s	0.022s – 2s
Correlation points	2112 incl cross, auto, and cross-polarization	2112
Frequency resolution	15.625kHz	<1kHz spectral line mode >125kHz continuum mode
Data Input Format	VDIF	VDIF mixed frame sizes

Data Input

The data path from antenna to correlator for real time data is shown in Figure 2.1. A data sender (such as a UniBoard configured as a digital receiver) at each station divides the sampled continuum signal into sub-bands. The sub-bands are packetized and transmitted across the network to the correlator.

The data senders must allocate destination IP addresses such that all the data for a given chunk of bandwidth arrives at a single correlator UniBoard. Each UniBoard can process sub-bands totaling 64MHz, with 1 to 8 bit resolution. Up to 32 stations can be processed simultaneously. If fewer stations are needed it is possible to trade off stations for bandwidth, for example 16 stations and 128MHz per UniBoard.

For pre-recorded data, smaller bandwidths can be processed at a multiple of real time, for example four 8MHz bands can be processed at double real time. Larger bandwidths than the real time bandwidth cannot be processed however, due to limitations on internal buffer sizes – the solution is to add more UniBoards.

¹ Note on units

Network transfer rates are base 10: 10Gbps = 10^{10} bits per second.

VLBI sample rates are binary in megasamples per second 1GSps = 1024×10^6 samples per second.

Memory sizes are binary: 1GByte = 2^{30} bytes

² This is the maximum bandwidth. For pre-recorded data smaller bandwidths and sub-bandwidths can be processed

³ Shorter integration times possible for fewer products

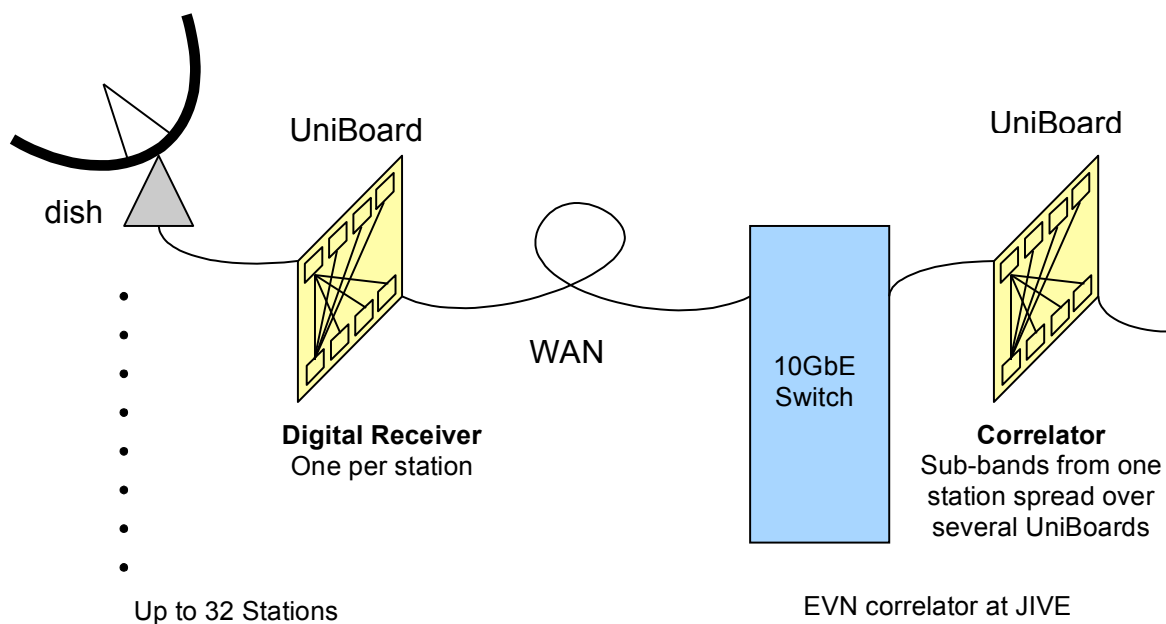


Figure 2.1: UniBoards at the Stations Send Data to the Correlator Over a 10Gb Network

The station data are distributed between the four ‘Front Node’ (FN) FPGAs on a UniBoard as shown in Figure 2.2. The table below shows the data rate into each FN FPGA for different bit resolutions

BW (MHz)	Nyquist	Stations	Pols	Resolution	Data Rate (Gbps)
64	2	8	2	1	2.048
64	2	8	2	2	4.096
64	2	8	2	4	8.192
64	2	8	2	8	16.384

A single 10GbE input port per FN is sufficient for all modes except 8-bit which requires 2 ports.

The FN FPGA performs all station-based processing, including compensating for network and geodetic delays, station clock offset correction, and conversion to the spectral domain using a polyphase filter bank and FFT. After the FFT the data are distributed amongst the ‘Back Node’ BN FPGAs with each BN receiving a quarter of the frequency points. The red, orange, green and blue lines in Figure 2.2 show the paths of each quartile of the frequency points from the FNs to the BNs. Each arrow represents a serial transceiver path with a capacity of 6.25Gbps.

The BN chips corner-turn the data and then perform the correlation on the frequency ordered data. A correlator engine in each BN can each process 16MHz of bandwidth from all 32 stations x 2 polarizations. The correlation products are exported from the BNs via the 10GbE CX4 connections.

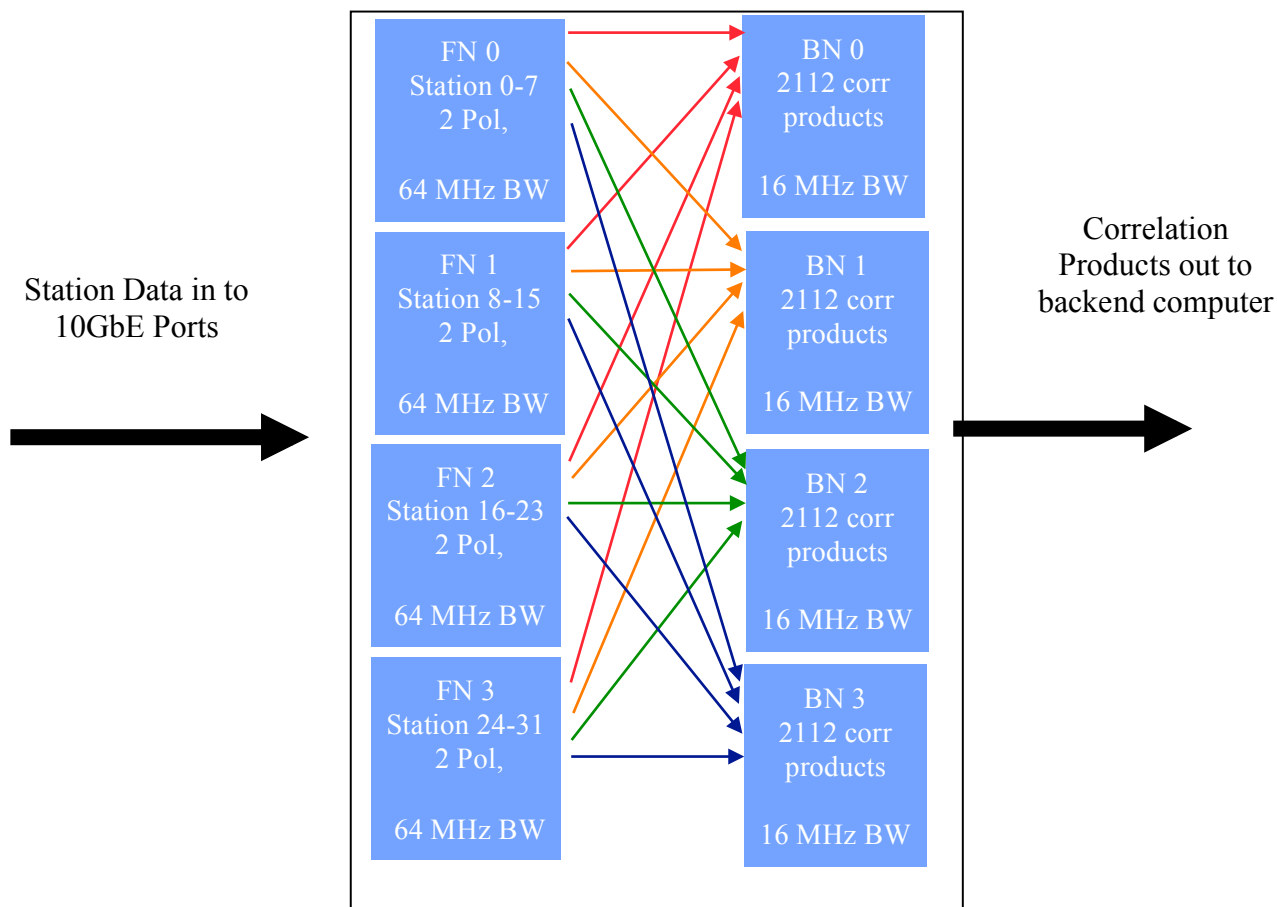


Figure 2.2: Data flow in an EVN Correlator UniBoard

For spectral line studies, it is possible to re-transmit a subset of the frequency points to another UniBoard for further high spectral resolution sub-banding. This can be done using a spare 10GbE port on either the FN or BN FPGA, but is not implemented as of June 2014.

Overview of Signal Flow

Figure 2.3 and 2.4 show the signal flow through the FN and BN respectively. The following sections discuss each block in more detail.

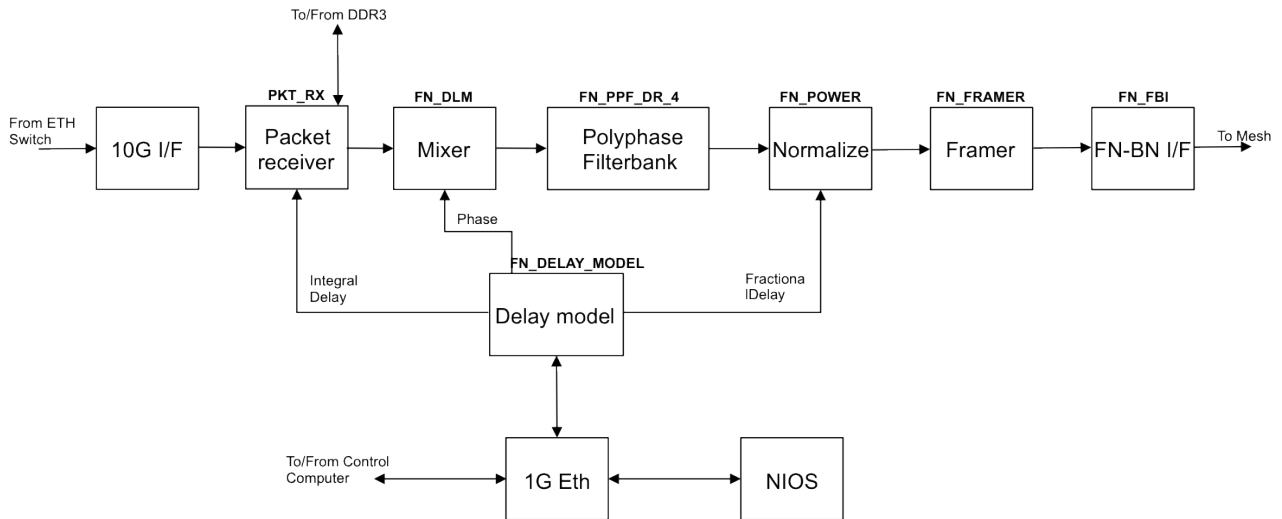


Figure 2.3: Signal flow through the FN

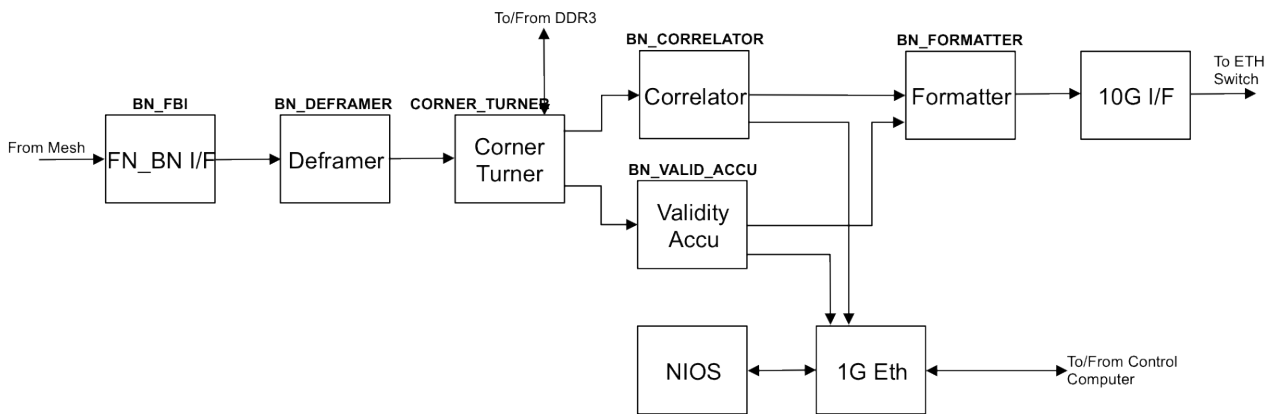


Figure 2.4: Signal flow through the BN

Packet Reception

Data are transmitted from the stations in VDIF [5] formatted jumbo UDP packets. All testing to date has used a fixed, Mark V compatible, frame length of 5000 bytes. However the firmware has been adapted to permit any valid frame length up to 8192 bytes as long as the frame length remains constant during an experiment. Valid frame lengths are shown in the table below:

Sub-band configuration	Valid frame lengths (bytes)
4 x 16MHz	1000, 1024, 1280, 1600, 2000, 2560, 3200, 4000, 5000, 5120, 6400, 8000
2 x 32MHz	as above plus 2048
1 x 64MHz (not implemented)	as above plus 2048, 4096

When two polarizations are used they must be transmitted in separate packets.

One packet contains one VDIF frame. The VDIF station ID and thread ID fields are ignored; instead a UDP port number is hard coded into the FN firmware for each station and sub-band. Because the same UDP port numbers are repeated in every FN, the combination of IP address and UDP port number is needed to fully identify each data stream. The UDP port to stream mapping is defined in [3].

Data inflow must start on a second boundary. The time field in the VDIF header is compared to a preset start time. After the start time data are stored in a 4 second deep circular buffer in a slot determined by the second, epoch and frame-within-the-second fields in the VDIF header. When data from different VDIF epochs are combined, the control system provides the required offset to convert each station to a common epoch prior to correlation.

Synchronization of Data and Delay Model

The control system coordinates sending data and delay model information to the UniBoard, and is thus aware of the fullness of the circular buffer. Data are read from the circular buffer and correlated when the control system instructs the UniBoard to process a batch of N integration periods. The integration period is set by the control system to an integer number of FFT periods (1 FFT period = 2048 samples). Real time and pre-recorded data are treated identically from the point of view of the UniBoard. Further details of the synchronization mechanism are given in [4].

Time Multiplexed Station Processing

The filter bank, including the blocks from Mixer to Normalize in Figure 2.3, has four input channels to process data from two dual-polarization stations simultaneously. Since its throughput is four times faster than real time, eight stations can be processed in four sweeps. The switching is done at the start of an integration period, with the filter-bank outputs being held in the corner turner memory until data from all eight stations has arrived. Once all eight stations have passed through the filter-bank, correlation begins while the filterbank processes the first pair of stations for the next integration period.

The system clock is 2% faster than real time to allow for some dead time between integration periods.

The aggregate real time bandwidth of the filterbank is 64MHz per station per polarization, currently configured as four bands of 16MHz. A second configuration, two bands of 32MHz, is currently under test and a single 64MHz band configuration is planned. The FPGAs must be re-programmed to switch between the configurations.

Delay and Phase Correction - Introduction

The control system sends a set of delay and phase coefficients per integration period. The delay models are per station, and the phase models are per sub-band per station. The coefficients are held in a FIFO until the correlator is instructed to process the corresponding data segment. The polynomial order and coefficient resolution required for the models to remain valid over the

maximum one-second integration are discussed by Small [6]. To summarize:

Delay model

The control computer calculates a geodetic delay model for each station and transmits it to the FN FPGAs as coefficients of a quadratic polynomial of the form

$$\tau = d_0 + d_1 t \quad \text{Equation 2.1}$$

in which t is time during the period during which the coefficients d_0 , d_1 are valid, and τ is the delay correction for that station.

Phase model

A phase correction is required because the delay correction is done at sub-band frequency, not sky frequency. This is calculated per band per station by the control system as

$$\phi = p_0 + p_1 t + p_2 t^2 \quad \text{Equation 2.1}$$

The following section provides more details on how the models are evaluated and applied in the UniBoard.

Delay and Phase Models - Implementation⁴

In the first order delay polynomial (Equation 2.1) t is simply a count of the FFT number within each integration period, since the UniBoard evaluates the delay model once per FFT. The coefficients d_0 and d_1 are 32 bit signed integers.

Coefficients are updated from the FIFO at the start of an integration period. The polynomial is evaluated once per FFT period by a simple accumulator. The delay is scaled such that the top 32 bits of the calculated delay represent the number of whole samples to adjust the circular buffer read pointer. The next 8 bits are fed to a look up table to translate the fractional time delay to a per-frequency bin phase rotation. This phase correction is applied to the data after the FFT in the `fn_power` module shown in figure 2.3, while the integer part is applied in the `pkt_rx` module.

For the second order polynomial (Equation 2.2) the three coefficients (p_0 phase, p_1 phase rate, and p_2 phase acceleration) are 48 bit signed integers. In this case t is a count of the sample number within the integration period.

As for delay, the phase coefficients are updated from the FIFO at the start of an integration period. The phase polynomial is evaluated every sample using a two stage accumulator. The top 9 bits of the phase accumulator, representing a 2π rotation full scale, are applied to the complex mixer in the `fn_dlm` module in Figure 2.3.

⁴ Note on transitional implementation. The delay model was initially implemented with 32 bit coefficients and the phase with 48 bits. This is the transitional implementation referred to in Small [5] and is currently in use as of June 2014. Work is currently underway to widen the coefficient signal paths (coefficient FIFOs, registers and adders) to support 48 bit for delay and 64 bits for phase.

Coefficient Resolution, Storage and Bandwidth

Both the delay and phase model coefficients are updated per integration. Given a realistic minimum integration time of 256 FFT periods, and a maximum of one second, the coefficients need to be updated between 1 and 61 times per second. The UniBoard provides a 128 word deep FIFO so that the control computer can send coefficients a second in advance without overflowing.

The coefficient storage requirement for the delay models is

$$32 \text{ bits} \times 2 \text{ coeffs} \times 8 \text{ stations} \times 128 \text{ fifo depth} = 64\text{k bits}$$

The coefficient storage requirement for the phase models is

$$48 \text{ bits} \times 3 \text{ coeffs} \times 8 \text{ stations} \times 4 \text{ sub-bands} \times 128 \text{ fifo depth} = 576\text{kbits}$$

The bandwidth needed to transmit half this data each second is approximately 416 kbps per FN.

Evaluation of the Models

Figure 2.5 shows the path of the delay coefficients through the FIFOs and into the evaluator. At the start of an integration period, the next pair of coefficients is read from the FIFO into the delay and delay rate registers as shown by the green arrows. Note that the delay is inserted at bit 20, effectively multiplying it by 2^{20} relative to the delay rate. This allows the relatively large constant delay, and much smaller delay rate to be represented by only 32 bit coefficients.

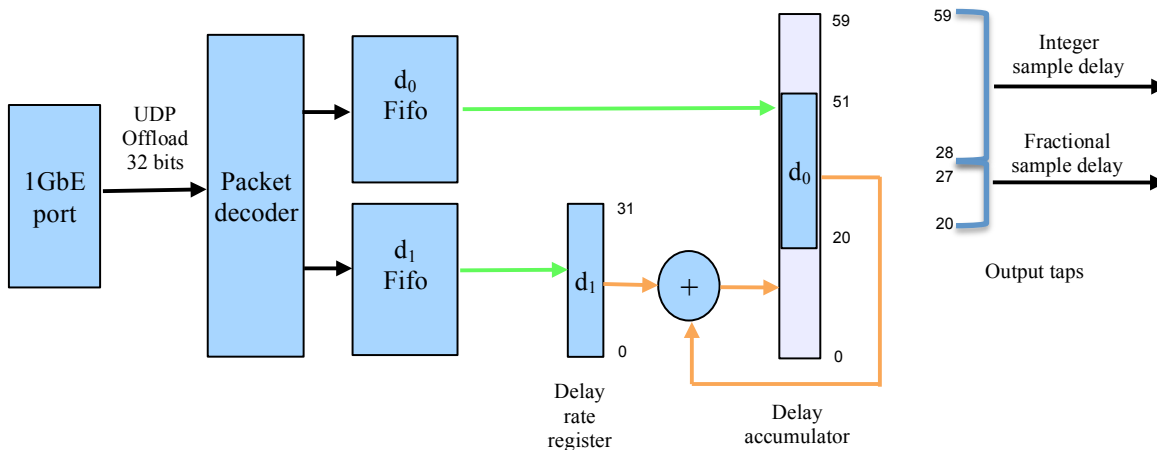


Figure 2.5: Evaluation of the Delay Model

After every FFT period the delay model is evaluated by adding the rate onto the value in the accumulator, along the path shown by the orange arrows in Figure 2.5. The new delay value is tapped off at the bit positions shown to the right of the Figure. The 32-bit integer part is sent to the logic streaming data from the circular buffer to the filter bank. When its value changes by ± 1 , one sample is skipped or repeated.

The next eight bits represent the fraction delay to the nearest $1/256^{\text{th}}$ of a sample. They are delayed by four FFT periods to align the correction with the centre of the filter bank window, and then applied as a per-frequency bin phase correction to the FFT outputs. Figure 2.7 later in this section shows where the delay and phase models are applied to the data.

The phase models are evaluated using the accumulators and registers shown in Figure 2.6. Again the green registers are loaded with fresh p_0 , p_1 and p_2 coefficients from the FIFO (not shown) at the start of an integration period. The phase models are evaluated every sample while data is flowing. All the registers and accumulators are 48 bits wide and permitted to overflow since phase can wrap round. The top nine bits of the output register are tapped off and fed to the sine/cosine lookup tables in the mixer module as shown in Figure 2.7.

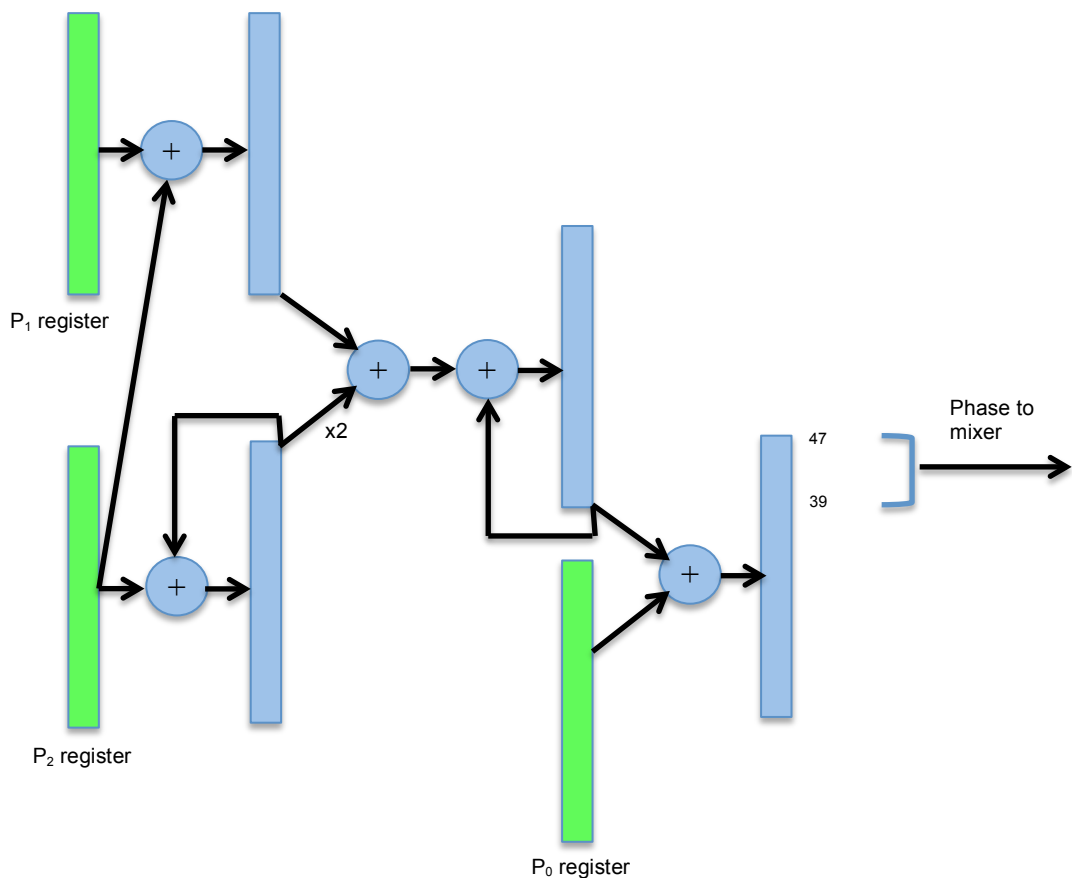


Figure 2.6 Evaluation of the phase model

The system of adders and accumulators evaluates the following:

$$\phi = p_0 + \Sigma(p_1 + p_2 + 2\Sigma p_2)$$

Iteration	Value
p_0	
$p_0 + p_1 + p_2$	
$p_0 + 2p_1 + 4p_2$	
$p_0 + 3p_1 + 9p_2$	
$p_0 + 4p_1 + 16p_2$	
$p_0 + 5p_1 + 25p_2$	

The values in the gradient and sine/cosine look-up tables were generated using Matlab.

Phase and Delay Coefficient Transmission

Coefficients are transmitted to the FPGAs via the UDP offload port of the 1GbE module. The packet format is shown below. Note that this is the format used in the transitional implementation, not the final version.

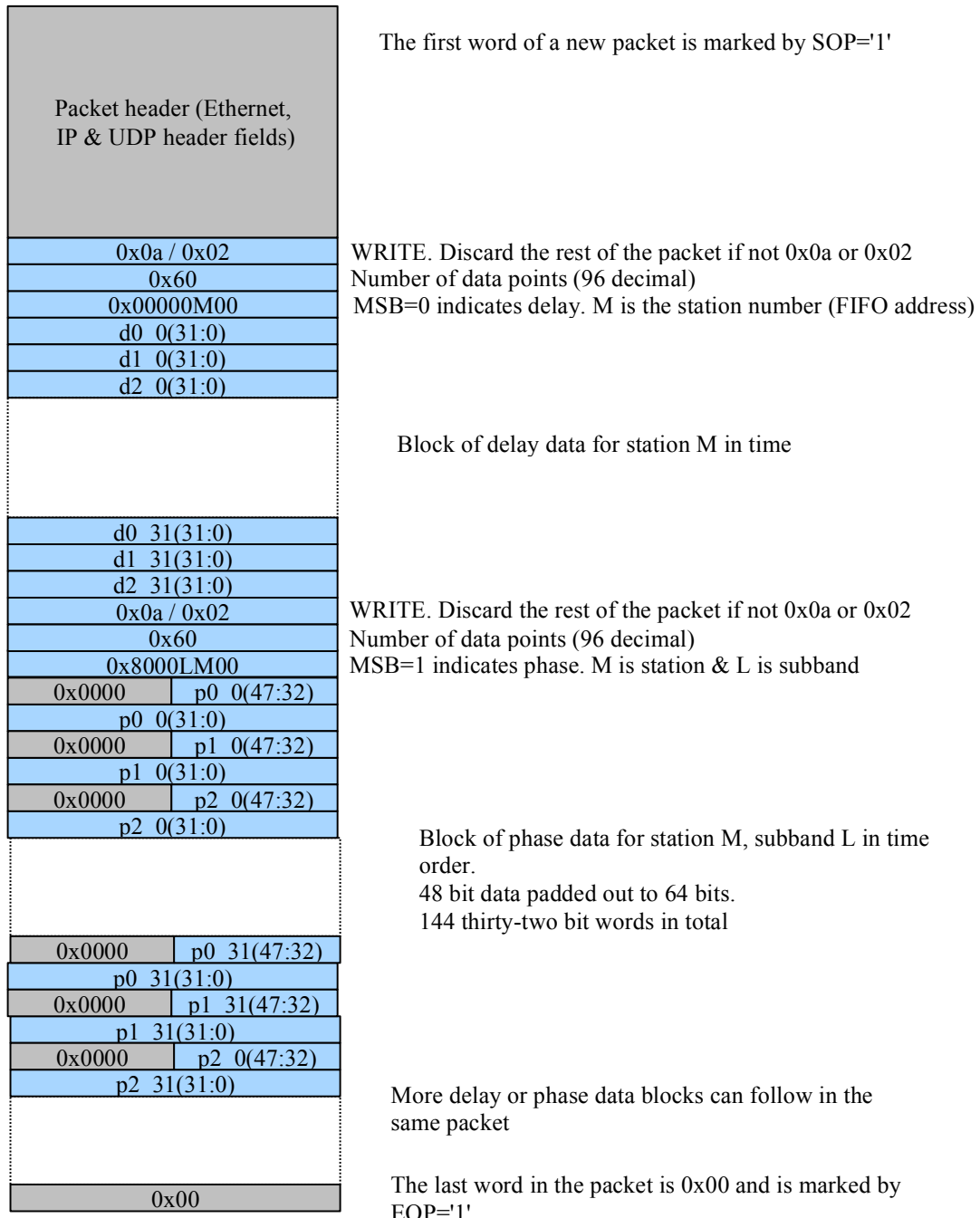


Figure 2.8: UDP Delay/Phase Coefficient Packet Format

Polyphase Filterbank

The filter bank is a complex 2048-point (per sub-band) polyphase design with a window length of 6 taps per point. It is implemented as a double-rate design with throughput twice the input sample rate. At the output the duplicate frequency bins are dropped leaving a 1024-point single sided spectrum whose data rate matches the input. The frequency bin size is 15.625kHz for a nominal 64MHz aggregate input bandwidth.

Data from the circular buffer first enter a quadrature mixer (block `fn_dlm` in Figure 2.3) where they are converted to complex, Doppler shift corrected, samples. They then pass to the polyphase filter bank, composed of a pre-filter structure and FFT (`fn_ppf_dr_4`). The pre filter structure applies a Blackman Harris window function, chosen to give good out-of-band rejection at the expense of a somewhat rounded frequency bin shape, as shown in the simulation in Figure 2.9.

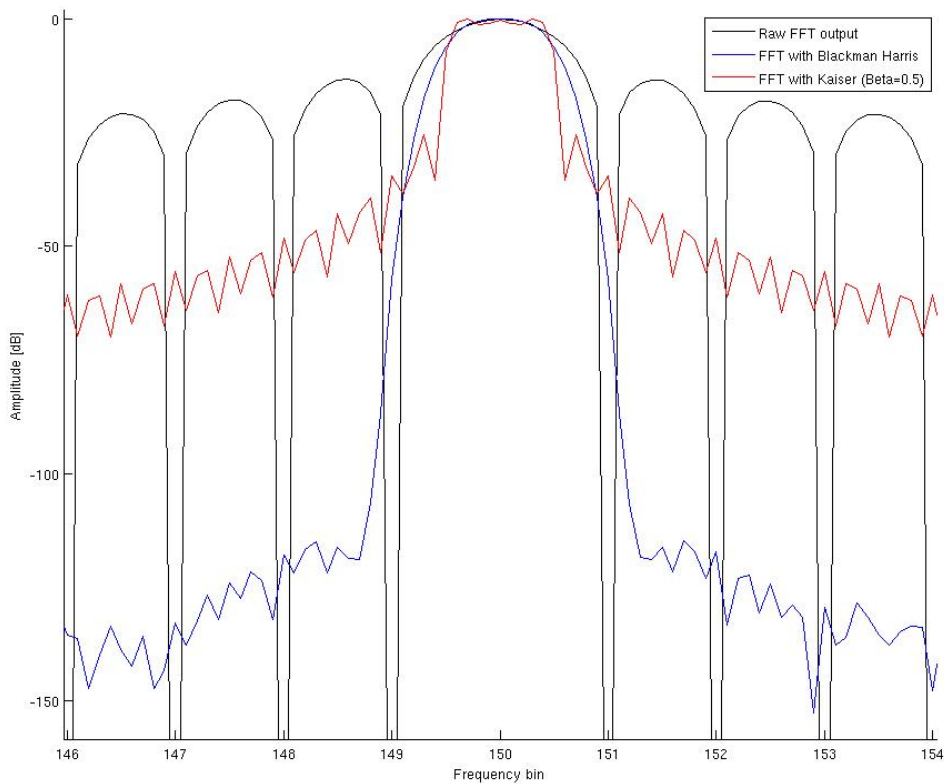
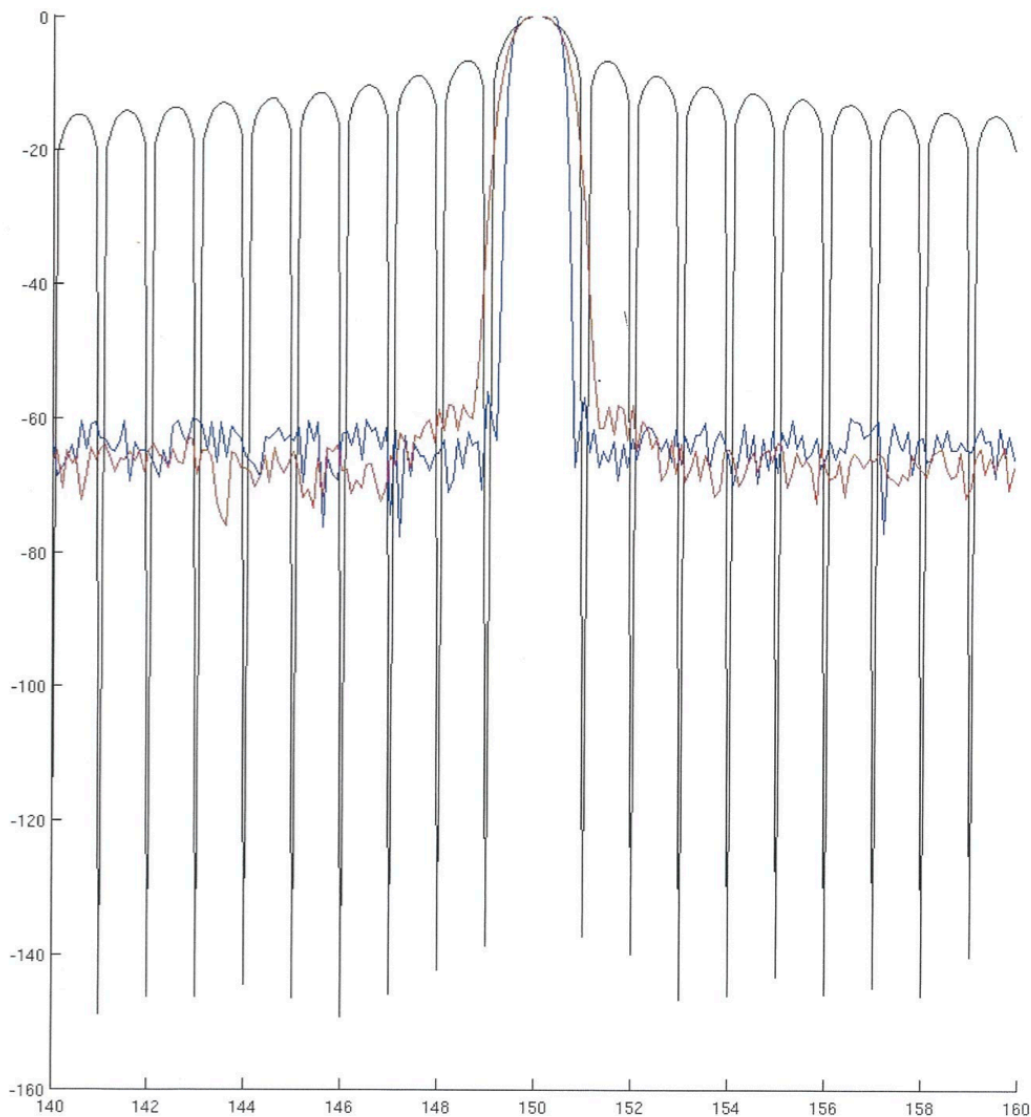


Figure 2.9 *Comparison between Blackman Harris window, Kaiser window and FFT*

In this Matlab simulation, the signal strength in bin 150 was recorded as the input signal was swept from bins 146 to 154 in steps of a tenth of a bin. The simulation used the same 16 bit fixed point coefficients as the hardware, but a floating point FFT.

Figure 2.10 compares Blackman-Harris windows with 6 and 16 taps per frequency bin (for a 1024 point FFT). The longer window has a narrower central lobe, and therefore better noise rejection from neighboring bins. However further out the rejection is not significantly better for the 16-tap version. Since memory available for the taps window is limited in the FPGA, 6 was chosen for now.



black = fft only
 red lines = Blackman-Harris 6144 taps (bin 149: -28.6 dB)
 blue lines = Blackman-Harris 16384 taps (bin 149: -56.14 dB)

Figure 2.10 Comparison between Blackman Harris windows with 6 (red) and 16 (blue) taps per point (1024 point FFT integrated over 128k samples)

Note that although the length of the window is limited by hardware constraints, it is possible to load an alternative weighting function into the coefficient store at run time. The procedure for generating a new set of coefficients in Matlab is described in [7]

Architecture

The architecture of the pre filter structure and FFT is shown in Figure 2.11. Data enter a shift register with taps every 2048 samples. The newest sample enters from the left while the next tap selects the sample 2048 clocks earlier and so on. The taps are fed to multipliers where the data are

weighted with the selected values from the coefficient memory. On each clock the data shift one tap to the right, while the coefficient selector moves to the next row. After 2048 clocks the coefficient selector returns to row 0 at the same time as the first sample emerges at the second tap.

The outputs from the six multipliers are added together and fed into the FFT module. Note that the data entering the pre filter structure are complex, so the weights are applied to both the real and imaginary parts.

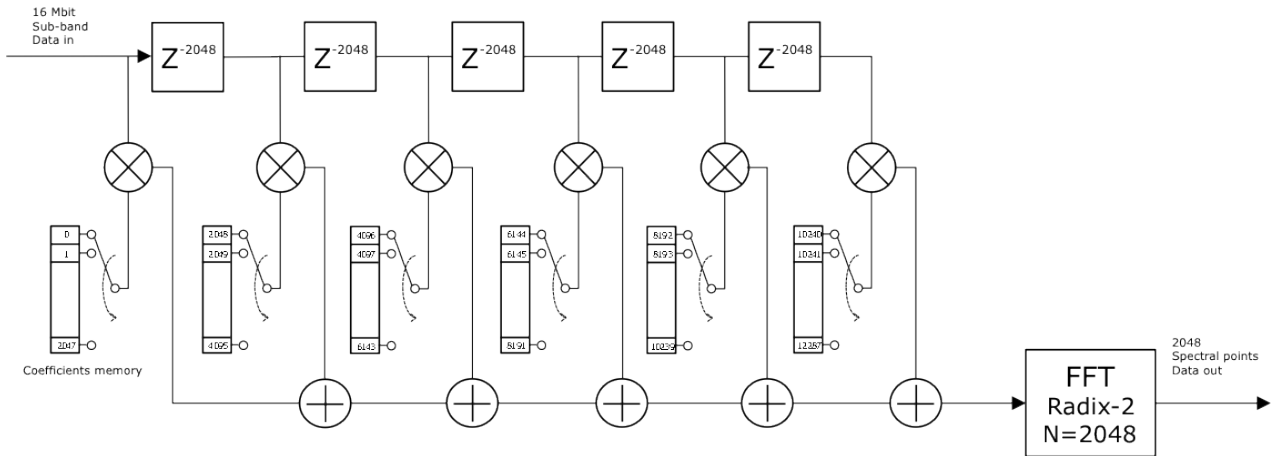


Figure 2.11 Architecture of the pre filter structure

As previously mentioned, the filterbank processes two stations with two polarizations simultaneously. In the current configuration, the aggregate 64MHz bandwidth is comprised of four sub-bands A, B, C & D. In fact the filter banks are implemented as two time-multiplexed branches, with the top branch processing A and C, while the lower branch handles B and D. After the FFT, when the duplicate spectral points are discarded, the four sub-bands are merged into a single 4-times time multiplexed stream. This arrangement is repeated for the other three channels as shown in Figure 2.12 below.

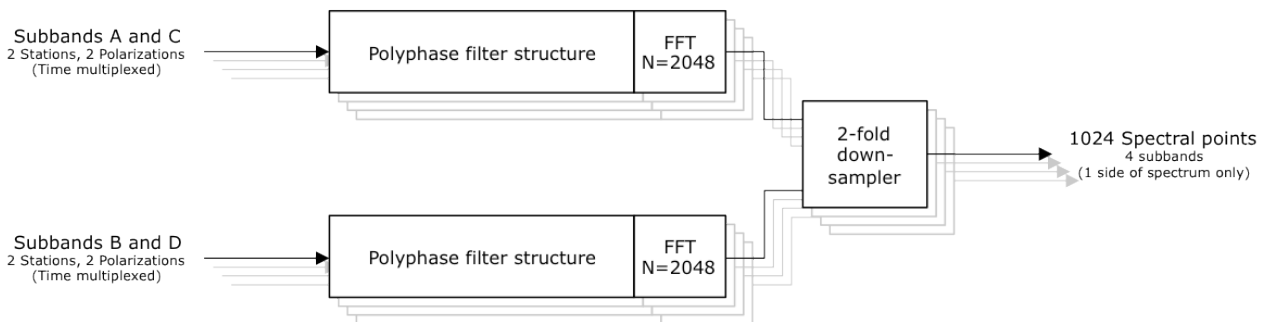


Figure 2.12 Time-multiplexed filter banks for 4 sub-bands, 2 stations, 2 polarizations

After the filter bank the sub-sample part of the delay correction is applied and the outputs truncated to 8 bits complex. The tap point for the truncation can be adjusted via the control system. Next the framer module divides the aggregate spectrum into four equal parts and transmits a quarter to each of the BNs. In this way the same chunks of spectrum from the four FNs are brought together for correlation in a single BN.

Corner Turner

The corner turner is needed because there is not enough memory in the FPGA to hold accumulated products for all 1024 frequency bins and 2112 correlation products. Instead the data for all stations and both polarizations are built up in one of the DDR memories over an integration period. When the data set is complete they are read out in frequency bin order: all the data for a given frequency bin are correlated and the products dispatched to the back end computer before moving to the next frequency bin.

The two DDR memories are swapped every integration so that one is always available for new data while the previous data set is being correlated. This allows continuous data flow and efficient read/write access to the DDR. The corner turner has a couple of advantages: data can be read out in natural frequency order at no extra cost, and it permits the time multiplexed filter bank architecture discussed earlier. A disadvantage is that the maximum integration time is limited by the available DDR memory size: one second with current 4GB modules.

Correlation Engine

The 2112 correlation products are computed by 132 complex multiply-accumulate cells which each calculate sixteen products sequentially. The throughput matches the input rate at a nominal 256MHz clock, but as in the FN the clock is run 2% faster to allow for dead time between integrations.

In Figure 2.13 below the black dots represent the cross correlation MAC cells and the red dots the (identical) auto correlation cells. The numbers below and to the right represent the 32 stations presented to the MAC cells during the first four passes. During these four passes all pol 0 x pol 0 products are calculated. A further 12 passes then compute pol 0 x pol 1, pol 1 x pol 0 and pol 1 x pol 1 for all 32 stations.

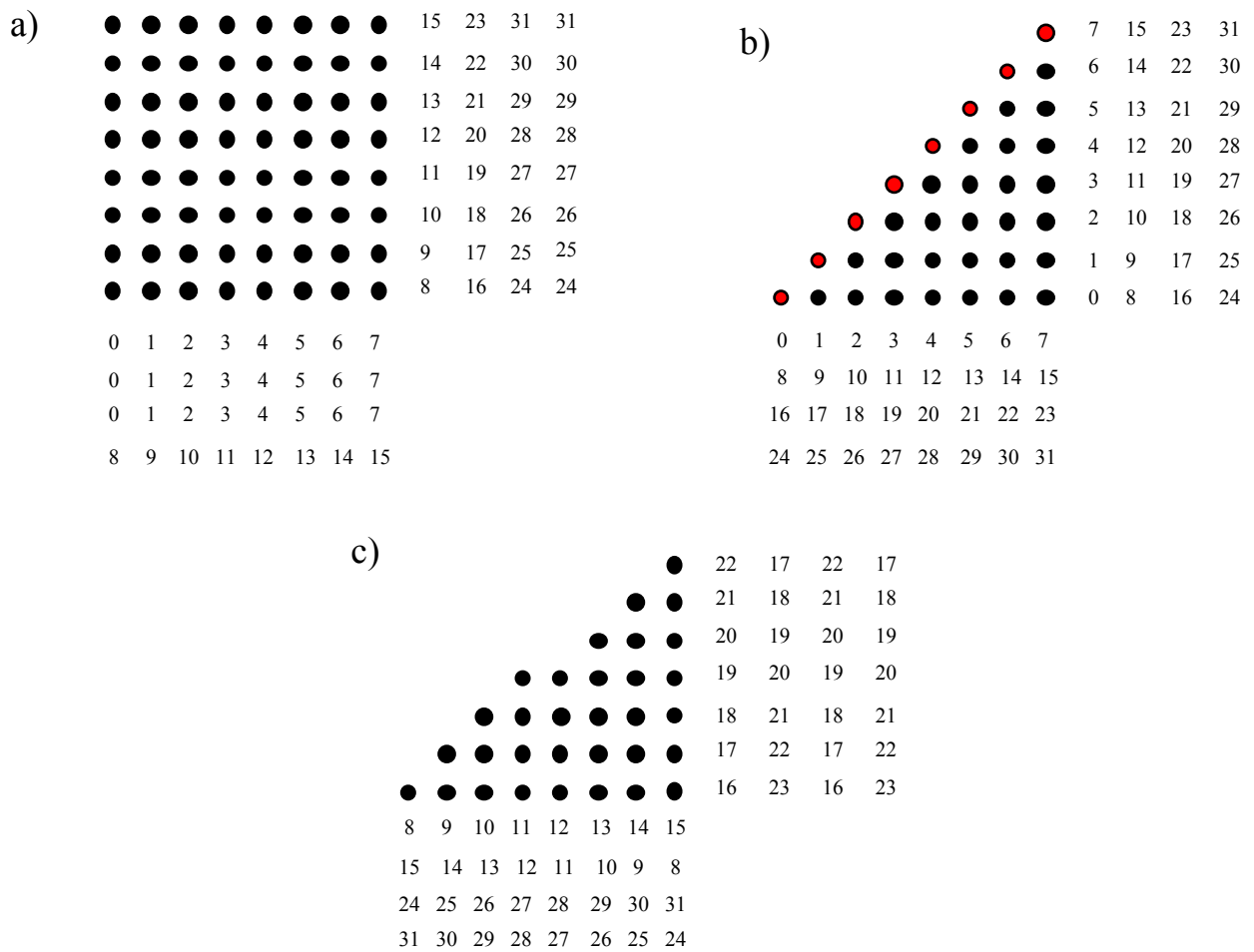


Figure 2.13: *Correlation Points Processed by 132 MAC Cells*

Figure 2.14 illustrates the architecture of a single multiply-accumulate cell. The two complex input signals $a+jb$ and $c+jd$ are fed in from the left. For autocorrelations $a=c$ and $b=d$. The accumulator memory is a 72-bit wide, 16 word deep dual port RAM. On every clock one of the 16 intermediate results is read out and fed to the accumulator adders with the correct pipeline delay to be combined with the next input value for that pair of stations.

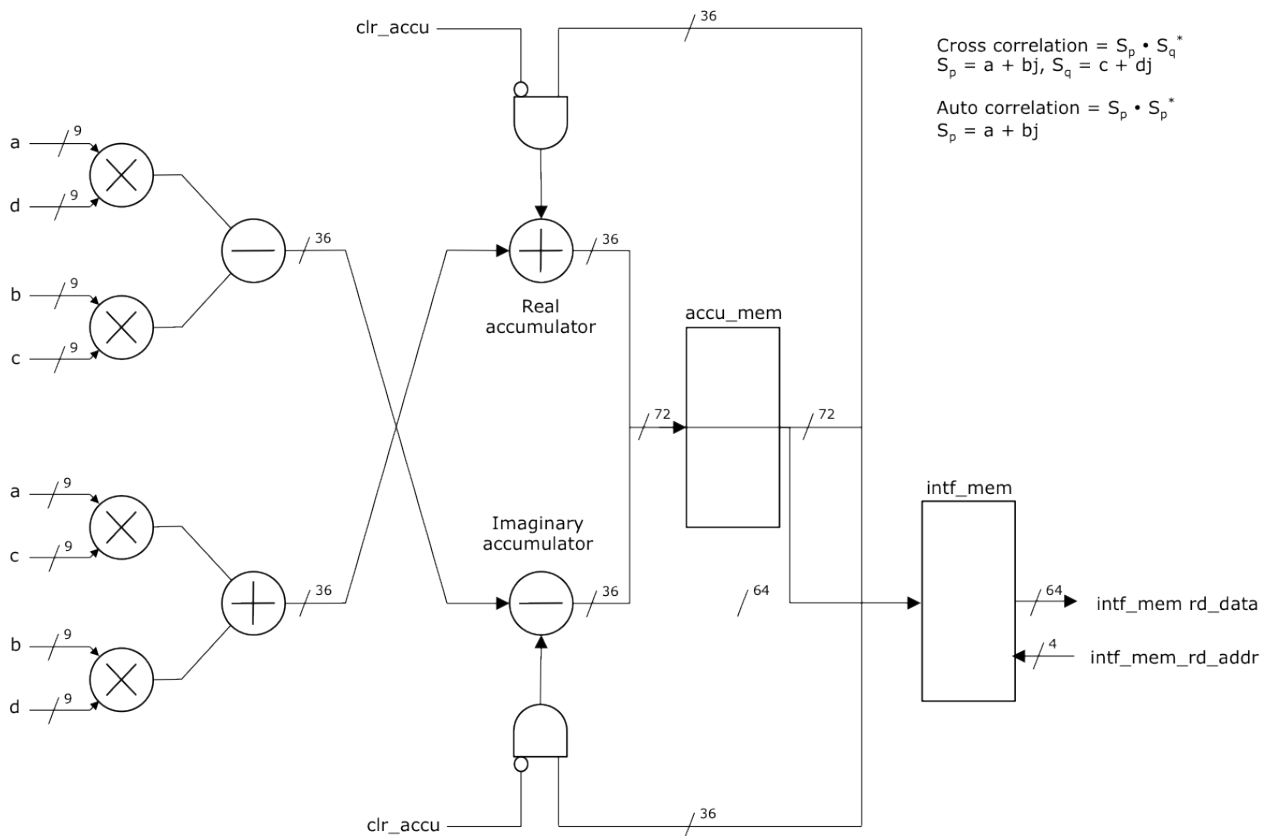


Figure 2.14 *Correlator Multiply-Accumulate Cell*

The cycle of sixteen passes is repeated with data from each FFT period in the integration time. On the last cycle the accumulated products are transferred to a dual port memory, the interface memory, where they can be read out and dispatched over the ten gigabit Ethernet port. The next frequency bin is then processed, starting with `clr_accu` held high for the first cycle to clear the accumulator RAM.

It is not necessary to send all 2112 products over the port. The control system knows the mapping between the FN input streams and the products calculated in each BN. It sets bits in a register bank to select which products must be sent. The output formatter logic reads the selected products from via the interface shown on the right of Figure 2.14 while the next frequency bin is being processed. Implementation details are given by Pirruccio [8].

Validity

Validity bits are carried through the correlator in parallel with the data. The validity of the arriving data is determined and stored per VDIF frame: the frame either arrives or it doesn't; if it arrives the VDIF valid bit indicates the validity of the entire frame. All the bits in the validity store are initially '0'. When a valid packet arrives the bit for its row is set '1'. When, later, that row has been completely read out from the buffer its validity bit is reset to '0'. If a valid packet does not arrive to fill that row by the time it is read again, the validity bit remains '0'.

As data are read from the buffer and fed into the filter bank, the corresponding validity bits are

sampled. The validity of the FFT output is calculated such that the whole of the FFT is marked invalid if any of the contributing data are invalid. Since a 6-tap polyphase window is used, the contributing data include the previous six FFT periods, that is 2048×6 samples. The first six FFT periods during an integration period are always invalid while the polyphase structure is filling with new data. Details of how the validity is calculated are given by Pirruccio [9].

The framer module (fn_framer in Figure 2.3) checks the validity status of each FFT output and substitutes zeros in all frequency bins of an invalid FFT. The invalid, zeroed, frame is then passed to the BN and correlated in the same way as valid data, but does not contribute to the products.

The validity bits are corner turned along with the data and accumulated in a parallel 'validity accumulator' simultaneously with the data correlation. Thus for every correlation product a corresponding validity count is generated which can be used to normalize the data.

Bit Truncation

Input

The data read out from the circular buffer are padded to 8 bits regardless of their original sampled resolution. The mixer LO signal is a 9-bit signal from a cosine lookup table whose phase input is also 9 bits. The mixer is implemented with 9-bit multipliers whose output is truncated to 14 bits before entering the polyphase filter bank.

Filter bank

In the pre-filter structure, the 14-bit data are multiplied by 18-bit coefficients and the outputs of three multipliers summed to 35 bits, of which 12 bits are passed on to the FFT.

The FFT truncates to 18 bits at every stage, and drops a bit every stage to prevent bit growth. Simulation showed that this gave identical output amplitude to the radix-4 Lofar FFT, and that it would be possible to drop a bit only every second stage without overflow.

The fine delay phase rotations are applied to the FFT outputs using an 18 bit complex multiplier to give a 37-bit result. This is then truncated to 9 bits at the point set by the control system.

The truncations before and after the FFT are balanced (ie both positive and negative numbers are truncated towards zero) to avoid introducing DC or wideband bias.

Corner Turner

Nine bit data are transported to the BNs, but it was found inefficient to store them in 64 bit wide DDR. So they are truncated again to 8 bits in the corner turner.

Correlator Engine

The correlator engine is built from 9 bit complex multipliers and accumulates the results to 36 bit real plus 36 bit complex. Similarly the validity accumulator is equipped with 32 bit accumulation registers.

Correlator Product Output

Frequency ordered Architecture

Because of the corner turning architecture, the correlation engines process the frequency bins sequentially within an integration period. After each frequency bin integration all single-, cross- and auto-correlation products are finished and must be dispatched to the backend computer while the next frequency bin is processed.

Each BN currently contains one correlation engine that can process 1024 frequency bins and 2112 products. The validity bits are accumulated in parallel with the data and sent in the same packet as the data.

Output Format

The correlation products are sent to the backend computer in UDP packets. The packet has a four 32-bit word header to identify the data by its frequency bin, correlator engine, FPGA and UniBoard number. Two 32-bit word fields contain the 'time stamp'. In fact this is simply a count of the number of integrations in the scan, from which the back end processor can calculate the time of the first sample in the integration period.

A packet may contain any number of correlation products plus their validity counts, provided the MTU of 9000 bytes is not exceeded.

For integration times $< 0.5s$ the products fit into two 32 bit words. For longer integrations the data are padded to 64 bit words before transmission to the backend computer. A flag in the header denotes whether the data fields are 32-bit or 64-bit. If there are an odd number of products in a packet, one additional padding product with zero real, imaginary and validity fields, is appended at the end.

The following table shows the first 7 words for the 32-bit case. For 64-bit data the real part of the first product would occupy words 4 and 5, the imaginary part words 6 and 7, and the validity count word 8.

Word 0	VVV ₃	F ₁	Chip ID ₈	CE ₂	reserved ₆	Frequency bin # ₁₂
Word 1	reserved ₈		Payload size ₁₂		First product # ₁₂	
Word 2	Integer no of seconds since epoch to start of integration period ₃₂					
Word 3	No of samples since last second to start of integration period ₃₂					
Word 4	Data for first product (read) ₃₂					
Word 5	Data for first product (imaginary) ₃₂					
Word 6	Data for first product (validity) ₃₂					

First 7 words of an output packet for 32 bit correlation products. Msb at left.

<i>Word 0</i>		
Bits 0-11	12	Frequency bin number (0-1023)
Bits 12-17	6	Reserved
Bits 18-19	2	Correlator engine number within an FPGA
Bits 20-27	8	Chip (node) ID. Bits 20-22 are the FPGA; bits 23-27 are the board number.
Bit 28	1	Flag to indicate 32/64 bit data representation (0/1)
Bits 29-31	3	Header version code (0-7) default 0
<i>Word 1</i>		
Bits 0-11	12	Number of the first product in this packet (0-2111)
Bits 12-23	12	Payload size. The number of products in this packet (not including padding when payload size is odd).
Bits 24-31	8	Reserved
<i>Word 2</i>		
Bits 0-31	32	Integer no. of seconds at start of integration period
<i>Word 3</i>		
Bits 0-31	32	Sample no. within second at start of integration period

Function of the header fields in the output packet

Appendix A FPGA Resources

Totals include other minor modules, signal taps and so on.

Front Node

Stage	Multipliers (18x18 equivalent)	ALUTs	SRAM(kbits)	Registers
Packet Receiver	4	17261	2434	19910
Nios Controller	4	2489	333	1642
1GbE Control port	0	3713	88	5555
Delay Module	4	9915	928	14383
Complex Mixers	16	440	9	552
Filter Bank	472	23854	6469	33263
Interface to BN	0	2296	4	3135
Total/Available	500/1288	70743/182400	10354/14200	85543/182400

Back Node

Stage	Multipliers (18x18 equivalent)	ALUTs	SRAM(kbits)	Registers
Interface from FN	0	1211	3	1035
Nios Controller	4	2316	332	1459
1GbE Control port	0	3705	88	5468
Corner Turner	0	12442	1435	13381
Correlator Engine	528	15113	152	29578
10GbE Product out ports	0	9254	134	5557
Total/Available	532/1288	52870/182400	2204/14200	68528/182400

References

- [1] “The UniBoard Correlator System – An Overview”, Verkouter, H., JUC Memo 16, June 2014
- [2] “The UDP/IPv4 FPGA/UniBoard command protocol”, Verkouter, H., updated 2 October 2012
- [3] “EVN Correlator Startup Guide”, Hargreaves, J. E., JUC memo #4, 18 October 2012
- [4] “Correlator timing and synchronization”, Verkouter, H., JUC memo #11, 11 October 2012
- [5] “VLBI Data Interchange Format (VDIF) Specification”, Release 1.0, Ratified 26 June 2009, Madrid, Spain. Available on the UniBoard Wiki
- [6] “JIVE Uniboard Correlator Review Memo: The Delay Model”, Small, D., JUC Memo 14, updated 27 June 2014
- [7] “Filterbank Coefficients Generation”, Salvatore Pirruccio, JUC Memo 13b, May 13, 2014
- [8] “Product Table”, Salvatore Pirruccio, JUC Memo 13c, May 13, 2014
- [9] “Validity Store and Processor”, Salvatore Pirruccio, JUC Memo 13f, May 13, 2014