Uniboard Digital Receiver
Design document

G. Comoretto[1], A. Russo[1], G. Tuccari[2], A. Baudry[3], P. Camino[3], B. Quertier[3]

Arcetri Technical Report 5-2011

[1]INAF - Osservatorio Astrofisico di Arcetri
[2]INAF - Istituto di Radioastronomia
[3]Université de Bordeaux 1, Laboratoire d'Astrophysique de Bordeaux (LAB)

Revision 3.0 - March 2011

Abstract

The Uniboard is a project for a general purpose board, containing several large FPGAs, to be used primarily as a general component for the next generation VLBI correlator. The Digital Receiver Working package deals with the implementation of a wide-band frequency demultiplexer, that divides the received radio band (up to a bandwidth of 4 GHz) into enough narrow band channels (128 MHz bandwidth or lower) to completely cover it. Each channel can be independently positioned across the input band, and may have a different bandwidth and signal representation (number of bits per sample). Output is formatted as VDIF packets, encapsulated in UDP packets routed over eight 10Gb fiber links.

The document describes the general architecture and performance of the proposed design.

Contents

1 Introduction

1.1 Scope

In this report the Digital Receiver application for the Uniboard hardware platform[1] is described. The report is a revised version of the Digital Receiver Initial Design Document [2]

The document starts with a very general description and high level specifications in chapter 2. Chapter 3 contains a detailed functional block description. Chapters 4 and 5 contain a more detailed description of the implementation for each component, respectively in the back node and front node FPGAs. A summary of the programming interface is contained in chapter 6.

Separate documents contain:

- Detailed specifications for each component,[3] with expected performances, filter shapes, latencies, etc.
- Programming interface,[4] more detailed than the summary contained in chapter 6.
- Structure of the design,[5] and description of the main VHDL modules
- Simulation testbench and simulation results[6]

1.2 Glossary

The following terms and acronyms are used in this document:

ADC: Analog to Digital Converter. An ADC converts a continuous signal to a discretely sampled digital signal, represented by a finite number of bits.

Bandwidth: The frequency span of a signal. For a complex signal it is equal to its sampling rate, for a real signal it is half the sampling rate. The **effective bandwidth** is the portion of the available bandwidth that is not distorted/attenuated by the signal processing operations.

BN: Back Node. The four FPGAs closer to the Uniboard backplane, and connected to the ADC input busses.

Channel: A portion of the input signal in the frequency domain, as processed by the digital receiver. In this document we will use the

- **FFT channel:** The portion of the input signal at each FFT output port.
- **DBBC channel:** The portion of a FFT channel filtered and frequency translated by a DBBC.

Channel group: A set of four FFT channels that share the same outputs of the FFT processor first section. Described in chapter 3.1.

DBBC: Digital Baseband Converter. A component that selects a portion of its input signal, converts it to near zero frequency, and filters it to a selectable bandwidth.

DDR: Double Data Rate. A method for transferring clocked data using a clock with half the data rate. Input data are latched on both edge of the clock.

DDS: Direct Digital Synthesizer. A digital component that produces a sinusoid of prescribed frequency.

Digital receiver: A system that processes a radio signal using only digital components. The signal to be processed is sampled by an ADC and down-converted, filtered, resampled by various digital components. [1]

DSP: Digital Signal Processing. Any technique used to digitally process a signal. All components in a digital receiver implement DSP operations.

[1] In a common use of this term, it is implied that only signal amplification, not frequency translation, is performed between the antenna and the ADC. Here we do not make this assumption.

E-VLBI A VLBI network in which the antennas are connected to the correlator using high speed Internet connections.

FFT: Fast Fourier Transform.

FN: Front Node. The four FPGAs closer to the front side of the Uniboard, and connected to the high-speed 10Gb links.

FPGA: Field Programmable Gate Array: a digital component whose functionality is defined by a externally loadable configuration. The Uniboard is composed by 8 large FPGAs.

IP: the Internet Protocol

LVDS: Low Voltage Differential Signaling. A standard for high speed digital interfaces, using a couple of wires for each data line.

Polyphase filter: A filter in which different samples are convolved by a different function. Polyphase filters can be used to control the shape of the FFT channels.

PRNG: Pseudo Random Noise Generator. A digital component that produces a deterministic, but highly randomic, sequence of numbers, used to simulate real noise.

Recirculation: A technique in which the same component is used at a higher clock rate to perform several operations at a lower clock rate. In the DBBC the filter multipliers are recirculated to compute the signal convolution on a much longer filter length.

SFDR: Spurious Free Dynamic Range: is the ratio (usually in dB) between the maximum input signal (could be an interferring signal) and the strongest spurious ghost signal.

UDP: User Datagram Protocol. One of the main Internet protocols, with minimum overhead, packetized transport mechanism, and no control over possible packet loss.

Uniboard: The general purpose board used in this project.

VDIF: A standard format for VLBI data. Described in [8].

VHDL: VHSIC Hardware Description Language. A language used tor describe high-speed digital circuits. VHDL circuital description can be translated by an appropriate program to the corresponding circuit implementation on a FPGA.

1.3 References

References

[1] Sjouke Zwier, Gijs Schoonderbeek: UniBoard V2.0 Board Description (2011)

[2] G. Comoretto, A. Russo, G. Tuccari, A. Baudry, P. Camino, B. Quertier: *Uniboard Digital Receiver Initial design document* (2009)

[3] G. Comoretto, A. Russo, G. Tuccari, A. Baudry, P. Camino, B. Quertier: *Uniboard Digital Receiver Specification Document* (2011)

[4] G. Comoretto, A. Russo, G. Tuccari, A. Baudry, P. Camino, B. Quertier: *Uniboard Digital Receiver Programming Manual* (2011)

[5] G. Comoretto, A. Russo, G. Tuccari, A. Baudry, P. Camino, B. Quertier: *Uniboard Digital Receiver VHDL Design Description* (2011, in preparation)

[6] G. Comoretto, A. Russo, G. Tuccari, A. Baudry, P. Camino, B. Quertier: *Uniboard Digital Receiver Simulation Testbench and Results* (2011, in preparation)

[7] a W.H. Press, S.A. Teukolsky, W.T Vetterling, B.P. Flannery: *Numerical Recipes, 3^{rd} Edition*, chapter 7. Cambridge University Press (2007)

[8] VDIF task force: *VLBI Data Interchange Format (VDIF) Specification", release 1.0* (2009), `http://www.vlbi.org/vdif/docs/VDIF%20specification%20Release%201.0%20ratified.pdf`

2 General description

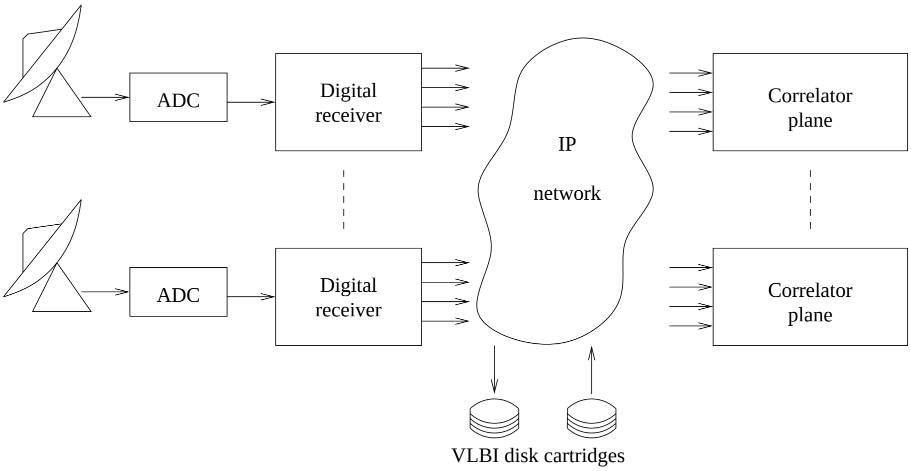

Figure 1: Distributed network correlator. Signal from each antenna is divided into separate bands in the frequency domain, and sent over a general IP network as UDP packets. Packets from all antennas at the same frequency are correlated together in separate correlator planes

The *digital receiver* application is a general purpose signal processing component that divides a wideband input signal into smaller bands, and packetizes these bands into Ethernet packets, to be processed by a separate equipment (e.g. a VLBI correlator). This *frequency demultiplexing* is essential to process the wideband signal using digital equipment that has a clock rate much lower than the signal sampling rate.

A concept for a net distributed VLBI correlator is shown in fig. 1. Several antennas send the radio signal, formatted as VLBI packets, over a fast Ethernet connection. These data can be either directly correlated in a distributed VLBI correlator, if the available IP bandwidth allows it, or stored locally in Ethernet connected VLBI data cartridges. The distributed correlator units (*planes*) process the same frequency region for all antennas, and different planes can be located at completely different geographic locations. The corner turning function (from different frequency regions of the same antenna to different antennas for the same frequency region) is performed by the IP network.

2.1 Specifications

The *digital receiver* part of the correlator must provide frequency demultiplexing of the receiver band, with the highest possible bandwidth and complete frequency coverage. The first version of the Digital Receiver design, described here, will use a 4 GHz (8 GS/s) sampler, and will provide 64 output channels.

These channels are sent as E-VLBI packets over 8 output 10Gb links. With a 50% overhead (8B/10B encoding, IP, VLBI header, collision avoidance) an aggregate bandwidth of 4 GHz (e.g. 64 channels of 64 MHz each) can be sent using 30% or 60% of the total IP bandwidth, respectively for 2 and 4 bit quantization.

The main specifications are:

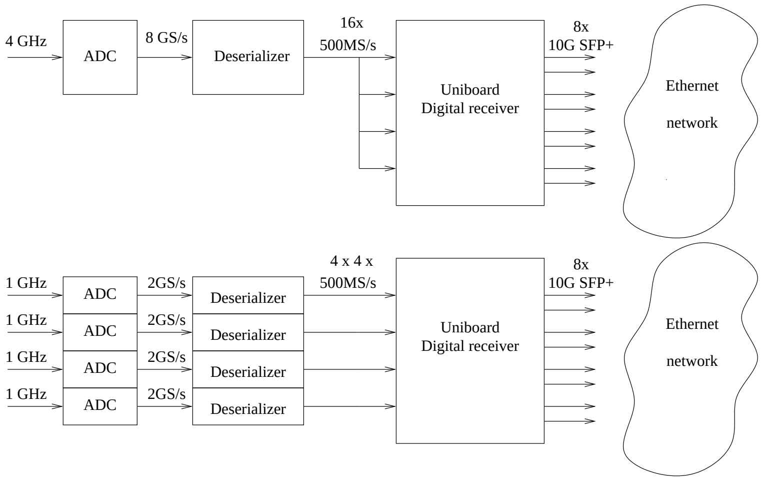

Figure 2: 8 GS/s sampler and 4x2GS/s sampler configurations

- Input band: 4 GHz (8.192 GS/s), [2] with 6 or 8 bit samples. [3] It will be possible to use the board with two 4.096 GS/s or four 2.048 GS/s samplers (fig. 2)
- Input format: 16 byte-wide LVDS inputs, carrying 16 consecutive samples at 512 MS/s (1024 MS/s in the advanced design), with DDR clock. Sample-to-sample skew acceptable. An independent bidirectional I^{2}C line is available for ADC control. The actual format for ADC sample transport is currently undefined, and could involve high speed serial links, e.g. a number of CX4 ports, also available on the board.
- Output band: selectable from 1 to 128 MHz (2 to 256 MS/s), 1 to 8 bit encoding, real samples, selectable USB or LSB, start frequency arbitrarily positionable inside the input band (with some restrictions)
- Output format: Standard VLBI frame, encapsulated in a standard VDIF UDP IP packet. All parameters programmable. Packet size is limited to standard *jumbo packets* (8 kB, 64 kbit)
- Number of output bands: 64, to completely cover the input band with 64 MHz channels (4096 MHz/64 MHz)
- RFI immunity: a out-of-band interfering signal must be attenuated at least by 80 dB

3 General structure

To implement the specifications listed in chapter 2.1, an architecture with a polyphase FFT followed by a series of digital BBCs has been proposed.

The conceptual (simplified) schematic of the design is shown in fig. 3. The input signal is first divided into equispaced bands *FFT channels* by a polyphase FFT, and each FFT channel is further filtered by an array of digital baseband converters (DBBC). Each *DBBC channel* is then formatted and sent over a IP port. Each DBBC can be programmed for an arbitrary position in the input band (with restrictions), and output bandwidth.

Each Uniboard has 16 input LVDS ports, rated for 500 MS/s at 8 bit per port. The total aggregate bandwidth is thus 8 GB/s, enough for a 8-bit ADC with a data rate of 8 GS/s. It is possible to use an

[2] Standard VLBI frequencies are 1 MHz times a power of 2. In this report we will assume this. For simplicity, frequencies multiple of 1024 MHz will be rounded to the next GHz.

[3] No commercial ADC with these characteristics is currently available. A 6 bit 8 GS/s ADC is currently under study at the Laboratoire dAstrophysique de Bordeaux. These specifications are therefore tentative, and subject to changes.

ADC with 4 bit samples, at a data rate of 16 GS/s, but this possibility will be considered in a future extension of the project. A test signal can be internally substituted to the input signal, and used for diagnostic purposes.

The system is physically distributed over the 8 FPGAs composing the board. The 4 back node FPGAs implement the test signal generator, the filter section of the polyphase filterbank and the first 4 stages of the FFT, and the 4 front node FPGAS implement the last 2 stages of the FFT, the DBBCs, and the formatter/IP layers.

The design will use the general framework developed as part of the Uniboard project. Standard components will be used for FPGA-to-FPGA intercommunications, IP formatting, and for the standard control structure. Generic peripherals for the Nios processor use Altera library modules.

All signal processing components are implemented as Nios memory mapped components. Signal processing data path is implemented using Nios streaming interfaces. A simple Nios processor is present in each FPGA in the board. This network of processors, accessible through a dedicated 1Gb Ethernet, can be used for system control and monitor, and to update FPGA personalities locally stored in FLASH.

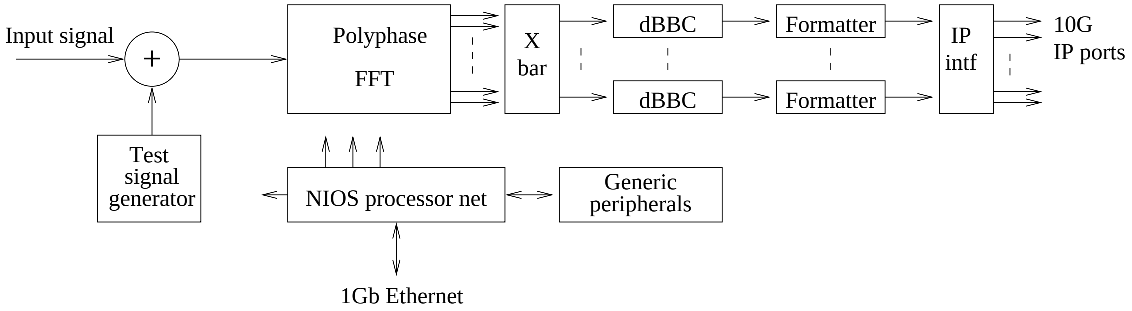

Figure 3: Structure of the digital receiver application. It is composed of a polyphase FFT, that initially divides the input band into 32 sub-bands, and an array of 64 DBBC, with individual VDIF formatters. The system is controlled by a network of Nios embedded processors, communicating through a 1 Gb Ethernet

3.1 Polyphase FFT filterbank

The first stage of the design is a 64 point parallel FFT, with 32 possible complex outputs (only positive frequencies of the input signal) spaced 1/32 of the input bandwidth. Each FFT output has a bandwidth of 1/16 of the input bandwidth, with a 50% overlap between adjacent channels.

The board processes in parallel 16 consecutive samples. The 16 LVDS input ports are distributed among the four *backnode* (BN) FPGAs, with each BN FPGA receiving one quarter of the total samples. Due to this distributed signal input, the FFT block must be distributed among the FPGAs in the board. To minimize vertical interconnections between consecutive samples, a decimation-in-time structure has been used.

Samples are presented in bit reversed order: consecutive samples are sent to BN chips BN0, BN2, BN1 and BN3. This is the natural order if successive stages of 1:2 deserialization are used, as shown in fig. 4. If one or two of the first two stages of time demultiplexing are omitted, the same design automatically produces 2 or 4 independent FFT filterbanks, respectively at 4 or 2 GS/s.

In each FPGA the input signal is further demultiplexed by a factor of 4 (channel spacing of 128 MHz). A polyphase filter structure is then applied to each sample.

The 64 point FFT block is divided among the BN FPGAs, that implement the first 4 stages, and the FN FPGAs, for the last two stages. The first stages are just 16 point complex FFT, and are identical for all BN FPGAs. As the input is real, only outputs with positive frequencies (0 to 8) are independent. Outputs 0 and 8 are real, and are combined in a single complex channel (channel 0), while outputs 1 to 7 are complex. Each channel is 256 MHz wide. These 8 outputs are sent in couples to the front node (FN) FPGAs, where the remaining two stages of the global FFT is implemented.

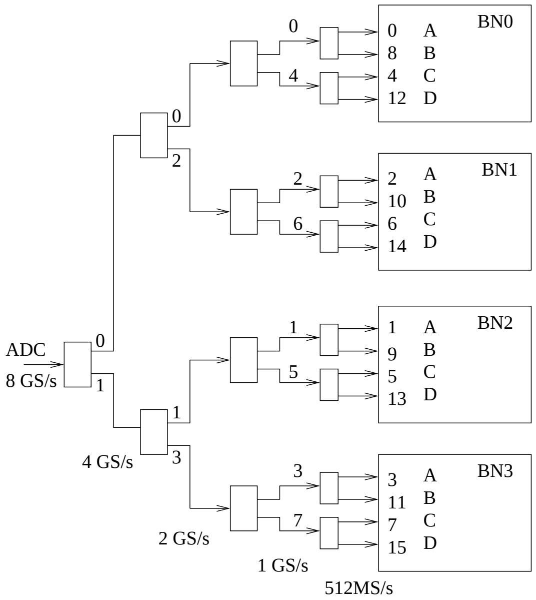

Figure 4: Deserialization of the ADC samples, and time sequence at the Uniboard input ports

The conceptual structure is shown in fig. 5 and 6, where one of 4 identical input FPGAs and one half of 4 output FPGAs are shown. The filled circles represent multiplication by the appropriate exponential (twiddle factors), and empty circles the sum/difference of the inputs. Dashed branches are not computed, as they are the complex conjugate of some other (computed) signal.

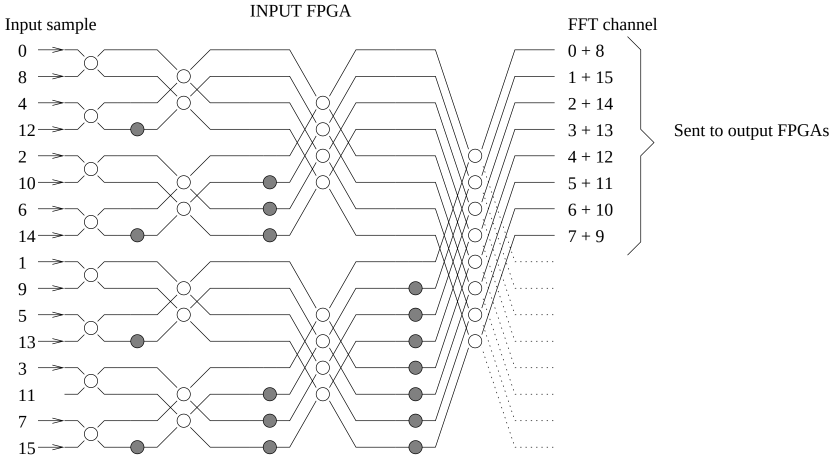

Figure 5: Structure of the FFT algorithm. Butterfly stages are divide between 4 input FPGAS (one shown) and 4 output FPGAs (next figure)

The corresponding outputs from each input FPGA are combined together in a output FPGA, using a 4 × 4 butterfly stage (fig. 6a). The four outputs of this stage represent 4 output channels of a 128-point FFT. The complex conjugate relationship among positive and negative frequencies have been used to delete the unnecessary butterflies. For example the structure in fig. 6a computes channels 1, 17, 33 and 49, but channels 33 and 49 (negative frequencies in a 64 point FFT) are interpreted as the complex conjugates of channels 31 and 15.

In the particular case of channel 0 from the input FPGA, the outputs for channels 8 of the first FFT

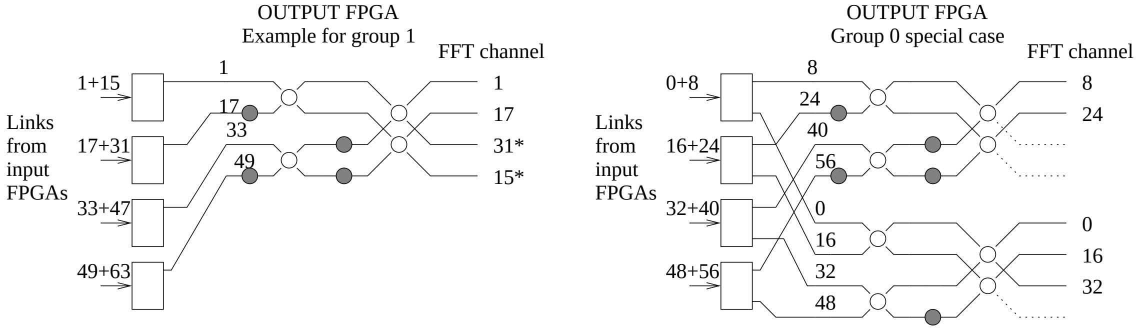

Figure 6: Structure of the FFT algorithm. Output FPGA (last 2 stages). A total of 8 groups are needed for the complete FFT. a) generic block (example for group 1). b) Dedicated block for group 0

are combined as usual, and those for channel 0 are combined in a second dedicated butterfly (fig. 6b). This latter does not require multipliers, as all twiddle factors are either 1 or i. The complete structure produces 65 output channels, 2 real with a band of 128 MHz, and 63 complex, with a band of 256 MHz. Odd and even channels overlap, to completely cover the input bandwidth. The position of the FFT channels in the input bandwidth is shown in fig. 8.

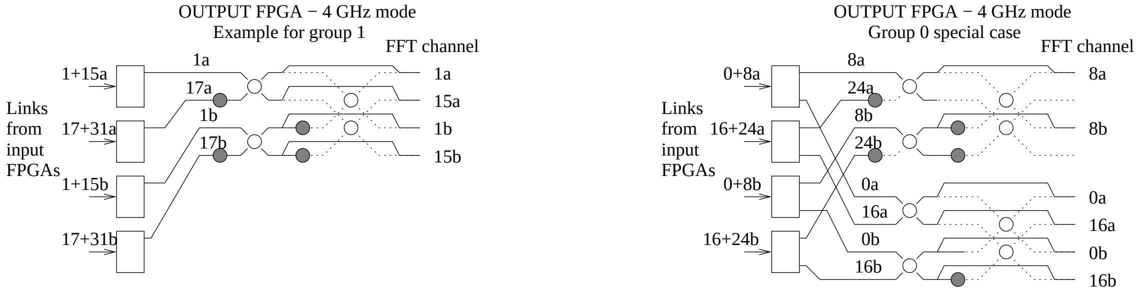

Figure 7: Output stage for 32 channels FFT. a) generic block, b) special case for channel group 0.

If the input signal is sampled at a lower data rate, it is possible to use a slightly modified butterfly stage (fig. 7) to analyze two signals at 4 GHz bandwidth. For 4 signals sampled at 2 GHz bandwidth all the required processing has already been performed in the BN FPGA.

Each input FPGA produces 8 complex outputs. The corresponding outputs of the four input FPGAs are combined together in a FN FPGA, and produce four FFT channels. These channels do not depend on the other FPGA outputs, and can be considered forming a group. There are thus 8 groups, that are numbered from 0 to 7, from the first numbered channel in the group. Group 0 contains all the channels centered at a multiple of 1 GHz. Groups $k = 1 \dots 7$ contain channels k, $16 - k$, $16 + k$ and $32 - k$ (see figure 8).

The available data bandwidth in the interconnecting mesh between BN and FN FPGAs allows two groups to be processed in each FN FPGA. Therefore the 16 DBBC implemented in each FPGA can process only portions of the input bandwidth belonging to two channel groups. To simplify routing, these groups are always a consecutive pair. This poses constrains in the position of the output DBBC channels, as a DBBC channel must be placed completely inside a FFT channel.

Summarizing, each one of the 16 DBBC in a FN FPGA can extract a portion of maximum 128 MHz from a set of 8 FFT channels, belonging to two consecutive FFT groups (fig. 9). Each FN FPGA can process one of these four sets, chosen arbitrarily. Each FFT channel is 256 MHz wide (nominal), with about 200 MHz effectively usable, and with a nominal overlap of 50% between adjacent FFT channels.

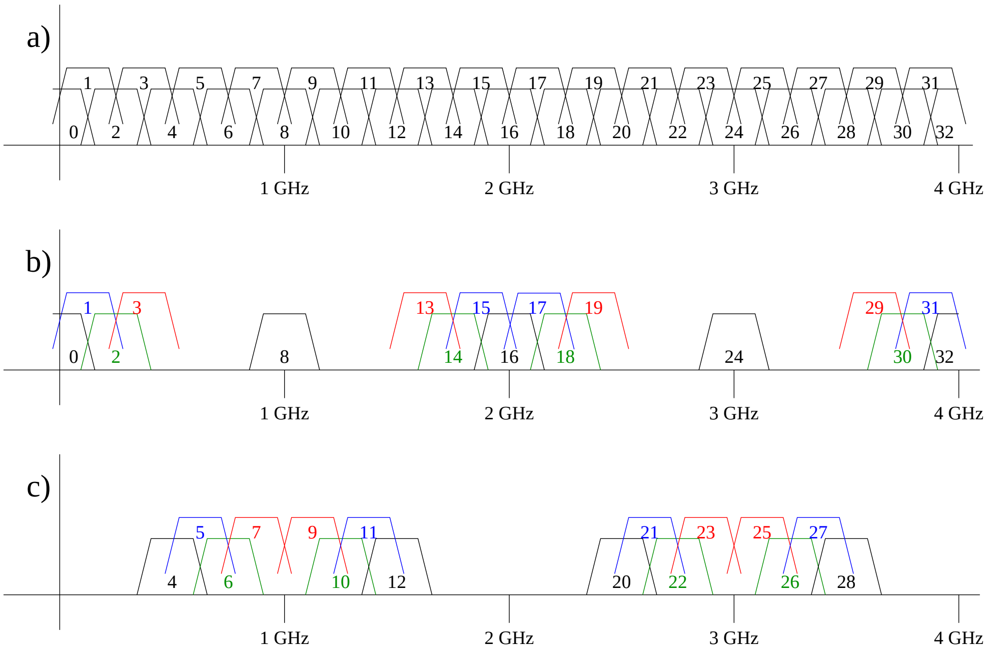

Figure 8: Position of the FFT channels in the input bandwidth. (a) All channels; (b) Channel groups 0 (black), 1 (blue), 2 (green), 3 (red); (c) Channel groups 4 (black), 5 (blue), 6 (green), 7 (red)

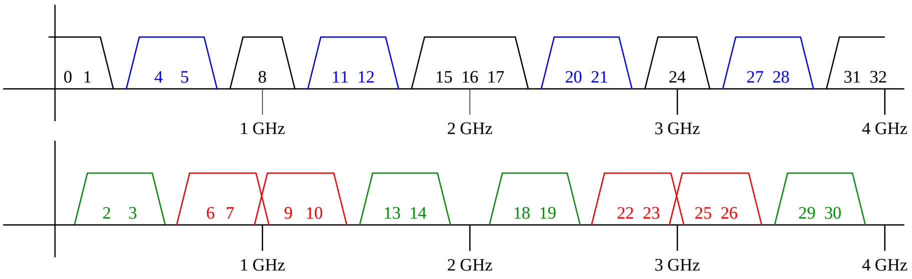

Figure 9: Position of the four FFT channel group pairs Channel groups 0-1 (black), 4-5 (blue), 2-3 (green), 6-7 (red). Each FN FPGA can analyze the portion of frequencies covered by one of these groups

3.2 Digital Baseband Converter and formatter

The Digital Baseband Converter selects a portion of one of the 8 available FFT channels, and further filters it to a programmable bandwidth between 1 and 128 MHz. The center frequency of the filtered band can be programmed within the FFT channel.

The filter design adopts tap multipliers recirculation, to obtain a passband shape that scales with the bandwidth, with an usable bandwidth around 90% of the nominal (Nyquist) one. The output signal is then converted to real, and decimated to the appropriate Nyquist rate.

Each DBBC has a dedicated VDIF formatter, that packetizes blocks of consecutive samples together with timing and ancillary informations. Block length is limited by the maximum IP packet size. We adopted the standard limitation for (nonstandard) Jumbo IP packets, i.e. 9 kbytes. For memory usage considerations the actual limit is 8192 bytes, minus the IP, UDP and VDIF overheads. A further limitation is due to the restriction that VDIF packets must evenly divide one second. For design simplicity we also assume that the VDIF packet is composed by 32 bit words, and thus typical packet size is either 5000 or 8000 bytes.

VDIF packets are sent encapsulated into UDP packets. Each packet may have a different length and destination port/IP address. Since the decimation can be different, it is natural to have different packet rates. Packets are then buffered, using a dual memory scheme, and are queued for sending when ready. A simple scheduler picks the next ready packet and sends it to a 10Gb IP interface. Each interface collects the packets generated by 8 DBBCs.

3.3 Test signal generator

A test signal generator is a very useful complement of any digital signal processing system. It allows to test the system without a real signal being available, both during system setup and during actual operations.

For a radioastronomy instrument, the test signal should include

- A truly Gaussian white noise, with a RMS noise spectral density N, and a statistics good enough to perform deep integrations (at least several minutes) with a measurement noise σ_N in accordance to the radiometer equation: $\sigma_N = N/\sqrt{\tau B}$
- A few (2 are usually enough) spectral sinusoidal tones, with amplitude well below the noise total power
- a comb of calibration tones with predictable phase relationship

The generator must produce 32 parallel samples representing 32 consecutive inputs form the simulated ADC. It is physically split into 4 components, one for each of the 4 input FPGAs. These components must be properly synchronized, using the 1pps and 1ms sync pulses.

4 Input FPGA (Back node)

The block diagram of one input FPGA is shown in fig. 10. Not shown are several components common to all Uniboard FPGAs (configuration FLASH PROM, temperature sensor, control LEDs, etc.).

Most of the system is instantiated as a Nios system on a chip (SOPC). The system is built using the SOPCbuilder Altera tool, using components *encapsulated* provided by Altera, by the project, or developed ad hoc. Several components are developed by the project: LVDS receiver, interconnecting mesh interface, Ethernet, and all the common components not shown in the figure. The *1st stage* custom module and the test signal generator have been developed for this application.

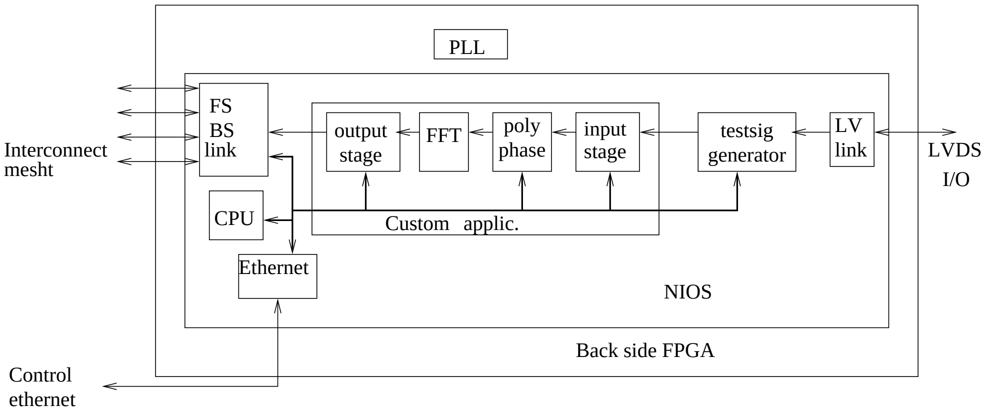

Figure 10: Structure of the input FPGA

The structure of the first part of the polyphase filter is shown in more detail in fig. 11.

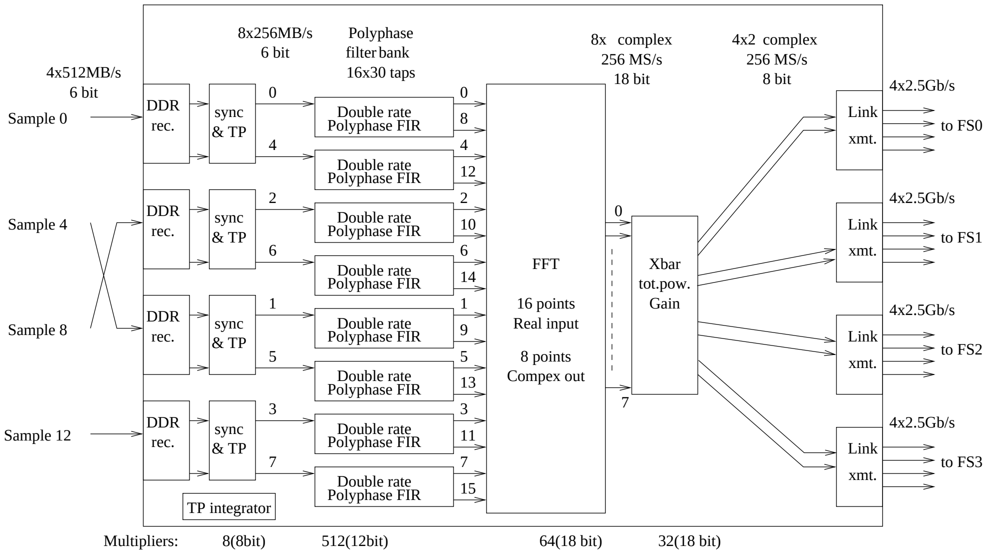

Figure 11: Structure of the input FPGA component

4.1 Test signal generator

The test signal is composed of a Gaussian white noise, two monochromatic lines of arbitrary amplitude and frequency, and a comb pattern. Each of these components is produced by a separate generator, multiplied by a gain factor and summed together and to the ADC input. The result is then re-quantized to 6 bit, and sent to the Digital Receiver component.

The circuit is effectively disabled if all gains are set to zero, and only the input values are sent to the Digital Receiver. This is the default configuration at power-up (all internal registers set to 0).

The generator is distributed among the four BN FPGAs, Each generator component must be programmed in a consistent way to produce a meaningful signal. Phase coherence is guaranteed by synchronizing all signals to the board wide 1PPS signal.

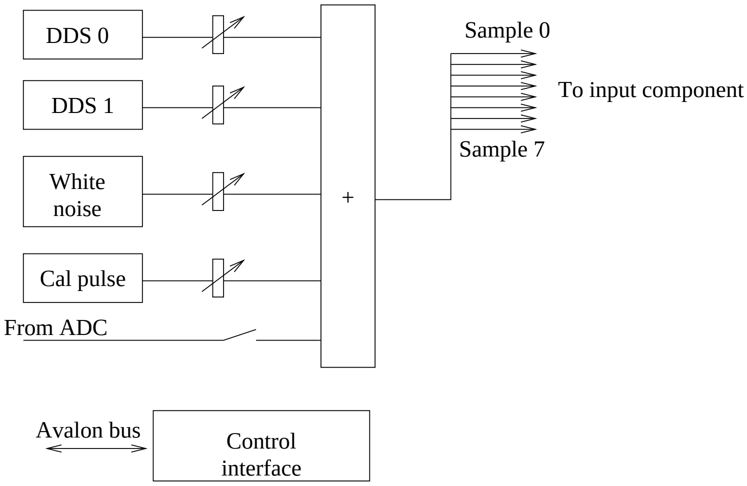

Figure 12: Test signal generator

4.1.1 White noise

It is not easy to generate a truly random number. We used the 64 bit xor-shift method described in the Numerical Recipes book[7], that is equivalent to a series of independent LFSR. Having 64 bits per clock cycle allows to obtain 8 uniformly distributed 8-bit quantities per generator, one for each of the 8 inputs in each FPGA.

A sufficient approximation to a Gaussian statistics can be the sum of N uniform generators, by the central limit theorem. A value of $N = 8$ produces a signal Gaussian enough for test purposes.

4.1.2 Monochromatic tone

Two monochromatic tones are produced using two independent direct digital synthesizer. An appropriate phase offset for each generator is applied automatically, in order to generate a 8GS/s interleaved sinusoid.

4.1.3 Phase calibration tone

A phase calibration tone is a comb of sinusoidal tones with a known phase. Phase calibration tones are produced in parallel at equispaced frequencies, usually multiple of the tone spacing $\Delta\nu_t$. This spacing is usually chosen in order to have a few tones in the channel bandwidth. Having just one tone allows an easy measurement of its phase, from the raw data, while having at least two allows for accurate measurement of the system group delay, but requires dedicated hardware/software for tone extraction.

The best way to generate such a tone is to produce a comb of pulses with a repetition rate Δt equal to $1/\Delta\nu_t$. The tone must be present in just one of the 32 samples.

4.2 Input from ADC

The input samples are provided on a backplane parallel interface. In this version of the design interface is composed of 16 identical paths (4 for each FPGA), with a 8 bit differential data, a differential clock, and an auxiliary low speed serial data link. To accommodate for 8 GS/s, each link must be operated at 512 MS/s clock rate (DDR at 256 MHz clock). Internally each link is converted to two data streams at 256 MHz clock, for a total of 8 samples. The DDR receiver/demultiplexer uses a standard library component from Altera.

After DDR demultiplexing, the sampled signal is represented as a total of 32 data streams, 8 for each FPGA. The convention adopted is that stream number indicates the sequence number in a group of 32 consecutive samples, with 0 the older and 32 the most recently sampled.

Samples are presented to the board in bit reversal order. This bit reversal is a natural consequence of the DDR demultiplexing and of the structure of the double rate polyphase algorithm, as shown in fig. 4.

As noted in chapter 2.1, the input format can be different, i.e. it may use serial CX4 ports. In this case, the input module would be different, but the sample coding and order would be the same, and after in-board demultiplexing samples are always presented to the digital receiver module as a total of 32 samples. Everything after this point should be independent from the sample transport layer.

Input streams must be synchronized to allow for different propagation paths in the electronics. Synchronization is achieved by substituting sampler symbols with a repetitive data-pattern, and measuring the delay with respect to system wide timing signals. In this way it is not essential to synchronize the FPGAs in the board to the picosecond level required by the signal speed, but each FPGA can process data within a relatively wide time window. Once measured, the synchronization delay for each sample input port (either LVDS or CX4) is applied in a short FIFO.

Several parameters of the input signal can be measured. The measurement is performed on each of the 8 input data paths in each FPGA, and then summed together and integrated for a programmable integration time. These values must be further summed together by the external control software.

Parameters to be measured include:

- Total power: the square of the input samples
- DC offset: the average of the input samples

- State counter: the number of occurrences of a particular (programmable) sample value. By performing a state count on all possible input values, a histogram of the input signal can be computed.
- RD check: the bit sequence on a selectable signal bit is compared to a pseudo-random signal generated by a LFSR, to check data integrity. It is assumed that for this test the ADC can generate the same pseudo-random signal.

The total power integrator uses a common structure, reused for all the similar total power meters in the design.

4.3 Polyphase filter

The polyphase filter section of the FFT is implemented in a distributed way across the 4 BN FPGAs. The filter tap coefficients depend on the FPGA position, and on the total input bandwidth. They are stored in small ROMs and selected by the control software.

Each data stream is processed in two short FIR filters (fig. 13), that produce at each clock cycle two filtered samples, with indices i and $i + 32$. As the data rate is doubled with respect to a conventional polyphase FFT, on odd clock cycles a phase slope is introduced in the data by the filter. To remove this, on odd cycles samples i and $i + 32$, that are produced by the same filter branch, are exchanged.

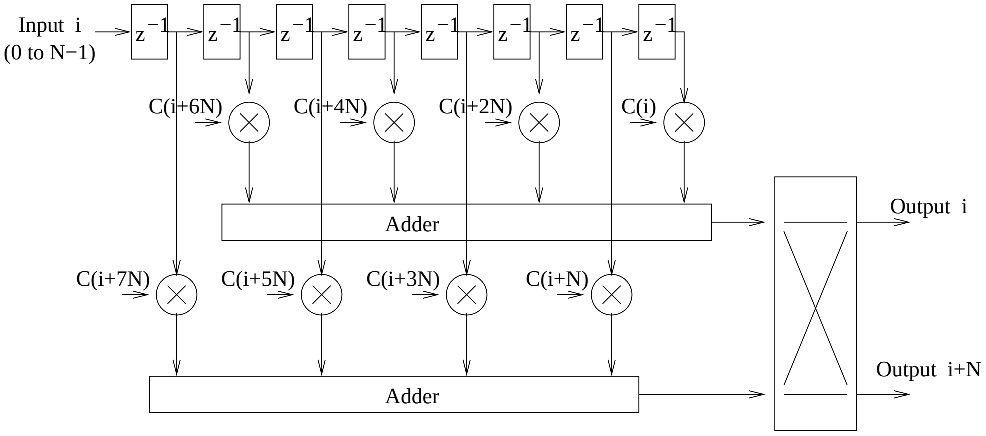

Figure 13: Double rate polyphase filter branch. On odd cycles the two outputs are exchanged, to remove phase slope

The polyphase length is limited by the available resources in the FPGA, but giving the small resource utilization in the other parts of the circuit a total of 512 taps per FPGA can be safely allocated, for a total filter length of 2048 taps. This allows a very sharp transition region at the channel edges, and a very good out-of-band rejection. A filter with usable band of 80%, and more than 90 dB of stop-band attenuation has been designed, and used in the simulations. This means that the central 200 MHz of each FFT channel can be used, with an overlap of 56 MHz between adjacent channels.

Altera Stratix4 offer 9, 12 or 18 bit multiplier size, but 12 bit tap coefficients provide roughly 60 dB stopband rejection. To achieve better than 80 dB rejection, 18 bit multipliers have been used.

Multipliers have fixed coefficients, but coefficients depend on the FPGA chip position and input bandwidth. It is convenient to have a single personality for all identical (FN or BN) FPGAs, and thus coefficients are stored in small LUT-based ROMS, with the appropriate coefficient selected using an address corresponding to the chip position.

It is relatively simple to code the VHDL description in order to generate these ROMs from a linear tap coefficient file, so changing filter response is relatively simple.

The samples computed by the polyphase filter are 24 bit wide, so a rescaling is necessary. Considering up to 5 bit growth (when all the input signal is contained in a single FFT channel) input to the FFT must not exceed 13 bits. The polyphase output is thus rescaled by a fixed factor of 2^{-11}.

4.4 FFT block (first 4 stages)

The 16 real outputs from the polyphase block are processed inside a 16 channel FFT processor, with real inputs and complex outputs. The processor is structured as two conventional 8 channel decimation-in-time FFTs, with complex outputs, and a butterfly stage in which only the first output channels are retained. The other channels can be reconstructed considering that channel $16 - k$ is the complex conjugate of channel k. Channel 0 and 8 are real, and are combined together as the real and imaginary part of a single *pseudo complex* channel 0.

Input from the polyphase filter block is 13 bit, that grows to 15 bits in the first two (multiplierless) FFT stages. Successive FFT stages use conventional 18 bit hard multipliers. Twiddle coefficients are identical for all chips. Output of each multiplier is rescaled to 18 bits, with rounding, and FFT output is 18 bit, complex.

A total of 64 18-bit multipliers are used in this stage.

4.5 Output stage

The output stage computes the total power in each FFT output. Each output can be scaled, in order to optimally use the available link bandwidth. Typically the best quantization efficiency occurs for a RMS value around 1/10 of the total span, i.e. 3 bits must be reserved for accommodate the Gaussian tails of the noise statistic.

Total power is computed for each frequency channel, is integrated for a programmable integer number of milliseconds, and can be read back by the embedded Nios computer. The control program must measure the total power outputs in all chips, and program the scale factor to the same value for all corresponding FFT outputs.

A crossbar switch allows up to two channel groups to be sent to any FPGA. The selection is arbitrary, e.g. all the FPGAs can process the same two channel groups. Both the gain and the signal selection must be the same for all input FPGAs, as they represent consecutive samples of the same FFT bands.

The total number of hard multipliers required in each stage is shown in the bottom line of figure 11. Most of them (512) are used in the polyphase filter, with 64 used in the FFT and 32 in the output stages.

Each BN FPGA is connected to each FN FPGA by four fast signals, implementing a x4 fast serial link. Three of the four signals are bounded to a hard transceiver, with several functionalities available in hardware. The increased speed of these transceivers make it advantageous to use just these three links, instead of all 4. The link is implemented in hardware in the Stratix4 FPGAs, and is instantiated using the `altGX` Quartus macro. The module `mms_tr_nonbonded`, developed by the Uniboard project, instantiates a transparent link, with a number of equivalent parallel connections depending on the link speed.

Each link must transport a total of 4 real samples, at 256 MS/s. The number of bits in each sample determines the overall SFDR. As shown in [2], a minimum of 8 bits/sample must be used, for 70 dB SFDR, with each extra bit providing an improvement of 6 dB. The link uses 8bit/10bit encoding, for better data integrity, so the minimum link speed is 256x4x10/3 MHz = 3.41 GHz.

5 Output FPGA (Front node)

The block diagram of one front node FPGA is shown in fig. 14.

Each FPGA receives samples from the same two FFT channel groups, with four different phases, from the 4 input FPGAs. Each signal is composed of complex samples at 256 MS/s, representing the output of the previous stages of the 64 channel FFT. For each group the four signals represent FFT outputs at index $j + 16k$, with j and k the index of the channel group and of the input FPGA, respectively.

The complex conjugate of each input sample represents the output at index $16 - j + 16k$, apart for channel group 0, where the real and imaginary parts correspond to channels $16k$ and $16k + 8$ respectively.

From these samples, the FPGA compute the final part of the FFT, obtaining up to 8 *FFT channels*, each one representing a 256 MHz wide portion of the input signal as a 256 MS/s complex signal.

From these FFT channels up to 16 real data streams are computed, with independently selectable bandwidth and position. Bandwidth can be chosen from 1 to 128 MHz (2 to 256 MS/s), with a position

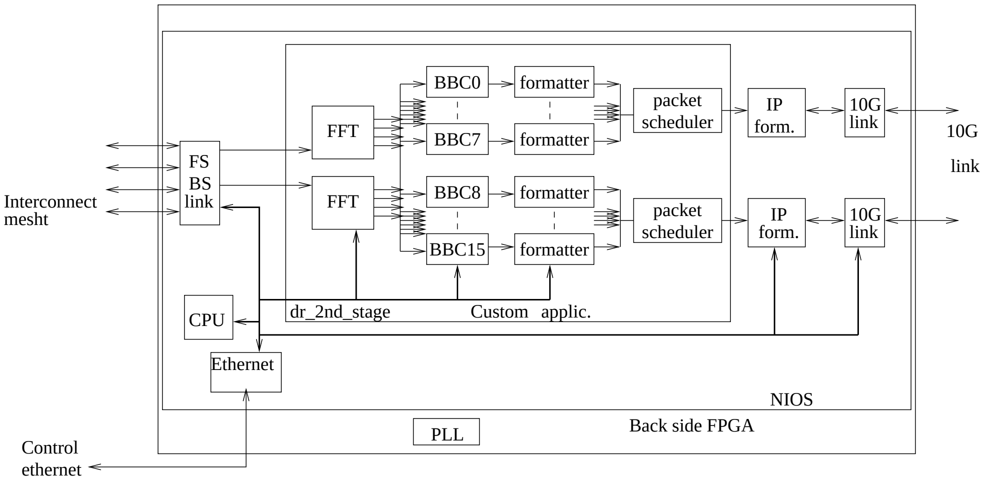

Figure 14: Structure of the output FPGA

resolution of 0.01 MHz[4]

5.1 Input section

As in the input FPGA, the first block after the link receiver is used to correctly align samples from different links and to measure the sample synchronization.

No total power measurement is needed, as this is already measured in the input FPGA for correct re-quantization (see chapter 4.5). Signal integrity can be measured after the FFT butterfly, configured in bypass (2 GHz) mode, using pseudo random sequences.

5.2 FFT block

Samples from each link are sent to a 4 × 4 FFT butterfly stage. Two stages are present, in order to process channels for up to 2 different FFT blocks. Each block computes 4 complex outputs, (see fig. 6), but in case of channel block 0 the output 0 is composed of two independent real signals.

This stage must be highly configurable, to support a wide range of possible configurations. The main configuration parameter is the channel group number, that must be compatible with the output selected in the input FPGA.

FFT twiddle coefficients depend on the channel group number, with 16 possible values for each twiddle multiplier. For group 0 the real and imaginary parts of the signal correspond to independent FFT channels, and are sent to two separate 4 × 4 butterflies. The imaginary part is sent to the *normal* butterfly, producing FFT channels 8 and 24, while the real part is processed in a dedicated, multiplierless, butterfly that computes outputs 16, 0 and 32. Channels 0 and 32 are then combined in a single pseudo-complex signal (see fig. 6).

The block can be configured to analyze four 2GS/s or two 4GS/s signals. For 2GS/s operations the block is completely bypassed. For 4 GS/s, the first stage of the butterfly operates normally, while the second is bypassed (see fig. 7).

Butterfly stage output is complex, with 18 bit integer representation. An output stage similar to the one in the input FPGA allows each individual channel to be rescaled and its total power measured. The samples are then re-quantized to 8 bits.

The FFT block requires 32 multipliers, with other 32 multipliers for total power and first rescaling stages. This requires 16 of the available 161 DSP blocks in the FPGA.

[4] Actual frequency resolution is much higher, 8192 MHz/2^{32}, i.e. 1.9 Hz, but the mechanism used to maintain phase coherency of the LO, described in chapter 5.3 works only for a local oscillator frequency multiple of 10 kHz.

5.3 DBBC blocks

The FFT stage is followed by an array of digital BBCs. A minimum of 16 DBBC is required to split the 128 MHz of the non overlapped portion of each FFT channel into 64 MHz DBBC channels. Each DBBC is composed of a complex LO/mixer, a complex low-pass filter with variable decimation, and a complex-to-real conversion stage. The low pass filter has a cutoff frequency of half the output bandwidth. The filter output is multiplied by $\exp(2\pi i t/4)$, where t is the index of the output clock, and the real part only is retained. Changing sign of the exponential reverses the frequency scale (USB or LSB).

Each DBBC can select its input among the 8 possible outputs of the FFT stage. It is possible to select as input the whole complex signal, or its real or imaginary component, interpreted as a real value. This is useful to select the first and last FFT channels combined together in the FFT channel 0. The complex LO is used to select the central frequency of the BBC channel. It can be programmed with high resolution (8 GHz / 2^{32}). If the frequency is programmed as close as possible to a multiple of 10 kHz, and its phase is reset using an internally generated 10 kHz reference, the LO emulates a frequency step exactly multiple of 10 kHz (standard used in many VLBI observations).

The filter uses a variable decimation scheme, where the total number of multipliers are fixed and operate always at the maximum speed. The filter length is thus proportional to the decimation factor, and the filter shape remains roughly constant (same fraction of usable output bandwidth) and does not degrade with the decimation factor.

The complex-to-real conversion requires alternately only the real or the imaginary part of the filtered signal. Each branch (real or imaginary) of the filter thus compute only alternate samples, with a decimation factor D equal to the ratio of the input (complex) to output (real) bandwidths. A minimum decimation of 2 is used (128 MS/s, 64 MHz bandwidth for each filter, 256MS/s, 128 MHz for the real output), where each multiplier is recycled by a factor of 2. Exploiting filter symmetry, filter length is thus $2D$ times the number of available multipliers.

Considering the number of available DSP blocks, up to 48 multipliers are available for each DBBC complex filter, i.e. up to 24 each of the real and imaginary parts. Maximum filter length for bandwidth decimation D is thus $96D$ taps, with 18 bit resolutions. A simpler filter, with length $64D$, has been initially used. Filter length determines the maximum stopband rejection and the usable portion of the Nyquist band. For a stopband rejection of 80 dB, the usable bandwidth is respectively 93% and 90% for the two filter lengths, with a passband ripple of 0.03 dB peak-peak.

A last total power meter and rescaling stage is used to adjust the signal level, and the output is requantized to 8 bits. The output level spacing is chosen with a threshold across zero, and equally spaced levels. This means that code i represents an interval of possible values centered at $i + 1/2$, and allows to re-quantize the signal by just discarding the least significant bits of the sample code.

5.4 Output formatter

The signal from the DBBC is formatted in a VLBI packet, encapsulated in a UDP *jumbo packet*, and sent to the correlator using one of the output links.

The output link bandwidth is limited by the actual physical connection between the antenna and the correlator, and is usually much less than the Uniboard output throughput. For example using 16 output channels with 4 bit representation and 64 MHz per channel (128 MSample/s), a total of 8 Gb/s is required. Considering packetization and overhead this data rate is close to saturate a single 10Gb link. To avoid bottlenecks two links are used, one for each group of 8 BBCs. Thus the Uniboard connects with 8 10Gb links, and an external (commercial) router may be required for merging all links to a single physical link.

Each BBC has its dedicated packetizer. The data from the BBC is quantized to 1, 2, 4 or 8 bits per sample, samples are grouped into 32 bit words, and VDIF header is added at the beginning of each packet. The VDIF header contains informations about the absolute time, the quantization, an identifier for the station and the data thread, the frame length, and up to 4 user defined parameters.

Each VDIF packet is transmitted as a single UDP packet. A UDP pseudo-header (UDP header plus destination IP and port informations) is added before the VDIF header. The UDP packet is stored in a dual buffer memory, and the associated UDP checksum is computed in the process.

All VDIF and UDP parameters can be specified independently on a per-BBC basis. In particular it is possible to specify different frame length (up to 8k bytes), quantization, destination address, source and destination UDP ports. The source IP address is hardware related, so each physical output link must have a unique address.

When a packet is complete, it is scheduled for transmission on the output link. A scheduler sends all the scheduled packets using a simple round-robin [5] algorithm to the Ethernet interface (ARP-UDP packetizer). This component is derived from the IP module developed by the system team. It implements the ARP protocol, enquiring and collecting the mapping informations between the MAC and IP addresses, the PING protocol, and assemblies the IPv4 header for the UDP packets received by the scheduler.

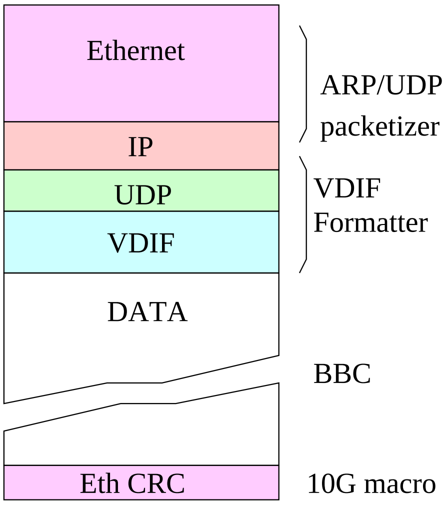

Figure 15: Ethernet packet structure. The raw data samples are encapsulated in a VDIF packet, a UDP header and checksum is then added, and the resulting frames are sent as IPv4 compliant jumbo packets

6 Programming interface

The detailed programming interface for the Digital Receiver is described in [4]. The assignment shown here is not definitive, but is shown to provide more details on the available system functionalities.

6.1 Test signal generator

The address space for the generator is shown in table 1. It occupies a total of 64 bytes organized as 16 32-bit registers.

The test point select register is actually not used. It has been kept as it provides an useful ”do nothing” register, that can be written and read back without affecting anything in the circuit.

The control register performs various control functions that require just one or a few bits. Its bit usage is shown in table 2.

Bit 0 switches off the ADC input. When it is set, the generator output does not depend on the ADC samples. In normal operation it is set to 0 and all amplitude values (register 2) are also set to 0, disabling the generator. Bits 2-3 select the FPGA. It affects the DDS phase and the PRDG seed. Bit 4 loads the frequency register of the DDS on the next 1ms pulse. It is automatically de-asserted when the register has been loaded.

[5] The round-robin algorithm s a commonly used algorithms for several clients needing a shared resource at random intervals. Clients are scanned circularly, and if one is found ready it is serviced just once. It is relatively simple to implement and reasonably fair, i.e. a single channel cannot monopolize the service.

Address	Write	Read
0x00	Test point select	Test point read-back
0x04	Global control	Global status
0x08	Amplitude gains	=
0x10	DDS 0 frequency	=
0x14	DDS 0 phase	=
0x18	DDS 1 frequency	=
0x1c	DDS 1 phase	=
0x20-3c	PRDG 0-7 seed	=

Table 1: Register mapping for Uniboard Digital Receiver test signal generator

bit	control	status
0x00	ADC off switch	=
0x02-03	Chip position	=
0x04	Load DDS	=
0x05	Reset DDS phase at each ms pulse	=
0x06	Reset PRDG at each ms pulse	=
0x10	Enable cal tone	=
0x11-13	Cal tone sample	=
0x14-16	Cal tone period	=

Table 2: Register mapping for general control register. Status bits specified as "=" are a copy of the corresponding control bits

Bit 5 resets the DDS phase register at each 1ms pulse. If the DDS frequency is set as close as possible to a multiple of 10 kHz, the resulting tone remains phase coherent to a true multiple of 10 kHz, with only a minor glitch every ms.

Bit 6 resets the PRDG every ms. In this way the signal processed by the system repeats itself every ms, simplifying debugging.

Bit `0x10` enables the phase calibration tone generator. A single sample of amplitude `0x200` is inserted in the sample selected by bits `0x11-13` with a period selected by bits `0x14-16`.

Bits `0x14-17` select the phase cal pulse repetition rate. Minimum repetition rate is 2 clocks (128 MHz), for binary value 000, up to 256 clocks (1 MHz) for binary value 111. The pulse is synchronized to the ms pulse. Register 2 specifies the gain in 4 bytes. Each byte is interpreted as a 8 bit unsigned value. From the LS byte, the values control the gain for the two DDS, for the white noise generator, and for the pulse generator.

Registers 4 to 7 specify the frequency and phase for the 2 DDS. Phase is specified in terms of turns divided by 2^{32}, and frequency in the same units per elementary sample (8192 MHz).

Registers 8 to 15 specify the initial seed for the eight PRDG. Only 32 least significant bits are specified, the 32 most significant bits are computed from this value XORed with 16 copies of the FPGA position.

6.2 First stage component

The first stage (BN) component is seen on the Avalon memory mapped bus as an address space of 128 bytes, organized as 32 32-bit registers. Not all bits of each register are used, or read back. A summary of the registers usage is shown in table 3.

Bit assignment for most registers is tentative, and are used to specify the general control capabilities available.

Register 0 is used to select the signals that are connected to the 8 `testio` lines. The 32 bits are grouped in 8 4-bit groups, and each group select one of 8 lines. Lines 0 and 1 are possible inputs, their use as output test points is TBD. Lines 2 and 3 drive two LEDs, so they must be used for pulsed signals

Register	Address	Write	Read
0x00	0x00	Test point select	Test point read-back
0x01	0x04	Global control	Global status
0x02	0x08	Interrupt control	Interrupt status
0x04-07	0x0c-1f	unused	unused
0x08	0x20	Input total power control	Input total power
0x09	0x24	FFT total power control	unused
0x0a-0b	0x28-2f	unused	unused
0x0c-0f	0x30-3f	Input link control	Input link status
0x10-18	0x40-63	FFT gain	FFT total power
0x19-1b	0x64-6f	unused	unused
0x1c-1f	0x70-7f	Output link control 0-3	Output link status 0-3

Table 3: Register mapping for Uniboard Digital Receiver input chip

(with pulse stretcher), or slow status signals. Lines 4-7 are connected to test posts for oscilloscope probes. The particular mapping of the lines connected to each test point is still to be determined.

This register can be read back, as part of a simple integrity test.

Register 1 controls the general behavior of the board. Bits are assigned as in table 4.

bit	control	status
0x00	Low power mode	=
0x02-03	Chip position	=
0x10	Input TP ready reset	Input TP ready
0x11	FFT TP ready reset	FFT TP ready
0x12	FFT overflow reset	FFT overflow
0x18-1e	PLL phase adj	PLL status
0x1f	PLL reset	PLL unlock

Table 4: Register mapping for general control register. Status bits specified as "=" are a copy of the corresponding control bits

Register 2 is used for interrupt control, if interrupt will be used by the module in future versions, and is unused in this version. Interrupt can be useful for total power reading, for the serial line from/to the ADC, and for error conditions.

Register 8 (table 5) controls the integration time and general functionality of the input total power meter. Total power is read from the same register.

bit	control
0x00-01	TP function: 0 = Total power, 1 = DC offset 2 = State counter, 3 = RD check
0x02	General enable
0x08-0f	Reference status or RD check line
0x10-1c	Integration length
0x1d-1f	Integration prescaler

Table 5: Total power control register

Integration time is expressed in 1ms intervals The register is used to specify the quantity to be monitored, that can be selected between the following:

- Total power: the square of the input signal

- DC offset: the average of the input signal
- Status: counts the number of samples identical to a specified value, specified as a 8 bit value in the control register. This is useful to build an histogram of the ADC sampled data.
- Random data check: counts the number of errors in one of the 8 input lines, specified as a 3 bit value. Is used to check electric integrity over the input lines

The integration is performed on a sum of these values over the 8 parallel input samples. Each sample can be individually enabled/disabled using the input link control register. For most cases, all inputs are used, but for debug a single line (random data check) or a single data stream can be examined.

Integration result must be rescaled to compensate for different integration time. The number of bits discarded range from zero to 14, in steps of 2. The total power reading is a 31 bit quantity, with the most significant bit used to signal an overflow.

Register 9 controls the integration time and number of bits discarded in the FFT total power detector. The register is similar to the input total power detector, but only total power and DC offset functions are available. The corresponding read register is unused as the total power value is read for each individual FFT channel.

Registers `0x0c` to `0x0f` control each of the 4 input blocks. This register is used to specify setting of the alignment FIFO, to read the synchronization detector, and to enable/disable each individual input channel for total power metering.

Registers `0x10` to `0x18` are used to read the signal level at the 9 FFT outputs, and to adjust the gain of each output channel. Channels 0 and 8 are combined together, so the quantization scheme for channel 0 is used also for channel 8, but they have independent total power meters and usually have different rescaling factors. The total power reading is a 31 bit quantity, with the most significant bit used to signal an overflow.

Registers `0x1c` to `0x1f` are used to select the signals sent to each of the front side FPGAs. Each register controls the 4 links directed to a specific FPGA, selecting the quantization scheme used (4 or 8 bits) and the two FFT channels sent over the 4 links. It is possible to substitute the FFT output samples with a 32 bit internally generated pseudo-random sequence, or with a synchronization pattern.

The high speed link interface is implemented using a library component, and is programmed using a dedicated interface.

6.3 Second stage component

The addressing space of the output section is 256 bytes long, organized as 64 32 bit registers. The register address mapping is shown in table 6. The first 8 positions (only 3 used) are used for general control. The next 8 are used for the formatter and output sections. Next 16 registers control the input and FFT blocks, and the last 32 control each of the 16 DBBCs.

Registers 0, 1 and 2 are similar to corresponding registers in the input FPGA. FFT and BBC total power registers (`0x10, 0x11` specify the integration time, the number of bits discarded in the result, and the function integrated.

The FFT total power can be used to check the statistics and the signal integrity for the input lines, by placing the FFT stage in the *2 GS/s FFT mode*.

FFT control register allows to specify FFT mode, and output gain. FFT modes are listed in table 7.

First 16 modes are the normal 64 channel FFT modes. If the input signals are for group k, output channels can be either $k, k+16$ (first half), or $32-k, 16-k$ (second half).

Modes `0x10-1f` are used for a 4 GHz input bandwidth (32 output channels). In this case both outputs correspond to channel k (first half) and $32-k$ (second half) for the two ADCs connected to the top and bottom input FPGAs. In this mode, the first butterfly is not modified, and the second one is bypassed, with the two outputs corresponding to the top outputs of the first butterflies.

Modes `0x20, 21` are used for 2 GHz ADCs. In this mode, each input FPGA completely process the signal sampled by a single ADC, and the butterfly simply copies its inputs (either the upper or the lower 2) to its outputs.

DBBC control register specifies the band, gain, and other functions for each one of the 16 DBBCs. Bit mapping for this register is shown in fig. 8. Bits $0-2$ select the decimation, and thus the band.

register	address	write	read
00	0x00	Test point select	Test point read-back
01	0x04	General control	General status
02	0x08	Interrupt register	Interrupt status
03-07	0x0c-1f	unused	unused
08-0f	0x20-3f	Output formatter	Formatter status
10	0x40	FFT Total power ctl	FFT Total power status
11	0x44	BBC Total power ctl	BBC Total power status
12-13	0x48-4f	unused	unused
14-17	0x50-5f	Input 0-3 FIFO ctl	Input 0-3 FIFO status
18-1f	0x60-7f	FFT 0-7 control	FFT 0-7 total power
20	0x80	BBC 0 ctl	BBC 0 status
21	0x84	BBC 0 frequency	BBC 0 total power
. . .	. . .	. . .	
3e	0xf8	BBC 15 ctl	BBC 15 status
3f	0xfc	BBC 15 frequency	BBC 15 total power

Table 6: Register mapping for Uniboard Digital Receiver output chip

Mode	Function
00-07	64 channel FFT, channel group 0-7, first half
08-0f	64 channel FFT, channel group 0-7, second half
10-17	32 channel FFT, channel group 0-7, first half
10-1f	32 channel FFT, channel group 0-7, second half
20	16 channel FFT, inputs 0 & 2
21	16 channel FFT, inputs 1 & 3
30-3f	disabled

Table 7: FFT modes

Bits $3-5$ set the BBC input selection. If bit 6 is set, the output is substituted with a pseudo-random sequence. Bit 7 drives the reset signal, and completely disables the device (useful also to reduce power).

Bits $8-10$ introduce a phase offset, in 45 degrees steps, in the local oscillator.

The two following bits are used to check the overflow and ready states of the total power, that are latched until explicitly reset. Bits $16-24$ control the output gain.

Bit	Control	Status
2-0	Band select 0 = 1/2 7 = 1/256	=
5-3	Input selection	=
6	Test pattern generation	=
7	Reset	=
10-8	Phase offset	=
11	TP OVF reset	TP OVF
12	TP ready reset	TP Ready
24-16	Output gain	=

Table 8: Register mapping for the BBC Control/Status registers