Uniboard Digital Receiver
Initial design document
Revision 2.0

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

[1]INAF - Osservatorio Astrofisico di Arcetri
[2]INAF - Istituto di Radioastronomia
[3]Observatoire de Bordeaux

Abstract

The Uniboard is a project for a general purpose board, containing several large FPGAs, to be used as a general component for the next generation VLBI correlator. The Digital Receiver Working package deals with the implementation of a wideband frequency demultiplexer, that divides the received radio band (up to a bandwidth of 8 GHz) into enough narrow band channels (64 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).

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

1 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.

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

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

FFT: Fast Fourier Transform

channel: A portion of the input signal in the frequency domain, as processed by the digital receiver. Here 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.

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

Bandwidth: The frequency span of a signal. For a complex signal it is equal to its samplig 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.

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

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, or to implement

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.

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

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

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

SFDR: Spurious Free Dynamic Range: is the ratio (usually in dB) between the maximum input signal (usually a RFI) and the strongest spurious interferring signal.

Uniboard: The general purpose board used in this project.

[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.

2 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. With a goal input bandwidth of 8 GHz and output bandwidth of 64 MHz, this means that a complete coverage requires 128 channels.

The initial version of the Digital Receiver design will use a 4 GHz (8 GS/s) sampler, with 64 output channels. Extension to a 8 GHz, 16 GS/s sampler will be considered, but will be actively studied in a possible extension of the project.

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.

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

The specifications are thus:

- Input band: 4 GHz (8.192 GS/s), with 6 or 8 bit samples [2] Possibility of using the board with two 4.096 GS/s or four 2.048 GS/s is desired
- Input format: 16 byte-wide LVDS inputs, carrying 16 consecutive samples at 512 MS/s (1024 MS/s in the advanced design), with DDR or QDR clock. Sample-to-sample skew acceptable. An independent bidirectional serial line is available for ADC control
- Output band: selectable from 1 to 64 MHz (2 to 128 MS/s), 1 to 8 bit encoding, real samples, selectable USB or LSB, start frequency arbitrarily positionable inside the input band (with some restrictions TBD)
- Output format: Standard VLBI frame, encapsulated in a standard E-VLBI UCP IP packet. All parameters programmable
- Number of output bands: 128 desirable, at least 64 (4096 MHz/64 MHz)
- RFI immunity: a out-of-band interfering signal must be attenuated at least by (80?)dB (TBD)

3 Architecture

To implement these specifications, an architecture with a polyphase FFT followed by a series of digital BBCs has been proposed. The input band is first divided in a number of wide *FFT channels*, and then each FFT channel is divided in a number of narrow *DBBC channels.*

The first stage is a 64 point FFT algorithm, with 32 possible complex outputs (only positive frequencies of the input signal) spaced 1/32 of the input bandwidth.

The algorithm is divided between 4 input (*backnode, or BN*) and 4 output (*front node, or FN*) FPGAs. The conceptual structure is shown in fig. 1 and 2, 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, 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.

The polyphase section uses a decimation-in-time structure, to minimize vertical interconnections between consecutive samples. The board processes in parallel 8 consecutive samples, 2 for each of the 4 *input* FPGA's. FPGA A0 processes samples 0 and 4, FPGA B0 samples 2 and 6, FPGA C0 samples 1 and 5 and FPGA D0 samples 3 and 7. Each FPGA further demultiplexes the input signal by a factor of 8 (channel spacing of 128 MHz), and implements the polyphase filter branches for the relevant samples. The 16 samples are processed by a 16 point FFT, with real inputs and complex outputs. 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.

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

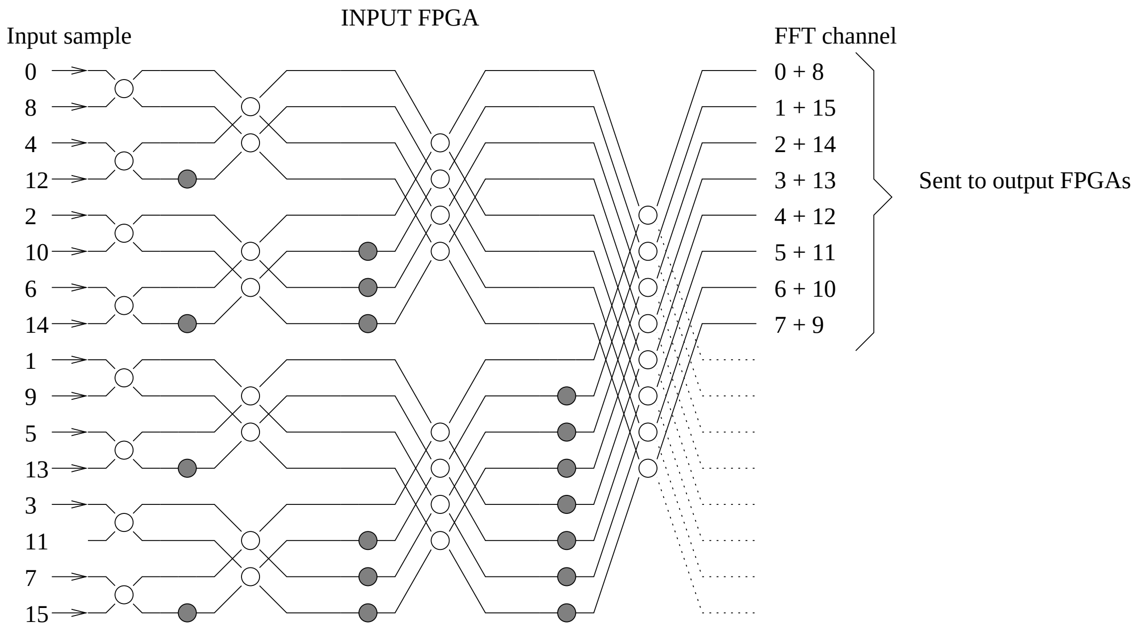

Figure 1: Structure of the FFT algorithm. Butterfly stages are divide between 4 input FPGAS (one shown) and 4 output FPGAs (next figure) 1)

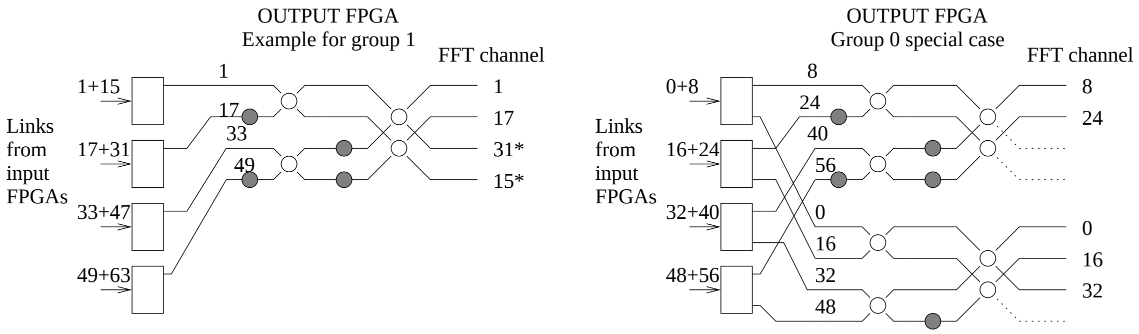

Figure 2: Structure of the FFT algorithm. Output FPGA (last 2 stages). A total of 8 groups are needed for the complete FFT. a) generic block (example for group 1). b) Dedicated block for group 0

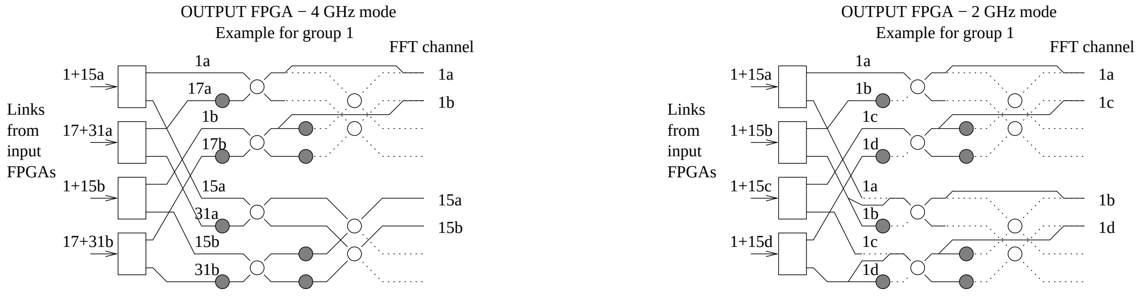

Figure 3: Output stage for 16 channels (right) and 32 channels (left) FFT.

The corresponding outputs from each input FPGA are combined together in a output FPGA, using a 4×4 butterfly stage (fig. 2a). The four outputs of this stage represent 4 output channels of a 128-point FFT. In the particular case of channel 0 from the input FPGA, the outputs for channels 8 of the first FFT are combined as usual, and those for channel 0 are combined in a second dedicated butterfly (fig. 2b). This latter does not require multipliers, as all twiddle factors are either 1 or i. Thihe 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. 4.

If the input signal is sampled at a lower data rate, it is possible to use a slightly modified butterfly stage (fig. 3) to analyze two signals at 4 GHz bandwidth, or four at 2 GHz bandwidth.

Each input FPGA produces 8 complex outputs. The corresponding output of the four input FPGAs can be combined together 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 $k = 1 \ldots 7$ contain channels k, $16 - k$, $16 + k$ and $32 - k$ (see figure 4).

The available data bandwidth between input and output FPGAs allows two groups to be processed in each output FPGA. Therefore the 16 DBBC implemented in each FPGA can process only portions of the input bandwidth belonging to two channel groups. This poses constrains in the position of the output DBBC channels, as a DBBC channel must be placed completely inside a FFT channel. Band limitation is quite severe. Assuming four 2.5 GHz serial links, a maximum of 20 bit per sample (16 bit using a 8B/10B coding) can be transmitted. This means just one group of FFT channels can be processed by each output FPGA, with a representation of 8 or 10 bits per real sample.

To use the board at higher input bandwidth, either higher link speed (theoretically up to 6.5 Gb/s can be acheved) or a lower number of bit per samples must be used.

4 Input FPGA

The block diagram of one input FPGA is shown in fig. 5.

The input samples are provided on a backplane parallel interface. The 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 accomodate 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 links are sorted in such a way that the 8 samples arriving at each input FPGA correspond to 8 equally spaced samples, with an equivalent sample rate of 2 GS/s. Samples arriving at input FPGAs BN0 to BN3 are interleaved for a total sample rate of 8 GS/s, with an offset of 0, 1, 2 and 3 samples, respectively.

Input streams must be syncronized to allow for different propagation paths in the electronics. Syncronization can be achieved by substituting sampler symbols with a repetitive datapattern, and measuring the delay with respect to system wide timing signals. In this way it is not essential to syncronize the

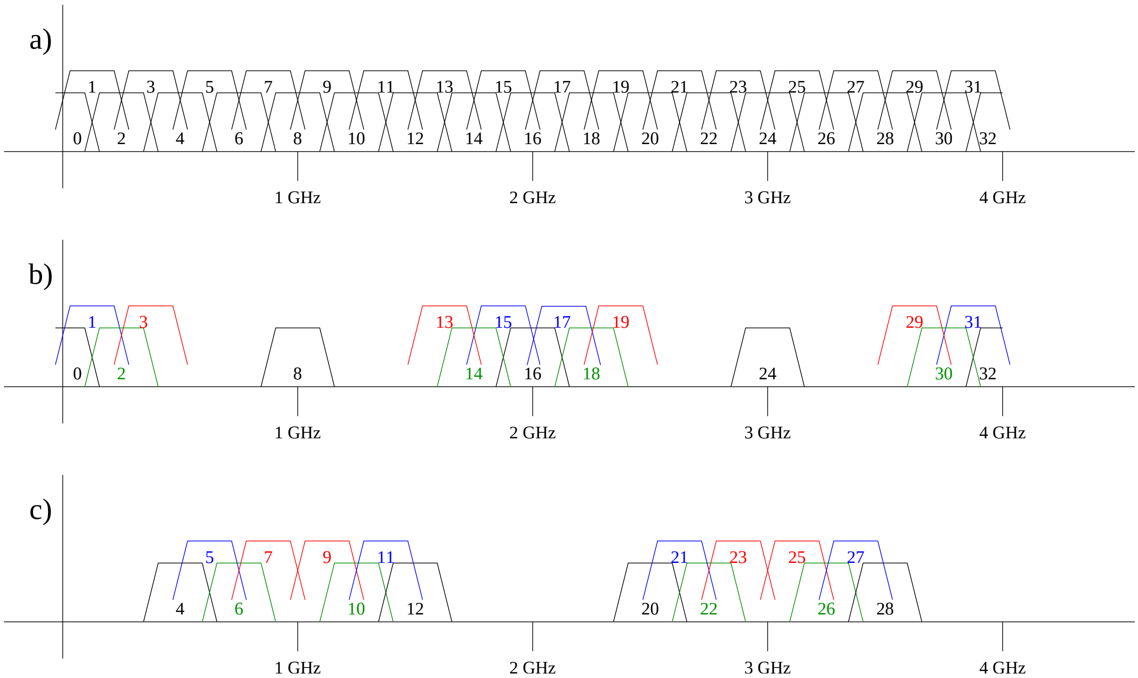

Figure 4: Position of the FFT channels in the input bandwidth. (a) All channels; (b) Channel groups 0 (black), 1 (blue), 2 (green), 3 (red); (c) Channel groups 4 (black), 5 (blue), 6 (green), 7 (red)

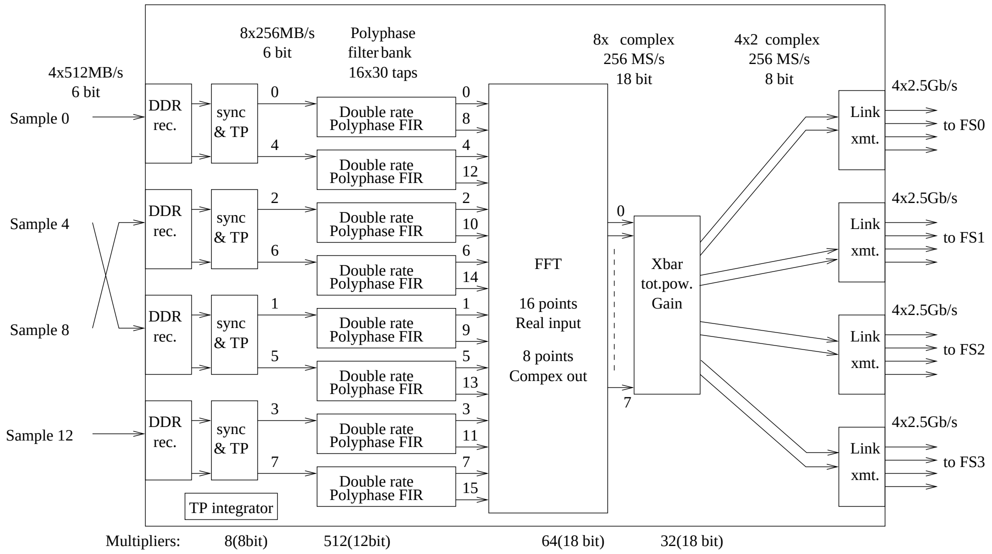

Figure 5: Structure of the input FPGA

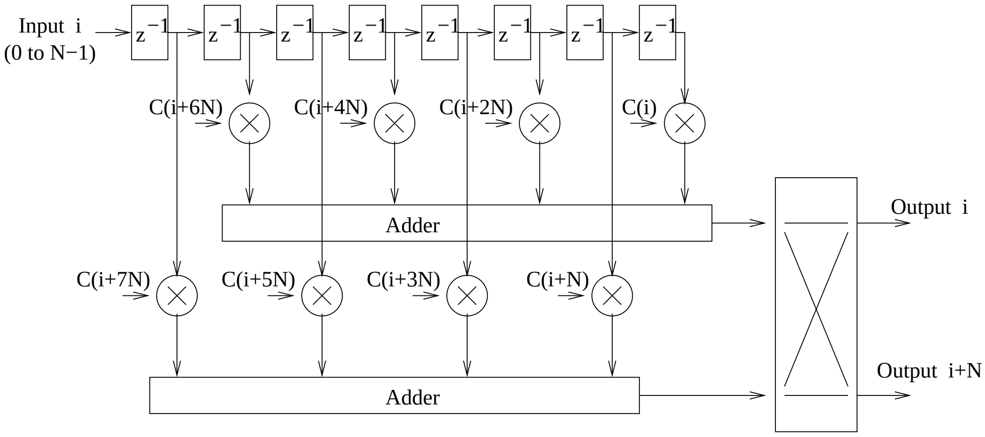

Figure 6: Double rate polyphase filter branch. On odd cycles the two outputs are exchanged, to remove phase slope

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 syncronization delay for each channel is applied in a short FIFO.

The polyphase filterbank is implemented in a distributed way across the 4 FPGAs. 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.

4.1 Input from ADC and bit reversal operation

The input samples are provided on a backplane parallel interface. The 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 accomodate 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. Samples are provided at the 16 board inputs in partially *bit reversed* sequence: FPGA BN0 processes samples 0, 4, 8 and 12, FPGA BN1 samples 1, 5, 9 and 13, FPGA BN2 samples 2, 6, 10 and 14 and FPGA BN3 samples 3, 7, 11 and 15. The remaining bit reversal operation is performed inside the FPGA.

The second and third stream are swapped in the internal FPGA routing (fig. 5), and the remaining of the bit reversal is a natural consequence of the DDR demultiplexing and of the structure of the double rate polyphase algorithm.

4.2 Polyphase filter

Each data stream is processed in two short FIR filters (fig. 6), that prodice 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 $i + 32$ are exchanged.

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 86%, and more than 90 dB of stopband attenuation has been designed, and used in the simulations.

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

Multipliers have fixed coefficients, but coefficients depend on the FPGA chip position. It is convenient to have a single personality for all identical (FS or BS) FPGAs, and thus coefficients are better stored in a memory, either loaded at system startup (e.g. organizing all coefficients as a single long shiftregister, serially loaded at startup), or as 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.

Samples from the polyphase filter are 24 bit wide, so a rescaling is necessary. Considering up to 5 bit growth (all the power in a single FFT channel) 11 bits must be discarded. Samples to the FFT are 13 bits wide.

4.3 FFT block

These 16 real outputs 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 *complex* channel 0.

Input from the polyphase filter block is 13 bit, trat 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.

4.4 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 compare the corresponding total power outputs in all chips, and program the corresponding scale factor to the same value.

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.

Each BS FPGA is connected to each FS FPGA by four fast signals, implementing a x4 fast serial link. Each link carries a real data stream, with subsequent samples directly serialized on the link. The number of bits in the output samples determine the overall SFDR, as examined in chapter 6. 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 is implemented in hardware in the Stratix 4 FPGAs, and is instatiated using the `altGX` Quartus macro. Link speed is 8, 10, 16 or 20 times the sample rate, i.e. 2, 2.6, 4.1 or 5.1 Gbps, if no complex data rate change is assumed. The link normally operates using a 8b/10b encoding, that provides better data integrity at the expense of a 20% bandwidth loss (8 bit samples transmitted as 10 bit), but it is in principle possible to operate it at 2.6 Gbps with 10 bit samples. Using a 2/3 data rate change, 10 bit samples can be transmitted using a 3.4 GHz link, and reaching approx. 85dB of SFDR. The most appropriate sample rate will be determined by analyzing the performances of the serial link on the board.

A total power detector is present also on the input signal. The detector is distributed across the 4 BS FPGAs, and the measurement is obtained summing together the result for each FPGA. This implies that the settings and the integration timing must be the same for all chips.

The total number of hard multipliers required in each stage is shown in the bottom line of figure 5. Most of them (512) are used in the polyphase filter, with 64 used in the FFT and 32 in the output stages.

4.5 Control and monitor points

Each chip is seen as a custom NIOS peripheral, with a standard Avalon port for monitor/control, four 10 bit Avalon streaming inputs for the ADC inputs and four streaming outputs for the high speed links.

The standard port uses 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 1.

Address	Write	Read
0x00	Test point select	Test point readback
0x01	Global control	Global status
0x02	Interrupt control	Interrupt status
0x03-07	unused	unused
0x08	Input total power control	Input total power
0x09	FFT total power control	unused
0x0a	ADC Serial control/transmit	ADC status/receive
0x0b	unused	unused
0x0c-0f	Input link control	Input link status
0x10-18	FFT gain	FFT total power
0x19-1b	unused	unused
0x1c-1f	Output link control	Output link status

Table 1: Register mapping for Uniboard Digital Receiver input chip

Bit assignment for most registers is TBD, but the main functions are outlined here for all registers. All assignments are 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 16 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 (with pulse stretcher), or slow status signals. Lines 4-7 are connected to test posts. The particular mapping of the lines connected to each test point is TBD.

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

Register 1 controls the general behavior of the board. Bits are tentatively assigned as in table 2.

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 2: 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 is used by the module (still TBD). Interrupt can be useful for total power reading, for the serial line from/to the ADC, and for error conditions.

Register 8 (table 3 controls the integration time and general functionality of the input total power meter. Total power is read from the same 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

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 3: Total power control register

- 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, but only total power and DC offset functions are available. The corresponding read register is unused as total power is read for each individual FFT channel.

Register `0xa` is used to send/receive data over the ADC serial data line. The line is basically a bidirectional UART, and its exact functionality is still TBD.

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 syncronization 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 signal 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 pseudorandom sequence, or with a syncronization pattern. If allowed by the IP, link parameters (e.g clock phase) can also be adjusted.

These registers control also specific functionalities of the high-speed link macrofunctions. This part is completely TBD.

5 Output (front side) FPGA

The block diagram of one output FPGA is shown in fig. 7.

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 250 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.

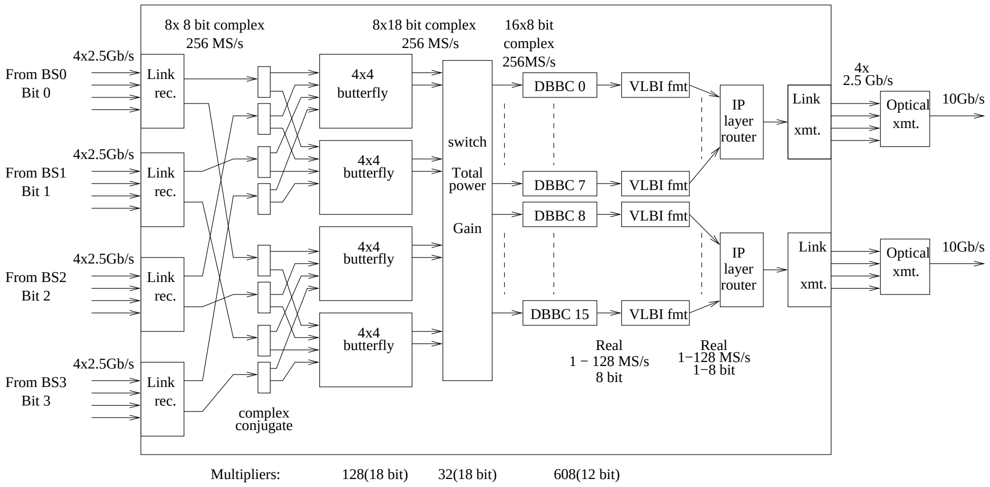

Figure 7: Structure of the output FPGA

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

From these signals, up to 16 real data streams are computed, with independently selectable bandwidth and position. Bandwidth can be chosen from 1 to 64 MHz (2 to 128 MS/s), with a position resolution of 0.01 MHz.

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, to check for data integrity using pseudo random sequences, and to measure the sample syncronization.

No total power measurement is needed, as this is already measured in the input FPGA. Signal integrity can be measured after the FFT butterfly, configured in bypass mode.

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. 2), 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. Also depending on the group number, it is possible to select the real, imaginary, complex direct or complex conjugate of the input samples.

To support 2 GS/s and 4 GS/s ADCs, the output of just one or two input FPGAs are analyzed. For 2GS/s operations, two of the four input signals are simply copied to the outputs. For 4 GS/s, the first stage of the butterfly operates normally, while the second allows to select two of the four inputs (see fig. 3).

Butterfy 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 requantized to 8 bit.

The FFT requires 128 multipliers, with other 32 multipliers for total power and first rescaling stages. This requires 40 of the available 161 DSP blocks in the FPGA.

5.3 DBBC blocks

The FFT stage is followed by an array of digital BBC's. A minimum of 16 DBBC is required to split each of the 256 MHz FFT channels 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 possibile outputs of the FFT stage. It is also possibile 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 real channels from the FFT.

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) does not degrade with the decimation factor. A minimum rate decimation of 2, or bandwidt decimation of 4 is used (128 MS/s, 64 MHz bandwidth), at which each multipler is recycled by a factor of 4. Exploiting filter symmetry, filter length is thus $2D$ times the number of available multiplers, with D the bandwidth decimation factor.

Considering the number of available DSP blocks, a total of 7 blocks are availablefor each DBBC, excluding final total power and rescaling. The LO mixer can be implemented using a single block with 18 bit resolution, leaving 6 blocks for the filter, i.e. 3 for each of the real and imaginary parts. Possible filter lengths for bandwidth decimation D are thus either $36D$ taps, with 12 bit resolution, or $24D$ taps, with 16 bit resolutions. Resolution determines the maximum stopband rejection, and the approximate filter performaces are respectively 60 dB rejection, 92% useful bandwidth (12 bits), or 85 dB stopband rejection, 80% bandwidth (18 bits).

A last total power meter and rescaling stage is used to adjust the signal level, and the output is re-quantized to the required number of bits.

5.4 Output formatter

The signal is then formatted in a VLBI packet, encapsulated in a UTP *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 card output bandwidth. For example using 16 output channels with 3 bit representation and 60 MHz per channel (128 MSample/s), a total of 6 Gb/s is required. Considering packetization and coding efficiency, this data rate is within the bandwidth of a single 10G link. To avoid bottlenecks two links are used, one for each group of 8 BBCs. Thus the Uniboard connects with 8 10G 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 addedd at the beginning of each packet. The VDIF header contains informations about the absolute time, the quantization, an ID for the station and the data thread (BBC and board), 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 and port, source port. 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 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.

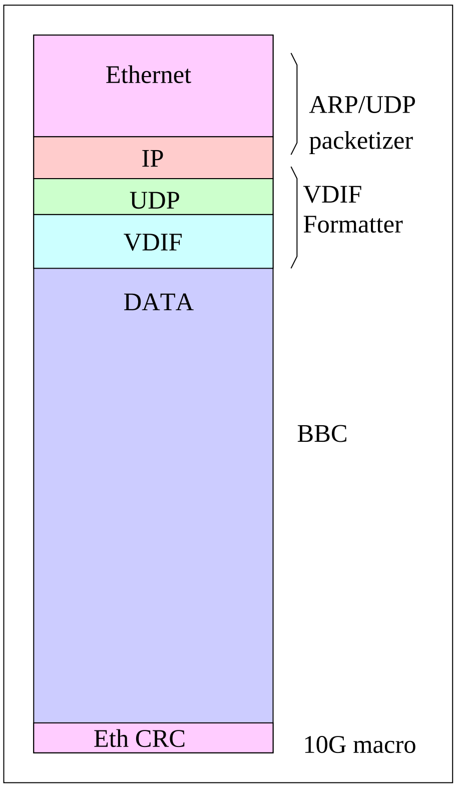

Figure 8: Ethernet packet structure. The raw data samples are encapsulated in a VDIF packet, a UDP header and checksum is then added, and the resulting frames are sent as IPv4 compliant jumbo packets

5.5 Programming interface

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 4. The first 8 positions (only 3 used) are used for general control. Next 8 are used for the formatter and output sections, and are completely TBD at the moment. 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 *16 channel FFT mode*.

address	write	read
00	Test point select	Test point readback
01	General control	General status
02	Interrupt register	Interrupt status
03-07	unused	unused
08-0f	Output formatter	Formatter status
10	FFT Total power ctl	FFT Total power status
11	BBC Total power ctl	BBC Total power status
12-13	unused	unused
14-17	Input 0-3 FIFO ctl	Input 0-3 FIFO status
18-1f	FFT 0-7 control	FFT 0-7 total power
20	BBC 0 ctl	BBC 0 status
21	BBC 0 frequency	BBC 0 total power
...	...	...
3e	BBC 15 ctl	BBC 15 status
3f	BBC 15 frequency	BBC 15 total power

Table 4: Register mapping for Uniboard Digital Receiver output chip

FFT control register allows to specify FFT mode, and output gain.

FFT modes are listed in table 5.

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.

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 5: FFT modes

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. 6. Bits $0-2$ select the decimation, and thus the band. Bits $3-5$ set the BBC input selection. If bit 6 is set, the output is substituted with a pseudorandom 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 $16-24$ and bits $25-26$ control respectively the output gain and the adopted quantization scheme.

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	=
26-25	Output quantization	=

Table 6: Register mapping for the BBC Control/Status registers

6 Simulations

To test the conceptual design, a simulator program has been written.

The simulator implements the FFT algorithm with floating point mathematics, and thus do not take into account rounding errors. Explicit rounding has been performed between the "input FPGA" and "output FPGA" sections, to test the effects of limited communication bandwidth. Filter taps for the polyphase filter are quantized to 18 bits.

In all tests the program analyzes a sequence of input samples, produced by a generator program, and computes all the 32 FFT channels, with a 1/16 decimation factor. The output of all 32 channels is then stored on disk. Each of the 32 outputs is processed by a FFT total power spectrometer, with a variable number of spectral points. The spectrometer uses Blackmann-Harris windowing, on segments of data with an overlap of 50% between successive segments.

A first test has been performed using a "colored" Gaussian noise, i.e. a dataset composed of a white noise with a specific spectral content. The same dataset has been analyzed before and after the polyphase filter, in order to compare the two spectra. The result is shown in fig. 9. The black line (original spectrum) completely covers the colored lines (spectra of each FFT channel, shifted to the correct place), apart from the points near the channel edges.

The remaining tests have used a simulation vector consisting of a pseudo-random Gaussian noise, with a fixed seed, with and without a sinusoidal tone added (respectively signal and reference dataset). The tone has a period of 20 clock cycles, and an amplitude of 0.1 times the Gaussian noise RMS. The vector length is 160 Msamples, limited by disk space considerations.

The spectrum of both the signal (noise plus tone), and of the difference between the signal and the reference has been calculated. In a perfect, linear system this latter should give the pure spectrum of the monochromatic tone. In the real case, quantization noise is different in the two datasets, but is anyway much lower than the input Gaussian noise, and thus allows to determine the presence of ghost tones much easily.

Using 18 bit quantization between the two stages, the spectrum is shown in fig. 10. Individual FFT channels are colored in sequence. Each spectral point is typically observed in 2 different channels, and

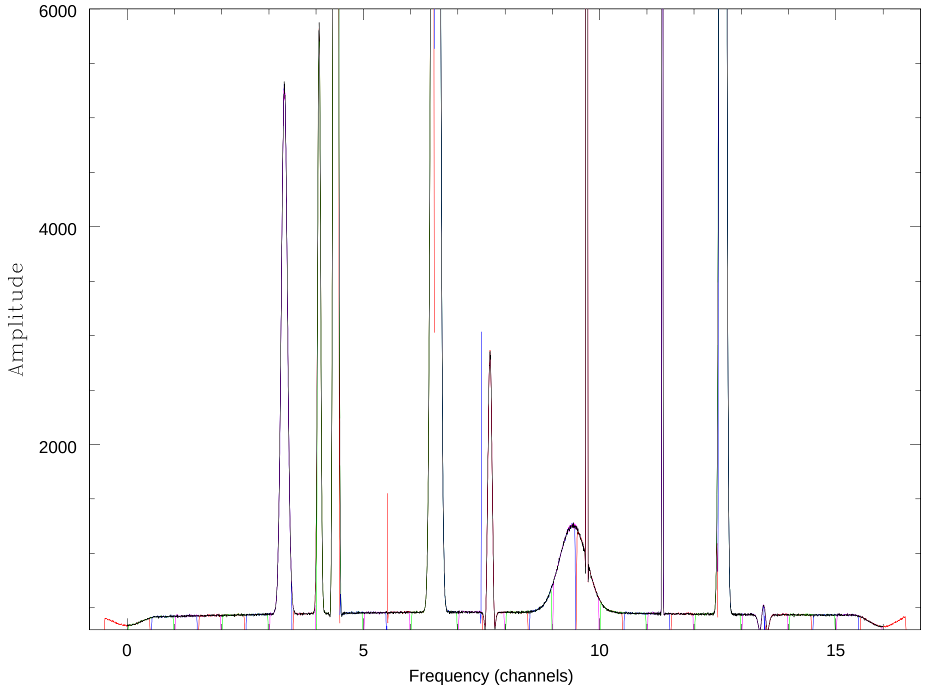

Figure 9: Spectrum of a pseudorandom Gaussian noise with predefined spectral density. Black: obtained with a 8192 point FFT spectrometer; Colored: obtained with polyphase filter followed by 32 256-point FFT spectrometers

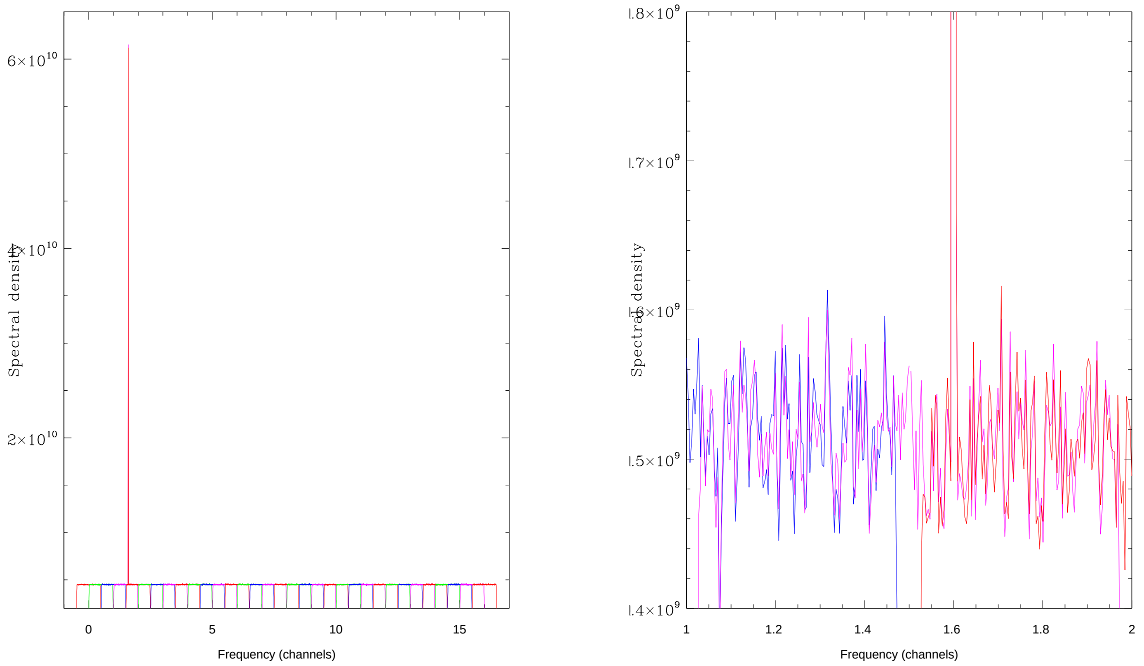

Figure 10: Spectrum of white Gaussian noise plus a spectral line. Left: whole spectrum with 32 FFT channels. Right: Zoom of FFT channels 2 (blue), 3 (purple) and 4 (red)

the relative spectra overlap with good accuracy, as seen in fig. 10(right).

If the signals between the two stages are quantized, we expect both an increase in noise, due to the extra quantization noise, and the possibility of ghost lines, due to the nonlinearity in the quantization process. This latter is difficult to predict, but if the signal power does not exceed the noise power in each channel group, the distortion cancels statistically, and only a (small) increase in the noise is expected.

To test this, the spectrum of the difference between the signal and reference datasets has been computed for various quantizations between the input and output FPGAs. In fig. 11 the cases for no quantization (24 bit mantissa floating point numbers) and 18 bit quantizations is shown. For no quantization (left) the ghost lines are around 82 dB below the carrier, as expected from the filter specifications. Noise is basically the quantization noise expected for a 24 bit representation. For 18 bit quantization, the noise in the signal and reference datasets is not identical for those channels containing the spectral tone. For the chosen line frequency this happens for channel groups 3 and 4. In these channels therefore we have a higher level of measurement noise, and it is difficult to measure the spurious free dynamic range. To overcome this problem, a FFT with a higher number of spectral points has been computed, increasing the amplitude of the coherent tone correspondingly. The right spectrum of fig. 11 has been computed with 16K points per FFT channel instead of 256, and the resulting SFDR is close to the previous value. No ghosts are expected in the remaining groups, and none are seen above the -82 dB level seen before.

The problem is worse with 8 and 6 bit quantizations. In these cases, to increase the line amplitude the length of the FFT has been increased up to 1 million points, i.e. the maximum practical value for datasets containing 5 million points. For 8 bit cases, the line then has an amplitude of about 70 dB above the noise in each spectral point, and a spurious free dynamic range of 70 dB is thus demonstrated.

The same test, using 6 bit quantization, show no ghosts up to 65 dB below the tone. Tests using hardware models are necessary to assess higher SFDR.

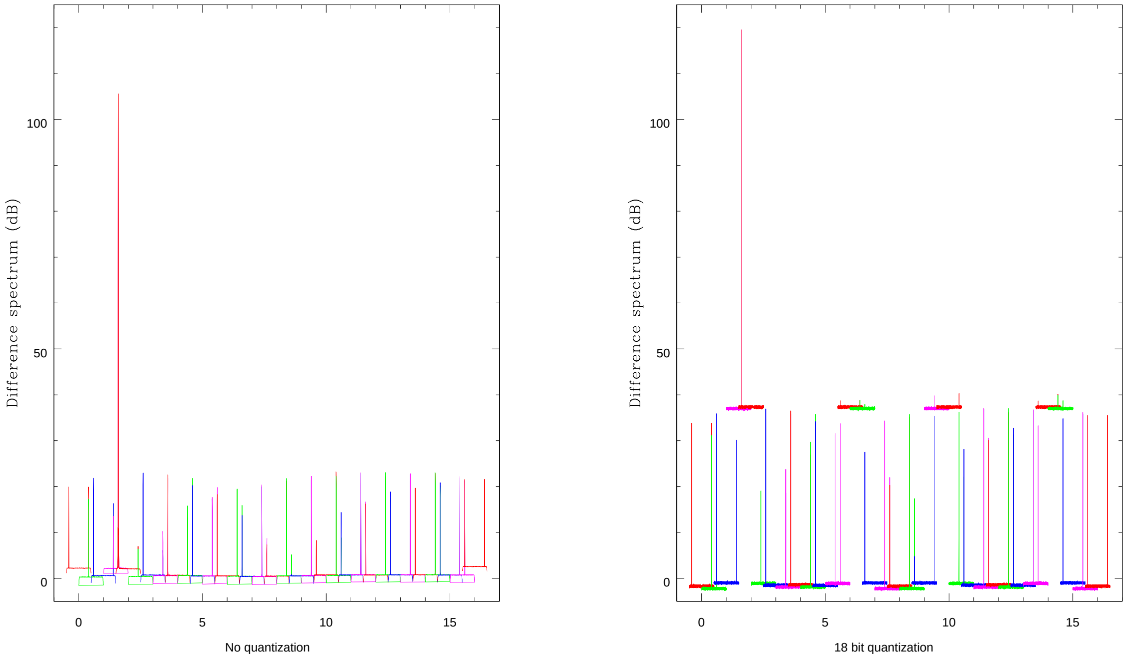

Figure 11: Spectrum of the difference between source (noise+tone) and reference (noise only). Left: using no quantization, right: with 18 bit quantization between input and output FPGAs

Contents

1 Glossary **1**

2 Specifications **2**

3 Architecture **2**

4 Input FPGA **4**
4.1 Input from ADC and bit reversal operation 6
4.2 Polyphase filter 6
4.3 FFT block 7
4.4 Output stage 7
4.5 Control and monitor points 8

5 Output (front side) FPGA **9**
5.1 Input section 10
5.2 FFT block 10
5.3 DBBC blocks 11
5.4 Output formatter 11
5.5 Programming interface 13

6 Simulations **14**