# FP7- Grant Agreement no. 283393 – *RadioNet3*

Project name: Advanced Radio Astronomy in Europe

Funding scheme: Combination of CP & CSA

Start date: 01 January 2012 Duration: 48 month

# Deliverable 8.4

## Firmware design document: digital receiver

Due date of deliverable: 2014-01-31

Actual submission date: 2014-02-04

Deliverable Leading Partner: ISTITUTO NAZIONALE DI ASTROFISICA (INAF), Italy

An European project supported within the 7th framework programme

# 1 Document information

| | |
|---|---|
| Document name: | UniBoard$^2$ Firmware design document – digital receiver |
| Type | Report |
| WP | 8 (UniBoard$^2$) |
| Authors | Giovanni Comoretto (INAF) |

## 1.1 Dissemination Level

| **Dissemination Level** | | |
|---|---|---|
| **PU** | Public | X |
| **PP** | Restricted to other programme participants (including the Commission Services) | |
| **RE** | Restricted to a group specified by the consortium (including the Commission Services) | |
| **CO** | Confidential, only for members of the consortium (including the Commission Services) | |

## 1.2 Terminology

| | |
|---|---|
| **ADC** | Analog to Digital Converter |
| **BF** | BeamFormer |
| **bps** | Bits per second |
| **BW** | Bandwidth |
| **channel** | *Frequency* band, unit output of the filterbank. For a two stage architecture, *Coarse channel*: and *Fine channel* refer to the unit output of the first and second stage filterbanks. |
| **COTS** | Commercial Of The Shelf |
| **DDR** | Double Data Rate |
| **DSP** | Digital Signal Processing |
| **EMI** | Electro-Magnetic Interference |
| **Firmware** | Embedded or real-time code that runs on a microprocessor (e.g. written in C), or describes a programmable logic (e.g. written in HDL) |
| **FFT** | Fast Fourier Transform |
| **FPGA** | Field Programmable Gate Array |
| **Hardware** | Boards, sub-racks and COTS equipment |
| **HDL** | Hardware Description Language |
| **IO** | Input-Output |
| **IP** | Intellectual Property |
| **PHY** | Physical interface (layer1 of OSI model) |
| **QSFP+** | SFP for 40Gb Ethernet |
| **PFB** | Polyphase Filterbank |
| **RF** | Radio Frequency |
| **RFI** | Radio Frequency Interference |
| **SFP** | Small Form-factor Pluggable transceiver. An optical interface module, available with different performance and range characteristics, pluggable in a common socketed cage |
| **SFP+** | SFP for 10Gb Ethernet |
| **SOPC** | System on Programmable Chip. A processor and associated peripherals implemented using programmable logic. Typically used to control the application on the FPGA |
| **Subband** | Frequency band, unit output of the filterbank |

## 1.3 References

[1] A. Szomoru: “UniBoard$^2$ Work Package description”, RadioNet3 283393 (2011)

[2] G. Schoonderbeek: “UniBoard$^2$ Architecture-Hardware design document”, ASTRON report INFRA2011-1.1.21 (2014)

[3] G. Comoretto, A. Russo, G. Tuccari, A. Baudry, P. Camino, B. Quertier: “Uniboard Digital Receiver Design document”, Arcetri Technical Report 5-2011

http://www.arcetri.astro.it/images/data/Reports/11/5_2011.pdf

[4] G. Comoretto, G. Knittel, A. Russo: “Uniboard Pulsar Receiver Design document” (2012)

[5] F. J. Harris: “Digital Receivers and Transmitters Using Polyphase Filter Banks for Wireless Communications”, IEEE Trans. on Microwave Theory And Techniques, 51, 4(2003)

[6] Analog Devices: “JESD204B Survival Guide”

http://www.analog.com/static/imported-files/tech_articles/JESD204B-Survival-Guide.pdf

[7] G. Comoretto, A. Russo, G. Tuccari: “A 16 channel FFT multiplexer”, Arcetri Technical Report 1-2009

http://www.arcetri.astro.it/ricerca/rapporti-tecnici/205-reports/564-09-1

[8] G. Comoretto: “A design method for very large FIR filters”, Arcetri memo series 3/2012

[9] G. Comoretto: “A base 10 FFT core for FPGAs”, IEEE International Conference on Acoustics, Speech, and Signal Processing, Florence, May 4-9 2014 (in press)

[10] G. Comoretto, A. Melis, G. Tuccari: “A wideband multirate FFT spectrometer with highly uniform response” Experimental Astronomy **31**, 59-68 (2011).

[11] 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

## 1.4 Content

# 2 Introduction

A radio telescope receives a wideband RF signal, with typical bandwidths, in current systems, of the order of several GHz. Only a portion of this signal is analysed, because of hardware limitations, presence of human made interfering signals, or absence of useful astronomic information. Moreover for technical limitations it is often advantageous to process the signal in relatively small bands, up to some tens of MHz, and thus to divide the input bandwidth into many small narrow bands. This task is performed in an astronomic radio receiver. An example of such a system is the MARK-x family of VLBI terminals, in which up to16 couples of narrow bands (from less than1 MHz to 32 MHz) can be arbitrarily positioned inside an input band of the order of 1 GHz.

A digital receiver performs the same function using digital signal processing. The radio signal, either just amplified up to a viable level or also converted in frequency, is digitized by a wideband ADC and then decomposed into narrow bands. These bands can be processed locally, e.g. for single dish spectroscopy or for connected element interferometry, stored on disk, for conventional VLBI, or sent to a high speed Internet link for real time e-VLBI.

## 2.1 Prior work in the field

A digital receiver application has been designed for the UniBoard platform [3]. The design interfaces with a high speed analog-to-digital converter, providing an instantaneous bandwidth of up to4GHz. A dual stage filterbank channelizes the digitized signal, providing a set of up to 64 VLBI channels. Each channel represents a portion of the input band with an arbitrary start frequency and width, that is filtered, down-sampled and formatted according to the VDIF format.

Another digital receiver has been developed for the FP7 project “Beacon in the Dark”, for a pulsar timing machine [4]. It receives two polarizations with a bandwidth of 3.2 GHz each, and produces a total of 144 partially overlapping channels of 24 MHz each. These channels are positioned in order to avoid heavily RFI contaminated spectral regions (Fig. 1).

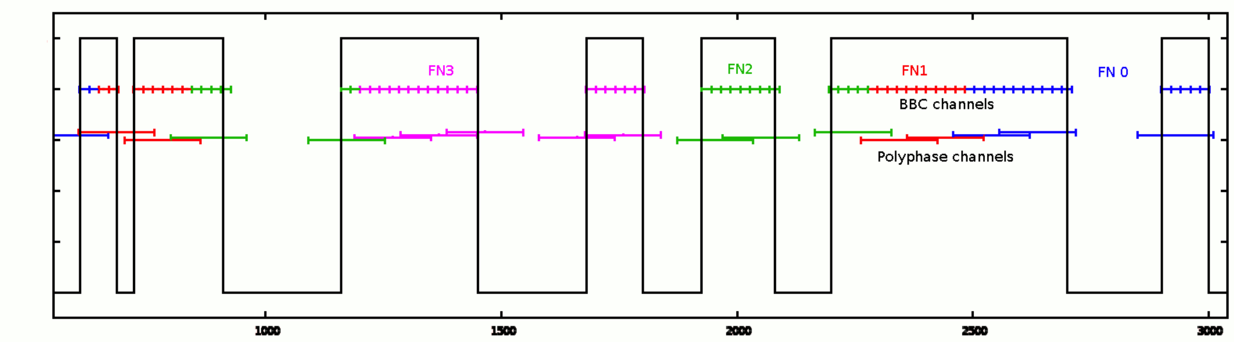


Figure 1: Coarse and fine channelizer bands for the UniBoard Beacon digital receiver. Black line denotes portions of the input band that are observed or filtered out in the RF receiver. Larger bars denote broad channels, colour coded for the UniBoard nodes where they are implemented. Smaller bars denote fine channels.

## 2.2 The UniBoard$^2$

The UniBoard$^2$ platform [2] is a high performance DSP platform that represents an evolution of the previous UniBoard, developed as part of the RadioNet-FP7 project. In the most likely proposed architecture it is composed of 4 identical nodes, with one FPGA per node. The initial version will use Arria10 FPGAs, replaced by Stratix10 in the final version. Each node is interfaced to 6 front panel QSFP+ cages, that can be connected to a 40 Gbps optical link, and to a passive backplane (Fig. 2). The back plane provides up to 72 high speed serial links per node. An option of 48-bit LVDS lanes per node has been considered, but probably will not be present.

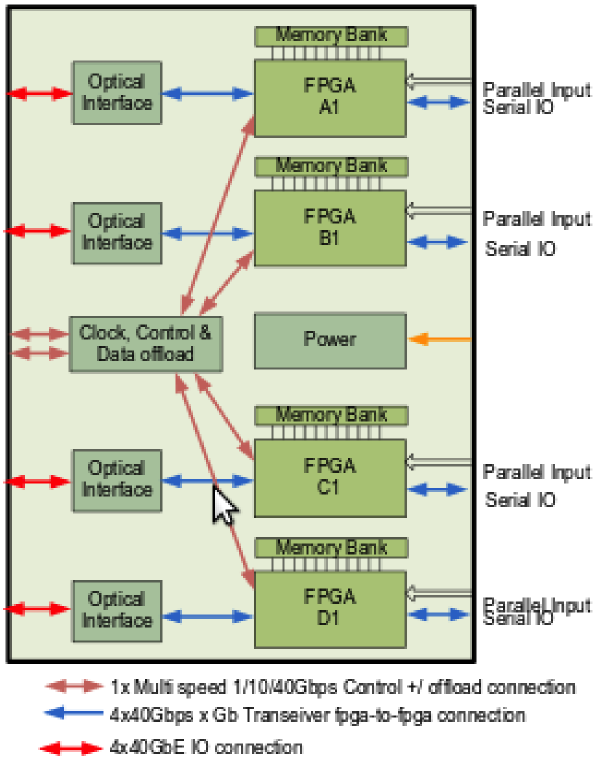


Figure 2: UniBoard$^2$ single column concept

Local memory is provided by two (or possibly 4) DDR4 modules, and can be extended using external Hybrid Memory Cube (HMC) modules connected via the backplane serial links. However in Arria10 family the maximum DDR4 speed is limited to 2666 MTransfer/s, and the HMC support is preliminary.

Node-to-node interconnection is performed using dedicated backplanes. A standard backplane with two QSFP+ modules and a complete 4 × 4 mesh will be available together with the initial version of the board.

The FPGA initially used will be an Altera Arria 10GX1150, with 660 general purpose usable pins, 96 high speed transceivers and a total of 3036 18 × 19 bit multipliers.

The final board will use Stratix10 FPGAs. This device will have the same pin out, and then the same number of transceivers and pin count, but a much higher number of internal resources, and an internal clock rate at up to 1 GHz. The external link bit rates will increase to 26 GHz, with 56 GHz on selected lines. Memory interface will support the maximum transfer rate of 3200MT/s of the DDR4 standard, and the Hybrid Memory Cube standard interface modes are full supported.

This design is based on the expected performances of the Arria10 family, but can be easily scaled to take advantage of the Stratix10 increased performances. The larger number of multipliers can be used both to increase the number of available output channels, and to increase the internal sample size, for higher spurious signal rejections in high RFI environments. The higher internal clock speed and link bit rate can be used to scale up the total RF receiver bandwidth to 8-10 GHz, matching the available bandwidth of current centimetre and millimetre radio astronomy receivers.

## 2.3 The UniBoard$^2$ Digital Receiver application

The Digital Receiver application will be redesigned for the UniBoard$^2$. The increased capabilities of the new board can be exploited in various ways.

- The larger number of resources per FPGA, and the higher clock speed allows to implement a complete digital receiver using a single node. This allows a single UniBoard to implement 4 independent digital receivers, e.g. for a dual polarization, dual frequency receiver for geodynamic VLBI observations.
- The Arria family may use multipliers with a larger word size. This allows for better noise performance and RFI immunity in critical design modules.
- The extra resources available in a single FPGA allows for different options both in the first and in the final channelization stages. The application is then designed as a modular library, in which independent elements can be tailored and assembled according to the application needs. In this way it can be adapted to widely different situations.

The general structure is similar to the UniBoard ad Beacon digital receivers, and is shown in Figure 3.

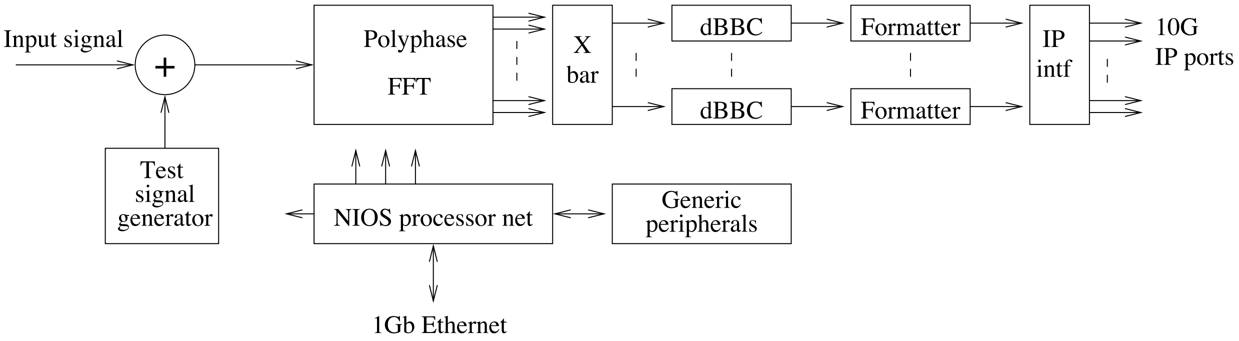


Figure 3: Structure of a dual stage digital receiver

It is composed of an ADC interface (chapter 4), a test signal generator (chapter 5), a first stage channelizer that provides uniformly spaced *broad channels* (section 6), and an array of second stage channelizers, providing the final channelization. A VDIF formatter then sends the channelized data as VDIF-over-UDP Ethernet packets (section 8).

Different options are available for the second stage channelizer (section7). The basic UniBoard digital receiver provides individually tunable channels, using a *digital baseband converter* architecture (chapter 7.1). It is possible to individually adjust each channel bandwidth (and sample rate) and position, allowing for maximum flexibility, e.g. when both wideband continuum and narrow band spectral observations must be performed simultaneously. The total number of channels is however limited, that is not usually a problem for VLBI, where the total bandwidth is limited by the data transport, but may constitute a limitation for single dish or connected elements interferometry.

For the situations requiring most of the input band to be processed a dual stage polyphase filterbank is more appropriate (chapter7.2). Some flexibility can still be retained by choosing which channels to process or to transmit for further processing, and allowing some forms of tunability. Both conventional, non-overlapping, and overlapping filterbanks can be implemented.

A generic library of commonly used functions and modules has been written for the application. The library includes commonly used data types like complex number representation and operations, serial and parallel FFT in radix 2, 4 and 5, conventional and oversampling polyphase filtering, sine/cosine and random number generation.

The digital receiver modules use also the libraries of generic digital signal processing modules and control structure written as part of the UniBoard project.

# 3 Digital Receiver common control structure

The UniBoard project has defined a standard protocol for communication with the UniBoard applications. The protocol allows read, write and modify of 32 bit registers across the system, and is agnostic about the register content. Any intelligence resides on an external computer, programmed using an high level language(C, Python, Erlang, ... ). High-level commands or procedures are decomposed in single register access commands, sent to the FPGA through the control Ethernet link.

Inside each FPGA a SOPC module runs a standard program, independent from the application that decodes UDP command packets. The simpler UDP protocol allows to implement the control program without the relatively large overhead of a TCP-IP stack. A set of standard peripherals is used for debug, program timing, download of personalities in the on-board flash memory, and generic control and monitor functions.

Each module has an address space composed by a number of consecutive 32 bit words. In the digital receiver modules we adopted a convention for the first four register positions, as in Table1.

| Word | Offest | Write register | Read Register |
|---|---|---|---|
| 0 | 0x00 | Test point | Readback test |
| 1 | 0x04 | — | Indetification |
| 2 | 0x08 | Control | = |
| 3 | 0x0c | — | Status |
| 4+ | 0x10+ | App. dependent | App. dependent |

Table 1: Common registers in all module interfaces

The first (offset zero) register controls a test point utility. The test point is associated to one (or more) external line that can be connected, for debug purposes, to an external pin of the FPGA. If the register is set to zero, the external line is at a fixed *LOW* level, to minimize electric activity and allowing multiple modules to share the same external line (OR-ing together the module lines). A number of internal signals can be connected to the external pins, and probed with an oscilloscope, using the value specified in the register for selection. The register can be read back, thus allowing a simple check of the module interface functionality without affecting the module functionality.

The identification register reports a fixed value, hardcoded in the module firmware. It can be used to check for the presence of specific modules, and to report the module version. Writing the register has no effect.

The control register has individual bits, or bit groups, mapped to specific functionalities of the module, at a global level. 32 bits are usually sufficient to set most of simple parameters, like bandwidth, start/stop, synchronization, reset. These bits, when read back, report the current setting of the associate option, allowing for a read-modify-write setting of individual options leaving the rest of the configuration unchanged.

The module general status can be checked using the status register. The register is usually polled and not written. Some bits report a Boolean status of some parameter (e.g. an overflow condition, or completion of a total power integration). These bits are reset by pulsing high (writing a 1-0 sequence) the same bit.

Registers after the first four are module dependent, and will be described in the detailed module description.

When a total power function is available, it is controlled using a total power register. Often a single total power register can control an array of total power meters, that reasonably use the same configuration. The format of the total power register is the same for all instantiations of the total power function, and is reported in Table 2.

| bit | control |
|---|---|
| 0x00-01 | TP function:<br>0=Total power, 1=DC offset<br>2=State counter, 3=RD check |
| 0x02 | General enable |
| 0x04-07 | Integration prescaler |
| 0x08-0f | Reference status RD check line |
| 0x10-1c | Integration length (ms) |

Table 2: Total power control register

The total power function allows to collect different statistics on the data stream, beyond the total power. The state counter counts the number of samples matching a given pattern, and the Random Data checker checks data integrity by comparing the received bit sequence on a specific sample bit with a pseudo random sequence given by a predefined polynomial. Integration time can be specified as an integer number of milliseconds. To allow for much different integration times, a programmable number of bits in the accumulator result is discarded before transferring the result in the 32 bit total power register.

# 4 Interconnection with the ADC

The input stage is not developed as part of the design, as it depends heavily on the choice of the ADC. Here we only give some general design considerations, and a few examples of possible ADC interfaces.

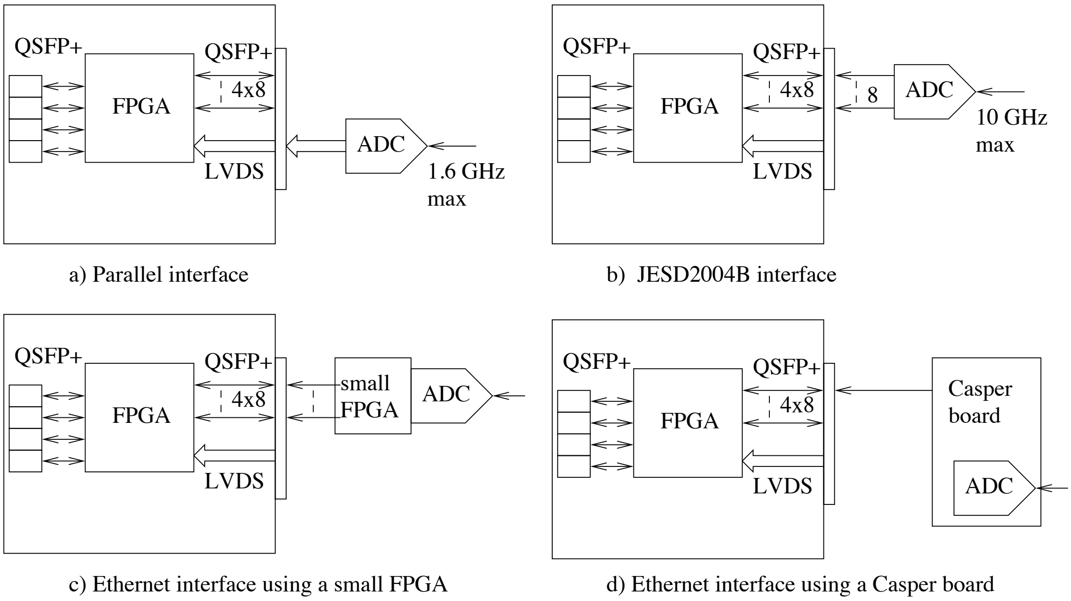
The simplest interface is a parallel LVDS (Fig. 4-a). The Arria FPGA has a maximum LVDS speed of 1.6 GSps, and each node has a total of 4 8-bit LVDS lanes. The maximum bandwidth for a single node system with 8 bit/samples is therefore 6.4 GSps, limiting the maximum ADC bandwidth to 3.2 GHz. To exploit larger bandwidths more nodes must be used in parallel, or more lanes should be added to the backplane interface (that conflicts with the available FPGA pin count).

The proposed UniBoard[2] may have no parallel inputs. In this case an ADC must be interfaced using high speed serial links, either directly or using an interposed FPGA. For example the industrial standard JESD204B [6] allows an ADC to interface using multiple links with a serial speed of up to 12.5 Gbps. 8 links are required for a 4 GHz bandwidth, at 8 bit data rate (Figure 4-b). These can be physically transported using two QSFP+ optical links.

Very few ADCs with the required sample rate(8GSps) are available. Custom ADCs and a digitization module based on commercial components are now being developed as part of the UniBoard project by the University of Bordeaux. The ADC will be interfaced with a small FPGA, using serial links for the output.

Standard Ethernet frames, or frames using the simplified *Uthernet* protocol (a point-to-point address less version of the Ethernet), developed as part of the UniBoard project, can be used for the transport layer (Figure 4-c).

For smaller bandwidth, the system can interfaces with a small CASPER board, using one of the commercial ADCs available in the CASPER project (Figure4-d).



a) Parallel interface

b) JESD2004B interface

c) Ethernet interface using a small FPGA

d) Ethernet interface using a Casper board

Figure 4: ADC interfacing to the UniBoard$^2$

# 5 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 radio astronomy 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 $\sigma_N$ in accordance to the radiometer equation $\sigma_N = N/\sqrt{\tau B}$,
- a few (2 in the UniBoard$^2$)pure sinusoidal tones, with power ranging from a small fraction to most (e.g. 90%) of the total power in the samples,
- a comb of calibration tones with predictable phase.

A block schematics of the generator is shown in Figure 5. The generator operates on a parallel sample representation of the data. Both the parallelization factor and the sample bit width are parameterized.

All these components have programmable parameters (amplitude, frequency, phase) and are added to the input signal. The input signal can be blanked, to provide only the test signal.

The component is identical to the one used in the UniBoard and in the Beacon digital receivers, and is described in more detail in the relative Design Documents [3,4].

# 6 First stage polyphase filterbank

The ADC sampling frequency is much higher than the internal frequency of the FPGA logic. ADC data is thus represented in time multiplexed form. A first stage of frequency multiplexing, in which the input band is split into adjacent, overlapping channels provide several advantages:

- data is represented in continuous time series, requiring much less routing,
- data representation becomes complex valued, allowing for easier signal processing,
- any RFI contained in a single channel is less likely to produce interferences outside the channel,
- portions of the input band not containing useful astronomic data can be discarded at an earlier stage of the processing.

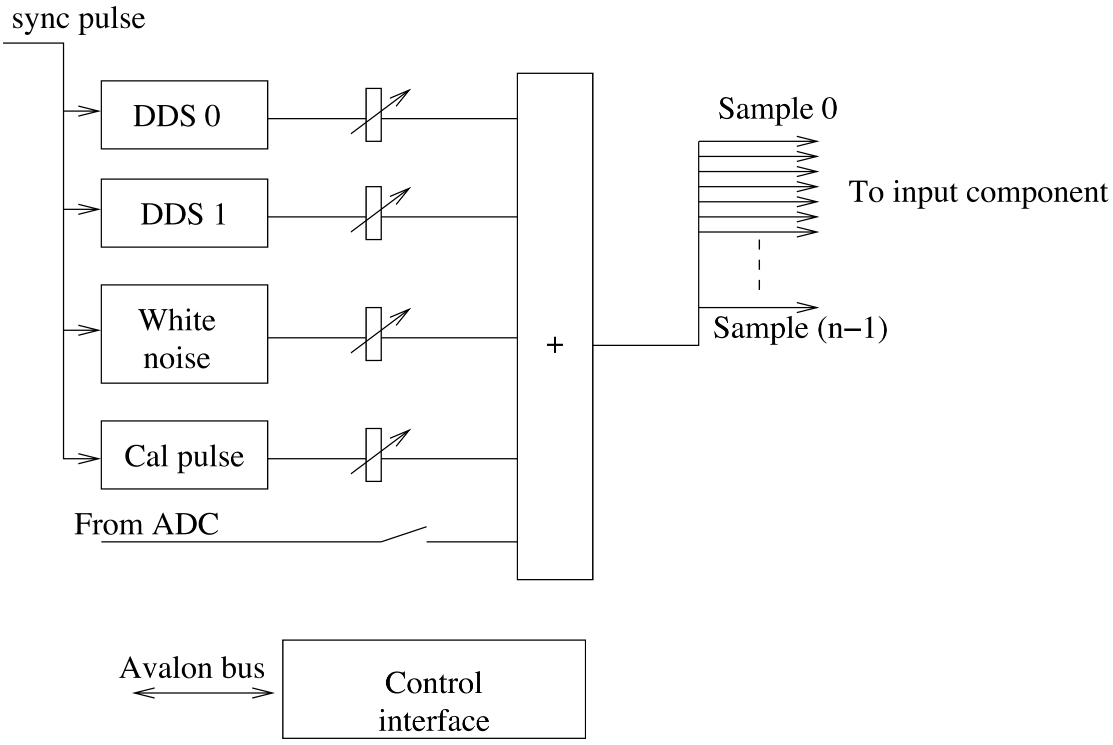


Figure 5: Test signal generator

The most efficient channelization scheme for equispaced frequency channels is the polyphase filterbank (PFB) [5]. It is composed by a series of short FIR filters that basically convolve the input sample sequence with an appropriate window function, and a conventional N-point FFT, that transposes the FIR response of the windowing function to the N frequency channels of the FFT. Output channels may overlap by an arbitrary amount. In our case we choose to use a 50% overlap, i.e. the channel nominal width is twice the channel spacing. This allows the shaping filter specifications to be very relaxed, using a minimum of resources for the band shape filter, and the overlap region allows for a large freedom in successive processing. E.g. in the digital receiver application the channelized signal is further filtered to a band of typically 1/4 that of the coarse filterbank. If the effective overlap among adjacent coarse channels is 25%, the narrow channel can be arbitrarily positioned in any point of the input band (Figure 6).

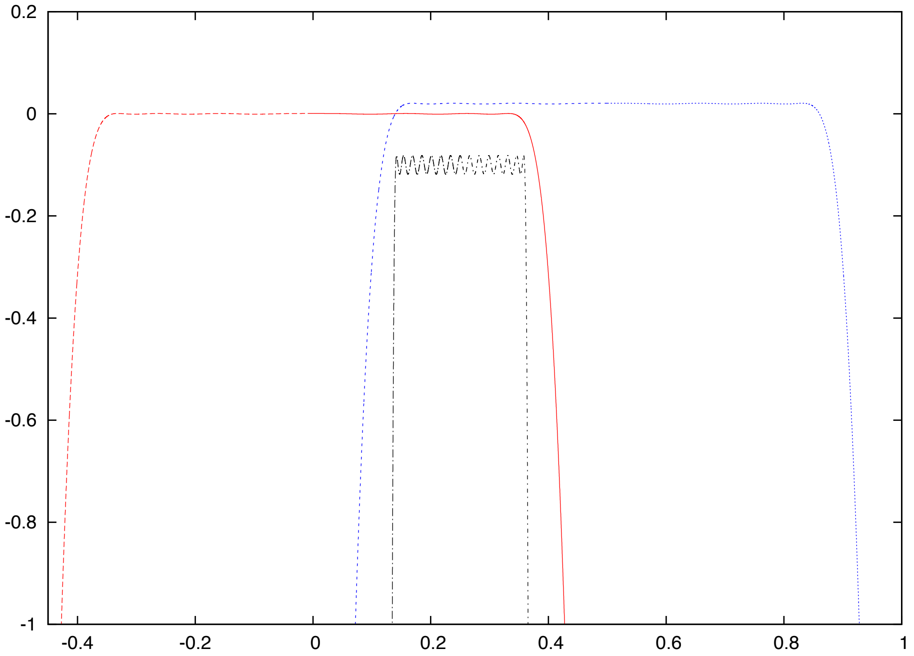


Figure 6: Bands for two consecutive coarse channels, and for a 1/4 band fine channel. Horizontal scale in channel width, vertical scale in dB. Overlap allows the fine channel to be arbitrarily placed in the input band

## 6.1 Polyphase filter

The structure for a *N* input double rate polyphase filter structure is shown in Fig. 7 (left). Each filter computes the convolution of the input data stream for samples *i* and *i* + *N*i (Fig. 7 right). FFT uses a parallel, decimation in time architecture, which requires the input samples to be presented in bit reversed order. The bit reversal can be performed naturally using a hierarchic time demultiplexing of high seed samples. As the FFT output has a rate double with respect to the natural decimated input sample rate, the phase of the odd output channels rotates by π at every output sample. To correct this, FFT inputs *i* and *i* + *N* are exchanged at each clock cycle.

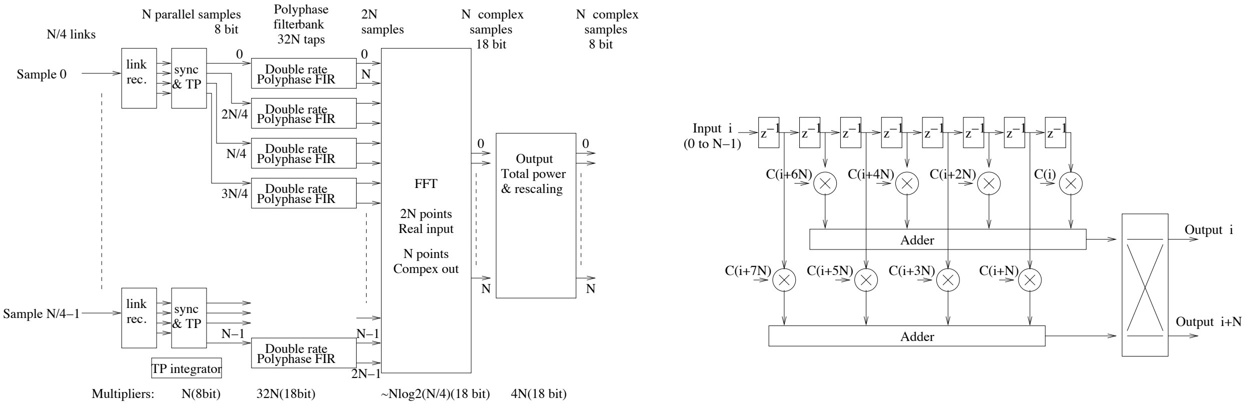


Figure7: Generic polyphase filterbank structure (left) and filter element of a double rate (50% overlap) PFB(right)

## 6.2 FFT block

The filterbank receives real valued samples at the input. The FFT can then be optimized for real input, halving the number of required multipliers by exploiting the conjugate relation $y(N - f) = y(f)^*$, and computing only half the outputs at each FFT stage. An example for *N*=16 is shown in Fig. 8. The required number nm of (real) multipliers for a *2N* FFT block (*N* real outputs) is slightly less than *2N* $\log_2$ (*N*/2). The twiddle factor implementation automatically uses no multiplies for exp(iπ/2), two multipliers for exp(iπ/4), and four multiplers for the general case.

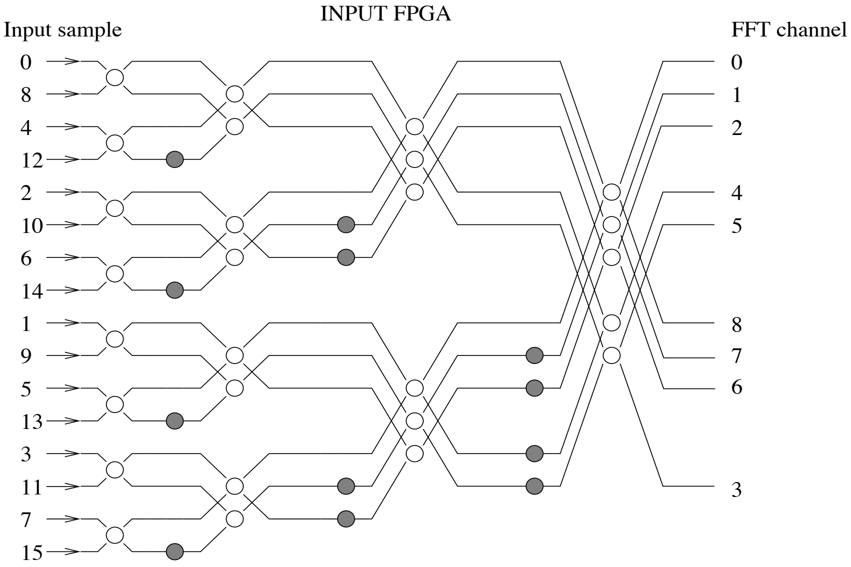


Figure 8: Real valued input 16 channel FFT

The FFT size, *2N*, depends on the input and output sample rates. For the baseline digital filter application the sample rate is 8 GHz and the internal sample frequency is 250 MHz, resulting in a convenient, power-of-2 ratio *N* = 32, and a FFT size of 64. The FFT block requires 204 real multipliers. The filter, for the band-shape of Fig. 6, requires 512 taps, and 512 real multipliers.

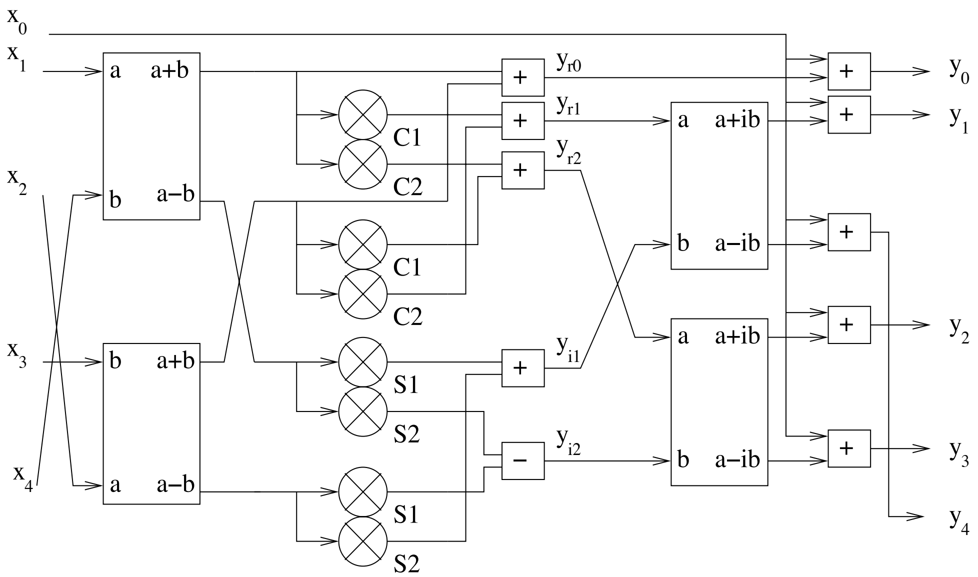


Figure 9: Base 5 butterfly block. *C1 = cos(2/5π), C2 = cos(4/5π)* and *S1 ,S2* are the corresponding sin

The limitation of a base 2 FFT, however, often imposes inefficient output sample rates. A base 5 butterfly module has been developed [9] (Fig. 9), allowing for decimal FFT sizes. An example of a 40 input FFT, appropriate for a clock decimation factor of 20, requiring a total of 170 multipliers, is shown in Fig. 10.

The output channels must be rescaled, to compensate differences in signal level across the input band, and re-quantized, to reduce the complexity of the following stages. A total power detector is therefore included for each output channel, together with a programmable re-quantization to 8-12 bit resolution.

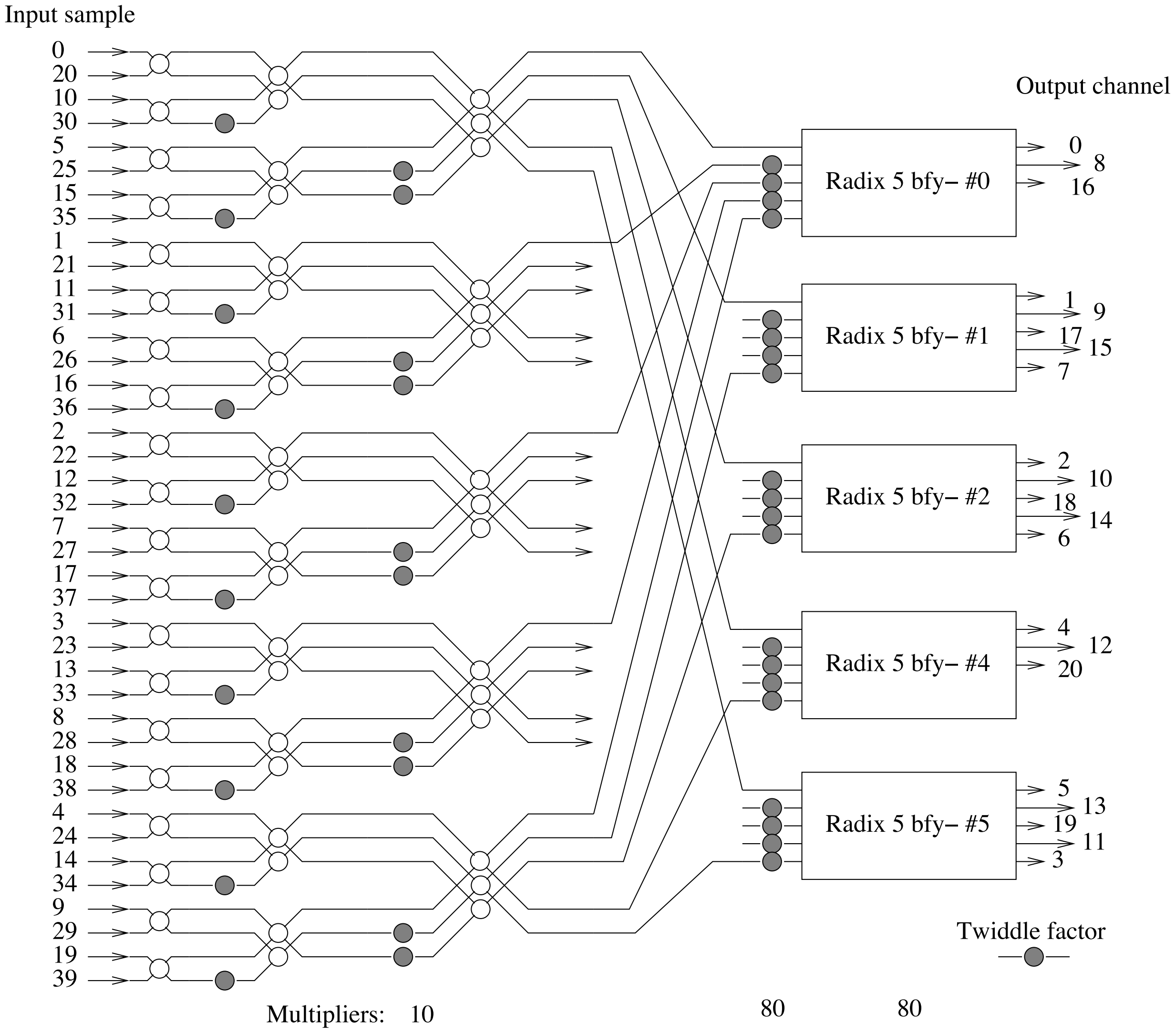


Figure 10: Real valued input 40 channel FFT

# 7 Second stage channelizer

The first stage channelizer produces coarse channels with a band of 250 MHz. Usually VLBI correlators work on much smaller channel bands, up to 64 MHz and possibly 128 MHz. Although correlators with complex signal representation are becoming increasingly common, most existing VLBI receivers use real valued samples. For this reason it is necessary to divide the coarse *channels* into one or more narrow channels, with real valued samples.

Three options are possible:

- If the number of required channels is small, one can implement a conventional digital down-conversion structure, composed of a complex local oscillator and mixer, a complex (real valued) decimating FIR filter, and a complex to real up-conversion stage. This allows the maximum flexibility for channel width and spacing: channels may overlap, and *zooming modes*, in which the same spectral region is observed with different spectral resolution, are available (Fig. 11a).
- If the channels are equispaced, with a bandwidth equal to the spacing, and a small *dead region* can be allowed between the channels, a conventional polyphase filter can be used (Fig. 11b). Channel bandwidth/spacing is fixed, requiring a personality reload to change FFT length, but the origin of the fine channel group in any given coarse channel can be shifted, and individual channels may be skipped.
- If the channels are equispaced with a bandwidth different from the channel separation, an over-sampling structure can be used. This allows a small overlap among adjacent channels, avoiding the *holes* in the spectrum associated with finite filter edges. Again, the bandwidth and overlap is fixed in the design, but each fine channel group can be tuned in block, and individual channels can be skipped. (Fig. 11c).

These options are described in detail in the following sections.

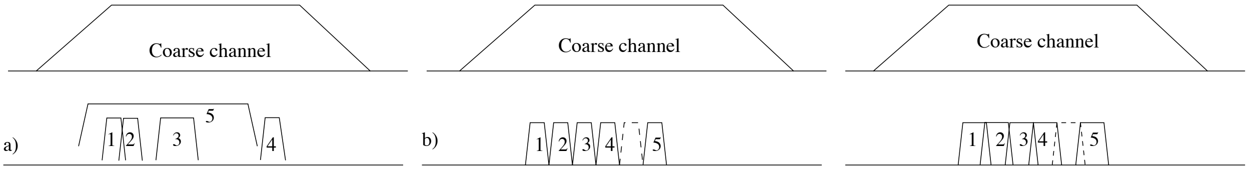


Figure11: Options for the second stage filter:(a) independent DBBCs; (b) conventional polyphase; (c) Oversampling polyphase.

## 7.1 Digital baseband converter

This option allows for maximum flexibility, but only a limited number of independent channels can be implemented. Considering the resources in a Arria10 FPGA, and the required multipliers used by the coarse channelizer, about 48 channels can be fitted in a single node. Structure of the filter is shown in Fig. 12. An input selector allows the channel to be fed from one of four coarse channels, i.e. to be tuned across about 1/8 of the total input bandwidth. The channel centre frequency is selected using a local oscillator and a complex mixer. By synchronizing the local oscillator to an external 0.1 ms timing signal it is possible to tune the local oscillator to an exact multiple of 10 kHz, according to the VLBI standard frequency scheme.

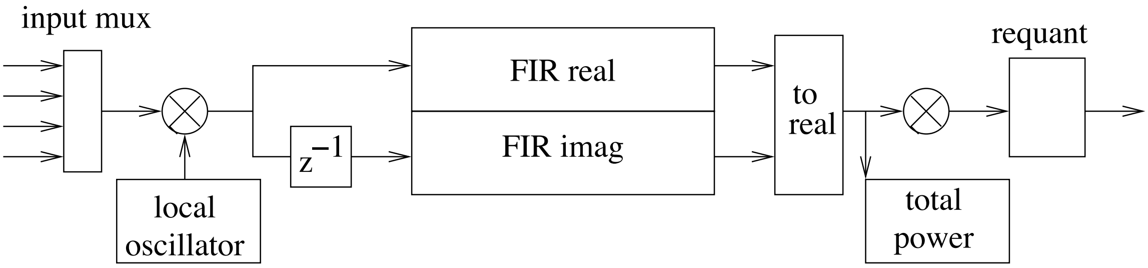


Figure 12: Digital baseband converter structure

The decimating filter uses tap recirculation and symmetry to reduce the number of required multipliers. For a band decimation by a factor *D* the filter uses *32D* taps, symmetric, and a total of 32$D_m$ tap values are stored to implement all filters up to a maximum decimation of $D_m$. 8 20kb memory blocks are used to store tap values for decimations from 2 (128 MHz band) to 256 (0.5 MHz).

With this architecture a total of 32 real multipliers are used to obtain a self-similar band-shape, with a usable band of 88% the Nyquist band. Filter coefficients are computed using the Remez algorithm, for an equiripple design. For higher decimations, with more than 500 taps per filter, the technique described in [8] has been used. Analytic, closed form filter coefficients using a *sin(t)/t* and a Dolph-Chebyshew tapering results in slightly worse performance, but filter computation is much simpler, and can be performed automatically in the VHDL code. Parameters for both filters are listed in Tab. 3 and plotted in Fig. 13.

| Filter type | Bandwidth | Pass-band | Stop-band |
|---|---|---|---|
| | % Nyquist | DB (p-p) | dB |
| Remez equiripple | 88% | 0.035 | 80 |
| Dopl-Chebyshew | 86% | 0.015 | 72 |

Table 3: Digital baseband converter filter parameters

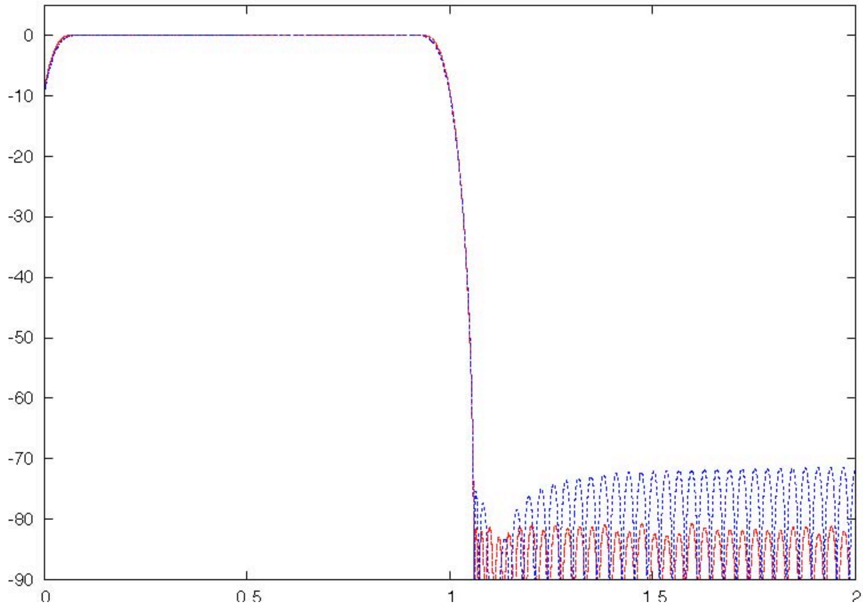


Figure13: Digital baseband converter example bandwidth, using an equiripple Remez (red) and a Dolph-Chebyshew tapered sin(*x)/x* (blue) design.

The FIR filters produce decimated and filtered complex samples, at a sample rate equal to the bandwidth. A real valued data stream at twice the sample rate is obtained by shifting the imaginary portion by half decimated sample, and by upconverting the result by half band.

## 7.2 Polyphase 2nd stage channelizer

When the application requires uniformly spaced channels with constant channel width, a polyphase filterbank is a more efficient solution with respect to individual channels. The general structure of a PFB is shown in Fig. 14, and is similar to that of the first stage channelizer, with some differences:

- both the input filter and the FFT operate on serial data, instead of parallel data,
- the input data samples are complex, while the output samples are real,
- the origin of the FFT channelization can be shifted by ±1/2 fine channel, using a local oscillator and mixer,
- the FFT uses a radix-4 algorithm, thus allowing 4 channels to be processed in parallel.,
- as only the central portion of the coarse channel is used, only half of the fine channels are processed.

Thus each base block has 4 inputs and 2 outputs, with the fine channels output serially.

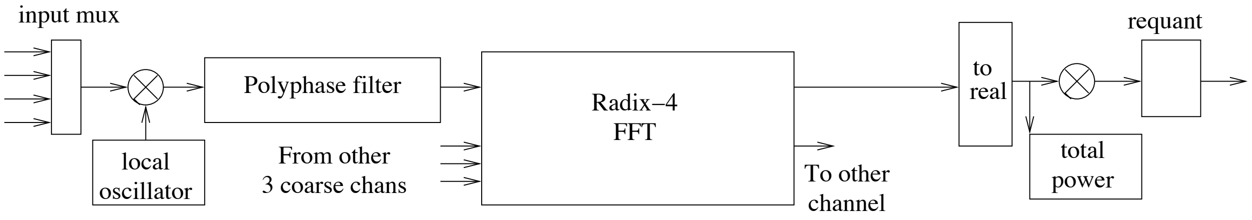


Figure 14: Polyphase filterbank block structure

The channel width and spacing are fixed for any particular design, but can be chosen with some degree of flexibility to suit different applications. Most VLBI continuum observations do not require a continuous frequency coverage, and a non overlapping channelization scheme can be adopted. In this case the channel shaping filter is chosen to reject the adjacent channels to a high level, at the expense of a ”hole” between channels.

For applications where a continuous frequency coverage is required, the filter portion is modified in order to slightly increase the output data rate, while maintaining the same channel spacing. This allows for a small overlap between adjacent channels that cover the ”hole” used to prevent aliasing. Feasible oversampling values are listed in table 4, together with the required filter resources, i.e. multipliers in the FIR section, increase in clock frequency.

| $O_f$ | Cutoff freq. | Stop freq. | N. of taps | N. of mult. | Data rate incr. (%) |
|---|---|---|---|---|---|
| 32/26 | 0.5/*N* | 0.7308/*N* | 12*N* | 24 | 23.1 |
| 32/27 | 0.5/*N* | 0.6852/*N* | 15*N* | 30 | 18.5 |
| 32/28 | 0.5/*N* | 0.6428/*N* | 18*N* | 36 | 14.3 |

Table 4: Oversampling factor and related polyphase filter characteristics for an N channel PFB

To implement the oversampling a dual clock structure is used (Fig. 15). The input samples are placed in a FIFO memory using the original sample rate, and extracted from the FIFO in longer blocks at the oversampled sample rate. Successive stages of the filter delay rewind the sample stream at the beginning of each FFT frame, to provide the required overlap in the time sequence. A phase rotation is needed at the FFT output to compensate for the fractional overlap between frames [5].

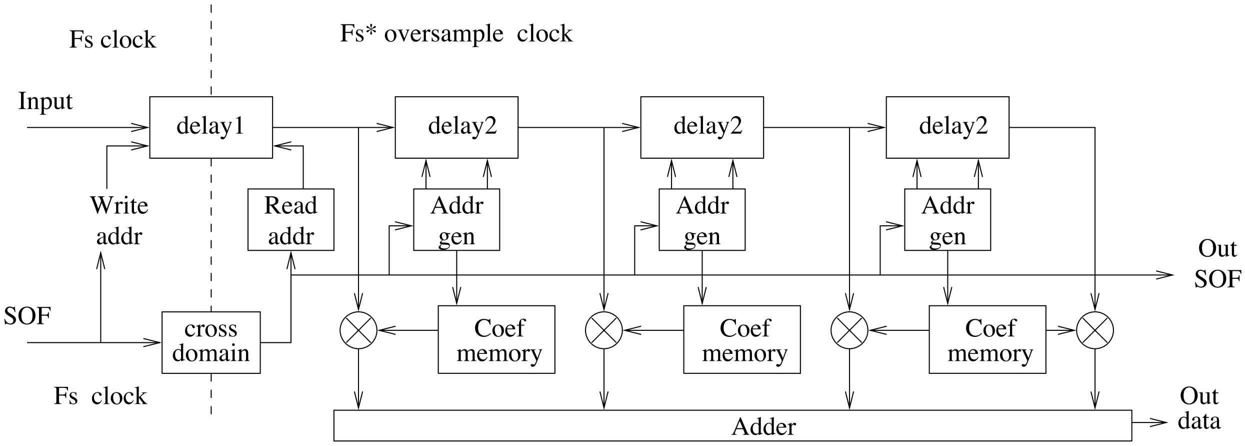


Figure 15: Serial oversampling polyphase filter

The FFT engine operates in parallel on 4 independent coarse channels, for a total processed bandwidth of ≈ 500MHz. The structure of the channel ordering is such that at any given moment only two outputs present data relative to the inner (good) portion of each overlapping coarse channel (see [10] for details). Therefore the output data rate from the FFT engine is half the input rate, or each FFT block will produce two output streams for a total bandwidth of 512 MHz.

FFT size depends on the required output channel width. For VLBI applications typical FFT sizes range from 4 to 64 channels, i.e. a 512 MHz segment of data produce from 8 to 128 fine channels.

Each block of 4 coarse channels requires about 170 real multipliers, mainly for the filter section. The total number of multiplers for 32 channels is thus 1500, less if not all coarse channels are processed.

Data is then stored in a corner turning memory and data buffer. If the number of fine channels is not excessive, local (internal) memory can be used. Arria10 FPGAs memory blocks have 20 kbit size, i.e. 8 blocks are required for a 8KB (jumbo frame) dual buffering storage. With about500blocksavailablefor buffering, 32 VDIF channels can be implemented. The number of output channels can be much larger, because the VDIF format allows several (up to the packet length) logical channels to be transmitted in one frame.

# 8 VDIF / UDP interface

The VDIF standard [11] has been developed for the storage and interchange of VLBI data. A VDIF packet is composed of a fixed size header and a frame of sequential samples. Samples may refer to a number (constrained to a power of 2) of synchronous samples, sampled at a constant data rate. Sample size can range from 1 to 32 bits per sample, trading precision for bandwidth.

A block schematics of the formatter and UDP/IP interface is shown in Figure 16. In this application sample size is constrained to a jumbo frame Ethernet packet. Each VDIF frame is encapsulated in a standard packet containing only a 32 bit sequential frame number and the payload, inside a standard UDP Ethernet packet. This formatting, up to the UDP packet, is performed directly inside the second stage channelizer, thus allowing different destination addresses for each channel, and stored in a double buffer memory structure. A stream arbiter, developed as part of the standard Uniboard stream library, selects the first ready packet and queues it to an Ethernet UDP/IP standard core.

The UDP/IP core provides the source address and UDP port, and the IP header and checksum, and implements the ARP protocol layer. The core is part of the standard Uniboard library. The formatted Ethernet packet is then transmitted using a commercial Ethernet MAC IP.

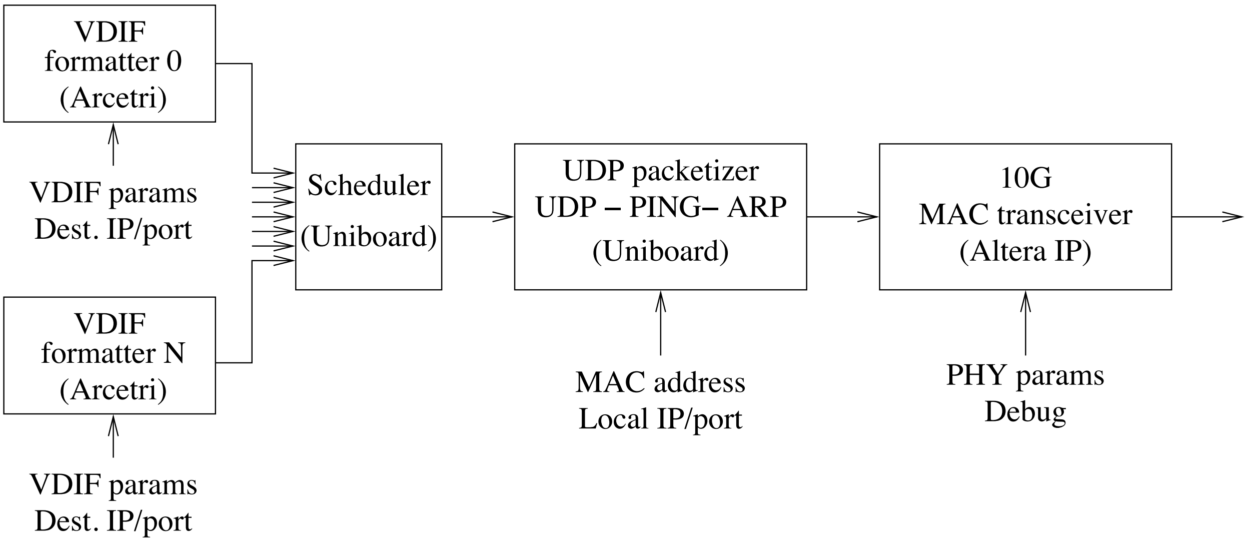

VDIF formatter 0 (Arcetri)
VDIF params Dest. IP/port
VDIF formatter N (Arcetri)
VDIF params Dest. IP/port
Scheduler (Uniboard)
UDP packetizer UDP – PING– ARP (Uniboard)
MAC address Local IP/port
10G MAC transceiver (Altera IP)
PHY params Debug

Figure 16: Concept oft he VDIF/Ethernet interface.

## Copyright