A 512 Mhz Polyphase Filterbank with overlapping bands

A.Melis[1], G.Comoretto[2]

[1]INAF - Osservatorio Astronomico di Cagliari

[2]INAF - Osservatorio Astrofisico di Arcetri

Arcetri Technical Report N° 1/2011

Abstract

The polyphase filterbank(PFB) algorithm is an efficient way to implement a uniformly distributed multi-channel filterbank using a Fast Fourier Transform, in order to reduce the necessary clock frequency to process the acquired samples into an FPGA. The simplest implementation of a PFB produces adjacent bands with small holes between the sub-bands; to overcome this problem, a PFB with overlapping bands is one of the best solutions.

In this report a PFB with a particular overlapping channel algorithm is described. The system has been implemented into the dBBC system, based on a Xilinx FPGA Virtex 5 LX220.

1 Introduction

Radioastronomic signals have usually a very large instantaneous bandwidth, up to several GHz, while currently available digital signal processing hardware, especially Field Programmable Gate Array(FPGA), have maximum clock rates of the order of a few hundred MHz. Therefore a multiplexing technique of some sort is necessary to process the acquired data at a reduced clock frequency. The most efficient way to perform this function is to use a filterbank, in order to split the incoming wideband signal into separate sub-bands of reduced bandwidth and sample rate.

A very efficient way to perform this function is the polyphase filterbank (PFB), a multi-rate filter structure followed to a Fast Fourier Transform (FFT) engine[1]. The filter is a low pass finite impulse response (FIR), and its bandshape is translated in frequency by the FFT algorithm. In its simplest form it implements a filterbank with adjacent bands. Due to the finite transition region at the end of the FIR passband, it produces small holes between the sub-channels, as shown in figure 1:

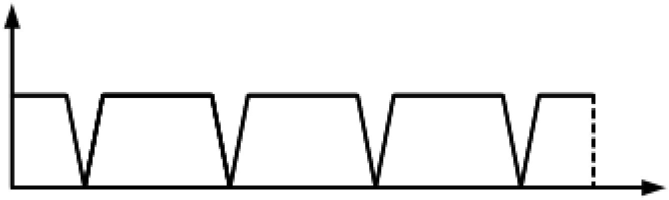


Figure 1: Non-Overlapping filterbank response

This approach has already been used for a non-overlapping filterbank for VLBI observations, described in a previous report[2]. We address to this report for the mathematical treatment of PFB.

To overcome the gap problem, a PFB with overlapping bands has been designed; the system, intended mainly for single dish spectroscopy, has been implemented into a Xilinx FPGA Virtex 5 LX220.

2 Algorithm description

As mentioned above, a PFB is basically a N-point FFT performed by a narrow polyphase filter followed by a small FFT engine as shown in figure 2. The signal is deserialized into N streams at 1/N the original data rate, and each stream is preprocessed by a short filter leg. Tap coefficents of each filter leg correspond to the taps of the original low-pass filter, spaced by N taps: first leg processes taps (0, N, 2N, …) second leg taps (1, N+1, 2N+1...) and so on.

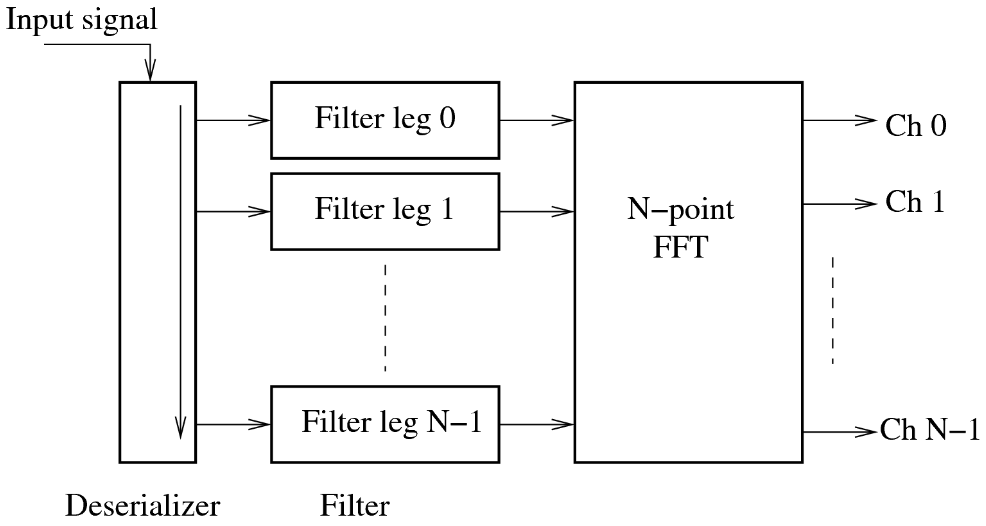


Figure 2: Typical block diagram of a PFB

The original filter shape is translated in frequency by the N-point FFT engine producing N adjacent replicas, as shown in figure 1. For a real valued signal (as in our case), only outputs 0 to N/2 of the FFT engine are used (N/2 + 1 in total), the others produce just the complex conjugate (frequency mirrored) version of the same bands.

To eliminate the gaps between adjacent sub-bands an ideal rectangular filter would be required, with infinite sharpness. To reduce these gaps to a reasonable value a filter with an impractically large size would be needed. A more practical solution is to double the size of the FFT, and thus the sub-channel spacing, using the same sub-channel width. Then only the central half of each sub-channel is actually used, leaving a large space for the transition band.

This solution has been adopted, with a sub-channel width of 128 MHz (-64 MHz to +64 MHz) and a channel-to-channel spacing of 64 MHz. The prototype filter shape is shown in figure 3, with the horizontal scale in MHz:

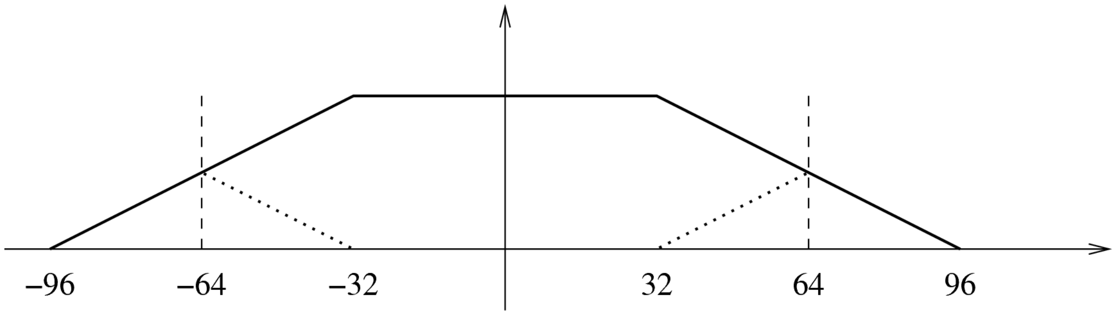


Figure 3: Filter shape implemented. Useful band is half the total band.

The useful band is from -32 to +32 MHz, with the transition band extending up to 96 MHz and aliased back in the 32 to 64 MHz region. Having such a large transition band allows for a smaller filter size, and for better performance.

In our case the initial deserialization factor is 8 (from 1024 to 128 Msample/s), and N=16. The resulting output sub-channels are depicted in fig. 4

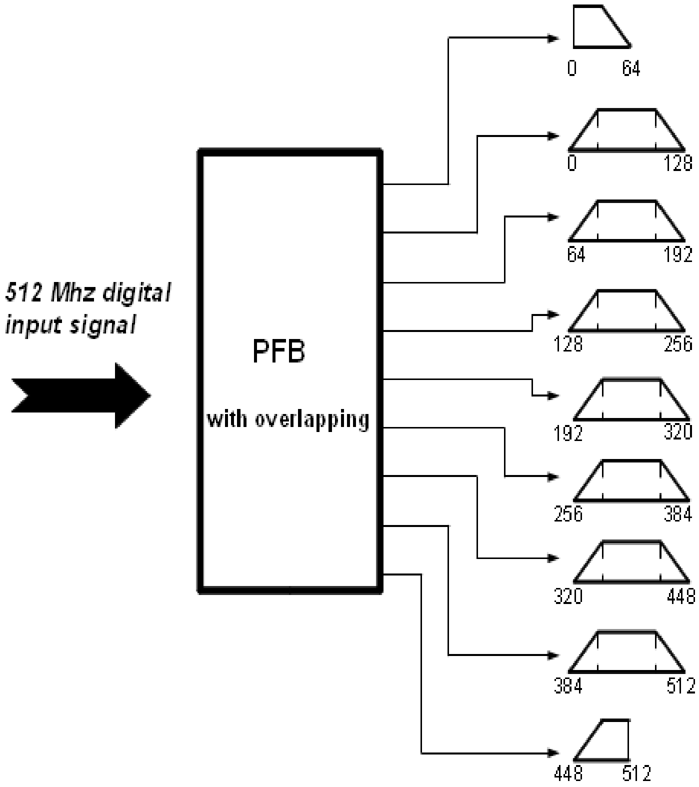


Figure 4: 512 Mhz input signal decomposition

First and last sub-channels have real representation, with the same signal on positive and negative frequencies, the others use a complex representation. Each sub-band is baseband converted; by choosing one of them, the flat band of interest always extend from -32 to 32 MHz. A smaller region can be selected by mixing the signal with a local oscillator of appropriate frequency and filtering the resulting signal by a programmable filter.

3 FPGA implementation

The chip used for this application is a Xilinx XC5VLX220 FPGA, that is contained into the CORE2 boards of the Digital Base Band Converter[3]. The code has entirely been written in the VHDL programming language avoiding any specific dependence on the Xilinx hardware so we were able to render the code completely portable to different FPGA families, even not Xilinx. The design functionality has been simulated using Active-HDL and Modelsim tools, while the code has been converted to a Xilinx netlist using the Synplify synthesis program and finally translated to a physical design in the target chip by using the Xilinx ISE 12 tool.
Figure 5 shows the signals resulting from a swept sinusoidal input sent to the PFB:

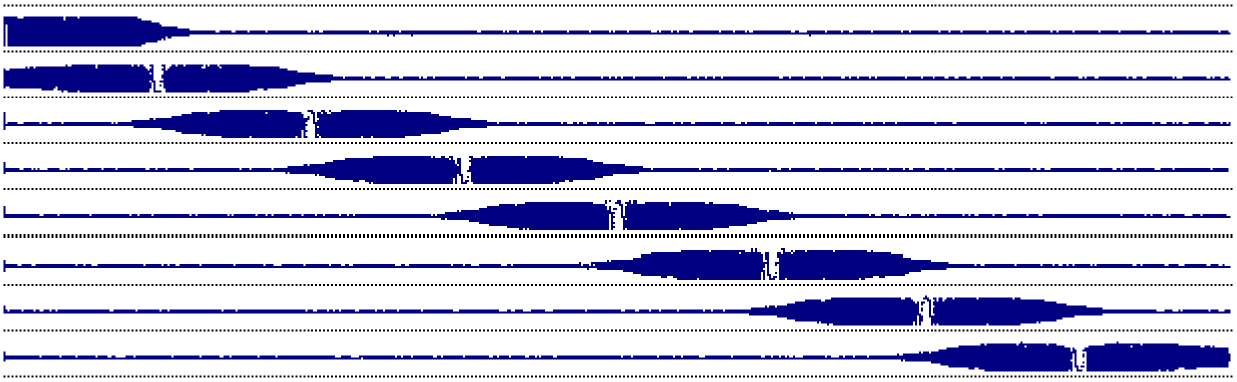


Figure 5: Sub-bands resulting from a swept sinusoidal signal (simulation with Active-HDL)

3.1 Polyphase Filter

The filter implemented is a symmetric low pass FIR filter. It has been synthesized using the well known Misell algorithm.

As shown in figure 3, a very extended transition band has been specified. Therefore we could exploit the available resource to achieve a better passband flatness and insulation amongst different channels. The adopted filter has 64 symmetric taps, an in-band flatness of 0.07 dB p-p, and an out-of-band rejection of 87 dB.

In particular, in order to achieve an insulation of almost 90 dB 18 bit tap coefficients have been used.

Figure 6 shows the filter's coefficients graphically:

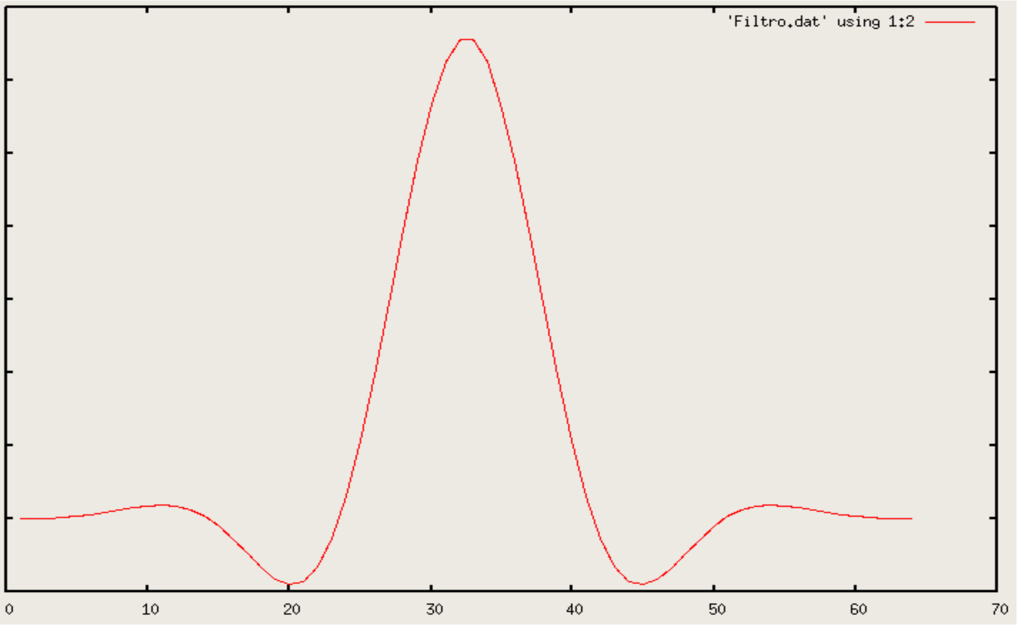


Figure 6: 64 filter's coefficients

3.2 FFT

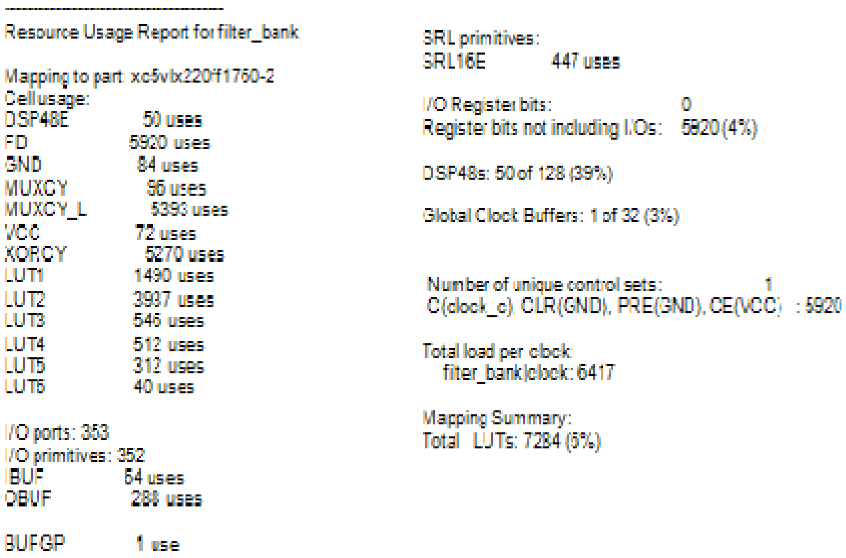
The FFT engine processes real data stream from the eight legs of the polyphase filter and provides sixteen complex data stream of which the latest eight represent the complex conjugate of the first eight, thus we discard them.

The design employed is a division in time FFT processor composed of four stages. The first has real inputs and no multipliers are used because twiddle factors involved are just ± 1. Outputs also are real. The second stage provides complex outputs, but no multipliers are used because twiddle factors involved are ± i or ± 1. The third and fourth stage perform the final calculations, twiddle factors are computed employing a dedicated VHDL package.

Finally, a correction is needed because some sub-bands are reversed due to the double length of the FFT processor. Therefore an alternate multiplications for +1 and -1 is done for the odd-numbered sub-channels to reverse back the frequency scale.

3.3 Results

In figure 7 Synplify Premier synthesis results are shown:



```
---------------------------------------
Resource Usage Report for filter_bank

Mapping to part: xc5vlx220ff1760-2
Cell usage:
DSP48E          50 uses
FD            5920 uses
GND             84 uses
MUXCY           96 uses
MUXCY_L       5393 uses
VCC             72 uses
XORCY         5270 uses
LUT1          1490 uses
LUT2          3937 uses
LUT3           546 uses
LUT4           512 uses
LUT5           312 uses
LUT6            40 uses

I/O ports: 353
I/O primitives: 352
IBUF            64 uses
OBUF           288 uses

BUFGP            1 use
```

```
SRL primitives:
SRL16E         447 uses

I/O Register bits:                   0
Register bits not including I/Os:    5920 (4%)

DSP48s: 50 of 128 (39%)

Global Clock Buffers: 1 of 32 (3%)


Number of unique control sets:              1
C(clock_c), CLR(GND), PRE(GND), CE(VCC)    : 5920

Total load per clock:
   fiter_bank|clock: 6417

Mapping Summary:
Total  LUTs: 7284 (5%)
```

Figure 7: synthesis report of the PFB

No dedicated multipliers are used for the polyphase filter because, with a 128 MHz frequency clock and fixed coefficients, multiplication can be calculated with logic resources.

The FFT engine needs 50 DSP48E (hard multiplier) blocks, a little less than 40% of the 128 available DSP, therefore two PFB can be implemented into a single chip. It is possible to further optimize the design, reordering the FFT process in order to eliminate computations required for the unused outputs and removing multipliers with unitary twiddle factors.

The PFB has been implemented in the general framework of the dBBC control structure[4], and tested using the digital-to-analog converter present in the system, confirming the performance achieved in the simulation. It is now available as a general purpose module, that can be integrated in larger designs.

4 Conclusions and future works

In this paper we described a 512 MHz polyphase filterbank with overlapping bands to overcome the problem of the holes between adjacent sub-band.

The system has been implemented on a Xilinx Virtex 5 LX220 FPGA, that is contained into the Core2 boards of the Digital Base Band Converter(DBBC).

We are developing a 1 GHz PFB with the same configuration. In this case 128 DSP are not enough and an optimization of the DSP resource usage is necessary.

Finally, we are developing, for the DBBC, a particular spectrometer composed of two-stage PFB in which the first is the system described in this paper and the second is a serial PFB. The overall system will be part of a scanning FFT spectrometer in which the flat parts of each band are achieved automatically avoiding waste of resource.

References

[1] Harris, Fredric J. *Multirate signal processing for communication systems.* Upper Saddle River, NJ: Prentice Hall PTR. ISBN 0-13-146511-2 (2004)

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

[3] G. Tuccari: *DBBC – a Wideband Digital Base Band Converter.* IVS 2004 General Meeting proc., 234 (2004) ftp://ivscc.gsfc.nasa.gov/pub/general-meeting/2004/pdf/tuccari3.pdf

[4] G. Comoretto, G. Tuccari: *Reference design for the Digital BBC Architecture*. Arcetri Technical Report 2-2008

Table of contents