The Medicina IRA-SKA Engineering Group

VIIP:
a PCI programmable board.

G. Bianchi
L. Zoni
S. Montebugnoli

IRA N. 374/05

Abstract

This document treats the development of a hardware for acquisitions and data processing, programmable, employable in radioastronomy and in other applications. The board has high speed, high resolution analog to digital converters (105MSPS and 14 bit) and has 3 million gate Fpga with 600 I/O pins fully configurable. It allows a Pci interface to connect the board in a Pc with 32/64 bit, up to 66 MHz bus Pci. Use of this board for back-end used in radio astronomy, allows to acquire up to 100 MHz bandwidth and decompose it in a number of channels depending of the datapath width. If datapath is an 8 bit word, the number of channels is 8k and therefore very high resolution is reached.

Introduction

Radioastronomy always needs new innovative technologies to increase the sensibility of the antennas for the research of distant radio sources. This has led to the development of the VIIP board, which implements a digitalization of the signal, a polyphase filter bank and a Pci bus interface. VIIP board is the first block of a radioastronomical digital back end that will allow the testing of new technologies developed for the S.K.A.. Particularly, it will be able to perform low resolution spectrum analysis, polarimetry, Pulsar research and, with the addition of a Corner Turn Memory (CTM) and a correlator after VIIP board, it will be possible to create radio maps and to perform very high resolution spectrum analysis. Beamforming tests and RFI mitigation algorithms will be able to be implemented. The costs are not high and they depend, essentially, on the Field Programmable Gate Array (FPGA), the core of the card. The FPGAs find, day-by-day, new applications that reduce their costs and increase their availability. Besides, for the same reasons, the FPGAs follow technological development of the integrated electronics, inheriting their high integration and the speed. This allows the analysis of wide bandwidths and the possibility of implementing a high number of functionality on the same hardware. The VIIP board uses a Xilinx FPGA, from Virtex-II family. In particular, the component used is the XC2V3000, composed by 3.000.000 gates, working up to 420MHz. In the proposed application, FPGA implements a Polyphase Filter Bank (PFB) that divides the input band in several channels, allowing extremely higher performances in comparison to a classical filter bank implemented through FFT: adjacent channels rejection can be 80 dB, while 13 dB is the one of a classical FFT filter bank. In this case, a strong interference can be isolated easily, simply erasing the channels that contain the interference itself.

Features

ADC resolution: 14 bit.
Max bandwidth: 100MHz (50MHz every input).
Pci interface to connect the board in a Pc with 32/64 bit up to 66 MHz bus Pci.
3 million gates available on the programmable logic.

> Implementation example: Polyphase filter bank up to 8k channels whit 8+8 bit of datapath (I+Q) and up to 16k channels whit 4+4 bit of datapath.

Board description

The VIIP board is a 10 layers board and can be inserted in a Pci slot. It has two inputs for analog signals and one input for the clock. It is composed by two ADCs, which digitize the signals coming from analog inputs, one FPGA, which implements a polyphase filter bank, and a bridge for VIIP bus Pci connection. VIIP board can work in two different modes, depending on how the FPGA is programmed:

1. It can receive two analog input signals from two different IF line (figure 1).
2. It can receive I and Q components of a quadrature demodulated analog input signal (figure 2). In this case a direct conversion is necessary before VIIP board.

In the first case a DDC (Digital Down Converter) performing a digital quadrature demodulation (I and Q) is implemented on FPGA, before PFB. In the second case, this operation is made analogically before VIIP board.

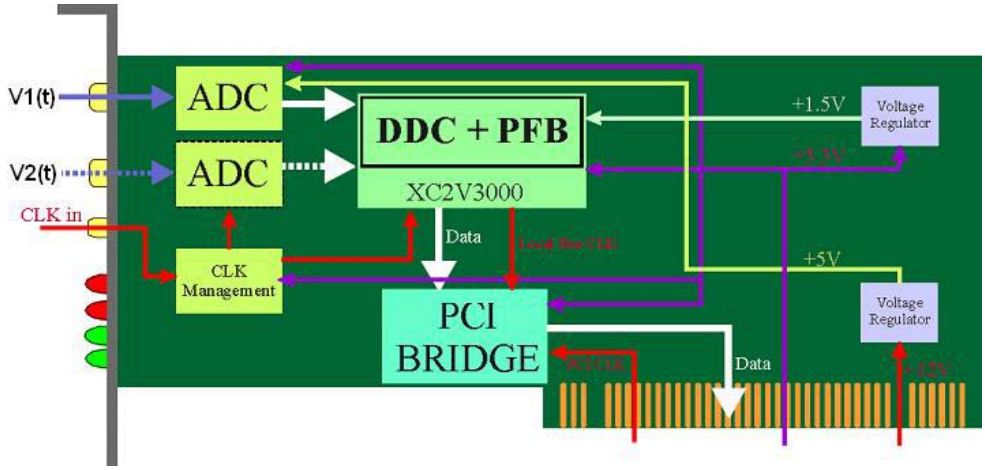

Fig. 1: VIIP configured whit two independents IF input.

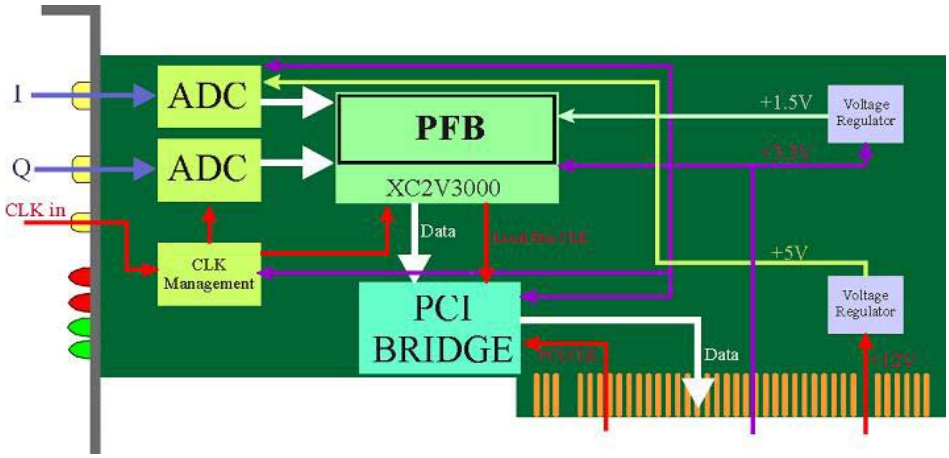

Fig. 2: VIIP configured with in quadrature input signals.

Due to the fact that the first option operates on two real bandwidths, the sampling frequency has twice the value with respect the mode 2 (having the same bandwidth per channel). Therefore the dataflow speed is doubled than the second configuration. The PCI datarate is obviously doubled in this case. Also, we have a high area occupation in the FPGA due to the necessity of the implementation of the DDC. In the second case the signal is acquired by only one receiver, but the clock works with half sampling frequency because it sample a complex bandwidth, consequently the bit rate is slower and the DDC is not needed.

Clock signal can be provided in two modes:

1. Sinusoidal waveform. In this case, by a squarer and a PLL assembler on VIIP board, it is possible to have a clock up to 100MHz.
2. PECL square waveform (f_{max}=100MHz).

An other input on the VIIP board is a PPS (Pulse Per Second) synchronism signal: it provides a temporal mark every second. PPS is provided by PECL logic level.

In figure 3 it is reported the VIIP board.

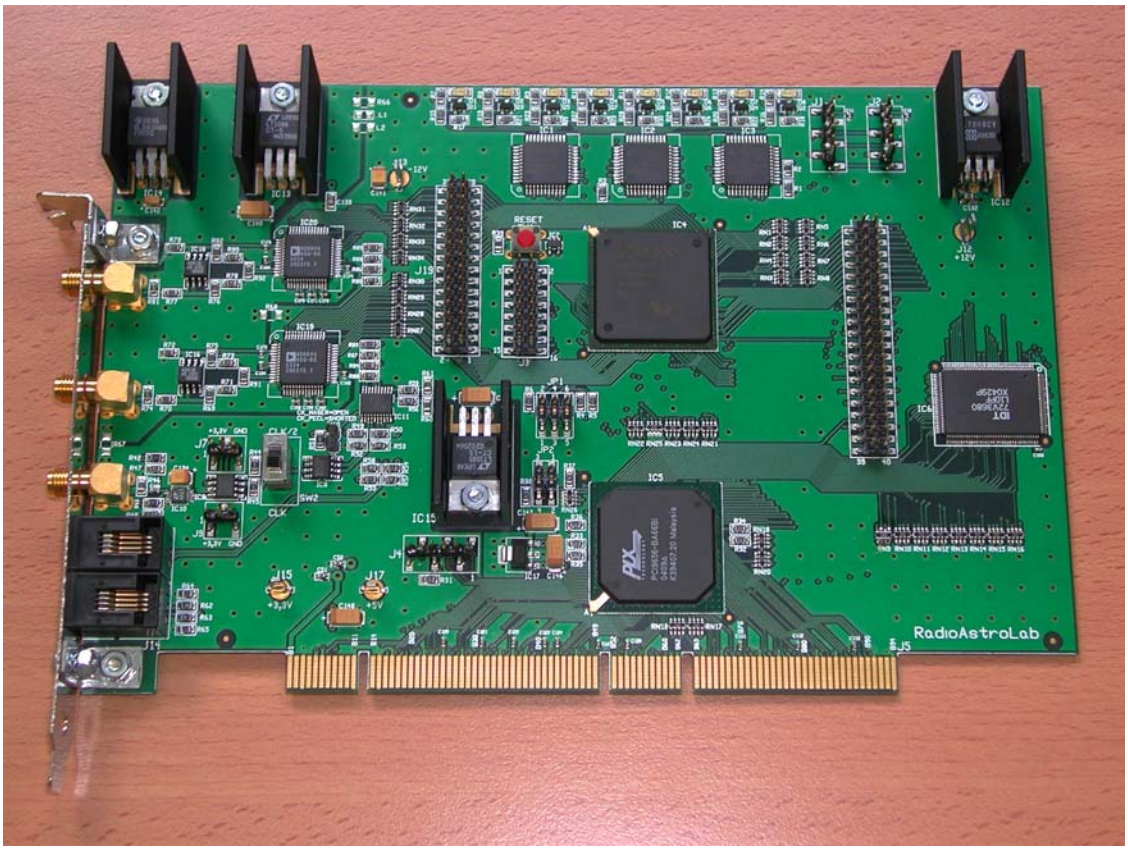

Fig. 3: VIIP board.

Results

Initially, we have tested power supplies. We did this test on a board with only voltage regulators mounted. Without BGA components installed we tested supply levels on all ballpads and we verified that all levels are correct.
The connections test was made by the PCB manufacturers so we have a right PCB.
Ourselves made an additional test on the FPGA configuration path (PROMS and FPGA configuration pins) and we verified all ok.
After all power supply tests was made, we have supplied the VIIP board. We have verified the case chip's temperature (voltage regulators, etc.) and, for security reasons, we decided that ADC's ones require a forced cooling. We are planning to mount a little cooling fan on future release of the board.
We tested clock distribution, analog part and analog to digital conversion module: following tests have provided good results.
The board test proceeded with the FPGA programming (a fundamental item!). So we have connected the Parallel Cable III of Xilinx and verified that Impact, the programming tool, recognized the configuring chain. We programmed the FPGA with a dummy application than switches board leds on. We verified that all leds was inverted. After a little manual operation all seemed to be ok.
We tested the FIFO through the FPGA; we implemented a finite state machine that generates all control signals of FIFO and measured voltage levels on inputs and outputs pins. We supplied a 30 MHz clock to the output and input clock pins of FIFO and verified that there're no speed problems.
Tests on VIIP board are currently in progress.

Appendix:

Polyphase Filter Bank theory.

Polyphase filter bank is a system with some passband filter. They have same input but different output and every output corresponds to a different transfer function. Every output of the bank corresponds to one channel. Input and output are in the time domain.

The simplest way to implement a digital filter bank is to plan passband filter for every channel. This method introduces some drawbacks, for example scarce efficiency. In fact, if we would build a filter bank through FPGA, we could not have an elevate number of taps every filter and the out band rejection could not be very high as in reality it is desired to have. Polyphase filter bank is more efficient in comparison to filters bank FIR. In fact, in the polyphase filter bank is necessary to plan a low-pass filter only, whose transfer function will be translated in frequency for creating the various filters. The same taps of the low-pass filter are also the taps of every single filter of the bank. In this mode it is possible to have an high number of taps every filter and an high out band rejection.

A polyphase filter bank is composed by two parts (figure A.1):

- Convolution.
- DFT.

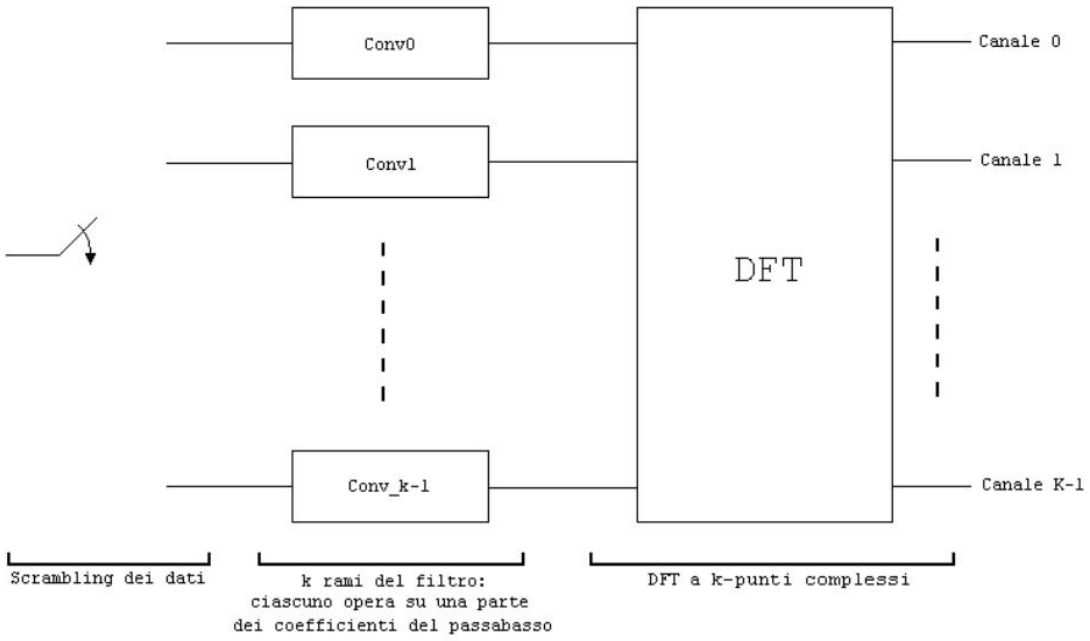

Figure A.1: Schematic blocks of the polyphase filter bank.

Convolution part is composed by low-pass filter as above descripted. This low-pass transfer function will be translated in frequency through DFT twiddle factor, creating the various channels of the bank. DFT computes the outputs only when all its input are relevant; this happens only when the last convolution block has performed its operation. Every filter channels work with a rate reduced by an K factor, with K as the number of channels.

$$S_{out} = \frac{S_{in}}{K} \qquad \text{(A.1)}$$

Take a filter bank with K filters, each costituited on a low-pass filter transfer function translated and centered on the following frequencies:

$$\omega_K = 2\pi \frac{k}{K} \qquad k= 0,1,2,...,K-1$$

Taking *h(n)* as the pulse response sampling of the original low-pass filter, the response of the filter number k is the following:

$$h_k(n) = h(n)e^{j\varpi_k n} \qquad \text{(A.2)}$$

This is a complex response in the time domain, with absolute value of the transfer function equal at the initial transfer function, but centered at ϖ_k frequency. Considering the sum of convolution (it is the output of a discrete time system vs input), for k-th filter we obtain:

$$y_k(n) = \sum_{p=1}^{KN_t} x(p)h_k(n-p) = \sum_{p=1}^{KN_t} x(n-p)h_k(p)$$

where KN_t is the total number of the filter taps. Effect the change of variable:

$$p=lK-i \qquad i=0,1,2,...,K-1$$

and consider the output of the k-th filter decimated by K:1, obtained to setting $n=mK$, the output expression of the filter is:

$$y_k(mK) = \sum_{i=0}^{K-1}\sum_{l=1}^{N_t} x\left[(m-l)K+i\right]h_k\left(lK-i\right) \qquad \text{(A.3)}$$

We now define the following term:

$$p_{i,k}(l) = h_k(lK-i) = h(lK-i)e^{j\varpi_k/K}e^{-j\varpi_k i} = h_k(lK-i) = p_{i,0}e^{-j\varpi_k i}$$

$$x_i(m) = x(mK + i) \qquad i=0,1,...,K\text{-}1$$

from which, (A.3) becomes:

$$y_k(mK) = \sum_{i=0}^{K-1} e^{-j2\pi ki/K} \sum_{l=1}^{N_t} x_i(m-l)\, p_{i,0}(l) \quad \text{(A.4)}$$

(A.4) is the output of the bin k-th of a K-points complex DFT. The K points supplied as inputs of the DFT are outputs of the convolution branch, each of these calculates the convolution between the decimated sequences $x_i(m)$ and the group $p_{i,k}(l)$. In the designed filterbank, data x_i are complex. The number of taps forming the $h_k(n)$ sequence, shown in (A.2), is the same of the *h(n)* and it is:

$$N = K \times N_k \qquad \text{(A.5)}$$

where K is the number of channels of the bank and N_k is the number of elements of $p_{i,k}(l)$ (in practical, this value is the taps number of each FIR in each branch). Note that (A.4) shown the signal $y_k(mK)$ in the time domain and it is present a factor decimator equal to the number of channels. In conclusion, we can affirm that this equation is the mathematical model described in the figure A.1.

References

"Polyphase-Discrete Fourier Tranform Spectrum Analysis for the Search for Extraterrestrial Intelligence Sky Survey"
G.A. Zimmerman.
S. Gulkis.

"An Improved FX Correlator"
John Bunton
CSIRO, Telecommunications and Industrial Physics, Australia, 19 December 2000

"Implementing Correlators for the SKA"
John Bunton

Motorola, AN1405
"ECL Clock Distribution Techniques"
Prepared by Todd Pearson

ON Semiconductor
"Interfacing Between LVDS and ECL"
Prepared by Paul Lee