APPLICATION OF THE POWERFFT^{TM} PROCESSOR IN MILITARY SOFTWARE DEFINED RADIO APPLICATIONS

Laurens Bierens and Peter Beukelman
(Eonic B.V., Delft, Netherlands; laurens.bierens@eonic.com / peter.beukelman@eonic.com)

ABSTRACT

In this paper we present the application of the PowerFFT processor – a digital full floating point spectrum processing chip – in military communication applications, such as Software Defined Radio (SDR) and Communication Intelligence (COMINT). The PowerFFT is superior in terms of performance, flexibility, footprint, and power consumption for the most demanding front-end processing operations compared to DSP or FPGA only alternatives above. The PowerFFT ASIC is combined in an SDR system architecture with FPGAs and DSPs, but relieves these devices from the heavy duty jobs required in military communications, which means that the overall architecture typically consumes less power and has a smaller footprint.

1. INTRODUCTION

Radio communications deal with very heterogeneous protocols, air interfaces, waveforms, etc. because it is strongly dependant on the local circumstances for the user: emitter/receiver, bandwidth, signal to noise ratio, multiple transmission links and of course local regulations These circumstances not only vary from country to country, but also over time. For example, in January 2005, the Federal Communications Commission is planning to grant narrowband radio equipment which provides at least one voice channel per 6.25kHz of channel bandwidth and/or are capable of supporting a minimum data rate of 4800 bits per second per 6.25kHz of channel bandwidth [1]. This is a logical migration from wider channel spacing (respectively 12.5kHz and 25kHz) but it implies a complete re-design of the analog front-end of transceivers. The new design involves stronger analog constraints (filters with higher order) and even more with ascendant compatibility for multi-mode equipments.

The limited flexibility of the analog designs can be bypassed by a Software Defined Radio approach where elementary analog functions such as frequency mixers or filters are replaced by software and/or firmware modules. Today, the popular alternative solutions for these applications Parallel DSP computing and/or FPGA based hardware.

Parallel DSP computers are flexible as they can be programmed in standard C/C++-language. Nevertheless, the solution is bulky and the power consumption is proportionally high. Further, they require a substantial software engineering effort for optimal code implementation.

FPGA based solutions may be smaller, typically a single board containing a few FPGAs and some memory components. FPGAs are considered today's solution because of their reconfigurable nature, but for extreme processing requirements they are also rigid solutions as their processing capability is often inversely proportional to the amount of flexibility. Further, the high power consumption for FPGAs operating on such high data rates is often not desirable in embedded applications.

And although the roadmap of FPGAs tends towards lower power and higher performance, also digitizer technology will push processing requirements proportionally. Requirements may become even more stringent as more bandwidth often induces more functionality, more flexibility, and more complexity. Furthermore, it is expected that sensitivity requirements of digital equipment will push the arithmetic accuracy requirements of front-end processing to (almost) floating point equivalent, which means that the power consumption and silicon area double or triple whereas the processing performance remains the same.

The PowerFFT provides an ASIC-centered alternative to well known FPGA-based reconfigurable computing and parallel DSP computing architectures. The key advantages of PowerFFT over these architectures are lower power consumption, much smaller footprint, lower costs, higher performance, combined with floating point accuracy. This opens up a new generation of portable SDR and battery operated COMINT systems.

In the next section we describe the general requirements of the PowerFFT Channelizer platform. In the subsequent sections, we describe the functionality of the current platform and some implementation issues. We end up with a short outline of the reconfigurable system components.

2. THE POWERFFT CHANNELIZER PLATFORM

The PowerFFT Channelizer is a reconfigurable Software Defined Radio platform in which the flexibility and performance of Eonic's PowerFFT processor is combined with FPGA logic to realize a software defined channelization architecture. Channelization is a technique to extract channels from an analog received radio signal in a digital domain instead of in the analog RF domain, see e.g. [2]. Characteristics of available bandwidth with different channel spacing width are dependent from authorities of regulations for the radio spectrum. The main difficulty in channelization is that a single channel represents a small part of the transmitted bandwidth. Thus in an analog design, precise mixers and high order filters are required whereas in a digital design, long filters and long FFTs provide good separation of adjacent channels. These considerations make the computational capabilities critical.

Eonic has built the PowerFFT Channelizer platform in the small 3U form factor. The guideline of this architecture is to offer homogeneous hardware configuration using CompactPCI with a flexible software approach. The combination of a high-speed digitizer with FPGA processing facilities and the ASIC's real time processing capabilities of the PowerFFT brings a reconfigurable on-the-fly architecture able to process high data rate flow. Considering the current status of each modules of the PowerFFT Channelizer, a global outlook can be defined:

Process a wideband: single or dual ADC channels 14bits@105MHz or 12bits@210MHz;

- Characteristics of channelization:
  - – Number of channel is a user defined parameter; nu niet
  - – Multi-mode channels spacing: 5 kHz / 6.25 kHz / 12.5 kHz / 25 kHz …;
  - – Spectrum monitoring, process as much channels as possible (see Table 1);
- Essential modules of digital signal processing:
  - – IQ Demodulation: It turns a real signal into complex by using generally half-band filters and mixers; sampling frequency can be reduced by a factor 2;
  - – Digital Mixer: It provides fine tuning of channels. Usual techniques are: sine table multiplies, cordic, direct digital synthesis …;
  - – Channel separation, to deliver proper separated channels, some processing is required ?in regards of? the core of the system: the PowerFFT; according to the available resources, channels sharpening could be achieved at different degree using flexible FFT processing and polyphase filtering.

Table 1. Flexible modes of operation of the PowerFFT Channelizer.

Channel Spacing	No. of channels	Input bandwidth	Sampling freq.
5 kHz	8192	40.96MHz	81.92MHz
6.25 kHz	8192	51.2MHz	102.4MHz
12.5kHz	4096	51.2MHz	102.4MHz
25kHz	2048	51.2MHz	102.4MHz
User Defined	User Defined	<105MHz	<105MHz

3. FUNCTIONAL DESCRIPTION

3.1. General functionality

The main functionality of the current PowerFFT Channelizer is the analysis of channels with 6.25kHz channel bandwidth within a certain analysis band, see Figure 1. As we intent to use an FFT-based channelizer for computational efficiency, the number of channels extracted from the analog input signal must be a power of two, say N. The number of channels in the analysis bandwidth, however, is flexible, say M. Usually we expand the analysis bandwidth to the nearest power of two, say $B_{analysis}$ = N×6.25kHz and set the sampling frequency at twice this value fs = $2B_{analysis}$. As an example, we use N = 8K in this paper, and thus fs = 102.5MHz.

Note that after sampling the digital signal is real and thus the analysis band as well as its mirror is in the digital signal. Therefore we apply an IQ demodulation directly behind the digitizer and reject the mirror band. The digital signal is then complex at half the sample frequency.

Figure 1 illustrates the channelizing process. The frequency bins are the outputs of the 8K-point FFT. Figure 2 shows the steps in the different steps in the channelizing chain. In the next subsections we describe these steps in detail.

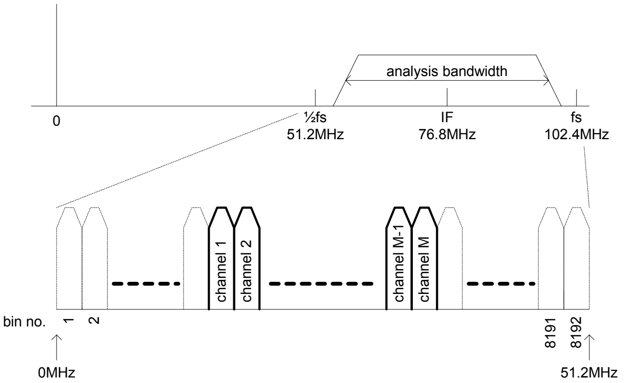


Figure 1 Channelizing for 8K channels of 6.25kHz.

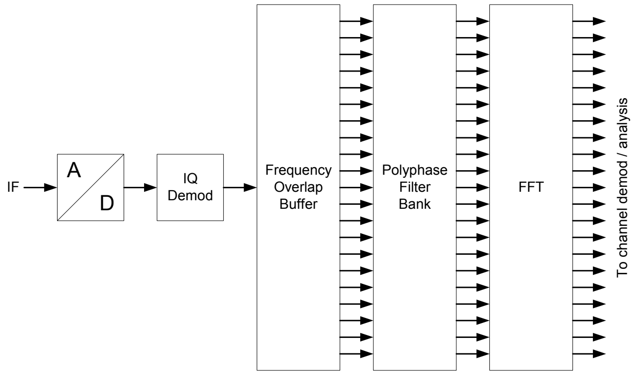


Figure 2 Functional channelizing chain.

3.2. IQ Demodulation

The IQ Demodulation is applied in two steps. First the digital signal is digital down converted with ¼ fs, which means in practice multiplying the digital samples with the sequence 1, -j, -1, j, 1, -j, -1, j, …. Secondly the mirror band is rejected and the sampling frequency is halved using a low-pass FIR filter with ¼fs cut-off frequency followed by factor 2 decimation. This process is illustrated in Figure 3. Note that the FIR filter has even order, is symmetric, and each odd tap is zero, see Figure 4.a.

Figure 4.b shows the frequency response of the used FIR filter. Observe that due to the limitations of the FIR filter the suppressed frequency components close to the cut-off frequency can fold into the channels.

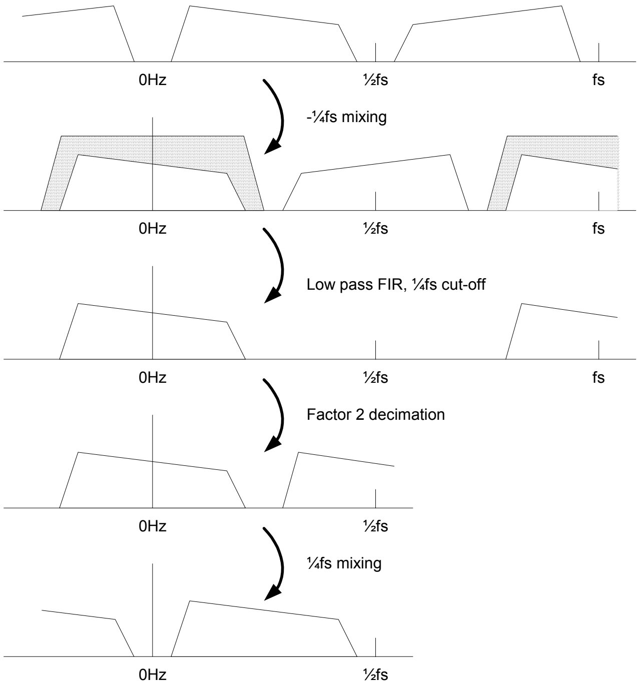
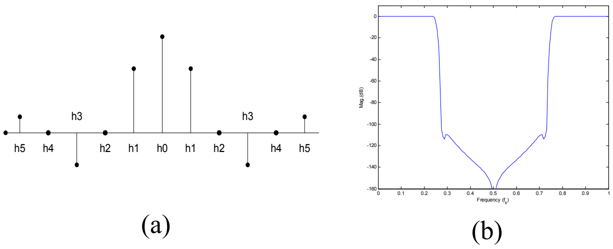


Figure 3 IQ demodulation.



(a)

(b)

Figure 4 IQ demodulation filter taps (only limited number of taps) (a) and typical filter spectrum (b)

3.3. Frequency Overlap Buffer

The basic goal of the PowerFFT Channelizer is to extract N channels from the received analog signal, and analyze them digitally. In our case, we extract 8K channels of 6.25kHz using an 8K-pts FFT filter bank from a 51.2MHz analysis bandwidth. Theoretically, we could decimate each channel down to a sample frequency close to the channel bandwidth, but in practice aliasing effects will cause to fold adjacent channel into the channel of interest. Obviously, sharp FIR filter banks can reduce this effect but with the sharp sampling of the channels very long FIR filters must be used, which is not desirable for reasons of computational complexity.

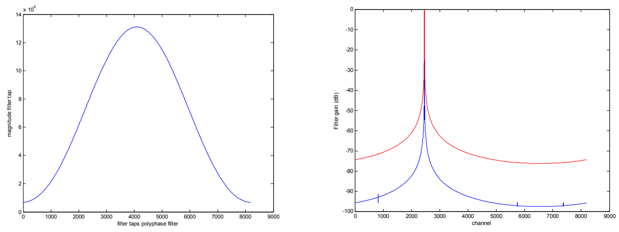
A better approach is to minimize this aliasing effect by oversampling the channels and use shorter FIR filters. This oversampling is accomplished by applying a so-called Frequency Overlap Buffer. The principle is illustrated as follows. Let the digitized input signal input signal be denoted as the vector [… x_3 x_2 x_1], and let the FFT-length (number of channels) be N = 8K. Suppose that we reshape the input vector into an N-row matrix as follows:

$$\begin{bmatrix} \cdots & x_3 & x_2 & x_1 \end{bmatrix} \xrightarrow{\textit{Frequency Overlap Buffer}} \begin{bmatrix} \cdots & x_{2Q+1} & x_{Q+1} & x_1 \\ \cdots & x_{2Q+2} & x_{Q+2} & x_2 \\ & \vdots & \vdots & \vdots \\ \cdots & x_{2Q+N} & x_{Q+N} & x_N \end{bmatrix}$$

with Q = N / S and S the oversampling factor per channel. Typical oversampling factors that we allow are 1, 2, 4, 8, 12, and 16 (where 1 is sampling at exactly the channel bandwidth).

3.4. Polyphase Filter Bank

Polyphase filtering is a well-known technique to improve the channel separation further using efficient implementation of long FIR channel filters [3]. The equivalent channel FIR filter for an N-channel system is a modulated version of a PxN-taps filter. It is implemented as a filter bank of N filters each having P taps, the so-called Polyphase Filter Bank. Typical numbers of P are 1, 2, and 4, where P = 1 is equivalent to a windowed FFT.



.. (a) (b)

Figure 5 Polyphase window for 8K-channel system (a), and response with and without polyphase filter (input signal in band 2457) (b).

3.5. FFT

The N-points FFT is realized in two steps: firstly a 16-pts fixed point FFT implemented in FPGA, and secondly an N/16-pts floating-point FFT (including twiddle multiplication) onto the PowerFFT processor. This algorithm is known as the twiddle factor algorithm [4]. As the PowerFFT can execute a floating-point FFT with variable length from 16 to 1K pts complex at 100MSPS, we have created an FFT engine for FFTs from 256 to 16K pts at 100MSPS. Thanks to the floating point nature of the PowerFFT the implementation has a very arithmetic accuracy by nature. The 16-pts FFT processor in FPGA is 18 bits fixed-point, but due to the limited amount of flexibility required we have been able to optimize the accuracy for this architecture.

4. IMPLEMENTATION ISSUES

4.1. General PowerFFT Channelizer architecture

Figure 6 shows the digital PowerFFT Channelizer pipeline architecture.

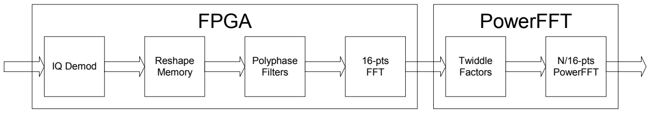


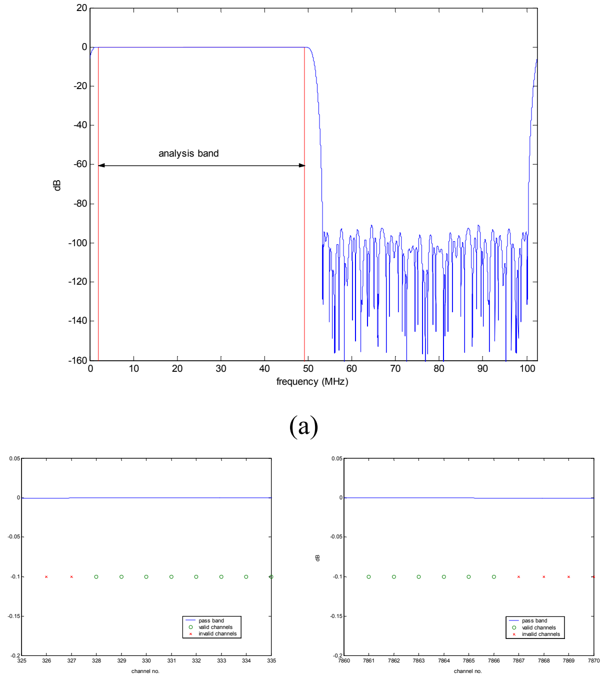
Figure 6. A basic digital channelizer pipeline architecture.

4.2. I/Q Demodulation

The Fs/4 shift of the spectrum as depicted in Figure 4 (i.e. multiplying with the sequence 1, -j, -1, j, 1, -j, -1, j, ….) only involves multiplexers and sign changes. The resulting signal is complex, but both real and imaginary samples contain 50% zeros.

The convolution of the low-pass FIR filter with the mixed data stream allows simplification of the hardware implementation. If real and imaginary parts are considered separately, one stream will get all the odd filter coefficients and the other one the even coefficients. Looking at the filter taps (Figure 4.a) it can be seen that the filter getting the even coefficients is zero except h0. This means the filter can be replaced by a FIFO, introducing a delay. The odd samples of the filter are symmetrical, so only half the number of multipliers is needed. Since the sample frequency of the complex signal is also halved, each multiplier running at incoming sample frequency can be used to calculate two taps.

In this firmware the implemented filter has 20 filter sections. Each filter section is mapped on a fixed point 18bit FPGA hardwired DSP blocks available in the Altera FPGA. The ratio of the smallest and largest taps is equal to the maximum resolution of 18 bits, in other words more filter taps would just add zeros. Each section effectively handles eight filter taps utilizing a) the factor two sample rate reduction, b) the filter symmetry, and c) the 50% zeros in the impulse response. Thus with 20 filter sections we effectively implement a 160 taps low-pass FIR filter.



(a)

(b) (c)

Figure 7 Realized 160-taps IQ Demodulation filter (fixed point) with effective analysis band (a), invalid first channels in analysis band (b) and invalid last channels in analysis band (c).

After low pass filtering the spectrum is shifted back with Fs/4. The effect of the IQ demodulation in the channels close to DC (channels 1, 2, …) and close to Fs/2 (channels …, 8K-1, 8K) depends on two aspects:

- the pass band of the of the low-pass filter;
- the stop-band of the low-pass filter in the mirror band (before factor 2 decimation).

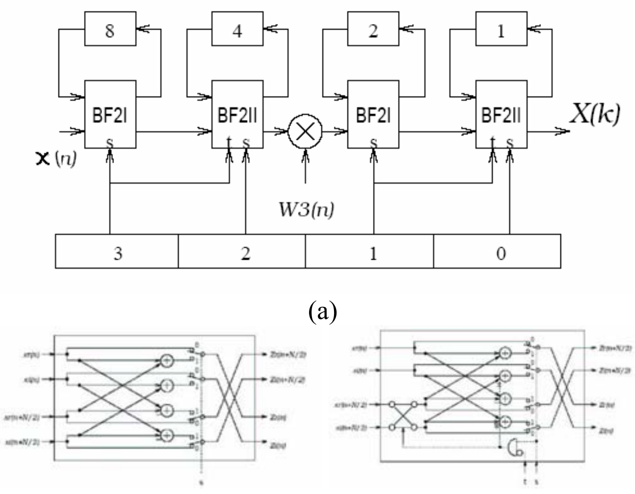
In Figure 7 these effects are illustrated for an 8K channelizer. Figure 7.a shows the filter spectrum after rounding the taps to 18bits. In Figure 7.b and Figure 7.c the valid and invalid channels first and last channels are shown. Our criteria to validate the channels is a pass-band ripple less than 0.5dB and a mirror stop-band of less than -90dB.

4.3. Reshape Memory

The frequency overlap and reshape functions are combined in FPGA in the so-called Reshape Memory. The overlap can be set by the number of rows the read counter has to advance for the next read from the reshape memory. The reshaping is implemented as follows. The N-pts FFT is split up into a N/16 16 pts FFTs followed by 16 512-pts butterflies (twiddle multiplies followed by an FFT). The 16 pts FFTs is executed on samples spaced N/16 points apart. This reshaping writes the incoming complex samples from the I/Q demodulator in rows of N/16 points wide. The data is then read from columns of 16 rows long. By this “corner turning operation”, the samples fed to the 16-pts FFT are spaced N/16 points apart.

4.4. 16-pts FFT

The16-pts FFT performs fixed point FFTs of 16 points in a pipelined structure. The architecture is described in [5]. The data path width is 18bit because of FPGA hardwired DSP blocks available in the Altera FPGA. A dual port memory on the output takes care of bit reversal. The FFT is implemented with four sections, see Figure 8. The bit reversal is done with a dual port memory of 16 points. Data is written and read in linear and bit reversed mode alternately.



(a)

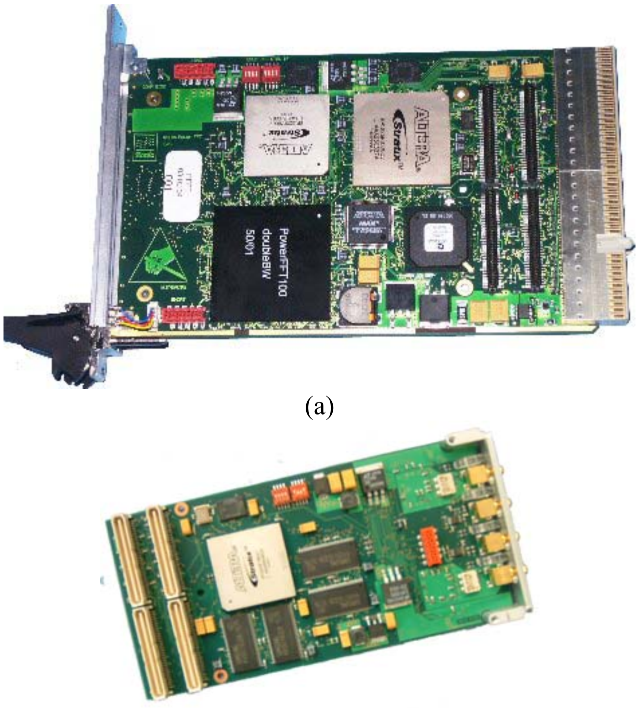
(b) (c)

Figure 8. 16-pts pipelined FFT (a) and BF2I (b) and BF2II (c) sections.

5. RECONFIGURABLE HARDWARE

The PowerFFT Channelizer architecture is mapped onto a Eonic’s 3U CPCI PowerFFT board (Figure 9.a), including a 105MSPS Digitizer CMC module (Figure 9.b). The architecture is housed in a CPCI rack with Pentium-based Host PC including WindowsXP. The IQ Demodulation, Reshape Memory, Polyphase Filter, and 16-pts FFT are implemented on an Altera Stratix EP1S40 on-board the Digitizer CMC module. The Twiddle Factor multiply and the N/16-pts FFT is implemented on the CPCI PowerFFT board.

The full N-channels data can be output to either the Host PC through the CompactPCI bus (non-real-time) or to three high-speed LVDS Power Links (200MBPS each) through the back plane. The latter can be used to connect to Eonic’s accompanying CPCI TigerSHARC DSP board or reconfigurable CPCI FPGA board for channel processing.



(a)

(b)

Figure 9 Eonic’s reconfigurable CPCI PowerFFT board (a) and Dual 105MSPS Digitizer CMC module (b).

6. CONCLUSIONS

In this paper we have presented the PowerFFT processor in military communication applications, such as Software Defined Radio (SDR) and Communication Intelligence (COMINT). The PowerFFT provides an ASIC-centered alternative to well known FPGA-based reconfigurable computing and parallel DSP computing architectures. The key advantages of PowerFFT over these architectures are lower power consumption, much smaller footprint, lower costs, higher accuracy and higher performance. This opens up a new generation of portable SDR and battery operated COMINT systems. The PowerFFT has an unprecedented performance in sustained multi-dimensional spectrum processing operations. It combines flexibility (programmable) with high-accuracy (floating point) with extreme performance (6 GFLOPS per processor) and – uniquely – with low-power consumption (typically 1W per processor).

We have elaborated on the PowerFFT Channelizer architecture, a device designed for real-time front-end signal processing operations in portable and battery powered military communication equipment. Typically, these operations occur directly behind the digitizer and involve signal enhancement, polyphase filtering, spectrum analysis and synthesis as well as other functions. At the same time, digitizer technology advances towards broader bandwidths and higher resolutions allowing full digital IF/RF signal processing. Commonly available are 14bit@105MHz and 12bit@210MHz ADC components, but newer components already digitize up to 400MHz signals at 14bit. The challenge hereby is to process the full digital bandwidth with sustained rate utilizing multi-mode, reconfigurable and flexible devices.

7. REFERENCES

[1] http://wireless.fcc.gov/services/ind&bus/operations/equipment.html

[2] Lee Pucker, “Channelization Techniques for Software Defined Radio”, SDR Forum 2003.

[3] P.P. Vaidyanathan, “Multirate Digital Filter, Filter Banks, Polyphase Networks and Applications: A Tutorial”, *Proc. Of the IEEE*, 78(1):56-93, January 1990

[4] Oran E. Brigham, *The Fast Fourier Transform*, Prentice-Hall, 1974

[5] S.He and M.Torkelson, “A new approach to pipelined FFT processor”, in Proc. Of the 10th Intern. Parallel Processing Symp. (IPPS), pp 766-770, April 1996