We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

FPGA Based Serial and Single-Clock Cycle Pipelined Fast Fourier Transforms in a Radio Detection of Cosmic Rays

Zbigniew Szadkowski

Additional information is available at the end of the chapter

1. Introduction

Results from various cosmic rays experiments located on the ground level, point to the need for very large aperture detection systems for ultra-high energy cosmic rays. With its nearly 100% duty cycle, its high angular resolution, and its sensitivity to the longitudinal air-shower evolution, the radio technique is particularly well-suited for detection of Ultra High-Energy Cosmic Rays (UHECRs) in large-scale arrays. The present challenges are to understand the emission mechanisms and the features of the radio signal, and to develop an adequate measuring instrument. Electron-positron pairs generated in the shower development are separated and deflected by the Earth's magnetic field [1], [2], hence they introduce an electromagnetic emission. During shower development, charged particles are concentrated in a shower disk of a few meters thickness. This results in a coherent radio emission up to about 100 MHz. Short but coherent radio pulses of 10 ns up to a few 100 ns duration are generated with an electric field strength increasing approximately linearly with the energy of the primary cosmic particle inducing the extended air showers (EAS), i.e. a quadratic dependence of the radio pulse energy vs. primary particle energy. In contrast to the fluorescence technique (e.g. used in the Pierre Auger Observatory [3]) with a duty cycle of about 12% (fluorescence detectors can work only during moonless nights), the radio technique allows nearly full-time measurements and long range observations due to the high transparency of the air to radio signals in the investigated frequency range.

The radio detection technique will be complementary to the water Cherenkov detectors and allows a more precise study of the electromagnetic part of air showers in the atmosphere. In addition to a strong physics motivation, many technical aspects relating to the efficiency, saturation effects and dynamic range, the precision for timing, the stability of the hardware

developed, deployed and used, as well as the data collecting and system-health monitoring processes will be studied and optimized.

EAS are investigated in several experiments utilizing different detection techniques (scintilators, water Cherenkov and fluorescence detectors). Signals in the detectors depend on several parameters such as the energy, the type of the primary particle, a distance from the core, the angle of registered shower, etc. Usually the triggering conditions are chosen such as to detect as wide as possible classes of events. However, sometimes the standard trigger conditions are not optimized for the specific class of events, which are either not registered at all or for which the registration efficiency is poor. In experiments utilizing water Cherenkov detectors, signals from photo-multipliers (PMTs) are usually digitized in ADCs and next processed by often-sophisticated electronics. In order to increase the signal/noise ratio coincidence techniques are widely used. Typically signals from PMTs are analyzed on-line in both amplitude and time domains. Strong signals in all PMT channels, corresponding to energetic showers detected near the core, are registered because of many-fold coincidence single bin trigger with a fixed thresholds. Showers detected far from the core give much lower signals usually spread in time. Such events are detected by the other type of trigger investigating the structure of signal in some period (in a sliding time window).

The structure of signals detected in water Cherenkov tanks and generated by horizontal showers depend strongly on the point of the EAS initialization. "Old" showers generated by hadrons early in the atmosphere give flat muonic front; showers generated by deeply interacting neutrinos are characterized by a curved front (radius of curvature of a few km), a large electromagnetic component and with particles spread over a few microseconds interval [4]. In both cases muonic front produces a bump, which can be a starting signature of horizontal showers. The bump for the "old" showers is shorter and sharper than for the "young" ones and results in a larger contribution in higher Fourier coefficients. For "young" showers, with relatively smooth shape of a signal profile, the lower Fourier components should dominate. The on-line analysis of the Fourier components may trigger specific events.

The existing software procedures, available as commercial IP routines, can calculate Fourier coefficients effectively utilizing a FFT algorithm. However the software implementation is too slow to be able to trigger events in the real time. On-line triggering requires the hardware implementation calculating multi-point DFT with a sufficient speed. Modern powerful FPGAs can do this job, however, the resource requirement increases dramatically with the number of points. The analysis time interval should be a reasonable compromise between the time resolution and the resources occupancy in the FPGA.

2. DFT

The discrete Fourier transform (DFT), of length N, calculates the sampled Fourier transform of a discrete-time sequence at N evenly distributed points $\omega k = \frac{2\pi k}{N}$ the unit circle. The following equation shows the length-N forward DFT of a sequence x(n):

$$\bar{X}_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2i\pi kn}{N}} \qquad k = 0, 1, ...N-1 \tag{1}$$

The following equation shows the length-N inverse DFT:

$$\tilde{x}_k = \frac{1}{N} \sum_{n=0}^{N-1} \bar{X}_n e^{\frac{2i\pi kn}{N}} \qquad k = 0, 1, ... N-1 \qquad (2)$$

The complexity of the DFT direct computation can be significantly reduced by using fast algorithms that use a nested decomposition of the summation in equations one and two-in addition to exploiting various symmetries inherent in the complex multiplications. Such algorithms are the Radix-r Decimation-in-Time (DiT) or Radix-r Decimation-in-Frequency (DiF) Fast Fourier Transforms (FFT), which recursively divides the input/output sequence into N/r sequences of length r and requires $log_r N$ stages of computation.

The commercially offered FFT processors for FPGA applications require several clock cycles to accomplish calculation of all complex DFT coefficients. Each stage of the decomposition typically shares the same hardware, with the data being read from memory, passed through the FFT processor and written back to memory. Each pass through the FFT processor is required to be performed $log_r N$ times. Popular choices of the Radix are r = 2, 4, and 16. Increasing the Radix of the decomposition leads to a reduction in the number of passes required through the FFT processor at the expense of device resources. Such an approach is very widely useful for many applications, where timing is not crucial. However, there are areas, where the FFT coefficients (based on a new set of samples) have to be known in each clock cycle. Commercial FFT processors, unfortunately, cannot be used. This approach requires special algorithms optimized for a particular solution.

2.1. Radix-2 : Decimation-in-Time and Decimation-in-Frequency

The Radix-2 algorithm is the simplest FFT one. The decimation-in-time (DIT) Radix-2 FFT recursively partitions a DFT into two half-length DFTs of the even-indexed and odd-indexed time samples. For the Radix-2 DiT, we get :

$$\bar{X}_k = \sum_{n=0}^{N-1} x_n e^{-2i\pi kn/N} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-i\frac{2\pi kn}{N/2}} + e^{-i\frac{2\pi k}{N}} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-i\frac{2\pi kn}{N/2}} = \qquad (3)$$

$$= DFT_{\frac{N}{2}}[x_0, x_2, ..., x_{N-2}] + W_N^k \times DFT_{\frac{N}{2}}[x_1, x_3, ..., x_{N-1}]$$

For the Radix-2 DiF, we get :

$$\bar{X}_{2k} = \sum_{n=0}^{N-1} x_n W_N^{2kn} = DFT_{\frac{N}{2}}\left[x_n + x_{n+\frac{N}{2}}\right] \qquad (4)$$

$$\bar{X}_{2k+1} = \sum_{n=0}^{N-1} x_n W_N^{(2k+1)n} = DFT_{\frac{N}{2}}\left[\left(x_n - x_{n+\frac{N}{2}}\right) W_N^n\right]$$
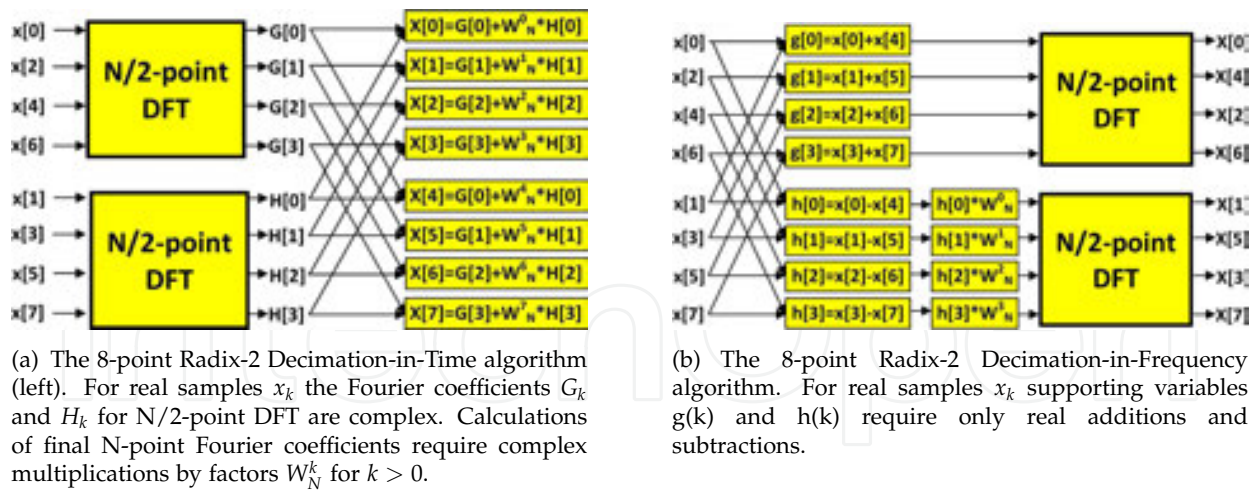
(a) The 8-point Radix-2 Decimation-in-Time algorithm (left). For real samples x_k the Fourier coefficients G_k and H_k for N/2-point DFT are complex. Calculations of final N-point Fourier coefficients require complex multiplications by factors W_N^k for $k > 0$.

(b) The 8-point Radix-2 Decimation-in-Frequency algorithm. For real samples x_k supporting variables g(k) and h(k) require only real additions and subtractions.

Figure 1.   Splitting of N-point DFT on two N/2-point parallel procedures for Decimation-in-Time (left) and Decimation-in-Frequency (right), respectively, on the basis of the 8-point Radix-2 algorithms.

The N-point DFT can be easily split on two N/2-point transforms. The outputs from DFT procedures are complex. So, a calculation of final DFT coefficients by using DiT algorithm requires complex multiplication for final merging data from parallel DFT procedures with lower order (i.e. multiplication of twiddle factors W_N^k) :

$$W_N^k = e^{-i\frac{2\pi k}{N}} \tag{5}$$

by $G[k]$ and $H[k]$ in Figure 1. For the DiF algorithm the 1st stage requires additions and subtractions only. Odd indexes require additional multiplications, however, even indexes remain without modifications for the next N/2-point DFT procedure (compare Figures 1a and 1b).

2.2. Radix-4 algorithm

The Radix-4 algorithm consists of four inputs and four outputs. The FFT length is 4^p, where p is the number of stages. A stage is half of Radix-2. The Radix-4 DIF FFT divides an N-point DFT into four N/4 -point DFTs, then into 16 N/16 -point DFTs, and so on.

For Radix-4 DiF, we get :

$$\bar{X}_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2i\pi kn}{N}} = \sum_{n=0}^{N/4-1} x_n e^{\frac{-2i\pi kn}{N}} + \sum_{n=N/4}^{N/2-1} x_n e^{\frac{-2i\pi kn}{N}} + \sum_{n=N/2}^{3N/4-1} x_n e^{\frac{-2i\pi kn}{N}} + \sum_{n=3N/4}^{N-1} x_n e^{\frac{-2i\pi kn}{N}} =$$

$$= \sum_{n=0}^{N/4-1} e^{-\frac{2i\pi kn}{N}} \left[x_n + (-i)^k x_{n+N/4} + (-1)^k x_{n+N/2} + (i)^k x_{n+3N/4} \right] \tag{6}$$

This algorithm is widely used, however, as it is shown in a next section, the simple application of the DiT or DiF algorithms in all sequential steps remains still an area for further optimization.

2.3. Architectures of the Altera®'s FFT MegaCore® functions.

2.3.1. Streaming architecture

The Radix-4 decomposition, which divides the input sequence recursively to form four-point sequences, has the advantage that it requires only trivial multiplications in the 4-point DFT, it is the chosen Radix algorithm in the Altera® FFT MegaCore® function. This results in the highest throughput decomposition, while requiring non-trivial complex multiplications in the post-butterfly twiddle-factor rotations only. In cases where N is an odd power of two, the FFT MegaCore automatically implements the Radix-2 pass on the last pass to complete the transform.

To maintain a high signal-to-noise ratio throughout the transform computation, the FFT MegaCore function uses a block-floating-point architecture, which is a compromise point between fixed-point and full-floating point architectures. In the fixed-point architecture, the data precision needs to be large enough to correctly represent all intermediate values throughout the transform computation. For large FFT transform sizes, the FFT fixed-point implementation that allows for word growth can make either the data width excessive or can lead to a loss of precision.

In the floating-point architecture each number is represented as a mantissa with an individual exponent, while this leads to greatly improved precision, floating-point operations tend to demand increased device resources.

In the block-floating point architecture, all of the values have an independent mantissa but share a common exponent in each data block. Data is input to the FFT function as fixed point complex numbers (Figure 2).
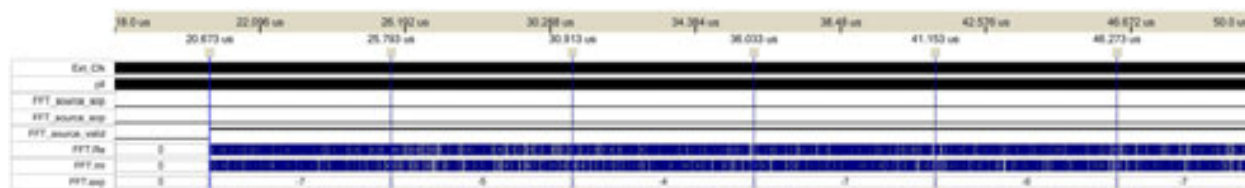

Figure 2. A simulation of the Fourier transform for the Altera® library routine of 1024 points and for streaming architecture. Each block of 1024 Fourier coefficients (Fc) is scaled by the factor FFT.exp. Fourier coefficients are provided in a serial way, each pair of real and imaginary parts of a single Fc in a single time bin. All Fc are calculated in 1024 time bins. FFT.sop (start of package) and FFT.eop (end of package) indicate begin and end of each 1024-point block.

The block-floating point architecture ensures full use of the data width within the FFT function and throughout the transform. After every pass through the Radix-4 FFT, the data width may grow up to $log_2(4\sqrt{2}) = 2.5$ bits. The data is scaled according to a measure of the block dynamic range on the output of the previous pass. The number of shifts is accumulated and then output as an exponent for the entire block. This shifting ensures that the minimum of least significant bits (LSBs) are discarded prior to the rounding of the post-multiplication output. In effect, the block-floating point representation acts as a digital automatic gain control. To yield uniform scaling across successive output blocks, you must scale the FFT function output by the final exponent [5].

2.3.2. Variable streaming architecture

The variable streaming architecture uses two different types of architecture, depending on which representation: the fixed-point or the floating-point is selected. For the fixed-point data representation, the FFT variation uses a Radix-2^2 single delay feedback architecture, which is a fully pipelined architecture. For the floating point representation, the FFT variation uses a mixed Radix-4/2 architecture. For a length N transform, log4(N) stages are concatenated together. The Radix-2^2 algorithm has the same multiplicative complexity of the fully pipelined Radix-4 architecture, but the butterfly unit retains the Radix-2 architecture. In the Radix-4/2 algorithm, a combination of Radix-4 and Radix-2 architectures are implemented to achieve the computational advantage of the Radix-4 architecture while supporting FFT computation with a wider range of transform lengths. The butterfly units use the DIF decomposition.

The fixed point representation allows for natural word growth through the pipeline. The maximum growth of each stage is 2 bits. After the complex multiplication the data is rounded down to the expanded data size using convergent rounding. The overall bit growth is less than or equal to log2(N)+1. The floating point internal data representation is the single precision floating point (32-bit). Floating point operations provide more precise computation results but are costly in hardware resources. To reduce the amount of logic required for floating point operations, the variable streaming FFT uses "fused" floating point kernels. The reduction in logic occurs by fusing together several floating point operations and reducing the number of normalizations that need to occur [5].

3. An FPGA based RFI filter for radio detection of cosmic rays

3.1. A physical background

The energy threshold of radio detection of cosmic rays is limited by the considerable radio background and noise. The very high level of radio frequency interferences (RFI) in the FM and the short wave band has to be eliminated by a band pass filter amplifier. Within the remaining receiver bandwidth of 30 to 80MHz the noise at the quiet-rural environment of cosmic-rays experiments is dominated by the frequency dependent galactic noise [6] with noise temperatures of 5000K at 60 MHz.

In addition to galactic noise, there is still a human made background. This background consists of continuous signals, as from a few radio and TV stations, and transients produced by machines. Without an effective trigger, a stable and low-level energy threshold is not guaranteed. Furthermore, the data rate for communication of the triggered data to the central DAQ would exceed the available power budget.

For self-triggered measurements, the data will be digitized and processed in real time by a powerful FPGA chip. The narrow peaks in the frequency domain due to radio frequency interferences have to be strongly suppressed before building a trigger. These peaks are removed by a median filter. The filter works in the frequency domain using the Fast Fourier Transform (FFT) routine provided by Altera®. Furthermore, the phase of the signal deformed by the steep band pass filter is reconstructed by a deconvolution in the frequency domain.

The median FPGA filter eliminates mono-frequent carriers, but broadband radio pulses from cosmic showers are not affected. After a second inverse FFT, signals are converted back to

the time domain. This chain of the digital signal processing strongly enhances the signal to noise ratio, and thus improves the radio pulse detection sensitivity (Figure 3).

Due to the Nyquist theorem, the used 80 MHz band should be sampled with at least 160 MHz. An application of 16-bit ADCs with such a sampling rate would be a challenge for the price, the power consumption and PCB routing to keep a reasonable noise level. The used practical option is an 12-bit ADC with 180 MSPS, leaving sufficient space for the anti-aliasing filter and implementing a high and low gain channel to obtain the required dynamic range.
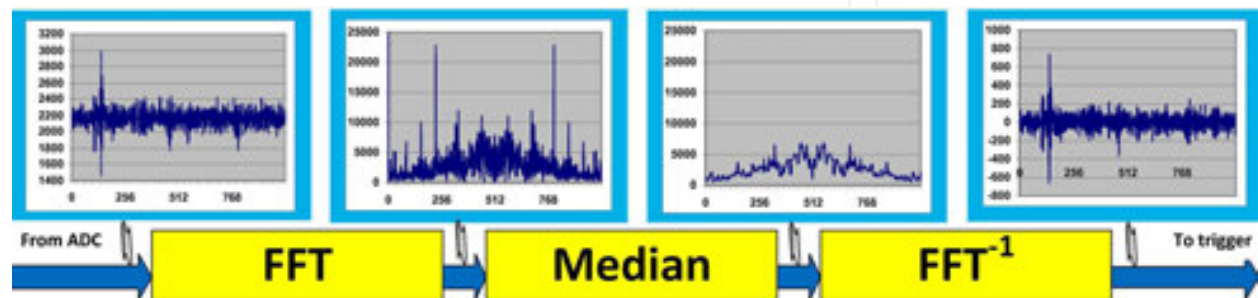
Figure 3. A diagram showing a (FFT + Median filter + iFFT) chain cleaning the signal from the RFI contamination. The 1st graph shows the ADC input as unsigned data with an offset of ca. 2300 ADC-counts, the 2nd - the absolute values of FFT coefficients in the frequency domain, the 3rd - FFT coefficients "decontaminated" by the median filter and 4th - signal converted back to the time domain. Additionally, the 0th FFT coefficient has been zeroed. Thus, the cleaned signal in the time domain is represented as signed data without the offset. The amplitude of the signal remains roughly the same and the noise is considerably reduced.

The necessary filtering accuracy requires at least 1024-point Fourier transforms. For the 180 MHz sampling, it corresponds to 360 kHz resolution in the frequency domain. Shorter transformation blocks give too rough filtering and may affect real signals from showers. For these parameters, the RFI filter has been developed and optimized [7].

3.2. Selection of the FFT architecture

The Altera® FFT MegaCore offers 4 types of FFT engines with various architectures :

- streaming
- variable streaming
- burst
- buffered burst

calculating the FFT and iFFT in real-time. All architectures can be implemented a fixed point FFT, whereas the variable streaming architecture can also be configured in a floating point data representation. A comparison of resource occupancy of different architectures is given in Table 1. Parameters are shown for 12-bit and 16-bit data processing.

architecture	TCC	BTC	LE wizard 12-bits	memory bits 12-bits	DSP 12-bits	LE wizard 16-bits	memory bits 16-bits	DSP 16-bits
streaming	1024	1024	3723	155 648	24	4952	155 648	24
variable streaming								
fixed bit reverse	1024	1024	6139	31792	48	7175	39 976	56
floating bit reverse	1024	1024	23000	73 568	128	23000	73 568	128
fixed natural order	1024	1024	6139	82 992	48	7175	100 380	56
floating natural order	1024	1024	23000	139 104	128	23000	139 104	128
burst (single engine)	1113	3162	2814	57 344	24	3804	57 344	24
burst (4 engines)	345	2394	7864	114 688	96	11 136	114 688	96
buffered burst (single engine)	1103	1291	3202	122 880	24	4197	122 880	24
buffered burst (4 engines)	335	1099	8517	245 760	96	11 885	245 760	96

Table 1. An utilization of resources for various FFT architectures at 12-bit and 16-bit data processing. The 2nd column shows Transform Calculation Cycles (TCC), required by the Altera® wizard, the 3rd - Block Throughput Cycles (TBC), the 4th - required Logic Elements (LE), the 5th - required memory bits, the 6th - required Digital Signal Processing blocks. Parameters in columns 4th to 6th correspond to the 12-bit data processing, in columns 7th to 9th to the 16-bit processing, respectively.

For the RFI filtering scheme shown on Figure 3 sampled ADC data have to be processed continuously in real-time. "Continuously" means that any dead-time is not acceptable. Data can be processed in blocks of fixed length, but no any sample can be ignored. This requirement eliminates two architectures: burst and buffered burst, because for i.e. 1024-point (and 1024 clock cycles when samples appear from the ADC output) these architectures require more than 1024 clock cycles for processing (BTC = 3162, 2394, 1103 and 1099 for burst and buffered burst and single and 4 engines, respectively). For any configuration the fundamental requirement of no dead-time is not obeyed.

The floating point representation for the variable streaming architecture requires huge amount of logic elements and DSP blocks. For two cascade FFT engines for two polarization channels almost all resources could be utilized for the FFT engines only. There would not be resources for other tasks. Additionally, the Altera®'s documentation shows that the registered performance for this architecture is much below our expectations (on the level of 110 MHz, while we need at least 180 MHz for the signal processing).

Some FFT applications require the FFT + the user operation + the iFFT chain. In this case, a careful selection of the input and output order can significantly save a memory and a latency. If the input to the first FFT is in the natural order and the output is in the bit-reversed order, the FFT engine operates in a mode with a minimal resource utilization (called Engine-only mode). Thus, if the iFFT operation is configured to accept bit-reversed inputs and produces natural order outputs (iFFT is operating again in Engine-only mode), only the minimum amount of memory is required, which provides a saving of N complex memory words, and a latency saving of N clock cycles, where N is the size of the current transform.

However, in the case of the RFI filtering by the median filter the sequence of FFT coefficients in the frequency domain has to be natural, to eliminate/suppress narrow-band peaks. The FFT routines have to be working with Engine with bit-reversal modes only. Two architectures: (a) streaming and (b) variable streaming with the natural order and the fixed-point data representation survived the selection.
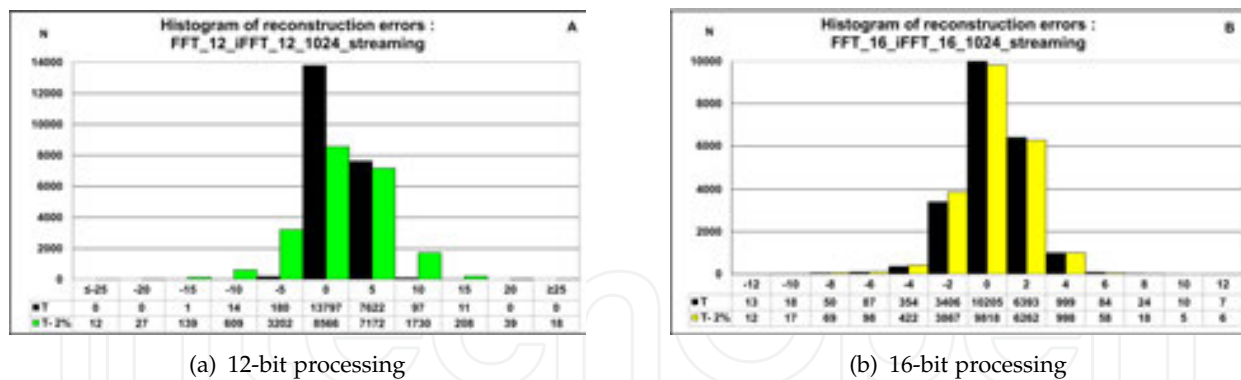
(a) 12-bit processing   (b) 16-bit processing

Figure 4. Histograms of reconstruction errors for the streaming architecture (differences between the original ADC data and data after application of the 12-bit (a) and 16-bit (b) wide FFT and inverse FFT). "T" denotes ideal configuration with zeroed offset. However, for a small shift of 2% only, error distributions become wider and more flat.

3.3. Streaming architecture

The streaming architecture accepts as input a two's complement format with a complex data vector of length N, where N is the desired transformation block length. The function output is given as a complex vector in the natural order. An accumulated block exponent is given to indicate any data scaling that has occurred during the transformation to maintain precision and maximize the internal numerical signal-to-noise ratio.

The signed block exponent, used for scaling of internal signal values, remains constant for a full data block. For relatively small variations of the signal samples x_n (typical for noise background), but with not negligible pedestal the Fourier component $\bar{X}_0$, may be relatively large whereas the $\bar{X}_{n \neq 0}$ components are rounded off to relatively small values. This may cause large errors of the reconstructed signals after going through the FFT/iFFT chain. Hence, the pedestal has to be subtracted carefully from the input signal. Errors of the reconstruction for the 1024-point transforms of a real event signal recorded in real cosmic rays experiment are shown in Figure 4.

The streaming architecture introduces, unfortunately, significant distortions of signals in a data processing for the FFT+iFFT cascade chain. The reconstruction errors for the 12-bit processing are on unacceptable level of 10 and more ADC-counts. The 16-bit configuration introduces smaller reconstruction errors and maybe used for real data processing, however, an influence of the data processing errors have to be carefully take into account for the final trigger and recorded data.

Figure 5 shows a possible optimization, where 12-bit data is processing in 14-bit FFT engine and 2 lower significant bits are grounded and treated as potentially fractional part.

3.4. Variable streaming architecture

The 12-bit input FFT routine with the variable streaming architecture yields 25-bit Re/Im Fourier coefficients. Processing of both buses with the full width in the iFFT procedure would be too spendthrift and slows down the speed significantly. A reasonable compromise for a selection of the input lines driving the iFFT routine is required.

Figure 6 shows that cropping the output FFT bus to 12 bits provides already a good reconstruction. The error is on the level of one ADC-count. This is achieved at the expenses of 2000 additional LEs and 24 additional DSP blocks. However, this architecture's maximum clock frequency of roughly 200 MHz (for selected FPGA from Cyclone® III family) is too low.

3.5. Aliasing and leakage removal

The incoming data stream must be chopped into blocks to be processed by the FFT routine. If signal pulses are located close to the border of a block, aliasing occurs. It manifests by a spurious contribution in the opposite border of the block and in the neighboring block as well. This effect may cause spurious pulses and has to be eliminated. The leakage effect is caused by the finite length of the blocks, acting like an applied rectangular window function. Thus, a signal amplitude leaks from one frequency bin to another. By using a suitable window function, the leakage effect can be reduced. To keep algorithmic costs low, we use a window function with a constant middle part like a trapezoidal shape or a Tukey-window.
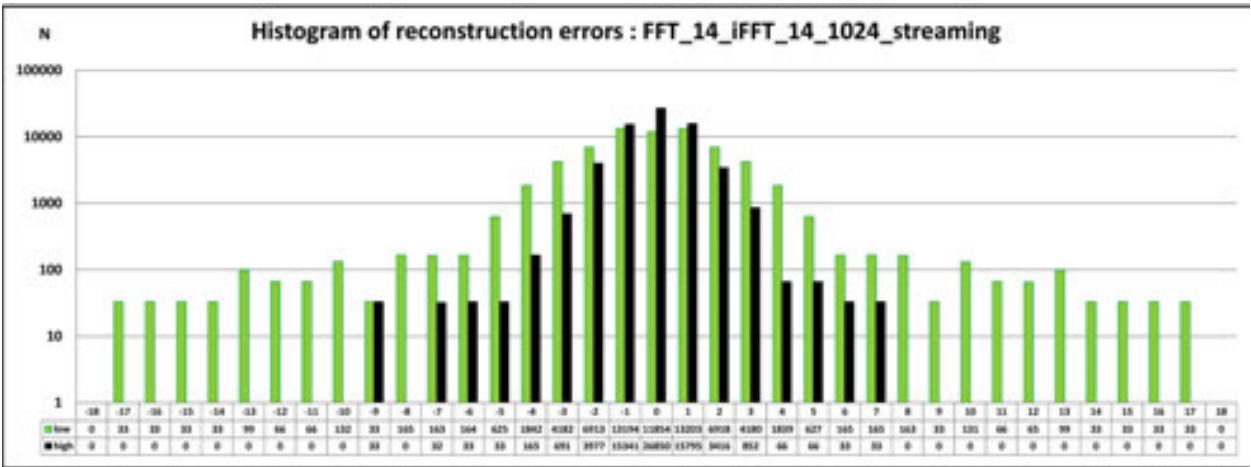

Figure 5. Histograms of reconstruction errors for the streaming architecture (differences between the original ADC data and data after application of the 14-bit wide FFT and inverse FFT). The width of input data is 12 bits connected to low 12 bits (starting from LSB) ("low") or to higher 12 bits (starting from MSB (high). For "low" configuration 13th and 12th input bits are connected to the sign (11th) bit. A distribution of the reconstruction errors is rather wide. For the "high" configuration 0th and 1st are grounded and they play role of a fractional zeroed input part. For a such modification of input connection only, the error distribution is significantly narrower.
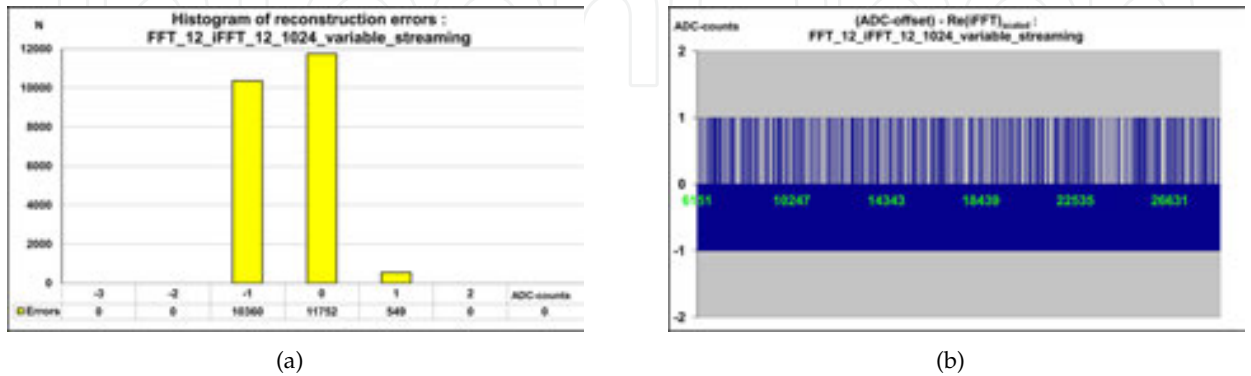

(a)

(b)

Figure 6. Histogram (a) of reconstruction errors for the variable streaming architecture (differences between the original ADC data and data after application of the 12-bit wide FFT and inverse FFT). The right plot (b) shows differences for raw data.

Both problems can only be solved, without introducing dead time between the blocks, by using an overlapping routine [7]. Therefore the filter engine must run in another clock domain with higher frequency. Preliminary estimation shows that for an overlapping of N = 32 errors due to an aliasing contribution is acceptable, however for a better safety margin N = 64 is preferred. N = 128, allows a total removal of aliasing effect, however this option requires too high over-clocking according to Table III. An odd value like N = 73 seems to be a valid compromise, although requiring some special modules to assure a seamless hand over of the data stream between the different clock domains.
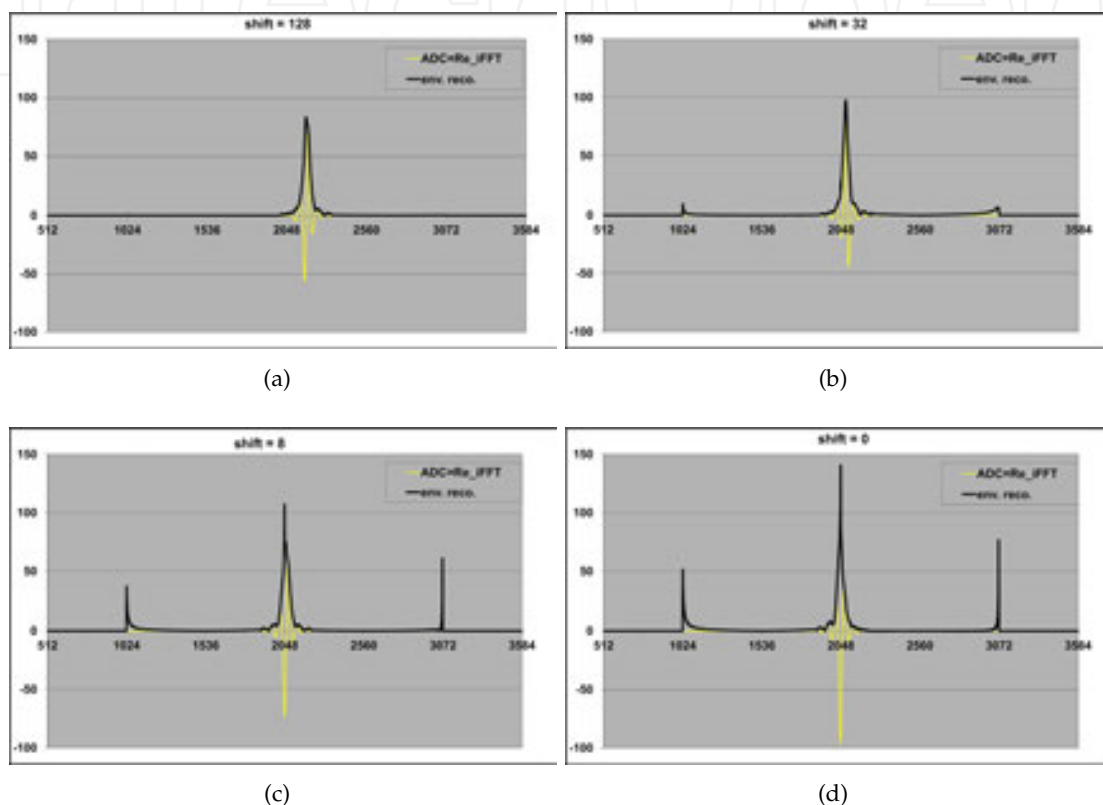

Figure 7. An example of spurious envelopes due to aliasing, when a signal appears close to the border of converted blocks, 128, 32, 8 and exactly on the border, respectively

Figures 7 show a potential danger if the aliasing were ignored. If the signal appears relatively far from the end of the block border (i.e. 128 time bins for 1024-point conversion) the envelope of the signal is reconstructed rather good (Figure 7a). There is no any false peaks, which could be recognized as spurious triggers. If the signal appears relatively close the end of the block border (Figure 7b) one can observe some spurious wings on the borders of neighboring blocks. However, if a relatively strong signal appears close to the block border (Figure 7c) the spurious peaks are created on both borders and there is a very high danger that these spurious peaks can be mistakenly taken as a trigger. If the signal appears exactly on the border of two blocks (Figure 7d), the spurious peaks can get an amplitude of more than 30 % of real signal. An additional procedure removing a spectral leakage has to be absolutely used to keep a high reliability of the system.

3.6. Simultaneous processing of two signals with perpendicular polarizations

Each antenna station measures radio signals in two opposite polarization channels. Thus, it would be straightforward to use two FFT engines for calculating the frequency domain signal, while setting their imaginary input to zero. A more efficient way is to exploit the symmetries of the FFT. Therefore the data streams of both antenna channels (N windowed signal samples f_j and g_j) are connected to the real respectively imaginary component input of the FFT engine. The resulting output components, H_n, are given in (7).

$$H_n = \sum_j e^{2\pi ijn/N}(f_j + ig_j) \tag{7}$$

The H_n can then easily be disentangled into the Fourier components, F_n and G_n, by the following equations (8)

$$H_n + H_{N-n}^* = 2F_n , \quad H_n - H_{N-n}^* = 2iG_n, \tag{8}$$

The (N-n) indices in (8) in a real time system correspond to a time reversed order. The H_n and H_{N-n}^* are synchronized by a routine inverting the order of the H_n^* like First In Last Out (FILO) and by using a delay routine for the H_n in parallel. Doing so, the amount of needed FFT engines can be reduced from two to one.

After the iFFT, the envelopes $f_{env}(t)$ and $g_{env}(t)$ (Figure 8) of the output signal $x(t)$ have to be created to allow the following trigger algorithms to discriminate specific pulse shapes in each channel.
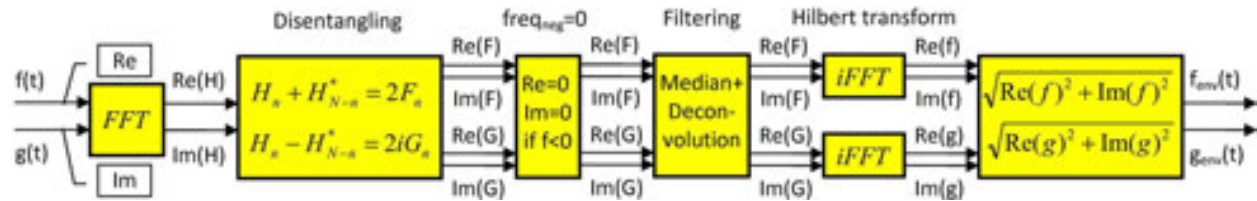

Figure 8. Schematic view of the resources-optimized implementation of the used two antenna chains with opposite polarization, each consisting of FFT, median filter, deconvolution, Hilbert transform Im(f) and Im(g) and FIR filters.

4. Wavelets

Let us investigate a time series X, with values of x_n, at time index n. Each value is separated in time by a constant time interval Δt. The wavelet transform $W_n(s)$ is just the inner product (or convolution) of the wavelet function with our original time series:

$$W_n(s) = \sum_{m=0}^{N-1} x_m \psi^* \left[\frac{(m-n)\Delta t}{s} \right] \tag{9}$$

where the asterisk (*) denotes complex conjugate. The above sum can be evaluated for various values of the scale s (usually taken to be multiples of the lowest possible frequency), as well as all values of n between the start and end dates.

It is possible to compute the wavelet transform in the time domain according to (9). However, it is much simpler to use the fact that the wavelet transform is the convolution between the two functions X and ψ, and to carry out the wavelet transform in Fourier space using the Fast Fourier Transform (FFT). In the Fourier domain, the wavelet transform is :

$$W_n(s) = \sum_{k=0}^{N-1} \bar{X}_m \bar{\Psi}^*(s\omega_k) e^{i\omega_k n\Delta t} \tag{10}$$

Unlike the convolution, the FFT method allows the computation of all n points simultaneously, and can be efficiently coded using any standard FFT package.

Wavelets coefficients allow an estimation of the signal power. The global wavelet spectrum, defined as the time average over a series of p-wavelet powers, can be expressed as [8]:

$$\bar{W}^2(p) = \frac{1}{N} \sum_{k=0}^{N-1} |W_k(p)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\bar{X}_k \times \bar{\Psi}_k(p)|^2 \tag{11}$$

A sum of products of Fourier coefficients calculated in a FFT32 routine for ADC data (x_n) in each clock cycle with pre-calculated Fourier coefficients of a reference wavelet gives an estimation of the signal power for selected type of the wavelet. Only a single FFT32 routine for the on-line calculation of Fourier coefficients for data is needed. Fourier coefficients for various wavelets can be calculated earlier and be available for final power estimation as constants.

A fundamental limitation for the on-line wavelet analysis in the FPGA is an amount of embedded DSP multipliers. A multiplication by an utilization of logic elements is rather inefficients. The Quartus® II environment for an Altera® FPGA programming provides parametrized FFT routines with various architectures: streaming, variable streaming, burst and buffered burst. However, all routines deliver the FFT coefficients in a serial form (Figure 4). No any Altera® routine allows calculating all FFT coefficients simultaneously.

If FFT coefficients are spread in time, the wavelet transform can be also calculated in a serial way (in a single clock cycle only a single pair of $\bar{X}_n$ is multiplied by a single pair of ψ^*), however, a product will strongly depend on a relative position of $\bar{X}_n$ and ψ^*. If the variables are shifted between themselves, even strong signal may give a negligible final contribution. Some additional procedure is needed, which could tune a wavelet transform regarding to the Fourier transform of ADC samples.

This problem can be automatically solved if all Fourier coefficients were provided simultaneously in each clock cycle. A synchronous multiplication with Fourier coefficients of wavelets would give required power estimation independently of any relatively configurations of these variables. The Fourier coefficients of selected wavelets are fixed, a sliding window of N ADC samples gives all Fourier coefficients in each clock cycle. This assures that for some set of samples (if a signal appears) the product of both transforms may give a significant contribution and may be used as a trigger.

The radio signal is spread in a time interval of an order of couple hundred nanoseconds, most of registered samples gave a time interval below 200 ns. The frequency window in the atmosphere, where a signal suppression is on an acceptable level (the atmosphere is relatively transparent) is ca. 30-80 MHz. According to the Nyquist theorem the sampling frequency should be at least twice higher than the maximal frequency in a investigated spectrum. The anti-aliasing filter should have the cut-off frequency of ca. 85 MHz. Taking into account some width of the transition range for the filter (from pass-band to stop-band) the final sampling frequency should not be lower than 180 MHz (200 MHz in our considerations). This frequency corresponds to 5 ns between rising edges of the clock.

The interval of 160 ns (estimated as sufficient time interval for radio signals) requires 32-point Fourier transform calculated in each clock cycle.

5. General algorithm

Let us consider a DFT $\bar{X}$ of dimension N

$$\bar{X}_k = \sum_{n=0}^{N-1} x_n W^{nk} \quad where \quad W = e^{-2i\pi/N} \quad and \quad k = 0, \ldots N-1 \tag{12}$$

If N is the product of two factors, with $N = N_1 N_2$, the indices n and k we can redefined as follows: $n = N_1 n_2 + n_1$,

where $n_2 = 0,\ldots,N_2\text{-}1$ and $n_1 = 0,\ldots,N_1\text{-}1$, $k = N_2 k_1 + k_2$, $k_2 = 0,\ldots,N_2\text{-}1$ and $k_1 = 0,\ldots,N_1\text{-}1$

$$\bar{X}_{N_2 k_1 + k_2} = \sum_{n_1=0}^{N_1-1} W^{N_2 n_1 k_1} W^{n_1 k_2} \times \sum_{m_2=0}^{N_2-1} x_{N_1 m_2 + m_1} W^{N_1 m_2 k_2} \tag{13}$$

For the Radix-2 algorithm: $N = 2^t$, $N_1 = 2$ and $N_2 = 2^{t-1} = N/2$. Hence,

$$\bar{X}_k = \sum_{n=0}^{N/2-1} (x_{2n} + W^k x_{2n+1}) W^{2nk} \tag{14}$$

If we split the sum as follows

$$\bar{X}_k = \sum_{n=0}^{N/4-1} x_{2n} W^{2nk} + \sum_{n=N/4}^{N/2-1} x_{2n} W^{2nk} + + \sum_{n=0}^{N/4-1} x_{2n+1} W^{2nk} + \sum_{n=N/4}^{N/2-1} x_{2n+1} W^{2nk} \tag{15}$$

and afterwards, if we redefine indices and group the sums, we get

$$\bar{X}_k = \sum_{n=0}^{N/4-1} (x_{2n} + (-1)^k x_{2(n+N/4)}) W^{2nk} + + W^k (\sum_{n=0}^{N/4-1} (x_{2n+1} + (-1)^k x_{2(n+N/4)+1}) W^{2nk}$$

$$\tag{16}$$

We can introduce the new set of variables defined for n = 0,...,N/4-1 as follows:

$$A_{2n} = x_{2n} + x_{2n+N/2} \qquad A_{2n+1} \quad = x_{2n+1} + x_{2n+1+N/2} \qquad (17)$$
$$A_{2n+N/2} = x_{2n} - x_{2n+N/2} \qquad A_{2n+1+N/2} = x_{2n+1} - x_{2n+1+N/2}, \qquad (18)$$

we get

$$\bar{X}_k = \sum_{n=0}^{N/4-1} (A_{2n} + W^k A_{2n+1}) W^{2nk} \qquad (19)$$

$$\bar{X}_k = \sum_{n=0}^{N/4-1} (A_{2n+N/2} + W^k A_{2n+1+N/2}) W^{2nk} \qquad (20)$$

for k even and odd respectively.

x_n represent signals in time domain. They can be easily available from outputs of shift registers clocked synchronously with the ADC. The DFT coefficients $\bar{X}_k$ can be expressed by new set of variables A_m, Because A_m are simple linear combination of x_m, they can be calculated by typical adders (eqs.(17) and sub-tractors (eqs.(18) in a single clock cycle. The input values x_n are real and positive, since they represent the signal in the real time.

Coefficients of DFT in the real domain additional simplify due to the following symmetry:

$$Re(\bar{X}_k) = +Re(\bar{X}_{N-k}) \qquad Im(\bar{X}_k) = -Im(\bar{X}_{N-k}) \qquad (21)$$

The Radix-2 algorithm allows regrouping of inputs elements in the DFT expression in order to utilize some symmetries of Fourier coefficients. In a single step of the Radix-2 algorithm we can redefine the "new" set of variables by some mathematical expression of the "old" ones. This step will correspond to an elementary process in the pipeline chain. The redefinition of variables in eqs.(17) corresponds to the 1st stage of the pipeline. Splitting the sum (14) reduces of coefficient W^k set from 0,...,N-1 for input x_n to 0,...,$\frac{N}{2}$-1 for (19-20). The 1st stage utilizes the feature of the twiddle factors related to the 1st stage of the pipeline.

$$W_A = W^{N/2} = e^{-i\pi} = -1 \qquad (22)$$

So, the 1st stage can be implemented in a very simple way. The implementation of the multi-points algorithm requires multiple pipeline stages and apart from adders and sub-tractors also requires multipliers, which correspond to the W^k coefficients relating to the fractional "angle" $e^{-2ik\pi/N}$. The Radix-2 algorithm used in the next stage reduces again the abundance of W^k coefficients due to the next twiddle factors' related to the 2nd stage of the pipeline.

$$W_B = W^{N/4} = e^{-i\pi/2} = -i \qquad (23)$$

The W_B suggests the similar splitting structure in the 2nd pipeline stage as in the 1st one (minus in (23) as in (22)), however the imaginary unit imposes the DFT calculation separately

for their real and imaginary parts. If we split the sum in (20) similar as in (15), we get for k = 0,2,...,N-2

$$\bar{X}_k = \sum_{n=0}^{N/8-1} [(A_{2n} + (-i)^k A_{2n+N/4})W^{2nk} + +(A_{2n+1} + (-i)^k A_{2n+1+N/4})W^{(2n+1)k}] \quad (24)$$

Let us consider separately two subset of odd indices: k=4n and k=4n+2 (n = 0,...,N/4-1)

$$\bar{X}_{4p} = \sum_{n=0}^{N/8-1} [(A_{2n} + A_{2n+N/4})W^{8np} + (A_{2n+1} + A_{2n+1+N/4})W^{(2n+1)4p}] \quad (25)$$

Notice that $\bar{X}_0$ and $\bar{X}_{N/2}$ are real.

$$\bar{X}_{4p+2} = \sum_{n=0}^{N/8-1} [(A_{2n} - A_{2n+N/4})W^{8n(4p+2)} + (A_{2n+1} - A_{2n+1+N/4})W^{(2n+1)(4p+2)}] \quad (26)$$

If we introduce new variables

$$B_{2n} = A_{2n} + A_{2n+N/4} \qquad B_{2n+1} \qquad = A_{2n+1} + A_{2n+1+N/4} \quad (27)$$
$$B_{2n+N/2} = A_{2n} - A_{2n+N/4} \qquad B_{2n+1+N/2} = A_{2n+1} - A_{2n+1+N/4} \quad (28)$$

we get

$$\bar{X}_{4p} = \sum_{n=0}^{N/8-1} (B_{2n} + B_{2n+1}W^{4p})W^{8np} \quad (29)$$

$$\bar{X}_{4p+2} = \sum_{n=0}^{N/8-1} (B_{2n+N/4} + B_{2n+1+N/4}W^{4p+2})W^{8np+4n} \quad (30)$$

However, repeating the above procedure for odd indices related to the eq.(20) gives more complicated formulas, which cannot be simplified due to complex coefficients $W^{4(n+p)}$ (eq.31).

$$\bar{X}_{4p+q} = \sum_{n=0}^{N/8-1} W^{2n(4p+q)} [(A_{2n+N/2} + A_{2n+1+N/2}W^{4p+q}) \mp j(A_{2n+3N/4} + A_{2n+1+3N/4}W^{4p+q})] \quad (31)$$

where $\mp$ corresponds to q = 1,3 respectively. Next simplification is possible due to symmetries of trigonometric functions. However, general considerations give relatively complicated formulas, which seem to be unnecessary here.

6. 16-point algorithm

For N = 16 and odd indices we get

$$\bar{X}_{4p+q} = (A_8 \mp jA_{12}) + (i)^p(A_9(-1)^pW^q - iA_{15}W^{4-q})+$$

$$+ (-i)^{\frac{q-1}{2}}(-1)^pW^2(A_{10} \mp iA_{14}) \pm (i)^p(A_{11}W^{4-q} - iA_{13}(-1)^pW^q) \tag{32}$$

Since of $W^4 = -i$, all coefficients can be expressed as a linear combination of the complex base W^1, W^2, W^3

$$W^1 = e^{-i\frac{\pi}{8}} = cos(\frac{\pi}{8}) - i \cdot sin(\frac{\pi}{8}) = \alpha - i \cdot \beta \tag{33}$$

$$W^2 = e^{-i\frac{\pi}{4}} = cos(\frac{\pi}{4}) - i \cdot sin(\frac{\pi}{4}) = \gamma(1 - i) \tag{34}$$

$$W^3 = e^{-i\frac{\pi}{8}} = cos(\frac{3\pi}{8}) - i \cdot sin(\frac{3\pi}{8}) = \beta - i \cdot \alpha \tag{35}$$

Symmetries in (33-35) allow the following simplification. Notice that

$$W^2(A_{10} \mp iA_{14}) = \gamma(A_{10} \mp A_{14})(1 - i) \tag{36}$$

$$A_9(-1)^pW^q - iA_{15}W^{4-q} = \widehat{\mathcal{X}}[(-1)^pA_9 - A_{15}] - i\widehat{\mathcal{Y}}[(-1)^pA_9 + A_{15}] \tag{37}$$

$$A_{11}W^q - iA_{13}(-1)^pW^{4-q} = \widehat{\mathcal{Y}}[A_{11} - (-1)^pA_{13}] - i\widehat{\mathcal{X}}[A_{11} + (-1)^pA_{13}] \tag{38}$$

where $\widehat{\mathcal{X}} = \alpha, \beta, \widehat{\mathcal{Y}} = \beta, \alpha$ for q = 1,3.

We can extend the set of variables $(27 - 28)$ also to odd indices of $\bar{X}$

$$B_{8,12} = A_{8,12} \quad B_{9,15} = A_9 \pm A_{15} \quad B_{10,14} = A_{10} \pm A_{14} \quad B_{11,13} = {}_{11} \pm A_{13} \tag{39}$$

Formulae (39) show that the entire 2nd pipeline stage can be built also from only adders and sub-tractors. Signals $A_{8,12}$ have to be delayed in parallel shift registers in order to assure synchronization with adjacent ones.

For N = 16 the DFT coefficients can be expressed by the B_n variables as follows

$$
\begin{aligned}
Re(\bar{X}_0) &= & B_0 + B_1 + B_2 + B_3 && Re(\bar{X}_8) = B_0 - B_1 + B_2 - B_3 && (40)\\
Re(\bar{X}_4) &= & B_0 - B_2 &&&& (41)\\
Re(\bar{X}_2) &= & B_4 + \gamma \cdot (B_5 - B_7) && Re(\bar{X}_6) = B_4 - \gamma \cdot (B_5 - B_7) && (42)\\
Im(\bar{X}_2) &= & -B_6 - \gamma \cdot (B_5 + B_7) && Im(\bar{X}_6) = B_6 - \gamma \cdot (B_5 + B_7) && (43)
\end{aligned}
$$

$$Re(\bar{X}_{1,7}) = \quad B_8 \pm \alpha B_{15} + \gamma B_{14} \pm \beta B_{13} \qquad Re(\bar{X}_{3,5}) = \quad B_8 \pm \beta B_{15} - \gamma B_{14} \mp \alpha B_{13} \quad (44)$$
$$Im(\bar{X}_{1,7}) = \mp B_{12} - \beta B_9 \mp \gamma B_{10} - \alpha B_{11} \qquad Im(\bar{X}_{3,5}) = \pm B_{12} - \alpha B_9 \mp \gamma B_{10} + \beta B_{11} \quad (45)$$

The next, 3rd pipeline stage requires implementation of 10 multipliers calculating products from (44-45), 3 adders, 3 sub-tractors and 4 shift registers : according to the following formulae

$$C_0 = B_0 + B_2 \qquad C_1 = B_1 + B_3 \qquad C_2 = B_0 - B_2 \qquad C_3 = B_1 - B_3 \qquad (46)$$
$$C_5 = B_5 + B_7 \qquad C_7 = B_5 - B_7 \qquad (47)$$
$$C_4 = B_4 \qquad C_6 = B_6 \qquad C_8 = B_8 \qquad C_{12} = B_{12} \qquad (48)$$
$$C_{9A} = \alpha \cdot B_9 \qquad C_{11A} = \alpha \cdot B_{11} \qquad C_{13A} = \alpha \cdot B_{13} \qquad C_{15A} = \alpha \cdot B_{15} \qquad (49)$$
$$C_{9B} = \beta \cdot B_9 \qquad C_{11B} = \beta \cdot B_{11} \qquad C_{13B} = \beta \cdot B_{13} \qquad C_{15B} = \beta \cdot B_{15} \qquad (50)$$
$$C_{10} = \gamma \cdot B_{10} \qquad C_{14} = \gamma \cdot B_{14} \qquad (51)$$

The 4th stage utilizes 2 multipliers, 5 adders, 5 sub-tractors and 4 shift registers

$$D_{5,7} = \gamma \cdot C_{5,7} \qquad D_{0,1} = C_0 \pm C_1, \qquad D_{8,14} = C_8 \pm C_{14}, \qquad D_{10,12} = C_{10} \pm C_{12} \qquad (52)$$
$$D_9 = C_{9A} - C_{11B} \quad D_{11} = C_{11A} + C_{9B} \qquad D_{15} = C_{13A} - C_{15B} \qquad D_{13} = C_{13B} + C_{15A} \quad (53)$$
$$D_{2,3,4,6} = C_{2,3,4,6} \qquad (54)$$

Finally, the set of DFT $\bar{X}_k$ in the 5th stage coefficients is calculated by 6 adders and 6 sub-tractors supported by 4 shift registers.

$$Re\bar{X}_{0,4,8} = D_{0,2,1} \qquad Re\bar{X}_{1,7} = D_8 \pm D_{15} \qquad Re\bar{X}_{2,6} = D_4 \pm D_7 \qquad Re\bar{X}_{3,5} = D_{14} \mp D_{13} \quad (55)$$
$$Im\bar{X}_4 = -D_3, \qquad Im\bar{X}_{1,7} = \mp D_{10} - D_{11} \quad Im\bar{X}_{2,6} = \mp D_6 - D_5 \quad Im\bar{X}_{3,5} = \mp D_{12} - D_9 \quad (56)$$

Figure 9 shows the internal structure of the 16-point FFT algorithm. As shown later (compare Figure 15), this algorithm is higher optimized in comparison to a pure DiF approach.

The algorithm with the 16-point FFT was tested on the 3rd generation of the Auger surface detector Front-End Board (Figure 10) [9], [10]. The 1st [12] and the 2nd [13] generations of the Front-End Boards could not support the FFT algorithms due to a lack of FPGA resources. However, the FFT algorithm seems to be less efficient than the DCT approach. The DCT algorithm implemented into the 4th generation Front-End with the CycloneIII® EP3C40F324C7 (Figure 11) passed successfully tests on the field recognizing short peaks with an exponentially attenuated tails characteristically for signals generated by very inclined showers.

7. 32-point FFT algorithm

For 32-point Discrete Fourier Transform $\bar{X}$

$$\bar{X}_{k=0,...,31} = \sum_{n=0}^{31} x_n e^{-i\pi kn/16} \tag{57}$$

where x_n as samples from an ADC chip are real. The formula (57) can be split on two or more parts by rearranging of the sum and indices. The standard approach of a formula simplification is a Radix-2 Decimation-in-Time (DiT) (Figure 1a) or Decimation-in-Frequency algorithm (DiF) (Figure 1b) one.

For Radix-2 DiT, we get the formula 3. N-point DFT can be easily split on two N/2-point transforms. Outputs from DFT procedures are complex. So, a calculation of final DFT coefficients by using DiT algorithm requires the complex multiplication for final merging
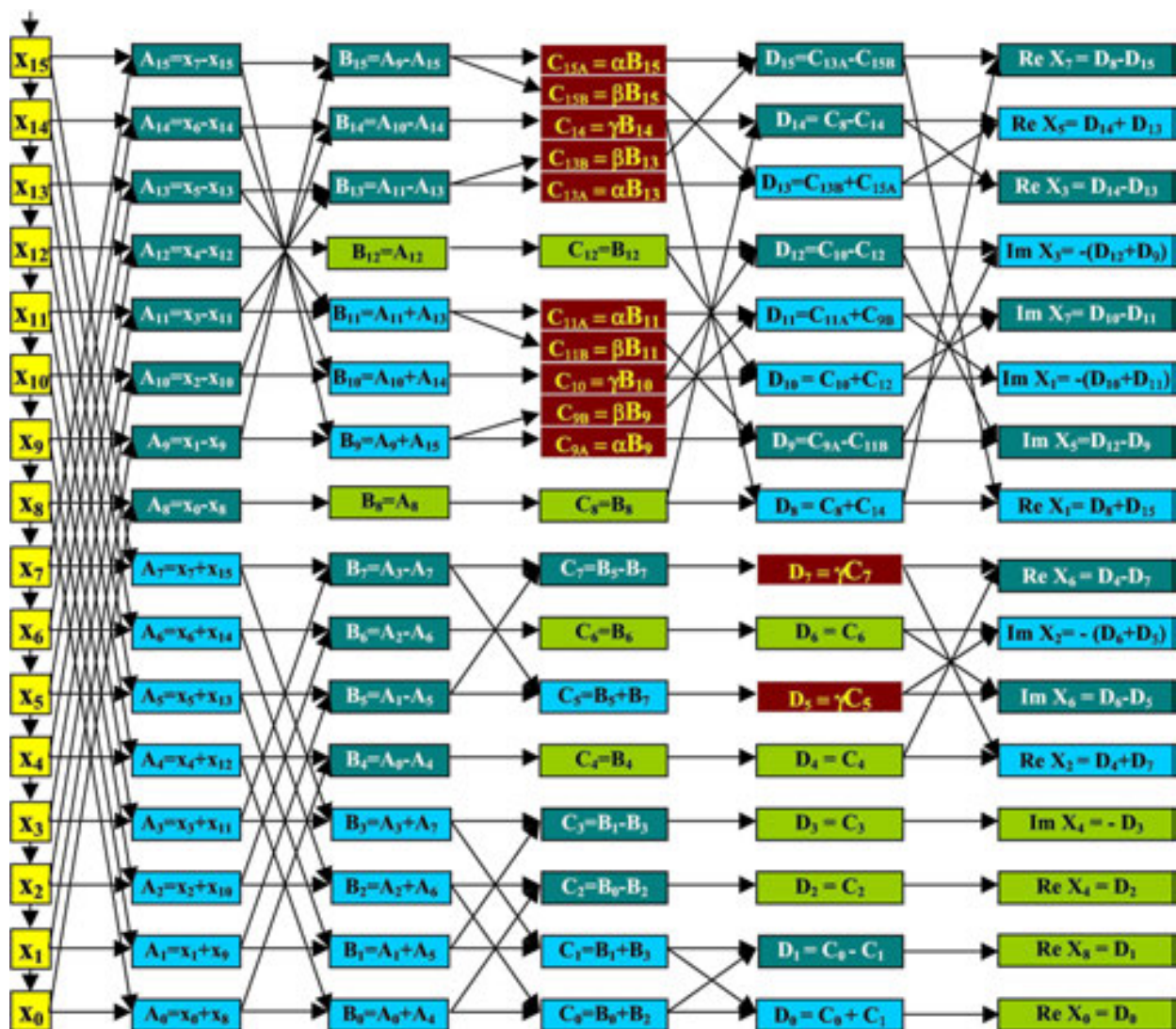

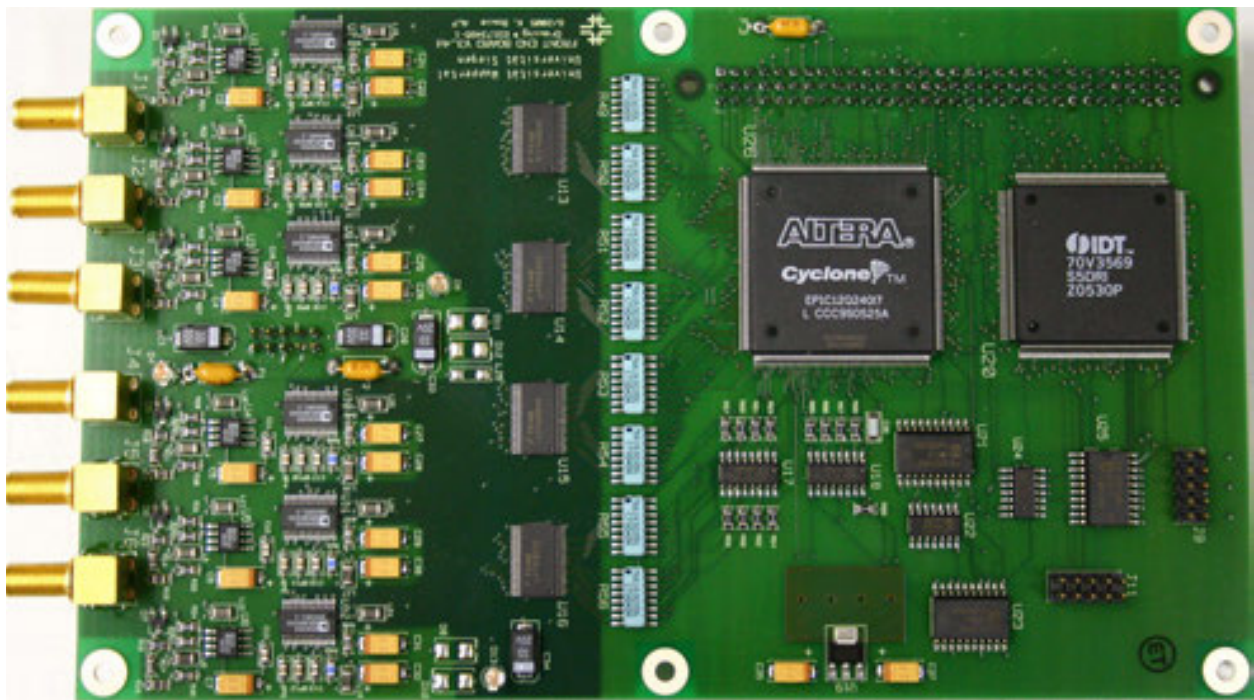**Figure 9.** A global pipeline internal structure of FFT_16 [11] .

Figure 10. The 3rd generation of the Front-End Board with Cyclone® FPGA EP1C12Q240I7 used in more than 800 surface detectors in the Pierre Auger Observatory on the Argentinean pampas. The EP1C12Q240I7 does not contain DSP blocks. The multipliers had to be implemented from logic elements according to the scheme on the Figure 9.
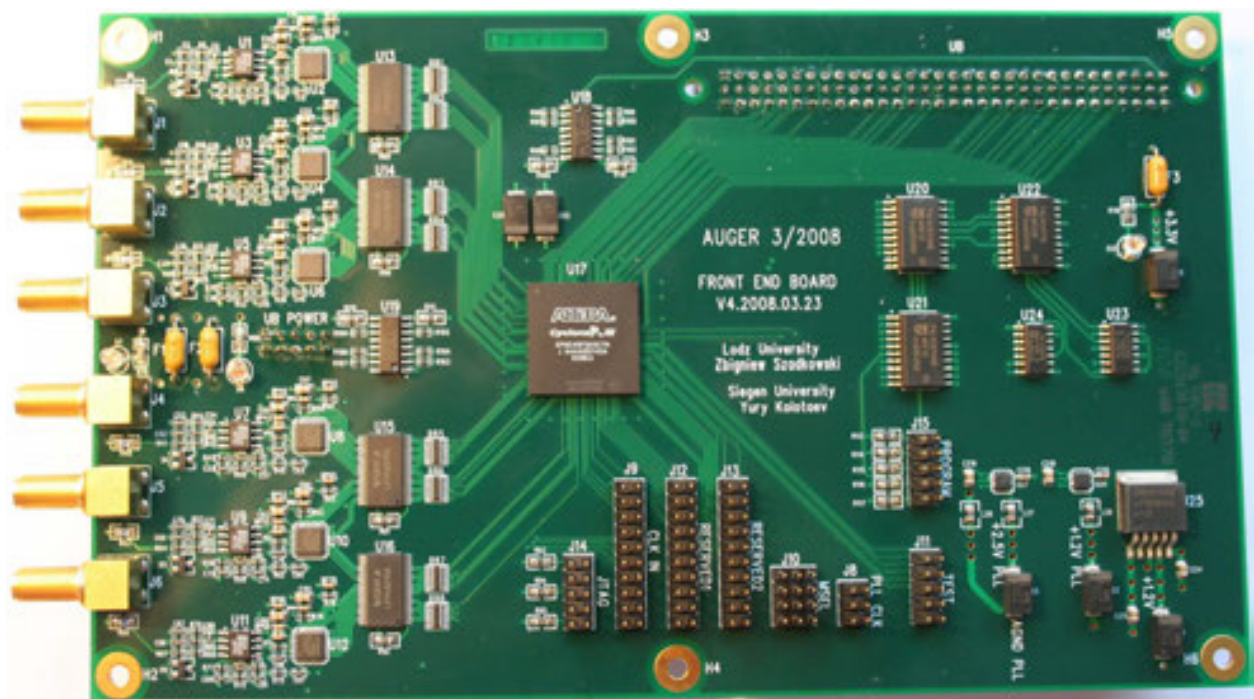

Figure 11. The 4th generation of the Front-End Board with CycloneIII® FPGA EP3C40F324I7. The EP3C40F324I7 contains DSP blocks and it is possible to implement even a sophisticated algorithm like DCT engines for a recognition of horizontal or very inclined showers. This board has been used also for preliminary testing of the wavelet trigger and the signal filtering based on a chain: FFT+Median filter+iFFT.

data from parallel DFT procedures with a lower order i.e. multiplication of twiddle factors W_N^k :

$$W_N^k = e^{-i\frac{2\pi k}{N}} \tag{58}$$

by $G[k]$ and $H[k]$ in Figure 1. Altera® provides a library routine of the complex multiplication in the FPGA (Figure 12a), however, for i.e. 16x16 bits operation requires 6 DSP embedded 9x9 multipliers even in most economical (canonical) mode. Generally, the complex multiplication in the FPGA is rather resource-spendthrift and if possible it should be replaced by the multiplication of real variables.
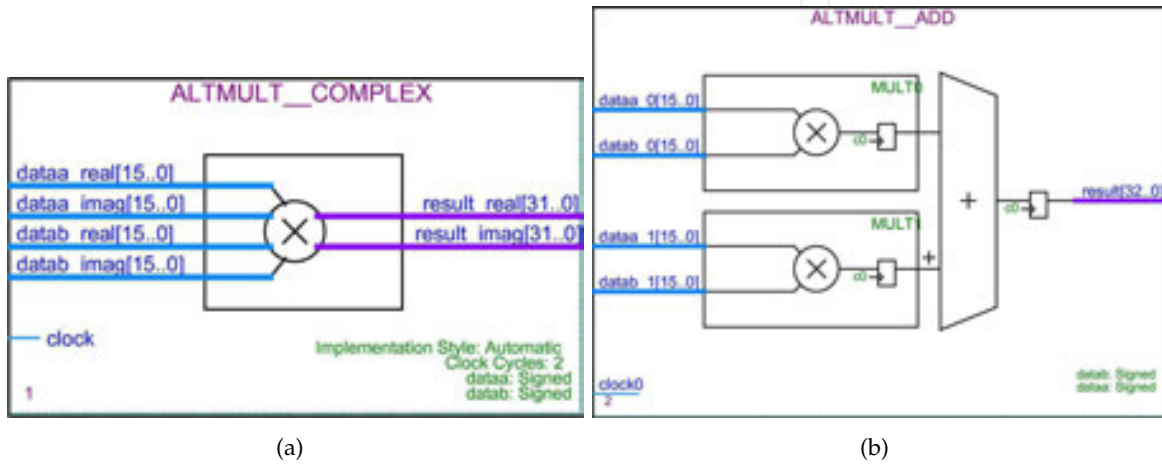

(a)   (b)

Figure 12. The ALTMULT_COMPLEX and ALTMULT_ADD procedures provided by Altera®. For a calculation of $|W_k|^2$, dataa_0 = datab_0 and dataa_1 = datab_1. The ALTMULT_ADD routine requires 4 DSP 9×9 multipliers. It is used in E_bin pipeline stage for odd FFT indices (Figure 17). Inputs $dataa\_0, 1$ are used for C_k, $datab\_0, 1$ for constants α, β, ξ, η, σ and ρ. The routine requires two clock cycles. Sub-products are registered in MULT0 and MULT1 DSP blocks, respectively. Thus, the sum appears in the next register stage.

For the Radix-2 DiF, we get the formula 4. The standard Radix-2 Decimation-in-Frequency algorithm (DiF) rearranges the DFT equation (57) into two parts: computation of the even-numbered discrete-frequency indices $\bar{X}(k)$ for k=[0,2,4,...,30] and computation of the odd-numbered indices k=[1,3,5,...,31]. This corresponds to a splitting N-point DFT into two k = N/2-point routines. The first corresponding twiddle factor is $e^{-i\frac{2\pi}{N}\frac{N}{2}} = -1$. The first operations are simple sums and subtractions of real variables (see Figure 1b). Each operation related to the consecutive twiddle factor will be performed in a single clock cycle.

The algorithm of Decimation in Frequency used for the 32-point DFT allows splitting eq. 57 as follows:

$$\bar{X}_{k=2p} = \sum_{n=0}^{15} A_n e^{-i\pi kn/8} \quad \Rightarrow \quad FFT16_{even} \tag{59}$$

$$\bar{X}_{k=2p+1} = \sum_{n=0}^{15} A_{n+16} e^{-i\pi(2p+1)n/16} \tag{60}$$

$$A_n = x_n + x_{n+16} \qquad A_{n+16} = x_n - x_{n+16} \qquad n = 0, 1, ..., 15 \tag{61}$$

The next twiddle factors are:

$$W_B = e^{-i\pi/2} = -i \qquad W_C = e^{-i\pi/4} = \gamma(1-i) \qquad W_D = e^{-i\pi/8} = \alpha - i\beta \qquad (62)$$

$$W_E = e^{-i\pi/16} = \xi - i\eta \qquad W_F = e^{-3i\pi/16} = \sigma - i\rho \qquad (63)$$

$$\gamma = cos(\pi/4) \qquad \alpha = cos(\pi/8) \qquad \xi = cos(\pi/16) \qquad \sigma = cos(3\pi/16) \qquad (64)$$

$$\beta = sin(\pi/8) \qquad \eta = sin(\pi/16) \qquad \rho = sin(3\pi/16) \qquad (65)$$

The scheme developed on the pure Radix-2 Decimation in Frequency algorithm is presented in Figure 15. The algorithm takes into account only FFT coefficients with indices k = 0,...,15. Due to real input data ($x_{0,...31}$) the higher FFT coefficients have well known symmetry : $Re\bar{X}_{32-n} = Re\bar{X}_n$ and $Im\bar{X}_{32-n} = -Im\bar{X}_n$ ($n > 0$). The calculation of $\bar{X}_{0,...15}$ according the pure Radix-2 DiF algorithm requires 8 pipeline stages. For $\bar{X}_{0,4,8,12,16}$ 2 pipeline stages are necessary only for a synchronization.

According to the eq. (59) all $\bar{X}_{0,2,4,...,14}$ with even indices could be calculated by the algorithm presented in [11]. Variables x_n in Figure 2 in [11] were be replaced by variable of A_n according to eq. (61). An application of a modified algorithm reduces an amount of 9×9 multipliers from 12 to 10 only and shorten a pipeline chain on stages (the last 2 stages are simple registers for synchronization) (see Figure 16).

Let us notice that for the odd indices stages B and C for k=16,...,19 and k = 24,...27 are pure delay lines, while for neighboring indices k=20,...,23 and k = 28,...31 mathematical operation are performed in a cascade. Let us multiply $A_{16,...19}$ and $A_{24,...27}$ by the factor $\lambda = \gamma^{-1}$. Then to adjust variables in the C stage for odd FFT coefficients (for k = 20,21,22,33 and k = 28,29,30,31)

$$C_k = \lambda \times \gamma = B_k \qquad (66)$$

Thus, by such a redefinition, The C stage for the odd FFT indices is a pure pipeline stage. It can be removed with one of pipeline stage for the even FFT indices. In order to come back to the correct values coefficients in F stage can be simple redefined

$$\alpha' = \gamma \times \alpha \quad \beta' = \gamma \times \beta \quad \xi' = \gamma \times \xi \quad \eta' = \gamma \times \eta \quad \sigma' = \gamma \times \sigma \quad \rho' = \gamma \times \rho \qquad (67)$$

but for indices k = 16, 20, 24 and 28 we have to use additional 4 multipliers. Nevertheless, at this cost we save one pipeline stage and depending on a width of buses in the final FFT coefficients we save at least of 1000 logic elements.

We can save a next pipeline stage and more ca. 1000 logic elements but again at the cost of additional utilized multipliers. The algorithm used for indices k = 2,6,10,14 is neither Decimation in Time nor Decimation in Frequency. The eq. (60) can be rewritten as follows:

$$\bar{X}_{k=2p+1} = A_{16} + A_{24} + \sum_{n=1}^{7} (cos\phi B_{n+24} - isin\phi B_{n+16}) \qquad \phi = \frac{\pi(2p+1)n}{16}$$

$$B_{16,24} = A_{16,24}$$
$$B_{n+16} = A_{n+16} + A_{32-n} \qquad n = 1,...,7 \tag{68}$$
$$B_{n+24} = A_{n+16} - A_{32-n} \qquad n = 1,...,7$$

A development of the algorithm according to eq. (22) would allow a reduction of the next pipeline stage, but unfortunately at the cost of additional 16 ALTMULT_ADD routines (64 DSP blocks) (see Figure 12b).

If the speed is not a factor, sums of products in the E_bin routine can be performed in a single clock cycle instead of two cycles as shown on Figure 17. Thus, $D_{26,20,24,28}$ shift registers are not necessary and can be removed. A shorter chain for the odd indices allows removing also the last pipeline chain for even indices and saving totally more than 1000 logic elements without the cost of additional multipliers. However, we should be aware, that a registered performance significantly decreases from ca. 220 MHz to only 158 MHz for EP3C120F780C7.

8. Wavelet power calculation

The reference wavelets are real, however, their Fourier transform are already complex. An elementary product from eq. (11) is a product of two complex numbers: Fourier coefficients of data and Fourier coefficient of a reference wavelet. The simplest way is to use the Altera$^{®}$ routine from Figure 12. However, due to a fact that the wavelet Fourier coefficients are predefined constant and finally we are going to calculate a module of a complex product as well as $|W \times \Psi|^2 = |W|^2 \times |\Psi|^2$, we can calculate only $|W|^2$ and next as real number multiply by a next real $|\Psi|^2$.

The FFT32 routine from Figure 17 utilizes 96 DSP 9×9 multipliers. For a calculation of $|W_k|^2$, the ALTMULT_ADD routine utilizes 4 DSP 9×9 multipliers for each index k, totally 60 ($|W_0|$ is trivial). $|W_k|^2 \times |\Psi_k|^2$ products use next 30 DSP 9×9 multipliers.

This algorithm can be implemented only in very powerful modern FPGA chips. The FPGA families ACEX$^{®}$ or Cyclone$^{®}$, currently used in surface detectors, do not contain DSP blocks. Even CycloneIII$^{®}$ EP3C40F324I7 [14] used for DCT trigger tests ([15], [16]) does not consist of a sufficient amount of DSP blocks to implement the wavelet trigger.

The biggest FPGAs from the CycloneIII$^{®}$ EP3C120F780C7 (Figure 13) and CycloneIV$^{®}$ EP4CE115F29C7 (Figure 14) families with 576 and 532 DSP multipliers, respectively, allow the implementation of the FFT32 routine (96 DSP blocks) + "Module" block (60 DSP blocks) + 14 or 11 "engines" (30 DSP blocks each) simultaneously for a power estimation of 14 or 11 various reference wavelets, respectively.

Table 2 shows results calculated and measured in the Altera$^{®}$'s development kit DK-DSP-3C120N for various variants for Cyclone$^{®}$ III EP3C120F780C7 (a heart of this development kit). Results do not fully agree with our expectations. A reduction of a single pipeline stage decreases a resource occupation on ca. 410 (not 640) logic elements. This

Figure 13. The test system based on a development kit with Altera® CycloneIII® FPGA EP3C120F780C7 supported by two daughter boards: AD/DA Data Conversion Card (left) with two ADCs (150MHz sampling) and two DACs (250 MHz), as well as the Industrial Communication Board (ICB-HSMC)(right) allowing a connection via the galvanic isolated RS485 ports.
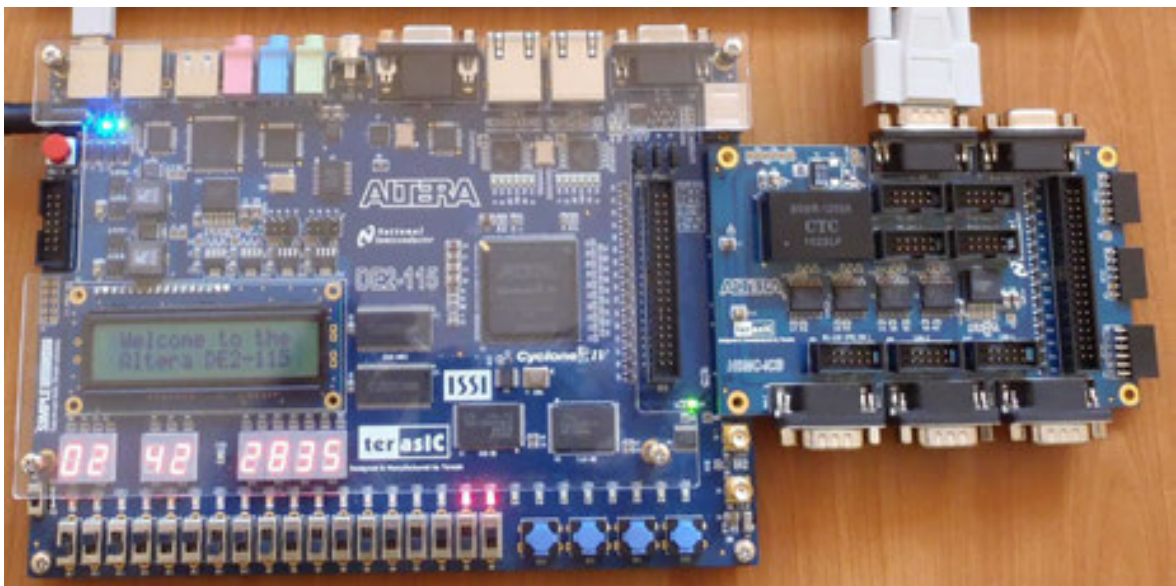

Figure 14. Test system based on a development kit with Altera® CycloneIV® EP4CE115F29C7 supported by ICB-HSMC daughter board.
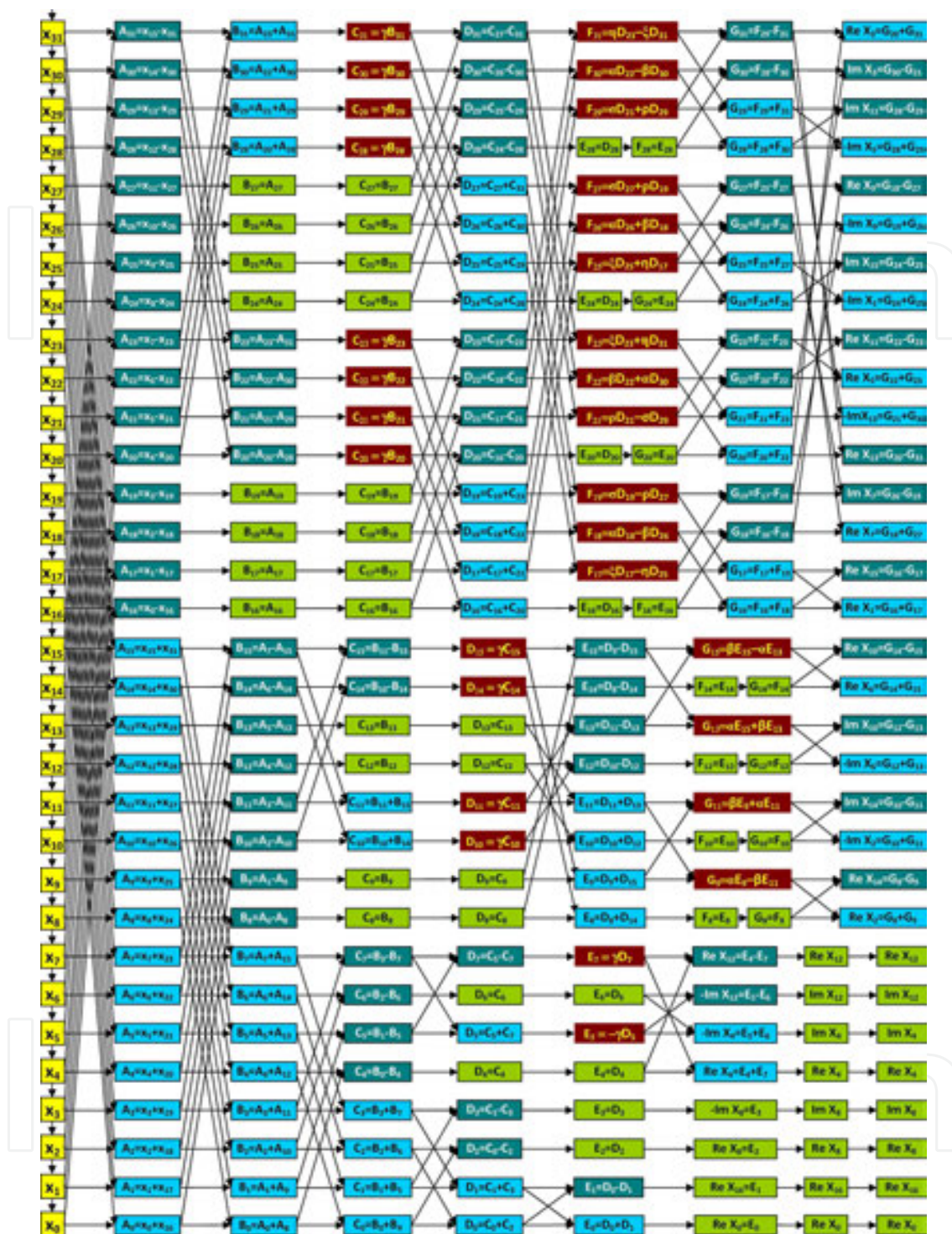
Figure 15. An internal structure of the FFT32 FPGA procedure. The algorithm uses 14 single clock-cycle multipliers (i.e. $F_7 = \gamma D_7$ - each utilizes two 9x9 DSP multipliers) and 16 two clock-cycles multipliers (i.e. $N_7 = \beta G_7 - \alpha H_7$ - each utilizes four 9x9 DSP multipliers). Totally, the algorithm needs 92 9x9 DSP multipliers.

may be due to optimization processes performed by the Quartus® II compiler to achieve the maximal registered performance. Nevertheless, for all comparisons the speed in the "optimized" design is higher than for the "pure DiF". For a development of wavelet engines the "optimized variant has been selected as potentially faster.
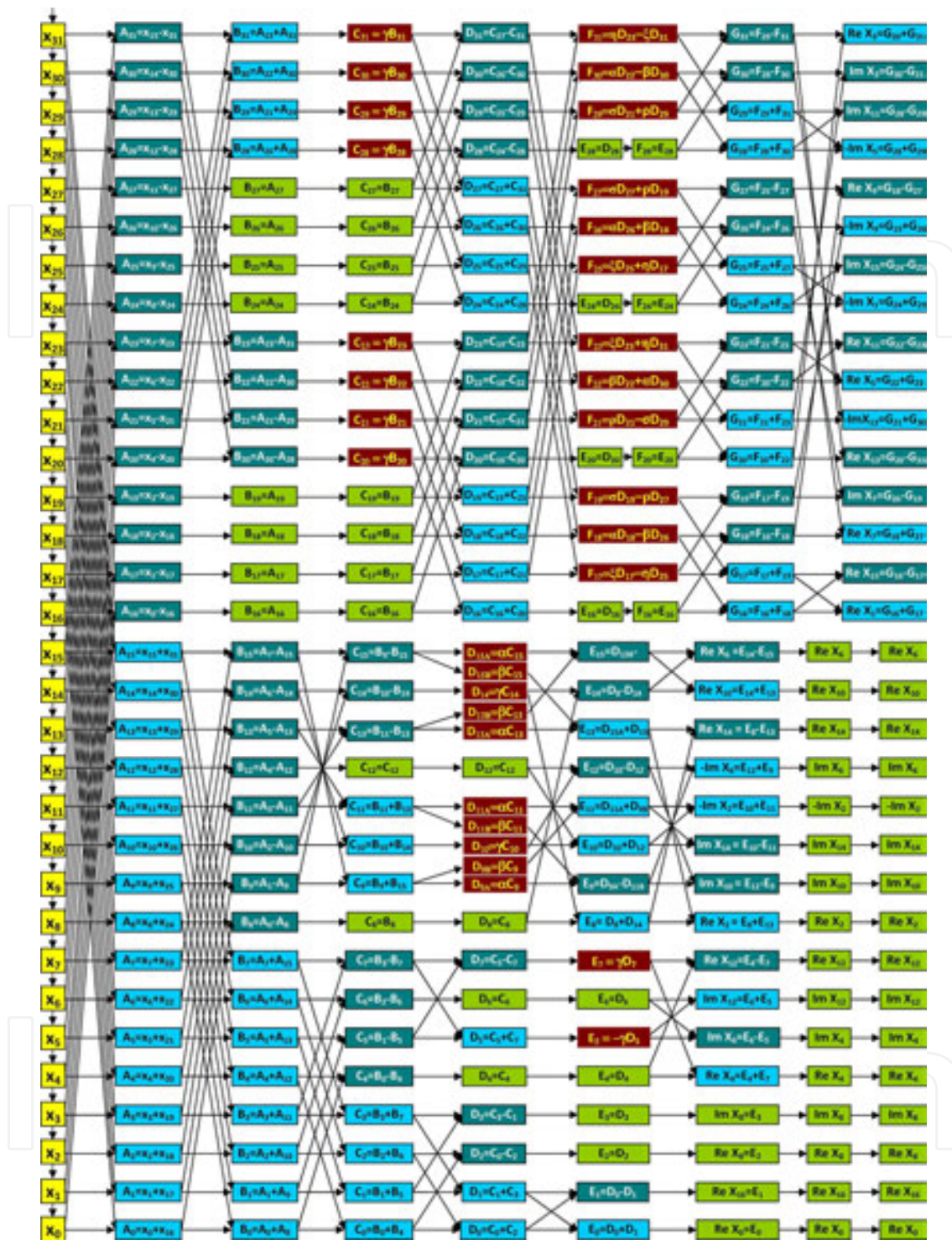
Figure 16. A modified structure for $\bar{X}_{2,6,10,14}$ allowing a reduction of two 9×9 multipliers and shorten a pipeline chain on two stages (shift registers still used for synchronization).

The Quartus® II compiler estimated a power consumption for the core, a static mode and for the I/O sector. As possible, the output of registers were multiplexed to reduce an amount of output pins (all pins were achieved to HSMC connectors on the development board). According to expectation the power for I/O increase ca. linear with a number of used pins. The static power consumption is on a level ~100 mW. It is a reasonable level. In comparison
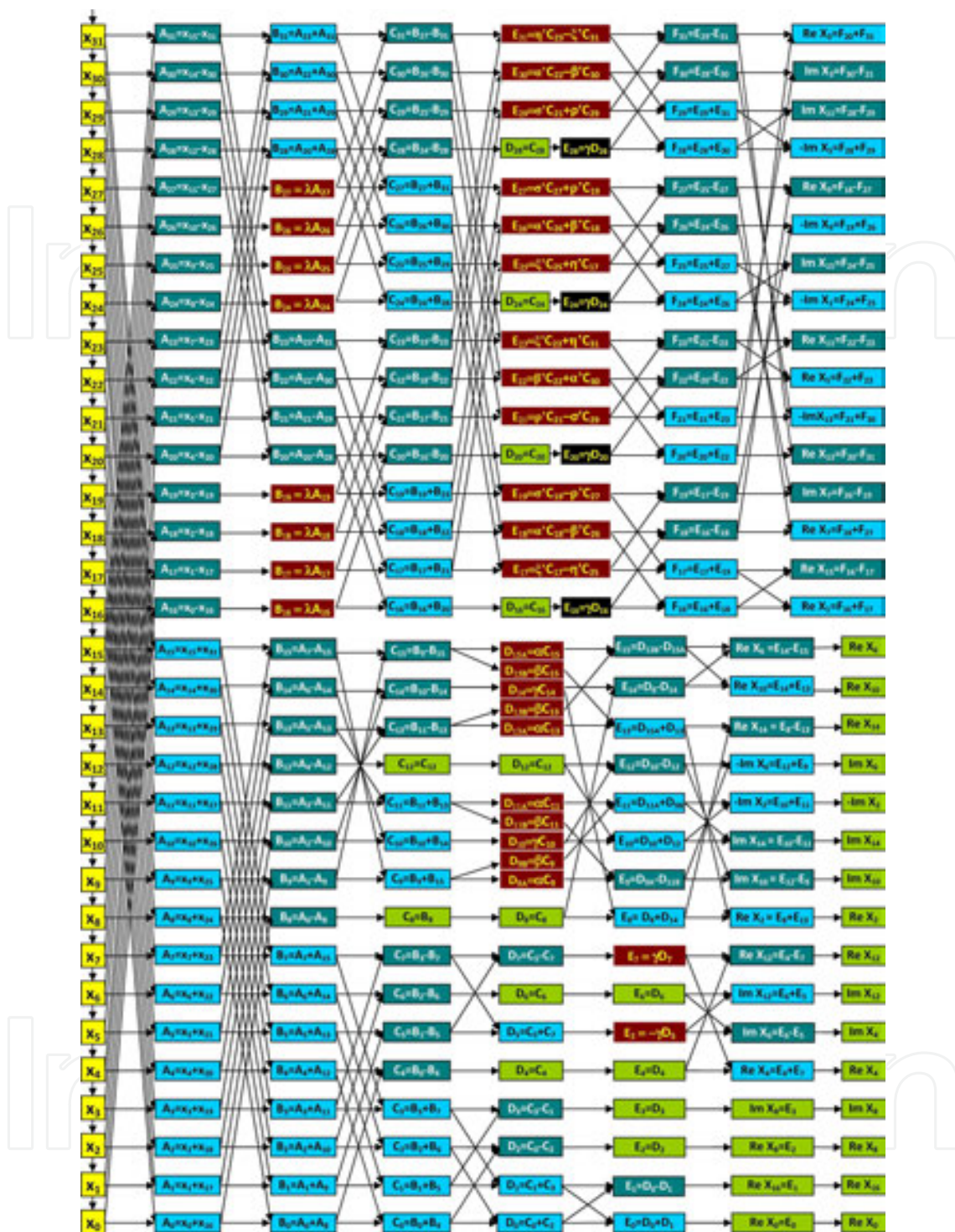
Figure 17. An optimized structure with a reduced a single pipeline stage at the cost of only 4 additional multipliers (8 DSP 9×9 blocks).

the Stratix® III chips have a huge power consumption in a static mode of ~ 600 mW, which significantly limited their application in systems supplied from solar panels. The power consumption for the "optimized" variant is ~ 35 mW higher than for the "pure DiF" solution. The additional 35 mW is not a factor, if it allows an improvement of the safety margin for the register performance. The EP3C120F780C7 allows the implementation of 14 wavelet engines.

config	logic elements	DSP	pins	fmax (MHz)	power sim. core (mW)	power sim. I/O (mW)	power mea. core (mW)	power mea. I/O (mW)
pure FFT32 pure DiF	4712 - 4%	92 - 16%	25 - 5%	236	557	65	580	170
pure FFT32 optimized	4301 - 4%	96 - 16%	25 - 5%	241	589	65	588	170
plus Module pure DiF	4990 - 4%	152 - 26%	25 - 5%	245	750	68	779	170
plus Module opt	4541 - 4%	156 - 27%	25 - 5%	246	787	68	783	170
1 wavelet - 24-bit opt	4726 - 4%	186 - 32%	29 - 5%	235	861	88	840	240
1 wavelet - 16-bit opt	4265 - 4%	186 - 32%	21 - 4%	228	814	66	790	170
4 wavelets - 16-bit opt	5478 - 5%	276 - 48%	81 - 15%	212	1134	215	1040	240
8 wavelets - 16-bit opt	5967 - 5%	396 - 69%	161 - 30%	204	1591	413	1363	360
12 wavelets - 16-bit opt	7060 - 6%	516 - 90%	241 - 45%	208	1980	612	1691	478

Table 2. Resources Occupancy and Power Consumption for the Cyclone III FPGA - EP3C120F780C7 for 200 MHz PLL Global Clock

Family	FPGA	config	logic elements	DSP	Slack Fast (ns)	Fmax Slow 0°C (MHz)	Fmax Slow 85°C (MHz)
Cyclone IV	EP4CE115F29C7	12 wavelets	7120 - 6%	516 - 97%	2.594	234	214
Cyclone V	5CGXFC7D6F31C6	12 wavelets	6933 - 6%	156 - 100%	2.111	**195**	**196**
Cyclone V	5CGXFC7D6F31C6	4 wavelets	3177 - 3%	111 - 71%	2.169	227	228

Table 3. Resources Occupancy and Timing for the Cyclone® IV and Cyclone® V FPGAs for 200 MHz PLL Global Clock

A design with 12 engines has been tested. The power consumption is on a level of ∼100-110 mW per the wavelet engine. It gives ∼2 W for 12 engines. This may be a challenge for an autonomous system supplied from solar panels.

Measurements of the power consumption for all considered variants show some discrepancies with simulations. The Measured power consumption for the core increases slower with new wavelet engines than simulations show. Almost 300 mW lower power taken by the FPGA (in comparison to simulations) for 12 engines gives optimistic predictions for the future applications. The power consumption for the core seems to be ca 15% overestimated in simulations. On the other hand, the power consumption for the I/O section is unpredictable much higher than for simulations. However, differences decrease with a higher amount of active pins. This, actually, is not a problem, I/O pins have been attached for test only. In real applications almost all variables are utilized as internal nodes. The power optimization is highly recommended.

Designs have been also implemented into EP4CE115F29C7 from the Cyclone® IV family of Altera® used in a development kit DE2-115 (Terasic). According to the Altera®'s specification, the power consumption for the Cyclone® IV family is 30% less than for the Cyclone® III one. However, the Terasic's development kit does not contain any system allowing a measurement of the power consumption on the board.

For the Cyclone®IV EP4CE115F29C7 timing shows a pretty good safety margin.

9. Spectral leakage

For the serial FFT processing the input data have to be chopped into blocks to be processed by the FFT routine. If signal pulses are located close to the border of a block, aliasing occurs. It manifests by a spurious contribution in the opposite border of the block and in the neighboring block as well. This effect may cause spurious pulses and has to be eliminated. The problem can only be solved, without introducing dead time between the blocks, by using an overlapping routine. Therefore the FFT engines have to be over-clocked. Practically for 1024-length blocks aliasing is reduced to a negligible level, when two blocks are overlapped during 64 time bins [7]. For parallel data processing, when all set of Fourier coefficients is available for each clock cycle FFT engines aliasing can be eliminated by a selection of a set of these coefficients not significantly affected. If a reduced set of Fourier coefficients is taken for data analysis, there is a possibility to increase an amount of wavelet engines for simultaneously analysis of more reference wavelets.

10. Design improvement

The new Altera$^{®}$'s FPGA family - Cyclone$^{®}$ V provides the industry's lowest system cost and power, along with performance levels that make the device family ideal for high-volume applications. A total power consumption compared with the previous generation (Cyclone$^{®}$ IV) is reduced up to 40%.

The biggest FPGA from the Cyclone$^{®}$ V E family 5CEA9 (with logic only without ARM-based hard processor system (HPS) contains 684 DSP 18×18 multipliers + 342 variable-precision DSP blocks (DSP blocks include three 9×9, two 18×19, and one 27×27 multiplier). Assuming roughly a single 18×18 multiplier is equivalent to two 9×9 ones, 5CEA9 could implement FFT32 + 18 engines for various 18 reference wavelets. However, the 5CEA9 FPGA is not yet available even for compilation (latest Quartus$^{®}$ II version 12.0). An estimation for 12 wavelet engines for 5CGXFC7 FPGA shows the scarcity of DSP blocks. Fast multipliers are replaced by logic elements, which significantly reduced the register performance for slow models, below our requirements. Nevertheless, if all multiplication all implemented in the fast DSP blocks (see Table 3 Cyclone$^{®}$ V for 4 wavelet engines only), timing is perfect. This allows anticipating also a perfect timing for the 5CEA9 chip. Expected total 58% less power consumption (30% and next 40% of reduction of power consumption from Cyclone$^{®}$ III to Cyclone$^{®}$ V) gives an estimation of 840 mW for 12 and 1260 mW for 18 wavelet engines, respectively. It is acceptable level of the power consumption for currently used supply systems in cosmic rays experiments.

11. Conclusions

The FFT32 routine has been successfully and cost-effectively implemented into the powerful FPGA EP3C120F780C7 from the Cyclone$^{®}$ III family used in a development kit DK-DSP-3C120N (Altera$^{®}$) and EP4CE115F29C7 from the Cyclone$^{®}$ IV family of Altera$^{®}$ used in a development kit DE2-115 (Terasic).

Nevertheless, both FPGAs from Cyclone$^{®}$ III and IV families were treated as an engineering test platform for a development of the algorithm and a timing verification. The prototype targeted for real detection of radio signals coming from air showers developing in the atmosphere will be built on a basis of Cyclone$^{®}$ V family.

The Pierre Auger Observatory is worldwide the largest cosmic ray experiment and operates its southern observatory since 2004. Results from Auger South have shown that the spectrum of cosmic rays has a characteristic cut-off at ca. 50 EeV; that events with higher energy arrive anisotropic; and that cosmic rays at highest energies are probably built from heavy nuclei. These results define the requirements for the next generation experiment: it needs to be considerably increased in size, it needs a better sensitivity to composition, and it should cover the full sky. Such a facility, AugerNext, will be specified within the next 3-5 years.

The innovative research studies are needed in order to prepare an AugerNext proposal fulfilling the demands. Requested resources are primarily focused in the areas: consolidation of the detection of cosmic rays using MHz radio antennas, proof-of-principle of cosmic rays microwave detection, testing the large-scale application of new generation photo sensors, generalization of data communication techniques, and developing a new technique of muon detection with surface arrays. Studies for such a next generation cosmic ray experiment and the utilization of detection methods are principle elements of the ASPERA /ApPEC roadmaps.

ASPERA-2 [18] supporting these efforts is the project of "The Innovative Research Studies for the Next Generation Ground-Based Ultra-High Energy Cosmic-Ray Experiment: AugerNext".

Acknowledgements

Author details

Zbigniew Szadkowski

* Address all correspondence to: zszadkow@kfd2.phys.uni.lodz.pl

University of Łódź, Department of Physics and Applied Informatics, Faculty of High-Energy Astrophysics, Łódź, Poland
The author is the member of the Pierre Auger Collaboration since 1999

References

[1] H. R. Allan, "Radio emission from extensive air showers", Prog. in Elem. Part. and Cos. Ray Phys., vol. 10, pp. 171, 1971.

[2] H. Falcke, P.W. Gorham, "Detecting radio emission from cosmic ray air showers and neutrinos with a digital radio telescope", *Astropart. Phys.* vol. 19, pp. 477-494, July 2003, ISSN: 0927-6505

[3] J. Abraham et al., [Pierre Auger Collaboration], "Properties and Performance of the Prototype Instrument for the Pierre Auger Observatory", *Nucl. Instr. Meth.*, ser. A, vol. 523, pp. 50-95, May 2004, ISSN: 0168-9002

[4] X. Bertou, P. Billoir, O. Deligny, C. Lachaud, A. Letessier-Selvon, "Tau Neutrinos in the Auger Observatory : A New Window to UHECR Sources ", astro-ph/ 0104452.

[5] http://www.altera.com/products/ip/dsp/transforms/m-ham-fft.html

[6] G. A. Dulk, W. C. Erickson, R. Manning, and J.-L. Bougeret, "Calibration of low-frequency radio telescopes using the galactic background radiation", *A&A*, vol. 365, pp. 294-300, Jan. 2001, ISSN (Print Edition): 0004-6361, ISSN (Electronic Edition): 1432-0746

[7] A. Schmidt, H. Gemmeke, A. Haungs, K-H Kampert, C. Rühle, Z. Szadkowski, "An FPGA Based Trigger and RFI Filter for Radio Detection of Cosmic Rays", *IEEE Trans. Nucl. Science.* vol. 58, no 4, pp. 1621-1627, Aug. 2011, ISSN: 0018-9499

[8] Z.Ge, Significance tests for the wavelet power and the wavelet power spectrum, *Ann. Geophys.*, 25, pp. 2259-2269, 2007, ISSN: 1593-5213

[9] Z. Szadkowski, K-H Becker, K-H Kampert, "Development of a new first level trigger for the surface array in the Pierre Auger Observatory based on the Cyclone™ Altera® FPGA", *Nucl. Instr. Meth*, vol. A545, pp. 793-802, June 2005, ISSN: 0168-9002

[10] Z. Szadkowski, et. al, "The 3rd Generation Front-End Cards of the Pierre Auger Surface Detectors: Test Results and Performance in the Field", *Nucl. Instr. Meth*, vol. A606, pp. 439-445, July 2009, ISSN: 0168-9002

[11] Z. Szadkowski, "16-point Discrete Fourier Transform based on the Radix-2 FFT algorithm implemented into Cyclone™ FPGA as the UHECR trigger for horizontal air showers", *Nucl. Instr. Meth*, vol. A560, pp. 309-316, May 2006, ISSN: 0168-9002

[12] Z. Szadkowski, D. Nitz, "Implementation of the first level surface detector trigger for the Pierre Auger Observatory Engineering Array", *Nucl. Instr. Meth*, vol. A545, pp. 624-631, June 2005 ISSN: 0168-9002

[13] Z. Szadkowski, "The concept of an ACEX® cost-effective first level surface detector trigger in the Pierre Auger Observatory", *Nucl. Instr. Meth*, vol. A551, pp. 477-486, Oct. 2005, ISSN: 0168-9002

[14] Z. Szadkowski. "Trigger Board for the Auger Surface Detector with 100 MHz Sampling and Discrete Cosine Transform". *IEEE Trans. Nucl. Science*, vol. 58, no 4, pp. 1692-1700, Aug. 2011, ISSN: 0018-9499

[15] Z. Szadkowski, "A spectral 1st level FPGA trigger for detection of very inclined showers based on a 16-point Discrete Cosine Transform for the Pierre Auger Experiments", *Nucl. Instr. Meth*, vol. A606, pp. 330-343, July 2009, ISSN: 0168-9002

[16] Z. Szadkowski "An optimization of 16-point Discrete Cosine Transform Implemented into a FPGA as a Design for a Spectral First Level Surface Detector Trigger in Extensive Air Shower Experiments", *Applications of Digital Signal Processing*, InTech, ISBN 978-953-307-406-1, Croatia, 2011.

[17] Z. Szadkowski, "FPGA implementation of the 32-point DFT for a wavelet trigger in cosmic rays experiments", Real Time Conference, Berkeley, CA, June 2012

[18] http://www.aspera-eu.org/