FPGA Based Signal-Processing for Radio Detection of Cosmic Rays

Adrian Schmidt, Hartmut Gemmeke, *Member, IEEE*, Andreas Haungs, Karl-Heinz Kampert, Christoph Rühle, and Zbigniew Szadkowski, *Member, IEEE*

***Abstract*—For the observation of ultra high-energy cosmic rays (UHECRs) by the detection of their coherent radio emission an FPGA based trigger and radio frequency interference (RFI) filter was developed. Using radio detection, the electromagnetic part of an air shower in the atmosphere may be studied in detail, thus providing information complementary to that obtained by water Cherenkov detectors which are predominantly sensitive to the muonic content of an air shower at ground. For an extensive radio detector array, due to the limited communication data rate, a sophisticated self trigger is necessary. However, radio signals in the frequency range of 30–80 MHz are significantly contaminated by RFI and human made distortions. The digitized signals are converted from the time to frequency domain by a FFT procedure, then a deconvolution and RFI-filters are applied to correct the frequency response and to suppress the RFI. Finally the filtered data is transformed back into the time domain by an iFFT, also generating an envelope as a base for the final self-trigger. To avoid leakage effect and to create an overlap of successive data blocks, trapezoidal windowing is applied with internal overclocking. The algorithms for two polarization channels have been successfully implemented in a single FPGA and tested in a prototype board with 180 MHz sampling rate, 16-bit dynamic range, and 12-bit resolution.**

***Index Terms*—AERA, cosmic rays, DSP, FPGA.**

I. Introduction

RESULTS from the Southern Pierre Auger Observatory, as well as the baseline design of the Northern Observatory, 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 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 developement are separated and deflected by the Earth's magnetic field, hence introduce an electromagnetic emission [1], [2]. 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 as given by simulations with REAS [3] and MGRM [4]. 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.

Several experiments like LOPES [5], CODALEMA [6], and MAXIMA [7], [15] have been deployed to study EAS radio emission. All three collaborations are working together at the Auger Engineering Radio Array (AERA) at the Pierre Auger Observatory. In contrast to the fluorescence technique with a duty cycle of about 12% (e.g. used in the Pierre Auger Observatory [8]), 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 dimension of AERA will be about 20 km^2. For such an area we expect about 5000 identified radio events per year, allowing a study of both scientific and technological questions. 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. In the following, we will describe one of the solutions developed for the AERA project, where we will start with the description of the necessary technical requirements and limitations.

II. Requirements and Limitations

The energy threshold of radio detection of cosmic rays is limited by the considerable radio background and noise. The very high level of RFI in the FM and short wave band has to be eliminated by a band pass filter-amplifier. Within the remaining receiver bandwidth of 30 to 80 MHz the noise at the quiet-rural environment of the Pierre Auger Observatory is dominated by the frequency dependent galactic noise [9] with noise temperatures of 5000 K at 60 MHz.

Manuscript received June 16, 2010; revised February 17, 2011; accepted March 31, 2011. Date of publication May 05, 2011; date of current version August 17, 2011. This work was supported in part by German Bundesministerium für Bildung und Forschung (BMBF) Grants 05A08VK1 and 05A08PX1.

A. Schmidt, H. Gemmeke, and C. Rühle are with the Karlsruhe Institute of Technology, Institute for Data Processing and Electronics, D-76344 Eggenstein-Leopoldshafen, Germany (e-mail: adrian.schmidt@kit.edu; hartmut.gemmeke@kit.edu; christoph.ruehle@kit.edu).

A. Haungs is with the Karlsruhe Institute of Technology, Institut für Kernphysik, D-76021 Karlsruhe, Germany (e-mail: andreas.haungs@kit.edu).

K.-H. Kampert is with the University of Wuppertal, Fachbereich C-Mathematik und Naturwissenschaften, D-42097 Wuppertal, Germany (e-mail: kampert@uni-wuppertal.de).

Z. Szadkowski was with the University of Wuppertal, D-76021 Karlsruhe, Germany. He is now with the Department of Physics and Applied Informatics, University of Łódź, 90-236 Łódź, Poland (e-mail: zszadkow@kfd2.phys.uni.lodz.pl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNS.2011.2141151

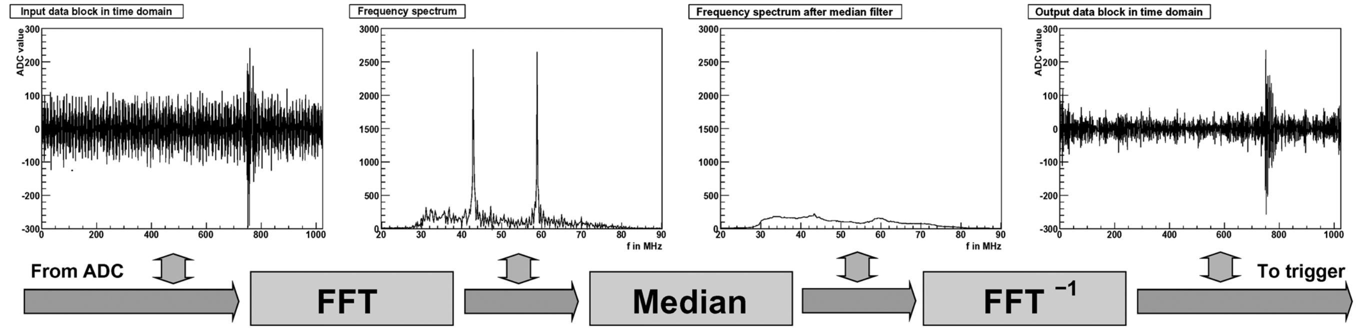

Fig. 1. A diagram showing a (FFT+Median filter+iFFT) chain cleaning the signal from the RFI contamination. The 1st graph shows a simulated input block of 1024 ADC samples, the 2nd—the absolute values of FFT components in the frequency domain in arbitrary units, the 3rd—FFT components "decontaminated" by the median filter and 4th—signal converted back to the time domain. The amplitude of the signal remains roughly the same and the noise is considerably reduced.

In addition to galactic noise, there is still a human made background. This background consists of continuous signals from a few radio and TV stations and transients produced by machines. Furthermore a steady running beacon transmitting in our wanted band is foreseen to allow a precise time calibration in the range of 1 ns [10]. This demanding noise situation requires a sophisticated trigger system to guarantee a low false trigger rate with a stable and low-level energy threshold.

For self-triggered measurements, the data is digitized and processed in real time by the powerful Altera Cyclone III FPGA chip EP3C80F780C6. 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 filter eliminates mono-frequent carriers, but broadband radio pulses from cosmic showers are not affected. With a second, inverse FFT (iFFT), signals are converted back to the time domain. This suppresses carriers down to the level of the surrounding background. At conditions as seen by LOPES in Karlsruhe the signal-to-noise ratio is increased by 20 to 200% depending on the activity level of radio and TV stations in our band, and thus improves the sensitivity of radio pulse detection, see Fig. 1.

At each station, the signals from two antennas with opposite polarization are recorded. Below 150 m core distance, our trigger-efficiency reaches at least 0.5 for showers of about 0.67 EeV [11]. Because of the high signal strength in the shower core, due to coherent radio emission, and the steep signal drop-off with distance, a high dynamic range of 16 bits is necessary.

According to the Nyquist theorem, the used 80 MHz band must be sampled with at least 160 MHz. Using 16-bit ADCs with such sampling rate would be challenging in respect of price, power consumption and PCB routing to keep a reasonable noise level. As a practical solution, we use 12-bit ADCs with 180 MSPS, fulfilling the sampling theorem with ease, and implementing a high and low gain channel to obtain the required dynamic range.

The necessary filtering accuracy requires at least 1024-point Fourier transforms. For 180 MHz sampling, this 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.

TABLE I
UTILIZATION OF RESOURCES FOR VARIOUS FFT ARCHITECTURES FOR A BLOCK LENGTH OF 1024 SAMPLES

architecture	LEs	memory kb	DSPs	max frequency MHz
12-bit streaming	3611	117	24	250
14-bit streaming	4201	137	24	250
12-bit variable streaming	5736	31	48	208
floating point	23000	72	128	111

The 2nd column shows required Logic Elements (LE) after compilation, the 3rd—memory bits, 4th—Digital Signal Processing (DSP) blocks, 5th—simulated maximum clock frequency (MHz), respectively.

III. OPTIMIZATION OF THE PURE FFT+IFFT CHAIN

The Altera FFT MegaCore offers so called streaming and variable streaming architectures calculating the FFT and iFFT in real-time, the transformation direction (FFT or iFFT) of an FFT engine can be specified. Both architectures implement a fixed point FFT whereas the variable streaming architecture can also be configured to use a floating point data representation. A comparison of resource occupancy and max. clock frequency of different architectures is given in Table I. For the floating point architecture, the max. clock frequency of 111 MHz is far to low. The goal is to implement the full trigger system with the discussed filters for two polarizations. Without the optimizations introduced later, each FFT/iFFT chain requires two engines.

In the following chapters, series and functions in small letters, e.g. $x_n, x(t)$, denote values in time domain, while series and functions in capital letters denote their Fourier transforms, e.g. $x(t) = F^{-1}(X(f))$.

A. The 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 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.

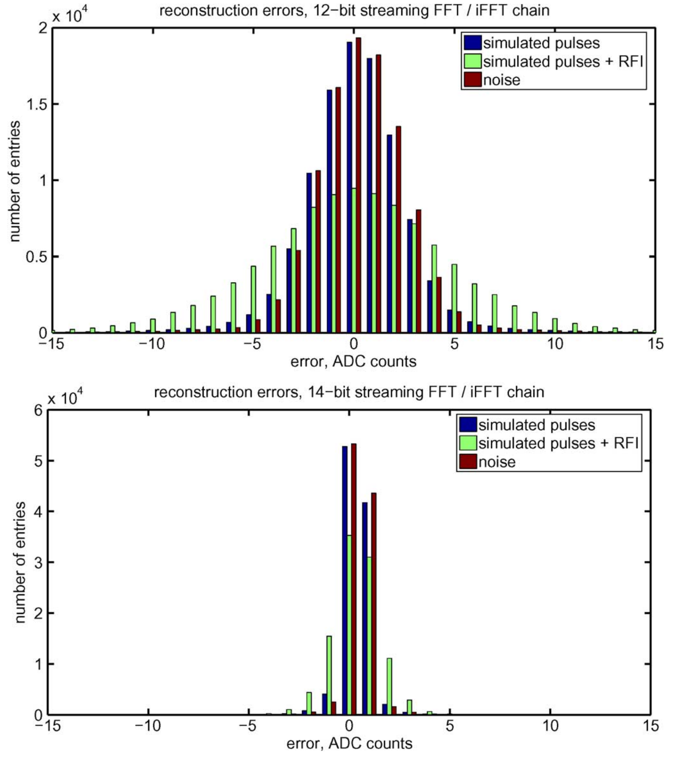

Fig. 2. Histograms of reconstruction errors for the streaming architecture (differences between the original ADC data and data after application of the 12-bit (above) and 14-bit (below) wide FFT and inverse FFT).

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_i (typical for noise background), but with not negligible pedestal the Fourier component X_0, $X_0 = \sum_{i=0}^{N-1} x_i$ may be relatively large whereas the $X_{n,n\neq 0}$ 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.

Reconstruction errors for the 1024-point transform of 100 REAS simulated pulses [3], also under the influence of RFI, and noise are shown in Fig. 2. Due to their AC-coupling, the ADC inputs have high-pass characteristics, so the pedestal was assumed to be already corrected to zero. The histogrammed values where calculated by subtracting the input values of the FFT/iFFT chain from its corresponding output values. At 12-bit resolution, the reconstruction error is about 2.6 ADC-counts standard deviation for noise and also for pulse signals. It rises to about 5.1 ADC-counts if the pulse signals are polluted by RFI, in this example 2 radio carriers with 50 ADC-counts amplitude each. This effect is also caused by the internal scaling of the FFTs. The X_{car} corresponding to the carrier frequencies become very large, hence the resolution left over for the $X_{n,n\neq car}$ decreases.

In order to preserve a higher resolution for the $X_{n,n\neq car}$ the bus width of the FFT routines can be increased to 14 bits. Then, the higher 12 bits of the FFTs/iFFTs input respectively output bus are used for the signal, while the lower 2 bits are connected to zero. This way, the signal at the iFFTs output becomes much larger than the quantization noise caused by the multiplication units in the FFT engines. The reconstruction error decreases to about 0.7 ADC-counts respectively 1.4 ADC-counts in the presence of RFI. Nevertheless, the 14-bit FFT engine is at the expenses of additional 590 LEs, while the max. clock frequency still keeps at 250 MHz, limited by the FPGAs clock tree.

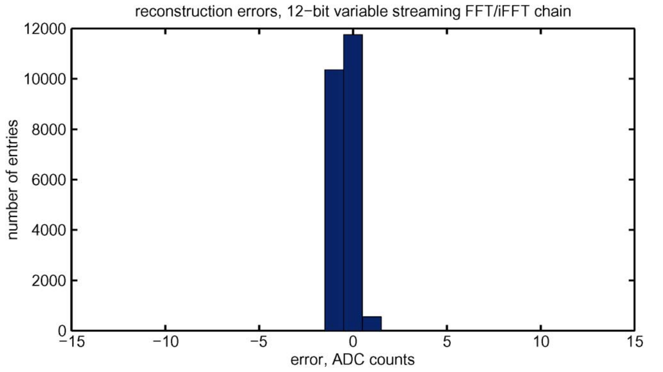

Fig. 3. Histogram of reconstruction errors for the FFT+iFFT routines with the variable streaming architecture. The reconstruction error is on the level of one LSB. No error contribution for $|ADC - counts| \geq 2$.

B. The Variable Streaming Architecture

A 12-bit input FFT routine with the variable streaming architecture yields 25-bit Re/Im Fourier components. Processing of both busses 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.

Fig. 3 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 we will see in the following sections, that this architecture's maximum clock frequency of roughly 200 MHz is too low. For this reason, the 14-bit streaming architecture is chosen for the FPGA implementation, as compromise between necessary speed and precision.

IV. Deconvolution

In order to suppress the strong man-made radio carrier signals below 30 MHz and above 80 MHz and to fulfill the Nyquist theorem, the signals of the antennas are filtered using an analog band filter before being sampled by the ADCs. The filters with the constant transfer function $H(f)$ are of 34th order. They have a non-constant group delay (Fig. 4), thus leading to dispersion and decrease of the amplitude of the output signal pulses $P(f)$. In the frequency domain, the resulting output signal of the filter can simply be calculated using (1):

$$P(f) = H(f) \cdot S(f) \tag{1}$$

Knowing $H(f)$ from measurements, it is now easy to get back the undispersed $S(f)$ by inverting (1). This operation can be implemented into the FPGA by placing a complex multiplication unit directly after the FFT engine which multiplies the output data of the engine with the precomputed coefficients of $1/H(f)$

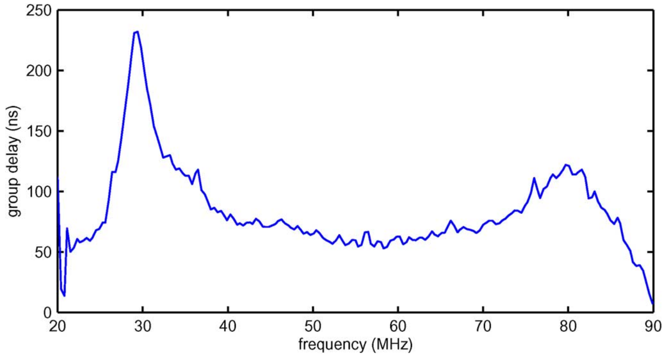

Fig. 4. Group delay caused by the analog filter.

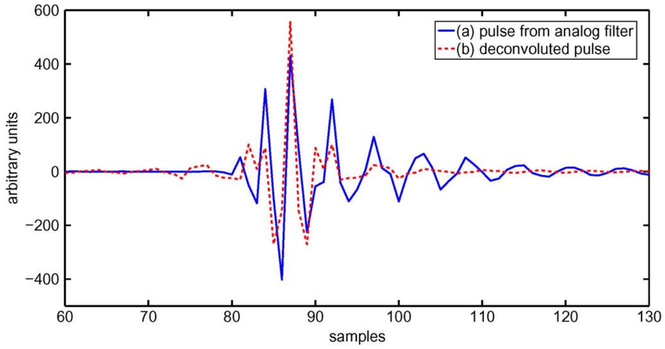

Fig. 5. Simulation of an input pulse disturbed by analog filtering (a), and the resulting pulse after deconvolution (b) at 180 MSPS. The bipolar pulse shape is dominated by the band pass filter from 30 to 80 MHz.

stored in a RAM. This way, an ideal, zero-phase band filter is constructed.

After transforming the signal back into the time domain, the amplitude of the de-convoluted pulses increases by about 20% compared to the input pulses (Fig. 5). Since the galactic and electronic noise is completely uncorrelated, the amplitude S/N ratio increases by the same amount.

V. Median Filter

The aim of the RFI filtering is to remove interference contaminations in the frequency domain. This can be done by applying a median filter with a sliding window on the absolute values of the Fourier components $|X_n|$. The median filter is a nonlinear filter, more used to remove the impulsive (salt and pepper) noise from images. The median $\tilde{X}_{n+5}$ of e.g. a sorted list of 9 elements $|X_n| \ldots |X_{n+9}|$ would be the 5th element. It is a more robust method than the traditional linear filtering because it preserves sharp edges, but it has much higher computational costs.

Both real and imaginary Fourier components $\Re(X_n), \Im(X_n)$, outgoing from the FFT routine will be scaled by the median filter output according to (2), with $\grave{X}_n$ as result, cleaned from interferences.

$$\Re(\grave{X}_n) + i \cdot \Im(\grave{X}_n) = (\Re(X_n) + i \cdot \Im(X_n)) \cdot \frac{\tilde{X}_n}{|X_n|} \qquad (2)$$

Numerous algorithms for median filters have been developed, mostly for image processing. The suggested architectures of data sorting can be divided into two types: parallel and serial. In the case of RFI filtering, a high-speed filter with flexible length is required. Preliminary estimations suggest the length of 9 cells. A 9-point median filter allows a removal of interference peaks with a width of 4 frequency-bins. With 180 MHz sampling and 1024 points, FFT transform corresponds to 1.4 MHz bandwidth, which should be enough to eliminate television carriers.

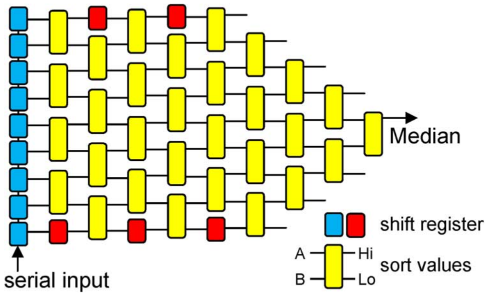

Fig. 6. Classical systolic array for sorting of 9 data buses with 30 full (two registers) and 5 reduced basic (one register) nodes.

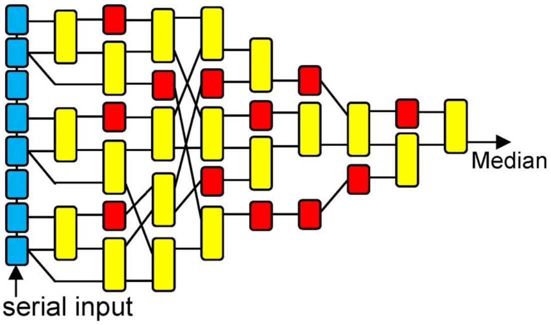

Fig. 7. The optimized structure of sorting algorithm with optimized pipelined synchronization with 19 full and 12 reduced basic nodes.

To calculate the median over n values, all n input values must be valid. Thus, for a block of data, the first and the last n/2 values, of the median filtered signal are not valid. As a substitute for the first respectively last n/2 values, we use the first respectively last valid value of the block as constant.

Many papers [12], [13] present optimized algorithms but with only basic nodes (sorting nodes in Fig. 6) neglecting the pipeline stages needed for synchronizations in a high-speed design. Such structures would correspond to a pure combinatorial logic, extremely slow in a practical implementation. Taking into account pipelined synchronization stages, the superiority in resource occupancy over the classical solution decreases.

A. *Classic Systolic Array*

Classic systolic array for sorting of 9 pixels, as shown in Fig. 6, requires a total of 30 full basic nodes (for median 6 of them can be neglected). As shown in Fig. 7 Smith's algorithm [12], improved in [13], can still be optimized. Taking into account basic nodes, the improvement factor is 30/19; however, with synchronization pipeline stages, it decreases to 65/50. This shows that the classic systolic array can still be useful for sorting of a long vector. The 9-point median filter has been implemented according to Fig. 7, and for longer filters according to Fig. 6.

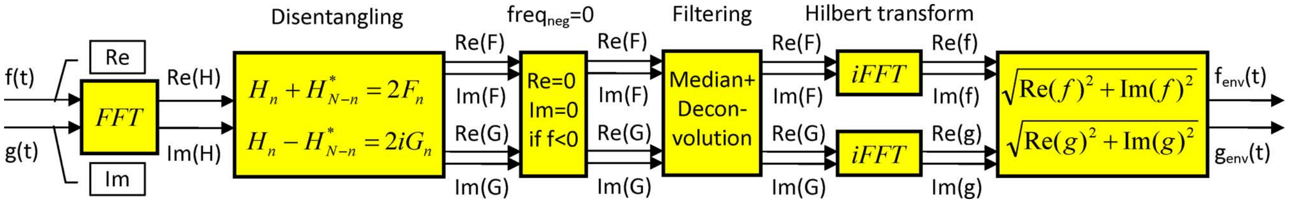

Fig. 8. Schematic view of the resources-optimized implementation of the used two antenna chains with opposite polarization, each consisting of FFT, median filter, deconvolution, Hilbert transform Im(f) and Im(g) and FIR filters.

TABLE II
UTILIZATION OF RESOURCES FOR VARIOUS MEDIAN FILTERS

width	Algorithm	LE	Speed MHz	Latency clk. cycl.
9	Classical systolic	822	250	10
9	[12] optimized	689	250	10
9	SRC_9	555	250	10
15	Classical systolic	3160	231	16
15	SRC_15	957	250	13
25	Classical systolic	6819	250	26
25	SRC_25	1627	231	18
31	SRC_31	1976	216	21

B. Serial Rank Computer

The serial rank computer (SRC) [14] operates on a FIFO with two attributes: the value of each entry and its rank. With each clock cycle, a new entry is added to the FIFO and the oldest one is removed. Both the new and the oldest entries are compared with each FIFO value. If the old and new entries are both higher or both lower than a FIFO value, the associated rank is kept. If the new entry is higher and the old one is lower, the associated FIFO rank is decreased by one. Otherwise, the rank is increased by one. The rank of the new entry is calculated by a sum over all the results of the comparisons. The median value is selected by a multiplexer choosing the value with a rank of half the median filter width.

The advantage of this implementation is that the resource occupancy only grows linearly with the median width. As a disadvantage, the speed is limited. In particular for larger median widths, the output multiplexer and summing up the new entry's rank are speed-critical. By pipelining these two operations over 2 clock cycles, the speed according to Table II can be improved to fulfill the requirements.

Because its better scalability and lower resource occupancy, the SRC was chosen for the final FPGA implementation. Its maximum clock frequency is still high enough for our purposes.

VI. COMPUTING TRICKS

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 (3).

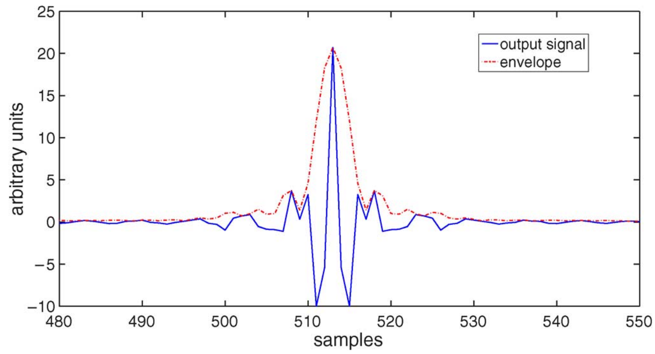

Fig. 9. Envelope of deconvoluted pulse of Fig. 5 created by using the Hilbert transformation.

$$H_n = \sum_j e^{2\pi ijn/N}(f_j + ig_j) \qquad (3)$$

The H_n can then easily be disentangled into the Fourier components, F_n and G_n, by the following (4)

$$H_n + H^*_{N-n} = 2F_n, \quad H_n - H^*_{N-n} = 2iG_n, \qquad (4)$$

(N-n) indices in (4) in a real time system correspond to a time reversed order. Synchronization of the H_n and H^*_{N-n} is done 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 iFFT, the envelope $x_{en}(t)$ (Fig. 9) of the output signal $x(t)$ has to be created to allow the following trigger algorithms to discriminate specific pulse shapes.

This can be done using the Hilbert transform $\tilde{x}(t) = x(t) * (1/\pi t)$ of $x(t)$, where $*$ denotes the convolution operator. Then the envelope is:

$$x_{en}(t) = |x_a(t)| = \sqrt{\tilde{x}(t)^2 + x(t)^2}, \qquad (5)$$

To get $x_a(t)$, respectively $\tilde{x}(t)$ and $x(t)$ at once, (6) and (7) can be used:

$$x_a(t) = F^{-1}\left(X_a(f)\right) = x(t) + i \cdot \tilde{x}(t) \qquad (6)$$

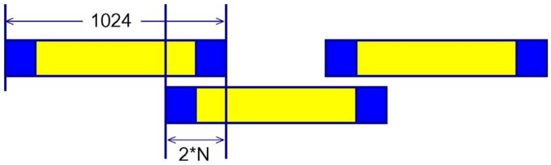

Fig. 10. A schematic of overlapping routine. Light gray rectangles correspond to the range of data, kept as RFI-filtered signal. Dark gray rectangles correspond to neglected data.

with

$$X_a(f) = \begin{cases} 2X(f), & f > 0 \\ X(f), & f = 0 \\ 0, & f < 0 \end{cases} \tag{7}$$

After iFFT of the processed signal, the envelope can be calculated according to (5) as the absolute value of the complex output of the iFFT. For the implementation into the FPGA drafted in Fig. 8, this means that after FFT, only the positive frequency components have to be taken into account when applying the deconvolution and median filter. As long as negative frequency components are streamed out of the FFT engine, these functions can fall back into a power saving sleep mode.

VII. 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, 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.

Both problems can only be solved, without introducing dead time between the blocks, by using an overlapping routine as shown in Fig. 10. Therefore the filter engine must run in another clock domain with higher frequency. Preliminary estimation shows that for an overlapping of $\mathrm{N} = 32$ errors due to an aliasing contribution is acceptable, however for a better safety margin $\mathrm{N} = 64$ is preferred. $\mathrm{N} = 128$, allowing a total removal of aliasing effect, however requires too high overclocking according to Table III. An odd value like $\mathrm{N} = 73$ 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.

TABLE III
Estimation of PLL Overclock Frequency Needed for the Overlap of Consecutive Blocks of 180 MHz Sampling

N	Scaling factor	PLL ratio	PLL frequency
128	1/(1-2*128/1024) = 1.333	4/3	240 MHz
73	1/(1-2*73/1024) = 1.166	7/6	210 MHz
64	1/(1-2*64/1024) = 1.143	8/7	205.7 MHz
32	1/(1-2*32/1024) = 1.067	16/15	192 MHz

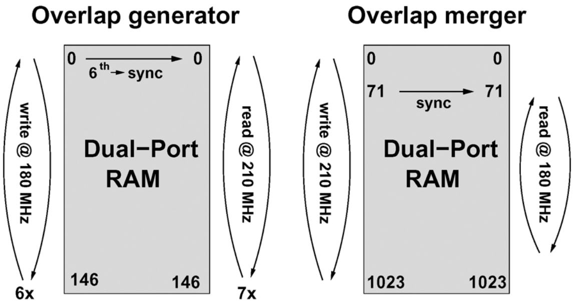

Fig. 11. *Left*: The overlap generator writes 882 values to a dual-port RAM at 180 MHz, at the same time it reads 1024 values at 210 MHz, then stops for 5 cycles. *Right*: The overlap merger writes 1024-point blocks from the FFT to RAM at 210 MHz, only 882 values are read out at 180 MHz. To ensure gapless operation, correct synchronisation is crucial.

The module to create the overlap is drafted in Fig. 11: Incoming data is continuously written into a dual-port RAM like into a ring-buffer. Each sixth time at write address 0, the read address is also synchronized to 0. As the read address pointer is running with a higher frequency, it is always ahead of the writing pointer, delivering the old data. After 7 passes, when the read pointer almost catches up to the write pointer it is reset to zero again by the synching process. This way $6 * 147 = 882$ values are written, while $7 * 147 = 1029$ are read. As the FFT can only process 1024 samples, the FFT is stopped for 5 clock cycles after each block. This effectively reduces the overlap to $\mathrm{N} = 71$, not introducing dead time as the lost 5 samples are repeated in the next block anyhow.

To merge the overlapping blocks back to a continuous data stream, the overlap merger as described in Fig. 11 writes the 1024-point block from the iFFT into a dual-port RAM. As soon as the 72th entry is written, the reading pointer is synchronized to address 71. As the read pointer runs slower than the write pointer, it will always run behind. After reading 882 values the read address is reset to 71 by the synching process. The first 71 and the last 71 values which are affected by the window function are never read. As the used window function is 1 in the middle, no further amplitude correction is necessary.

VIII. Trigger

The envelope generated according to Fig. 8 contains high-frequency harmonics, due to the calculation of the absolute value. To remove these contibutions we use a low-pass FIR filter with Blackman window and a cut-off frequency 50 MHz before triggering.

For pulse detection, we use a dynamic threshold, which is set to a fixed factor of typically 6 above the background RMS. Thereby, we can handle varying environmental conditions like day-night effects with a reasonable trigger rate. The shape of detected pulses is analysed, and cuts on pulse parameters, such as the pulse width and rise/fall time, are used to reject noise transients. Finally, a coincidence condition with several detector neighboring stations initiates the readout and storage of the original signal trace.

TABLE IV
THE REGISTERED PERFORMANCE AND POWER CONSUMPTION FOR CYCLONE III EP3C80F780C6

Mode	Config	LE %	Mem %	DSP %	Speed MHz	Power mW
Classic	iFFT_12	40	12	43	201	2 661
Classic	iFFT_16	47	13	49	196	3 026
Classic	Hilb_12	60	19	64	204	3 915
Optimized	Hilb_14	30	17	32	221	2 051

Columns 3–5 show the percental FPGA usage in respect of Logic Elements, Memory, and Multipliers. The 6th column—maximum clock rate, 7th—power consumption. The rows list the classical implementation of a FFT-iFFT chain (12-bit and 16-bit), a FFT-Hilbert transform chain, and our optimized 14-bit FFT-Hilbert transform implementation.

IX. RESOURCES OCCUPANCY AND POWER CONSUMPTION

Different configurations of FFT-engines were tested. Although favoring a variable streaming architecture for its higher numerical precision, its speed according to Table I is not sufficient when using overlapping with higher clock frequency of 210 MHz for leakage removal. The streaming architecture offers higher operating speed, the precision can be improved by using higher bit-width according to Fig. 2. A width of 14 bits turned out to be a good compromise allowing sufficient processing speed, and a good precision.

For a precise envelope generation based on the Hilbert transform we normally need 2 iFFT routines per channel. This would increase the FPGA usage to a level of 60% for two channels (see Table IV), which leaves insufficient space for the subsequent trigger algorithm, I/O-routines and a controlling processor. By optimizing the algorithm, we can half the number of used FFT-engines, and reduce the FPGA usage to a tolerable value of 30%. This also significantly reduces the power consumption as shown in Table IV, which is an important improvement for a system driven by solar energy.

X. CONCLUSION

Due to the low signal to noise ratio in the noise-polluted radio band, self triggered radio detection of cosmic rays is a challenge requiring elaborate signal conditioning.

For this purpose we successfully developed an RFI filter, removing mono-frequent carriers in Fourier-space without affecting broadband signal pulses. In addition the signal pulse amplitude is improved by deconvoluting the analoge filter dispersion. After generating the envelope using a Hilbert transform, the enhanced signal is fed into the trigger algorithm.

In order to avoid aliasing and leakage effect when transforming into Fourier-space and back, we use a window function with overlapping data blocks, which requires an overclocking of the RFI filter module. To fulfill the raised speed requirements, we chose the faster streaming FFT routine over the more accurate, but slower variable streaming architecture.

By exploiting symmtries of the FFT for real input signals, we could half the number of used FFT engines from a total of 6 to 3. Thereby the ressource occupancy and power consumption could be significantly reduced, which is an important issue for a solar powered system.

The real-time signal processing of two channels requires high calculation power, which is provided cost-effectively by the Altera Cyclone III FPGA EP3C80F780C6. Although the FPGA with the required fast speed grade is only available in a commercial temperature range, the system can be operated at frequencies up to 210 MHz in an ambient temperature range from −25 to +60°C, as tested in a conditioning cabinet.

ACKNOWLEDGMENT

The authors would like to thank the LOPES Collaboration for providing the environment to perform the presented studies.

REFERENCES

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

[2] H. Falcke and 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, Jul. 2003.

[3] M. Ludwig and T. Huege, "REAS3: Monte Carlo simulations of radio emission from cosmic ray air showers using an 'end-point' formalism," *Astropart. Phys.*, vol. 34, pp. 438–446, Mar. 2011.

[4] K. Werner and O. Scholten, "Macroscopic treatment of radio emission from cosmic ray air showers based on shower simulations," *Astropart. Phys.*, vol. 29, pp. 393–411, July 2008.

[5] H. Falcke *et al.*, "Detection and imaging of atmospheric radio flashes from cosmic ray air showers," *Nature*, vol. 435, p. 435, 2005, [LOPES Collaboration].

[6] O. Ravel *et al.*, "Radio detection of cosmic ray air showers by the CODALEMA experiment," *Nucl. Instrum. Methods Phys. Res. A*, vol. 518, pp. 213–215, Feb. 2004.

[7] A. M. van den Berg, "Radio detection of cosmic rays at the southern auger observatory," in *Proc. 31st Int. Cosmic Ray Conf.*, Łódź, Poland, Aug. 2009 [Online]. Available: http://arxiv.org/abs/0908.4422v1

[8] J. Abraham *et al.*, "Properties and performance of the prototype instrument for the pierre auger observatory," *Nucl. Instrum. Methods Phys. Res. A*, vol. 523, pp. 50–95, May 2004.

[9] G. A. Dulk, W. C. Erickson, R. Manning, and J.-L. Bougeret, "Calibration of low-frequency radio telescopes using the galactic background radiation," *Astron. Astrophys.*, vol. 365, pp. 294–300, Jan. 2001.

[10] F. Schroeder *et al.*, "Time calibration of the radio air shower array LOPES," in *Proc. 31st Int. Cosmic Ray Conf.*, Łódź, Poland, Aug. 2009 [Online]. Available: http://arxiv.org/abs/0907.3098v1

[11] S. Fliescher, "Radio detectorarray simulation, a full simulation chain for an array of antenna detectors," *Nucl. Instrum. Methods Phys. Res. A*, vol. 604, pp. 225–229, Mar. 2009.

[12] J. L. Smith, "Implementing median filters in XC4000E FPGAs," *XCell*, vol. 23, no. 4, p. 16, 1996.

[13] M. A. Vega-Rodríguez, J. M. Sánchez-Pérez, and J. A. Gómez-Pulido, "An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection systems," in *Proc. 10th Mediterranean Conf. Control and Automation*, Lisbon, Portugal, 2002.

[14] B. Hirschl and L. P. Yaroslavsky, "FPGA implementation of sorters for non-linear filters," in *Proc. 12th. Eur. Signal Processing Conf.*, Vienna, Austria, 2004 [Online]. Available: http://www.eurasip.org/Proceedings/Eusipco/Eusipco2004/defevent/papers/cr1074.pdf

[15] T. Huege *et al.*, "Radio detection of cosmic rays in the pierre auger observatory," *Nucl. Instrum. Methods Phys. Res. A*, vol. 617, pp. 484–487, May 2010.