**FPGA-Based SOC Architecture for Fog and Edge Computing Applications**

Ilya Tarasov1[0000-0001-6456-4794] and Dmitry Potekhin1

1MIREA – Russian Technological University, Vernadsky Avenue 78, 119454 Moscow, Russia

[tarasov\_i@mirea.ru](mailto:tarasov_i@mirea.ru)

**Abstract**. The article examines an example of the architecture of a digital device of the ‘system on a chip’ class, which implements data collection from sensors of physical quantities and their preliminary processing using integral transformations. This approach is in line with the current trends in fog and edge computing, which provide for distributed data collection, preprocessing and transmission of only conversion results over wireless networks, which can significantly reduce the required traffic. The architecture is designed to accommodate conflicting requirements for high computing performance, the presence of a large number of external interfaces and lower power consumption, therefore, its implementation requires specialization of components in relation to the problem being solved. The architecture variant was implemented on the basis of FPGA with FPGA architecture. Examples of IoT devices include a high-resolution wearable cardiograph with multi-channel myograph capability with WiFi and BlueTooth interfaces, and an acoustic emission meter in power equipment. Devices are based on FPGA Xilinx Spartan-7 with external controllers for wireless interfaces.

**Keywords**: Internet of Things, Fog Computing, SoC, FPGA.

1. Introduction

Fog computing and edge computing systems are becoming more widespread as new demands in the field of measurement automation emerge, including both industrial automation and wearable Internet of Things [1-5]. New areas of application of computing and communication devices form new requirements for their characteristics. In particular, the performance and power requirements required for wearable devices may not always be met by widely used general purpose hardware platforms. If we consider the specifics of fog and edge computing, we can draw attention to the fact that we are talking, on the one hand, about high-performance devices that process signals at the point of their receipt, and on the other hand, their mobile nature determines the requirements for low power consumption. Therefore, at the present time, the urgency of creating a specialized element base that has sufficient performance to perform pre-processing of signals, and at the same time has a reduced power consumption, is increasing. The combination of these properties is achievable when developing specialized solutions.

1. Problem Description

When designing data collection systems, an important problem is the overload of communication channels when trying to collect data directly from sensitive elements. The capabilities of modern ADCs allow generating data streams up to several billions of samples per second (Gsamples/sec), which obviously exceeds the capabilities of modern wireless communication channels. In addition, the features of distributed data collection systems imply the presence of many data sources, which significantly exacerbates the problem, since even with a small data flow from one source, the need to connect multiple sources proportionally increases the amount of information transmitted.

Using a layered architecture that includes data hub middle tiers can alleviate the problem, but this approach should avoid the emergence of thick tree networking architectures. This architecture implies an increase in network traffic when moving from the sensor-hub layer to the next layer, for example, the hub-to-PC layer. The concept of fog and edge computing is a timely solution to this problem, since it allows you to organize preliminary data processing at the point of receipt, which in some cases significantly reduces the traffic required for transmission to a higher level. At the same time, the practical implementation of this approach requires the fulfillment of a number of conditions:

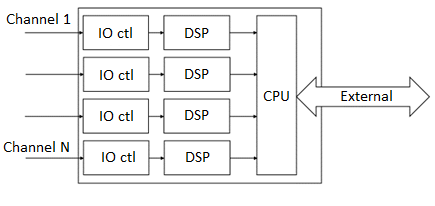
– the problem being solved must objectively allow a decrease in traffic transmitted for analysis; this can be the case when a decision on the measurement results is made on the basis of spectral, statistical or other similar characteristics of the measured signal, implying the use of transformations that reduce the amount of data (for example, transformations based on the calculation of integrals or convolutions have such a property);

– the hardware edge computing platform provides the required level of performance to implement the algorithm at the level of the primary network of sensors or a local hub that receives data from a limited number of sensors.

The use of primary concentrators in distributed measuring systems that collect data from a limited number of sensors allows unloading the higher levels of the hierarchy of such a system. You can pay attention to the fact that in this case the characteristics of the primary concentrator must be studied in detail. On the one hand, its performance and the throughput of external interfaces should be sufficient, but on the other hand, redundancy of characteristics is likely to cause an increase in weight and dimensions, energy consumption and cost, and also complicate operation. Therefore, this article discusses the use of FPGAs for building specialized devices designed to work as data concentrators in distributed measuring systems, which also perform preliminary processing in order to reduce traffic to higher system elements. This FPGA application follows the concept of fog/edge computing.

A known disadvantage of FPGAs with FPGA architecture is the worst performance of clock frequency and specific power consumption compared to CPU / GPU due to the presence of configurable connections and the implementation of logical expressions based on programmable truth tables based on static memory (LUT). At the same time, modern FPGAs have a large number of hardware-implemented components on the chip, the characteristics of which are not degraded relative to solutions comparable in the technological process, since these components are not configurable and do not contain redundant switching components. Traditionally, such devices are static dual-port memory blocks (BRAMs), DSP48 multiply-accumulate components, and, for some FPGA families, high-speed serial transceivers (MGTs). Hardware-based components deliver high absolute performance for FPGAs.

Thus, it is required to consider the architecture of a system that is a concentrator of data from a set of external sources. Such a system, implemented in an FPGA, is designed to connect external data sources, perform preliminary signal processing in order to reduce the amount of transmitted data and connect external interfaces for integrating a data concentrator into a higher-level system. At the same time, it is necessary to pay attention to the efficient use of FPGA hardware components and to compensate for the negative effects from configurable logic cells that have obviously worse technical characteristics. A variant of the considered architecture is given on the Figure 1.



**Fig. 1.** Hub architecture of the data acquisition subsystem for implementation on the basis of FPGAs.

The ability to configure FPGAs is of interest mainly in the system development process. This is due to the fact that it is advisable to debug and refine digital signal processing algorithms on a workable prototype of the device, while the computing performance, the structure of data processing channels and the interaction of hardware accelerators with the processor can change significantly when obtaining experimental results. Therefore, when developing new systems with requirements that cannot be specified in advance, it is useful to use a reconfigurable hardware platform.

FPGA configuration is complemented by the ability to program an embedded processor, the so-called. software processor. While some FPGA families use hardware cores, such as the Cortex-A core in the Xilinx Zynq-7000 and Xilinx Zynq MPSOC families, some of the FPGA's programmable logic cells can be configured to replicate the operation of the processor device.

The characteristics of the soft processor used to control the fog computing device are of significant interest for analysis, mainly at the design stage. Despite the general impression of sufficient computing performance for embedded electronics, power consumption and the ability to integrate components on a chip are significant for the architectures considered. Therefore, the development of specialized digital devices of the ‘system on a chip’ class is of practical interest from the point of view of eliminating functional redundancy.

Despite the fact that modern soft processors often support the use of cache memory created on the basis of on-chip static FPGA memory, this approach has a number of design and system drawbacks. First of all, the use of external heap memory complicates PCB routing, increases overall power consumption, and generally complicates device design. From the point of view of the system architecture, the appearance of interacting memory components does not allow predicting the exact execution time of critical segments of the code, since a cache miss can occur at an arbitrary point in time. There is a tendency to use Tightly Coupled Memory in real time systems.

For FPGA-based systems with full control of the design at the development stage [1], you can either explicitly indicate the need to use static memory as the main one, or apply an appropriate architectural approach when designing a specialized control processor based on the use of static memory as the only one available to the processor. In combination with a focus on architectures with a high code density (for example, stack architectures with a zero-operand instruction system), this in many cases makes it possible to control hardware accelerators without deploying a standard processor subsystem on a chip, which in this subclass of devices will be functionally redundant.

The control processor of a heterogeneous computing system can have a moderate performance sufficient for solving control problems. This is due to the fact that the control processor is not a critical unit in such a system that determines the performance in the main problem being solved. The hardware costs of implementing the control processor are overhead costs, since the FPGA resources spent on its implementation are diverted from solving basic computational problems. Therefore, the size of the control processor should be reduced while maintaining an acceptable level of functionality and flexibility in implementing the underlying algorithms. The presence of conflicting requirements makes it questionable to use the term ‘minimization of hardware costs’, since formally a processor architecture can be obtained that meets the mathematical criterion of minimality, but inconvenient for practical work with applied software.

For the control processor, tight integration with hardware accelerators is desirable, including through the creation of specialized system buses. Versatility and scalability are of lower priority because the integration of the processor and accelerators occurs during the system architecture design process. In this case, simplification of communication protocols will have a more noticeable positive effect on the characteristics of the project compared to the potential possibility of changing the system architecture.

1. FPGA Application in Fog and Edge Computing

Modern FPGAs with FPGA architecture provide essential capabilities for the problem being solved. Important are the number of external pins, which determines the throughput of external interfaces and the connectivity of sensors, as well as the total performance of the digital signal processing subsystem based on DSP sections. The comparative characteristics of the Xilinx FPGA families intended for the development of entry-level systems are given below in Table [6]. It can be noted that a number of FPGA-based products provide for the use of external controllers for wireless interfaces, for example, for the Minized board [7].

**Table.** Comparative characteristics of the Xilinx FPGA families intended for the development of entry-level systems.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Spartan-7 | Artix-7 | Zynq-7000 |
| Logic cells, k. | 6 – 100 | 12 – 215 | 23 – 85 |
| Block RAM, Mbit | 0.18 – 4.3 | 0.72 – 13.1 | 1.8 – 4.9 |
| DSP slices | 10 – 160 | 40 – 740 | 60 – 220 |
| MGTs | – | 2 – 16 | 0 – 16 |
| Max data rate of MGT, Gbit/s | – | 6.6 | 6.6 |
| Programmable inputs/outputs, max. | 400 | 500 | 328 |

An important factor affecting the characteristics of such a subsystem is the ability to reduce the amount of data transmitted to a device at a higher level of the hierarchy. Algorithms with this property include integral transforms, for example, the Fourier transform and the wavelet transform. The basis of the wavelet transform is the calculation of the convolution integral of the form:

.

The form of this formula makes it possible to assert that when the integration is replaced by the summation of discrete readings, the amount of information received decreases in proportion to the increase in the number of readings. Therefore, data processing algorithms based on the calculation of the wavelet transform can transmit the values of the convolution integral as the processing results, rather than the samples themselves.

Integral transforms are convenient for FPGA-based implementations, which, as shown in the above Table, even for entry-level families provide the developer with 10 to 740 ‘multiply and accumulate’ hardware modules (DSP slices). With a technically achievable clock frequency of 300-400 MHz (which corresponds to the features of tracing projects with real, rather than peak operating frequencies), the total performance of the digital signal processing subsystem can range from 3 to 300 GMAC/s (billions of operations ‘multiply and accumulation’ per second).

The calculation of the wavelet spectrum (or ‘wavelet density’) is performed in the same way as the Fourier spectral density. The difference is that instead of a harmonic series, the wavelet function acts as an analyzing function. In general, only two requirements are imposed on the wavelet function:

1. Localization in the time domain (i.e., the function should decay with distance from the center along the t axis).

2. Absence of a constant component (the integral of the function must be equal to 0).

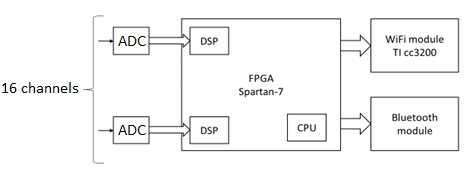
The use of wavelet analysis opens up wide prospects, since the synthesis of modulating windows for wavelet functions allows, in a number of cases, to obtain sufficiently high-quality results. By combining the time intervals and the attenuation coefficient of the Gaussian window, it is possible to obtain a set of frequency response of the wavelet function, which differ in the width of the spectrum and the amount of suppression of the Gibbs effect. At the same time, the absence of an analogue of the fast transform for the Morlet wavelet function makes it necessary to calculate the wavelet density by the “direct method” - by repeating the operation “multiply with accumulation” for each of the frequencies of interest. To do this, you can effectively use hundreds and thousands of DSP48 slices, which are available in modern FPGAs.

1. Examples of Systems

Based on the described approach, a number of projects were carried out, implying the connection to the FPGA of a set of sensors that generate a continuous stream of data for analysis. An FPGA-based concentrator was used to perform integral transformations (Fourier analysis, wavelet analysis) and then prepare data for transmission over a wireless interface.

The project of a wearable cardiograph / myograph is a device of the "Internet of Things" class, implemented on the basis of a multichannel ADC, FPGA and WiFi / BlueTooth wireless modules. Medical monitoring devices are currently of interest to a number of researchers [8-10] and are considered as a type of wearable Internet of Things devices. The block diagram of the device is shown in Figure 2.

Digital signal processing consists in performing band-pass filtering using wavelet filters in the myogram recording mode (measuring muscle activity). The transmission of a raw signal in this mode is of no practical interest, since information on muscle activity is contained in the amplitude of the frequency bands selected for analysis. For signal processing, we used bandpass wavelet filters described by the authors in [11].



**Fig. 2.** Block diagram of a cardiograph/myograph with WiFi and BlueTooth wireless interfaces.

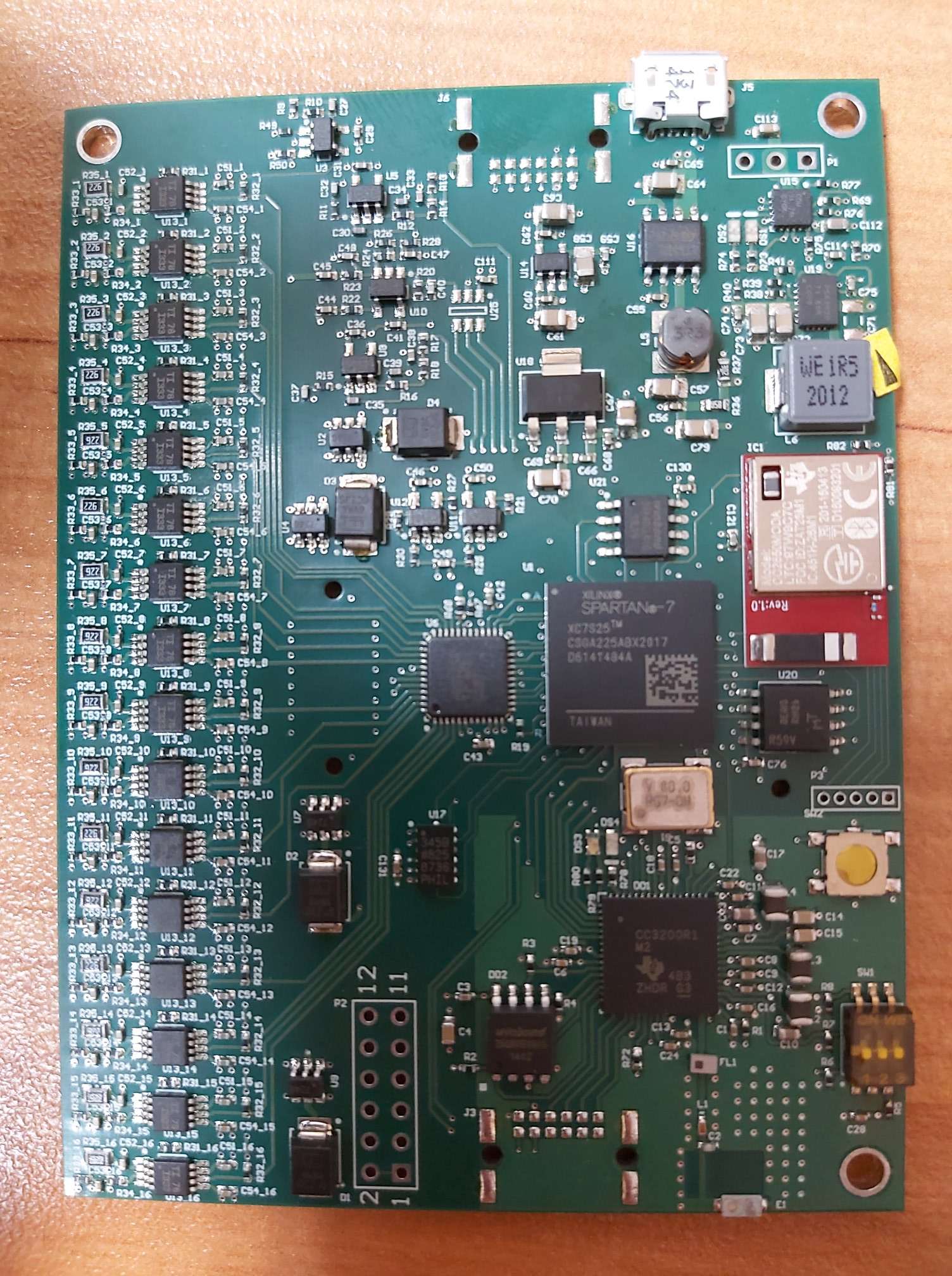
In the mode of taking the cardiogram, digital signal processing can be simplified and reduced to filtering high-frequency interference. It can be noted that the so-called. a high-resolution ECG assumes analog-to-digital conversion with a frequency of 2 kHz, so the total load on the wireless interfaces turns out to be small even when transmitting the original data received from the ADC. However, in this case, the FPGA solves another important problem - ensuring the continuity of the measurement process and buffering data. This property is important when using wireless communication channels, in which transmission is carried out in burst mode, and in addition, local fluctuations in the transmission rate are possible due to changes in the reception conditions.

In addition, the independent input of 16 channels of analog-to-digital conversion without using time division multiplexing is a difficult task for a processor with single-threaded instruction execution. The implementation on the basis of FPGA cells and Xilinx XtremeDSP digital signal processing units of a signal preprocessing subsystem with a wavelet transform in real time provided a significant unloading of the control processor, reducing its role in data processing to reading FIFOs with ready conversion results.

The use of an additional digital node that buffers data allows for the collection, processing and transmission without loss of signal fragments in those moments when the connection is unstable or switching to another frequency channel, which may occur for wireless interfaces.

The prototype of the device is shown in Figure 3. The device uses a Xilinx Spartan-7 FPGA with a logical volume of 25 thousand cells, a single-chip WiFi TI CC3200 controller, and a BlueTooth module. Power is supplied from an external 5 V battery, which allows you to use a wide range of sources designed to power mobile devices. The meter's dimensions allow you to place it in your pocket or on your belt, including for online recording of muscle activity during sports exercises.

The presence of a processor subsystem on the board allows recording signals into flash memory for their subsequent transmission to a higher-level information system for analysis. This makes the device non-critical to the availability of access to a wireless connection, allowing the collection and preliminary analysis of data autonomously. This property is useful for monitoring muscle activity when exercising outdoors, where wireless communication with the nearest WiFi site is likely to be unavailable, and constant use of mobile communication may be undesirable.



**Fig. 3.** The cardiograph/myograph with wireless WiFi and BlueTooth interfaces.

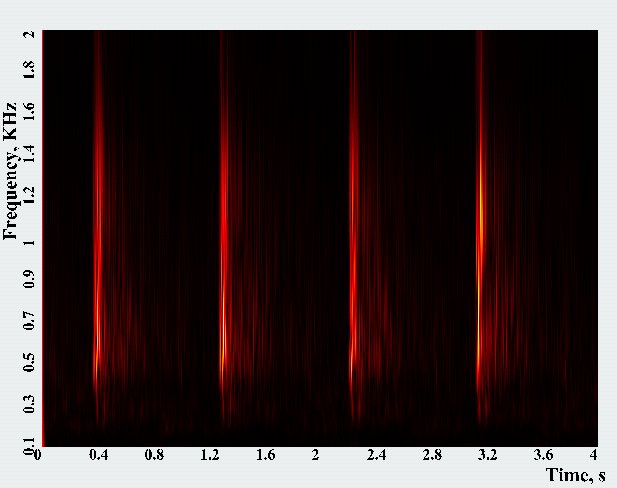
Another example of FPGA-based edge computing is the PD meter. A partial discharge is an electrical discharge in insulation, the duration of which is between several tens of nanoseconds. Partial discharge short-term shunts the insulation of high-voltage equipment, which leads to a short-term change in the current in the circuit and is accompanied by acoustic noise. These phenomena can be registered by various methods. Two approaches can be distinguished:

- direct observation of current surges;

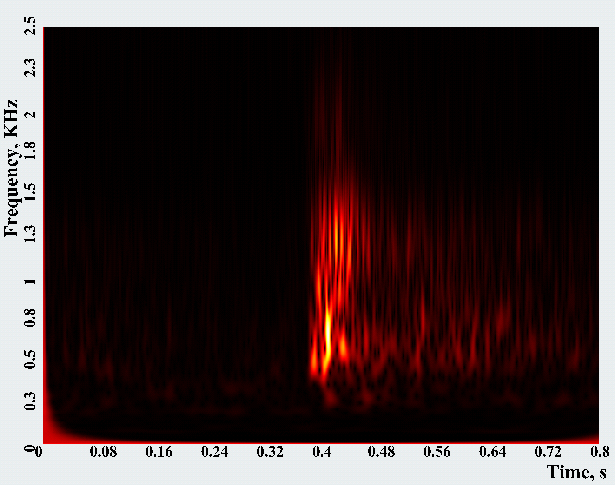
- measurements of acoustic noise, for example, using a piezoelectric sensor.

Partial discharges appear in the weak point of high-voltage equipment and lead to the gradual development of a defect and destruction of insulation.

Figure 4 shows the spectrum of noise in working high-voltage equipment. Figure 5 shows the noise spectrum in the same equipment in a pre-emergency state. This information can be used to create methods for automated diagnostics, but this requires additional research to obtain an experimental collected base of acoustic noise.

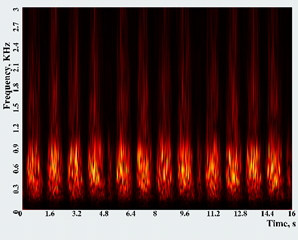


a)

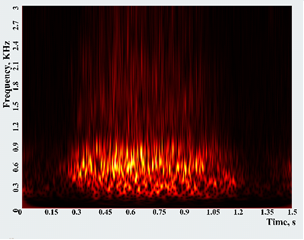


b)

**Fig. 4.** Noise spectrum in high-voltage equipment (a) and enlarged fragment (b).



a)



b)

**Fig. 5.** Noise spectrum in a pre-emergency state in high-voltage equipment (a) and an enlarged fragment (b).

The above examples give an idea of the possibilities of foggy and edge calculations in the case of using a specialized element base that performs preliminary signal analysis and transfer only the conversion results. In the examples shown, the stated architectural approach was used to perform multichannel wavelet analysis of signals.

1. Conclusions

The use of FPGA-based signal preprocessing made it possible to use the developed system on a chip to significantly reduce the amount of data required for transmission over wireless communication networks. This approach allows the development of mobile devices for measuring and processing signals in real time with the formation of distributed monitoring networks that can be included in larger information systems - for example, industrial automation systems, medical or sports monitoring systems. The achieved positive effect from the use of FPGAs is a decrease in the amount of data transmitted from the hub, as well as the possibility of clarifying the structure of the device in the process of its design and updating the hardware component during operation.

References

1. Morabito, R., Cozzolino, V., Ding, A.Y., Beijar, N. and Ott, J.: Consolidate IoT edge computing with lightweight virtualization. IEEE Netw. **32**(1), 102–111 (2018).

2. Hamm, A.,Willner, A. and Schieferdecker, I.: Edge Computing: A Comprehensive Survey of Current Initiatives and a Roadmap for a Sustainable Edge Computing Development. In: Proceeding of WI2020, pp. 694–709. GITO Verlag. Potsdam (Mar. 2020). DOI: 10 . 30844 / wi 2020 g1 - hamm. arXiv: 1912.08530.

3. Pfandzelter, T., Hasenburg, J. and Bermbach, D.: From Zero to Fog: Efficient Engineering of Fog-Based IoT Applications. Mobile Cloud Computing Research Group Technische Universit at Berlin & Einstein Center Digital Future (August 19, 2020). {tp, jh, db}@mcc.tu-berlin.de

4. Pfandzelter, T. and Bermbach, D.: IoT data processing in the fog: Functions, streams, or batch processing? In: Proc. of DaMove, pp. 201–206 (Jun. 2019).

5. Darwish, T.S.J. and Bakar, K.A.: Fog based intelligent transportation big data analytics in the Internet of vehicles environment: Motivations, architecture, challenges, and critical issues. IEEE Access **6**, 15679–15701 (Mar. 2018).

6. <https://www.xilinx.com/products/silicon-devices/cost-optimized-portfolio.html>

7. <http://zedboard.org/product/minized>

8. Chetelat, O., Ferrario, D., Proenc¸a, M., Porchet, J.-A., Falhi, A., Grossenbacher, O., Delgado-Gonzalo, R., Della Ricca, N.and Sartori, C.: Clinical validation of LTMS-S: A wearable system for vital signs monitoring. In: EMBC’2015, pp. 3125–3128 (2015).

9. Segarra, C., Delgado-Gonzalo, R., Lemay, M., Aublin, P.-L., Pietzuch, P. and Schiavoni, V.: Using trusted execution environments for secure stream processing of medical data. In: Lect. Notes Comput. Sc. **11534**, pp. 91–107 (2019).

10. IHE PCD Technical Committee: Medical equipment management (MEM): Medical device cyber security.White paper, IHE International, Inc., (Oct. 2015).

11. Tarasov, I.E., Potekhin, D.S.: REAL-TIME KERNEL FUNCTION SYNTHESIS FOR SOFTWAREDEFINED RADIO AND PHASE-FREQUENCY MEASURING DIGITAL SYSTEMS. Russian Technological Journal **6**(6), 41-54 (2018) (in Russian). https://doi.org/10.32362/2500-316X-2018-6-6-41-54