

DOI: <https://doi.org/10.15276/aait.06.2023.7>
UDC 004.383

Investigation of reconfigurable hardware platforms for 5G protocol stack functions acceleration

Viktor A. Melnyk^{1), 2)}

ORCID: <https://orcid.org/0000-0002-5046-8002>; viktor.melnyk@kul.pl. Scopus Author ID: 57200786767

Vladyslav V. Hamolia²⁾

ORCID: <https://orcid.org/0000-0003-2444-0775>, vladyslav.v.hamolia@lpnu.ua. Scopus Author ID: 57222387801

¹⁾ The John Paul II Catholic University of Lublin. 14, Al. Racławickie. 20-950 Lublin, Poland

²⁾ Lviv Polytechnic National University, 12, Bandery Str. Lviv, 79013, Ukraine

ABSTRACT

Open RAN and 5G are two key technologies designed to qualitatively improve network infrastructure and provide greater flexibility and efficiency to mobile operators and users. 5G creates new capabilities for high-speed Internet, Internet of Things, telemedicine and many other applications, while Open RAN enables open and standardized network architectures, which reduces cost and risk for operators and promotes innovations. Given the growing number of users and data volumes, the purely software implementation of certain functions of the 5G protocol, and especially computationally complex ones, requires significant computer resources and energy. These, for example, are low-density parity-check (LDPC) coding, FFT and iFFT algorithms on physical (PHY) layer, and NEA and NIA security algorithms on Packet Data Convergence Protocol (PDCP) layer. Therefore, one of the activity areas in the development of means for 5G systems is the hardware acceleration of such functions execution, which provides the possibility of processing large volumes of data in real time and with high efficiency. The high-performance hardware basis for implementing these functions today is field-programmable gate array (FPGA) integrated circuits. Along with this, the efficiency of the 5G protocol stack functions hardware acceleration depends significantly on the size of the data packets transmitted to the hardware accelerator. As experience shows, for certain types of architecture of computer systems with accelerators, the acceleration value can take even a negative value. This necessitates the search for alternative architectural solutions for the implementation of such systems. In this article the approaches for hardware acceleration using reconfigurable FPGA-based computing components are explored, their comparative analysis is performed, and architectural alternatives are evaluated for the implementation of a computing platform to perform the functions of the 5G protocol stack with hardware acceleration of PHY and medium access control (MAC) layers functions.

Keywords: Open radio access network; 5G; hardware acceleration; field programmable gate array

Copyright © Odessa Polytechnic National University, 2023. All rights reserved

For citation: Melnyk V. A., Hamolia V. V. "Investigation of reconfigurable hardware platforms for 5G protocol stack functions acceleration". *Applied Aspects of Information Technology*. 2023; Vol. 6 No. 1: 84–99. DOI: <https://doi.org/10.15276/aait.06.2023.7>

INTRODUCTION

Today, 5G systems are gradually but surely entering the market of telecommunications services. In contrast to telecommunication systems of previous generations, the 5G systems are "open", that is, those in which hardware and software are disaggregated, with open interfaces and the possibility of virtualization [1]. Such a technological model contributes to the rapid development of the market of telecommunication services, increased competition and innovations. Accordingly, manufacturers of the means for 5G systems, guided by open standards, can independently create hardware and software components of systems and develop their architecture.

The 5G protocol stack includes many components that provide various network functions, such as data transmission, session management, security, network resource management, and others.

As the amount of data processed in a 5G network grows exponentially, as well as the number of users and devices interacting in the network, a hardware acceleration of the 5G protocol stack functions is required [2], [3], [4]. Hardware acceleration is aimed to distribute data processing between different devices and resources in the network, reducing the load on central servers and increasing the speed of data processing. In addition, hardware acceleration allows for data security and protection against attacks, in particular preventing DDoS attacks and performing other network functions quickly, in real time.

PUBLICATIONS REVIEW AND STATEMENT OF THE PROBLEM

Scientific works devoted to hardware acceleration of 5G protocol stack functions execution are an active area of research in the field of mobile networks and network technologies. First of all, this concerns the functions of the lower layers: PHY and MAC, as well as PDCP, which have a high

© Melnyk V., Hamolia V., 2023

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0>)

computational complexity. In particular, the work [2] reviewed various hardware acceleration techniques that are used to optimize the performance of 5G network functions.

In the work [5] the implementation is proposed of 5G DU Low-PHY layer functions into an FPGA-based SmartNIC (Network Interface Card) exploiting the Open Computing Language (OpenCL) framework to facilitate the integration of accelerated 5G functions within the mobile protocol stack. In the work [6] a Low-Density Parity Check (LDPC) decoder accelerator implementation in FPGA is proposed, and the process of its integrating into the system is shown. In addition, there are many other not less important works that consider the specifics of implementing the 5G protocol stack in cloud computing environments, the deployment of graphics processing units (GPUs) as accelerators, etc.

Along with this, the existing publications do not sufficiently cover research on alternative architectural solutions aimed at reaching the maximum speedup gain on 5G protocol stack execution with use of hardware acceleration of PHY, MAC and PDCP layers functions. Not enough attention has been paid to the features of data processing in hardware accelerators. In particular, the question of the influence of the size of the data packets to be processed on the acceleration value, and how exactly this influence changes with a change in the architecture of the hardware accelerator, the way the accelerator is connected to the system and the organization of its work, remain unexplained.

THE GOAL AND TASKS OF THE RESEARCH

The goal of this article is to study architectural alternatives for the implementation of a computing platform to perform the functions of the 5G protocol stack with hardware acceleration and reach the maximum speedup gain on PHY, MAC and PDCP layers functions execution, such as LDPC coding-decoding, FFT and iFFT, NEA and NIA security algorithms.

For this, an analysis of the challenges faced by developers of hardware accelerators for PHY, MAC and PDCP layers functions in 5G will be carried out. In particular, this is the effect of the size of the data packets to be processed on the acceleration value. An overview of the approaches for acceleration with reconfigurable logic devices in the context of 5G systems will be given. The architecture of reconfigurable hardware platforms and types of equipment available on the market suitable for hardware acceleration of 5G protocol stack functions will be investigated.

1. Challenges to the 5G protocol stack functions acceleration

A number of experiments investigating the performance characteristics of individual algorithms from the 5G protocol stack on hardware platforms with reconfigurable accelerators have shown that the most traditional approach to acceleration - connecting the accelerator board to the host via a high-performance peripheral interface (e.g. PCI-Express) - is ineffective. In particular, when transferring the execution of individual functions of the PDCP layer to the accelerator, the obtained acceleration indicators turned out to be negative for the majority of experimental data sets. At the same time, the performance indicators of the IP Cores themselves implemented in the FPGA accelerator are high - about 2 orders of magnitude higher compared to those on the host CPU.

As the accelerators, the Xilinx Alveo U250 and U200 Data Center Accelerator Cards and the Intel Arria 10 FPGA boards were used in the experiments. The connection interface with the host was PCIe Gen4x8. Its theoretical performance is 8GT/s, 16,000 MB/s. Data packet sizes used in experiments were of 50B, 1KB and 8KB.

The obtained characteristics showed that the acceleration downgrade for packets of the smallest size is in a range of 3 orders of magnitude. With larger packet sizes the acceleration results are improving, and with packets larger than 1 MB, the acceleration becomes positive, and when processing data portions of 100 MB, it is more than 20 times greater. Thus, it can be concluded that the major delay is introduced during the transmission of the data packet to the accelerator and backwards, and not during its processing.

Since the 5G packet size is anyway smaller than 1 MB, which is needed for positive acceleration, there is a task of researching alternative technical solutions that would make it possible to obtain positive acceleration of data packets of a typical size of the order of 1 KB. The communication bottleneck, which occurs in the above version of connecting the accelerator, takes place for two reasons:

1. The method of organizing data exchange with the accelerator: during the experimentation a typical look-side accelerator was used, not an in-line accelerator.

2. The deployment of PCIe peripheral interface to communicate with the accelerator, which corresponds to a loosely coupled architecture.

Let's consider these reasons in more detail.

Look-aside versa inline acceleration

The principal difference between look-aside and inline acceleration approaches is that in the first, only selected functions are sent to the accelerator from the CPU and then back, while in inline acceleration parts of or the whole data flow and functions are coming through the accelerator (see Fig. 1) [7].

In the case of look-aside acceleration, the CPU is free to use its cycles to process other useful tasks, while the accelerator is working on the data to be accelerated. Once the CPU receives processed data back from the accelerator, it can switch back to the original processing context and continue the pipeline execution until the next function to be accelerated comes up. Thus, the look-aside acceleration requires intensive data transfer between CPU and accelerator and therefore their tight integration is preferred.

In the inline acceleration case, a part of or the entire layer functions pipeline can be offloaded to the accelerator, potentially allowing for a less data heavy interface between CPU and accelerator. The acceleration solution can in this case be a mix of programmable and “hard” blocks, and there is a trade-off between flexibility and efficiency.

Deployment of a loosely-coupled FPGA

The structure of computer systems with a loosely coupled reconfigurable logic is introduced in [8]. Although the accelerator boards (e.g. Xilinx Alveo U250, Alveo U200, Intel Arria 10 FPGA), on which the experiments were conducted, are modern and equipped with a high-performance PCIe Gen4x8 interface, there are still additional time overheads for

the organization of interaction (drivers, runtime systems, etc.), which have a significant impact on the acceleration value.

Therefore, the loosely coupled architecture is effective for applications with non-intensive data exchange, which can be done in parallel with processing on an accelerator.

2. An overview of approaches for acceleration in context of 5G systems

We can single out several hardware platforms to solve the task of accelerating the 5G protocol stack functions execution, which will be based on the type of architecture of the reconfigurable hardware accelerator. Along with this, we should analyze them in a context of solving the problem of the “bottleneck” (caused by both look-aside acceleration principle and loose coupling) in the interaction of the system core (host, CPU) with the accelerator. The splitting of the functions of the 5G protocol stack between the host and the accelerator should be such that the data transfer delay does not negate the acceleration value, i.e.:

$$T_{\text{CPUonly}} \gg (T_{\text{host}} + T_{\text{acc}}) + T_{\text{com}}, \quad (1)$$

where T_{CPUonly} is a time of functions execution without acceleration, T_{host} is a time of execution of part of the functions by the host interacting with the accelerator, T_{acc} is a time of functions execution by the accelerator, and T_{com} is information transfer time from the host to the accelerator and backwards when executing functions. The equation (1) must hold even when the T_{host} and T_{acc} periods do not overlap in time.

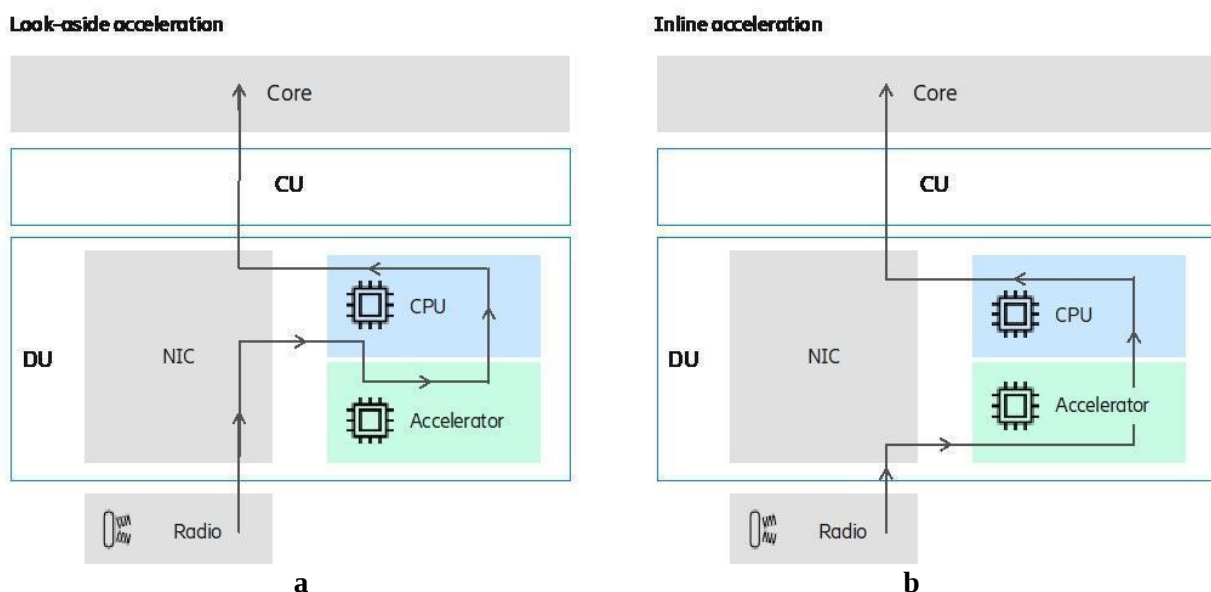


Fig. 1. Illustrating look-aside (a) versus inline (b) acceleration

Source: compiled by the [8]

According to the results of the analysis of the previously performed experiments (see section 1), it seems that such a technical solution is hardly feasible with a loosely coupled, look-aside accelerator board (connected to the host via the peripheral interface).

As alternatives, the following architecture options should be considered and explored: reconfigurable FPGA- or SoC-based SmartNIC,

- tightly-coupled FPGA-based accelerator, and
- integrated reconfigurable SoC-based platform.

In the first solution, the key to solving the “bottleneck” problem is the inline organization of data flow through the accelerator. It does not involve the transfer of data from the host CPU to the accelerator and back during the execution of individual functions on the FPGA, but the execution of a full or partial set

of functions of the lower layers of the protocol stack on the FPGA immediately after receiving data during the downlink, and further transfer of the results to the host CPU to perform the rest of the functions, without the need for further invocation of the accelerator when performing the functions on higher layers. The same principle of operation is during the uplink – the data are sent from the host CPU to the accelerator after execution of the higher layers’ functionality, while the accelerator executes the remaining lower layers tasks and transmits the data out. This can be shown by the diagrams in Fig. 2a and Fig. 2b. In this figure (a) the diagram is shown for inline acceleration with a SmartNIC that does not contain a programmable processor, and in (b) – for a SmartNIC that combines an FPGA and a programmable processor.

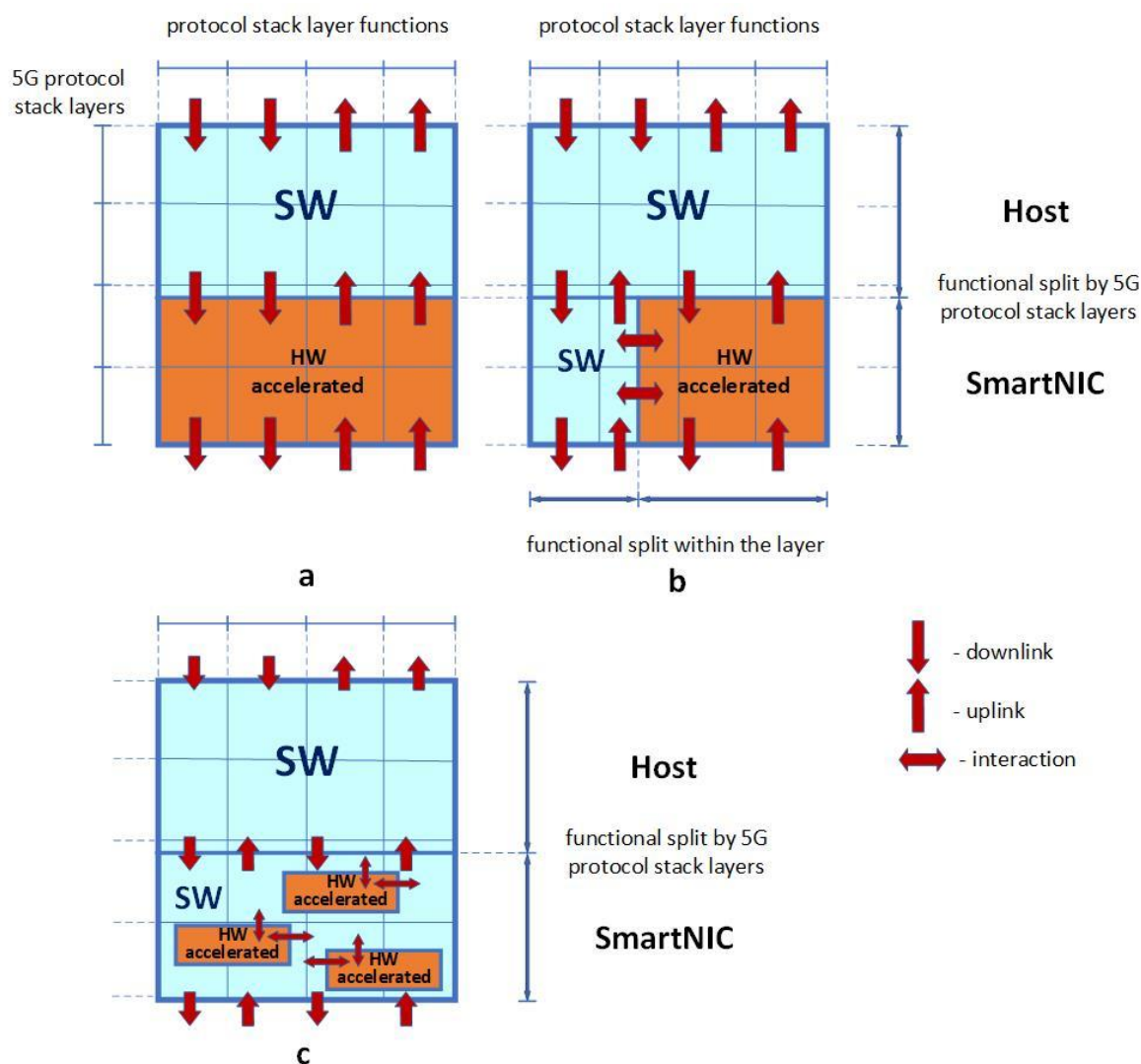


Fig. 2. Defining a set of functions for acceleration with SmartNICs:
a – inline without programmable processor; b – inline with programmable processor;
c – look-aside with programmable processor

Source: compiled by the authors

In the following two solutions, the key to solving the “bottleneck” problem is a fast and parallel information exchange, which is ensured in the first case by the “tight” interaction of the CPU with the accelerator through the appropriate interface and connection method, and in the second case by the close integration of the CPU and reconfigurable logic within the silicon chip. The splitting of functions according to this approach is illustrated in Fig. 2c and Fig. 3.

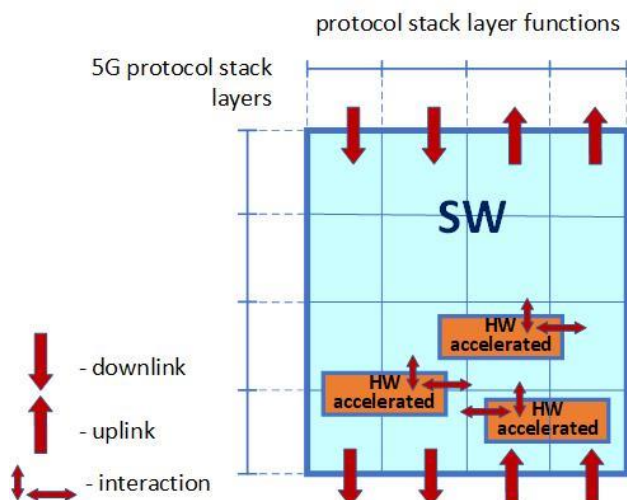


Fig. 3. Defining a set of functions for tight interaction of the CPU with the accelerator

Source: compiled by the authors

In this case, the splitting of functions between the host and the accelerator, as well as the nature of the interaction is not restricted by information flow, since the intensity and nature of data transfer does not have a significant impact on the acceleration value. This makes it possible to effectively implement the concept of a look-aside acceleration, as well as inline acceleration.

In order to choose a technical solution that best meets the requirements of a particular implementation platform, it is necessary to conduct market analysis and research and evaluate the available accelerators according to the following parameters:

- 1) method of information processing;
- 2) method of connecting the accelerator to the host;
- 3) potential throughput of the interaction channels between the host and the accelerator;
- 4) potential performance of the accelerator;
- 5) potential performance of the host (for integrated and tightly-coupled accelerator platforms).

This will be done in the following sections of this paper.

3. Acceleration with a reconfigurable FPGA- or SoC-based SmartNIC

The approach

This approach makes it possible to achieve acceleration with the split of functions of the 5G protocol stack according to the schemes shown in Fig.2. This is done in such a way that all HW-accelerated fragments are performed by the SmartNIC, and the rest (SW) is performed by the host (e.g., CPU, server) if the SmartNIC does not contain CPU, or is split between the SmartNIC's and host's CPUs if it does.

Corresponding diagrams of information processing acceleration with the FPGA-based SmartNIC are shown in Fig. 4 for both cases respectively.

Basic structures

Today, the market of SmartNICs is intensively developing in terms of increasing productivity and throughput of interfaces. Some of them are not reconfigurable, but just programmable, based on CPUs or GPUs.

The reconfigurable SmartNICs available on the market can be divided into 3 categories:

1) FPGA-based. They perform acceleration according to the scheme shown in Fig. 4a with the split of functions as shown in the diagram given on Fig. 2a.

2) FPGA-based, which contain separate CPU and FPGA chips, and

3) SoC-based, in which the CPU and reconfigurable logic are integrated into one chip.

The last two categories make it possible to implement acceleration with the split of functions of the 5G protocol stack as in the diagram shown in Fig. 2b and Fig. 2c, when a part of the software-implemented functions at the layers to be accelerated in the reconfigurable logic is performed in the programmable processors in the SmartNIC, while the rest of the functions (SW) – by the host (e.g., CPU, server), as is shown in Fig. 4b.

Available equipment

Among the FPGA-based SmartNICs, it is worth paying attention to the following:

- Napatech NT200A02 SmartNIC and NT100A01 SmartNIC (based on Xilinx's Virtex UltraScale+ VU5P FPGA) [9] with Link-Inline Software package.

- NickelBoards IRYA Smart NIC, (based on Xilinx Virtex Ultrascale+VU9P which offers upto 2586000 logic cells) [10].

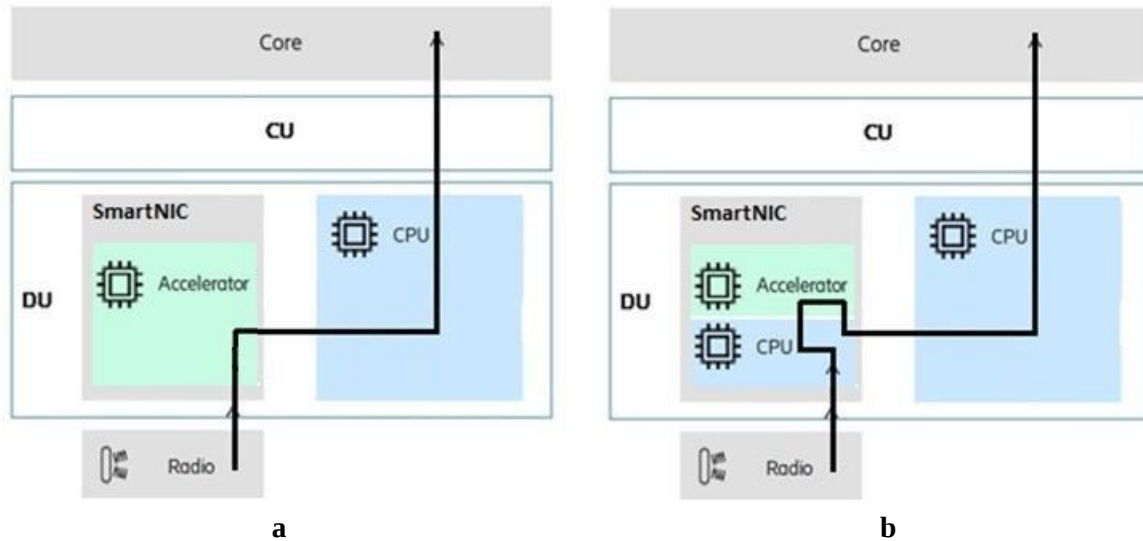


Fig. 4. Data processing acceleration with the FPGA-based SmartNIC:
a – if the SmartNIC does not contain CPU; b – if it does

Source: compiled by the authors

Among the FPGA-based SmartNICs with separate CPU and FPGA chips, it is worth paying attention to the following:

- Inventec FPGA IPU C5020X (Intel Xeon D CPU + Stratix 10 DX FPGA, connected via PCIe Gen3x8) [11]
- Silicom C5010X Data Center SmartNIC (Intel Xeon D CPU + Stratix 10DX FPGA, connected via PCIe Gen3x8.) [12].

Among the SoC-based SmartNICs, it is worth paying attention to the following:

- Silicom FPGA SmartNIC N6010/N6011 (Intel AgileX Based) [13]
- WNC FPGA SmartNIC WSN6050 Series (Intel AgileX Based) [14].
- NickelBoards ADYA TelcoNIC FEC Acceleration Card for 5G DU, (based on Xilinx Xilinx Zynq Ultrascale + MPSoc & RFSoc) [15].
- Silicom TimeSync SmartNIC FB4XXVG@Z21D for hardware acceleration for mobile 4G and 5G Baseband Units or Distributed Units (based on a Xilinx Zynq UltraScale+ RFSoc FPGA) [16].

These facilities are designed for 4G/5G vRAN Acceleration.

Summary on SmartNICs

The approach is effective under the conditions of feasibility to implement a sufficient (or entire) set of functions in SmartNIC, i.e. in two alternative cases:

1. If it is possible to hardware implement all the functions of the lower layers of the 5G protocol stack, which contain the functions nominated for

acceleration - use an FPGA-based SmartNIC. If this case is impossible/impractical, then:

2. If it is possible to execute software-implemented functions of the lower layers of the 5G protocol stack in the SmartNIC, which contain the functions nominated for acceleration - use of the SmartNIC with a programmable processor (FPGA- or SoC-based).

Otherwise, the use of SmartNIC is unlikely to be appropriate.

4. Acceleration with a tightly-coupled FPGA (In-Socket accelerators)

The approach

This approach makes it possible to achieve acceleration with the split of functions of the 5G protocol stack according to the scheme shown in Fig. 3. This is done in such a way that all HW-accelerated fragments are executed by the accelerator, and the rest (SW) by the host (CPU).

In this case, the diagram of information processing acceleration fully corresponds to those shown in Fig. 1. Both acceleration approaches will be effective: inline and look-aside.

Basic structures

Two types of structures of computer systems with tightly coupled reconfigurable logic are introduced in [8]: with tightly coupled reconfigurable logic connected to the memory bus (a) and with tightly coupled reconfigurable logic connected to the processor bus (b). Here, the reconfigurable logic is connected to the universal processor via the system

bus, which ensures high-speed interaction. An accelerator here operates similarly to a coprocessor.

Computer platforms that use AMD and Intel processors are characterized by quite wide possibilities of integration of various devices. Having summarized and systematized the available research and the state of the market, it is possible to single out *three generations* of tightly coupled accelerators architecture, which are reviewed in following subsections of this section of the paper.

The first generation (2006-2008): Intel QuickAssist Technology, AMD Torrenza, HyperTransport and FSB buses

In 2006-2007, AMD initiated the Torrenza platform to improve support for the integration of specialized coprocessors in systems based on Opteron microprocessors directly connecting them to the HyperTransport links. The connection diagram of the main elements of the AMD Torrenza platform, which shows various options for connecting FPGA-based accelerators, is given in [17].

The following connection options are possible for this platform:

- A card with an HT (HyperTransport) interface, which is directly installed in the processor socket.
- A card with an HT interface that is installed in an HTX socket (Hyper Transport eXpansion - a specification of a slot-type interface that has a direct connection to the microprocessor via HyperTransport).
- A card with a PCIe interface that connects to the HT socket via an HT/PCIe bridge.

Intel QuickAssist Technology accelerates cryptographic and compression workloads by offloading the data to hardware capable of optimizing those functions. The principles of integration of the main elements of the Intel platform along with the corresponding diagram are shown in [17].

The following connection options are possible for this platform:

- a card with an FSB interface, which is installed directly into the processor socket;
- a card with a PCI Express interface.

In general, this approach is mainly used to create integral reconfigurable computer systems with high computing power, which have a special architecture design, dedicated printed circuit boards for placing and organizing the interaction of nodes, etc. Examples of such systems are:

- Cray XD1 system, combining universal 64-bit AMD Opteron 200 processors and Xilinx Virtex 4

LX160 FPGAs connected by a RapidArray network [18].

- RASC (Reconfigurable Application Specific Computing) architecture of the SGI and its NUMA-link communication structure. A feature of this communication structure is low data transmission delay, high bandwidth and tight integration of specialized and universal computing elements into a united environment with globally addressed memory.

About a decade ago, several companies offered complete tightly coupled accelerator boards. Below are a few examples.

- The Reconfigurable Processor Unit (RPU) by DRC Computer. It is represented by models of the RPU110 series. The hardware includes a Xilinx Virtex 4 FPGA and additional components that enable the use of motherboard resources. These RPUs are plugged directly into an open 940-pin Opteron system socket and directly access the adjacent dynamic memory and processor on the high-speed HyperTransport bus with nanosecond latency.

- Reconfigurable accelerator RCHTX by Celoxica. A typical accelerator in the RCHTX series is the CELOXICA RCHTX-XV4, which communicates with an AMD processor via the HyperTransport bus.

As of today, Celoxica has repurposed its activities, accelerators are not explicitly shown among their products on the web page, although there is a mention of FPGAs and Celoxica cards. The DRC Computer does not operate today; their accelerators are not available on the market.

The second generation (2008-2017): QPI, Intel Xeon+FPGA Platform, AAL

Tightly coupled systems of the first generation were dependent on the specification of the equipment used. Taking into account the complexity and duration of development and production of systems on the one hand and the rapid evolution of universal processors and reconfigurable logic devices on the other, this was their significant drawback – the systems quickly became “obsolete”.

It can take six months or longer to design and build a complex custom PCB containing processor(s), FPGA(s), memory, interfaces, and other components. Application code developers must wait until communication mechanisms are designed, built, and debugged so that their application software will interface to the board hardware. This communication layer adds to the cost of developing new accelerator technology. It also locks in application software to the specific communication mecha-

nism used on a specific board design. Software may need to be substantially changed when board design changes or even when new versions of accelerators are released [19].

Therefore, along with the moving ahead to new interfaces, Intel QuickAssist Technology includes both third-party In-Socket Accelerator (ISA) FPGA modules and an Acceleration Abstraction Layer (AAL) developed by Intel. The very implementation of AAL is a characteristic feature of the second generation tightly coupled accelerators. The AAL provides a consistent interface for application software so that underlying accelerator and general-purpose processor hardware can evolve independently and application software can more easily scale. The AAL does not define domain-specific libraries or functions. Instead, it provides consistent interfaces that existing libraries and frameworks can use to interface to hardware accelerator modules (see Fig. 5).

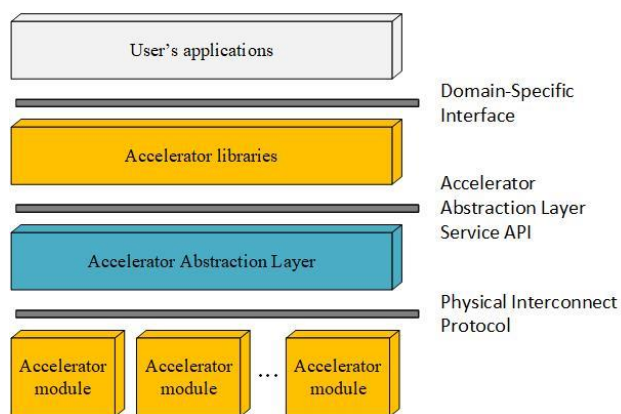


Fig. 5. Intel Acceleration Abstraction Layer
Source: compiled by the authors

When the FSB interface was retired and replaced by QPI, Intel proposes a Xeon+FPGA Platform [20].

The Intel Quick Assist Technology is licensed to select vendors and consists of: QPI Referenced RTL, Software and Applications, Simulation Environment, and Validation Environment. One of the vendors is Pactron, offering QPI Software Development Platforms [21] based on Intel Quick Assist Technology.

It is stated in [19] that Intel has been working with third parties such as XtremeData to develop a comprehensive approach to hardware-based acceleration. However, there is no information on such developments on the XtremeData web page. In publication [22] it is told about the XD2000i Family of In-Socket Accelerators, which at the time of the publication issuing is available, as well as about the XD2000i Development System, that has a single unit price of \$23.000.

The third generation (from 2017 till now): UPI, CXL, giants' integration

Intel's purchase of Altera in 2015 for \$16.7 billion and AMD's purchase of Xilinx in 2022 for \$49 billion were bright events of recent years in the semiconductor market. These events, without exaggeration, have a fundamental impact on the further development of computer engineering, because 2 giants of the market of universal processors have acquired 2 giants of the market of reconfigurable logic devices. Therefore, we should expect further development of the direction of heterogeneous computing and a leap in the evolution of reconfigurable computer systems.

We will see how the acquisition of Xilinx by AMD will be reflected in the near future. Let's consider today's trends in the development of tightly coupled systems of the Intel architecture.

The Intel Ultra Path Interconnect (UPI) replaced the Intel QuickPath Interconnect (QPI) in Xeon Skylake-SP platforms starting in 2017. The UPI is Intel's proprietary (the same as the QPI) point-to-point cache-coherent interconnect which facilitates Intel Xeon symmetric multiprocessing support. It is a low-latency coherent interconnect for scalable multiprocessor systems with a shared address space. Supporting processors typically have two or three UPI links. Operating at up to 11.2 GT/s, each UPI link has a bandwidth of 22.4 GB/s. Intel says that compared to standard PCIe, UPI allows for up to 37 % lower latency on memory coherent applications.

A number of Intel FPGAs of the latest series are primarily prepared for the creation of tightly coupled reconfigurable systems. In particular, this applies to Intel Stratix 10 [23] and Intel Agilex FPGA and SoC FPGA [24] with UPI and PCIe Gen4 interfaces.

Further on, Intel Agilex with PCIe Gen5 is built on those capabilities. Instead of using UPI, in the PCIe Gen5 generation of silicon, Intel adopts CXL (Compute Express Link) Interconnect. Intel said it may bring UPI to Agilex and 10nm, but actually, the plan is to use CXL.

The CXL Interconnect is a processor (or processor to processor) interconnects rather than a data center interconnects like Ethernet. The CXL is being an alternate protocol running over the PCIe (Gen 5.0) physical layer. Intel says it needs a new class of interconnect because while PCIe is great, future computing will need low latency communication with coherent memory pools rather than isolated pools with PCIe. Intel is not going to use UPI because the company thinks that would have been the wrong solution for heterogeneous computing and an open ecosystem [25].

The key advantage of CXL over QPI and UPI is that it is an open standard – openly accessible and usable by anyone.

It is noteworthy that the latest Intel developments, such as the fourth generation Xeon Sapphire Rapids processor and Agilex FPGAs family are featuring both CXL and PCIe Gen5 [26]. An actual version of CXL specifications was released in 2022, and it is based on PCIe 6.0 PHY.

Summary on tightly coupled accelerators

Summarizing the above, we can conclude that tightly coupled architecture is a good potential computing platform for hardware acceleration of 5G functions. Its advantages are scalability and high speed of interaction between the host and the accelerator. The disadvantages include the need to use licensed interfaces (QPI and UPI, although the CXL open standard is also available today) and technologies (Intel QuickAssist Technology), as well as the lack of COTS solutions, that requires for creation of custom system architecture and raises the complexity and cost of system design.

Along with this, it is worth conducting an additional comparative analysis with today's accelerators of loosely coupled architecture in terms of the speed of interaction between the host and the accelerator. After all, the official PCIe 5.0 standard which came out in May 2019 brought 128 GBps of throughput (PCIe is most often used in loosely coupled architecture). Moreover, in 2022 the final PCIe 6.0 specification has already appeared, which offers twice the throughput of the previous version, and on 21 June 2022, PCI-SIG announced the development of PCI Express 7.0 specification with 128 GT/s raw bit rate and up to 242 GB/s per direction in x16 configuration. As an argument, let's compare the throughput of interfaces used in tightly coupled architecture with PCIe (see Table 1.) [27].

Thus, additional experimentations on the characteristics of recent PCIe-connected loosely coupled accelerators may be needed. It is also necessary to further investigate delays introduced by the interaction through the peripheral interface, which may be critical for the use of loosely coupled accelerators and reduce the high theoretical bandwidth of the new generations of PCIe. This requires practical experiments.

5. Acceleration with an integrated reconfigurable SoC-based platform

The approach

This approach, same as the previous one, makes it possible to achieve acceleration with the split of

functions of the 5G protocol stack according to the scheme shown in Fig. 3. This is done in such a way that all HW-accelerated fragments are executed by the accelerator and the rest (SW) by the host (CPU). Both computing components (CPU and FPGA) are integrated into one VLSI (see Fig. 6). In this case, the diagram of information processing acceleration fully corresponds to the one shown in Fig. 1. Both acceleration schemes will be effective – inline and look-aside. The approach can be considered in case when the SoC resources are sufficient to implement entire 5G protocol stacks functionality with required technical characteristics, at first in terms of performance.

Basic structures

There are 2 basic types of integrated reconfigurable accelerators and corresponding two types of computer platforms with integrated reconfigurable logic [8]:

- with reconfigurable logic integrated into the ALU of the universal processor (on-core accelerator);
- with a universal processor and specialized accelerators integrated into the reconfigurable logic (on-chip accelerator).

They make it possible to achieve the highest speed of interaction between a universal processor and an accelerator. In the first case, accelerators are used as functional units of a universal processor by expanding its instruction set. In the second one, the universal processor is embedded into the reconfigurable environment (as programmable fabric) and performs the functions of controlling accelerators, being implemented “hardly” or as a soft IP core.

Architecturally, the first approach corresponds to the concept of a processor with an extended instruction set (EISP). Such architecture consists of one or more specialized hardware functional units which can accelerate critical portions of an application kernel; for example, the body of an inner loop for an algorithm or a sequence of trigonometric functions. This type of accelerator is located inside, or very close to, the processing core. The CPU core shares key resources (register file, memory-management unit and L1 data cache) with the integrated accelerator (IA), and thus stalls until the IA completes execution. From a hardware viewpoint, however, IAs can pose integration challenges. First, they further complicate the design of the CPU. Second, they can pose timing closure challenges, since it is common to require the IA logic to meet the same clock-frequency constraints that are set for the CPU. Third, they have limited portability across different system designs, since it is often necessary to adapt the accelerators' interfaces to CPU-dependent structures.

Table 1. Comparison of the throughput of interfaces used in tightly coupled architecture with PCIe

Interface	Lanes	Transfer Speed	Theoretical Bandwidth (unidirectional)	Typical Bandwidth (in Practice)
PCIe Gen 3	x16	8 GT/s,	16 GB/s	12,1 GB/s
PCIe Gen 4	x16	16 GT/s,	32 GB/s	26,2 GB/s
PCIe Gen 5	x16	32 GT/s,	64 GB/s	No data
CXL (based on PCIe Gen 5)	x16	32 GT/s,	64 GB/s	No data
Intel QPI	x20	9,6 GT/s,	19,2 GB/s	No data
Intel UPI	x20	10,4 GT/s,	20,8 GB/s	No data
AMD HyperTransport Link 3.1	-	6,4 GT/s	12,8 GB/s	No data

Source: compiled by the [27]

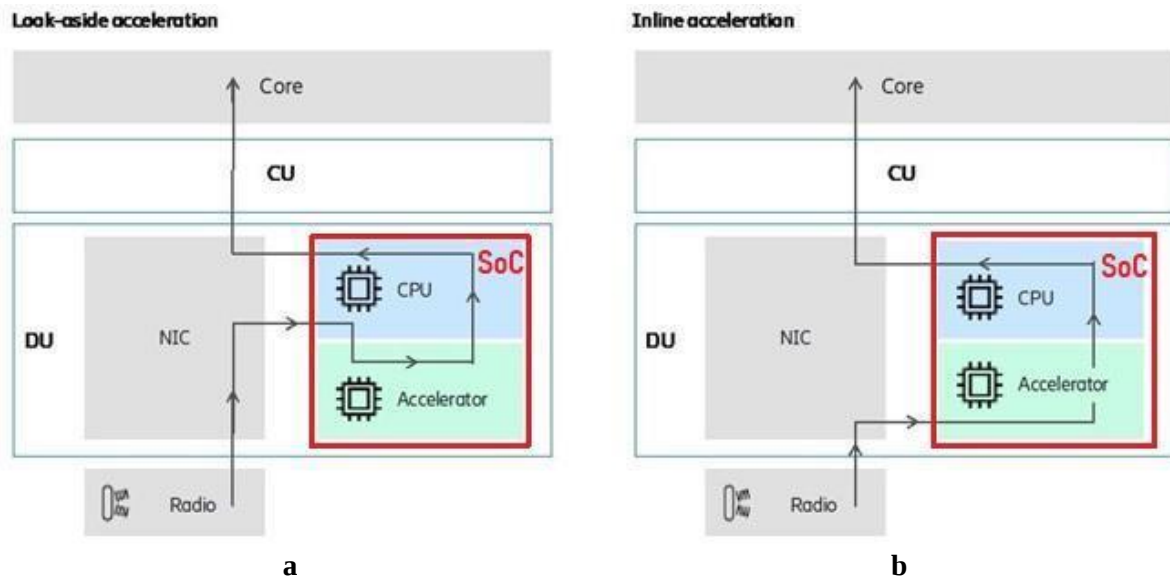


Fig. 6. Data processing acceleration with the integrated reconfigurable SoC-based platform:
a – look-aside; b – inline

Source: compiled by the authors

According to the studies highlighted in the work [28], the speedup is on average much greater for tightly coupled accelerators (TCAs) than for an integrated one. Based on this, we can conclude that the use of EISP-based integrated architecture (on-core) is unlikely to be appropriate for acceleration of 5G protocol stack functions.

The second approach architecturally corresponds to the concept of a specialized system on a chip (SoC). It is theoretically inferior in speed, since operations are performed by accelerators not as CPU instructions, but as external ones, through the driver and the OS (if it is not a bare metal). However, it has advantages over the previous one, which are that the accelerators are separated from the CPU core, their operation is not performed within the instruction cycle, does not depend on the contents of the cache memory, they can communicate with the memory via DMA and have their own memory. These advantages are generally characteristic of tightly coupled architecture. Considering the speedup results shown in [28], it can be predicted that the speedup value for

the integrated on-chip architecture will be even higher.

Therefore, this type of the integrated architecture can be considered as a potential computing platform for the hardware acceleration of 5G functions. However, one should take into account the basic limitation of integrated accelerators, namely: the limitation of reconfigurable logic resources by the capacity of the chip and the lack of the possibility of their increase.

Let's consider what the leading companies, primarily Intel and AMD, offer in this direction today.

SoC-based integrated hardware accelerator platforms by AMD Xilinx

Today AMD Xilinx offers 4 groups of products that we can use as a SoC-based integrated hardware platform to implement acceleration of the 5G protocol stack functions:

1. Versal Adaptive Compute Acceleration Platform (ACAP) [29]. The Versal AI Core is effective for 5G Wireless Beamforming tasks [30]. Versal

Prime Series is effective for application in 5G xHaul and as a SmartNIC [31].

2. Zynq-7000 SoC family devices, which are equipped with dual-core ARM Cortex-A9 processors integrated with 28nm Artix-7 or Kintex-7 based programmable logic with up to 6.6M logic cells. It is the first generation of Xilinx fully programmable SoCs, and its performance characteristics are hardly sufficient for their use as a hardware accelerators platform for performing 5G functions. In particular, this applies to the Processing System, which is based on a dual-core ARM Cortex A9 CPU.

3. Zynq UltraScale+ MPSoC, which are equipped with Dual Arm Cortex-A53 and Dual Arm Cortex-R5F processors integrated with 16nm FinFET+ Programmable Logic. An important difference between it and the Zynq-7000 SoC is the availability of PCIe Gen4 and significantly higher performance in the Processing System part.

4. Zynq UltraScale+ RFSoC. According to AMD Xilinx, it is “the Industry’s Only Single-Chip Adaptable Radio Platform” [32]. This type of SoC is particularly interesting for the implementation of 5G systems given that the chips are equipped with hardened IP for critical Digital Front-End processing for up to 400MHz bandwidth for 5G mass radio deployment (Zynq RFSoC DFE) [33].

SoC-based integrated hardware accelerator platforms by Intel Altera

In one of its white papers [34], Intel declares that “*With Intel Xeon-D CPUs, Intel Agilex FPGAs, Intel eASIC devices, and ASIC technologies, Intel is the only company on the planet that has an end-to-end silicon solution for 5G radio access networks*”.

A number of Intel’s and its partners’ products of accelerator cards and SmartNICs are available. These products are equipped with the Agilex SoC as an integrated hardware acceleration platform.

Intel Agilex SoCs features among others include:

- Intel Advanced 10nm FinFET (3rd generation) process;
- Hard PCI Express Gen4 x16 (up to 16 Gbps per lane) and Gen5 x16 (up to 32 Gbps per lane) IP blocks;
- CXL Hard IP block for cache-coherent and memory-coherent interfacing to Intel Xeon CPU;
- Quad-core 64-bit ARM Cortex A53 embedded processors running up to 1.4 GHz.

In contrast to Xilinx’s SoCs, Intel Agilex SoCs do not contain hard IP cores for direct use in 5G sys-

tems, such as LDPC. At the same time, there is the AES-256 encryption hard IP core. However, the current 3GPP specifications specify for encryption the AES-128 algorithm from the AES family.

Summary on integrated accelerators

An integrated accelerator makes it possible to effectively implement both inline and look-aside acceleration with less design complexity compared to a tightly coupled accelerator. It should be noted that in the SoC-based SmartNICs the same SoCs are deployed that were discussed above. This means that this approach can be considered in case when the SoC resources are sufficient to implement entire 5G protocol stack functionality with required technical characteristics, at first in terms of performance.

In terms of functionality, Xilinx’s SoCs are better suited for our tasks than Intel’s, as they contain hard IP blocks for LDPC encoding/decoding and FFT/iFFT. In terms of the processing system, these SoCs are similar, as they both use built-in ARM CPUs.

Problematic issues of the application of integrated reconfigurable accelerators:

1. Such accelerators are designed as complete modules (SoM) that are to be integrated into the existing information and communication infrastructure, or to be installed as a VLSI on a specially designed (custom) PCB.

2. The volume of reconfigurable logic resources and the performance of the processing system are strictly limited in the SoC.

6. Comparison of acceleration approaches

Table 2 shows comparison of accelerators of different types according to basic characteristics (except for the price, which can vary widely for each type of accelerator), based on the results of the above analysis. The Table 3 lists their main advantages and disadvantages and provides important comments on application.

The presented in Tables 2 and 3 results of analysis can be used as a basis for developing a technical solution for the implementation of a computing platform to execute the functions of the 5G protocol stack with hardware acceleration aimed to reach the maximum speedup gain on 5G protocol stack execution of PHY and MAC layers functions.

Table 2. Comparison of accelerators of different types according to basic characteristics

Accelerator type Characteristics	SmartNIC-based	LCA	TCA	IA
Method of data processing	Inline	Inline and look-aside	Inline and look-aside	Inline and look-aside
Method of connecting the accelerator to the host	Loosely coupled	Loosely coupled	Tightly coupled	Very tightly coupled
Potential throughput of the interaction channels	Not applicable	Low	High	Very high
Potential performance of the accelerator	High / moderate, scalable (we can replace the NiC without affecting the host)	Very high, scalable (we can replace the accelerator without affecting the host)	Very high / high, scalable but restricted by the host computer design features	Moderate / high but lower as comparing to LCA/TCA approaches, not scalable (fixed)
Potential performance of the CPU	Not applicable	Not applicable	Scalable, but rigidly defined by the PCB(s) topology	Fixed
Amount and scope of reconfigurable logic resources, scalability	Scalable, but restricted by device features	Scalable, not restricted	Scalable, but rigidly defined and restricted by the PCB(s) topology	Fixed

Source: compiled by the authors

Table 3. Main features of accelerators deployment

Accelerator type	Advantages	Disadvantages	Comments
SmartNIC-based	Accelerator is easily deployable and replaceable (PCIe connection), and independent from the host computer.	Requires for appropriate functional split, allowing for inline acceleration.	• Moderate to high acceleration is reachable, depending on FPGA capacity (finally this is NIC, not an accelerator for HPC); • There are (CPU+FPGA)-based SmartNICs, allowing for execution of a part of CPU-driven SW code in the SmartNIC.
LCA	• Accelerator is easy deployable and replaceable (PCIe connection), and independent from host computer; • We can reach for required acceleration gain by choosing an accelerator of the required performance.	Communication bottleneck: data transfer size and intensity have to be minimized.	There are SoC-based (CPU+FPGA) LCAs, allowing for offloading part of CPU-driven SW code from CPU to the accelerator's universal processor.
TCA	No communication bottleneck.	• Hardly feasible, especially in short development time due to lack of COTS solutions on the market. • Accelerators are hardly replaceable.	Can be a very effective solution for long-term stable products.
IA	No communication bottleneck.	Resource-constrained device.	Moderate to high acceleration is reachable, for tasks requiring not a high computational resource.

Source: compiled by the authors

CONCLUSIONS

In 5G telecommunication systems, there is a need for fast processing of data packets on lower layers, in particular PHY, MAC, PDCP. Given the relatively small size of the data packet, execution of individual functions of these layers using hardware acceleration is a challenging task. In the course of practical experiments, it turned out that the most typical way to gain acceleration – connecting the reconfigurable accelerator board to the host via a high-performance PCIe peripheral interface, is ineffective.

The reason for acceleration downgrade is the communication bottleneck, which is caused by 1) a look-side method of organizing data exchange with the accelerator, and 2) deployment the a loosely coupled accelerator architecture via the PCIe peripheral interface. In order to resolve the communication bottleneck, in this paper the following architecture options were investigated:

- reconfigurable FPGA- or SoC-based SmartNIC,
- tightly-coupled FPGA-based accelerator, and integrated reconfigurable SoC-based platform. An extensive analysis of architectural approaches to implement a reconfigurable hardware platform for acceleration of 5G protocol stack functions was carried out.

It has been found that:

1. The SmartNIC deployment can be effective under the conditions of the feasibility to implement a sufficient (or entire) set of functions in SmartNIC, i.e. in two alternative cases:
 - a. If it is possible to hardware implement all the functions of the lower layers of the 5G protocol stack, which contain the functions nominated for acceleration - use an FPGA-based SmartNIC. If this case is impossible/impractical, then:
 - b. If it is possible to execute software-implemented functions of the lower layers of the 5G protocol stack in the SmartNIC, - use of the SmartNIC with a programmable processor (FPGA- or SoC-based).

Otherwise, the use of SmartNIC is unlikely to be appropriate.

2. The tightly coupled architecture is a good potential computing platform for hardware acceleration of 5G functions. Its advantages are scalability and high speed of interaction between the host and the accelerator. The disadvantages include the need to use licensed interfaces (QPI and UPI, although the CXL open standard is also available today) and technologies (Intel QuickAssist Technology), as well as the lack of COTS solutions, that requires for creation of custom system architecture and raises the complexity of system design. Along with this, with the release of new generations of PCIe - 5.0 and 6.0 with their high bandwidth, a loosely coupled architecture with its other advantages can be a good alternative to a tightly coupled one.

3. An integrated SoC-based accelerator makes it possible to effectively implement both inline and look-aside acceleration with less design complexity compared to a tightly coupled accelerator.

Two issues of their deployment are following:

- a. Such accelerators are designed as complete systems on modules that are to be integrated into the existing information and communication infrastructure, or to be installed as a VLSI on a specially designed (custom) printed circuit board. The question arises of the feasibility and expediency of such an application as an alternative to a traditional server platform.
- b. The volume of reconfigurable logic resources and the performance of the processing system are strictly limited in the SoC.

The results of the analysis are summarized in tables. They can be used as a basis for developing a technical solution for the implementation of a computing platform to perform the functions of the 5G protocol stack with hardware acceleration aimed to reach the maximum speedup gain on PHY and MAC layers functions.

REFERENCES

1. Redana, S., Bulakci, Ö., Mannweiler, C. *et al.* “5G PPP Architecture Working Group – View on 5G Architecture”. Version 3.0 (3.0). Zenodo. 2019. DOI: <https://doi.org/10.5281/zenodo.3265031>.
2. Borromeo, J. C., Kondepu, K., Andriolli, N. & Valcarenghi, L. “An overview of hardware acceleration techniques for 5G functions”. *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. Bari: Italy. 2020. p. 1–4, <https://www.scopus.com/authid/detail.uri?authorId=57189099331>. DOI: <https://doi.org/10.1109/ICTON51198.2020.9203242>.
3. Chamola, V., Patra, S., Kumar, N. & Guizani, M. “FPGA for 5G: Re-configurable Hardware for Next Generation Communication”. In *IEEE Wireless Communications*. 2020; 27 (3): 140–147,

<https://www.scopus.com/authid/detail.uri?authorId=55329792700>. DOI: <https://doi.org/10.1109/MWC.001.1900359>

4. Civerchia, F., Pelcat, M., Maggiani, L., Kondepu, K., Castoldi, P. & Valcarengi, L. “Is OpenCL driven reconfigurable hardware suitable for virtualising 5G infrastructure?” In *IEEE Transactions on Network and Service Management*. 2020; 17 (2): 849–863, <https://www.scopus.com/authid/detail.uri?authorId=57193016471>. DOI: <https://doi.org/10.1109/TNSM.2020.2964392>.

5. Borromeo, J. C., Kondepu, K., Andriolli, N. & Valcarengi, L. “FPGA-accelerated SmartNIC for supporting 5G virtualized radio access network”. *Computer Networks*. 2022. Vol. 210. ISSN 1389-1286, <https://www.scopus.com/authid/detail.uri?authorId=57189099331>. DOI: <https://doi.org/10.1016/j.comnet.2022.108931>.

6. Papatheofanous, E. A., Reisis, D. & Nikitopoulos, K. “LDPC hardware acceleration in 5G open radio access network platforms”. In *IEEE Access*. 2021; 9: 152960–152971. DOI: <https://doi.org/10.1109/ACCESS.2021.3127039>.

7. Riback, M. & Toorell, O. “How to get the most out of 5G mid-band in Cloud RAN”. 2022. – Available from: <https://www.ericsson.com/en/blog/2020/10/how-to-get-the-most-out-of-5g-mid-band-in-cloud-ran>. – [Accessed: 7 th March, 2023].

8. Todman, T., Constantinides, G., Wilton, S., Mencer, O., Luk, W. & Cheung, P. “Reconfigurable computing: Architectures, design methods, and applications”. *IEE Proceedings on Computers and Digital Techniques*. 2005; 152 (2): 193–207.

9. “SmartNICs – Napatech.” *Napatech*. – Available from: <https://www.napatech.com/products/>. – [Accessed: 9-th March, 2022].

10. “SSL Acceleration, and OvS Off-Load | IRYA Smart NIC.” Network interface cards (Smart NIC), Network Accelerator Cards. – Available from: <https://www.nickelboards.com/irya-smart-network-interface-card-xilinx-virtex-ultrascale-xcvu9p-xcvu7p>. [Accessed: 9-th March, 2022].

11. “Inventec FPGA IPU C5020X.” *Inventec*. – Available from: <https://ebg.inventec.com/en/product/Accessories/Smart%20NIC%20Card/Inventec%20FPGA%20IPU%20C5020X#0>. – [Accessed: 9-th March, 2022].

12. “Silicom C5010X Data Center SmartNIC for Accelerated Virtualization.” *Napatech*, – Available from: <https://www.napatech.com/support/resources/data-sheets/silicom-c5010x-data-center-smartnic/>. – [Accessed: 9-th March, 2022].

13. “Silicom FPGA SmartNIC N6010/6011.” *Silicom Denmark*. – Available from: <https://www.silicom.dk/product-details/silicom-fpga-smartnic-n601x/>. – [Accessed: 9-th March, 2022].

14. Wistron NeWeb Corporation (WNC). “FPGA SmartNIC Acceleration Card WSN6050 Series | Network Infrastructure | Network Access Solutions | Wistron NeWeb Corporation.” WNC. – Available from: https://www.wnc.com.tw/index.php?action=pro_detail&top_id=106&scid=142&tid=99&id=287. – [Accessed: 9-th March, 2022].

15. “ADYA TelcoNIC | 5G Acceleration Card.” *Network Interface Cards (Smart NIC), Network Accelerator Cards*. – Available from: <https://www.nickelboards.com/adya-5g-acceleration-card>. – [Accessed: 9-th March, 2022].

16. “TimeSync SmartNIC FPGA FB4XXVG@Z21D.” *Silicom Denmark*. – Available from: <https://www.silicom.dk/product-details/timesync-smartnic-fpga-fb4xxvgz21d/>. – [Accessed: 9-th March, 2022].

17. Melnyk, A. & Melnyk, V. “Personal supercomputers: Architecture, design, application”: Monograph (in Ukrainian). *Lviv Polytechnic National University Publishing*. 2013. 516 p.

18. “CRAY XD1 DATASHEET.” *The University of New Mexico*. 2004. – Available from: http://www.hpc.unm.edu/~tlthomas/buildout/Cray_XD1_Datasheet.pdf. – [Accessed: 9-th March, 2020].

19. Carlston, P. “Tightly coupling FPGAs with X86 Processors.” *Signal Processing*, INTEL. – Available from: <http://signal-processing.mil-embedded.com/articles/tightly-coupling-fpgas-x86-processors/>. – [Accessed: 9-th March, 2021].

20. Gupta, P. K. “Intel Xeon + FPGA Platform for the Data Center”. In *FPL’15 Workshop on Reconfigurable Computing for the Masses*, September 2015. – Available from: <https://reconfigurablecomputing4themasess.net/files/2.2%20PK.pdf>. – [Accessed: 9-th March, 2020].

21. “Pactron’s V Series QPI software development platforms.” *Oactron Inc.* – Available from: <https://www.pactroninc.com/qpi/Pactron%20V%20series%20QPI%20SDP%20102115.pdf>. – [Accessed: 9-th March, 2022].
22. Maxfield, C. “XtremeData’s XD2000i in-Socket accelerator based on multiple stratix III FPGAs.” *EETimes*. 2008. – Available from: <https://www.eetimes.com/xtremedatas-xd2000i-in-socket-accelerator-based-on-multiple-stratix-iii-f>. – [Accessed: 9-th March, 2020].
23. Kennedy, P. “Intel stratix 10 adds UPI and PCIe Gen4 Ready for CXL.” *ServeTheHome*. 2019. – Available from: <https://www.servethehome.com/intel-stratix-10-adds-upi-and-pcie-gen4-ready-for-cxl/>. – [Accessed: 9-th March, 2021].
24. “Intel Agilex FPGA and SoC FPGA Family.” *Intel*, – Available from: <https://www.intel.com/content/www/us/en/products/details/fpga/agilex.html>. – [Accessed: 9-th March, 2021].
25. Kennedy, P. “New CXL details at intel interconnect day 2019 – ServeTheHome”. *ServeTheHome*. 2019. – Available from: <https://www.servethehome.com/new-intel-cxl-details-at-intel-interconnect-day-2019>. – [Accessed: 9-th March, 2020].
26. Nassif N. *et al.* “Sapphire rapids: The next-generation Intel Xeon Scalable Processor”. 2022 *IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco: CA, USA. 2022. p. 44–46, <https://www.scopus.com/authid/detail.uri?authorId=54400389500>. DOI: 10.1109/ISSCC42614.2022.9731107.
27. “Performance characteristics of common transports and buses - Microway.” *Microway*. – Available from: <https://www.microway.com/knowledge-center-articles/performance-characteristics-of-common-transports-buses>. – [Accessed: 9-th March, 2021].
28. Cota, E. G., Mantovani, P., Di Guglielmo, G. & Carloni, L. P. “An analysis of accelerator coupling in heterogeneous architectures”. *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. San Francisco: CA, USA. 2015. p. 1–6. <https://www.scopus.com/authid/detail.uri?authorId=55368592500>. DOI: <https://doi.org/10.1145/2744769.2744794>.
29. “Versal: The first adaptive compute acceleration platform (ACAP)”. *Xilinx*. 2020. – Available from: https://www.xilinx.com/content/dam/xilinx/support/documents/white_papers/wp505-versal-acap.pdf. – [Accessed: 9-th March, 2021].
30. “System-Level benefits of the Versal Platform.” *Xilinx*. 2022. – Available from: https://www.xilinx.com/content/dam/xilinx/support/documents/white_papers/wp539-versal-system-level-benefits.pdf. – [Accessed: 9-th March, 2022].
31. “Versal Prime Series VMK180 Evaluation Kit.” *Xilinx*. – Available from: <https://www.xilinx.com/products/boards-and-kits/vmk180.html>. – [Accessed: 9 th March, 2022].
32. “Zynq UltraScale+ RFSoc.” *Xilinx*, <https://www.xilinx.com/products/silicon-devices/soc/rfsoc.html>. Accessed 9-Mar. 2022.
33. “Wired & Wireless Communications.” *Xilinx*. – Available from: <https://www.xilinx.com/applications/wired-wireless.html>. – [Accessed: 9-th March, 2022].
34. “Build Cost-Effective & Efficient 5G Radios with Intel Agilex FPGA.” *Intel*. – Available from: <https://www.intel.com/content/www/us/en/products/docs/programmable/build-efficient-5g-radios-with-agilex-fpga.html>. – [Accessed: 9-th March, 2021].

Conflicts of Interest: the authors declare no conflict of interest

Received 15.01.2023

Received after revision 28.03.2023

Accepted 02.04.2023

DOI: <https://doi.org/10.15276/aait.06.2023.7>

УДК 004.383

Дослідження реконфігурованих апаратних платформ для прискорення виконання функцій стеку протоколів 5G

Мельник Віктор Анатолійович^{1), 2)}

ORCID: <https://orcid.org/0000-0002-5046-8002>; viktor.melnyk@kul.pl. Scopus Author ID: 57200786767

Гамоля Владислав Вікторович ²⁾

ORCID: <https://orcid.org/0000-0002-2954-455X>; skovbasa@ukr.net. Scopus Author ID: 57222387801

¹⁾ Люблінський Католицький Університет Івана Павла II, Al. Raclawickie 14, 20-950 Люблін, Польща

²⁾ Національний університет «Львівська політехніка», вул. Степана Бандери, 12, Львів, 79013, Україна

АНОТАЦІЯ

Open RAN та 5G – це дві ключові технології, які покликані якісно вдосконалити мережеву інфраструктуру і надати більшу гнучкість і ефективність мобільним операторам і користувачам. 5G створює нові можливості для високошвидкісного інтернету, Інтернету речей, телемедицини та багатьох інших застосувань, а Open RAN впроваджує відкриті та стандартизовані мережеві архітектури, що дає можливість зменшити вартість та ризик для операторів і сприяє подальшим інноваціям. З огляду на зростаючу кількість користувачів і об'ємів інформації, суто програмна реалізація окремих функцій протоколу 5G, а особливо обчислювально складних, вимагає значних комп'ютерних ресурсів і енергії. Тому одним з напрямків активної діяльності в розробленні засобів для 5G систем є апаратне прискорення виконання таких функцій, що забезпечує можливість обробки великих об'ємів даних в реальному часі та з високою ефективністю. У цій статті автори досліджують можливості реалізації апаратного прискорення з застосуванням реконфігурованих комп'ютерних компонентів – пристроїв на базі програмованих логічних інтегральних схем (ПЛИС), порівнюють і оцінюють архітектурні альтернативи для реалізації обчислювальної платформи для виконання функцій нижчих рівнів стеку протоколів 5G (PHY та MAC) з апаратним прискоренням.

Ключові слова: відкрита мережа радіодоступу; 5G; апаратне прискорення; програмовна логічна інтегральна схема
Copyright © Національний університет «Одеська політехніка», 2023. Всі права захищені

ABOUT THE AUTHORS



Viktor A. Melnyk - Doctor of Engineering Sciences, Professor of Department of Natural and Technical Sciences, John Paul II Catholic University of Lublin, Al. Raclawickie 14, 20-950 Lublin, Poland, Professor of the Department of Information Technologies Security, Lviv Polytechnic National University, 12, St. Bandery Str. Lviv, 79013, Ukraine
ORCID: <https://orcid.org/0000-0002-5046-8002>; viktor.melnyk@kul.pl. Scopus Author ID: 57200786767

Research fields: Computer systems architecture research and design; IP cores design; high-performance reconfigurable computer systems design; computer data protection; cryptographic processors design and implementation; wireless sensor network security

Мельник Віктор Анатолійович - доктор технічних наук, професор кафедри Природничих і технічних наук, Люблінський католицький університет ім. Івана Павла II, Al. Raclawickie 14, 20-950 Lublin, Польща. Професор кафедри Безпеки інформаційних технологій, Національний університет «Львівська політехніка», вул. Ст. Бандери, 12. Львів, 79013, Україна



Vladyslav V. Hamolia - PhD student, Department of Information Technologies Security, Lviv Polytechnic National University, 12, St. Bandery Str. Lviv, 79013, Ukraine
ORCID: <https://orcid.org/0000-0003-2444-0775>; vladyslav.v.hamolia@lpnu.ua. Scopus Author ID: 57222387801

Research fields: Heterogeneous computing; computer systems architecture research and design; machine learning; deep learning

Гамоля Владислав Вікторович - аспірант кафедри Безпеки інформаційних технологій, Національний університет «Львівська політехніка», вул. Ст. Бандери, 12. Львів, 79013, Україна