TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 4, August 2021, pp. 1291~1306
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i4.18400

FPGA based technical solutions for high throughput data processing and encryption for 5G communication: A review

P. Visconti[1], R. Velazquez[2], Carolina Del-Valle-Soto[3], R. de Fazio[4]
[1,4]Department of Innovation Engineering, University of Salento, Lecce, Italy
[2]Facultad de Ingeniería, Universidad Panamericana, Aguascalientes, Mexico
[3]Facultad de Ingeniería, Universidad Panamericana, Jalisco, Mexico

Article Info

Article history:

Received Nov 2, 2020
Revised Mar 18, 2021
Accepted Mar 30, 2021

Keywords:

5G communication
Advanced encryption standard
Field programmable gate array
High-speed processing
Xilinx ZCU102

ABSTRACT

The field programmable gate array (FPGA) devices are ideal solutions for high-speed processing applications, given their flexibility, parallel processing capability, and power efficiency. In this review paper, at first, an overview of the key applications of FPGA-based platforms in 5G networks/systems is presented, exploiting the improved performances offered by such devices. FPGA-based implementations of cloud radio access network (C-RAN) accelerators, network function virtualization (NFV)-based network slicers, cognitive radio systems, and multiple input multiple output (MIMO) channel characterizers are the main considered applications that can benefit from the high processing rate, power efficiency and flexibility of FPGAs. Furthermore, the implementations of encryption/decryption algorithms by employing the Xilinx Zynq Ultrascale+MPSoC ZCU102 FPGA platform are discussed, and then we introduce our high-speed and lightweight implementation of the well-known AES-128 algorithm, developed on the same FPGA platform, and comparing it with similar solutions already published in the literature. The comparison results indicate that our AES-128 implementation enables efficient hardware usage for a given data-rate (up to 28.16 Gbit/s), resulting in higher efficiency (8.64 Mbps/slice) than other considered solutions. Finally, the applications of the ZCU102 platform for high-speed processing are explored, such as image and signal processing, visual recognition, and hardware resource management.

Corresponding Author:

P. Visconti
Department of Innovation Engineering
University of Salento
Road to Monteroni, Ecotekne Campus, Lecce, 73100, Italy
Email: paolo.visconti@unisalento.it

1. INTRODUCTION

The 5G-based communication technology is revolutionizing several fields of human and industrial activities, opening several perspectives for enhancing productivity, connectivity, automation, scalability and the interaction between humans and machines or networks, for instance, in health-care provisioning, industry 4.0, and internet of things (IoT) fields [1]. The 5G systems/networks are featured by larger network capacity, wider bandwidth, higher spectrum efficiency, and throughput, as well as low latency and power consumption, enabling high mobility and dense connectivity thanks to software-defined networking (SDN) as well as network function virtualization (NFV) approach [2]. These last provide to the network great flexibility and ability to be reconfigured for supporting different network functionalities, such as address translation, tunnelling, load balancing, and security tasks. Three different models are employed in the 5G systems' development, low-energy massive machine type communication (mMTC), ultra-reliable low-latency communication (URLLC), and high throughput mobile broadband (eMBB). The first one, including smart city and IoT, requires high-density and low-frequency packet transmission, whereas, in URLLC systems, such as unmanned aerial vehicle (UAV) and tactile internet, low-latency and high-reliability are the main prerogatives. Finally, in mMTC systems, like 3D video streams, gigabyte internet, wide capacity and high throughput are crucial. The IoT, unmanned aerial vehicle (UAV), vehicle-to-vehicle (V2V), and machine-to-machine (M2M) communication are examples of applications that can benefit from the improved performances offered by 5G communication, allowing to wriggle out from proprietary interfaces featuring most of the systems above [3]-[7].

Furthermore, low power consumption and cost minimization are two key requirements for designing the 5G network nodes. The development and design of cloud radio access networks (C-RAN) enables high energy efficiency and improved resource allocation, employing a single baseband processing unit (BBU) serving several remote radio heads (RRHs) [8], [9]. However, certain reconfigurability is required for the hardware infrastructure to support multiple services, waveforms, access methods and easily update/add the network functionalities. In this context, the field programmable gate array (FPGA) devices represent the ideal platform for obtaining the aforementioned goals. Thanks to their intrinsic capability to parallel and high-speed processing and their flexibility both in the design phase and during the task execution, using the dynamic partial reconfiguration (DPR), the FPGA devices allow network hardware high performances and reconfigurability. Therefore, this approach enables the simple addition of new functionalities and operating modalities related to the reconfigurable sections, leaving unchanged the static sections [10]. Furthermore, the functional density of implemented computing applications can be improved by the runtime reconfiguration (RTR) techniques and compared through suitable methods [11]. Also, the FPGA platforms offer optimal performances in terms of development time, power efficiency, and costs.

In the present review paper, an overview of main FPGA devices' applications in the 5G networks/systems is presented, exploiting their flexibility and advanced performances above described. In particular, different FPGA implementations of 5G building blocks, such as C-RAN controller accelerator [12], network slicer using network function virtualization (NFV) [13], Cognitive Radio systems [14] and characterizers for multiple input multiple output (MIMO) channels [15], are reported in the scientific literature, as well as integrated modulators, and encoders/decoders, all benefiting of the improved performances guaranteed by the FPGA platforms. Afterwards, innovative architectures of encryption/decryption algorithms designed by employing the Xilinx Zynq Ultrascale+ MPSoC ZCU102 platform are discussed. Furthermore, a high-speed and lightweight implementation of the AES-128 algorithm is introduced, properly developed for a high-frequency short-range communication apparatus, named "wireless connector" [16]. The Xilinx ZCU102 FPGA platform constitutes the main section of the "wireless connector" system, managing all the baseband tasks, including the encryption/decryption of the data packets [17]. The proposed AES-128 implementation relies on a pipelined framework, enabling for each clock period the contemporary elaboration of different rounds on successive data packets, resulting in higher data throughput. In particular, the designed encryption solution can operate up to a 220 MHz clock frequency, obtaining time durations to perform data encryption and decryption both equal to just 10 clock periods; also, a data throughput greater than 28 Gbit/s is supported, thanks to the employed pipelined approach along with the fast solutions for implementing the Substitute Bytes operation. Finally, an in-depth overview of high-speed data processing applications, such as image and signal processing, visual recognition, and hardware resource management, based on Xilinx ZCU102 platform, is presented.

Therefore, the paper is arranged as follows: in section 2 are presented the main FPGA devices' applications in 5G networks/systems, whereas, in section 3, the implementations of encryption/decryption algorithms by Xilinx Zynq Ultrascale+ MPSoC ZCU102 platform are described in the sub-section 3.1; the comparison of our high-speed and resource-efficient AES-128 encryption/decryption system [16] with other similar implementations is introduced in the sub-section 3.2. Finally, an overview of the main applications of FPGA devices, based on the Xilinx ZCU102 platform, for high-speed processing is reported in section 4.

2. APPLICATIONS OF FPGA PLATFORMS IN 5G COMMUNICATION SYSTEMS

Due to their great flexibility and wide applicability, FPGA platforms are used in various application fields, such as video and imaging processing, military applications, automotive, electronics for specific processing, and more. These devices are commonly used for prototyping application-specific integrated circuits (ASICs) or processors. In this section, the literature analysis concerning the FPGA implementations is summarised, covering two major topics: high-throughput data processing applications and encrypting/decrypting algorithms, lingering on convenient designs and efficient architectures, and novel ideas.

In this paragraph, the main applications of FPGA-based platforms in 5G communication systems are investigated, in which high-throughput, low latency and high-security constraints are required for supporting new services, such as internet of everything (IoE), providing Internet connection to a myriad of devices [18]. These requirements are achievable by employing a robust and reliable hardware infrastructure. In this context, the FPGA represents an optimal solution for guaranteeing reconfigurability, high bandwidth, energy efficiency and enabling intrinsically parallel architectures to allow multiple computation kernels. The FPGA reconfigurability allows dynamically tuning the transmission capacity by changing the implemented services according to the capacity requirements. Therefore, the main FPGA producers are proposing on the market new FPGA-based platforms, with software programmability, multi-band functionality, and multi-standard hardware optimization along with hardware-level security. Several building blocks of a 5G infrastructure can be implemented using an FPGA-based platform, improving their overall performances. Specifically, efficient FPGA-based implementations of accelerators for cloud radio access network (C-RAN) controllers and network slicer using network function virtualization (NFV) are proposed in the literature, thus promoting the resource sharing and extending the functionalities of radio access network (RAN) [19]. Since FPGA can change its operative modalities in real-time, this allows us to share resources in C-RAN. In particular, in [20], the authors proposed a new FPGA-based framework to increase the performance of the data plane of the Edge-to-Core segment in the 5G virtualized and softwarized networks, providing a new design approach of 5G-aware traffic processing. This method has been implemented on an FPGA platform to test the prototype and evaluate its performances, so demonstrating its superior performances. Specifically, empirical tests have shown that the developed approach and prototype provides significant improvements in network congestion conditions, reducing the packet loss ratio from 49% to 4% in the worst scenario. Also, the FPGA-based prototype allows to obtain good scalability in the number of rules, as well as a minimum processing delay (i.e., 4 μs), thus complying with the latency required for 5G applications.

Besides, in [21], the authors presented a reconfigurable architecture of a universal-filtered multicarrier (UFMC) system for 5G communications to select the number of sub-carriers in a sub-band, leaving unchanged the hardware resources, and the used filter among many available by single selection line. A commercial Virtex 5 FPGA platform was used to test and analyze the baseband signal. The carried-out tests have revealed that the quantization error of the reconfigurable UFMC system was comparable with the error bound (i.e., 3.051×10^{-5}). Ferreira *et al.* introduced an FPGA-based baseband modulator suitable for non-contiguous carrier aggregation, C-RAN elaboration, and 4G/5G waveform coexistence [22]. The proposed design employs virtualization techniques for expanding the exploitation of the resources beyond the available ones for the Zynq XC7Z020; also, by applying the dynamic frequency scaling (DFS) technique, significant improvements in processing latency (-18.4% compared to those reported to [23]) and power consumption (up to 88% less) have been obtained.

Concerning FPGA employment in coding operations, in [24], the authors proposed quasi-cyclic low-density parity-check (LDPC) codes that rely on a base matrix comprising several sub-matrices, including the cyclically right-shifted identity one and the null one. The Vivado HLS (High-level synthesis) tool has been used to synthesize the proposed coding scheme on the Xilinx Kintex-7 FPGA platform. In order to test the coding scheme, an IEEE 802.11n compliant decoder has been implemented on the same FPGA chip using a highly-parallel framework to achieve higher throughput (i.e., 608 Mb/s). The experimental results revealed that the supported throughput is higher by four compared with the corresponding CPU-based implementation.

Besides, in the scientific literature, several software-defined wireless communication systems operating on 5G mmWave are proposed [25], [26]. Specifically, Riberio *et al.* developed a software-defined FPGA implementation of the orthogonal frequency division multiplexing (OFDM) transceiver for the 5G physical layer [27]. Some FIFO registers are placed between the functional blocks to mitigate critical path and high throughput for complex designs. A COTS Xilinx ML605 platform equipped with an AD-FMCOMMS1-EBZ RF board is employed to implement the system, covering a frequency range from 400 MHz up to 4 GHz. The realized system prototype was featured by scalable bandwidth (20-61.44 MHz), reaching a 500 Mb/s throughput value by 256-QAM (Quadrature amplitude modulation) modulation scheme. Also, in [28], the authors presented the orchestration and reconfiguration control architecture (ORCA) project, aiming to develop an SDR FPGA-based architecture based on low latency and high-performance accelerators, implementing multiple virtualized and concurrent instances. A demonstrator was realized using two WiFi, eight Zigbee, and one BLE instances, using an FPGA platform (namely, the Xilinx ZC706 evaluation board) over a 40 MHz bandwidth. The developed architecture demonstrated rapid development time, as well as, the software-oriented approach enables us to create and handle multiple radios. Similarly, Duarte *et al.*, in [29], presented an end-to-end FPGA-based communication system; it includes a baseband and low-IF (intermediate frequency) processing unit, allowing on-the-fly modulation (up to 256-QAM), and a high-frequency RF front-end, both at the receiving and transmitting sides. Besides, an OFDM scheme was employed, obtaining 2% error vector magnitude (EVM) lower than the current threshold for the 256-QAM. Also, in [30], an FPGA-based wireless communication system was proposed, equipped with Bluetooth connectivity. The system includes an AES encryption block for making secure the data transfer between two radio stations; a prototype of the developed communication system was realized employing the RC10 FPGA development board. Comparing the developed framework with other ones presented in the literature, the first has demonstrated a higher data rate and lower power consumption.

As previously discussed, the NFV is a crucial element for developing the 5G networks; Pinneterre *et al.* developed a new FPGA-based virtualization approach, called vFPGAmanager, that enables orchestrated acceleration of resource allocation for virtual machines, uni-kernel, and containers [31]. The proposed framework enables dynamic remote orchestration using a set of commands associated with the FPGA accelerators and virtual machine status through an innovative communication mechanism's interface. The experimental results demonstrated that the controller could serve the incoming commands from the orchestration with negligible overhead (in the worst case, it employed 18079 µs for performing 1000 commands).

Furthermore, FPGA can be exploited for characterizing massive multiple input multiple output (MIMO) channels for both the local data elaboration and the standardization of the technology [32]-[35]. In this context, Huang *et al.* discussed the development of an efficient geometry-based complex MIMO channel emulator, using an interactive-based method [36]. Besides, they analyzed and studied the trade-off featuring the emulator performances between resource utilization and channel accuracy. The proposed emulation approach employs an iterative framework for generating the geometry-based channels and optimizing the word length and refresh rate to reduce the FPGA's memory and hardware utilization. Furthermore, the experimental results indicated that the developed emulator could be implemented using an FPGA device (Virtex 4 VFX100), allowing to process up to 19 TX/RX antennas pairs in real-time. Also, in [37], the authors presented a 2X2 MIMO Generalized frequency division multiplexing (GFDM) transceiver, using National Instruments USRP-RIO platforms, using a Xilinx Kintex-7 FPGA, combined with Labview software, so obtaining a communication chain deployed both in hardware and software. A prototype of the developed transceiver, operating on 1.2-6 GHz frequency interval with 40 MHz bandwidth, allowed to determine the performances of GFDM, namely low latency, out-of-band (OOB) emission, and high reliability. In particular, experimental tests demonstrate that -48dBm OOB radiation has been obtained due to pulse-shape filtering applied to the sub-carrier, enabling to apply the GFDM in a highly fragmented spectrum scenario or cognitive radios framework [38]-[40]. Ayouby *et al.* introduced a novel combining framework for the optimal generalized diversity receiver for 5G MIMO channels, called generalized maximum ratio combining (GMRC), each represented by a single input multiple output (SIMO) and binary phase shift keying-spatially modulated (BPSK-SM) channel, obtained from a proper combination of diversity channels [41]. This work, derived from a previous implementation [42], substituted all operations into addition and multiplication, more efficiently hardware implemented by an FPGA platform. Furthermore, a pipelined framework was used in the proposed solution, resulting in a higher throughput value. The FPGA-based solution of the proposed combination scheme was efficient in terms of resource utilization and speed, reaching an operative frequency higher than 180 MHz due to the above-described solutions. Given the superiority of the GMRC scheme with respect to the MRC one and the efficient FPGA implementation, the GMRC solution is suitable for future 4G or 5G wireless MIMO receivers [43].

Finally, in this section, different scientific works involving FPGA devices in communication applications, and, mainly in 5G networks, have been analyzed and discussed. All the applications exploit the reconfigurability, high operational frequency and attitude to parallel computing offered by the FPGA platforms. Table 1 summarizes the scientific works discussed above, highlighting the application typology, employed FPGA platform, and benefits of FPGA implementation as shown in Appendix.

3. IMPLEMENTATION OF ENCRYPTING/DECRYPTING ALGORITHMS WITH XILINX ZYNQ ULTRASCALE+ MPSoC ZCU102 PLATFORM

In this section, the state-of-art of innovative cypher/decipherer implementations is reported, all hinged on the Xilinx Zynq Ultrascale+ MPSoC ZCU102 FPGA platform described in detail in the next sub-section 3.1. Afterwards, in sub-section 3.2, we presented our novel architecture of the AES-128 algorithm, featured by high data throughput (up to 28.16 Gbit/s) and low FPGA utilization of hardware resource (only 3262 slices), and comparing it with other similar solutions reported in the literature.

In [44], the authors introduced a multi-processor architecture to implement the computationally expensive fan-vercauteren (FV) homomorphic encryption scheme by employing both an FPGA and an ARM-based processor for carrying out several homomorphic processes in the cloud. They used the Halevi, Polyakov, and Shoup optimization techniques for reducing the computational load due to parallel polynomial multiplication algorithm with high-precision arithmetic. Specifically, a Xilinx Zynq UltraScale+ MPSoC ZCU102 board has been deployed for developing and testing the proposed programmable architecture.

Through parallel computation cores and block-level pipeline methods, a 200 MHz operative frequency was reached, thus achieving a data rate of 400 homomorphic multiplications per second, thirteen times faster than the heaviest optimized software implementation on the Intel i5 processor.

A dataset partitioning solution based on the FPGA platform has been presented in [45], to parallel execute software programs. To benchmark this architecture, the authors have used four different applications: a 256-bit AES encryption algorithm, a hotspot application for thermal simulation, a NBody application to measure the particle behaviours under the influence of a force, and a general matrix multiplication (GEMM) application multiplying two 1024x1024 floating-point values. The dataset was distributed into fixed-length blocks and, at each iteration, several data chunks were entrusted to a computing unit, and only that subset can be loaded to the unit. Filter classes and pipeline derived by Intel Threading Building Blocks (Intel TBB) were employed to implement parallelism and a two-stage pipeline. A serial filter constitutes the first stage for calculating the data chunk size and allocates it to the following idle computing unit. The second stage was a parallel filter to simultaneously communicate and process the data chunks in all the computing units. The obtained results demonstrated that the developed architecture could provide up to 86.23% of the throughput achievable for the AES application, up to 82.50% for hotspot application, up to 94.06% for GEMM application, and up to 111.51% for NBody application.

In [46], the authors presented an AES-GCM architecture with efficient utilization of digital signal processor (DSP) slices and block random access memory (BRAM) tiles, implementing Drimer’s round-based architecture, for performing both AES and the multiplication in 10 cycles. Furthermore, a fully unrolled pipelined architecture, employing the AES Tbox approach, was reported to carry out the AES encryption and the multiplication in 1 cycle and append the GCM mode of operation with optimized GF (2^{128}) (Galois field) multiplier. In the round-based architecture, the sequence of SubBytes and Shift-Rows steps were exchanged, and the 128-bit blocks of data were split into 1-byte chunks, applied to the Shift-rows function. To derive the Sbox outputs and multiplied version of SBox outputs, each byte was provided to its corresponding Tbox. Each column was constituted by four Tbox adjacent outcomes, combined to obtain the corresponding MixColumns output. In the last round, all the Tboxes generated the Sbox output and the MixColums function was skipped. In the unrolled pipelined architecture, the operation flow was the same as the round-based, with all the rounds implemented in an unrolled pipelined modality for faster execution with more area utilization. The results demonstrated that the round-based architecture uses 899 LUT, 1036 FF, 139 BRAM, and 685 DSP. In contrast, the unrolled pipelined architecture employs 785 LUT, 1043 FF, 17.5 BRAM, and 72 DSP, resulting in fewer resources usage.

Kim *et al.* have proposed the SafeDB (spark acceleration on FPGA clouds with Enclaved data processing and Bitstream protection), a complete and systematic security framework for the confidential bitstream data provided by the on-cloud applications [47]. The framework employed a 256-bit AES algorithm for ensuring the bitstream security and two asymmetric key-based security systems, namely the Public-Key Infrastructure (PKI) and elliptic-curve cryptography (ECC). The first one was used for sharing the authentication key between the FPGA and the user, generated in hard-wired logic using PKI and ECC. The AES key protection was guaranteed for each device by a physically unclonable function (PUF) - based scheme, where the private key, employed for the bitstream decryption, was derived and stored in the FPGA non-volatile memory, whereas the public shared key was generated using the private key. In Figure 1, the data and bitstream flow from a client to cloud service provider (CSP) was depicted; this last processed the data and executed the client’s application, constituted by two sections, namely the house-keeping and kernel codes. The first one was a software application executed on the CSP, whereas the kernel code elaborated the user data employing the Xilinx CAD tool carried out in the FPGA on CSP. The FPGA locally decrypted the incoming encrypted (using the AES decryption module) data from the bitstream, executed the kernel code, and encrypted (using the AES encryption) again the outcoming data. A management tool, named FPGA-as-a-Service (FaaS), carried out the initial protection configuration (such as passing metadata) of the end-to-end communication between FPGA and CSP. The performances evaluation and the hardware utilization were estimated using three benchmark applications, namely word count (WC), sobel filter (SF), and logistic regression (LR). Based on the results provided by the last, with 64 GByte computational load, a performance improvement of up to 1.36x was obtained compared to the baseline; instead, for the SF application with 192 GB workload, just a 1.12x improvement in execution time was demonstrated.

3.1. Zynq Ultrascale+ MPSoC ZCU102 board: overview

The Xilinx Zynq Ultrascale+MPSoC ZCU102 board enables the quick prototyping of industrial, automotive, communications, and video applications (Figure 2). It relies on the Zynq Ultrascale+ XCZU9EG-2FFVB1156E multiprocessor system-on-chip (MPSoC), which combines a powerful processing system (PS) and the efficient programmable logic (PL) section within a single package. The PS section includes three main processing units:

- Application processing unit (APU) based on the quad-core ARM Cortex-A53.
- Real-time processing unit (RPU) based on the dual-core ARM Cortex-R5.
- Mali-400 graphics processing unit (GPU).

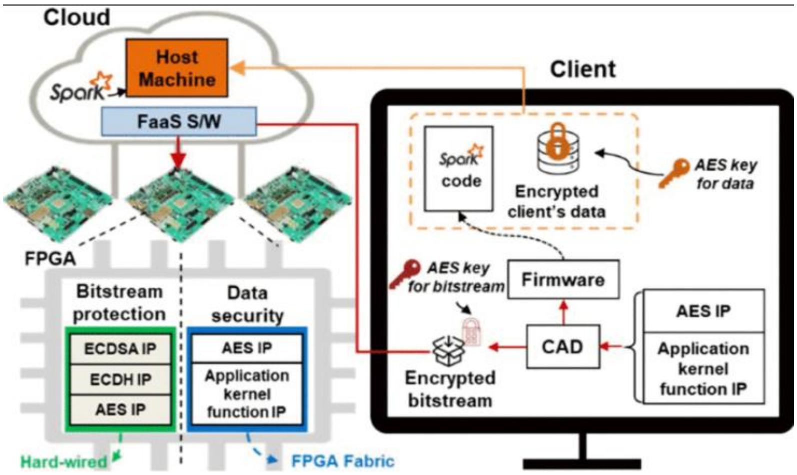

Figure 1. Graphical representation of the SafeDB security framework presented in [47]

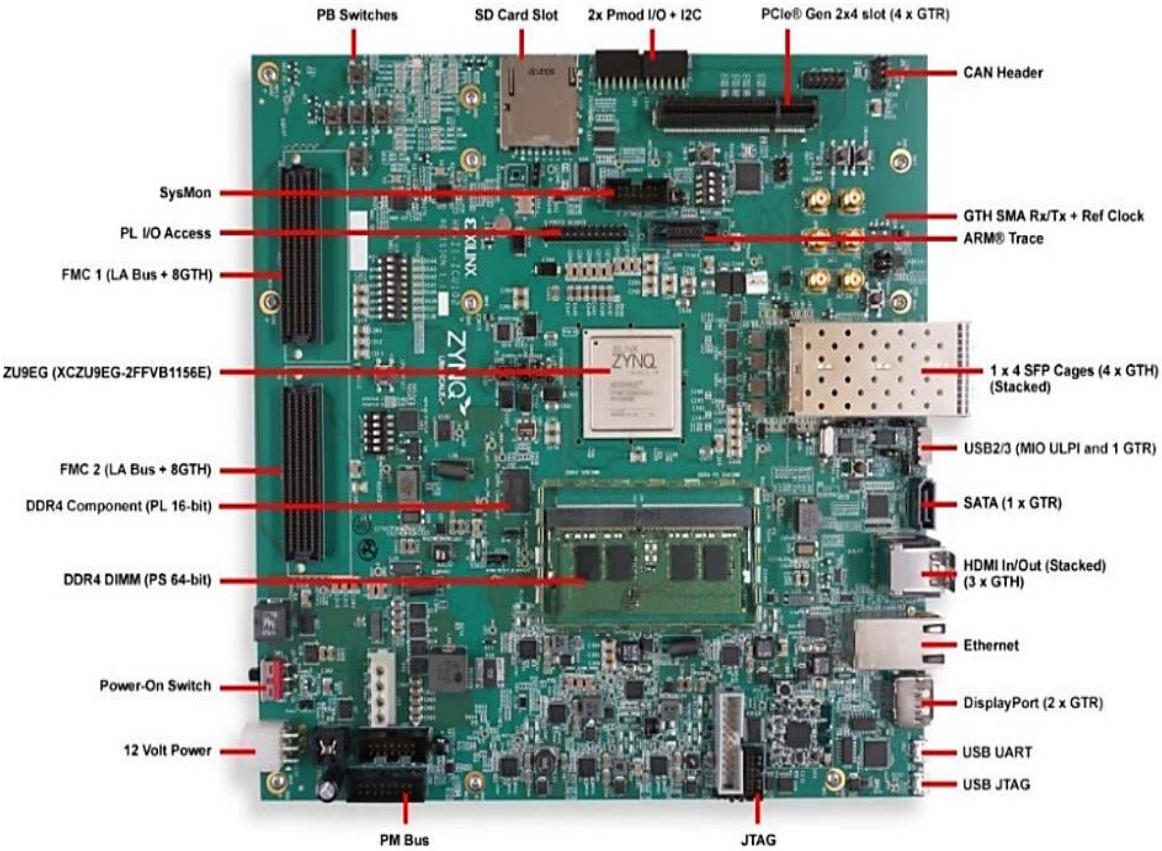

Figure 2. Zynq Ultrascale+MPSoC ZCU102 board with highlighted the main components

The PS-side contains numerous peripherals, made available through three Multi-use I/O (MIO) banks, including 78 pins. Also, on the PL-side, there are five high-density (HD) banks for a total of 120 pins and four high-performance (HP) banks corresponding to 208 pins. The board also includes 4GB DDR3 (Double Data Rate), small outline dual in-line memory module (SO-DIMM) memory for the PS needs and 512 MB DDR4 memory for the PL-section ones. The ZCU102 board offers a PCIe (Peripheral Component Interconnect Express) slot, two mezzanine card interfaces for hardware expansion, universal serial bus (USB), high-definition multimedia interface (HDMI) interfaces, and RJ45 ports (Registered Jack type 45) for the ethernet connection.

The ZCU102 board provides several clock oscillators using the SI5341B clock generator (PS reference clock), the SI570 (manufactured by silicon labs, PL reference clock) I^{2}C (Inter-Integrated Circuit) programmable oscillator (300 MHz default), and another SI570 programmable oscillator (156.2 MHz default). Furthermore, the board includes the SI5324 clock generator for the HDMI clock recovery and variable clock oscillators for the XCZU9EG MPSoC. The board has SMA connectors, named J79 (P-side) and J80 (N-side), to provide external clocks for the GTH transceivers. Also, the ZCU102 platform is equipped with an MSP430 microcontroller that communicates with the onboard programmable devices by the I^{2}C interface. The user interface for system control, provided by Xilinx, enables us to check and manage the board's programmable features like the clocks, FMC (FPGA Mezzanine Card) functionalities power systems, and the PS-Side GTR (Gigabit Trans-Receiver) transceiver selection.

The embedded processor is a 64-bit quad-core ARM Cortex-A53, based on ARMv8-A architecture, operating with a 1.5 GHz clock frequency and 64-bit or 32-bit operative modalities. Each Cortex-A53 core is equipped with a separate 32 KB L1 instruction and data cache memories and a shared 1 MB L2 cache memory. The real-time processor is a 32-bit dual-core ARM Cortex-R5 (each equipped with the same previous chance memory and a 128 KB Tightly Coupled Memory-TCM), based on the ARMv7-R architecture, and reaching a 600 MHz maximum clock frequency. The graphics processor is an ARM Mali-400, supporting a single geometry processor and a two-pixel type one, supporting 667 MHz clock frequency; also, it has 64 KB L2 cache read-only memory and 4x/16x anti-aliasing support. The platform management unit (PMU), a dedicated user-programmable processor, monitors the board's power usage, error management, and system initialization before the booting stage; it employs a battery power mode to maintain security configuration and an RTC (real-time clock) also when the board is powered off. The PMU is equipped with a read-only memory containing a set of instructions such as the startup sequence, interrupts, and power-up/power-down requests; also, it stores the system-power state at all times PS-level and propagation logic errors.

There are two memory typologies available in Zynq Ultrascale+ systems, namely a 256 KB RAM on-chip memory (OCM) and off-chip DDR memories. The first stage boot loader (FSBL) is loaded by the OCM from the boot device; after loading the FSBL in the OCM, either APU or RPU processor executes it. The PS is internally divided into three power regions, isolated from each other by the PMU, allowing functional isolation between regions; each power region can be associated with a power mode, where only some of the components are active. In full-power mode, all sections are fully operating, thus the most energetically expensive mode; in low-power mode, the active components include the RPU, OCM, TCMs, and all peripherals except serial advanced technology attachment (SATA) and DisplayPort. The battery-power modality is featured by the lowest power consumption and includes a battery-backed RAM (BBRAM) to store the encryption key and an RTC supported by an external clock generator to keep time also whatever the system is turned off. The Zynq UltraScale+ MPSoC PS has four high-speed serials I/O (HSSIO) interfaces that support the protocols: PCI Express®; SATA 3; DisplayPort interface (with video resolution up to 4K x 2K-30 (30 Hz pixel rate); USB 3.0; Serial GMII (Gigabit Media Independent Interface).

The input-output processor (IOP) peripherals are interfaced with the external devices through a shared bank of up to 78 dedicated multiplexed I/O pins. Each peripheral can be mapped on multiple devices concurrently by using pre-set pin groups. The PL section can access most IOP interface signals; if the 78 pins are not enough, standard PL I/O pins can be used. Furthermore, extended multiplexed I/O (EMIO) allows unmapped PS peripherals to access the PL I/O pins to extend the interfacing capability of the MPSoC. The ZU9EG uses a 16nm FinFET technology. The PL section includes 208 HP and 120 HD I/O pins, 24 GTH 16.3 Gb/s high-speed transceivers, and a monitoring system for detecting the chip temperature as well as internal voltages and currents. It also contains new high-performance peripheral interfaces, such as 1G ethernet and four Gen2 PCIe interfaces.

The overall FPGA resources are organized, according to the clock management, in a column-and-grid layout; some of the PL resources are dedicated to the processing system for implementing transceivers, memory interface logic, clocking circuit, and I/O interfaces. Other blocks included inside the PS, such as the PCIe interface, configuration logic, and monitoring system, are integrated into the SoC. Generally, FPGAs have dedicated clock routes, known as clock regions, distributed in a chip region. For the UltraScale+ architecture, the clock regions are 60 CLBs height, corresponding to a bank of 52 I/O interfaces, 24 DSP slices, 12 block RAMs, or four transceiver channels [48]. The clock region width, in terms of CLBs number, affects timing repeatability inside them, regardless of the availability of device resources and their distribution. The clock region includes vertical and horizontal clock routing to distribute the clock signal within a region; these horizontal and vertical clock routes can be divided at the clock region edge to obtain a flexible, high-performance, low-power clock distribution architecture.

A multi-layered ARM advanced microprocessor bus architecture (AMBA), Advanced eXtensible Interface (AXI) bus connects the MPSoC blocks and the PL portion, allowing multiple simultaneous master-slave transactions. The AXI bus is designed with the shortest paths to connect the memory blocks and support high throughput connections to the slave blocks. The CPU, direct memory access (DMA) controller, and a combined entity representing the masters in the IOP generates the AMBA AXI bus data, supervised by the interconnection's quality of service (QoS) block.

3.2. Comparison of proposed AES-128 encryption/decryption algorithm with other works presented in the scientific literature

As above described, in [16], we proposed a high-speed and resource-efficient implementation of the well known AES-128 encryption/decryption algorithm, developed for a custom high-frequency (around 60 GHz), short-range (1-10 m) communication system, named "wireless connector". A Xilinx ZCU102 development board has been employed as the core section of the developed communication apparatus, implementing all the baseband tasks, such as modulation/demodulation, coding/decoding, and encryption/decryption to ensure communication security. Particularly, the proposed AES-128 encryption/decryption system employs a pipelined approach, enabling concurrent processing of multiple data packets each clock cycle within the 10-rounds elaboration, distinguishing the AES-128 cypher. Also, together with the development of the fast implementation of Sub Bytes operation through a 32-bit 16x16 Sbox matrix, the processing time of each AES-128 round is reduced to only one clock cycle. The behavioural and post-implementation simulations, along with onfield tests carried out after the board programming, demonstrated that a 220 MHz maximum clock frequency is sustained by both the cypher and decipherer; also, only ten clock periods are needed to provide encrypted and plaintext data packets, respectively, and loading new data packets every clock cycle, thus resulting in data throughput over 28 Gbit/s ($i.e. 128\frac{bit}{packet} \times 220\ MHz = 28.16\ GHz$). Also, a rapid key expansion algorithm has been developed, combining, by combinatorial operators, the current sub-key with those at the previous step treated with the Sbox, thus obtaining the 44 sub-keys involved in the ten rounds of AES-128 algorithm in only 174.55 ns.

The proposed AES-128 cypher and combined encryption/decryption system are compared with different high-speed pipelined implementations presented in the scientific literature, supporting similar operating frequency and throughput (Table 2). Besides, the FPGA device employed by each considered work is indicated because it affects its performances. To compare the different solutions, the efficiency has been chosen as a merit figure since it considers the obtained throughput jointly and used hardware resources.

$$Efficiency = \frac{Data\ Throughput\ [Mbps]}{n°\ of\ used\ slices}\ [\frac{Mbps}{slice}] \tag{1}$$

Table 2. Performances comparison of proposed AES-128 pipelined solution with other implementations proposed in the literature

Design	Platform	Frequency [MHz]	Throughput [Gbit/s]	Slices	Efficiency [Mbps/Slice]
Wang *et al.* [49]	XC6VLX240T	319.3	40.87	5927	6.90
Good *et al.* (Enc/Dec) [50]	XC3S2000-5	196.1	23.65	16693	1.42
Daoud *et al.* (Enc) [51]	XC7Z020	192.0	1.29	431	2.99
Granado-Criado *et al.* (Enc) [52]	XC2V6000-6	194.7	24.92	3576	6.97
Zhang *et al.* (Enc/Dec) [53]	XCV1000E-8	168.4	21.56	11022	1.95
Henzen *et al.* (Enc) [54]	XC5VSX240T	233.00	119.30	14799	8.06
Our solution (Enc) [16]	XCZU9EG	220.0	28.16	3262	8.63
Our solution (Enc/Dec) [16]	XCZU9EG	220	28.16	10278	2.74

From the results in Table 2, it is evident that the AES-128 implementation proposed in [16] supports a relatively high data throughput (up to 28.16 Gbit/s) but uses fewer hardware resources compared to other similar works reported in the literature; therefore, it obtains a higher value of efficiency. By comparing the proposed solution with those reported in [52], the first supports a higher data throughput (i.e. +13.0%) but uses fewer FPGA slices (i.e. -8.78%), thus obtaining a higher efficiency (i.e. +23.81%). Considering the combined encryption and decryption system and comparing it with the solution proposed in [53], the first supports a higher data throughput (i.e. +30.6%), but also requires a lower FPGA hardware resources utilization (i.e. -6.7%), thus resulting in higher efficiency (i.e. +40.5%).

4. HIGH-THROUGHPUT DATA PROCESSING APPLICATIONS USING XILINX ZYNQ ULTRASCALE+ MPSoC ZCU102

This section explores the applications of the Xilinx Zynq Ultrascale+ MPSoC ZCU102 FPGA platform for high-speed data processing in several fields, such as image and signal processing, visual recognition, and hardware resource management. In [55], the authors proposed a novel architecture for parallel multi-view high-efficiency video coding (HEVC) decoder, using the Xilinx Zynq UltraScale+ MPSoC ZCU102 board as a hardware accelerator for complex operations. The proposed method can optimally decompress in real-time 3 videos with 1920 x 1080 pixels resolution using the low power processor. To improve the compression efficiency, the motion prediction between two consecutive frames is needed, so increasing the computational load and waiting time of the inter-view parallel implementation. Therefore, the authors developed a similar method that relies on the decompression order variation, implementing an inter-frame parallel approach with no data dependency between the frames. The frame dependence was defined based on the waiting time and frame frequency, allowing simultaneous elaborations of independent frames. The authors demonstrated that the MV-HEVC algorithm provided a throughput eleven times higher than the 3D-HTM16 software and a real-time decompression of a 388p 3-views video.

Huang *et al.* introduced an expandable FPGA-based digital pre-distortion (DPD) system, guaranteeing linear processing in 5G mm-wave transceivers transmitting wideband modulated signals [56]. The DPD engine architecture, implemented into FPGA, allows processing multiple samples per each clock cycle, operating at a clock rate of 300 MHz, achieving a scalable linearisation bandwidth up to 2.4 GHz. An undersampling transmitter observation receiver (TOR) was developed to dynamically update the DPD coefficients and capture the power amplifier (PA) distortion. The TOR is equipped with a single analog to digital converter (ADC) and an apposite training algorithm to update the DPD coefficients to change PA non-linearity. The authors demonstrated that the DPD engine using envelope complexity-reduced volterra-series (ECRV) for linearising 5G mm-wave wideband OFDM signals, providing performances comparable to CRV methods but consuming less power and occupying fewer hardware resources.

A 3D convolutional neural networks (CNNs) accelerator, suitable for embedded systems, is described in [57]. The accelerator employs a pipelined architecture and implements parallel computations using several multiply-and-accumulate (MAC) units to accelerate the inference task in CNNs; each vector performs simultaneously three vector convolutions. Furthermore, it carried out 3D convolution of feature maps, up to 256x256 pixels, and 64 AXI channels, working with a kernel size of 3 and 8-bit or 16-bit fixed-point logic. The optimization was realized using pragma directives and SDSoC functions. The proposed implementation stood out for low power consumption and excellent performances, reaching 32.08 GOP/s and an efficiency of 3.58 GOPs/W.

Véstias *et al.* proposed an optimized and scalable architecture that improves inference execution times of CNNs, by using static and dynamic zero-skipping and weight pruning and by applying an 8-bit fixed-point representation based on FPGAs [58]. The architecture consists of two separate sections; the first one dedicated to convolutional layers and another to fully connected layers, allowing them to apply different optimization techniques independently. In the convolutional layers, the complete feature map is stored; then, the data from several blocks are loaded, including their weight coefficients, which are multiplied with the weights of kernels, avoiding zero activations for calculation. Kernels and memory of activations are partitioned and stored in separate memories allowing parallel reading. The convolutional layer module reads eight activations per clock cycle, but only one non-zero activation per cycle is sent. Furthermore, in the convolutional layers, to reduce the dispersion of weights in the kernels and the overhead due to the index information of the sparse weights' vector, they adopt the block pruning method. The technique prunes weights block instead of single weights, reducing the overhead data and enabling parallel MACs efficiently. The fastest solution was the architecture with the implementation of zero-skipping, static and dynamic pruning, and dual-rate memories, which shows 464 GOP/s and 216 GOPs/W in the Zynq XC7Z020, 1344 GOP/s, and 145 GOps/W in the Zynq XC7Z045.

In [59], the authors proposed hardware implementation of the you only look once (YOLO) object detector based on a mixed-precision CNN. The authors used a half-precision (16-bit) CNN in parallel for both the classification and the localization, as well as the binary (1-bit) precision CNN for the feature extraction. A half-precision weight cache is used for the former convolutional circuit, whereas the binary one is used for the 2D convolutional binarised neural network. All weights are stored on the off-chip DDR memory. The authors trained the mixed-precision YOLOv2 with their designed training system by using chainer deep learning framework. The results showed that the proposed framework was featured by an 85.2% recognition accuracy, 35.71 Frames Per Second (FPS), and power usage of 4.5 W, thus ensuring an efficiency performance per power unit of 7.93 FPS/W.

Janus *et al.* presented two hardware implementations of the Gaussian mixture model (GMM) algorithm for foreground object segmentation and background modelling, capable of processing real-time video streams with resolution up to 3840x2160 pixels and 60 FPS [60]. The first approach provides each pixel with a dedicated background model and processes grayscale images. In contrast, the second gives each pixel a dedicated background model per clock cycle and processes red-green-blue (RGB) colour model images along with the grayscale ones. The background model is read via the AXI memory controller, and Gaussian distributions are sorted using a simple bubble sort algorithm in parallel to colour-space conversion. The authors employed a lossless compression algorithm for reducing the memory bandwidth (i.e., the RAM access time) for hardware implementation by reducing the background model's size. The whole system used 22 W and obtained 32.8 GOP/s for the first implementation and 20.7 GOP/s for the latter.

In [61], the authors proposed two optimization methods based on a virtual-channel (VC) router, one consisting of scoring crossbar arbitration and the other of arbitration interception. The first method processes the priority and round-robin arbitrations in parallel, assigning a score to the packets based on both their priority and current round-robin factor as weight. The packets with the highest score are transmitted to the next router (Figure 3). To prevent lower-priority packets from waiting in the virtual channels, the authors proposed an arbitration interception to overcome transmission congestion and latency, improving overall performance. If the high-priority channel fails to pass the crossbar arbitration, an arbitration interception signal is sent to the VC allocator, disabling the request of the high-priority channel, guaranteeing to the lower priority one the crossbar arbitration. The proposed router used 2644 LUTs and 1189 FF and showed 0.645 W power consumption.

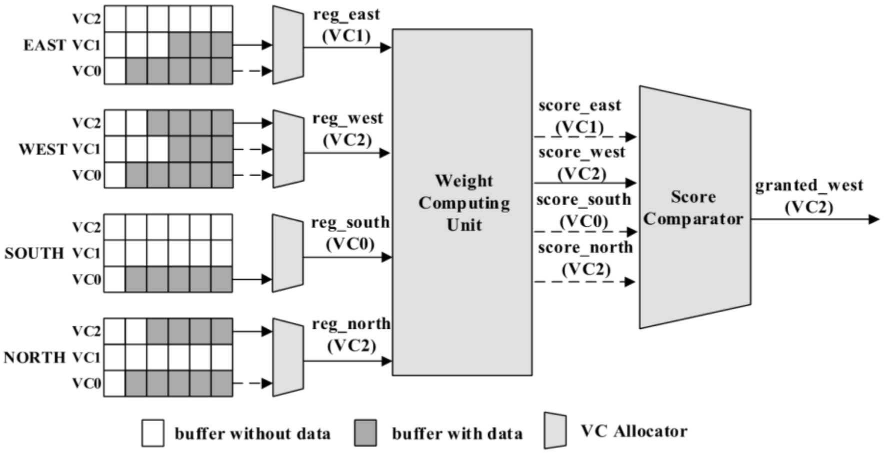

Figure 3. Scoring crossbar arbitration proposed in [61]

Shen *et al.* presented an FPGA-based service gateway user plane (SGW-U) system in mobile edge computing (MEC) aimed to 5G scenario [62]. The system includes a generic OpenFlow switch, and, for offload computational tasks, a programmable GPRS tunneling protocol (GTP) processor for GTP packets encapsulation and decapsulation. This last processor is programmed by programming protocol-independent packet processors (P4) code, consisting of two sub-systems, called the GTP Encap and GTP Decap. The OpenFlow switch manages the ethernet packets by forwarding them between the Backhaul and the edge servers. When they are transmitted from the Backhaul to the edge servers, the packets are offloaded by the switch to the Encap sub-system, where they are arranged as GTP packets by adding the correspondent overhead. Similarly, the packets are offloaded to the Decap sub-system when transmitted from edge servers to the Backhaul, and the GTP headers are removed. The system achieves a 10 Gb/s throughput for each port, with a 5 μs processing latency and a total of 40 Gb/s speed considering only four ports.

A non-volatile memory express over remote processor messaging (NVMe-over-RPMsg) software solution is described in [63], for emulating a remote storage system without requiring VMs. The proposed solution represents a high scalable framework for virtualizing remote storage systems such as NVMe solid state drive (SSD), overcoming the limitation of the NVMe SSD emulated method by quick emulator (QEMU). A guest operating system (OS) runs local applications, as wells as a remote OS, the independent storage management software, replacing the PCIe with RPMsg protocol to deliver messages between the two OSs. The remote core manages the communication between the guest OS and remote OS and represents the RPMsg endpoint as well as the front-end of NVMe-over-RPMsg. The remote OS implements the back-end, by emulating an NVMe SSD controller and processing NVMe commands received from the guest OS. The obtained results indicated a performance boost than the QEMU-NVMe, reducing the read/write latency by 45.4% and scaling up to 1.74x the read/write throughput.

In [64], the authors benchmarked computer algorithms on three graphical accelerators for image processing applications: the ARM57 CPU, ZCU102 FPGA, and Jetson TX2 GPU, equipped with proprietary libraries (OpenCV, xfOpenCV, and VisionWorks, respectively). The algorithms, according to their functionality, are classified into six categories. The *input processing* category includes the pre-processing methods, including arithmetic methods, to convert the incoming format or several channels into another format. The *image arithmetic* category includes standard arithmetical or logical image processing operations; the algorithms can be distributed in different processing units regardless of data dependencies. The image filters-type algorithms compute the correlation between an input image and a kernel; in the image analysis, analytic kernels extract the image's features, including the colour distribution, the maximum and minimum pixel value, etc. The matrix product is included in the geometric transformation category. Finally, the composite kernels category contains all the kernels included in the above-described categories. The tests demonstrated the GPUs' superiority for standard and easy-to-parallelize methods, achieving a reduction of energy/frame ratio of 1.1-3.2 times than CPU and FPGA implementations. On the other hand, the FPGA works better than the other hardware accelerators for complex kernels, reducing the energy/frame ratio of 1.2-2.3 times.

Xinkai *et al.* proposed an FPGA-based architecture that combines owning processing unit, enabling parallel and pipelined processing with buffering capability. The novel method splits the filters using a preliminary feature map before performing the Winograd approach, called Wino-transCONV. This algorithm eliminates the multiplication with 0 value and, then applies the transposed convolution, through the classic Winograd-transformed processing, to the splitting and remapping stage's outputs [65].

Finally, they presented a parallel-aware memory partition technique to coordinate parallel operations and to achieve efficient data access. The dataflow of the developed Wino-transCONV algorithm is graphically summarized in the following Figure 4; as evident, the splitting (S1) and remapping (S5) are added to the fast Winograd algorithm (S2-S4). The S1 processing divides into four sub-filters windows the K×K complete filter window; afterwards, each of them is transformed into the related sub-inFM window. The S2 processing performs matrix transformations for the input feature map as well as the filter, whereas for the output feature map, the S4 step carries out matrix transformations. The S3 processing carried out the element-wise multiplications manipulation (EWMM), whereas the S5 processing rearranges the m×m sized, intermediate output patterns, computed by the S4, into a 2m×2m sized outFM matrix. The proposed implementation reaches 639.2 GOP/s on the Xilinx ZCU102 board and 162.5 GOP/s on the VC706 FPGA-based platform, obtaining a 2.2-factor improvement of the performance and up to 11.7-factor on processing throughput compared to other works present in the literature. Table 3 summarizes the scientific works discussed above, classifying them in terms of application and benefits offered by the FPGA-based compared to traditional implementations [66], [67].

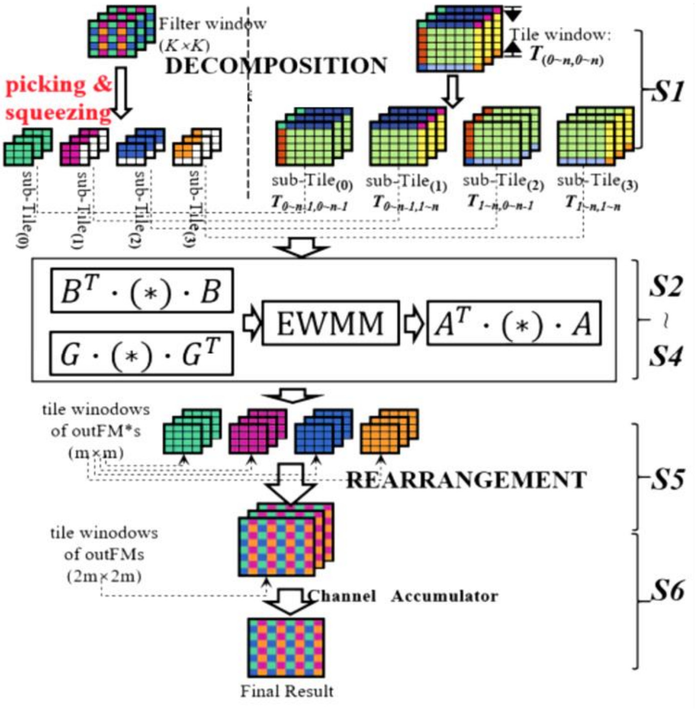

Figure 4. Wino-transCONV dataflow proposed in [65]

Table 3. Summarizing table of the scientific works discussed in section 4, related to image and signal processing, visual recognition, and hardware resource management using the ZCU102 platform

Work	Application	Benefits
Liu *et al.* [55]	Hardware Multi-View decoder	High parallel High throughput (11x faster compared to the software implementation)
Huang *et al.* [56]	Digital Pre-Distorsion system	Hardware-efficient High-bandwidth scalability
Ordoñez *et al.* [57]	3D CNN graphic accelerator	Reconfigurability Flexibility High throughput (32.08 GOP/S) High efficiency (3.58 GOP/s/W)
Véstias *et al.* [58]	CNN optimized with fixed-point representation, zero-skipping, and weight pruning operation	Reduced execution time (2.9 ms) High throughput (5.8X compared to traditional implementations) Low memory requirements
Nakahara *et al.* [59]	YOLOver2 object detector	Low recognition time (28 ms) Higher accuracy (mAP-mean average accuracy of 85.2%)
Janus *et al.* [60]	Gaussian mixture algorithm for video streams	Process 4K video in real-time High throughput (32.8 GOPS) High efficiency (6.98 GOPS/W) High accuracy (percentage of wrong classified pixels varies from 6% to 11%)
Guo *et al.* [61]	Virtual-Channel router	Reduced hardware resource (-10% LUT compared to a traditional router) Low packet latency High throughput (+165% compared to a traditional router)
Shen *et al.* [62]	Processing engine for packing and unpacking of GTP packet	High throughput (10 Gbps) Low latency (5 μs)
Zhang *et al.* [63]	NVMe for remote processing messaging (RPMsg)	Reduced latency (-45.4% compared to QEMU-NVMe system) High throughput (1.74x compared to QEMU-NVMe system)
Di *et al.* [65]	Accelerated Architecture for Wino-transCONV of GANs	High throughput (8.6x ÷11.7x compared to conventional Conv-baseline) High performance (2.2x compared to conventional Conv-baseline)

5. CONCLUSION

The FPGA-based platforms represent an ideal solution for high-speed processing due to their intrinsic capability to parallel elaboration as well as their flexibility both in the design phase and runtime step. Thanks to these advanced features, the FPGA devices are widely applied in 5G networks/systems for implementing critical tasks, such as accelerators for C-RAN controllers, network slicer employing NFV, and characterization of MIMO channels. In this review paper, we explored the applications of FPGA devices for high-speed processing; in particular, an overview of the main applications of the FPGA devices in the 5G networks/systems was presented, exploiting their flexibility and advanced performances. These features enable the network apparatuses to support different functionalities and operating modalities in runtime to dynamically respond to the communication requirements and improve resource and power efficiency. Thanks to hardware virtualization, the FPGA applications are opening new frontiers for developing the new generation of network device able to comply with the evolution of the communication architectures. Afterwards, an AES-128 cypher/decipher, implemented on the Xilinx Zynq Ultrascale+ MPSoC ZCU102 FPGA board, were presented, as well as, we introduced our efficient and performant AES-128 encryption/decryption system, suitably designed for a point-to-point, close-distance and high-throughput communication apparatus, named “wireless connector”. The encryption/decryption system is based on the ZCU102 FPGA development board, implementing a pipelined strategy, enabling the parallel elaboration, in each clock cycle, of multiple rounds on distinct consecutive data packets, obtaining a higher data rate. Specifically, the developed AES-128 cypher/decypher reaches 220 MHz clock frequency, spending ten clock cycle s for both encryption and decryption; a data rate greater than 28 Gbit/s is also achieved thank to the employed pipelined solution and rapid implementation of the Substitute Bytes step. Besides, our AES-128 encryption system is featured by efficient hardware resource utilization, obtaining an efficiency higher (i.e., 8.63 Mbps/slice) than similar solutions reported in the scientific literature. Finally, further applications of the Xilinx Zynq Ultrascale+ MPSoC ZCU102 platform for high-speed processing were explored. Exploiting the wide range of peripherals and the advanced performances in terms of data throughput and storing capability, the ZCU102 board represents a powerful and versatile tool to implement custom solutions in various operative scenarios, such as image and signal processing, visual recognition, and hardware resource management.

APPENDIX

Table 1. Summarizing table with the scientific works analyzed in section 2

Work	Application	FPGA Platform	Benefits
Ricart-Sanchez *et al.* [20]	5G-aware traffic processing	NetFPGA-SUME board (Xilinx Virtex-7 690T)	Reduced packet loss (4%) Good scalability Reduced process delay (4µs)
Kumar *et al.* [21]	Universal-Filtered Multicarrier (UFMC) system	Xilinx Virtex-5 XC5VLX20T-2FF323	Reduced quantization error (≈3.051X10^{-5})
Ferreira *et al.* [22]	Baseband modulator utilizing the DFS technique	Xilinx Zynq XC7Z020	Low processing latency (< 1 ms) Reduced power consumption (180 mW @100MHz)
Riberio *et al.* [27]	OFDM transceiver	Xilinx ML 605	High reconfigurability Reduced cost Low hardware utilization
Jiao *et al.* [28]	Multi-standards SDR architecture	Xilinx ZC706	Multiple transmissions using parallel virtual channels Parametric control of each channel Radio resources slicing
Duarte *et al.* [29]	SDR communication architecture	Xilinx VC707	Design flexibility Upgradability Reconfigurability Low cost
Pinneterre *et al.* [31]	Virtualization Manager	Xilinx Virtex-7	Dynamic remote management with commands function of hardware accelerator and VNF status.
Huang *et al.* [36]	MIMO channels emulation	Xilinx Virtex-7 VH870	Efficient emulations of a large number of rays AR(Autoregressive)-fading channel generator reduces the required memory by 95% than traditional LUT-based approach Better scalability
Danneberg *et al.* [37]	2x2 MIMO GFDM transceiver	NI USRP RIO platform (Xilinx Kintex-7 410T)	Design flexibility Reduced latency Low OOB emission (-48 dBm)
Ayoubi *et al.* [41]	GMRC receiver for MIMO channels	Altera Cyclone EP1C6F256	High throughput Low hardware utilization

REFERENCES

[1] S. Ahmadi, "Toward 5 G xilinx solutions and enablers for next-generation wireless systems," *White Paper: Xilinx MPSoCs and FPGAs*, pp. 1-32, Jun. 2016, Accessed: Oct. 27, 2020. [Online]. Available:/paper/Toward-5-G-Xilinx-Solutions-and-Enablers-for- Ahmadi/d45687958f1472bb0dc8b3e124f4376bfdc6ada1

[2] J. Bartelt, A. Betzler, G. Bianchi, S. Bjørnstad, and Y. Chang, "5G Programmable Infrastructure Converging disaggregated network and compUte REsources," Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Initial report on Data Plane Programmability and infrastructure components 3.1, Mar. 2018. Accessed: Oct. 27, 2020. [Online]. Available: https://5g-ppp.eu/5g-picture/

[3] P. Visconti, N. I. Giannoccaro, R. de Fazio, S. Strazzella, and D. Cafagna, "IoT-oriented software platform applied to sensors-based farming facility with smartphone farmer app," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1095-1105, Jun. 2020, doi: 10.11591/eei.v9i3.2177.

[4] P. Visconti, B. Sbarro, P. Primiceri, R. de Fazio, and A. L. Ekuakille, "Design and Testing of a Telemetry System Based on STM X-Nucleo Board for Detection and Wireless Transmission of Sensors Data Applied to a Single-Seat Formula SAE Car," *International Journal of Electronics and Telecommunications*, vol. 65, no. 4, pp. 671-678, Oct. 2019, doi: 10.24425/ijet.2019.130248.

[5] P. Visconti, B. Sbarro, and P. Primiceri, "A ST X-Nucleo-based telemetry unit for detection and WiFi transmission of competition car sensors data: firmware development, sensors testing and real-time data analysis," *International Journal on Smart Sensing and Intelligent Systems*, vol. 10, no. 4, pp. 793-828, Dec. 2017, doi: 10.21307/ijssis-2018-019.

[6] P. Visconti, P. Primiceri, and C. Orlando, "Solar powered wireless monitoring system of environmental conditions for early flood prediction or optimized irrigation in agriculture," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 7, pp. 4623-4632, Nov. 2016.

[7] A. Hofmann, R. Glein, L. Frank, R. Wansch, and A. Heuberger, "Reconfigurable on-board processing for flexible satellite communication systems using FPGAs," *2017 Topical Workshop on Internet of Space (TWIOS)*, Jan. 2017, doi: 10.1109/TWIOS.2017.7869767.

[8] F. Musumeci, C. Bellanzon, N. Carapellese, M. Tornatore, A. Pattavina, and S. Gosselin, "Optimal BBU Placement for 5G C-RAN Deployment Over WDM Aggregation Networks," *Journal of Lightwave Technology*, vol. 34, no. 8, pp. 1963-1970, Apr. 2016, doi: 10.1109/JLT.2015.2513101.

[9] B. M. Khorsandi and C. Raffaelli, “BBU location algorithms for survivable 5G C-RAN over WDM,” *Computer Networks*, vol. 144, pp. 53-63, Oct. 2018, doi: 10.1016/j.comnet.2018.07.026.

[10] W. Lie and W. Feng-yan, “Dynamic Partial Reconfiguration in FPGAs,” *2009 Third International Symposium on Intelligent Information Technology Application*, Nov. 2009, doi: 10.1109/IITA.2009.334.

[11] M. J. Wirthlin and B. L. Hutchings, “Improving functional density using run-time circuit reconfiguration [FPGAs],” *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 6, no. 2, pp. 247-256, Jun. 1998, doi: 10.1109/92.678880.

[12] M. Hadzialic, B. Dosenovic, M. Dzaferagic, J. Musovic, “Cloud-RAN: Innovative radio access network architecture,” *Int. Symposium on Electronics in Marine*, Sep. 2013, pp. 115-120.

[13] A. Aijaz, “Hap-SliceR: A Radio Resource Slicing Framework for 5G Networks with Haptic Communications,” *IEEE Systems Journal*, vol. 12, no. 3, pp. 2285-2296, Sep. 2018, doi: 10.1109/JSYST.2017.2647970.

[14] C.-I. Badoi, N. Prasad, V. Croitoru, and R. Prasad, “5G Based on Cognitive Radio,” *Wireless Pers Commun*, vol. 57, no. 3, pp. 441-464, Apr. 2011, doi: 10.1007/s11277-010-0082-9.

[15] S. Haene, D. Perels, and W. Fichtner, “System-level characterization of a real-time 4×4 MIMO-OFDM transceiver on FPGA,” *European Signal Processing Conf.*, Sep. 2007, pp. 1146-1150.

[16] P. Visconti, S. Capoccia, E. Venere, R. Velázquez, and R. de Fazio, “10 Clock-Periods Pipelined Implementation of AES-128 Encryption-Decryption Algorithm up to 28 Gbit/s Real Throughput by Xilinx Zynq UltraScale+MPSoC ZCU102 Platform,” *Electronics*, vol. 9, no. 10, Oct. 2020, doi: 10.3390/electronics9101665.

[17] Xilinx, “ZCU102 Evaluation Board User Guide,” 2019. [Online]. Available: https://www.xilinx.com/support/documentation/boards_and_kits/zcu102/ug1182-zcu102-eval-bd.pdf (Accessed Sep. 16, 2020).

[18] V. Chamola, S. Patra, N. Kumar, and M. Guizani, “FPGA for 5G: Re-configurable Hardware for Next Generation Communication,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 140-147, Jun. 2020, doi: 10.1109/MWC.001.1900359.

[19] Z. Wang, Q. Tang, B. Guo, J.-B. Wei, and L. Wang, “Resource partitioning and application scheduling with module merging on dynamically and partially reconfigurable FPGAs,” *Electronics*, vol. 9, no. 9, Sep. 2020, doi: 10.3390/electronics9091461.

[20] R. Ricart-Sanchez, P. Malagon, P. Salva-Garcia, E. C. Perez, Q. Wang, and J. M. Alcaraz Calero, “Towards an FPGA-Accelerated programmable data path for edge-to-core communications in 5G networks,” *Journal of Network and Computer Applications*, vol. 124, pp. 80-93, Dec. 2018, doi: 10.1016/j.jnca.2018.09.012.

[21] V. Kumar, M. Mukherjee, and J. Lloret, “Reconfigurable Architecture of UFMC Transmitter for 5G and Its FPGA Prototype,” *IEEE Systems Journal*, vol. 14, no. 1, pp. 28-38, Mar. 2020, doi: 10.1109/JSYST.2019.2923549.

[22] M. Lopes Ferreira and J. Canas Ferreira, “An FPGA-Oriented Baseband Modulator Architecture for 4G/5G Communication Scenarios,” *Electronics*, vol. 8, no. 1, Jan. 2019, doi: 10.3390/electronics8010002.

[23] A. R. Jafri, J. Majid, L. Zhang, M. A. Imran, and M. Najam-ul-Islam, “FPGA Implementation of UFMC Based Baseband Transmitter: Case Study for LTE 10MHz Channelization,” *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-13, Jul. 2018, doi: 10.1155/2018/2139794.

[24] S. Mhaske, H. Kee, T. Ly, A. Aziz, and P. Spasojevic, “FPGA-Based Channel Coding Architectures for 5G Wireless Using High-Level Synthesis,” *International Journal of Reconfigurable Computing*, vol. 2017, pp. 1-24, Jun. 2017, doi: https://doi.org/10.1155/2017/3689308.

[25] Q. Abdullah, M. Balfaqih, N. S. Mohd Shah, and N. Abdullah., “Maximising system throughput in wireless powered sub-6 GHz and millimetre-wave 5G heterogeneous networks,” *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 18, no. 3, pp. 1185-1195, Jun. 2020, doi: 10.12928/telkomnika.v18i3.15049.

[26] M. Jaoua, “Development of an FPGA-based High-Speed Wireless Communication System in the 60GHz frequency band for CERN facilities and 5G deployment,” *Digitala Vetenskapliga Arkivet*., 2018.

[27] C. Ribeiro and A. Gameiro, “A software-defined radio FPGA implementation of OFDM-based PHY transceiver for 5G,” *Analog Integr Circ Sig Process*, vol. 91, no. 2, pp. 343-351, May 2017, doi: 10.1007/s10470-017-0939-x.

[28] X. Jiao, I. Moerman, W. Liu, and F. A. P. de Figueiredo, “Radio hardware virtualization for coping with dynamic heterogeneous wireless environments,” *Cognitive Radio Oriented Wireless Networks*, Cham, 2018, pp. 287-297, doi: 10.1007/978-3-319-76207-4_24.

[29] L. Duarte, R. Gomes, C. Ribeiro, and R. F. S. Caldeirinha, “A Software-Defined Radio for Future Wireless Communication Systems at 60 GHz,” *Electronics*, vol. 8, no. 12, Dec. 2019, doi: 10.3390/electronics8121490.

[30] H. Tana, A. N. Sazish, A. Ahmad, M. S. Sharif, and A. Amira, “Efficient FPGA implementation of a wireless communication system using Bluetooth connectivity,” *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2010, pp. 1767–1770, doi: 10.1109/ISCAS.2010.5537610.

[31] S. Pinneterre, S. Chiotakis, M. Paolino, and D. Raho, “vFPGAmanager: A Virtualization Framework for Orchestrated FPGA Accelerator Sharing in 5G Cloud Environments,” *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2018, pp. 1-5, doi: 10.1109/BMSB.2018.8436930.

[32] N. Bel-Haj-Maati, N. Malhouroux, P. Pajusco, and M. Ney, “Massive MIMO Channel Measurements and Simulations at 3.7 GHz in Outdoor Environment,” *2019 23rd International Conference on Applied Electromagnetics and Communications (ICECOM)*, Sep. 2019, doi: 10.1109/ICECOM48045.2019.9163640.

[33] R. Ayoubi, J. P. Dubois, and O. Abdul-Latif, “FPGA Implementation of a Novel Receiver Diversity Combining Technique for Wireless SIMO Systems,” *2007 IEEE International Conference on Signal Processing and Communications*, Nov. 2007, pp. 37-40, doi: 10.1109/ICSPC.2007.4728249.

[34] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, “A flexible 100-antenna testbed for Massive MIMO,” *2014 IEEE Globecom Workshops (GC Wkshps)*, Austin, TX, USA, Dec. 2014, pp. 287-293, doi: 10.1109/GLOCOMW.2014.7063446.

[35] P. Murphy, C. Shepard, L. Zhong, C. Dick, and A. Sabharwal, “FPGAs Help Characterize Massive-MIMO Channels,” *XceLL Journal*, vol. 89, pp. 19-25, 2014.

[36] P. Huang, M. J. Tonnemacher, Y. Du, D. Rajan, and J. Camp, “Towards Massive MIMO Channel Emulation: Channel Accuracy Versus Implementation Resources,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4635-4651, May 2020, doi: 10.1109/TVT.2020.2980583.

[37] M. Danneberg, *et al.*, “Implementation of a 2 by 2 MIMO-GFDM transceiver for robust 5G networks,” *2015 International Symposium on Wireless Communication Systems (ISWCS)*, Aug. 2015, pp. 236-240, doi: 10.1109/ISWCS.2015.7454336.

[38] X. Liu, D. He, and M. Jia, “5G-based wideband cognitive radio system design with cooperative spectrum sensing,” *Physical Communication*, vol. 25, pp. 539-545, Dec. 2017, doi: 10.1016/j.phycom.2017.09.010.

[39] S. Bhandari and S. Joshi, “Cognitive Radio Technology in 5G Wireless Communications,” *2018 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Oct. 2018, pp. 1115-1120, doi: 10.1109/ICPEICES.2018.8897345.

[40] M. Jia, X. Gu, Q. Guo, W. Xiang, and N. Zhang, “Broadband hybrid satellite-terrestrial communication systems based on cognitive radio toward 5G,” *IEEE Wireless Communications*, vol. 23, no. 6, pp. 96-106, Dec. 2016, doi: 10.1109/MWC.2016.1500108WC.

[41] R. Ayoubi and J. Daba, “FPGA Design of Spatially Modulated Single-Input-Multiple-Output Signals in 5G Diversity Receivers,” *2019 IEEE Int. Conference on Communication, Networks and Satellite (Comnetsat)*, Aug. 2019, pp. 24-29, doi: 10.1109/COMNETSAT.2019.8844099.

[42] R. Ayoubi, J. Dubois, and R. Minkara, “FPGA Implementation of Generalized Maximal Ratio Combining Receiver Diversity,” *World Academy of Science*, vol. 4, no. 8, pp. 1264-1268, 2010.

[43] J. Dubois, R. Minkara, and R. Ayoubi, “Generalized Maximal Ratio Combining as a Supra-optimal Receiver Diversity Scheme,” *International Journal of Electronics and Communication Engineering*, vol. 4, no. 8, pp. 1259-1263, 2010.

[44] S. Sinha Roy, F. Turan, K. Jarvinen, F. Vercauteren, and I. Verbauwhede, “FPGA-Based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data,” *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb. 2019, pp. 387-398, doi: 10.1109/HPCA.2019.00052.

[45] S. Amiri, M. Hosseinabady, A. Rodriguez, R. Asenjo, A. Navarro, and J. Nunez-Yanez, “Workload Partitioning Strategy for Improved Parallelism on FPGA-CPU Heterogeneous Chips,” *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, Aug. 2018, pp. 376-380, doi: 10.1109/FPL.2018.00071.

[46] E. B. Kavun, N. Mentens, J. Vliegen, and T. Yalçın, “Efficient Utilization of DSPs and BRAMs Revisited: New AES-GCM Recipes on FPGAs,” *2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Dec. 2019, pp. 1-2, doi: 10.1109/ReConFig48160.2019.8994730.

[47] H.-Y. Kim, *et al.*, “SafeDB: Spark Acceleration on FPGA Clouds with Enclaved Data Processing and Bitstream Protection,” *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, Jul. 2019, pp. 107-114, doi: 10.1109/CLOUD.2019.00029.

[48] Xilinx, “UltraScale Architecture and Product Data Sheet: Overview,” 2020. [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf (Accessed Oct. 30, 2020).

[49] Y. Wang and Y. Ha, “FPGA-Based 40.9-Gbits/s Masked AES With Area Optimization for Storage Area Network,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 1, pp. 36-40, Jan. 2013, doi: 10.1109/TCSII.2012.2234891.

[50] T. Good and M. Benaissa, “AES on FPGA from the Fastest to the Smallest,” *Cryptographic Hardware and Embedded Systems- CHES 2005*, 2005, pp. 427-440, doi: 10.1007/11545262_31.

[51] L. Daoud, F. Hussein, and N. Rafla, “Optimization of Advanced Encryption Standard (AES) Using Vivado High Level Synthesis (HLS),” *CATA*, 2019, vol. 58, pp. 36-44, doi: 10.29007/X3TX.

[52] J. M. Granado-Criado, M. A. Vega-Rodríguez, J. M. Sánchez-Pérez, and J. A. Gómez-Pulido, “A new methodology to implement the AES algorithm using partial and dynamic reconfiguration,” *Integration*, vol. 43, no. 1, pp. 72-80, Jan. 2010, doi: 10.1016/j.vlsi.2009.05.003.

[53] Xinmiao Zhang and K. K. Parhi, “High-speed VLSI architectures for the AES algorithm,” *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, vol. 12, no. 9, pp. 957-967, Sep. 2004, doi: 10.1109/TVLSI.2004.832943.

[54] L. Henzen and W. Fichtner, “FPGA parallel-pipelined AES-GCM core for 100G Ethernet applications,” *2010 Proceedings of ESSCIRC*, Sep. 2010, pp. 202-205, doi: 10.1109/ESSCIRC.2010.5619894.

[55] W. Liu, W. Li, P. S. Un, and Y. Beom Cho, “High-throughput HW-SW implementation for MV-HEVC decoder,” *2018 International SoC Design Conference (ISOCC)*, Daegu, South Korea, Nov. 2018, pp. 226-228, doi: 10.1109/ISOCC.2018.8649965.

[56] H. Huang, J. Xia, and S. Boumaiza, "Novel Parallel-Processing-Based Hardware Implementation of Baseband Digital Predistorters for Linearizing Wideband 5G Transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 9, pp. 4066-4076, Sept. 2020, doi: 10.1109/TMTT.2020.2993236.

[57] J. Ordoñez and G. Sahonero-Alvarez, "A 3D Convolution Accelerator Implemented on FPGA Using SDSoC," *2019 7th International Engineering, Sciences and Technology Conference (IESTEC)*, Oct. 2019, pp. 677-681, doi: 10.1109/IESTEC46403.2019.00126.

[58] M. Véstias, R. Duarte, J. Sousa, and H. Neto, "Fast convolutional neural networks in low density fpgas using zero-skipping and weight pruning," *Electronics*, vol. 8, pp. 1-24, Nov. 2019, doi: 10.3390/electronics8111321.

[59] H. Nakahara, M. Shimoda, and S. Sato, "A demonstration of FPGA-based you only look once version2 (YOLOv2)," *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, Aug. 2018, pp. 457-458, doi: 10.1109/FPL.2018.00088.

[60] P. Janus and T. Kryjak, "Hardware implementation of the Gaussian Mixture Model foreground object segmentation algorithm working with ultra-high resolution video stream in real-time," *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, Sep. 2018, pp. 140-145, doi: 10.23919/SPA.2018.8563404.

[61] Y. Guo, H. Zheng, J. Wang, S. Xiao, G. Li, and Z. Yu, "A Low-Cost and High-Throughput Virtual-Channel Router with Arbitration Optimization," *2019 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*, Nov. 2019, pp. 75-76, doi: 10.1109/ICTA48799.2019.9012817.

[62] C.-A. Shen, D.-Y. Lee, C.-A. Ku, M.-W. Lin, K.-C. Lu, and S.-Y. Tan, "A Programmable and FPGA-accelerated GTP Offloading Engine for Mobile Edge Computing in 5G Networks," *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 1021-1022, doi: 10.1109/INFCOMW.2019.8845143.

[63] X. Zhang, Y. Li, and G. Chen, "NVMe-over-RPMsg: A virtual storage device model applied to heterogeneous multi-core SoCs," *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2020, pp. 821-826, doi: 10.1109/CCWC47524.2020.9031144.

[64] M. Qasaimeh, K. Denolf, J. Lo, K. Vissers, J. Zambreno, and P. H. Jones, "Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels," *2019 IEEE International Conference on Embedded Software and Systems (ICESS)*, Jun. 2019, pp. 1-8, doi: 10.1109/ICESS.2019.8782524.

[65] X. Di, H.-G. Yang, Y. Jia, Z. Huang, and N. Mao, "Exploring Efficient Acceleration Architecture for Winograd-Transformed Transposed Convolution of GANs on FPGAs," *Electronics*, vol. 9, pp. 1-21, Feb. 2020, doi: 10.3390/electronics9020286.

[66] J. J. Rodríguez-Andina, M. D. Valdés-Peña, M. J. Moure, "Advanced Features and Industrial Applications of FPGAs-A Review," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 853-864, August 2015, doi: 10.1109/TII.2015.2431223.

[67] I. F. Akyildiz, A. Kak, and S. Nie, "6G and Beyond: The Future of Wireless Communications Systems," *IEEE Access*, vol. 8, pp. 133995-134030, Jul. 2020, doi: 10.1109/ACCESS.2020.3010896.