ASIP for 5G and Beyond: Opportunities and Vision

Shahriar Shahabuddin, *Member, IEEE*, Aarne Mämmelä, *Senior Member, IEEE*,
Markku Juntti, *Fellow, IEEE*, and, Olli Silvén *Senior Member, IEEE*

***Abstract*—The tutorial discusses application-specific instruction-set processors (ASIP) and their potential for the fifth generation (5G) and beyond 5G networks. ASIP is a class of customized processor, typically designed for a single or a small set of applications, incorporating flexibility of a software implementation and efficiency of a register transfer level (RTL) based design. In this paper, we present advantages, methodology, tool sets and a summary of state-of-the-art ASIPs for communication systems. In addition, we identify key features of 5G and beyond 5G systems where flexibility is a key requirement and thus, ASIPs can play a pivotal role.**

***Index Terms*—ASIP, 5G, B5G, 6G, RTL, ASIC, FPGA, VLSI**

I. Introduction

The complexity of digital very large-scale integration (VLSI) systems has been increasing dramatically due to the advent of several disruptive technologies such as fifth generation (5G) wireless networks [1], Internet of Things (IoT) [2] and deep learning [3]–[5]. Deep sub-micron effects such as noise and quantum effects are becoming more profound as we are going towards smaller transistors. These deep sub-micron phenomena are not always properly captured at the register transfer level (RTL) designs, which are more susceptible to faults for fabrication process technologies under 10 nm [6], [7]. This means that several costly iterations are required for a rigorous verification of modern VLSI systems. On the other hand, software (SW) implementations on off-the-shelf general purpose processors (GPPs), digital signal processors (DSPs) and micro-controllers (μCs) typically cannot provide the required performance and energy efficiency due to their general purpose nature [8].

An application-specific instruction-set processor (ASIP) is a viable alternative for trading off the general purpose and application specific nature of the designs. An ASIP incorporates the flexibility of SW implementations and efficiency of the RTL-based designs. Hardware (HW)/SW co-design method, which is typically used to design an ASIP, can be applied to exploit the flexibility of SW or high speed of the HW whenever necessary [9]–[11]. An intuitive diagram to compare performance and flexibility of different implementations is shown in Fig. 1. ASIP lies between two extremes of SW implementations on GPP and application-specific integrated circuits (ASIC) designs. In addition to flexibility, another important advantage of using ASIPs is the reduced time to market. An ASIP is typically designed with a tool set and a C-programmable processor template. Thus, an ASIP reduces the design gap between available resources on a chip and development productivity in terms of the number of transistors per designer effort in person-months. The motivation of designing an ASIP is not to outperform pure monolithic HW accelerators in terms of throughput. A reasonable throughput and area efficiency performance (in bit/(s$\times$mm^2)) can be obtained through ASIP design without compromising programmability [12]. ASIPs are complementary to HW accelerators rather than mutually exclusive. As the semiconductor technology is already mature and it was forecast that "transistors could stop shrinking in 2021" [13], the focus of the research community has been shifting towards devices and systems, which also suit the "more than Moore" vision where system design proceeds vertically from applications [14]. As the emphasis on systems is increasing, the demands for flexible implementations, such as ASIP, will continue to increase [15].

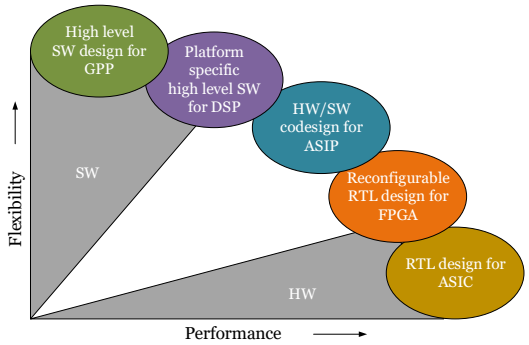

Fig. 1: Performance vs. flexibility of different implementations.

In this paper, a tutorial on ASIPs and their potential application for communication systems of 5G and beyond is presented. To our best knowledge, there is no recent tutorial or review on ASIPs. The most recent publications, which provide a general overview of the advantages and tool sets of ASIPs, have been published almost a decade ago [16], [17]. As a result, applications and potential of ASIPs for 5G and beyond are out of their scope. The rest of the paper is organized in the following way: In Section II, we briefly present ASIP design methodology and tool sets. In Section III, we present use cases of ASIPs for communication systems. In Section IV, we present algorithms and features of 5G where flexibility is a key requirement and thus, ASIP can play a pivotal role to reduce time to market. In Section V, we present ASIP applications for beyond 5G (B5G) networks, based on our vision of next generation communication systems. The

S. Shahabuddin is with Mobile Networks, Nokia, Finland, e-mail: shahriar.shahabuddin@nokia.com.

A. Mämmelä is with VTT Technical Research Centre of Finland, Finland, e-mail: aarne.mammela@vtt.fi.

M. Juntti and O. Silvén are with Centre for Wireless Communications (CWC), and Center for Machine Vision and Signal Analysis (CMVS), respectively, of University of Oulu, Finland, e-mail: firstname.lastname@oulu.fi.

conclusions are drawn in Section VI.

II. ASIP Design Flow and Development Tool Sets and

A programmable ASIP development is heavily dependent on tool sets because it is not straightforward to design an entire working processor from scratch. In this section, we review the generic ASIP design flow and a few HW/SW co-design tool sets and methods for ASIPs, which are commonly used for developing communication system IPs. These tool sets assist a VLSI designer to develop an ASIP without requiring a profound knowledge of processor architectures or instruction sets.

A. ASIP design flow

ASIP design typically starts with writing the target application in a high level language such as C/C++. At the same time, a processor is developed for the target application either from an architecture description language (ADL) or using a graphical user interface (GUI). In the case of the ADL based method, the input of the processor designer tool is an architecture model described in the ADL. They enable a high processor design efficiency by a declarative description style [18]. In a GUI based processor design method, the designer can modify a processor template in a GUI by adding or removing function units (FUs) such as adder, multiplier etc. The core tool in ASIP design tool sets is a retargetable compiler, which can generate assembly code for different target processor architectures [19]. Rather than manually writing assembly code, the retargetable compiler enables a designer to generate assembly code from a high level programming language. In Fig. 2, a generic framework for ASIP design is shown, where a retargetable compiler uses the processor model in addition to the application written in C/C++. Typically, a simulator is also part of the tool sets to analyze the execution of the C/C++ program on the processor with detailed information about the resource usages and cycle counts during the execution. If the designed processor does not meet the target performance, the designer can go back to the C/C++ code or modify the processor model. The final step involves the generation of hardware description language (HDL) of the processor. A HW database is typically provided with the ASIP design tool sets, which include HDL description of common FUs, register files (RFs), interconnects etc. The tool sets also allow the designer to build custom FUs to reduce the processing latency. In that case, the designer has to write the HDL description for the custom unit and add it in the HW database. An instruction for SW needs to be created for this custom unit, which can be used in the C/C++ code as an intrinsic. The generated HDL can be simulated, tested, and synthesized by third party tools.

B. ASIP development tool sets

1) PEAS: The practical environment for ASIP development (PEAS) environment takes architecture specification, available resource description and instruction format from the designer to generate HDL [20], [21].

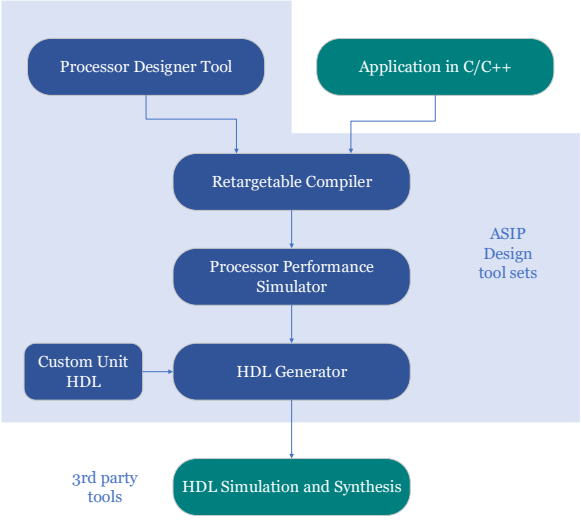

Fig. 2: A generic framework for ASIP design.

2) Tensilica: Tensilica Inc. developed configurable microprocessor cores, which are known as Xtensa [22]. There are several units of Xtensa which are optional and can be added if it is deemed necessary by the designer. Cadence acquired Tensilica and currently offers the tool set [23].

3) LISATek tool set: This is another integrated embedded processor development environment which was originally developed at the RWTH Aachen University, Germany [24], [25]. LISATek tool set uses an ADL called LISA 2.0 to describe an ASIP model.

4) Target tool suite: nML language was originally developed at the TU Berlin, Germany for describing processor instruction-sets [26]. Based on nML, a retargetable C-compilerand an instruction-set simulator generator were developed by IMEC, Belgium [27] and commercialized by Target Compiler Technologies, which is known as Target tool suite.

5) RISC-V and Rocket: Reduced Instruction Set Computer V (RISC-V) is an open-source and extensible instruction-set architecture (ISA), which has been gaining a lot of attention [28], [29]. RISC-V provides a programmable processor base for custom accelerators. An open-source tool set called Rocket Chip Generator tool, developed by the University of California, Berkeley, CA, can be used to generate RISC-V architectures with possible customized HW extension [30].

6) Synopsys ASIP designer: Both LISATek and Target tool sets were acquired by Synopsys to improve their processor design portfolio. Synopsys ASIP designer is currently the leading commercial ASIP design tool in the market [31].

7) TCE: The TCE tool set was developed at the University of Tampere, Finland [32]. TCE is an open source tool set for designing, implementing and simulating ASIPs with transport triggered architecture (TTA) [33].

8) Others: A few other tool sets such as MIMOLA [34], EXPRESSION [35], TiPi [36] are worth to mention here. We invite interested readers to browse through [8] to explore some other tools available for ASIP design.

III. ASIP USE CASES FOR COMMUNICATION SYSTEMS

In this section, we discuss the most important use cases of ASIPs for communication systems.

1) FIR filtering: Finite impulse response (FIR) filter is one of the early use cases of ASIPs pertaining to communication systems. In [37], an ASIP for FIR filtering is designed with PEAS-III system. The authors show that it is possible to rapidly modify the design and explore different micro-architectures. A scalable ASIP for FIR filtering, which is implemented with TCE, is presented in [38]. The authors use a variable number of complex multiply and accumulate (MAC) units to meet the requirements and conclude that most design complexity resides in SW.

2) Channel decoding: A plethora of ASIP implementations for channel decoders can be found in the literature [39]–[43]. A recurrent theme for the channel decoder ASIPs is supporting several decoding algorithms with a single ASIP core. For example, a reconfigurable ASIP to support both Viterbi and logarithmic maximum a posteriori probability (Log-MAP) decoding is presented in [41]. The flexible implementation, named FlexiTreP by the authors, is designed with the LISATek tool set and can support more than ten wireless communication standards. A single ASIP to support both turbo and low-density parity-check (LDPC) decoding can be found in [42]. The ASIP chip, called FlexFEC, is designed with Target tool suite and provides a performance-competitive solution to HW designs with an added advantage of flexibility. Another turbo decoder ASIP, which is designed with TCE, is presented in [40]. The TCE ASIP is comparable to pure HW implementations in terms of performance. However, the ASIP is programmed with assembly language and thus, the development time and verification efforts are also comparable to pure HW designs. Multiple ASIPs working in parallel for different parts of an algorithm is also an attractive solution to improve the overall performance. For example, a multiprocessor based turbo decoder is proposed in [39]. Each ASIP executes a soft-input soft-output decoding algorithm and thus can multiply the overall data rate of the turbo decoder. A similar multi-ASIP architecture can be found in [43].

3) MIMO detection: Multiple-input multiple-output (MIMO) symbol detection at the receiver has been another popular use case of ASIPs during last two decades. ASIPs for selective spanning with fast enumeration (SSFE) algorithm are presented in [44] and [45], which are developed with the TCE and Target tool sets respectively. An ASIP, designed with LISATek, to support various sphere decoding MIMO signal detection algorithms is presented in [46]. The authors of [44], [45], and [46] agree that ASIPs supporting a single MIMO detection algorithm cannot compete with RTL-based ASIC solutions in terms of area and power efficiency. However, in [44], the authors state that adding flexibility to the RTL-based MIMO detectors increases power dissipation swiftly and can be less energy efficient than an ASIP solution. A no-instruction-set-computer (NISC) ASIP for soft demapping can be found in [47]. A single ASIP supporting multiple detection algorithms, also known as multimode detector, can be found in [48]. The reconfigurable ASIP, named FLEXDET by the authors, supports minimum mean-square error (MMSE), successive interference cancellation and Markov chain Monte Carlo detection methods. A dedicated ASIC for MMSE takes half the area of FLEXDET. In other words, FLEXDET can support three detection algorithms in the same area of two ASICs for detection.

4) FFT: Fast Fourier transform (FFT) is another important use case for ASIPs and several FFT ASIPs can be found in the literature [49]–[51]. The FFTs presented in [49] and [50] are designed with Tensilica and LISATek, respectively, and comparable to ASIC solutions in terms of power efficiency. Recently, a flexible FFT is presented in [51] to support fourth generation (4G), wireless local area network (WLAN), and 5G systems. The programmable FFT ASIP is designed with Synopsys ASIP designer and can provide a higher data rate than most state-of-the-art FFT designs.

5) MIMO pre-processing: ASIPs are also used to implement pre-processing algorithms for a MIMO transceiver, such as QR decomposition and lattice reduction [52]–[54]. A scalable lattice reduction algorithm ASIP is presented in [53]. The authors configured a processor design template, called ADRES, to design their ASIP. In [54], A multi-ASIP architecture for lattice reduction is presented. Each ASIP supports a single iteration of the lattice reduction algorithm in [54]. Both lattice reduction ASIPs are programmable and can provide performance-competitive solutions compared to HW designs. In [52], an ASIP for complex-valued QR decomposition is designed with TCE. The QR ASIP solution is significantly inferior to systolic array based VLSI architectures in terms of performance as the main objective of the design is flexibility and programmability. With a modest effort, the ASIP can be modified to support also FFT operation.

6) Others: A plethora of ASIPs can be found in the literature for other applications pertaining to communication systems. An ASIP to support two broadcast precoding algorithms is presented in [55]. A hybrid precoding ASIP, designed with Synopsys ASIP designer, can be found in [56]. Other such applications include packet detection [57], multicarrier modulation [58], encryption [59] etc. ASIP implementations for a complete baseband chain can also be found in the literature [60], [61]. A low-power ASIP for digital baseband processing of ultra-wideband impulse radio is presented in [60]. In [61], an ASIP is presented for baseband processing of 3G and 4G mobile handsets. ASIPs used in other domains, such as image processing, can also be applied to communication systems [62], [63]. Numerous fully RTL based ASIPs can be found in the literature [64], [65]. For example, an ASIP for channel estimation is presented in [64], which is completely designed with Verilog. We focus only on ASIPs which are programmable with high level SW language, such as C, due to their superior time to market and flexibility.

IV. ASIP FOR 5G: OPPORTUNITIES

5G is the latest technology standard for broadband communications and the successor of 4G networks [66]. 5G radio (NR) interface promises a higher data rate by adopting technologies such as massive MIMO, millimeter (mm) wave

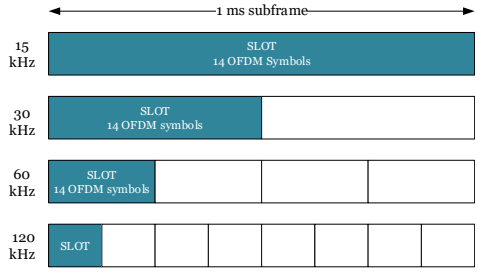

Fig. 3: 5G mixed numerology with flexible slot and symbol duration.

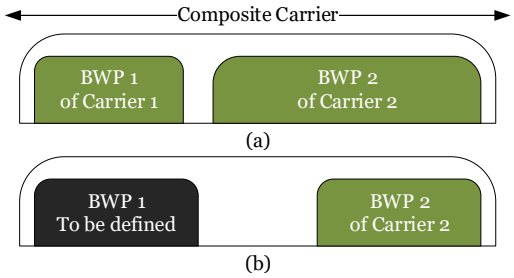

Fig. 4: 5G NR BWP: (a) different types of carriers inside a composite carrier, (2) a composite carrier with all signals yet to be defined.

communications, network slicing, etc. [67], [68]. We discuss potential use cases of ASIPs for 5G systems in this section.

1) Channel decoding for 5G: Unlike the 4G standard, the 5G NR adopted polar decoding for control channels and LDPC decoding for data channels [69]. Typically, IP cores for LDPC and polar decoders are developed or acquired separately by a network developer. A single ASIP dedicated for both LDPC and polar decoders can be a convenient solution. In [70], an ASIP to support four decoding algorithms, which includes LDPC and polar decoding, is presented. The quad-mode ASIP occupies much smaller area than the sum of four single-mode decoders. However, the ASIP is programmed with an assembly language. More research is necessary for C-programmable solutions.

2) Massive MIMO detectors for 5G uplink: Massive MIMO is a key 5G physical layer technology where the base station (BS) is equipped with a large number of antennas to support a large number of single-antenna user equipment (UEs). Due to the high complexity introduced by a large number of UEs and antennas, a new class of detectors based on approximate matrix inversion has been a popular choice for massive MIMO [71]. A plethora of HW implementation can be found in the literature for 128 antenna BS, and 8 UEs, i.e., 128×8. However, these detectors cannot provide satisfactory performance when the ratio of numbers between BS antennas and users are small, e.g., for a 128×32 system. A more conventional detection algorithm is necessary in such a case [72], [73]. Therefore, ASIPs can be a convenient solution to support separate detection algorithms in a single design for different MIMO configurations. ASIPs for massive MIMO detection have not been explored properly. In [74], a fully C-programmable ASIP is designed with the Synopsys tool set to support 128×16 MIMO detection. We believe there is an huge opportunity for ASIPs to support multimode massive MIMO detectors.

3) Mixed numerology for OFDM: Orthogonal frequency-division multiplexing (OFDM) is adopted for both downlink and uplink of 5G NR for multicarrier modulation [75]. However, the subcarrier spacing (SCS) of OFDM is no longer fixed to 15 kHz in 5G. Instead, SCS scales with $2^{\mu} \times 15$ kHz to support different quality of service (QoS) and latency requirements where $\mu = 0 \ldots 5$. Such a flexible SCS allocation enables 5G systems to change their slot and OFDM symbol durations as shown in Fig. 3. This concept of a flexible configuration for multicarrier modulation is known as mixed numerology where each numerology corresponds to a SCS. Mixed numerology requires a flexible HW implementation, which can be challenging. A typical RTL solution can use 120 kHz as a base, which has the lowest granularity in Fig. 3. As the number of slots or symbols are a multiple of the 120 kHz SCS, it is possible to adapt the counters and FSMs to support other SCS. However, the SCS used as the base can change due to product requirements. For example, a support of 240 kHz subcarrier spacing might be needed instead of 120 kHz. In that case, it will be very difficult or nearly impossible to update the RTL to support 240 kHz subcarrier. Therefore, an ASIP can be very useful to support the mixed numerology for 5G. An ASIP for general frequency division multiplexing (GFDM) is recently proposed in [58]. A similar approach can be adopted to support mixed numerology for OFDM and thus, presents a huge opportunity for ASIP designers.

4) Bandwidth parts for 5G: Bandwidth parts (BWP) is a key concept of 5G NR to enable more flexibility for resource allocation to a composite carrier [76]. 5G systems use signals with a wider bandwidth than earlier generations. However, UE capabilities will vary and BS also has to select carriers according to a UE capability. For example, 20 MHz and 40 MHz carriers can be aggregated together in the same composite carrier in 5G systems to serve a different purpose. In Fig. 4 (a), we present such a scenario where two carriers with different bandwidths are aggregated together. In Fig. 4 (b), we see another scenario where the network developers have to support a carrier whose bandwidth is yet to be defined. As a consequence, the channel filters, which isolate different carriers at the transceiver, will require a flexible implementation. Firstly, a variable number of coefficients will be required for the filtering to support different carrier types. Secondly, the filters needs to have a later field update capability for a carrier which is not yet properly defined. Therefore, ASIPs can be an ideal solution for such flexible channel filters. Improved approaches to [37], [38] can be investigated to provide a better solution.

5) Interfaces between RRH and BBU: Base stations today use a remote radio head (RRH), which is typically located at the tower, and a central baseband unit (BBU), which is typically located on the ground, to extend the coverage. The RRH and BBU are connected through fibre optic cables which follows a protocol called common public radio interface (CPRI) [77]. However, for 5G, the fiber optic cables will

carry much higher traffic compared to 4G systems. Therefore, telecommunications industries released an updated protocol specification for communications between RRH and BBU of 5G. This new protocol is known as enhanced CPRI (eCPRI). The eCPRI cores are typically designed as RTL by base station developers. However, this becomes challenging as eCPRI specification is constantly evolving. ASIPs can be an ideal candidate for IP cores of eCPRI and provide programmable solutions to support future updates as well as legacy protocols.

6) Millimeter wave beam management: The idea of beamforming has existed since the 3G era, however, this has become an essential feature for 5G systems due to the introduction of mm wave communications. As the path loss and atmospheric loss is high for mm waves, the transmitter requires to focus signals or beams from multiple antennas, to transmit a stronger signal towards the receiver. This process is commonly known as beamforming. The practical beamforming implementation is not so straightforward as it involves several steps such as beam sweeping, beam measurement, beam reporting, beam determination, etc. [78]. Each of these steps can be implemented in different ways. For example, beam sweeping technique is used to locate the receiver by transmitting in all pre-defined directions in different time intervals. This process can be done in three ways: (1) Both BS and UE use beam sweeping, (2) Only BS uses beam sweeping, (3) Only UE uses beam sweeping. Similarly, there are different ways to implement other phases of 5G beamforming. Therefore, the entire beam management process requires a flexible implementation to support different techniques. Due to its flexibility, an ASIP can be a good candidate to implement the beam management process. Generic beamforming ASIPs can be found in the literature [56], however, they do not address the beam management procedure.

V. ASIP FOR BEYOND 5G SYSTEMS: VISION

Research community has shifted its focus to B5G systems. The two promising candidates for physical layer for 6G networks are (1) cell-free massive MIMO and (2) recongifurable intelligent surface (RIS) or intelligent reflective surface (IRS) technologies. We address a few promising research areas for B5G systems where ASIPs can be applied.

1) Machine learning IPs for B5G: There has been a massive interest on applying machine learning (ML) for communication systems during past couple of years. We envision that ML will play a key role for B5G systems, and replace many existing telecommunications algorithms, which are built upon the fundamentals of information theory, statistics and optimization. For example, in [79], O'Shea and Hoydis proposed a deep learning solution for end-to-end communication systems, i.e., the entire transmitter and the receiver are replaced with a deep neural network for an extremely slow Rayleigh fading channel. In addition, a plethora of ML solutions have been proposed for different parts of a communication system over past couple of years. On the other hand, ASIPs have been previously used to accelerate ML algorithms [80]–[83]. In [80], an ASIP is designed with Synopsys ASIP designer tool to accelerate support vector machines (SVM) for streaming applications. In [81], a K-nearest neighbour (KNN) ASIP is designed with Cadence Tensilica tool. Both [80] and [81] provide $40\times$ - $650\times$ speed-up over SW implementations, however, they have not been compared to pure HW designs. In [83], an ASIP for neural network is presented for environmental monitoring in WSN. The ASIP is designed with Synopsys ASIP designer and enables runtime reconfiguration with a performance-competitive solution to a HW accelerator. In a nutshell, ASIPs can be an attractive solution for accelerating machine learning algorithms, which will be used for next generation communication systems. There are cases when an ASIP cannot match RTL performance. In [82], a convolutional neural network (CNN) is accelerated by modifying an Xtensa processor with Cadence Tensilica tool. According to the authors, the CNN ASIP provides flexibility, but the performance of RTL implementations is better as expected from Fig. 1.

2) IRS controller: As mentioned before, mm wave communications suffer from high attenuation and penetration loss. Therefore, IRS proposes creating an alternative path for communications if there is a blockage between the transmitter and receiver. The transmitter transmits to this alternative path with a reflective surface at the end and this surface is used to divert the beams towards the intended receiver [84] and can also be beneficial for mobile positioning [85]. The heart of the IRS mechanism is an IRS controller which applies the necessary phase shifts. The IRS controller will require flexible implementations due to a myriad of user locations, signal types, beamforming weights, etc. ASIPs can play a crucial role for implementing a flexible and high data rate IRS controller.

3) Cell-free massive MIMO signal processing: In a cell-free massive MIMO systems, the antennas are not centralized in a bulky base station, but distributed geographically to jointly serve a small number of UEs. A central BBU, which is connected to all the antennas and front-end circuitry through cables, controls the overall communication [86]. However, cell-free massive MIMO will require decentralized or partially decentralized approach for many signal processing algorithms such as channel estimation, precoding and detection, digital predistortion, etc. Therefore, flexible implementation will be required at the front-end to support different configurations of a cell-free massive MIMO systems. For example, an ASIP capable of matrix computations can be programmed for matched filtering, which is an initial step of the partially decentralized equalization. Thus, there will be a huge opportunity for programmable ASIP designs for signal processing of a cell-free massive MIMO system.

VI. CONCLUSION

ASIPs have been offering attractive solutions for communication systems since the 4G era. Due to the advent of 5G technology, the ASIP solutions will continue to play an important role for flexible wireless system implementations. As a complementary technique to conventional digital VLSI designs, ASIPs will reduce the time to market of 5G chip development and accelerate the roll-out of 5G systems. Finally, ASIP designs have large potential for B5G communication systems, such as cell-free massive MIMO.

REFERENCES

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. MIT Press Cambridge, 2016, vol. 1, no. 2.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[6] L. A. Naviner, J.-F. Naviner, G. dos Santos Jr, E. C. Marques, and N. M. Paiva Jr, "FIFA: A fault-injection–fault-analysis-based tool for reliability assessment at RTL level," *Microelectronics Reliability*, vol. 51, no. 9-11, pp. 1459–1463, 2011.

[7] P. C. Ward and J. R. Armstrong, "Behavioral fault simulation in VHDL," in *Proc. 27th ACM/IEEE Design Automation Conference*, 1990, pp. 587–593.

[8] O. Schliebusch, H. Meyr, and R. Leupers, *Optimized ASIP Synthesis from Architecture Description Language Models*. Dordrecht, The Netherlands: Springer Science & Business Media, 2007.

[9] S. Shahabuddin, O. Silvén, and M. Juntti, "Programmable ASIPs for multimode MIMO transceiver," *Journal of Signal Processing Systems*, vol. 90, no. 10, pp. 1369–1381, 2018.

[10] S. Shahabuddin, J. Janhunen, and M. Juntti, "Design of a transport triggered architecture processor for flexible iterative turbo decoder," in *Proc. Wireless Innovation Forum Conference on Wireless Communications Technologies and Software Radio (SDR WINCOMM)*, Jan. 2013.

[11] S. Shahabuddin, J. Janhunen, M. F. Bayramoglu, M. Juntti, A. Ghazi, and O. Silvén, "Design of a unified transport triggered processor for LDPC/turbo decoder," in *Proc. 2013 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, July 2013, pp. 288–295.

[12] D. Liu, *Embedded DSP Processor Design: Application Specific Instruction Set Processors*. Oxford, UK: Elsevier, 2008.

[13] R. Courtland, "Transistors could stop shrinking in 2021," *IEEE Spectrum*, vol. 53, no. 9, pp. 9–11, 2016.

[14] M. M. Waldrop, "More than Moore," *Nature*, vol. 530, no. 7589, pp. 144–148, 2016.

[15] L. Dake, C. Zhaoyun, and W. Wei, "Trends of communication processors," *China Communications*, vol. 13, no. 1, pp. 1–16, 2016.

[16] M. Imai, Y. Takeuchi, K. Sakanushi, and N. Ishiura, "Advantage and possibility of application-domain specific instruction-set processor," *IPSJ Transactions on System LSI Design Methodology*, vol. 3, pp. 161–178, 2010.

[17] K. Karuri and R. Leupers, *Application Analysis Tools for ASIP Design: Application Profiling and Instruction-Set Customization*. New York, NY, USA: Springer Science & Business Media, 2011.

[18] G. D. Micheli, *Synthesis and Optimization of Digital Circuits*. McGraw-Hill Higher Education, 1994.

[19] R. Leupers, M. Hohenauer, J. Ceng, H. Scharwaechter, H. Meyr, G. Ascheid, and G. Braun, "Retargetable compilers and architecture exploration for embedded processors," *IEE Proceedings - Computers and Digital Techniques*, vol. 152, no. 2, pp. 209–223, 2005.

[20] J. Sato, A. Y. Alomary, Y. Honma, T. Nakata, A. Shiomi, N. Hikichi, and M. Imai, "PEAS-I: A hardware/software codesign system for ASIP development," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 77, no. 3, pp. 483–491, 1994.

[21] M. Itoh, S. Higaki, J. Sato, A. Shiomi, Y. Takeuchi, A. Kitajima, and M. Imai, "PEAS-III: An ASIP design environment," in *Proc. 2000 International Conference on Computer Design*, Sep. 2000, pp. 430–436.

[22] R. E. Gonzalez, "Xtensa: A configurable and extensible processor," *IEEE Micro*, vol. 20, no. 2, pp. 60–70, 2000.

[23] L. Lavagno, I. L. Markov, G. Martin, and L. K. Scheffer, Eds., *Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology*. CRC Press, 2016.

[24] A. Chattopadhyay, H. Meyr, and R. Leupers, "LISA: A uniform ADL for embedded processor modelling, implementation and software toolsuite generation," in *Processor Description Languages*, P. Mishra and N. Dutt, Eds. Burlington: Morgan Kaufmann, 2008, ch. 5, pp. 95–130.

[25] R. Muhammad, L. Apvrille, and R. Pacalet, "Evaluation of ASIPs design with LISATek," in *Proc. International Workshop on Embedded Computer Systems*. Springer, 2008, pp. 177–186.

[26] A. Fauth, J. Van Praet, and M. Freericks, "Describing instruction set processors using nML," in *Proc. European Design and Test Conference. ED&TC 1995*. IEEE, 1995, pp. 503–507.

[27] G. Goossens, D. Lanneer, W. Geurts, and J. Van Praet, "Design of ASIPs in multi-processor SoCs using the Chess/Checkers retargetable tool suite," in *Proc. International Symposium on System-on-Chip*, 2006, pp. 1–4.

[28] S. Davidson *et al.*, "The Celerity open-source 511-core RISC-V tiered accelerator fabric: Fast architectures and design methodologies for fast chips," *IEEE Micro*, vol. 38, no. 2, pp. 30–41, 2018.

[29] K. Asanović and D. A. Patterson, "Instruction sets should be free: The case for RISC-V," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146, Aug. 2014.

[30] K. Asanović *et al.*, "The Rocket Chip Generator," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17, Apr. 2016.

[31] J. Kreku, K. Tiensyrjä, A. Wieferink, and B. Vanthournout, "ASIP exploration and design," in *Scalable Multi-Core Architectures: Design Methodologies and Tools*, D. Soudris and A. Jantsch, Eds. Springer, 2012, pp. 81–103.

[32] O. Esko, P. Jaaskelainen, P. Huerta, S. Carlos, J. Takala, and J. I. Martinez, "Customized exposed datapath soft-core design flow with compiler support," in *Proc. International Conference on Field Programmable Logic and Applications*, 2010, pp. 217–222.

[33] H. Corporaal, "Design of transport triggered architectures," in *Proc. 4th Great Lakes Symposium on VLSI*, 1994, pp. 130–135.

[34] P. Marwedel, "The MIMOLA design system: Tools for the design of digital processors," in *Proc. 21st Design Automation Conference Proceedings*. IEEE, 1984, pp. 587–593.

[35] A. Halambi, P. Grun, V. Ganesh, A. Khare, N. Dutt, and A. Nicolau, "EXPRESSION: A language for architecture exploration through compiler/simulator retargetability," in *Proc. Design, Automation, and Test in Europe*. Springer, 2008, pp. 31–45.

[36] M. Gries and K. Keutzer, *Building ASIPs: The Mescal Methodology*. Springer Science & Business Media, 2006.

[37] S. Kobayashi, K. Mita, Y. Takeuchi, and M. Imai, "Design space exploration for DSP applications using the ASIP development system PEAS-III," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2002, pp. III–3168–III–3171.

[38] P. Salmela, T. Jarvinen, J. Takala, and T. Sipila, "Scalable FIR filtering on transport triggered architecture processor," in *Proc. International Symposium on Signals, Circuits and Systems*, vol. 2, 2005, pp. 493–496, vol. 2.

[39] O. Muller, A. Baghdadi, and M. Jézéquel, "ASIP-based multiprocessor SoC design for simple and double binary turbo decoding," in *Proc. Design Automation & Test in Europe Conference*, vol. 1, 2006, pp. 1–6.

[40] P. Salmela, H. Sorokin, and J. Takala, "A programmable Max-Log-MAP turbo decoder implementation," *VLSI Design*, vol. 2008, 2008.

[41] T. Vogt and N. Wehn, "A reconfigurable application specific instruction set processor for Viterbi and Log-MAP decoding," in *Proc. IEEE Workshop on Signal Processing Systems Design and Implementation*, 2006, pp. 142–147.

[42] F. Naessens *et al.*, "A 10.37 mm^2 675 mw reconfigurable LDPC and Turbo encoder and decoder for 802.11n, 802.16e and 3GPP-LTE," in *Proc. Symposium on VLSI Circuits*, 2010, pp. 213–214.

[43] A. R. Jafri, A. Baghdadi, M. Najam-ul-Islam, and M. Jézéquel, "Heterogeneous multi-ASIP and NoC-based architecture for adaptive parallel TBICM-ID-SSD," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 3, pp. 259–263, 2017.

[44] J. Janhunen, T. Pitkänen, O. Silvén, and M. Juntti, "Fixed-and floating-point processor comparison for MIMO-OFDM detector," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 8, pp. 1588–1598, 2011.

[45] U. Ahmad, M. Li, A. Amin, M. Li, L. Van der Perre, R. Lauwereins, and S. Pollin, "Towards approaching near-optimal MIMO detection performance ONAC-programmable baseband processor," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 3893–3897.

[46] J. Rust, C. Osewold, and S. Paul, "Implementation of a low power low complexity ASIP for various Sphere Decoding algorithms," in *Proc. 17th European Wireless 2011 - Sustainable Wireless Technologies*, 2011, pp. 1–6.

[47] M. Rizk, A. Baghdadi, M. Jézéquel, Y. Mohanna, and Y. Atat, "NISC-based soft-input–soft-output demapper," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 11, pp. 1098–1102, 2015.

[48] X. Chen *et al.*, "FLEXDET: Flexible, efficient multi-mode MIMO detection using reconfigurable ASIP," in *Proc. 2012 IEEE 20th International*

Symposium on Field-Programmable Custom Computing Machines, 2012, pp. 69–76.

[49] X. Guan, Y. Fei, and H. Lin, "Hierarchical design of an application-specific instruction set processor for high-throughput and scalable FFT processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 3, pp. 551–563, 2012.

[50] T. Chen, X. Pan, H. Liu, and T. Wu, "Rapid prototype and implementation of a high-throughput and flexible FFT ASIP based on LISA 2.0," in *Proc. IEEE International Symposium on Quality Electronic Design*, 2014, pp. 681–687.

[51] S. Liu and D. Liu, "A high-flexible low-latency memory-based FFT processor for 4G, WLAN, and future 5G," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 3, pp. 511–523, 2019.

[52] P. Salmela, A. Burian, H. Sorokin, and J. Takala, "Complex-valued QR decomposition implementation for MIMO receivers," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 1433–1436.

[53] U. Ahmad, A. Amin, M. Li, S. Pollin, L. Van Der Perre, and F. Catthoor, "Scalable block-based parallel lattice reduction algorithm for an SDR baseband processor," in *Proc. IEEE International Conference on Communications*, 2011, pp. 1–5.

[54] S. Shahabuddin, J. Janhunen, Z. Khan, M. Juntti, and A. Ghazi, "A customized lattice reduction multiprocessor for MIMO detection," in *Proc. IEEE International Symposium on Circuits and Systems*, 2015, pp. 2976–2979.

[55] S. Shahabuddin, O. Silvén, and M. Juntti, "ASIP design for multiuser MIMO broadcast precoding," in *Proc. European Conference on Networks and Communications*, 2017, pp. 1–4.

[56] A. Pohekar, "ASIP design on behalf of hybrid beamforming in MIMO communication system," Master's thesis, University of Twente, 2019.

[57] R. Avez and S. Weiss, "A reconfigurable ASIP for 802.11 packet detection algorithm," in *Proc. IEEE International Conference on the Science of Electrical Engineering in Israel*, 2018, pp. 1–5.

[58] R. Wittig, S. A. Damjancevic, E. Matus, and G. P. Fettweis, "General multicarrier modulation hardware accelerator for the Internet of Things," in *Proc. IEEE Global Communications Conference*, 2019, pp. 1–6.

[59] K. Shahbazi, M. Eshghi, and R. F. Mirzaee, "Design and implementation of an ASIP-based cryptography processor for AES, IDEA, and MD5," *Engineering Science and Technology, an International Journal*, vol. 20, no. 4, pp. 1308–1317, 2017.

[60] C. Bachmann, A. Genser, J. Hulzink, M. Berekovic, and C. Steger, "A low-power ASIP for IEEE 802.15.4a ultra-wideband impulse radio baseband processing," in *Proc. 2009 Design, Automation and Test in Europe Conference Exhibition*, 2009, pp. 1614–1619.

[61] D. Liu, "Baseband ASIP design for SDR," *China Communications*, vol. 12, no. 7, pp. 60–72, 2015.

[62] S. Saponara, L. Fanucci, S. Marsi, G. Ramponi, D. Kammler, and E. M. Witte, "Application-specific instruction-set processor for retinex-like image and video processing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 7, pp. 596–600, 2007.

[63] H. Qi, Q. Huang, and W. Gao, "A low-cost very large scale integration architecture for multistandard inverse transform," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 7, pp. 551–555, 2010.

[64] M. Hamdy, O. A. Nasr, and A. F. Shalash, "ASIP design of a reconfigurable channel estimator for OFDM systems," in *Proc. International Conference on Microelectronics*, 2011, pp. 1–5.

[65] Y. Yokota, S. Yoshizawa, and H. Ochi, "ASIP implementation of CDI feedback and low complexity precoding for MU-MIMO system," in *Proc. International Conference on Communications, Management and Telecommunications*, 2015, pp. 88–93.

[66] M. Shafi *et al.*, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.

[67] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*. London, UK: Academic Press, 2018.

[68] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The new 5G radio access technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.

[69] 3GPP TS 38.211, *NR; Physical channels and modulation*, v16.1.0 (2020-03), Release 16.

[70] W. Qiao, D. Liu, and S. Liu, "QFEC ASIP: A flexible quad-mode FEC ASIP for polar, LDPC, turbo, and convolutional code decoding," *IEEE Access*, vol. 6, pp. 72 189–72 200, 2018.

[71] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019.

[72] S. Shahabuddin, M. Juntti, and C. Studer, "ADMM-based infinity norm detection for large MU-MIMO: Algorithm and VLSI architecture," May 2017, pp. 1–4.

[73] S. Shahabuddin, M. S. Islam, M. S. Shahabuddin, M. A. Albreem, and M. Juntti, "Matrix decomposition for massive MIMO detection," in *Proc. IEEE Nordic Circuits and Systems Conference*, Oct. 2020.

[74] S. Malkowsky, H. Prabhu, L. Liu, O. Edfors, and V. Öwall, "A programmable 16-lane SIMD ASIP for massive MIMO," in *Proc. IEEE International Symposium on Circuits and Systems*, 2019, pp. 1–5.

[75] A. A. Zaidi, R. Baldemair, V. Moles-Cases, N. He, K. Werner, and A. Cedergren, "OFDM numerology design for 5G New Radio to support IoT, eMBB, and MBSFN," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 78–83, 2018.

[76] F. Abinader, A. Marcano, K. Schober, R. Nurminen, T. Henttonen, H. Onozawa, and E. Virtej, "Impact of Bandwidth Part (BWP) switching on 5G NR system performance," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 161–166.

[77] A. De la Oliva, J. A. Hernandez, D. Larrabeiti, and A. Azcorra, "An overview of the CPRI specification and its application to C-RAN-based LTE scenarios," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 152–159, 2016.

[78] Y. R. Li, B. Gao, X. Zhang, and K. Huang, "Beam management in millimeter-wave communications for 5G and beyond," *IEEE Access*, vol. 8, pp. 13 282–13 293, 2020.

[79] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[80] A. Gupta and A. Pal, "Accelerating SVM on ultra low power ASIP for high throughput streaming applications," in *Proc. 28th International Conference on VLSI Design*, 2015, pp. 517–522.

[81] D. Jamma, O. Ahmed, S. Areibi, G. Grewal, and N. Molloy, "Design exploration of ASIP architectures for the K-Nearest Neighbor machine-learning algorithm," in *Proc. International Conference on Microelectronics*, 2016, pp. 57–60.

[82] S. Chidambaram, A. Riviello, J. M. PierreLanglois, and J. David, "Accelerating the inference phase in ternary convolutional neural networks using configurable processors," in *Proc. Conference on Design and Architectures for Signal and Image Processing*, 2018, pp. 94–99.

[83] J. Rust and S. Paul, "Design and implementation of a neurocomputing ASIP for environmental monitoring in WSN," in *Proc. IEEE International Conference on Electronics, Circuits, and Systems*, 2012, pp. 129–132.

[84] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality—What is next?: Five promising research directions for antenna arrays," *Digital Signal Processing*, vol. 94, pp. 3–20, 2019.

[85] H. Wymeersch, J. He, B. Denis, A. Clemente, and M. Juntti, "Radio localization and mapping with reconfigurable intelligent surfaces," *IEEE Veh. Technol. Mag.*, to be published.

[86] G. Interdonato, E. Björnson, H. Q. Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, 2019, Art. no. 197.