A Survey and Comparative Study of Embedded FPGA Architectures

Hajer SAIDI[†1,2,] Mariem TURKI[1], Abdulfattah OBEID[3], Zied MARRAKCHI[4] and Mohamed ABID[1,2]
hajer.saidi@enis.tn
[1] CES Laboratory, National Engineering School of Sfax, Sfax, Tunisia
[2] Digital Research Center of Sfax (CRNS), Sfax, Tunisia
[3] National Center for Electronics and Photonics Technology, King Abdul-Aziz City for Science and Technology (KACST), Riyadh, Saudi Arabia
[4] Mentor Graphics, Tunisia

Summary
In recent years, the modern embedded systems are in intense complexity growth, that is why the different constraints (power consumption, performance, area, and cost) become increasingly inevitable. In this context, re-configurable System on Chip (SoC) has been proposed to attain these goals and the higher flexibility. Many solutions are proposed to obtain the reconfigurability with the best compromise between different challenges. Thus, the embedded FPGA is one of the key solutions to ensure the reconfigurability into systems. In this survey, we describe a comprehensive overview of existing embedded FPGA, specific techniques and tools used for the development and the configuration of the core. Moreover, we present comparatives tables used as a guideline to select which architecture is adequate for the target application in terms of characteristics such as occupancy, area utilization and regularity.

Key words:
embedded FPGA, Architecture, Hardware, Re-configurable, IP

1. Introduction

Nowadays, embedded systems have an increasing interest in technology. For this reason, the requirements for high performance and low power consumption are becoming more and more important when designing modern embedded systems [1,2]. As shown in Figure 1, the re-programmable computing as CPU is considered very flexible since they can execute all tasks, but this solution is bad in terms of performance [3]. Conversely, ASICs improve performance [4], but they are optimized for a specific application. Considering these information, re-configurable systems/devices stand between re-programmable computing and ASICs [5,6].

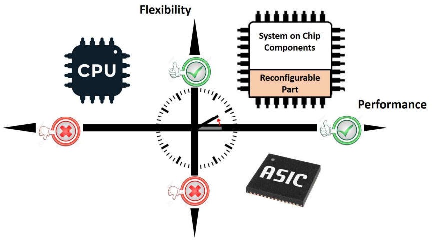


Fig. 1 Diagram of System's evolution

Embedded programmable logic devices (ePLDs) have a growing interest in the scientific and industrial communities thanks to innovations that have occurred during the last decades in the domains of microelectronics, Mechanical System Design (MSD), energy harvesting, WSN (Wireless Sensor Network), EPLDs can be applied in system-on-a-chip (SoC) designs [5]. In fact, making modifications to the hardware during both development and post manufacturing is the reason of increasing flexibility, widen their market, and curb soaring nonrecurring engineering (NRE) costs. In [1], authors studied the existing solutions to have a re-configurable SoC. They found that Embedded Field Programmable Gate Arrays (eFPGAs) offers computational advantages in terms of area and cost. An embedded FPGA (eFPGA) is an IP core that can be integrated into an ASIC or SoC to get the benefits of programmable logic with better latency, throughput, and power consumption. Quantity of look-up-tables (LUTs), registers, embedded memories and number of I/O ports can all be controlled when designing an eFPGA to make trade-offs between power and performance. Five years ago, eFPGA becomes available. Moreover, due to its flexibility and feature set, it invaded in the market very quickly. Now, it is commercially available from multiple vendors [7–11], on a wide range of foundries (TSMC, GlobalFoundries, SMIC and Samsung). In addition, eFPGA development is an interesting research field in many labs in the world (Sandia National Labs, DARPA, Boeing, Harvard, MorningCore China [12,13,13–17], Lip6 Paris and silicon valley research centers) [18]. These researches will be presented in section 3 with more details. eFPGA is also being used in a wide range of applications such as aerospace, communications, networking and most recently, artificial intelligence (AI), Internet of things, sensors networks, deep learning [19–22]. In last year's where eFPGA has been available, the industry and academic has proposed several generations of products. With each new generation, eFPGAs become more flexible and more usable for new applications, all driven by customer demand. In fact, the applications for eFPGA seem to be endless and in the future, this technology should be as pervasive as Arm processors are

Manuscript received August 5, 2020
Manuscript revised August 20, 2020

today [23]. The eFPGA IP has grown from simple and typical architecture which is programmed in Verilog with command line interfaces to earlier generation which is programmed by high level neural network model languages such as TensorFlow Lite and ONNX [19].

This research paper is a part of a research project called the Energy-Aware Re-configurable Node (EARN) project [24], which aims to design a low-power sensor node based on a system-on-chip for water pipeline monitoring applications. The system on chip includes a re-configurable part called "eFPGA". This paper will review the recent generations of eFPGA existing in literature [25,26] and industrial [8–10] In this survey, we analyze and emphasize the research key trends of embedded FPGA architecture, in order to guide the designer to select the most suitable IP for the designed application in terms of capacity, architectural structure, power consumption and latency.

The remainder of this paper is organized as follows. Next section presents the embedded FPGA characteristics. Next, the background used along with this research works in both academic and available industrial solutions are presented in section 3 and 4. Section 5 summarizes this study.

2. Embedded FPGA: Overview

Many factors such as the internal architecture and the IP implementation control the performance of the embedded FPGA. This section presents common concepts and definitions related to eFPGAs. The goal is to highlight the parameters of the performance optimization.

2.1 Embedded FPGA Versus FPGA

The increasing density of current programmable circuits, particularly FPGAs (Field Programmable Gate Arrays), allows the rapid prototyping of highly complex digital circuits. Also, it is possible to quickly test the performance of new architectural concepts: the complete implementation of a processor on FPGA circuits is now possible. Thus, the performance evaluation of software application is more than these offered by software simulators [27]. FPGA is defined as an integrated circuit including a network of programmable cells. Each cell can perform a function, chosen from several possible ones like shown in Fig.2-A. The interconnections are also reconfigurable. The most of FPGA components are mainly based on SRAM technology. The logic blocks in FPGAs are based on Look Up Tables (LUTs) to implement any Boolean function with 4, 5, 6 inputs with one or two outputs. The flexibility in FPGAs is mainly based on the interconnect network that allows any logic block to be connected to any other. This is also controlled by programming bits. Traditional FPGAs use 2D-mesh architectures that can require 10+ metal layers and take up much more area than the logic blocks themselves. In addition, the reconfigurability of some FPGA circuits opened new research Fields:

- Methodologies for designing re-configurable systems, with the ability to adapt new requirements environments or variable constraints.
- The possibility to be embed in ASIC systems and so considered as embedded FPGA [28].

Embedded FPGAs is an IP which is like the larger stand-alone FPGAs [29]. Its architecture is composed by arrays of computational logic blocks (LB) connected using programmable routing. EFPGA is equivalent to the core of an FPGA which can be integrated into the chip, just like others components: processors, DSPs and RAM ...[5,30,31]. An FPGA core is implemented with maximum metal layers and full-custom logic design to minimize area. However, an embedded FPGA needs to be designed differently: it must be modular to offer a wide range of sizes. The main use of FPGA is the prototyping but the eFPGA is useful more to be integrated into system on chip, which can make system more flexible by ensuring the reconfigurability in ASIC systems [32].

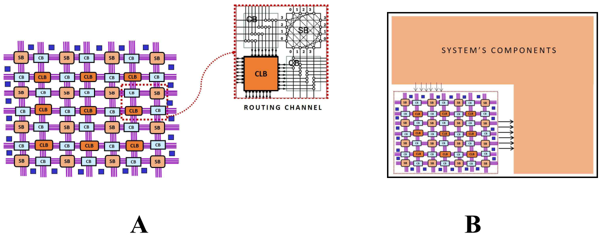


Fig. 2 FPGA VS eFPGA

2.2 Soft eFPGA Versus Hard eFPGA

Embedded Re-configurable FPGA can be provided as hardware cores (ASIC), defined as soft Intellectual Property (IP) also as an RTL design. This section tackles these two issues and describes each of them. Soft Re-configurable eFPGA is an IP which is described in behavioral RTL with hardware language [12,13,33]. Next, users implement this IP using CAD tools for synthesizing, mapping the RTL to standard cell library, placing and routing to generate the physical IC layout. This approach is better in flexibility than the hard solution, it is simple to use. However, design is constructed from CMOS logic gates like NORs and NANDs. Consequently, the implementation of this logic is poorly in soft IP, therefore overheads in area, delay, and power occur [33]. In the other side, Hardware eFPGA is a re-configurable hard block, which can be integrated into the design flow, but in this approach, designers cannot change the eFPGA core, because all the parameters are fixed from the beginning (speed, physical dimensions, power efficiency ...) [34]. The major advantage is that user don't need to re-design or build the programmable core, because all is ready to use, Full-custom design techniques maintain highly optimization of hard eFPGA core. However, vendors of hard IP typically offer a single eFPGA architecture that may not be ideal for a particular application.

2.3 Coarse Grain Versus Fine Grain

In this section, two families of eFPGA architecture are described: Fine Grain and Coarse Grain [35]. Like shown in Fig.3, fine-grain re-configurable architecture, means that embedded re-configurable logic architectures have homogeneous components. These architectures are characterized by their multi-function role in a system-on-a-chip. The coarse grain, architecture can include hardware blocks like (DSP, memory, adder) [36,37]. Coarse logic blocks simplify the implementation of complex functions [38]. These structures provide a much better area efficiency, and still being flexible enough for the target application domain. However, this higher flexibility can increase cost overhead. In the case of fine grain, it is easy to modify, and it has greater flexibility compared to coarse-grained.

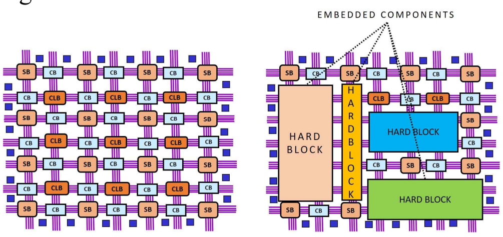


Fig. 3 Coarse grained Versus Fine Grained

In the present section, we detailed the concepts and the application of eFPGA. We presented the difference between the twice coarse/fine grained architectures, and soft/hard core IP. In the two next sections, we will present the existing solutions in both literature and industry.

3. EFPGA Solutions

3.1 Industry Examples:

Different works in industry will be presented in this section, to give a global overview on the existing works related to the eFPGA design.

- **Flex Logic:** Flex logic technologies is a company, which has introduced their innovative eFPGA core since 2015 [8]. The main objective is to update fixed functions in silicon, and to re-use the chip after fabrication. This company has their own hardware and software environments. Their eFPGA core called FLX 150; it has a meretricious architecture that goes up to (6*6, 7*7) lacks. It contains three global blocks: Boundary Less Radix Interconnect Network (RBB), Input / outputs and the logic Blocks. It has 20 Kbits RAM and 2.520 look-up-tables (6-input LUTs or dual 5-input LUTs). The total size is ranged between 2.520 LUTs to 122.500 LUTs. Flex logic proposed new patented architecture which uses only 5-7 metal layers. The architecture of the invented IP is presented in Table1. The EFLEX compiler is the software tool used to achieve the packing, placing, routing and generation of the configuration bit stream that will be loaded into the eFPGA core [11]. Flex logic can be integrated into SoC for many ways: RAM, Inputs/outputs, customize logic or co-processor. This IP can be used as: Software reconfiguration I/Os multiplexing, re-configurable accelerators, DSP accelerations and re-configurable cloud data centers.

- **Menta:** Since 2007, Menta has founded to deliver standard cells embedded FPGA IPs [39]. The main objective is to make the possibility to update the silicon post-production. Menta IP delivered its first version in 2009. Afterward four versions are delivered between 2013 and 2018 by Menta technology [40]. eFPGA IP core has a coarse-grained architecture as shown in Table 1. It contains embedded logic blocks (eLB) and embedded memory blocks (eMBs) and embedded application blocks (eAB) which can be inserted inside the array to increase performance for application-oriented design and to have high-speed arithmetic functions. Menta has their own tools to compile and configure the IP called Origami toolchains [41]; it includes RTL synthesis in HDL language (VHDL, System Verilog or Verilog) with additional support to SDC application design constraints. The Menta IP can be used in defense, aerospace, ADAS, data center systems and networking, to implement digital radio filters and cryptography algorithms.

- **Quick Logic:** Since 1988, Quick logic has founded. It started the researches to develop a programmable logic device, with lower power [42]. In 2016, this company announced their first version of eFPGA IP core, which is called ArcticPro. This IP is characterized by the ultra-low power. ArcticPro IP has fine grained architecture, it is an array ranging from

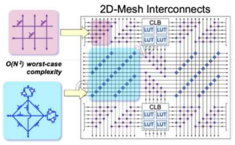
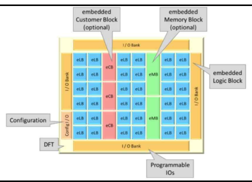
16x16 logic cells up to 64x64 logic cells. Like shown in Table 1, each logic cells contains Flexible Flip Flop, which can be driven by logic (4-input LUT or two independent 3-inout LUT). Using this architecture, the routing delay is reduced routing resource utilization in the IP core is improved. ArcticPro uses The Borealis eFPGA Compiler to easily evaluate and define a custom eFPGA logic cell array sizes and generate all the necessary design files for SoC integration (. cdl, v, .lib,. lef and .gds). ArcticPro also uses Aurora for Placement and Routing. The proposed eFPGA is used to control Sensor Hub, so that, the eFPGA is used to run sensor fusion at a very low power level.

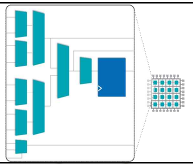
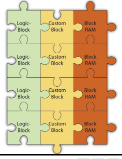
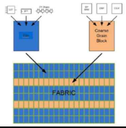
- **Achronix:** Since the middle of 2016, Speedcore eFPGA technology has been in production and shipping to end customers [10]. Speedcore core has a coarse grain architecture. It contains several blocks like the Re-configurable Logic Blocks (RLBs), RAMs Block (BRAMs), Logic RAMs (LRAMs) and Digital Signal Processing Blocks (DSP64s). These blocks are routing resources where the connection points between vertical and horizontal routing tracks are supported by component called Switch Boxes. The abstract view of the eFPGA is shown in Table 1. The size of Speedcore IP goes to 1.000 LUTs. Speedcore uses their own of the ACE design tools (Achronix CAD Environment) to configure the core by the required function. The IP can serve as hardware accelerator and re-configurable co-processor to implement a wide range of applications such as multimedia processing, cryptography [10].

- **Nano Explore:** NanoXplore is a very fast-growing company, which was developed in 2014 [9]. The eFPGA IP of NanoXplore is based on SRAM technology and contains programmable logic blocks, which has 4- inputs LUT. This IP offers an innovative solution to give flexibility in SoC and future ASIC. It is characterized by limited area. Its architecture is presented in Table 2 It has coarse grain blocks, it can support DSP and memory functions, LUTs, Carry chains. NanoXplore can deliver up to 750K LUT to Configure NX-eFPGA. And uses the mapping software called NanoXmap, to transform a synthesized RTL description into a bit stream downloadable in the NX-eFPGA through various hardware interfaces [43]. Thanks to mapping process used by NX, optimization in architecture has occurred, so a gain in the use of LUT appeared, which can reach 90%.

Table 1 presents a comparison between the industrial architectures, all the studied eFPGA are mesh based architectures, this is due to the regularity of this type of architecture. Indeed, when the number of CLB increases, regular rows and columns are added with the related connection blocks. In the other part, all the vendors proposed hard cores eFPGA to include in the system on chip design. This is could be benefice in terms of time to market and chip integration, but developers could not make modifications after the production process. Only Quick Logic proposed a fine-grained IP. All other eFPGA are coarse grained to add more flexibility in the application implementation. The Speed-core provides IP with a total size going up 2 M of LUTs. Finally, each IP is delivered with the corresponding CAD tools to implement the soft-core application into the hardware architecture.

Table 1 Industrial eFPGA Comparison

eFPGA	*Date*	*Architecture* *Mesh* *Hierarchical*	*Input-LUT*	*Soft/Hard* *Core*	*Coarse* *Fine* *Grain*	*Total Size (LUTs)*	*Company*	*CAD Tools*	*Domain*	*eFPGA Template*	*Delivered Generation*
Flex Logic [8]	**2015** **-** **2019**	**Mesh**	**6-input** **Or** **dual 5-input**	**Hard**	**Coarse**	**122,5**	**Flex Technology**	**EFIEX Compiler**	**Cloud Data Center**	 2D-Mesh Interconnects CLB LUT O(N²) worst-case complexity	**2**
Menta [39]	**2009** **-** **2018**	**Mesh**	**4-input**	**Hard**	**Coarse**	**200K**	**Menta Technology**	**Origami**	**Cryptography Radio filters**	 embedded Customer Block (optional) embedded Memory Block (optional) I / O Bank eLB eCB eMB embedded Logic Block Configuration Config I / O DFT Programmable IOs	**4**

Quick Logic [42]	**2016 - 2019**	**Mesh**	**4-input**	**Hard**	**Fine**	**Array sizes ranging from 8*8 to 64*64 logic cells**	**Quick Logic Technology**	**Aurora Software Tool Suite for eFPGA Design**	**Sensor hub, Hardware Accelerators, Machine Learning, Defense, Addressing, Security**		**2**
Speed core [10]	**2016 - 2019**	**Mesh**	**-**	**Hard**	**Coarse**	**From 5K to 2M**	**Achronix Semiconductor**	**ACE design tools**	**Cryptography, Multimedia Processing, Machine Learning, 5G Wireless, Networking Automotive**		**3**
NX-eFPGA [9]	**2014 - 2019**	**Mesh**	**4-input**	**Hard**	**Coarse**	**750 k**	**NanoXplore**	**NanoXmap**	**Space**		**3**

In this section we tried to address different emerging trends in industry for embedded FPGA. We presented some existing vendors of eFPGA IPs. We compared of the described solutions in terms of architecture, implementation, number of cells, target applications...

3.1 Academia Examples:

This section highlights the academic research related to eFPGA. The IPs are described in detail including the theoretical concepts and parameters developed to design the proposed eFPGA.

- **FASE eFPGA:** In [26], authors presented an automatic design flow for generating customized embedded FPGA fabric. The authors implement a matrix's FPGA embedded into SoC called "FASE". The basic elements of FASE eFPGA are: CLB (Configurable Logic Block) with 4-input LUT (Look Up Table), switches and configuration parts: Flip Flops + Shift register chain. Authors used VPR Tools [44] (placer and router) to generate the bit-stream file. They added a model to VPR tools for the generation of structured VHDL.

- **EFPGA Based on Multistage Switching Networks**: In [12,15] authors described a MSSN eFPGA. It is based on the utilization of Multistage network (MSSN). This topology is characterized by its routing infrastructures that is regular. The interconnection topology allows a hierarchical structure in order to reach connection between logic networks like shown in [45]. The general architecture is composed by: Inputs/Outputs, Logic Blocks (6:1 LUT, 2 LUT 4:2, multiplexers and FF), crossbars topology ensures the connections between LUT Blocks and the DSP Blocks. Typical size of MSSN eFPGA is 1 KLUT [15]. To ensure the configuration of an eFPGA core, authors used a complete CAD flow based on VPR [46], an academic open source tool for FPGA. For the routing step, authors have their own tool called MSSN-specific router.

- **ACLBs eFPGA:** Recently, in 2019, a patent [25] is delivered including new strategies in embedded FPGA fields. This IP is developed in Valbonne / Sophia Antipolis (FR). The architecture of the invented IP is presented in Table 2. It includes several Abuttable Configurable Logic Blocks (ACLBs) which are organized in island style. In this IP, each ACLB may be an instance of multiple programmable functional blocks including several ACLBs of the same or different types.

- **Overlay eFPGA:** The architecture conforms to a VPR template [46]. The template shown in Table 2 describes a regular array of cells (CLBs), where each on embeds several logic elements (BLEs). A BLE contains a Look-Up Table and a register. The cells are surrounded by routing channels, interconnected through switch boxes. The switch boxes can expose several connection schemes, but the Wilton [47] connectivity is most used. The CLBs can drive a portion of their close routing tracks, while some of the tracks can drive the CLBs inputs. The number of inputs for a CLB is usually less than the sum of inputs of its BLEs. Therefore, efficient resources sharing through a smart packing is a key issue [48].

- **EFPGAs based on a flexible architecture:** In [49], authors proposed an embedded FPGA based on a flexible architecture. Table 2 presents the architecture of the proposed IP. It has an island style composed by: LE (Logic Element) in a row and column, Routing Switch (RS), clusters and Connections Boxes (CB). The architecture template described above was formulated as a high-level description using MATLAB. Authors used exiting tools to achieve the synthesis, mapping and routing the target application (VPR), and Existing bit-stream generators like DAGGER [50] for the bit-stream generation.

- **ESL Re-programmable eFPGA:** In [16,51], authors proposed an eFPGA as soft IP. The general architecture is island style. The description architecture is written in VHDL. Since it is a soft IP there are no SRAM cells, Pass transistor or tri state buffer switches. The configuration element is a Flip Flop and switching element is a Multiplexer. The core is flexible since; the designer can select or choose all the fundamental parameters (LUT size, number of LUT, array size). However, this research work is done in close collaboration between startup Menta® technology and research center which makes more details not publicly available [17].

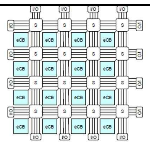
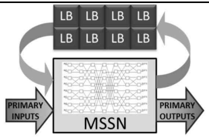
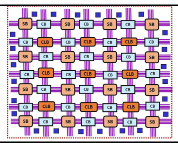
- **EFPGA Subsystem:** Recently, in 2019, researchers in [15] used Soft-core embedded-FPGA of [13]. Table 2 represents the eFPGA subsystem, composed of a frequency divider, a configuration loader, the PLD core and the interface toward APB, including the configuration registers which allow the processor to handle the subsystem.

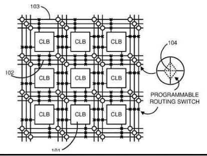
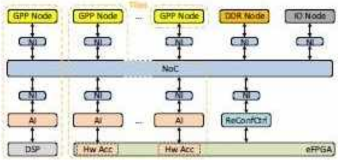
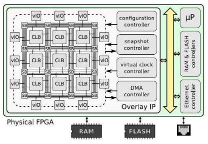
- **LIP6 eFPGA**: In [52], authors developed a traditional island-style programmable logic architecture, as presented in Table 2. The developed eFPGA contains the configurable logic blocks (CLBs) and bidirectional I/O pad resources which are connected together with routing networks. In [5], authors integrated the embedded IP into system on chip with multi fifo components in order to achieve the configuration, read and write data to test the target application.

- **FlexTiles:** In [53], authors proposed a hardware embedded FPGA. It has an island-style architecture. It has a typical architecture described with Verilog-To-Routing (VTR) framework. The FlexTiles platform uses an intermediary representation of the routing data called the Virtual Bit-Stream (VBS). Table 2 presents the architecture of FlexTiles Platform.

In Table 2, the academic eFPGA of the last decade are presented. In fact, we notice that the coarse grained architecture is appeared recently [14,25] . All the other works studied the fine-grained architecture due to its flexibility to implement the software application. The element component is the 4-inputs which is considered as the best solution in both mesh and hierarchical architectures. Most studied eFPGAs use the VPR tool to implement the software application. VPR offers a complete flow starting from the synthesis mapping and packing tools to the place and route tools. VPR can be adapted to target different eFPGA architectures. The performance of some eFPGA [5,15,52] is evaluated after integration into a system on chip. The integration aims to add flexibility and reconfigurability into the SoC even after production.

Table 2 Academia eFPGA Comparison

eFPGA	*Date*	*Architecture Mesh Hierarchical*	*Input- LUT*	*Soft Hard Core*	*Coarse Fine Grain*	*Total Size*	*Research Team*	*eFPGA Template*	*CAD Tools*
FASE	**2007**	**Mesh**	**4-input**	**Soft**	**Fine**	**-**	**GET Telecom Paris, CNRS – LTCI France**	**-**	**VPR**
ESL [51]	**2009 - 2011**	**Mesh**	**4-input**	**Soft**	**Fine**	**-**	**UM/CRNS France & Menta Company**		**Origami**
MSSN [12]	**2013 - 2019**	**Hierarchical**	**6-input or 2 4-input- 2**	**Soft**	**Fine**	**1 KLUTs**	**Bologna, Italy & STMicroelectronics**		**VPR**
LIP6 eFPGA [52]	**2013 - 2016**	**Mesh**	**4-input**	**Soft**	**Fine**	**-**	**LIP6- Paris France**		**VPR**

ACLBs [25]	**2019**	**Mesh**	**-**	**Soft**	**Coarse** **&** **Fine**	**-**	**Valbonne** **Sophia**		**Own** **Tool**
FlexTiles [53]	**2015**	**Mesh**	**-**	**Hard**	**Fine**	**-**	**Embedded Systems** **Ruhr-University** **Bochum, Germany**		**VPR**
Overlay [48]	**2018**	**Mesh**	**-**	**Soft- Hard**	**Fine**	**-**	**ENSTA Bretagne** **&** **Lab-STICC UMR**		**VPR**

In the present section, many researches are illustrated, presented and discussed. A comparative table is described in the last part of this section. In the last decade, many CAD tools and architectures are explored to obtain the best performance.

4. Conclusion

This paper presents a selective state-of-the-art review of embedded FPGA architecture and technologies. It classifies existing architectures related to the conception of eFPGA cores used technology from fine grain to coarse grain. These architectures have been detailed, discussed, evaluated and compared. Future trends and research directions have been enumerated. The aim of this paper is to present the different technologies which was not commonly expressed in previous review papers. Considering all the challenges and issues studied in this paper, new eFPGA architecture will be evaluated and explored. The target architecture should improve the logic occupancy and decrease the number of routing switches in order to ensure shorter critical path and better area. In this context, recent works study the possibility to design new architectures using new technologies such as Deep Learning frameworks: Tensor Flow.

Acknowledgments

The authors would like to thank Mentor Graphics Tunisia, the Digital Research center of Sfax and King Abdul-Aziz City for Science and Technology (KACST), which supports this work under a research grant (project no. 35/1012).

References

[1] Hsiung P-A, Santambrogio MD, Huang C-H. Reconfigurable System Design and Verification. CRC Press; 2018.
[2] Bueti S, Cranford Jr HC, Iadanza JA, Thiagarajan P, Ventrone ST. Clock distribution network, structure, and method for providing balanced loading in integrated circuit clock trees, 2009.
[3] Lai Y-H, Chi Y, Hu Y, Wang J, Yu CH, Zhou Y, et al. HeteroCL: A multi-paradigm programming infrastructure for software-defined reconfigurable computing. Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2019, p. 242–251.
[4] Razak H. High performance ASIC design: using synthesizable domino logic in an ASIC flow. Cambridge University, Cambridge 2008.
[5] Saidi H, Turki M, Marrakchi Z, Saleh MSB, Abid M. Embedded FPGA accelerator for Wireless Sensor Network nodes. 2016 11th International Design & Test Symposium (IDT), IEEE; 2016, p. 313–318.
[6] Loubach DS. A runtime reconfiguration design targeting avionics systems. 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), IEEE; 2016, p. 1–8.
[7] Embedded FPGA Under the Hood. Electronic Design 2018. https://www.electronicdesign.com/industrial-automation/article/21806004/embedded-fpga-under-the-hood (accessed August 17, 2020).
[8] https://flex-logix.com/ n.d. https://flex-logix.com/ (accessed August 17, 2020).
[9] NanoXplore n.d. https://www.nanoxplore.com/ (accessed August 17, 2020).
[10] Achronix Semiconductor Corporation | Achronix Semiconductor Corporation n.d. https://www.achronix.com/ (accessed August 17, 2020).

[11] eFPGA Archives Semiconductor Engineering n.d. https://semiengineering.com/tag/efpga/ (accessed August 17, 2020).

[12] Cuppini M, Scarselli EF, Mucci C. Design-space exploration of an eFPGA soft-core based on multi-stages switching networks. Proceedings of the 2013 9th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME), IEEE; 2013, p. 133–136.

[13] Cuppini M. Methodologies for Synthesizable Programmable Devices based on Multi-Stage Switching Networks. PhD Thesis. alma, 2015.

[14] Renzini F, Mucci C, Rossi D, Scarselli EF, Canegallo R. A Fully Programmable eFPGA-Augmented SoC for Smart Power Applications. IEEE Transactions on Circuits and Systems I: Regular Papers 2019;67:489–501.

[15] Renzini F, Cuppini M, Mucci C, Franchi Scarselli E, Canegallo R. Quantitative Analysis of Multistage Switching Networks for Embedded Programmable Devices. Electronics 2019;8:272.

[16] Ahmed SZ, Eydoux J, Rougé L, Cuelle J-B, Sassatelli G, Torres L. Exploration of power reduction and performance enhancement in LEON3 processor with ESL reprogrammable eFPGA in processor pipeline and as a co-processor. Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association; 2009, p. 184–189.

[17] Ahmed SZ. eFPGAs: Architectural Explorations, System Integration & a Visionary Industrial Survey of Programmable Technologies. Université Montpellier II-Sciences et Techniques du Languedoc, 2011.

[18] Yang Y, Huang Q, Wu B, Zhang T, Ma L, Gambardella G, et al. Synetgy: Algorithm-hardware co-design for convnet accelerators on embedded fpgas. Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2019, p. 23–32.

[19] Embedded Computing Design n.d. https://www.embedded-computing.com/ (accessed August 17, 2020).

[20] Wang J, Lou Q, Zhang X, Zhu C, Lin Y, Chen D. Design flow of accelerating hybrid extremely low bit-width neural network in embedded FPGA. 2018 28th International Conference on Field Programmable Logic and Applications (FPL), IEEE; 2018, p. 163–1636.

[21] Guo K, Sui L, Qiu J, Yu J, Wang J, Yao S, et al. Angel-Eye: A complete design flow for mapping CNN onto embedded FPGA. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 2017;37:35–47.

[22] Eldafrawy M, Boutros A, Yazdanshenas S, Betz V. FPGA Logic Block Architectures for Efficient Deep Learning Inference. ACM Transactions on Reconfigurable Technology and Systems (TRETS) 2020;13:1–34.

[23]
Src='https://Secure.gravatar.com/Avatar/366d6479d7f0270393041d47f6f23338?s=28 <img Alt='', #038;d=mm, Srcset='https://Secure.gravatar.com/Avatar/366d6479d7f0270393041d47f6f23338?s=56 #038;r=g', #038;d=mm, says #038;r=g 2x' class='avatar avatar-28 photo' height='28' width='28' /> Lydell Aaron. How To Integrate An Embedded FPGA. Semiconductor Engineering 2019. https://semiengineering.com/how-to-integrate-an-embedded-fpga/ (accessed August 17, 2020).

[24] Karray F, Garcia-Ortiz A, Jmal MW, Obeid AM, Abid M. Earnpipe: a testbed for smart water pipeline monitoring using wireless sensor network. Procedia Computer Science 2016;96:285–294.

[25] Martin G, Cavalli D, Firmin F. Embedded FPGA with multiple configurable flexible logic blocks instantiated and interconnected by abutment, 2019.

[26] Chaudhuri S, Danger J-L, Guilley S. Efficient modeling and floorplanning of embedded-FPGA fabric. 2007 International Conference on Field Programmable Logic and Applications, IEEE; 2007, p. 665–669.

[27] Kuon I, Tessier R, Rose J. FPGA architecture: Survey and challenges. Now Publishers Inc; 2008.

[28] Whatmough PN, Lee SK, Donato M, Hsueh H-C, Xi SL, Gupta U, et al. A 16nm 25mm 2 SoC with a 54.5 x Flexibility-Efficiency Range from Dual-Core Arm Cortex-A53 to eFPGA and Cache-Coherent Accelerators. 2019 Symposium on VLSI Circuits, IEEE; 2019, p. C34–C35.

[29] von Sydow T, Neumann B, Blume H, Noll TG. Quantitative analysis of embedded FPGA-architectures for arithmetic. IEEE 17th International Conference on Application-specific Systems, Architectures and Processors (ASAP'06), IEEE; 2006, p. 125–131.

[30] Nath APD, Ray S, Basak A, Bhunia S. System-on-chip security architecture and cad framework for hardware patch. 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE; 2018, p. 733–738.

[31] Saidi H, Turki M, Marrakchi Z, Obeid A, Abid M. Implementation of Reed Solomon Encoder on Low-Latency Embedded FPGA in Flexible SoC based on ARM Processor. 2020 International Wireless Communications and Mobile Computing (IWCMC), IEEE; 2020, p. 1347–1352.

[32] Kosan T, Talla J, Blahnik V. Efpga-based real-time hil simulator of induction motor drive. 2016 17th International Conference on Mechatronics-Mechatronika (ME), IEEE; 2016, p. 1–7.

[33] Aken'Ova V, Saleh R. A" soft++" eFPGA physical design approach with case studies in 180nm and 90nm. IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06), IEEE; 2006, p. 6–pp.

[34] Ferger M, Al Kadi M, Hübner M, Koedam M, Sinha S, Goossens K, et al. Hardware/software virtualization for the reconfigurable multicore platform. 2012 IEEE 15th International Conference on Computational Science and Engineering, IEEE; 2012, p. 341–344.

[35] Brant AD. Coarse and fine grain programmable overlay architectures for FPGAs. PhD Thesis. University of British Columbia, 2013.

[36] Srinivasan V, Govindarajan S, Vemuri R. Fine-grained and coarse-grained behavioral partitioning with effective utilization of memory and design space exploration for multi-FPGA architectures. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2001;9:140–158.

[37] Kumar TN, Almurib HA. Fine grain faults diagnosis of FPGA interconnect. Microprocessors and Microsystems 2013;37:33–40.

[38] Jain AK, Maskell DL, Fahmy SA. Are coarse-grained overlays ready for general purpose application acceleration on fpgas? 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE; 2016, p. 586–593.

[39] eFPGA IPs | Menta Efpga n.d. https://www.menta-efpga.com/efpga-ips (accessed August 17, 2020).

[40] Hall E. 33-3: Secure ASIC Architecture for Optimized Utilization of a Trusted Supply Chain for Common Architecture A&D Applications n.d.:4.

[41] Esteve E. How standard-cell based eFPGA IP can offer maximum safety, flexibility and TTM? Semiwiki n.d. https://semiwiki.com/efpga/7082-how-standard-cell-based-efpga-ip-can-offer-maximum-safety-flexibility-and-ttm/ (accessed August 17, 2020).

[42] QuickLogic - Provider of End-to-End Solutions for Endpoint AI n.d. https://www.quicklogic.com/ (accessed August 17, 2020).

[43] Lentaris G, Stratakos I, Stamoulias I, Maragos K, Soudris D, Lourakis M, et al. Project HIPNOS: Case study of high performance avionics for active debris removal in space. 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE; 2017, p. 350–355.

[44] Betz V, Rose J. VPR: A new packing, placement and routing tool for FPGA research. International Workshop on Field Programmable Logic and Applications, Springer; 1997, p. 213–222.

[45] Greenberg RI, Guan L. An improved analytical model for wormhole routed networks with application to butterfly fat-trees. Proceedings of the 1997 International Conference on Parallel Processing (Cat. No. 97TB100162), IEEE; 1997, p. 44–48.

[46] Rose J, Luu J, Yu CW, Densmore O, Goeders J, Somerville A, et al. The VTR project: architecture and CAD for FPGAs from verilog to routing. Proceedings of the ACM/SIGDA international symposium on Field Programmable Gate Arrays, ACM; 2012, p. 77–86.

[47] Masud MI, Wilton SJ. A new switch block for segmented FPGAs. International Workshop on Field Programmable Logic and Applications, Springer; 1999, p. 274–281.

[48] Le Lann J-C, Bollengier T, Najem M, Lagadec L. An Integrated toolchain for Overlay-centric System-on-chip. 2018 13th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), IEEE; 2018, p. 1–8.

[49] Neumann B, von Sydow T, Blume H, Noll TG. Design flow for embedded FPGAs based on a flexible architecture template. Proceedings of the conference on Design, automation and test in Europe, 2008, p. 56–61.

[50] Siozios K, Koutroumpezis G, Tatas K, Soudris D, Thanailakis A. DAGGER: A novel generic methodology for FPGA bitstream generation and its software tool implementation. 19th IEEE International Parallel and Distributed Processing Symposium, IEEE; 2005, p. 4–pp.

[51] Ahmed SZ, Fernandez M, Sassatelli G, Torres L. eFPGA architecture explorations: CAD & Silicon analysis of beyond 90nm technologies to investigate new dimensions of future innovations, 2009.

[52] Marrakchi Z, Mrabet H, Farooq U, Mehrez H. FPGA interconnect topologies exploration. International Journal of Reconfigurable Computing 2009;2009:6.

[53] Janßen B, Schwiegelshohn F, Koedam M, Duhem F, Masing L, Werner S, et al. Designing applications for heterogeneous many-core architectures with the FlexTiles Platform. 2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), IEEE; 2015, p. 254–261.