Annual Technical Volume

Computer Engineering Division Board 2020

Theme

Reconfigurable Computing and Machine Learning

The Institution of Engineers (India)

An Eco-system of Reconfigurable Architectures for Machine Learning

Sheetal Bhandari
Department of Electronics and Telecommunication Engineering
Pimpri Chinchwad College of Engineering, Pune, Maharashtra
✉ *sheetalubhandari@gmail.com*

ABSTRACT

Machine Learning (ML) algorithms, such as those used for image based search, face recognition, multi-category classification, and scene analysis are being developed that will fundamentally alter the way individuals and organizations live, work, and interact with each other. However, their computational complexity still challenges the state of the art computing platforms, especially when the application of interest is tightly constrained by the requirements of low power, high throughput, small latency, etc. Field Programmable Gate Array (FPGA) is an integrated circuit designed for lateral binding of application by a designer. Parallelism offered by thousands of Processing Elements (PEs), on-chip processor core and other resources available for digital designing makes FPGA a promising fabric for performance critical applications.

The goal of this paper is to present the potentials of FPGA for performance critical algorithms and design needs of ML. Also, to provide comprehensive details of software tools and hardware boards made available by the FPGA vendors with some discussion around implementations reported by researchers to motivate the AI community to use and experiment with FPGA.

Keywords: FPGA, SoPC, HW-SW Co-design, DPR, AI, ML.

INTRODUCTION

Artificial intelligence (AI) is evolving rapidly, with new neural network models, techniques and use cases are emerging regularly. While there is no single architecture that works best for all Machine Learning (ML) and Deep Learning (DL) applications, Field Programmable Gate Arrays (FPGAs) can offer distinct advantages over GPUs and other types of hardware in certain used cases [1].

In the field of electronics and computer engineering, hardware and software are two common approaches for implementing functionality. Hardware approach viz. Application Specific Integrated Circuits (ASIC), is an outcome of huge design and fabrication efforts as well as heavy Non Recurring Engineering (NRE) cost. It provides a solution with highly optimized resources for performing critical tasks. But it is permanently configured to only one application and cannot be changed at later stage. A software approach around General Purpose Processors (GPP) involves writing program to get the desired functionality. This approach provides the flexibility to change applications and perform a huge number of different tasks into one single chip.

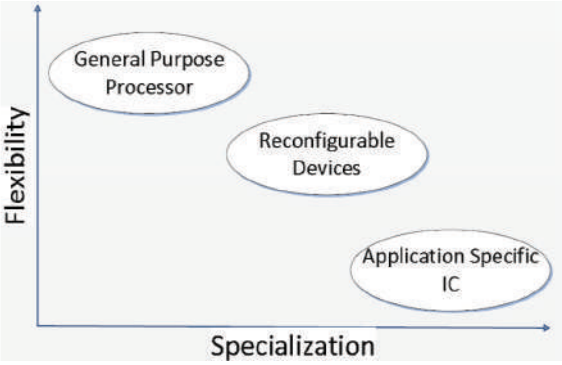

Figure 1: Domain of reconfigurable devices

When compared with ASIC, the cost of GPP based solution is less but it lacks in terms of performance, area efficiency and power dissipation.

FPGAs are revolutionary reconfigurable devices that blend benefits of ASIC and GPP. As shown in **Figure 1**, these devices provide better performance than GPP and higher flexibility than ASIC. Also, designing functionality on reconfigurable devices takes few days and less cost compared to ASIC design. The reconfigurable devices can be programmed and reprogrammed many times. Thus, reconfigurable devices have provided a trade-off between ASIC and GPP as it tries to achieve balance among cost, power, flexibility and design efforts [2].

Since their introduction in 1980, FPGAs has gained importance both in commercial as well as in research settings. Today, FPGAs are used for a wide sector of applications.

The computing requirements for ML pipeline have many different components, i.e., data pre-processing (data collection, cleaning, labelling, and maintenance), training, hyper parameter optimization, testing and validation, inference and deployment etc. It results into huge requirements of computations and data movements. Conventionally, Graphic Processing Units (GPUs) are used to cater theses requirements. FPGAs can deliver superior performance in ML applications, where low latency is critical and are therefore becoming widely adopted ML accelerators [3].

Contemporary FPGAs have many on-chip resources and features to support AI workloads with incredible accelerations. FPGA vendors have also created an eco-system of software tools and hardware boards to support such development to a great extent. Few researchers have also reported performance improvement while using FPGA for machine learning. This paper is organized as the following:

basic structure of FPGA and capacities of the state of the art of the device are given in Section-I. Available development platforms for implementing machine learning applications are summarized in Section-II. Section-III presents few success stories of researchers working in the domain of ML on FPGA.

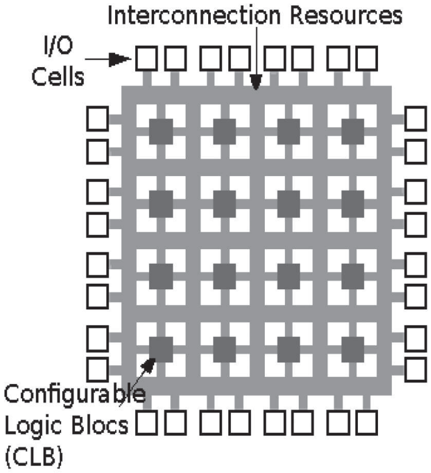

Figure 2: Basic structure reconfigurable devices

FIELD PROGRAMMABLE GATE ARRAY

FPGA includes two-dimensional array of general-purpose logic circuits as shown in **Figure 2**, whose functions are programmable. It can be programmed to implement virtually any set of functions. Input signals are processed by the programmed circuit to produce the desired set of outputs. Such input signals flow from user's system, through input buffers and through the circuit, and finally back out to the user's system via output buffers. The Configurable Logic Blocks (CLBs) in the gate array are connected together by routing resources to form a desired integrated circuit. The routing resources are connected to each other and to the logic elements in the gate array by programmable elements. Programming of the logic blocks, the routing network and the I/O cells can be selectively done to make the necessary interconnections that establish one configuration thereof to provide the desired system operation/function for a particular application [4].

Contemporary FPGAs contain thousands of logic blocks made-up of look-up tables and flip-flops and also a large variety of built-in digital components, such as, processors, memories, multipliers, transceivers and many more for implementing complex digital logic. Few powerful features and design techniques of state of the art FPGAs are given earlier literature [5, 6].

System on Programmable Chip (SoPC)

Now a days, FPGA embedded microprocessors are available along with reconfigurable array to form a complete System on Programmable Chip (SoPC) [7]. FPGA fabric can be paired with a GPP, this methodology is commonly referred as hardware-software (HW-SW) co-design. In this technique, most demanding section of an application is implemented into the FPGA fabric that accelerates the program execution. This technique is increasingly used in real time, high performance computing applications where computational kernels are performed on the FPGA while microprocessor continues to execute other program. Considerable speed up can be observed when applications or algorithms can make use of the massive parallelism offered by FPGAs [8].

Dynamic Partial Reconfiguration (DPR)

At present, FPGAs can support Dynamic Partial Reconfiguration (DPR). With DPR, a portion of the FPGA can be reconfigured, while the rest of the hardware mapped on FPGA continues to operate transparently with respect to the reconfiguration process. DPR offers countless benefits like adapting hardware algorithms during system runtime, share resources, reduced power consumption and shortening reconfiguration time. Advantages of using DPR feature in several application domains such as signal and image processing, graph algorithms, genetic algorithms, software defined radio, and adaptive control, video streaming and cryptography are being discussed in [9]. This feature is expected to provide benefits while implementing ML applications [10].

High Level Synthesis (HLS)

To enable designer fraternity cope with the advancements of FPGA devices, tools and techniques are also being upgraded. These techniques are expected to fully utilize the capacity and improve the productivity for FPGA. One such technique is the ability to write applications in High level languages (HLLs) rather than the traditional Hardware description languages (HDLs). High Level Synthesis (HLS) tools are able to target this problem. It accepts input as the algorithmic description of the application written in High level language like C/C++/System C etc. This algorithmic description is converted to the hardware description i.e. RTL netlist. A HLL description can typically be implemented faster and more concisely, reducing design effort and susceptibility to programmer error. In the recent years, HLS has made significant advances in both the breadth of HLS compatible input source code and quality of the output hardware designs. There are various HLS tools available, for e.g., academic tools like LEGUP, DWARV, BAMBU, etc. and commercial tools like SDK, Vivado HLS, etc [11].

Making use of above features available with the FPGAs, ML applications and AI workload is expected to obtain several advantages like great performance with high throughput and low latency, excellent value and cost, low power consumption and quick algorithm testing to get to the market fast.

DEVELOPMENT PLATFORMS

There are various manufacturers who provide FPGA development boards such as Altera (now part of Intel), Xilinx, Lattice Semiconductor, Atmel Corporation, Achronix, Microchips and Flex Logix etc. Intel and Xilinx are most widely used platforms. Both companies provide development board and Software Development

Kit (SDK) which support AI based applications as given in **Table 1**.

Table 1: ML Platforms by Leading FPGA Vendors

FPGA Vendor	**Software Development Kit (SDK)**	**Languages support**	**Hardware Development Boards**
Intel – (Former Altera)	Intel® Quartus ® Prime Design Suite	C, C++, VHDL, Verilog	Intel® Stratix® 10 GX FPGA L-Tile and Intel® Stratix® 10 GX FPGA H-Tile
Xilinx	Vitis Unified Software Platform 10 NX FPGAs	C, C++ or Python	Cloud: Xilinx Alveo™ cards U200, U250, and U50 Edge: Xilinx MPSoC evaluation boards ZCU102 and ZCU104
	Vivado Design Suite - HLx	C/C++ and IP-based design	Zynq UltraScale+ MPSoC Zynq-7000
	Xilinx ML Suite	C, C++, VHDL, Verilog	EDGE ZYNQ PYNQ. ZYNQ UltraSCALE+

Intel

The Intel® Stratix 10 NX FPGA delivers a unique combination of capabilities needed to implement customized hardware with integrated high-performance artificial intelligence (AI) [12]. These capabilities include High-performance AI tensor blocks, Abundant near-compute memory and High-bandwidth networking.

Tuned for AI arithmetic, the AI Tensor Block is estimated to provide up up to 15X more INT8 throughput than standard Intel Stratix 10 FPGA DSP block1 for high-compute density needed for high-throughput AI inference applications. Integrated memory stacks allow for large, persistent AI models to be stored on-chip, which results in lower latency with large memory bandwidth to prevent memory-bound performance challenges in large models. With up to 57.8 G PAM4 transceivers, Intel Stratix 10 NX FPGAs provide the scalability and the flexibility to implement multi-node AI inference solutions, reducing or eliminating bandwidth connectivity as a limiting factor in multi-node designs. The Intel Stratix 10 NX FPGA also incorporates hard Intellectual Property (IP) such as PCI Express* (PCIe*) Gen3 x16 and 10/25/100G Ethernet media access control (MAC)/physical coding sublayer (PCS)/forward error correction (FEC). These transceivers provide a scalable and flexible connectivity solution to adapt to market requirements.

These three sets of capabilities allow Intel Stratix 10 NX FPGAs to uniquely address the trend towards low latency and larger AI models requiring greater compute density, memory bandwidth, and scalability across multiple nodes as well as reconfigurable custom functions. The detailed guidelines to use this board through Quartus prime design suit is available as a user guide [13].

Xilinx

The Xilinx Machine Learning (ML) Suite users can easily integrate and deploy machine learning into applications. Pre-trained models are included to give users flexibility to quickly assess and choose the right model for their applications. Step by step tutorials and user guides are made available for easy use. It provides support for many common machine learning frameworks such as Caffe, MxNet and Tensorflow as well as Python and RESTful APIs [14]. The Vitis AI Library is a set of high-level libraries and APIs built for efficient AI inference with

Deep-Learning Processor Unit (DPU) available with FPGA. The Vitis AI Library provides an easy-to-use and unified interface by encapsulating many efficient and high-quality neural networks. This simplifies the use of deep-learning neural networks, even for users without knowledge of deep-learning or FPGAs. The Vitis AI Library allows users to focus more on the development of their applications, rather than the underlying hardware [15]. FPGA designers are given flexibility to develop algorithms or programs in software languages like C, C++, System C and Python, translate the code into a HDL, and dump onto advanced FPGA board without knowledge of VHDL or Verilog. High Level Design Suite (HLS) offers a new approach for ultra-high productivity with next generation C/C++ and IP-based design [16, 17].

Zynq-7000 devices are equipped with dual-core ARM Cortex-A9 processors integrated with

Artix-7 or Kintex-7 based programmable logic for excellent performance-per-watt and maximum design flexibility. With up to 6.6M logic cells and offered with transceivers ranging from 6.25Gb/s to 12.5Gb/s, Zynq-7000 devices can support highly differentiated designs. Zynq UltraScale+ MPSoC devices provide 64-bit processor scalability while combining real-time control with soft and hard engines for graphics, video, waveform, and packet processing [5].This very strong hardware and software support made available by FPGA vendors opens up opportunities for AI-ML engineers and practitioners to use FPGA as their development platform.

SUCCESSFUL IMPLEMENTATION

Few successful implementations are tabulated in **Table 2**. Earlier, literature presented hardware implementation of artificial intelligence algorithms [18]. Authors have reviewed around 169 papers published from the year 2009 to 2019 addressing the AI-ML implementation on hardware. Authors have identified that the implemented applications mainly from following categories of application like image processing, data mining, ambient intelligence, robotics, resources gauge model and global navigation. Out of these 169 applications, 122 are implemented on FPGA, 40 are implemented on GPU and 3 are implemented on ASIC respectively.

In order to bring completeness for the readers and appreciate recent efforts of researchers in implementing AI-ML algorithms on FPGA a short survey is presented here.

Some researchers thought that it was a futility using FPGA for CNN acceleration based on results studied after extensive review [19]. Techniques based on approximated multipliers, focusing on the parameters such as low-power, high throughput and limited hardware resources have been implemented [20]. Some literature discussed about the representation and classification methods for developing hardware for machine learning with the main focus on neural networks and mentioned that the use of advanced technologies such as Embedded Dynamic Random Access Memory (Edram) and Resistive Random Access Memory (ReRAM) for speed-up and to overcome conventional design problems [21]. Earlier, researchers developed the Low-latency Multi-Layer Perceptron (MLP) processor using field programmable gate arrays (FPGAs), and observed that the proposed FPGA design outperforms both CPU and GPU implementations, with an average speed up of 144x and 21x, respectively [22]. The challenges in deploying Deep Neural Network (DNN) on FPGAs during training phase were identified in earlier research on adopting FPGA for DNN computation to focus on on-chip resource usage, training efficiency, energy efficiency, and model accuracy [23].

Table 2: Successful Implementations

Sr. No.	Author Name	Algorithm	Platform
1	Ahmad Shawahna et. al. [19]	Deep learning	Virtex-5 based accelerator
2	SERGIO SPANÒ et.al [20]	Q-Learning algorithm	Xilinx Zynq UltrascaleC MPSoC ZCU106 Evaluation Kit
3	Pooja Jawandhiya et.al. [21]	Machine leaning	eDRAM and ReRAM
4	Ahmed Sanaullah et.al [22]	ANN	FPGA
5	Yudong Tao [23]	DNN	FPGA

CONCLUSION

AI on FPGA is a promising approach and it will play a crucial role in the development of AI-ML based applications. Using contemporary FPGAs is a way to place millions of PEs, equipped with internal processor, memory, within one device, will allow AI modality to be fully autonomous. As long as weights are stored inside the SoC, the network will consume much less power and works much faster. The features of state of the art FPGAs, software tool support and various design techniques provided by vendors is expected to open up opportunities for designers with varied skill sets like VHDL, Verilog, C, C++, etc. Many researchers are trying to exploit this eco-system for successful implementation and validation of results with respect to implementation accuracy, speed, power etc.

REFERENCES

1. Promwad, “Using FPGA in the Near Future: Trends and Predictions”, March 2020, https://hackernoon.com/using-fpga-in-the-near-future-trends-and-predictions-kr953yr9

2. Shuo Zhang, Yanxia Wu, Chaoguang Men, Hongtao He, Kai Liang, “Research on OpenCL optimization for FPGA deep learning application”, PLOS-ONE, October 10, 2019

3. Griffin Lacey, Graham W Taylor, Shawki Areibi (2016), “Deep Learning on FPGAs: Past, Present, and Future”

4. Michael Smith, “Application-Specific Integrated Circuits”, Addison-Wesley Professional; 1st edition (June 10, 1997)

5. Xilinx, “Zynq-7000 SoC Data Sheet: Overview”, DS190 (v1.11.1), July 2, 2018

6. https://www.intel.com/stratix10nx

7. https://www.intel.in/content/www/in/en/products/programmable/soc.html downloaded on 22/11/2020

8. https://www.intel.in/content/www/in/en/automotive/products/programmable/applications.html

9. Sheetal U. Bhandari, Shaila Subbaraman, Shashank Pujari, Rashmi Mahajan. 2009, “Real Time Video Processing on FPGA Using on the Fly Partial Reconfiguration”, Proceedings of the 2009 International Conference on Signal Processing Systems (ICSPS ‘09). IEEE Computer Society, USA, 244–247, DOI: https://doi.org/10.1109/ICSPS.2009.32

10. Jonny Shiton, Jon Fowler, Chris Chalmers, Sam Davis, Sam Gooch, “Implementing wavnet using Intel Stratix 10NX FPGA for real time speech synthesis”, Intel white paper https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/text-speech-synthesis-fpga.pdf

11. Razvan Nane, Vlad-Mihai Sima, Christian Pilato, Jongsok Choi, Blair Fort, Andrew Canis, Yu Ting Chen, Hsuan Hsiao, Stephen Brown, Fabrizio Ferrandi, Jason Anderson, Koen Bertels, “Survey and Evaluation of FPGA High Level Synthesis Tools”, IEEE transactions on Computer Aided Design of Integrated Circuits and Systems, vol. 35 No. 10, pp 1591-1604 Oct 2016

12. https://www.hpcwire.com/2020/06/18/intel-debuts-stratix-10-nx-fpgas-targeting-ai-workloads/

13. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-s10-fpga-devl-kit.pdf

14. https://www.xilinx.com/applications/megatrends/machine-learning.html

15. https://www.xilinx.com/support/documentation/sw_manuals/vitis_ai/1_0/ug1414-vitis-ai.pdf

16. https://www.xilinx.com/support/documentation/sw_manuals/ug1197-vivado-high-level-productivity.pdf

17. https://www.xilinx.com/support/documentation/sw_manuals/ug1197-vivado-high-level-productivity.pdf

18. Talib, M.A., Majzoub, S., Nasir, Q. et al., “A systematic literature review on hardware implementation of artificial intelligence algorithms”. J Supercomput (2020), https://doi.org/10.1007/s11227-020-03325-8

19. A. Shawahna, S. M. Sait, A. El-Maleh, “FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review”, IEEE Access, vol. 7, pp. 7823-7859, 2019, DOI: 10.1109/ACCESS.2018.2890150.

20. S. Spanò et al., “An Efficient Hardware Implementation of Reinforcement Learning: The Q-Learning Algorithm”, in IEEE Access, vol. 7, pp. 186340-186351, 2019, DOI: 10.1109/ACCESS.2019.2961174

21. Pooja Jawandhiya, “Hardware Design For Machine Learning”, International Journal of Artificial Intelligence and Applications (IJAIA), Vol.9, No.1, January 2018

22. Ahmed Sanaullah1, Chen Yang, Yuri Alexeev, Kazutomo Yoshii, Martin C. Herbordt, “Real-time data analysis for medical diagnosis using FPGA-accelerated neural networks”, Sanaullah BMC Bioinformatics 2018, 19(Suppl 18):490 https://doi.org/10.1186/s12859-018-2505-7

23. Y. Tao, R. Ma, M. Shyu and S. Chen, “Challenges in Energy-Efficient Deep Neural Network Training with FPGA”, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1602-1611, DOI: 10.1109/CVPRW50498.2020.00208.

ISBN 978-81-950662-8-5
Computer Engineering Division Board

The Institution of Engineers (India)
8 Gokhale Road, Kolkata 700 020
Phone : +91 (033) 40106264
Website : http://www.ieindia.org
e-mail : cpdb@ieindia.org