FPGA Implementation Strategies for Efficient Machine Learning Systems

Cristian Randieri[1], Valerio Francesco Puglisi[2]

[1]*eCampus University, Via Isimbardi, 10, Novedrate, 22060, Italy*
[2]*University of Catania, Dept. of Mathematics and Computer Science, Catania, Italy*

Abstract
The paper deals with the FPGA implementation of artificial neurons for Machine Learning systems. Nowadays Machine Learning is always more used in is different application fields and in many cases, FPGA implementations represent the best compromise between performance, and reduced power consumption. Modern FPGAs are equipped with specific circuits suitable for the implementation of the multiply and accumulate operation. These circuits called DSP blocks can be used for the implementation of the synapses of artificial neurons. However, the DSP blocks are not the only solution to implement this operation. This paper compares artificial neuron implementation considering DSP blocks based neurons and CLB-based ones. Comparisons are performed in terms of hardware resources, timing, and power consumption. Results show that DSP blocks based neurons are characterized by best performances in terms of power consumption and maximum frequency

Keywords
Machine Learning, FPGA, Neural Networks

1. Introduction

Machine Learning is a field of Artificial Intelligence based on statistical methods to improve performances of algorithms in data pattern identification [1, 2, 3]. Machine learning algorithms can be divided into three main categories: Supervised, Unsupervised, and Reinforcement Learning. The first two are characterized by training and inference phases. In Reinforcement Learning the training and inference phases are not separated. In the last decades, we assisted in an incredible spread of Machine Learning both in research and industry. The reasons are essentially two: the availability of data thanks to internet diffusion; The availability new circuits and devices optimized for Machine Learning applications [4, 5, 6, 7, 8]. These two reasons made possible the realization of Machine Learning systems that are always more used in different fields [9, 10, 11, 12, 13, 14]. Although in the first phase Machine Learning systems were mainly implemented in remote data centers, in the last few years there is an increasing diffusion of "Embedded Machine Learning". This paradigm involves the implementation of Machine-Learning systems inside objects, for example, cars, wearable devices, smartphones, etc. In this scenario, FPGAs play a crucial role thanks to their reconfigurability and high computing [15, 16, 17, 18, 19, 20]. Different from what happens for microprocessors and GPUs, FPGAs based design requires RTL design capabilities. Designers have always to consider aspects relative to the hardware design. These aspects require the engineers the capability to choose between different architectures and different hardware resources.

This aspect is especially true in the design of Machine Learning systems in which FPGA engineers must be able to identify the appropriate hardware resource for each operation [21]. This is the case of the FPGA implementation of artificial neurons that represent the basic element of artificial Neural Networks. This paper compares artificial neuron implementation of FPGA considering DSP blocks based neurons and CLB-based ones. Comparisons are performed in terms of hardware resources, timing, and power consumption.

2. Background

An artificial neuron is composed of several synapses that perform the multiplication between the inputs and precalculated weights (obtained during the training phase), a multi-input adder, and an activation function. The block diagram of an artificial neuron is shown in Fig. 1

The critical elements in terms of hardware complexity are the synapses and the nonlinear function. However, the nonlinear function can be simplified by replacing the traditional Sigmoid function with the Satlin one. In terms of equation the artificial neuron implements Eq.1, where ϕ is the activation function w_i are the weigths and finally x_i are the inputs.

$$y = \phi(w_i * x_i) \tag{1}$$

FPGA designers have essentially two possibilities for the weights implementation. The first one consists of the

ICYRIME 2022: International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Catania, August 26-29, 2022
cristian.randieri@unicampus.it (C. Randieri); valeriopuglisi@unict.it (V. F. Puglisi)

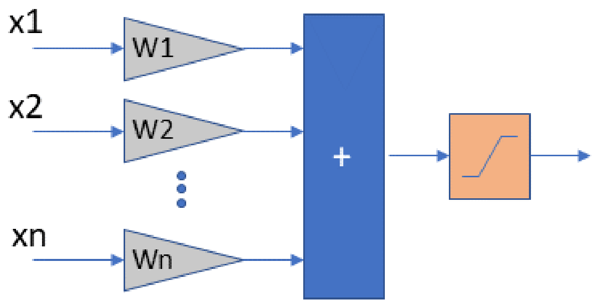


Figure 1: Artifical Neuron Block diagram composed of N synapses, a multi-input adder, and an activation function.

use of DSP Blocks while the second one consists of the use of the FPGA Logic Blocks.

2.1. DSP Blocks

FPGAs are nowadays used for digital signal processing (DSP) applications thanks to their capability to implement custom, fully parallel data-paths [22].

DSP applications make use of multipliers and accumulators that are best implemented in dedicated DSP slices.

Xilinx FPGAs for example have many dedicated, full-custom, low-power DSP slices, optimized for high speed, small size. During the design phase, designers can set the synthesizer to implement multiplications on these specific hardware resources. The number of DSP slices depends on the FPGA family. In general, more expensive FPGAs are equipped with a high number of DSP slices with respect to cheaper ones. In Fig. 2 is shown the block diagram of a XILINX DSP block provided in serie 7 FPGAs.

2.2. FPGA Logic Blocks

Another possibility for the implementation of multipliers consists in the use of the logic blocks contained in the FPGA slices (with reference to XILINX architecture) Logic blocks represent the basic elements of FPGA used for the implementation of the switching functions. In general a multiplier like any other digital circuit can be expressed in terms of switching functions. During the mapping phase of the design flow, the IDE used for the FPGA implementation maps the switching function in the logic blocks. The logic blocks are usually based on LUTs (Lookup Tables). In Fig 3 is show a simplified Logic Blocks.

It is composed of a LUT, a MUX, and a D Flip-Flop. The LUT is used for the implementation of the switching functions while the MUX manages the Flip-Flop that can be used to make sequential the switching function implemented in the LUT or as a single Flip Flop for the realization of fully sequential circuits.

This paper investigates the implementation of artificial neurons using DSP block and Logic blocks.

2.3. Methods

In order to perform the comparison, we code in VHDL an artificial neuron at RTL level. The number of synapses and the number of bits are application dependent. Consequently, we chose a specific setup for out experiment that is 3 inputs (that implies three synapses), and a 16 bit datapath.

For what concern the activation function, we replace the sigmoid function with the satlins. This considerably simplifies the hardware complexity of the circuit. In fact, the implementation of the sigmoid implies the use of LUT while the satlin function can be easily implemented using multiplexer and comparators.

Fig. 4 shows the synthesis options that give designer the possibility to choose how to synthetize the multipliers thougth the setting -maxdsp. There are three main possibility

- With -maxdsp =0 the synthetizer implement multipliers using hthe FPGA Logic Blokcs
- With -maxdsp = -1 the synthetizer choose autonoumosly the implementation strategy
- With -maxdsp = N with N>0 designer choose the maximum number of DSP blocks involved in the design. If for example a project requires 8 multiplies and -maxdsp = 2 then 2 multiplier will

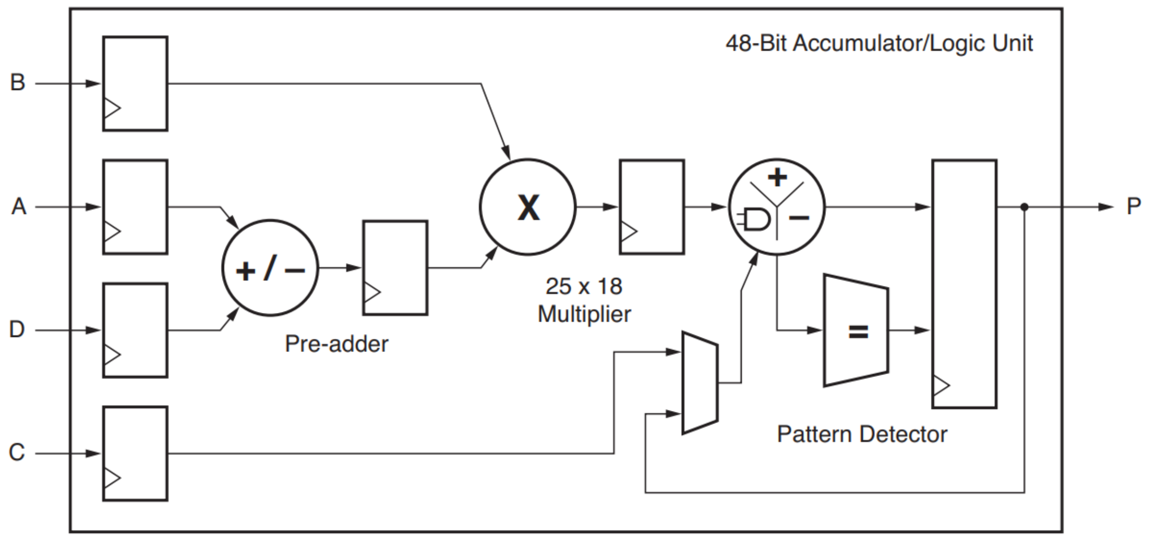


Figure 2: XILINX DSP block schematic.

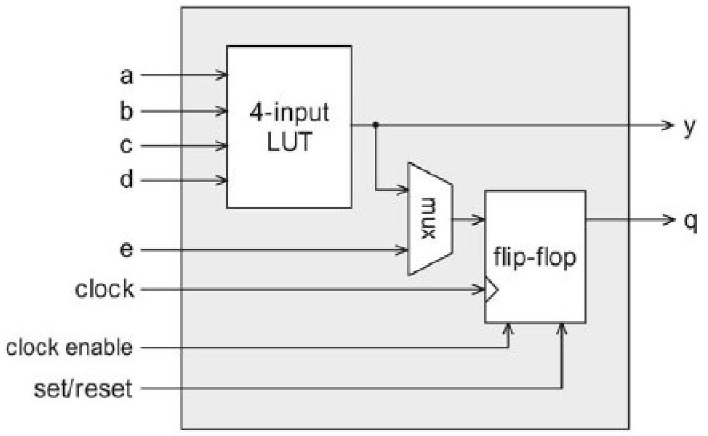


Figure 3: XILINX simplified Logic Block composed of a LUT, a MUX, and a D Flip-Flop

be implemented thougth DSP blocks while the other 6 will be implemented using Logic Blocks.

An alternative method to set this synthesis option consist in the writing o f a tcl command as follows:

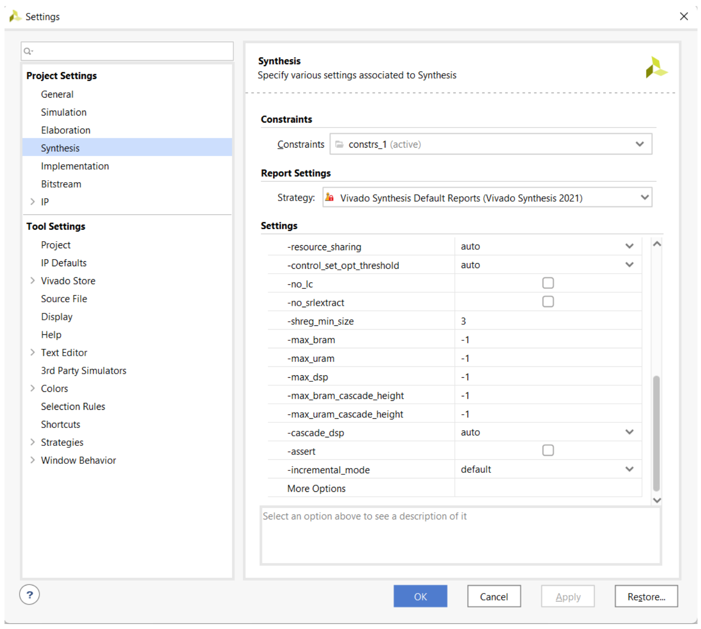
```
set_property
STEPS.SYNTH_DESIGN.ARGS.MAX_DSP 0
```

3. Experimental Results

The two architecture, the one using DSP blocks and the one implementing multipliers with Logic Blocks has been synthesized and implemented using the VIVADO toolchain on an kintex 7 FPGA device. In addition after the implementation has been performed a post-implementation test bench for two main reasons:

- Check the correct behavior of the artificial neouron
- Have an accurate power consumption estimation using the SAIF file.

The post-implementation simulation has been per-


Settings

Project Settings: General, Simulation, Elaboration, Synthesis, Implementation, Bitstream, IP

Tool Settings: Project, IP Defaults, Vivado Store, Source File, Display, Help, Text Editor, 3rd Party Simulators, Colors, Selection Rules, Shortcuts, Strategies, Window Behavior

Synthesis
Specify various settings associated to Synthesis

Constraints
Constraints: constrs_1 (active)

Report Settings
Strategy: Vivado Synthesis Default Reports (Vivado Synthesis 2021)

Settings

-resource_sharing	auto
-control_set_opt_threshold	auto
-no_lc	☐
-no_srlextract	☐
-shreg_min_size	3
-max_bram	-1
-max_uram	-1
-max_dsp	-1
-max_bram_cascade_height	-1
-max_uram_cascade_height	-1
-cascade_dsp	auto
-assert	☐
-incremental_mode	default
More Options	

Select an option above to see a description of it

Figure 4: XILINX VIVADO synthesis options. Thesetting max dsp allows designer to choose the implementation strategy for the multipliers

formed using the VIVADO simulator tool. Simulations have been performed by injecting random signals at the input of the DUT (the neuron). Input has been provided through a txt file. This simulation has been used not only to proof the correct behavior of the implemented system but also to generate the so-called SAIF file for the power estimation.

Switching Activity Interchange Format is an ASCII file that captures the switching activity in the design.

This file is needed for the Vivado power estimation tool. In fact, as it is known from the theory, CMOS circuits' power consumption is the composition of three main termsAmong these three terms the switching power represent the most important one an it depends on the switching activity of the circuit node [23].

it is defined in eq. 2,

$$P = \alpha C f V_{dd}^2 \tag{2}$$

Where a is the switching activity, C is the switching capacitance, f is the clock frequency and Vdd the supply voltage. Post implementation simulation allows to estimate the α parameter.

Power consumption results are shown in Tab1. DSP based neurons are characterized by a reduced power consumption. This feature is very important for embedded systems that are not directly connected to the power grid and are powered by batteries or energy harvesting sources.

Table 1
Power consumption (measured at 100Mhz)

Utilization	
DSP BLOKS	***LOGIC BLOCKS***
2.1 mW	3.5 mW

The maximum frequency reachable by the two architectures is shown in Tab. 2. Also in this case the best performance are obtained implementing the synapses using DSP blocks instead of Logic Blocks.

Table 2
Maximum frequency

Utilization	
DSP BLOKS	***LOGIC BLOCKS***
110 Mhz	89 Mhz

4. Conclusion

In this paper, we investigate the implementation of artificial neurons on FPGA. Modern FPGAs offer the possibility to implement multiplication on specific DSP blocks and this feature fits perfectly with the implementation of artificial neural networks. In fact, artificial neural networks are composed of artificial neurons requiring multiplications. Experiments has been performed to characterize artificial neurons implemented using with DSP block with artificial neurons implemented with Logic Blocks based multipliers Results show that DSP blocks based neurons are characterized by best performances in terms of power consumption and maximum frequency

References

[1] R. Avanzato, F. Beritelli, M. Russo, S. Russo, M. Vaccaro, Yolov3-based mask and face recognition algorithm for individual protection applications, in: CEUR Workshop Proceedings, volume 2768, 2020, pp. 41–45.

[2] G. Capizzi, G. Lo Sciuto, M. Woźniak, R. Damaševicius, A clustering based system for automated oil spill detection by satellite remote sensing, in: Artificial Intelligence and Soft Computing: 15th International Conference, ICAISC 2016, Zakopane, Poland, June 12-16, 2016, Proceedings, Part II 15, Springer, 2016, pp. 613–623.

[3] N. Brandizzi, V. Bianco, G. Castro, S. Russo, A. Wajda, Automatic rgb inference based on facial emotion recognition, in: CEUR Workshop Proceedings, volume 3092, 2021, pp. 66–74.

[4] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W.-m. W. Hwu, J. P. Strachan, K. Roy, et al., Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference, in: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 2019, pp. 715–731.

[5] R. Brociek, G. Magistris, F. Cardia, F. Coppa, S. Russo, Contagion prevention of covid-19 by means of touch detection for retail stores, in: CEUR Workshop Proceedings, volume 3092, 2021, pp. 89–94.

[6] S. Acciarito, A. Cristini, L. Di Nunzio, G. M. Khanal, G. Susi, An a vlsi driving circuit for memristor-based stdp, in: 2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME), IEEE, 2016, pp. 1–4.

[7] G. Lo Sciuto, G. Susi, G. Cammarata, G. Capizzi, A spiking neural network-based model for anaerobic digestion process, in: 2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), IEEE, 2016, pp. 996–1003.

[8] G. Capizzi, G. Lo Sciuto, C. Napoli, E. Tramontana, An advanced neural network based solution to enforce dispatch continuity in smart grids, Applied Soft Computing 62 (2018) 768–775.

[9] R. Giuliano, The next generation network in 2030: Applications, services, and enabling technologies, in: 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), IEEE, 2021, pp. 294–298.

[10] G. De Magistris, S. Russo, P. Roma, J. Starczewski, C. Napoli, An explainable fake news detector based on named entity recognition and stance classification applied to covid-19, Information (Switzerland) 13 (2022). doi:10.3390/info13030137.

[11] G. Capizzi, C. Napoli, S. Russo, M. Woźniak, Lessening stress and anxiety-related behaviors by means of ai-driven drones for aromatherapy, in: CEUR Workshop Proceedings, volume 2594, 2020, pp. 7–12.

[12] N. Dat, V. Ponzi, S. Russo, F. Vincelli, Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches, in: CEUR Workshop Proceedings, volume 3118, 2021, pp. 51–63.

[13] V. Ponzi, S. Russo, V. Bianco, C. Napoli, A. Wajda, Psychoeducative social robots for an healthier lifestyle using artificial intelligence: a case-study, in: CEUR Workshop Proceedings, volume 3118, 2021, pp. 26–33.

[14] R. Aureli, N. Brandizzi, G. Magistris, R. Brociek, A customized approach to anomalies detection by using autoencoders, in: CEUR Workshop Proceedings, volume 3092, 2021, pp. 53–59.

[15] H. Li, X. Fan, L. Jiao, W. Cao, X. Zhou, L. Wang, A high performance fpga-based accelerator for large-scale convolutional neural networks, in: 2016 26th International Conference on Field Programmable Logic and Applications (FPL), IEEE, 2016, pp. 1–9.

[16] G. Magistris, C. Rametta, G. Capizzi, C. Napoli, Fpga implementation of a parallel dds for wide-band applications, in: CEUR Workshop Proceedings, volume 3092, 2021, pp. 12–16.

[17] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions, Journal of big Data 8 (2021) 1–74.

[18] C. Ciancarelli, G. De Magistris, S. Cognetta, D. Appetito, C. Napoli, D. Nardi, A gan approach for anomaly detection in spacecraft telemetries, Lecture Notes in Networks and Systems 531 LNNS (2023) 393–402. doi:10.1007/978-3-031-18050-7_38.

[19] G. C. Cardarilli, L. Di Nunzio, R. Fazzolari,

M. Panella, M. Re, A. Rosato, S. Span, A parallel hardware implementation for 2-d hierarchical clustering based on fuzzy logic, IEEE Transactions on Circuits and Systems II: Express Briefs 68 (2020) 1428–1432.

[20] C. Napoli, G. De Magistris, C. Ciancarelli, F. Corallo, F. Russo, D. Nardi, Exploiting wavelet recurrent neural networks for satellite telemetry data modeling, prediction and control, Expert Systems with Applications 206 (2022). doi:10.1016/j.eswa.2022.117831.

[21] F. Silvestri, S. Acciarito, G. C. Cardarilli, G. M. Khanal, L. Di Nunzio, R. Fazzolari, M. Re, Fpga implementation of a low-power qrs extractor, in: Applications in Electronics Pervading Industry, Environment and Society: APPLEPIES 2017 6, Springer, 2019, pp. 9–15.

[22] Xilinx, 7 series dsp48e1 slice (2018).

[23] N. H. Weste, D. Harris, CMOS VLSI design: a circuits and systems perspective, Pearson Education India, 2015.