Mesopotamian journal of Computer Science
Vol. **(2021)**, 2021, **pp**. 13–17
DOI: https://doi.org/10.58496/MJCSC/2021/003 ; ISSN: 2958-6631
https://mesopotamian.press/journals/index.php/cs

Research Article

# Character Recognition By Implementing FPGA-Based Artificial Neural Network

Ahmed Hussein Ali [1],*, Mostafa Abdulghfoor Mohammed [2], Munef Abdullah Ahmed [3]

[1] *AL Salam University College Computer Science Department Baghdad, Iraq*

[2] *Great Imam University College, Baghdad, 10053, Iraq*

*3 Faculty of Al Hawija Technical institute, Northern Technical University, Mosel, 41002, Iraq*

**ARTICLE INFO**

Article History

Received 02 Nov 2020
Accepted 23 Dec 2020
Published 15 Jan 2021

Keywords

Neural Network

FPGA

OCR

ANN

VHDL

**ABSTRACT**

A non-linear applied math knowledge modelling tool, Artificial Neural Networks (ANN) are predominantly used to model complicated interactions between inputs and outputs or to look for patterns within the data. Using VHDL coding, we developed a generic hardware-based ANN. This classifier has been trained to recognize letters on a 4x4 binary grid that a user fills out using 16 toggle switches. An LCD shows the most likely classification that the ANN proposed. The ANN was taught to recognize 20 English character patterns and 9 Arabic character patterns on a 4x4 grid to showcase the viability of the FPGA execution of ANN. The Altera DE2 development and education board use the generated design file to flash the Altera Cyclone II FPGA. A training supervisor is also included in the design, which is responsible for training the ANN to recognize a set of 29 standard characters drawn from the English and Arabic alphabets. Positive patterns were found, with the ANN successfully recognizing all training characters. Finally, this study demonstrates the adaptability and boundless possibility of hardware-based implementation of ANN and the infinite possibility of prototyping and building recent systems with VHDL coding on an FPGA platform.

## 1. INTRODUCTION

With the advent of the extraordinary FPGA family of platforms, the manufacturing of electronic devices has entered an era of remarkable innovation. FPGAs are devices that have a full connectivity structure and a large number of building blocks like ports and logics (Figure 1). These interconnections can be programmed to link various logical blocks or slices to produce the required circuit [1]. These kinds of platforms represent the best possible answer for today's advanced digital infrastructures. In some cases, the parallel structure of a neural network (NN) can significantly improve its efficiency. Similarly, NN might be integrated into VLSI technology very quickly. The efficient application of a single neuron is critical to the complete success of every hardware-based neural network implementation. For NN hardware implementations, reconfigurable computing designs such as FPGAs are a great alternative. However, it is still challenging to implement high-neuron-count ANNs in FPGAs[2, 3].

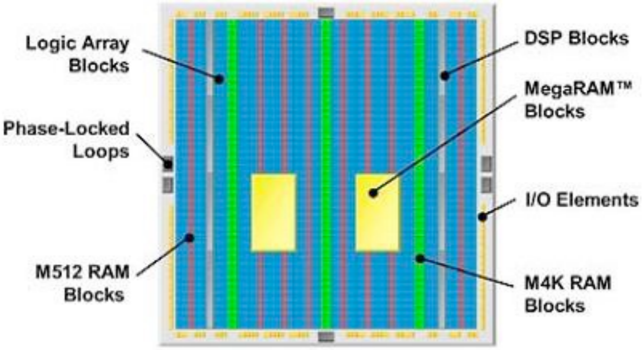

Fig.1. Altera FPGA Structure

*Corresponding author. Email: ahmedhusseinali@alsalam.edu.iq

The brain and other organic nervous systems serve as inspiration for the neural network (ANN) paradigm of data processing. This innovative data processing system architecture sets this paradigm apart from others. It is made up of multiple neurons; they are specially designed processing units that collaborate to tackle issues [4]. These neural interconnections are dense [5]. Like humans, artificial neural networks (ANNs) are learned from examples. An ANN can be trained to perform better at specific tasks like pattern recognition and data classification. Learning in biological systems necessitates remodeling the synapses that connect neurons [5]. All of what is mentioned also can be applied to ANNs. Since NN applications require a high number of computations, especially during the training phase, many efficient implementations of NN have been developed [6].

Optical character recognition, or OCR, is the process of transforming scanned images of handwriting, typed, or printed text into a machine-encoded form of that text. Data entry through paper sources, such as records, financial documents, mail, or any other printed records, is widespread. Commonly used for digitizing printed texts for use in electronic search, compact storage, online display, and machine operations, including machine translation, text-to-speech, and text mining, OCR is an integral part of the digitization process. OCR is studied in the context of pattern recognition, artificial intelligence, and computer vision.

Researchers have recently focused much on parallel processing technology like FPGAs. FPGA is programmable hardware on which practically any digital system can be constructed using HDL. In his paper, [7] proposed an ANN-based OCR. An NN is a massively parallel distributed processor built from simple processing units, and it is inherently successful at storing and making accessible experimental knowledge. Engineers typically utilize it for prototype designs before moving into ASIC because it is programmable, and numerous handy HDL languages have been developed. The construction of recognizing optical characters in this study was achieved on hardware employing an Altera Cyclone II EP2C35F672C6 FPGA chip. Based on this technique, the character recognition study has shown promising results, as demonstrated by the analysis results of the hardware.

## 2. PROBLEM STATEMENT

Although ANN has widespread use, most current implementations are software-based. Since the actual processing is done by the regular general-purpose processor of the computer system the ANN is installed on, this severely restricts the ANN's potential. Although ANN has been developed in hardware previously, there are significantly fewer hardware-based ANN implementations than software-based ones. This is because both the ANN and the resultant circuit are complicated when implemented in hardware. Constructing the ANN on a single chip is preferable from a practical standpoint. Furthermore, this caused another challenge, as the construction of an application-specific integrated circuit (ASIC) is exceedingly pricey.

Another problem with ANN hardware implementation with ASIC was the high expense of developing the complicated ANN circuit. In addition, ANN built on an application-specific ASIC cannot be used for various tasks. At this point, it becomes clear why it is vital to implement ANN on FPGAs. Historically, the FPGA's poor performance has kept it from being seriously considered as the hardware platform for implementing ANN. However, recent advancements in FPGA technologies have made ANN implementation on FPGA practical. The flexibility of FPGA makes it an ideal option for the small-scale development of ANN. The FPGA can be reconfigured in accordance with the ANN to be implemented, even though it has a much worse performance than ASIC.

## 3. OBJECTIVES

This study aims to validate the ANN method of character recognition using an FPGA-based hardware platform. This includes the ANN's purposeful training, the algorithms used to run the ANN, and the training and pedagogical guidelines used to teach the ANN to achieve the desired outcome. This study's primary goals are to develop an ANN based on an FPGA for use in a character recognition application and to be ready to acknowledge the type of characters needed to match the ANN's output with the actual outcome.

## 4. RESEARCH METHODOLOGY

The method started by a study phase, then progress to planning and development phase. After that, it will proceed into execution phase and is followed by error recording and correction. The result is then recorded and analyzed before being presented. The project should end when it produces the appropriate or acceptable results. The project will be able to put on keeping while it is concluded and the result data is presented.

There were multiple iterations of the design process for the embedded character recognition system. Many designs are realized on personal computers with simulators and pre-packaged software like MatLab and Java. Various feature extraction and categorization methods form the foundation of such systems. The most appropriate layout has been chosen and modified for embedded use. Iterations of the design, optimization and implementation phases are performed until the best possible performance has been achieved.

Although the benefits of a hardware implementation include improved performance, precision, and robustness, doing so requires a significant increase in design work and presents the drawback of being inflexible. More adaptability, reusability, and user-friendliness in the program architecture comes at the expense of speed. FPGAs enable concurrent hardware and software design, allowing users to reap the benefits of both methodologies simultaneously. The system's foundation consists of a microprocessor core and other hardware components. To make the processor and the rest of the system work together, a program must be written and loaded onto the computer. For this endeavor, we make use of the Altera DE2 board, which is powered by the Cyclone II EP2C35F672C6 FPGA chip.

In order to design with Altera CPLDs and FPGAs, the selected software is the Altera Quartus II Web Edition, Version 9.1 SP2. Altera Quartus II is a comprehensive development package. With version 9.1, Quartus II provides every instrument needed to create systems or prototypes using Altera's CPLDs and FPGAs. It's compatible with a wide variety of design notations and processes, including schematic diagram,block diagramss, Verilog HDL,state diagrams, and VHDL. The Quartus II development environment is being utilized for this initiative, and its requirements have been carefully studied. The machine includes the following features:

1. Microsoft Windows 7 Ultimate Operating System, OS version: 6.1.7601 Service pack1.
2. Altera Quartus II Web Edition Version: 9.1.
3. Intel Processor Core Duo T2450 @ 2.0 GHz and main memory of 2.5Gbyte.

The procedures involved in implementing a character recognition system are briefly described with reference to the flowchart shown in (Figure 2).

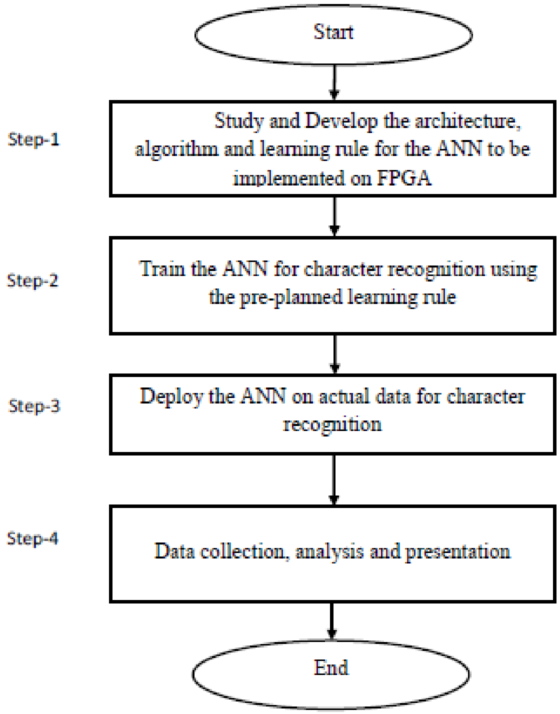
Start
Step-1
Study and Develop the architecture, algorithm and learning rule for the ANN to be implemented on FPGA
Step-2
Train the ANN for character recognition using the pre-planned learning rule
Step-3
Deploy the ANN on actual data for character recognition
Step-4
Data collection, analysis and presentation
End

Fig.2. Flowchart of Process

This section discussed a methodology of project, the projects is consist of two main part first one software and second hardware, and when gathering these two part is shown the project is involved several phases and steps starting from study and plan to ended of data collection and analysis and presentation phase to LCD, this system depended on the functional blocks representing the top-level architecture to the system to be developed on FPGA is also explain. Each functional block has its own architecture and behavior and responsible to do specific job in the system. Those functional block are connected together to produce desired behavior of the overall system for this project. The material in this chapter will be helpful for future project.

## 5. RESULTS AND DISCUSSION

Findings from a project utilizing an FPGA implementation of ANN for character recognition are reported here. We provide a comprehensive presentation and explanation of the system's synthesized top-level entity. After that, we compile the system's output and examine it for insights. Some discussions of systemic problems and overarching project goals are included.

The system's top-level entity displays the system's configuration, the functional blocks, and the links between them. To name a selection of the functional blocks:

- The ANN block (block ann:ann0) .
- ANN training supervisor block (pr:pr0).
- Pseudo-random number generator block (lfsr:lfsr0).
- Floating point processor block (float_alu:float_alu0).
- SRAM driver block (sram:sram0).
- LCD controller block (display_controller:display0).
- LCD driver block (LCDdriver:lcd0).

This project had produced favorable result. The system is capable to recognize 20 English character patterns and 9 Arabic character patterns (only 3 Arabic character patterns at a time) included in its training data set. During initialization, the system will enter initialization state before going to learning/training state where the ANN learn to recognize the total character patterns in the training data set. During this time the system will not be able to perform any recognition task until the ANN finishes the learning state. Users input the character patterns into the system using SW17 – SW2 toggle switches on-board the Altera DE2, and PB3 push-button switch is pressed to signal the ANN to perform recognition process of the input character pattern. The input pattern and recognition output is displayed on the on-board LCD module. Apart from some issues discussed earlier, this project is functional to recognize a total of 29 character patterns.

## 6. CONCLUSION

This project required utilizing many features of the Altera DE2 board. These include the LCD display, the switch inputs, LEDs, and the SRAM. The switches and LEDs were straightforward to use in the design. The SRAM and LCD required looking at datasheets to determine how the controlling state machines would need to be designed to meet the timing requires of the units . The many stages of the implementation allowed the experiments of different technologies to implement parts of the system, and to determine which worked the best.

The design shows the implementation of perceptron NN in FPGAs for the character recognition problem. The characters here are the English and Arabic language alphabets which are inputs to the network and after training, they are tested for recognition. These alphabets are represented in the form of a 4x4 matrix which means that there are 16 input neurons in the input layer of the NN, and getting outcomes of 20 English characters and 9 Arabic characters to be recognized by the network. The training input patterns and targets are converted to an appropriate form for the NN to process which in this case is the bipolar representation.

## 7. FUTURE WORK

The VHDL code for this project can be further optimized to obtain better result and performance. The resources usage such as multiplier can be further reduced to ensure that cost effectiveness. All of this issues, optimizations, and developments regarding this project are just a small fraction of what other tasks that can be done in the future.

## Conflicts Of Interest

None.

## Funding

The author's paper explicitly states that the research project did not receive any funding from institutions or sponsors.

## Acknowledgment

The author extends gratitude to the institution for fostering a collaborative atmosphere that enhanced the quality of this research.

## References

[1] P. Pelgrims, D. Driessens, and T. Tierens, "Embedded Systeemontwerp op basis van Soft-en Hardcore FPGA's," 2003.

[2] H. K. Ali, and E. Z. J. I. Mohammed, "Design artificial neural network using FPGA," vol. 10, no. 8, pp. 88, 2010.

[3] S. Jung, and S. J. I. T. o. I. E. su Kim, "Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems," vol. 54, no. 1, pp. 265-271, 2007.

[4] N. Shukla, A. J. J. o. E. S. Kumar, Computer Science, Engineering, and T. JECET, "Using Back-Propagation Recognition of Facial Expression," vol. 2, no. 1, pp. 39-45, 2013.

[5] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, and C. J. N. m. Peterson, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," vol. 7, no. 6, pp. 673-679, 2001.

[6] T. Nordström, B. J. J. o. p. Svensson, and d. computing, "Using and designing massively parallel computers for artificial neural networks," vol. 14, no. 3, pp. 260-285, 1992.

[7] S. J. I. J. o. A. R. i. C. E. Barve, and Technology, "Optical character recognition using artificial neural network," vol. 1, no. 4, pp. 131-133, 2012.