RESEARCH ARTICLE OPEN ACCESS

# Real Time Implementation of Nn-Based Aes

Fadhel. M. Almusailikh*, Mohamed. K. Shousha **

** Specialized Trainer (B), Electrical Engineering Department, ThePublic Authority for Applied Education and Training, Kuwait*

***Specialized Trainer (B),Electrical Engineering Department, The Public Authority for Applied Education and Training, Kuwait*

*Corresponding Author : Fadhel. M. Almusailikh*

**ABSTRACT**

In order to implement a neural network-based cryptosystem, a hardware platform with high speed and high efficiency is required, therefore the Field Programmable gated Array (FPGA) is said to be the most suitable choice in order to implement this system.

Due to the features of pipelining technique, the neural network was built by using it in order to minimize the connection between the neurons.

My proposed system shows the final hardware implementation of NN-based cryptosystem that is attained through programming the (Xilinx XCVirtex –E Family).

The data bus used is 128bits with a maximum clock frequency of (500MHz) and throughput of (4Gbps) per neural network.

The tool I have used in the implementation of the system was (Foundation ISE 3.1i).

**Keywords:** FPGA, Advanced Encryption Standard, Neural Networks Cryptosystem, Symmetric Block Ciphers**.**

---------------------------------------------------------------------------------------------------------------------------------------

Date Of Submission: 01-10-2019 Date Of Acceptance: 16-10-2019

---------------------------------------------------------------------------------------------------------------------------------------

## I. INTRODUCTION

In order to implement a neural network-based cryptosystem, a hardware platform with high speed and high efficiency is required, therefore the Field Programmable gated Array (FPGA) is said to be the most suitable choice in order to implement this system. Due to the features of pipelining technique, the neural network was built by using it in order to minimize the connection between the neurons.

My proposed system shows the final hardware implementation of NN-based cryptosystem that is attained through programming the (Xilinx XCVirtex –E Family). The data bus used is 128bits with a maximum clock frequency of (500MHz) and throughput of (4Gbps) per neural network.The tool I have used in the implementation of the system was (Foundation ISE 3.1i).

## 1. Basic Neuron Architecture and Implementation

The neurons are in charge of the performance of all calculations needed for the LMA. In addition, each of the feed-forward and update stages holds only the precise circuitry essential for that stage according to the ANN design. External RAM is used to hold the weight and target output data and error values. Each NN is required to have 4 inputs only, and the NN taught to work as a batch system, therefore, each NN replaced by 4 NN works in parallel, while each single one of them need to have one input only. In the end, the total number of NN would be 16, where all of them operates together in parallel, and every 4 has the same weight, in addition of having each 4 NNs connected to 1 update circuit in order to obtain LMA.

### 2.1 Activation Function

The activation function is employed by using the look-up tables method. A Lookup Table (LUT) is known to be defined as a simple storage device. Every output result is pre-computed and kept in the LUT. On the other hand, every input is considered as an address or reference number in the LUT, where it can be used to access the previously stored value that does not require any calculation. Hence, the process of acquiring data can be very quick and fast [1].

The LUT's storage is designed in way where it can produce up to 256 output values. An 8 bits address bus is used to represent our data within this range. Accordingly, a 256X8 ROM is required to operate the LUT of the activation function. This LUT should be in the bus within each of the feed-forward layers, in order that it may be given the f(net) for each single neuron of preceding layer to the input of following layer neurons. The activation function used in the ANN is nothing but the hyperbolic tangent tanh(x). The sigmoidal function tanh(x) can be expressed by the following formula:

O = tanh(x) = (enet – e-net) / (enet + e-net) ..(1)

As a result of computing the sigmoidal function that takes an input that ranges between [-1,1], that input which is taken from the data bus will have to be separated into 256 steps while the width of each single step is (0.0117)10. On the other hand, the output O is also multiplied by 256 in order to be with the range of [0,255].

Actually, the address bus of the ROM can be considered as the weighted input values of the data bus. Every value has to find-out its corresponding sigmoidal function magnitude from the LUT storage space [20]. The whole procedure of digging out the result of the activation function value O is shown in Figure1.

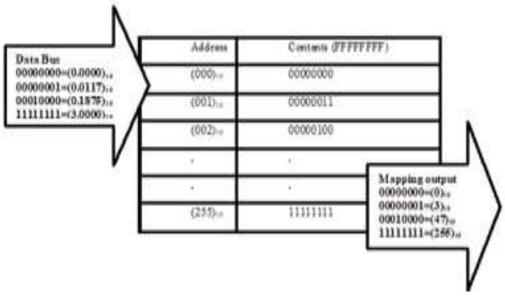

**Figure 1**. Scheme of the LUT (256X8bit) ROM

### 2.2 Multiplexed Interconnection

In order to limit the increase of interconnect and synaptic multipliers to NET(n), a time-multiplexed interconnection system is used in RRANN where n is the number of neurons enclosed by the NN. The main idea behind this structure is to link all output results of neurons on layer m withthe inputs of neurons on layer m+1 by using a bus multiplexer [1]. Every multiplexer within NN has something called output enable signal (EL),and it is used to enablethe neurons of that layer in order to put their output processing data to theneurons of the next layer.

A 6-bits counter is used in order to control the count of the enable signals to be able to count from (0-36) that can be a representation of the neurons in the chosen NN. Every single enable signal has a unique Boolean mathematical equation in order to represent the orders of the neurons linked to the multiplexer.

$E_1 = \mathrm{CNT}'2 \bullet \mathrm{CNT}'3 \bullet \mathrm{CNT}'4 \bullet \mathrm{CNT}'5$
…………………………………… (2)

$E_2 = ((\mathrm{CNT}2 + \mathrm{CNT}3) \bullet \mathrm{CNT}'4 \bullet \mathrm{CNT}'5) + (\mathrm{CNT}'2 \bullet \mathrm{CNT}'3 \bullet \mathrm{CNT}4)$……………………………..... (3

$E_3 = (\mathrm{CNT}'2 \bullet \mathrm{CNT}5) + (\mathrm{CNT}2 \bullet \mathrm{CNT}4) + (\mathrm{CNT}3 \bullet \mathrm{CNT}4 \bullet \mathrm{CNT}'5)$
…………………………………..… (4)

### 2.3 Feed-Forward Stage

The feed-forward stage’s job is to find the net values of every single node existing in the NN application. This requires broadcasting an input pattern over the network, one by one layer, until the net values for every output layer existing is done being calculated. This stage is also needed to be able to find the network’s output errors, where these errors can be used as an evaluation for the network’s output activations along with the chosen output pattern related to the supplied input pattern [2].

Figure 2 shows a block diagram of a feed-forward NN. It consists of 3 types of neurons with a total number of 37 neurons distributed on 4 layers. The neurons also get their weights from external RAM, so they can be used later in the update stages. Also there is a controller that operates the neurons of this stage, each one a time.

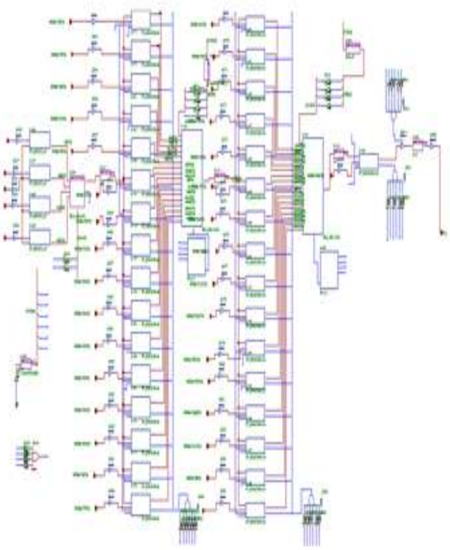

**Figure 2.** Block diagram of a feed-forward NN.

### 2.4 Update Stage

In order to adapt the weight, the update stage is used, where we can make the error is shown when the feed-forward stage is close to zero, the performance goal. The main idea behind the update stage is to make sure that the $\Delta W$ is calculated correctly. After the calculation process succeeds, then the result is used to be subtracted from the original weight. The learning algorithm to be used along this NN is the LMA method, where the update phase here can obtain the resulted output from the 4 NN then computes the error for every single output among them. The LMA method advises that in case of an error from the feed-forward stage, the learning rate (lr) should be presumed according to detailed implementation that can be as follows:

Do an update according to:

**wij(t+1) = wij(t) +** $\Delta W$ ………………....(5)

and then calculate $\Delta W$ as follows:

$\Delta W$ **= - E / lr**…………………….…. (6)

where: E is the sum of the sum of errors of fourparallel NNs.

Once the update produces the new output result that was computed in the feed-forward stage, assess the error at the new parameter vector.

In case that the error has raised as an outcome of the update, increment lr by a factor of 5, and then repeat the update process.

In case that the error has been minimized as an outcome of the update, decrement lr by a factor of 0.2 and then repeat the update process.

Hence, at the end of every update process, one training epoch is done.

## 3. The Contents of the Software Reports

There are multiple reports that exists including all the essential material about the simulation programs that can assist in carrying out the design in its final steps along the program.

1-The view synthesis report is specified by the symbol “syr”, which is considered as the main description of the of the program that has many parameters specified as follows:

I. The source parameters that contains the name of the program and the type of language.
II. Target parameters that has the output file name and format, the target technology and type of chip.
III. Source options that contains:

a) Entity name.
b) Finite State Machine (FSM) extraction.
c) FSM encoding algorithm.
d) FSM flip-flop type.
e) MUX extraction.
f) Resource sharing.
g) Complex clocks enable extraction.
h) Multiplier style.
i) ROM extraction. RAM extraction.
j) RAM style.
k) MUX style.
l) Decoder extraction.
m) Priority encoder extraction.
n) Shift registers extraction.
o) Logical shifter extraction.
p) XOR collapsing.
q) Register balancing.

iv. Target options that contains:

a) Additional 10 buffers.
b) Equivalent register removal.
c) Additional generic clock buffer (BUFG).
d) Global maximum fanout.
e) Register duplication.
f) Move first flipflop stage.
g) Move last flipflop stage
h) Slice packing
i) Pack JO registers into JOBs
j) Speed grade.

v. General options, which have the maximum approximation asynchronous delay in addition of having some details of that evaluation. This time is found to be a bit greater than the practical time delay, most of the time.

Also, “Bid” is the name that was given for the translation report that holds the warning and error messages that goes along with the three translation processes. Changing the EDIF netlist to be the Xilinx NGD netlist format is what these processes does. In addition, the timing specification and the logical design rule check, on the other hand, and report the following if found:Missing or untranslatable hierarchical blocks.

Invalid or incomplete timing constraints and removed logic summary.

Output contention, loadless outputs and sourceless inputs.

## 4. The Contents of the HardwareReports

In order to give a broad understanding of the implemented system, the hardware reports are descriptive for the second step of the design. There are eight reports in total, which were shown earlier in Figure 1. The map report is called “mrp”, and it has the warning and error messages specifying logic optimization and difficulties in mapping logic to physical resources.

In more details, the report contains the following information:

Errors-Warnings-Informational-RemovedLogic Summary-Removed Logic-lOB Properties-RPMs Guide Report-Area Group Summary-Modular Design Summary

As for the second hardware that was named “par”, mostly contains the place and route reports. These reports provide the following information:

1. The number of external lOBs.
2. The number of LOCed (not connected in the internally circuit) external lOBs.
3. The number of slices that are required to implement the design.
4. The overall placer score which measures the “goodness” of the placement, where that goodness score depends on the number of internal intersection nodes (Note that the lower score gives the better results.)
5. The timing summary at the end of the report details the design’s asynchronous delays, and most importantly the one that is named “dly”. This report has about 6000 pages that contains every single net in the design, and more importantly the delay of all loads on that net. The 20h1 highest net delays are listed at the top of the report.

Moreover, “lbs” was the name given for the IBIS model that belongs to the Virtex-E report. It contains the main explanation of the chip pins that belongs to the ICs, which are used in the employment of the system with every detail of the pins. This report contains the following points for each pin:

1- The job of the input/output pins and their connections.
2- The delay to input or output data in each pin.
3- All the currents in the main connections of each I/O pin with typically, the maximum and the minimum estimated current.
4- All the voltages of the point of the I/O port in each connection.
5- The input resistance value.
6- The input capacitor value.

The originator of the multi-pass place and route report named MPPR, gives the design scores and timing scores of the design in details, after inspecting every hardware report that contains an error or missing among them.

Another report is named "Ick", which delivers the information on the constraint conflicts, and that would be the back-annotate pin report. This happens, only if the pin of units and connections file (ucf) are discovered. With the aid of the provided design and those existing in the design, the ucf can do its job after the generation of the MPPR report.
During that, some conflicts may occur, and they are listed as follow:

1. The multiple pins could be constrained on the same net.
2. The same pin could have multiple nets.

In the end, we have the programming file generation report which has the name of "bgn", and it is mainly used to have the summary of bit generation options for the loading chip. This report will be used in the last step in the implementation of the system.

**5. Summary of Reports in the Implementation on Virtex-E:**

The Map Report of NN-based AES:

1. Design Information is;
   I. Command Line: map -p xcv3200e-8-cg1156-o map.ncdann.ngdann.pcf.
   II. Target Device: xcv3200cgl 156.
   III. Mapped Date: Started On Tue Jan 17 20:52:44 2006
2. Design summary is;
   I. Number of Slices: 25,552 out of 32,44878%
   II. Number of Slice Flip Flops: 5936 out of 64,896 9%
   III. Number of bonded lOBs: 489 out of 804 60%
   IV. Number of 4 input LUTs: 3,978 out of 64,896 6%
3. The Asynchronous Delay Report of AES: It is attempted to summarize this report in two main data pieces making use of the most critical indicators in this report, as follows:

The total equivalent gate counts for design are: 41,760

1. The simulated maximum operational frequency is 500MHz.
**2.** The maximum asynchronous delay time is 2 ns.

## II. CONCLUSION

The design of NN-based cryptographic systems is a good idea for the creation of very complex cryptographic systems that require no cryptographic analyzer or cracker.

However, it is necessary to know the number of adaptive iterations and the final weights of the encryption and decryption systems to start the system. Applying more plaintext / encryption text to the NN-based encryption system can reduce the error rate to a maximum (these must be zero errors). This will ensure the necessary confusion and dissemination in the encrypted text of the AES network-based encryption system.

However, further research is needed to understand how to attack this NN cryptosystem and protect it from possible attacks. The article also shows that such a powerful security algorithm can be implemented using a fast and customizable FPGA system. However, this requires additional research to implement the same circuit in Altera FPGA devices to make a comparison between Xilinx and Altera. apparatus.

## REFERENCES

[1]. J. Hadley, " **The Performance Enhancement of a Run-Time Reconfigurable FPGA System Throught the Reconfiguration",** M.Sc. thesis, Department of Electrical and Computer Engineering, Brigham University, Nov 1995.http://citeseer.ist.psu.edu/context/309218.pdf

[2]. A. Scwarzbacher, A. Brasching, Th. Eahl, P. Comiskey, J. Foley, **"Optimization and Implementation of the Arctan Function for the Power Domain",**Dublin Institute of Technology, Trinity College, Dublin, Ireland, 1999.

Fadhel. M. Almusailikh" Real Time Implementation of Nn-Based Aes" International Journal of Engineering Research and Applications (IJERA), vol. 9, no. 10, 2019, pp 75-78