International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.029 ISSN: 2581-4621

Perceptron Algorithm and Its Verilog Design

Dr. P.Harish[1], K.Jyoshna[2], K. Dinesh[3], K.Himabindu[4], C.Goutham Reddy[5], J.Indu[6]
Associate Professor/ECE, B.Tech student/ECE,Annamacharya Institute of Technology & Sciences, Tirupati, India.

ABSTRACT
The basic perceptron algorithm is important in supervised machine learning using artificial neural networks (ANN) because of its simple structure. While it cannot solve non-linear problems like XOR, its simplicity allows for perceptron to be built on a hardware design. Researchers have proposed ANN accelerators with complex memory units and specific registers, but simpler perceptron designs using Verilog HDL have also been developed for high efficiency and defect tolerance. This study focuses on the simplest perceptron design and includes one core for learning and four memory units for storing training data. The results show that the simple perceptron design can replace the defect-tolerant registers and memory units with proximate floating-point simulation, resulting in a tiny scale accelerator. The accuracy rate on the test set achieved 98% and the total area cost is only 0.0078 $mm2$
Keywords—Perceptron, Verilog, hardware design, machine learning, ANN, CNN

INTRODUCTION

The field of artificial intelligence has made significant progress with the development of CNN (convolution neural network), especially in supervised machine learning. Software languages can run algorithms on the CPU, and hardware can be designed to work with the CPU to accelerate calculations. The design of hardware plays a decisive role in the performance of the AI application.
Despite the increasing complexity of machine learning algorithms, AI-specific accelerator designs have been accomplished, such as accelerating calculations on GPU or FPGA or proposing the ASIC implementation of the neural network. However, the accelerators' performance is always compromised by communication with memory via DMA. Therefore, a state-of-the-art AI chip was proposed in 2014, followed by many high-level AI-specific chips with better accuracy and more functions.
The hardware design of AI-specific chips or accelerators is complicated because they have to satisfy most algorithms, such as CNN and RNN. Nevertheless, most neural networks can be seen as improved versions of ANN (artificial neural network), and the fundamental ANN algorithm is perceptron. Thus, if the perceptron accelerator with high scalability and compatibility are available, it will be easier to implement high-level hardware.
Previously, ANN hardware design was commonplace, but Olivier Temam proposed a better design due to the deficiency of spatial storage on neurons. This work reports on a perceptron design with simpler memory and less dimensionality, using a sign activation function and a 2-input port, which is less complicated than previous designs that use DMA and 90 inputs.

METHODOLOGY

Implement floating computing in hardware

In perceptron calculation, floating-point numbers are commonly used. However, in Verilog, variables are represented using 1s and 0s, which do not support floating numbers. The data that needs to be stored includes wi, xi, and the label. One solution is to multiply all the stored data by 512, which makes formula (1) the same as the original. However, formula (2) will change because the (label-y) value will now multiply with other enlarged data. To address this issue, the label should be stored in its original form, while other data should be stored as multiples of 512. By doing so, the difference between label and y will be enlarged when calculating formula (3), and all the formulas will be enlarged 512 times simultaneously. Once the computation is complete, the new weights should be divided by 512, and the floating-point results become the final results.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.029 ISSN: 2581-4621

The simple memory unit design

The DMA is a common component in hardware systems. It provides precise control and can function independently to support the CPU. However, its complex structure can make hardware design difficult. Therefore, it is necessary to use a simpler memory unit that allows for easy read and write operations. Additional details are provided in Table I.

TABLE I. Details of memory unit

Ports	Bit width	I/O	Description
ena	1	I	Enable
w	1	I	Write(1) or Read(0)
Data-in	16	I	Input data
Data-out	16	O	Output data
clk	1	I	Clock
rst	1	I	Reset
addr	8	I	Address

The data stored in the memory unit are 16-bit, as presented in Table I. The maximum actual value is 128 since the number is multiplied by 512. This is sufficient as most perceptron 2-dimensional input data is smaller than this value. However, larger data can be zoomed out simultaneously since the goal of the perceptron is to find a classified interface. The memory unit is controlled by the "w" port for reading and writing data. If "w" is set to 1, data is written into the memory unit at a specific address. When "w" is set to 0, data is output from memory at the same address to a request. The core performs real-time output calculations for the weight data and one of the training data. The output is used to update the weights based on the difference between the output and the label. The new weight data is output to the mem-w unit, concluding the first update. The core will continue reading new weights and training data for updating in a cycle. This process continues until there is no more training data to input.

RESULT AND DISCUSSION

Accuracy Result

The accelerator's accuracy is tested by using a particular dataset for both training and testing. The dataset has 700 lines of data, where each line represents the attributes of one data point, including x1, x2, and label data. The data is split into two parts: the training set, which includes 500 data points, and the testing set, which includes 200 data points. After training, the updated weight data is output through a mem-w-data-out port in serial, and the final result can be viewed in *-data-out ports. The output wave graph is shown below, which only displays some critical ports. The output data is presented, and other enable signals or middle variables can be excluded to avoid confusion.

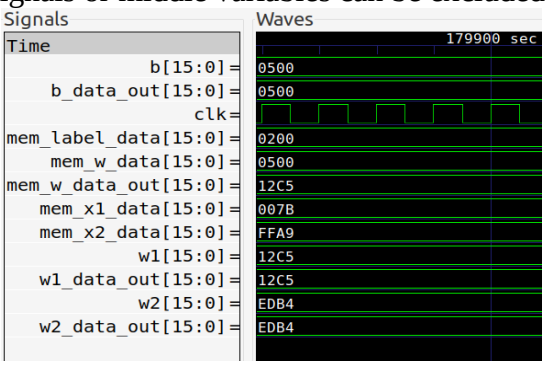

Figure 2: Part of the output wave graph

In the graph, the accuracy rate is 196/200, 98%; the words, compatibility is also the key factor in the

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.029 ISSN: 2581-4621

future design.

Synthesis Result

The Synopsys Design Compiler was used to synthesize the hardware design using a 25nm TSMC ASIC library to test for area and power cost. The synthesis began at 1.25 GHz and reached up to 1.9 GHz. The consumption results are summarized in Table IV. The table reveals that higher frequency leads to a larger area and power consumption. However, the total area is only 0.0078 $mm2$, which is merely 5.5% of a 4-way 32KB Cache. Due to its small scale and low power consumption, the hardware design is highly portable.

TABLE IV. Preliminary synthesis results

Frequency[GHz]	Area[mm^2]	Average power[mW]
1.25	0.0049	0.974
1.91	0.0078	1.798

precision is 92/92, 100%; the sensitivity is 92/96, 96%; the specificity is 104/104, 100%.

The top-level structure of the accelerator

The perceptron schematic is shown in *Fig. 1*.

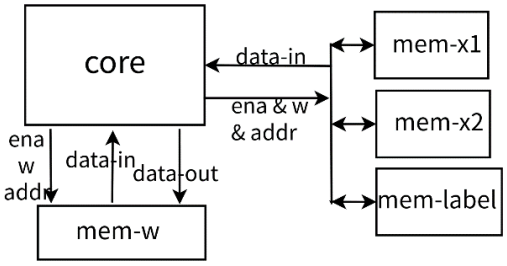

Figure 1: The top structure of the accelerator

The illustration displays that the perceptron comprises only five modules, which include one core module and four memory modules. The core module is responsible for reading and processing the data.

Discussion

To generate a dataset with two-category data and two-dimensional input, we used a Python script as such data is limited. Our dataset aimed to classify data based on the ideal classification interface of x1-x2=0. The interface in the transformation table is close to this ideal interface, resulting in a good performance of 96%-100%. However, the bias (b) value should be close to zero, but it turned out to be 2.5. As a result, the interface moved upward a little bit, causing the points labeled 0 to be predicted as 1, as shown in Table III. Another issue encountered in the simulation is the loss of overflowed data. The registers that hold the product of two 16-bit data should be at least 32-bit to prevent data loss due to overflow. Otherwise, the weight data cannot converge. It should be noted that these results were obtained from a specific dataset, and the accuracy may differ for other datasets. However, the robustness of the perceptron algorithm ensures that the performance remains good.

CONCLUSION

This paper presents a straightforward design for an accelerator to simulate a perceptron algorithm. While much research has focused on machine learning accelerators, it is important to also give attention to fundamental algorithms such as the perceptron. This design solves two main issues: performing floating-point calculations in hardware and utilizing simple memory units in the design.

DOI:10.46647/ijetms.2023.v07si01.029 ISSN: 2581-4621

By addressing these problems, a quick and uncomplicated accelerator has been created.

FUTURE WORK

Once the hardware design is complete, there are several additional tasks that need to be completed. First, the input port needs to be expanded to support multidimensional data, which requires an extension of the core module. Additionally, the memory unit can be enhanced with additional control ports and functions to ensure that all training data can be handled effectively. Finally, when the accelerator interacts with other standard chips, some ports may need to be reconfigured.

ACKNOWLEDGEMENTS

We express our gratitude to Li Sen for their assistance in preparing this paper. We also extend our thanks to the anonymous reviewers for their insightful feedback and suggestions.

REFERENCE

[1] Temam O. A defect-tolerant accelerator for emerging high-performance applications. Proceedings of the International Symposium on Computer Architecture, 2012, 40(3): 356-367.
[2] Holler M, Tam S, Castro H, and Benson R. An electrically trainable artificial neural network (ETANN) with 10240 "Floating Gate" synapses. IEEE Press, Piscataway, NJ, USA, 1990.
[3] Mauduit N, Duranton M, Gobert J, and Sirat JA. Lneuro 1.0: a piece of hardware lego for building neural network systems. IEEE Transactions on Neural Networks, 3(3):414-422, May 1992.
[4] Chen T, Du Z, Sun N, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine learning. Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems, 2014, 49(4): 269-284.
[5] Coates A, Huval B, Wang T, Wu DJ, and Ng AY. Deep learning with COTS HPC systems. Proceedings of the International Conference on Machine Learning, 2013.
[6] Chakradhar S, Sankaradas M, Jakkula V, and Cadambi S. A dynamically configurable coprocessor for convolutional neural networks. Proceedings of the International Symposium on Computer Architecture, page 247, Saint-Malo, France, June 2010. ACM Press.
[7] Liu D, Chen T, Liu S, et al. PuDianNao: A Polyvalent Machine Learning Accelerator. Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems, 2015, 43(1): 369-381.