NEURAL NETWORK ACCELERATOR IN FPGA CHIP

Marek Bohrn
Doctoral Degree Programme (2), FEEC BUT
E-mail: bohrn@phd.feec.vutbr.cz

Supervised by: Lukáš Fujcik
E-mail: fujcik@feec.vutbr.cz

ABSTRACT

This article describes implementation of multi core accelerator unit for artificial neural network computations into FPGA chip. The unit significantly speeds up computations required for realization of artificial neural networks. FPGA chips are selected because of their flexibility, high computing power and low price. The unit is built in FPGA language and the source code is optimized for Spartan-3 FPGA chip family which has enough resources for this purpose and provides highly parallelized computations. Many development boards are available for Spartan-3 chips, which provides easy development and use of the unit.

1. INTRODUCTION

Artificial neural networks are suitable to solve many different tasks such as pattern recognition, signal processing, classification, approximation of functions, prediction, compression of data and others.

Artificial neural networks' main advantage is, that they can be trained. Training is in most cases more effective than finding proper algorithm and programming. Optical character recognition can be presented as an example. There is no exact way to determine a character on the picture. The task can be split into many sub tasks like brightness balancing, filtering, contour detection, comparing, etc. All of these parts must be analyzed, converted to algorithms and interconnected into functional unit and then well tested. By utilization of artificial neural network, the task is simplified to training of the artificial neural network with proper patterns which illustrate possible shapes of letters for recognition.

2. THEORETICAL ANALYSIS

2.1 PLATFORMS FOR REALIZATION OF ARTIFICIAL NEURAL NETWORKS

There is a significant disadvantage in using the artificial neural networks. Their function requires a plenty of computing power. Most of calculations in artificial neurons are multiplying and adding. In some cases, these functions can be replaced by multiply-accumulate (MAC) function.

Usually the artificial neural networks are embedded into analog circuits, microprocessors, signal processors, computers and FPGAs. The analog circuits are not suitable to use because of their accuracy, reliability, complex composing and price. Usual microprocessors don't have enough memory and computing power to embed artificial neural networks for most tasks and they are not used widely. Most times the signal processors and computers are used for realization of artificial neural networks.

Signal processors have support for multiply-accumulate function and can perform it in one clock cycle. They are also easy to use in embedded applications, they can be programmed in C language and they are cheap solutions. Despite of these advantages, their computing power is limited due to their fixed structure and small number of computing cores (often only one or two).

The artificial neural network realization in computers is the simplest one; they have enough computing power and lots of memory. But their usage as embedded solutions is complicated and they are also expensive for this purpose.

The FPGA chips have perfect ratio between complexity, computing power, and price. Computations inside an FPGA chip can be effectively parallelized, their structure can be changed according to target application. Number of computation cores implemented inside FPGA can be much higher than in other solutions, typically up to a thousand.

For this project the FPGA chip family Spartan-3 from Xilinx was selected. These FPGAs contain hardware multipliers and memory blocks which are important for designing of a unit. The FPGA in this project is mounted into a development board that provides easy development and use of unit.

2.2 Artificial neural networks function

The artificial neuron is an elementary block of artificial neural network and the artificial neural network is formed by interconnecting the neurons. Artificial neuron consists of an input vector, synaptic weight vector, summation unit, and activation function (Figure 1).

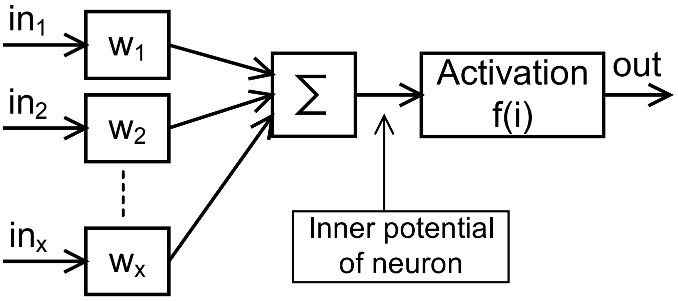

Figure 1: Artificial neuron

The neuron's function is following: the input vector is multiplied by weight vector and the results are summed in summation unit. Summation unit produces the inner potential of the neuron as shown in equation 1.

$$i = \sum_{n=1}^{x} w_n * in_n \qquad (1)$$

The inner potential is transferred into activation function. The activation function is, in most cases, step function, sigmoid or ramp function. Objective of the activation function is to saturate output into a defined interval, usually between 0 and 1.

The most used computations inside artificial neuron and artificial neural network are multiplying and adding. Spartan-3 FPGA chip used for realization of computation unit does not support multiply-accumulate function. Therefore the operations are computed in a pipe-lined circuit, which results in the latency of computation rise by one clock cycle, however, computation power of the unit remains the same.

Format of data inside the neural network is set to 18-bit fixed point. The integer part of number is represented by 6 most significant bits and fraction part by 12 least significant bits. It is enough for function of artificial neural network and respects internal 18-bit structure of multipliers inside the Spartan-3 FPGA chip.

The complete unit is optimized for feed-forward perceptron networks. It is also possible to implement different types of artificial neural networks inside the unit but their performance will be lower.

3. REALIZATION OF CIRCUIT

3.1 Complete circuit

Figure 2 shows a block diagram of complete accelerator unit. Main parts of the unit are control logic, computing blocks, activation function block and memory for storing results.

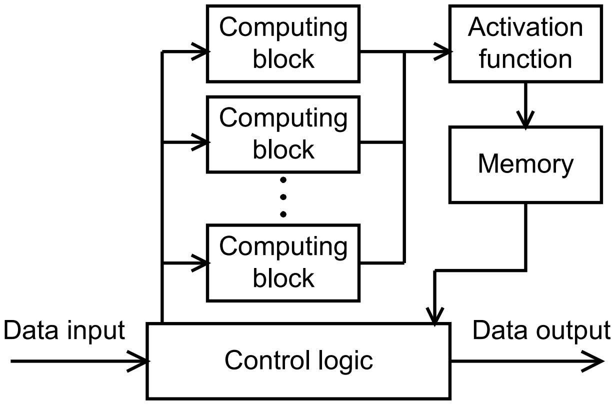

Figure 2: Block diagram of complete circuit

Control logic block contains information about map of implemented network. According to the map the control block is distributing data between input, computing blocks and output.

Inputs of computing blocks are interconnected into one bus. This arrangement allows for using less logic. The speed of artificial neural network is affected little because most times all the computing blocks require the same data on input.

There is only one bus for the outputs of computing blocks too. Data from computing blocks are transferred into activation function in a sequence according to the priority of the blocks. In this part of circuit, one bus is enough because data are stored most time inside the computing block and transferred only after the computation of the neuron is completed.

After activation function is performed, the result is stored in the memory. The memory contains actual value of each neuron result. Memory block is implemented in dual port RAM which allows for storing data from activation function and reading data by control logic simultaneously.

3.2 Computing block

Block diagram of computing block is shown in Figure 3. Input of computing block consists of data and weight value. The weight value is stored inside the block as a lookup table and control logic sends only pointer to required value. This arrangement reduces required logic and optimizes speed of the circuit.

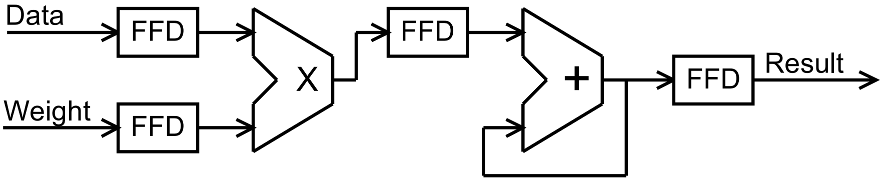

Figure 3: Block diagram of computing block

The computing block is arranged as a pipe-lined circuit. The first stage of pipe-line computes the product of data and weight value. The second stage contains adder and register which compute a sum of weighted inputs. After the computation of the neuron is done, the result is transferred into result register and waits for impulse to transfer into activation function via output bus.

Due to pipe-line arrangement of the computing block, the multiply-accumulate function is replaced by multiplying and adding and only latency of one clock cycle is added to the signal. The data throughput capacity and computing power is unchanged.

3.3 Activation function block

Figure 4 shows the activation function block as block diagram. The activation function block is arranged to a pipe-line which gives enough data throughput capacity for using only one activation function block for all outputs of computation blocks.

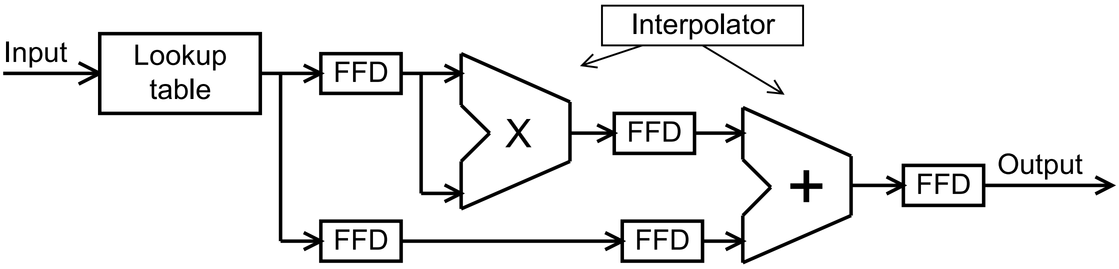

Figure 4: Block diagram of activation function block

Activation function is computed as an interpolation of the lookup table. The input value is transferred into the lookup table. Inside the lookup table there are two values stored. One value defines the level of interval and the second defines gradient inside the interval. At the first stage, gradient is multiplied by fraction part of the input value. Then, in the second stage, the multiplied value of the gradient is summed with level of interval that produces the result. The result is transferred to output stage and then to memory block where output values of neurons are stored.

3.4 Performance of the unit

The accelerator unit is programmed in VHDL language. This language is suitable for describing parallel digital circuits and pipe-line circuits. The code is optimized for use with Spartan-3 FPGA chip family.

Target chip XC3-S200 contains 12 blocks of RAM and 12 dedicated multipliers. This allows for creating accelerator unit with up to 10 computation blocks, one activation function and one memory block for results storage. Weight vectors are stored locally in computation blocks and can hold up to 10240 synapses. Maximum number of neurons implemented into the unit is 1024.

Source codes were optimized and final circuit can work with frequency up to 133 MHz. That gives 1330 millions of multiplications and summations in one second when using 10 computation blocks. That is enough for many applications of neural networks. The unit was tested with application that recognize hand written numbers and gives 336 000 recognized numbers per second.

4. CONCLUSIONS

Neural network computations require high computation power. Using FPGA chips, it is possible to create very effective computing accelerator for this purpose. Comparing to typical signal processors, it is possible to create very cost efficient solution with more than 10 times better performance.

Higher computing power can be achieved using different FPGA chips. For example latest Virtex-6 devices from Xilinx have more than 100 times more resources on chip than Spartan-3 and can run on much higher frequencies. That is ideal solution for neural networks computations.

REFERENCES

[1]Omonodi, A.; Rajapasake, J.: FPGA Implementations of Neural Networks, Netherlands, Springer 2006, ISBN 0-387-2845-0

[2]Fausett, L.: Fundamentals of Neural Networks, New Jersey, Prentice Hall 1994, ISBN 0-13-334186-0

[3]Xilinx company: Spartan-3 FPGA Family: Complete Data Sheet, Xilinx 2008, available at WWW: www.xilinx.com/support/documentation/data_sheets/ds099.pdf