electroscope

FPGA Implementation of Digital Cellular Neural Network

E. Raschman[1], D Ďuračková[1]

[1] Department of Microelektronics, Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava
Ilkovičova 3, 812 19 Bratislava
E-mail: emil.raschman@stuba.sk

Abstract:
This paper deals with hardware implementation of Digital CNN network in FPGA. We have implemented using the XILINX ISE development kit and the XC5VFX30T chip. The implementation in the FPGA has various advantages. The main advantage is the flexibility – simple modification of the neural network and relatively low cost since the chip is reprogrammable to almost any digital circuit of any given complexity. We have implemented several CNN networks and compared them with the hardware implementation achieved in our proposed CNN network presented in conference EDS 2008.

INTRODUCTION

The big part of electronic deals with processing and plotting various information. By one of most frequently information is image processing.. So how is increasing demand on quality so is increasing requirements on their processing. Circuits with quickest image processing are neural networks. Their calculation power is given by parallel processing of individual pixels of image. For image processing we used cellular neural networks. Network contains a lot of simple calculation elements (cells), which are working parallel. Basic principle of the CNN contains literatures [1], [2], [3], [4], [5] and [6]. Cellular neural network is usually non-linear cellular network. This is group of the spatially distribution cells, where every cell is neighbor to itself and is local connected with neighboring cells in some incidence markings *r* - neighborhood. Control input of CNN network is weight matrix, where each coefficient represented some weight (importance) correspondent to input. Then each input is multiplied with certain weight constant. Summarize this conjunction we get function ***Net.***

$$\boldsymbol{Net} = w_1 s_1 + w_2 s_2 + w_3 s_3 + \ldots + w_i s_i + \ldots + w_m s_m \qquad (1)$$

In this equation the coefficients *w* represent weights and coefficients *s* represent incoming signals from surrounding cells. Output of the cell *y* we get from non-linear transformation ***Net*** :

$$y = f\left(\boldsymbol{Net}\right) \qquad (2)$$

Function ***f()*** we call transfer or activation function. This function designates output state of cell. There exist some transfer functions [7] as for example sigmoid function, hard-limiter or threshold logic utilized in several applications.

Depended from properly choosing of weight matrix we can make that the CNN network is able for example noise removing. Choice of the matrix input conditions for CNN network, follows the network input data processing.

FPGA IMPLEMENTATION

The principle of the CNN networks is based on very simple principle similar to those of biological neuron. Network is consisting of quantity basic computing elements so-called cells. Incoming inputs of cell are multiplying by correspondent weight coefficient, adding and then making conversion through the transfer function. Because all cells realize information's processing in parallel, calculation power CNN network is direct proportional to the number of cells. The more cells contain network, the more information achieve synchronized processing. Therefore in design of the CNN network effort is focused to minimize cell size and thereby provide for maximum number of cells on chip. The base our new architecture CNN network are signals distributed in time, which are multiplication by AND gate.

The new digital architecture of CNN network

Size of the chip is one of biggest problems at designing CNN network. The most area of chip takes hardware multiplier unit, therefore we are searching other alternative multiplication.

The multiplication of the signal using the AND

How to alternate the possibility of multiplication is to design circuit, which multiplication input values and weight coefficient are by means of AND gate. The method of multiplication is based on the fact, that by multiplication the input value must be converted to the time signal and the weight value has to be special picked, so by the timing starts multiplication. We proposed a special coding for the weights.

We used a special system of 15parts, e.g. one cycle is divided into 15 parts. In such a time period is possible code 16 various values weight or inputs. Decomposition signal for corresponding weight values are displayed on the Fig. 1.

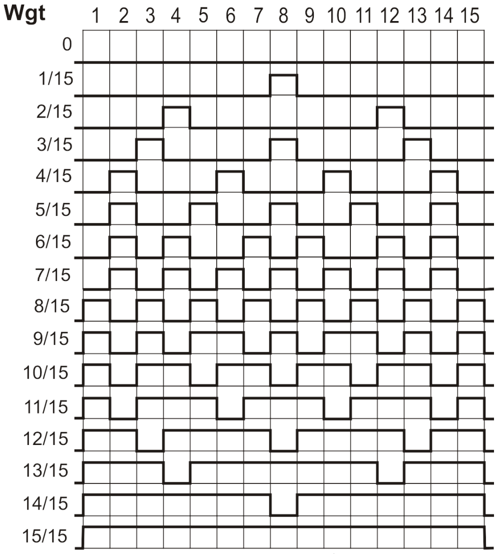

Fig. 1: Weights in proposed 15s systems of the timing

As the example we can input value x=8/15 multiply with the weight w=9/15, according the "Fig. 1" and input values must by conversion to the time interval from begin time axis corresponding with input size. We can get the output signal y = 5/15 from the time interval. The real value from the $x \cdot w = 0.32$ and the result after "Fig. 2" is $5/15 = 0.3\overline{3}$.

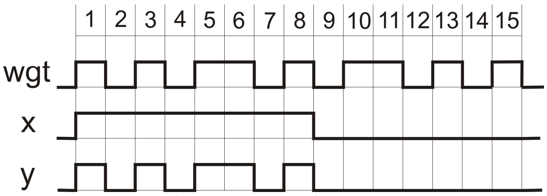

Fig. 2: An example of the evaluation for weight wgt=9/15 and the input x=8/15

Natural property this proposed method of multiplication is rounding. For verification effect of the rounding on the result CNN network we are create simulator as macro at Visual Basic for Application in Microsoft Excel. We used the simulator to recognize, that we can this rounding neglect. For example existence of multiple rounding the intermediate result caused that the final result network will be delayed about one iteration later, than in example without rounding.

The cell of CNN

The proposed circuit is a digital synchronous circuit. Designed circuit has been realized by means of descriptive language VHDL in development environment Xilinx. The cell contains several sub-circuits. The block diagram of cell is on the "Fig. 3". On the inputs of cell are 9 weight and 9 input signals and their corresponding signs. Eight input signals are attaches on the output of neighboring cells and one input, fifth in sequence is attaches on own output of cell, because in the CNN theory is the cell neighbor to itself. Then the inputs are multiplied with weights in logical *AND* gate and sign inputs are compared with the logical *XOR* gate. Size of weight must be special decomposition in time, so that pass-over *AND* gate are signals multiply. After multiplication of the are results counting in the block *counter* and consequentially converting over block *transfer function* is realizing the transfer function.

Converted signal over transfer function is coming through multiplexer *mx* to the block *converter*. Multiplexer *mx* allows enter input values to the network. Block *converter* has two functions: it contain register, where is saving result, from it is possible to read (*data out*) and circuit converting results on time interval corresponding size of results (*statex_o, sign_o*), which feed into surrounding cells.

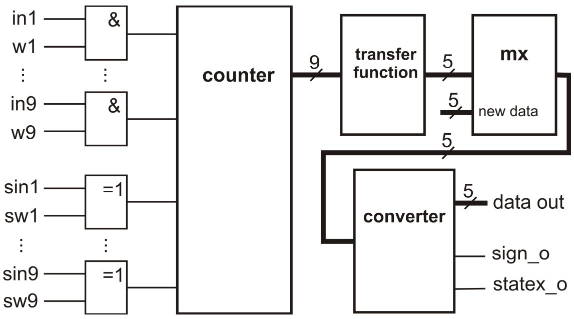

Fig. 3: Block diagram of cell of the CNN

The new architecture of the CNN network

CNN network consists from a field of cells, in that every cell is coupled with everyone of the nearest neighbor i.e. output one's cell is the input to all surrounding cells. These coupled cells in the CNN network are displayed on "Fig. 4". From this picture we can see, that every cell is on the fifth input coupled with its own output, because in the CNN theory is every cell neighbor to itself too.

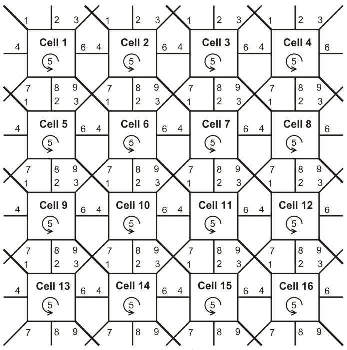

Fig. 4: Connections between cell of CNN

The circuts was programmed in description language VHDL in Xilinx ISE 10.1 and then implemented in FPGA chip (Virtex 5 - XC5VFX30T). After synthesis, the one cell of network contains 187 gates and calculation one iteration is 15 clock cycles for 5-bits rsolution. The maximum frequency of one cell of network with new architecture is 369MHz. The time need to calculation of one iteration (15 clock cycles) for maximal frequency is 40.65ns.

RESULTS

The main aim of our work was to propose new architecture of the neural CNN network with alternative way of multiplication, that us allow to reduce chip area by implementation of this CNN network. Our main comparison parameters were the speed and size (number of gates) of network. We implemented our new CNN network in FPGA chip. For comparison properties of network, we implemented in FPGA also our previous network presented in conference EDS 2008 [7] (Fig. 5) and standard CNN network with 5-bits parallel multipliers (Fig. 6).

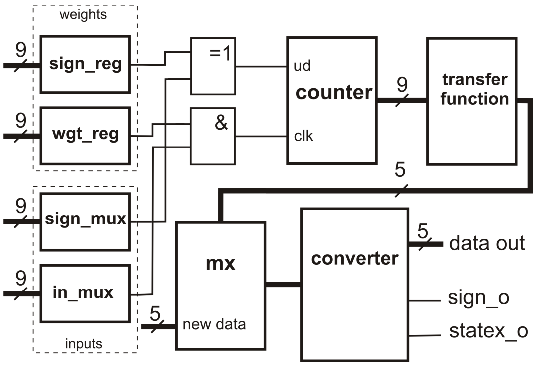

Fig. 5: Our previous CNN network

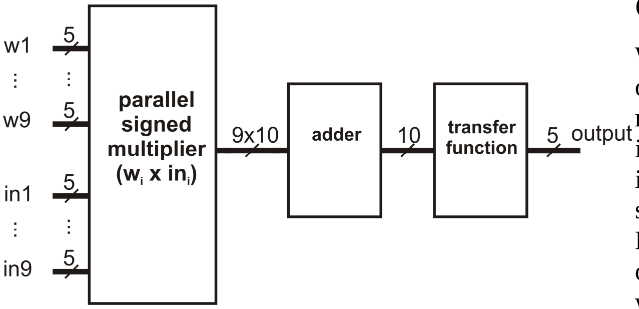

Fig. 6: Our previous CNN network

Parameters for all networks are in the table 1 and table 2. The main parameter proposed architecture was area consumption. Smallest area consumption network with new architecture (one cell contains 187 gates) and biggest area consumption network with parallel multipliers 1415 gates. Our previous network contains 461 gates.

Second parameter was speed of circuit. The smallest number of clock signals need to calculation of one iteration was standard network with parallel multipliers (4 clock cycles), though its maximal frequency is only 86MHz and because the time of calculation of one iteration is 46.5ns. Quickest was network with new architecture, [3] which need 15 clock cycles, but its maximal frequency is 369MHz, what is 40.65ns. Slowest was our previous model, which calculation of one iteration during 409ns.

In these facts see, that cell a new network takes 7.5 times less gates how standard network with parallel multipliers (what is 86.8% of gates) and its speed is little higher (calculation of one iteration is about 5.85ns shorter).

Table. 1: Parameters of Implemented CNN architecture

Cell of the CNN	Parameters	
	Number of clocks for one iteration	Number of gates
CNN with 5bit signed parallel multipliers	4 CLK cycles	1415
Our proposed CNN	135 CLK cycles	461
New design of the CNN	15 CLK cycles	187

Table. 2: Parameters of Implemented CNN architecture

Cell of the CNN	Parameters	
	Max. frequency	Time of calculation of one iteration
CNN with 5bit signed parallel multipliers	86MHz	46.5ns
Our proposed CNN	330MHz	409ns
New design of the CNN	369MHz	40.65ns

CONCLUSION

We proposed and implemented in FPGA digital chip of neural network, which used AND gate for multiplication of signals distributed in time. Network is working with 5-bits inputs and outputs. Output and input from cell can reach the values from -1 to 1 with step 1/15, what represented 31 gray shadows.

Design of the new architecture saves 86% gates in opposite to standard CNN with parallel multipliers, what allows implementation essentially more cell on the same area. Network speed is little increased (time of calculation of one iteration is smaller about 5.85ns).

The proposed network is fully cascade. So it allows to create network with optional number of cells.

ACKNOWLEDGMENTS

This contribution was supported by the Ministry of Education Slovak Republic under grant VEGA No 1/0693/08 and conducted in the Centre of Excellence CENAMOST (Slovak Research and Development Agency Contract No. VVCE-0049-07).

REFERENCES

[1] A. Muthuramalingam, S. Himavathi, E. Srinivasan, Neural Network Implementation Using FPGA: Issues and Application, International Journal of Information Technology Vol. 4 Num. 2, 2008-08-28.

[2] J. Larsen, Introduction to Artifical Neural Network, Section for Digital Signal Processing Department of Mathematical Modeling Technical University of Denmark, 1st Edition November 1999.

[3] Yang, C.-C.; S.O. Prasher; J.A. Landry. 2000. Application of Artificial Neural Networks to Plant Image Recognition in the Field. 2000 ASAE Annual International Meeting. July 9-12, 2000, pp. 147-153.

[4] Martin Hänggi a George S. Moschytz, Cellular Neural Network: Analysis, Design andOptimalization, Kluwer Academic Publisher, Boston, 2000, ISBN 0-7923-7891-1.

[5] Mohamed Boubaker, Khaled Ben Khalifa, Bernard Girau, Mohamed Dogui and Mohamed Hédi Bedoui, On-Line Arithmetic Based Reprogrammable Hardware Implementation of LVQ Neural Network for Alertness Classification, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008.

[6] H. F. Restrepo, R. Hoffman, A. Perez-Uribe, C. Teuscher, and E. Sanchez. A Networked FPGABased Hardware Implementation of a Neural Network Application In K. L. Pocek and J. M. Arnold, editors, Proceedings of the IEEE Symposium on Field Programmable Custom Computing Machines, FCCM'00, pages 337-338, Napa, California, USA, April 17-19, 2000. IEEE Computer Society, Los Alamitos, CA.

[7] E. Raschman, D. Ďuračková: The Novel Digital Design of a Cell for Cellular Neural, Proceedings of the Electronic Device and Systems, Brno, September, 2008.