

Realization of Three-dimensional DT-CNN on FPGA

Nguyen Tien Dat, Nguyen Tien Dzung, Thang Manh Hoang

Faculty of Electronics and Telecommunications
 Hanoi University of Science and Technology
 1 Dai Co Viet, Hanoi, Vietnam
 Email: thang@ieee.org

Abstract—In this paper, realization for the architecture of three dimensional cellular neural network is introduced for the first time. The complete parallelism is to maximize the performance of CNN. Implementation for 3D-CNN with the size of 4x4x3 on FPGA will be described. Simulation result shows the behavior of the network.

1. Introduction

Cellular neural network (CNN) was introduced by Chua and Yang in 1988 [1]. It is considered as a special case of artificial neural networks. Due to their parallel processing mechanism, the CNN has been successfully used for application requiring high speed processing such as image processing [2], pattern recognition [3], solving PDEs [4], etc. The structure of CNN is formed by local connection, and this feature allows ones to implement on VLSI [5, 6].

There are two approaches to realize CNN on digital platforms. The first is to develop a complete single layer network [7]. In order to emulate 3D-CNN, this layer must be shifts along the input array of its neighbor layers, so it requires architecture that allows to multiplexes inputs and stores current and next output values in memory. This architecture obtains the efficient resource usage and ability in emulating large size of 3D-CNN, but not gets an optimum computing performance. The other way is to direct implementation of entire 3D-CNN on chip. Due to massively parallelism of CNN, it can result the best performance of CNN. The main disadvantage of this approach is the usage of large resources. In this paper, the last approach is chosen for implementation.

In recent years, implementation of CNN has been carried out in various platforms i.e. computer simulation program, ASIC, emulated – digital in FPGA and DSP [8]. The analog ASIC implementation of CNN is the most powerful processor, but its disadvantages are long development time and high cost. Computer program and emulated CNN on reconfigurable chip, FPGA are suitable for verifying a model. In this paper, we present the design of complete 3D-Cellular Neural Network of the size of 4x4x3 with the template of 3x3x3 on FPGA.

2. Cellular Neural Networks

In theoretical, the structure of CNN can be of n-dimension (n=1, 2, 3...). Following, fundamental of 2D-CNN and 3D-CNN is described.

2.1. 2D-CNN

2D-CNN is the dynamical system of identical cells, a cell connects locally to neighbor cells to form a two dimensional array as illustrated in Figure 1.

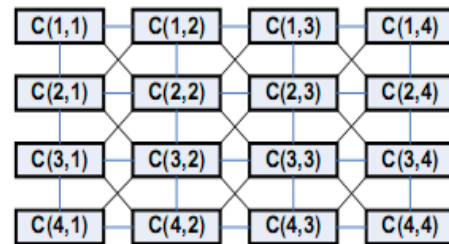


Figure 1. A two dimensional 4x4 CNN with neighboring distance $r=1$.

Cell $C(i,j)$ contains linear and nonlinear circuit elements. Specifically, it may include independent sources, a linear capacitor, linear resistors, linear and nonlinear controlled sources. A cell $C(i,j)$ couples with neighbor cells via the controlling input voltage and it receives feedbacks from the output voltage of neighbor cells $C(k,l)$. The templates $A(i,j;k,l)$ and $B(i,j;k,l)$ are the weight parameters of links between cells $C(i,j)$ and neighbor ones, $C(k,l)$. The behavior of CNN depends on these template values. The state of cell $C(i,j)$ is given by the following equation:

$$C \frac{dx_{ij}}{dt} = -\frac{1}{R} x_{ij} + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l) y_{kl} + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l) u_{kl} + z_{ij} \quad (1)$$

$1 \leq i \leq M, 1 \leq j \leq N$

The output of cell $C(i,j)$ is as

$$y_{ij} = \frac{1}{2} (|x_{ij} + 1| - |x_{ij} - 1|) \quad (2)$$

$1 \leq i \leq M, 1 \leq j \leq N$

2.2. 3D-CNN

In the 3D-CNN, the cells are similar to those in the 2D-CNN. The 3D-CNN is formed as collection multiple 2D-CNN layers, and connections between cells is not only in the existing layer of cell, but also to cells in the two neighbor layers (under and upper ones) as depicted in Figure 2. We consider the neighbor

as a sphere, so if $r=1$ the cell $C(i,j,k)$ has 26 neighbor cells. Hence, the templates have three indexes as $A(i,j,k)$; $B(i,j,k)$

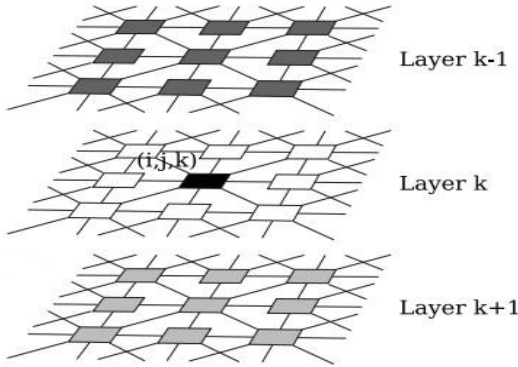


Figure 2. 3D-CNN with $r=1$

The state and output equations of 3D-CNN is derived as follow:

$$C \frac{dx_{ijk}}{dt} = -\frac{1}{R} x_{ijk} + \sum_{C(k,l) \in Nr(i,j)} A(i, j, k; l, m, n) y_{lmn} + \sum_{C(l,m,n) \in Nr(i,j,k)} B(i, j, k; l, m, n) u_{lmn} + z_{ijk} \quad (3)$$

$$y_{ijk} = \frac{1}{2} (|x_{ijk} + 1| - |x_{ijk} - 1|) \quad (4)$$

3. Three Dimensional CNN Design and Realization

Using the forward Euler discretization method for Eq. (3) with $R=C=1$ and time step $=1$, the single cell can be described by the DT-CNN dynamics:

$$X_{ijk}[q+1] = \sum_{k,l,m \in Nr(i,j,k)} A_{lmn}[q] y_{lmn}[q] + \sum_{k,l,m \in Nr(i,j,k)} B_{lmn}[q] u_{lmn}[q] + z_{ijk} \quad (5)$$

$$y_{ijk}[q] = \frac{1}{2} (|X_{ijk}[q] + 1| - |X_{ijk}[q] - 1|) \quad (6)$$

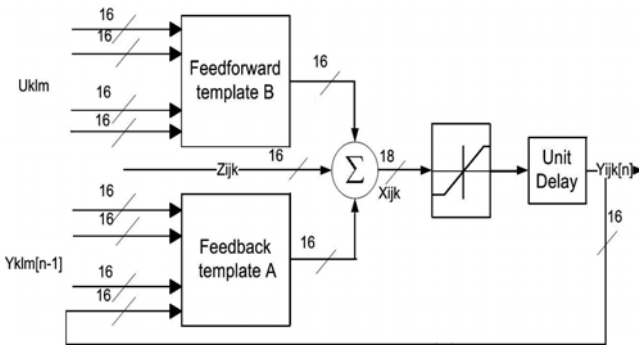


Figure 3. Discrete time model of cell

The 3D-CNN with the size of $4 \times 4 \times 3$ is considered. The discrete time CNN model in Eq. (5) is illustrated in Figure 3. The input data, the initial, output data, the bias and templates are in the format of 16 bit fixed-point signed value, (8.8), and the output state of cell is represented in 18 bit fixed-point signed value, (10.8). The architecture of single cell has two convolution blocks and non-linear function block which realize Eq. (5) and Eq. (6), respectively. These convolution units are composed simply of arithmetic adders and multipliers. The structure of convolution blocks is shown on Figure 4. The data from 27 neighbor cells is multiplied by the corresponding template coefficients, and then is truncated to 16 bit. After then, the result is added together by 5-pipelined-stages tree adders.

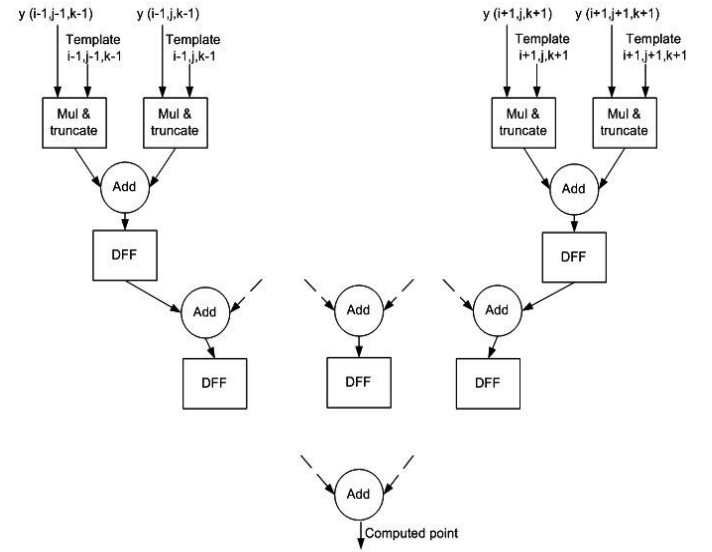


Figure 4. Architecture of convolution block

The datapath circuit is shown in Figure 5. The serial-to-parallel converter with output-latched reads the input data row-by-row, stores and feeds them simultaneously to all cells of CNN array, which composed of 3 layers of 4×4 cell array. The processed data are then stored in parallel to serial converter block.

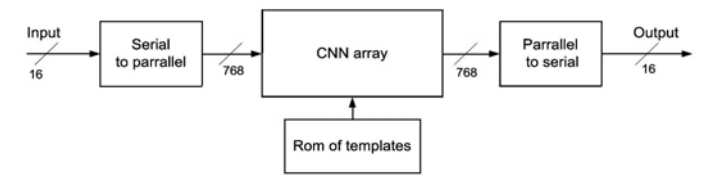


Figure 5. Datapath circuit architecture

Our proposed architecture has been implemented on Altera Cyclone II EP2C70F896C6 FPGA using VHDL. The synthesis result of place-and-route and speed are shown in Table 1.

Table 1. Report of resource consumption

Total logic elements	10,592
Total combinational functions	6,724
Total registers	7,339
Maximum frequency	186.64 MHz

4. Results

To verify the behavior of the system, we have simulated our architecture in ModelSim simulator tool.

For example, by setting zero for initial states of Layer 1(L1) and 3(L3), boundary conditions are set to zero, and templates are chosen as noise removal as below:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$z = 0$$

$$\text{Initial state (L2)} = \begin{bmatrix} -0.8 & 1.0 & -1.0 & -0.6 \\ 1.0 & 1.0 & 1.0 & -1.0 \\ -1.0 & 0.9 & -1.0 & -0.8 \\ -0.9 & -1.0 & -0.7 & -0.8 \end{bmatrix}$$

Simulation result of this example is shown in Figure 6. In this example, the stable state of cell is given by

$$x_{i,j,k} = 2y_{i,j,k} + y_{i-1,j,k} + y_{i+1,j,k} + y_{i,j-1,k} + y_{i,j+1,k} \quad (7)$$

where $y_{l,m,n}$ are output of cell at steady state. Initial state of layer 2 is similar to initial condition of Chua's 4x4 arrays as given in Figure 9(a) of [1]. Without interaction with neighbor layers, the output result of Layer 2 is same with the result depicted in Figure 9(c) of [1].

By keeping above templates and initial states and changing template A as

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

the stable state of cell in the inner-layer becomes

$$x_{i,j,k} = 2y_{i,j,k} + y_{i-1,j,k} + y_{i+1,j,k} + y_{i,j-1,k} + y_{i,j+1,k} + y_{i,j,k-1} + y_{i,j,k+1}$$

(8)

And, the outputs corresponding to this defined template are shown in Figure 7.

$$y(L1)=y(L3) = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad y(L2) = \begin{bmatrix} 1.0 & 1.0 & -1.0 & -1.0 \\ 1.0 & 1.0 & -1.0 & -1.0 \\ -1.0 & -1.0 & -1.0 & -1.0 \\ -1.0 & -1.0 & -1.0 & -1.0 \end{bmatrix}$$

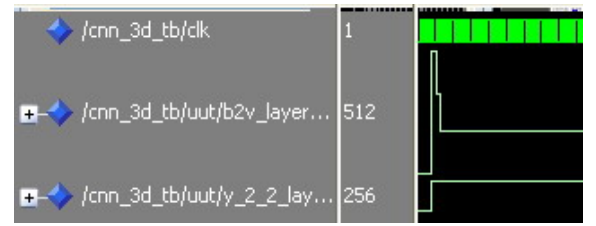
(a)

(b)

$$x(L1)=x(L3) = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad x(L2) = \begin{bmatrix} 4.0 & 3.0 & -3.0 & -4.0 \\ 3.0 & 2.0 & -4.0 & -5.0 \\ -2.0 & -4.0 & -6.0 & -5.0 \\ -4.0 & -5.0 & -5.0 & -4.0 \end{bmatrix}$$

(c)

(d)



(e)

Figure 6. Simulation result of CNN behavior. (a) Outputs of layer 1 and 3 at steady state. (b) Outputs of layer 2 at steady state. (c) Final states of layer1 and 3. (d) Final states of layer2. (e) Transient waveforms of cell C(2,2,2) simulated on ModelSim

$$y(L1)=y(L2)=y(L3) = \begin{bmatrix} 1.0 & 1.0 & -1.0 & -1.0 \\ 1.0 & -1.0 & -1.0 & -1.0 \\ -1.0 & -1.0 & -1.0 & -1.0 \\ -1.0 & -1.0 & -1.0 & -1.0 \end{bmatrix}$$

(a)

$$x(L1)=x(L3) = \begin{bmatrix} 5.0 & 4.0 & -4.0 & -5.0 \\ 4.0 & 1.0 & -5.0 & -6.0 \\ -4.0 & -5.0 & -7.0 & -6.0 \\ -5.0 & -6.0 & -6.0 & -5.0 \end{bmatrix} \quad x(L2) = \begin{bmatrix} 6.0 & 3.0 & -5.0 & -6.0 \\ 3.0 & -4.0 & -8.0 & -7.0 \\ -5.0 & -8.0 & -8.0 & -7.0 \\ -5.0 & -7.0 & -7.0 & -5.0 \end{bmatrix}$$

(b)

(c)

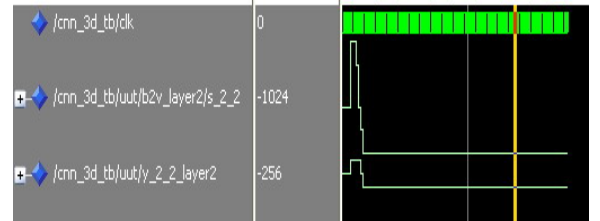


Figure 7. Behavior of CNN with different values of template A. (a) Outputs of three layers at steady state are the same. (b) Final states of layer 1 and 3. (c) Final states of layer2. (d) Transient waveforms of cell C(2,2,2) simulated on ModelSim

5. Conclusion

In this paper, we have described an approach to realize the 3D-CNN model with full parallel computation on FPGA platform. We also verify the approach by means of simulation for 3D-CNN with the size of 4x4x3. Future works will be focused on implementation of 3D-CNN with larger sizes.

Acknowledgments

This work has been supported by Ministry of Science and Technology of Vietnam with the project number ĐTĐL2009G/44. The authors would like to thank Vietnam's National Foundation for Science and Technology Development (NAFOSTED).

References

- [1] Chua, L. O. and L. Yang "Cellular neural networks: theory." *Circuits and Systems*, IEEE Trans. Circuits Syst 35(10): 1257-1272 ,1988.
- [2] Chua L. O and L. Yang, "Cellular Neural Networks: Application", IEEE Trans. Circuits Syst. Vol.35, Oct. 1998
- [3] L.O. Chua, T. Roska, "Cellular Neural Network and Visual Computing", Cambridge Univ. Press, Cambridge, 2002
- [4] T.Roska, L.O.Chua and D.Wolf "Simulating Nonlinear Waves and Partial Differential Equation via CNN "Circuits and Systems, IEEE Trans. Circuits Syst . Vol.42, p 807-820, Oct 1995.
- [5] T. Roska and L. O. Chua, "The CNN Universal Machine: An analogic array computer," IEEE Trans. On Circuits and Systems-II, vol. 40, pp. 163–173, 1993.
- [6] G. Linan, R. Dominguez-Castro, S.Espejo and A.Rodriguez-Vazquez, "ACE16k: A programmable focal plane vision processor with 128×128 resolution" Proc. of the 15th European Conference on Circuit Theory and Design, vol. 1, pp. 345–348, 2001
- [7] L.Raffo, S.B.Sabatini, G.M.Bisio "A reconfigurable architecture mapping multilayer CNN Paradigms"on 3rd International Workshop on Cellular Neural Networks and Their Applications ,December, 1994
- [8] Z. Nagy "Implementation of emulated digital CNN-UM architecture on programmable logic deveices and its applications" Ph.D. thesis , 2007.