2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

An Improved Emulated Digital CNN Architecture for High Performance FPGAs

László Füredi[†], Zoltán Nagy[‡], András Kiss[†] and Péter Szolgay[†] [‡]

†Faculty of Information Technology, Péter Pázmány Catholic University
Práter street 50/a, Budapest, Hungary
‡Cellular Sensory and Wave Computing Laboratory Computer and Automation Institute,
Hungarian Academy of Sciences
Lágymányosi street 11, Budapest, Hungary
e-mail: furla@digitus.itk.ppke.hu, nagyz@sztaki.hu, kissa@digitus.itk.ppke.hu, szolgay@sztaki.hu

Abstract—Cellular Neural Network (CNN) is a prototype Single Instruction Multiple Data (SIMD) like architecture, where the basic operation of this architecture is the weighted sum calculation. The emulated digital CNN-UM architecture was implemented and tested on different kind of array computers, eg. Cell Broadband Engine (Cell BE), Field-Programmable Gate Arrays (FPGAs), for utilizing the high performance of the digital microprocessors. The arithmetic unit of the original Falcon architecture was mainly optimized for the special features of the Xilinx Virtex-II architecture. Implementing the same architecture on the new Digital Signal Processor (DSP) optimized FPGAs will be inefficient. In order to achieve the highest possible performance the dedicated elements of the new FPGAs should be fully utilized. Therefore an improved arithmetic unit should be designed. According to the requirements of the new arithmetic unit the input data structure and the data-flow of the processor should be redesigned. Additionally the interconnection of the Falcon processing elements are optimized to utilize the specialized interconnect resources on the FPGA. Compared to the original Falcon processor with the modified implementation on the new FPGA families the clock frequency can be improved by 20 percent. Additionally the area requirement of the arithmetic unit is significantly reduced by utilizing the special features of the DSP blocks.

1. Introduction

In high performance processors the operation delay and the wiring delay is comparable. This effect is explicable with the scaling down of the technology. The increase of clock frequency the signal does not have enough time to reach the destination in one cycle. The adjacent computational elements can communicate faster because in short range the wiring delay is not significant. The effective architecture design's prime aspect is the locality precedence. This precedence is studied in an emulated digital CNN-UM implementation. A multi-layer CNN array can be used to solve the state equation of complex dynamical systems [1][2]. The CASTLE and Falcon emulated digital CNN chips were designed to reach this goal [3][4], where the accuracy, template size, cell array size and the number of layers can be configured. This paper describes synthesis and implementation methods used for the modified Falcon processor array on Virtex-5 and Virtex-6 FPGAs. The Falcon architecture is designed to solve the full signal range model of the CNN cell [5][6].

$$\begin{aligned}\dot{x}_{i,j}(t) &= \sum_{k=0}^{2\cdot n}\sum_{l=0}^{2\cdot n}\mathbf{A}_{k,l}\cdot x_{i+k-n,j+l-n}(t) + \\ &+ \sum_{k=0}^{2\cdot n}\sum_{l=0}^{2\cdot n}\mathbf{B}_{k,l}\cdot u_{i+k-n,j+l-n}(t) + I_{i,j} \qquad (1)\end{aligned}$$

where x, u and z are the state, input and the bias values of the CNN cell, n is the neighbourhood size, A is the feedback, B is the feed forward template. The templates are (2n+1)×(2n+1) sized matrices. The state equation of the CNN array is solved on the Falcon architecture by forward Euler discretization. The h time step value can be inserted into the templates A and B, these modified templates are denoted by $\hat{A}$ and $\hat{B}$. Usually the input values do not change for several time steps so the state equation (1) can be partitioned into two parts, the feedback (2) and the feedforward part (3).

$$x_{i,j}(m+1) = \sum_{k=0}^{2\cdot n}\sum_{l=0}^{2\cdot n}\hat{\mathbf{A}}_{k,l}\cdot x_{i+k-n,j+l-n}(m) + g_{i,j} \qquad (2)$$

$$g_{i,j} = \sum_{k=0}^{2\cdot n}\sum_{l=0}^{2\cdot n}\hat{\mathbf{B}}_{k,l}\cdot u_{i+k-n,j+l-n} + h\cdot I_{ij} \qquad (3)$$

The problem to be solved how to map the computational problem defined in (2) and (3) on a virtual array to a given physical FPGA where area/processor (logic slices, DSP slices), on-chip memory (Block Random Access Memory (BRAM)) and off-chip memory bandwidth are limited. Depending on the complexity of the operator a small amount of physical execution units can be implemented n << N×M (in 2D case) or N×M×L (in 3D case).The operator can be decomposed into small basic blocks which use either the logic resources (such as adders) or the dedicated resources (embedded multipliers) of the FPGA. The result of this process is a Virtual Cellular Machine optimized for the given application. The optimization can be focused on area, accuracy, speed, dissipated power etc. Main components are on-chip memory and the specialized execution unit.

2. The resources on an FPGA

The main configurable elements of the new Xilinx Virtex family is the Advanced Silicon Modular Block (ASMBL)[7]. The architecture is column based where each ASMBL column has specific capabilities, such as logic, memory, Input/Output, DSP, hard IP and mixed signal. By using different mix of the ASMBL columns domain specific devices can be manufactured. In the new architecture traditional 4-input Look-up Tables (LUTs) are replaced by 6-input LUTs. Each configurable logic block (CLB) is divided into two slices and every slice contains 4 6-input LUTs, 4 registers, and carry logic. In the new FPGAs the simple multipliers are replaced by complex DSP blocks called XtremeDSP (DSP48E) slices, it supports over 40 dynamically controlled operating modes including: multiplier, multiplier-accumulator, multiplier-adder/subtractor, three input adder, barrel shifter, wide bus multiplexers, wide counters, and comparators. The heart of the DSP48E is a 25bit by 18bit 2's complements signed multiplier with full precision 43-bit result. It also contains a 48bit Arithmetic Logic Unit (ALU) with optional registered accumulation feedback and support for SIMD operations. Additionally, hard wired 17 bit shift capability simplifies the construction of large multipliers, while optional pipeline registers enable even 550MHz operation. The number of DSP48Es is 1056 in a Virtex-5 SX240T and 2016 in a Virtex-6 SX475T FPGA. The other key configurable elements are the interconnect wires. In the contribution we especially focus on minimization of wire delays.

3. Architectural improvements

The new FPGA families have much more resources than the Vitrex-II FPGA which was used for the first implementation of the Falcon processor. For solving the discretized version of the CNN state equation a large number of multiplication is needed which can easily and efficiently implemented by using the dedicated elemnts (multipliers or DSPs) of the FPGAs. The aviable dedicated resources of the differnt FPGAs can be seen on Figure 1. Scaling up the original Falcon architecture on the new FPGAs in terms of the multipliers shows that on new FPGAs there are not enough configurable logic resources. If 32 original Falcon processor cores are implemented on the Virtex-II 3000 FPGA 94 percent of the configurable logic blocks and all of the multipliers can be utilized. Examining the aviable resources on the Virtex-5 SX50T, Virtex-5 SX240T and Virtex-6 SX475T devices 228, 182 and 140 percent of aviable logic block is required to implement the origianl Falcon processor when all multipliers are utilized as shown in Figure 2. The question is how to arrange the computation to use all multipliers while not overusing configurable logic blocks to implement the Falcon architecture on new FPGAs. The new modified type of architecture is shown on Figure 3, where the mixer and arithmetic units were changed. With these changes which will be described in the next sections all of the built-in DSP48E slices can used and the configurable logic block requirement of the processor core is also reduced.

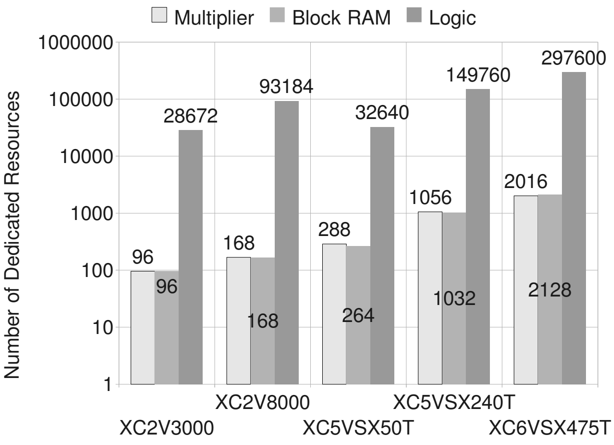

Figure 1: The Resorces of FPGAs

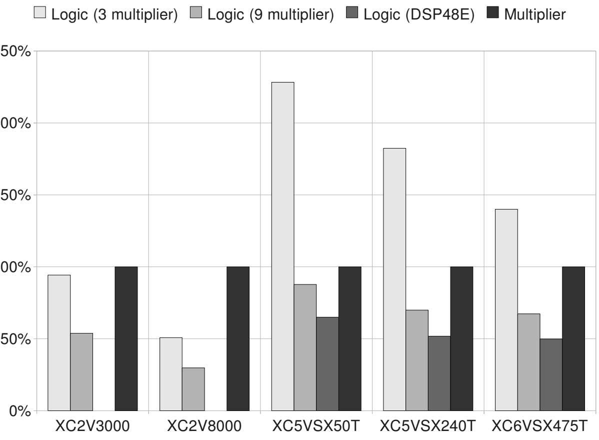

Figure 2: The logic usage in Falcon

3.1. Modified Mixer Unit

The structure of the mixer unit is shown in Figure 4. This unit contains one block of shift registers to store a window around the currently processed cell and two additional

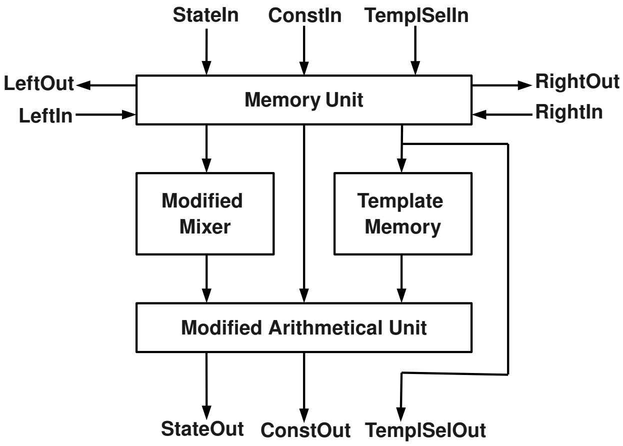

Figure 3: Structure of one Falcon processor core

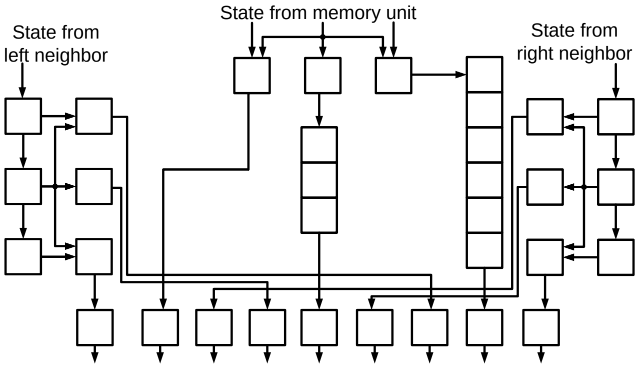

Figure 4: Structure of the modified mixer unit

block of shift registers which are used to store data from the left and right neighbors of the processor. The registers are connected serially and its outputs are also connected to the Sx inputs of the arithmetic unit. Communication between the neighboring processors is carried out through the left and right inputs without affecting the arithmetic unit. As a result the number of cycles required for the processing is reduced which increases the performance of the architecture and enables 100 percent utilization of the multipliers in the arithmetic unit.

3.2. Modified Arithmetic Unit

The Sx inputs of the arithmetic unit are connected to the state value outputs of the mixer unit while the Tx template values are connected to the output of the template memory. The precision of the state values is 25bit while the inputs are 18 bit wide. All of the multipliers and adders are implemented inside the cascaded DSP48E slices. Using this structure the dedicated connections between the DSP48E slices can be utilized. Therefore the operating frequency of the arithmetic unit is the maximum that the DSP slices allow. Depending on the speed grade of the FPGA the operating frequency of the arithmetic unit can reach 550MHz on the Virtex-5 and 650MHz on the Virtex-6 FPGAs. Only one external element a register is required to store the feedforward value of the computation and it comes from the memory unit. The modified arithmetic unit is shown in Figure 5, and it can be used in pipelined mode.

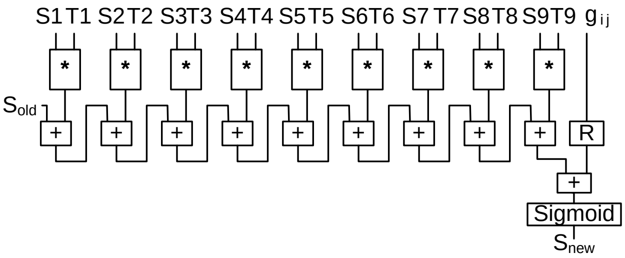

Figure 5: Structure of the improved arithmetic unit

4. Performance

Performance of the modified Falcon processor is compared to the speed of the software simulation. In the software simulation Intel Core 2 Duo E8400 and IBM CellBE processors are used. To simulate CNN array functions of the Intel Performance Primitives - Image Processing Library (IPP IPL) was used to help to optimize image- and vector-processing tasks. Performance of the software simulation depends on the size of the cell array. If the size is larger than 688×688 the performance drops to a lower level, due to the memory bottleneck and L2-cache memory occupancy. Even in a single Falcon processor configuration 38 percent performance improvement can be achieved compared to Intel Core 2 Duo processor. The easy scalability of the array makes it possible to connect severaly modified Falcon processor cores on one FPGA and get even more performance. Using the previously described architecture and utilizing all the 224 modified Falcon processors on the Virtex-6 FPGAs 145.6 billion cell iteration per second computing performance can be achieved. Our Virtex-6 FPGA based solution is 364 times faster compared to a high performance microprocessor, using all of the modified Falcon processors during the computation. Compared to a high performance Intel Core 2 Duo microprocessor, for a 1024 pixel ×1024 pixel picture instead of 38 template execution per second, 13885 template execution per second can be used with the Virtex-6 Falcon implementation.

5. Conclusions

An improved emulated digital CNN-UM architecture implementation was successful on our prototyping boards,

Table 1: Comparison of different implementations

	Implementations				
	Intel Core 2 Duo	Cell Processor 8 SPEs	FPGA		
			XC5VSX50T	XC5VSX240T	XC6VSX475T
Implementation type	Software (Intel IPP)[8]	Software (Cell SDK)	FPGA	FPGA	FPGA
Technology (nm)	45	65	65	65	40
Clock Frequency (MHz)	3000	3200	550	550	650
Number of Processing Elements	2 Cores	8 SPE	32 FPE	117 FPE	224 FPE
Million cell iteration/s	400	3627	17600	64350	145600
Speedup	1	9	44	160	364
Power Dissipation (W)	65	85	16	59	102
Area (mm^2)	107	2×253	N/A	N/A	N/A

using the Virtex-5 SX50T and Virtex-5 SX240T FPGA from Xilinx Inc. and implementation in simulation using the Virtex-6 SX475T FPGA. Our solution was optimized to the special requirements of the Virtex-5 and Virtex-6 FPGAs. The main parameters of the architecture is described and compared to the parameters of the software simulation of the CNN full signal range modell running on high performance processors such as Intel Core2Duo and IBM Cell.

6. Acknowledgements

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] Z. Nagy, Z. Vörösházi, and P. Szolgay, “Emulated Digital CNN-UM Solution of Partial Differential Equations,” *International Journal of Circuit Theory and Applications*, vol. 34, pp. 445–470, 2006.

[2] T. Roska, “An Overview on Emerging Spatial Wave Logic for Spatial-Temporal Events Via Cellular Wave Computers on Flows and Patterns,” *International symposium on nonlinear theory and its applications*, pp. 98–100, 2008.

[3] P. Keresztes, A. Zarándy, T. Roska, P. Szolgay, T. Hidvégi, P. Jónás, and A. Katona, “An Emulated Digital CNN implementation,” *International Journal of VLSI Signal Processing*, vol. 23, pp. 291–303, 1999.

[4] Z. Nagy and P. Szolgay, “Configurable Multi-layer CNN-UM Emulator on FPGA,” *IEEE Transaction on Circuit and Systems I: Fundamental Theory and Applications*, vol. 50, pp. 774–778, 2003.

[5] L. O. Chua and L. Yang, “Cellular neural networks: theory,” pp. 1257–1272, 1988.

[6] S. Espejo and et.al., “A VLSI-Oriented Continuous-Time CNN Model,” *International Journal of Circuit Theory and Applications*, vol. 24, pp. 341–356, 1996.

[7] “Xilinx product homepage,” http://www.xilinx.com, 2010.

[8] “Intel integrated performance primitives homepage,” http://software.intel.com/en-us/intel-ipp/, 2010.