ICSE2004 Proc. 2004, Kuala Lumpur, Malaysia

ASIC DESIGN OF A KOHONEN NEURAL NETWORK MICROCHIP

Avinash Rajah, Mohamed Khalil Hani
Dept. of Microelectronics and Computer Engineering
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81300 Skudai, Johor Bahru, Malaysia
Email : avin_rj@yahoo.com

Abstract **This paper discusses the Kohonen neural network (KNN) processor and its KNN computation engine microchip. The ASIC design of the KNN processor adopts a novel implementation approach whereby the computation of the KNN algorithm is performed on the custom ASIC microchip and its operations are governed by a FPGA based controller. Thus, the ASIC implementation of the KNN processor is derived through integration between a custom ASIC and FPGA. The 3.3V AMI 0.5um C05M-D process technology was used to achieve the VLSI design of the computation engine microchip and the entire design adopted the BBX cell based methodology, which is a viable alternative to conventional ASIC methodology.**

I. INTRODUCTION

Neural Networks are models of the human brain that have shown to possess the ability to learn. This has made it very suitable for problems that conventional computers are unable to solve, such as pattern recognition. With its increasing popularity, there is now a demand for dedicated neurohardware that offers low-cost high-speed performance with a compact implementation. Neurohardware have specialized architectures optimized for the implementation of neural networks. Thus, to cater to this need, a neuroprocessor core (KARN) targeting pattern recognition applications was designed to implement the Kohonen Neural Network (KNN) algorithm, one of the more popular neural paradigms. The KNN has been successfully implemented in various pattern recognition applications. In line with designing a high speed neuroprocessor implementing the KNN for real time applications, the KARN processor adopted the neuron-array based architecture, which is highly parallel, to match the inherent parallelism of the KNN. It sub-contracts the processing of each individual neuron of the KNN map to individual Processing Elements (PE). Due to the representation of each neuron of the implemented KNN map by individual PEs, the area requirement for realization of the architecture on a single ASIC chip proved to be impractical and rather expensive. In addition to that, different pattern recognition tasks and may command differing neuron-map sizes and hence the computation engine of the KARN processor cannot be implemented to an inflexible fixed map dimension. Thus, it was concluded that the matter was best resolved by detaching the controller and computation engine on separate ASICs.

The Neuron-Array Computation Engine (ACE) of the KARN processor is made up of scalable arrays of Processing Elements (PE) that emulate the Neuron-Map layer of the KNN. The PEs execute in a parallel manner to permit accelerated simulation of the KNN algorithm. Initial efforts targeting for FPGA device implementation revealed that even large devices such as the Altera APEX_20K200 were merely able to implement modest ACE designs of 4x4 PE arrays. Thus, VLSI implementation was employed to assure higher compaction and better execution speed. Also, the cost of a single VLSI chip was lower than an off-the-shelf FPGA device. The ACE was also designed to feature PE-level scalability and Chip-level cascadability. PE-level scalability promotes a simplified process for designing ACE chips with differing neuron-map dimensions. On the other hand, cascadability at chip level would avoid the impracticality of realizing large neuron-map dimensions of up to 50x50 neurons on a single ACE chip. Instead, construction of large maps of such sizes could be achieved merely through cascadation of a few ACE chips.

0-7803-8658-2/04/$20.00(c)2004 IEEE

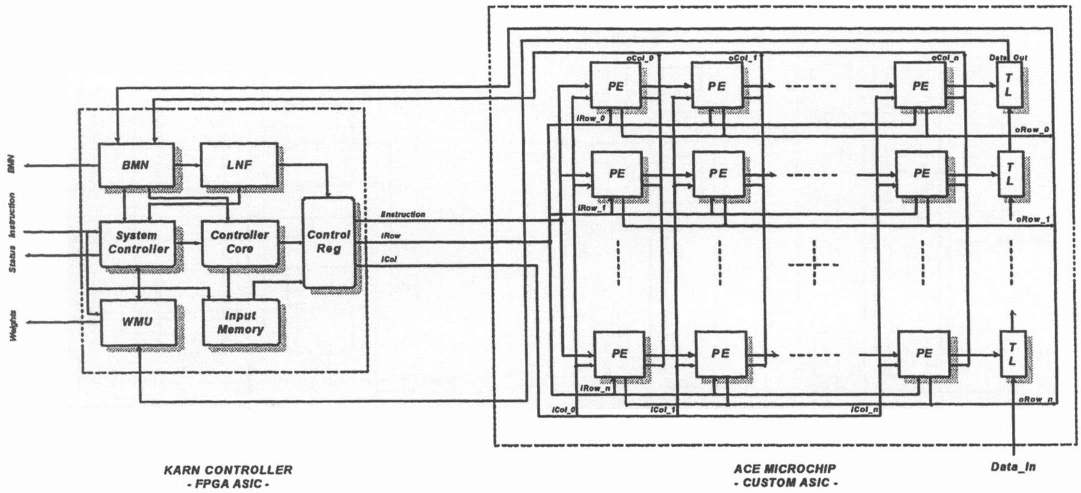

Fig. 1 Block Diagram Of KARN Neuroprocessor Core

II. THE KOHONEN ANN ALGORITHM

As with most Neural Networks, KNN consists of a learning phase and a recall phase. Algortihm 1 presents the learning phase algorithm while Algorithm 2 presents the recall phase algorithm.

Algorithm 1: The KNN Learning Algorithm

```
Step0: Initialize weights $w_{ij}$.
        Set topological neighborhood parameter.
        Set learning rate parameter.
Step1: While stop is false do steps 2 to 8
Step2: For each input vector x do steps 3 to 5
Step3: For each j, compute:
        $D(j) = \sqrt{\sum_i (w_{ij} - x_i)^2}$   ..(1)
Step4: Find index j such that D(j) is a minimum
Step5: For all units j within a specified
        neighborhood of j and for all i:
        $w_{ij}(new) = w_{ij}(old) + \alpha(x_i - w_{ij}(old))$ ..(2)
Step6: Update learning rate
Step7: Reduce radius of topological
        neighborhood at specified times
Step8: Test stopping condition
```

Algorithm 2: The KNN Recall Algorithm

```
Step1: For each j, compute:
        $D(j) = \sqrt{\sum_i (w_{ij} - x_i)^2}$
Step2: Find index j such that D(j) is a minimum
```

III. HARDWARE STRUCTURE OFTHE PROCESSING ELEMENT (PE)

The PE was designed to implement the Euclidean Distance (1) and Weight Update (2) computation of the KNN algorithm. In order to simplify the PE's circuitry, the Euclidean distance was substituted with the Manhattan distance and the adaptation factor, α, was restricted to the negative orders of two, i.e. 1, 1/2, 1/4, 1/8, 1/16. Although these alterations introduce some error into the measure, it is acceptable as a compromise between accuracy and speed of calculation [6]. The PE was designed to execute 7 different instructions, entirely relevant to the implementation of the Recall & Learn phase computations.

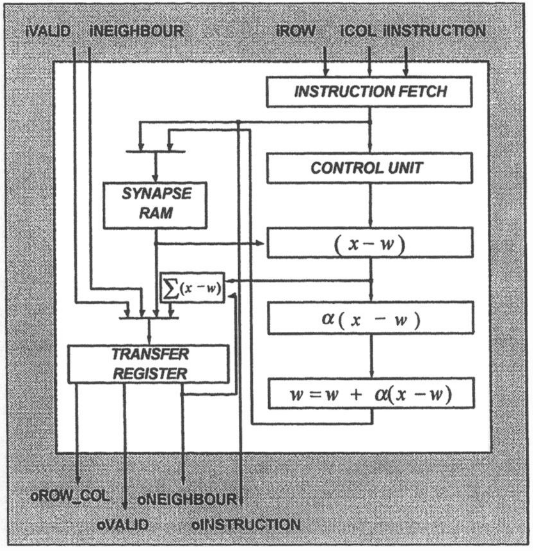

Fig. 2 PE Functional Block Diagram

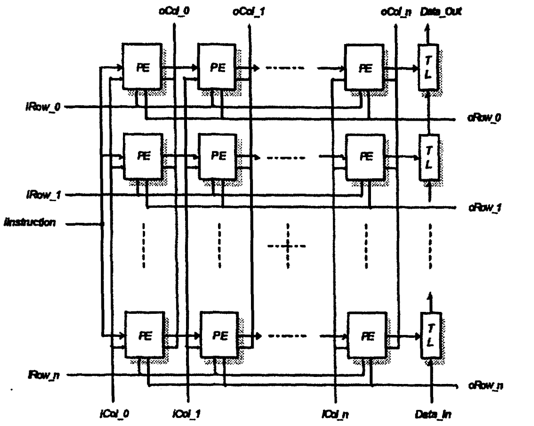

Fig. 3 The Scalable ACE Desig n Consisting Arrays Of PEs

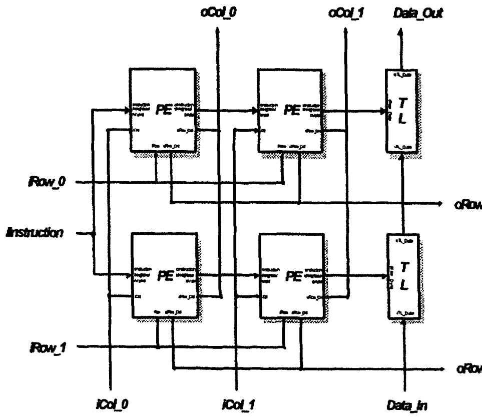

Fig. 4 The ACE_2x2 Functional Block Diagram

IV. THE STRUCTURE OF THE NEURON-ARRAY COMPUTATION ENGINE (ACE)

The scalable structure of the ACE design is illustrated in Figure 2. The scalable ACE design consists of only 2 submodules, being the previously described PE module and the Transmission Module (TL). All PEs and TL modules of the ACE are clocked with the same signal. Thus, its operation is thoroughly synchronized. The TL module is responsible in transmitting out all read PE weight vectors in a row-by-row fashion and also plays an important role in supporting the cascadability feature of the ACE. When an ACE is cascaded with other ACE chips and is not actively processing, the TL module transmits data from lower to upper ACE chips through it. As illustrated by Figure 2, the ACE design can be scaled to any array dimension, simply by placing an equal number of rows and columns of PEs to form a symmetrical array dimension. A single column of TL modules is then placed next to *row_n* to form the transmission bus. The intended ACE design is finally concluded by making appropriate signal connections between the used modules.

A 17bit instruction bus from the KARN processor controller is connected to the ACE to transmit instruction streams for PE operations. The instruction stream is received by the f^{t} row of PEs and subsequently transmitted to the following rows in a systolic manner. Individual row and column control signals, also originating from the KARN controller, is used to address specific PEs or rows and column groups of PEs and activates the appropriate rows and columns in an orderly manner for the RECALL, LEARN, LOAD and READ functions. The row-column signals are particularly important in synchronizing the countdown operation of all PE Transfer Registers in the BMN search function. Output row and column signals are used also in the BMN search operation to denote the winning PE's coordinate within the array.

V. VLSI DESIGN OF THE ACE_2X2 CHIP

For prototyping and proof-of-concept purposes, a 2x2 neuron-map ACE chip, dubbed the ACE_2x2, was designed using the AMI 0.5um CO5M-D process technology. The entire design, from design entry till tape-out adopted a new and unconventional BBX cell-based design methodology, developed in-house at UTM. The hardware modeling of the ACE_2x2 was done using VHDL and the in-house developed design entry tool, VHDLmg. Upon functional verification, the VHDL design of the ACE_2x2 was ratified for ASIC back-end design.

The Tanner L-Edit Layout tool was employed to produce the physical design of the ACE_2x2 through means of P&R and full-custom layout. The layouts of the ACE_2x2 sub-modules; the PE and TL modules were produced through P&R based on timing verified logic netlists produced by Synopsys DC. The P&R stage utilized cells from the AMI MTC35000 Standard Core Cell library. The IO ring design on the other hand was achieved through the full-custom manner and used pad-limited I/O cells from the AMI MTC351000 Standard I/O Cell library. The final full-chip design was then derived and found to be 16.9 sq.mm in area of 4.11um x 4.11um and to contain 41,028 transistors. Upon functional, timing and physical verification for process design & electrical rule compliance, the ACE_2x2 was taped out in GDSII format for submission to the IC foundry, Europractice IMEC, Belgium. An 84-pin PGA package was chosen to house the 49-I/O-signal ACE_2x2 design. The design is currently being fabricated at Europractice and expected back by late September 2004, given a turn-around period of 3 months.

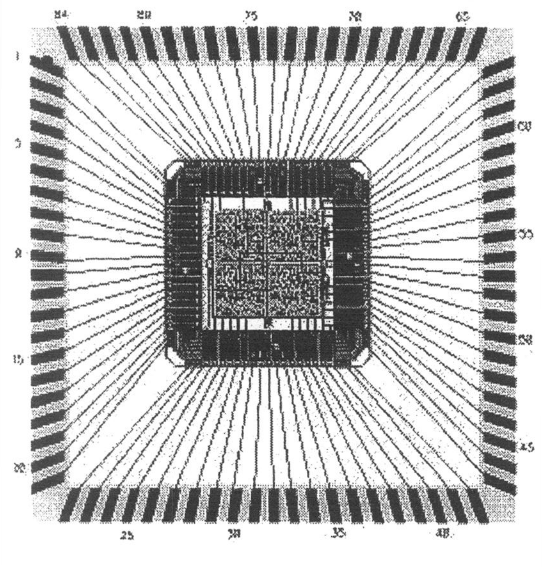

Fig. 5 ACE_2x2 Bonding Diagram

VI. Performance evaluation

Post-layout timing analysis for worst-case conditions of 2.7V supply voltage and 80 Centigrade environmental temperature ascertained the design to have a critical path of lesser than 12ns. Subsequently, final round of simulations functional operations of the ACE_2x2 at 86.66 MHz with the supply voltage being set at 3.3V and simulation temperature set at 27 Centigrade. For evaluation in terms of neurohardware performance metrics, the ACE design for a 50x50 chip is used instead of the ACE_2x2 and is able to deliver up to 37500 Million Connections Per Second (MCPS) and 27667 Million Connections Update Per Second (MCUPS). The delivered MCPS and MCUPS performances were significantly better than existing documented efforts.

VI. Conclusion

The paper presents the hardware structure of the scalable Neuron-Array Computation Engine (ACE) design and the VLSI design of a prototype ACE_2x2 chip. The ACE_2x2 chip is a composition of 4 PEs and 2 TL modules. The VLSI design was achieved using the AMI 0.5um C05M-D process and is able to execute at a minimum operating frequency of 86.66 MHz. The chip is currently being fabricated in Europractice IMEC, Belgium and is expected to be retrieved by late September, 2004. Upon retrieval, the chip will be integrated with the FPGA based controller of the KARN processor to implement the KNN algorithm for real time pattern recognition applications.

VII. References

[1] C. Lindsey. "Neural Networks in Hardware: Architectures, Products and Applications", *http://www.particle.kth.se/~lindsey),* 1998.

[2] T. Schoenauer, A. Jahnke, U. Roth, H. Klar, "Digital Neurohardware: Principles and perspectives", *Neural Networks in Applications, Magdeburg,* 1998.

[3] R. Togneri, Y. Attikiouzel, "Parallel implementation of the Kohonen algorithm on Transputer", *Int. Joint Conf. on Neural Networks(IJCNN91) Vol.2 Singapore,* 1991.

[4] M. Melton, Tan Phan, D. Reeves, D. Van den Bout, "The TInMANN VLSI Chip", *IEEE Trans. on Neural Networks, Vol.3, No. 3,* 1992.

[5] T.Kohonen, "The 'neural' phonetic typewriter", *Computer,* 1988.

[6] R Beale, T Jackson. "Neural Computing: An Introduction." *Adam Hilger, IOP Publishing Ltd, , p.25* 1990.

[7] S. Rueping, K. Goser, U. Rueckert. "A Chip for Selforganizing Feature Maps", *Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems, Italy,* 1994.

0-7803-8658-2/04/$20.00(c)2004 IEEE