SOLAR AND ITS HARDWARE DEVELOPMENT

Janusz Starzyk, Yongtao Guo and Zhineng Zhu
School of Electrical Engineering and Computer Science
Ohio University, Athens, OH 45701, U.S.A.
{starzyk, gyt, zhineng}@bobcat.ent.ohiou.edu

ABSTRACT

The focus of this paper is a novel parallel and self-organizing neural network hardware architecture and its development. This architecture is based on self-organizing learning array (SOLAR) algorithm reported earlier. This hardware architecture is aimed at hundreds of traditional reconfigurable field programmable gate arrays (FPGAs). Its reconfigurability scheme is distinct from the usual approach in which hardware can only be reconfigured via transferring new binary information to the configuration memory. SOLAR reconfigurability is derived from self-organizing learning process yielding dynamically self-reconfigurable structure. In this paper, we describe how this structure is being built.

1. INTRODUCTION

Artificial neural networks (ANNs), which originated from study of biological neural networks, are applied in many areas since their renaissance in the eighties. However, the implementation of ANNs has been dominated by software for its higher flexibility compared to hardware, although hardware [1] is more appropriate to implement highly parallel ANNs. This situation lasts until now even though the first parallel neurochip appeared at the beginning of nineties. One of the significant reasons for this are very limited hardware resources, especially the routing resources for ANN hardware implementation. The quadratic relationship [2] between the routing and number of neurons is one of the reasons limiting the growth of VLSI NN chip complexity. More hardware resources with fewer interconnections between neurons are required to solve this problem. To ease the area limitation of a single silicon chip, a 3 dimensional ANN implementation based on multiple chips organized in a space-efficient way was introduced [3]. Nevertheless a custom VLSI implementation requires significant non-recurring engineering (NRE) cost, therefore, programmable hardware based on FPGA is preferred for system prototyping.

In this paper, we describe SOLAR project aimed at modeling the biological neural networks – their non-software operation, ability of learning and problem solving, making associations, and eventually reason. We adopt FPGA as our hardware implementation platform. The objective of hardware implementation is to build the new self-organizing learning hardware by combing hundreds of computing cells implemented by high-end FPGAs to form a 3D structure. In this 3D structure, the parallelism is supported by computing cells with flexible interconnections. The self-organizing learning is based on a new machine learning algorithm [4] which combines knowledge from neural networks (NN) and information theory. Hardware reconfigurability is achieved by this self-organizing learning array involving flexible routing modules and hundreds of thousands of simple neurons. The routing module is the combination of the proposed configurable switching unit with various routing resources in FPGA.

The rest of this paper is organized as follows. In Section 2, self-organizing learning array architecture is introduced. Section 3 deals with SOLAR hardware development. Finally, a summary is given in Section 4.

2. SELF-ORGANIZING LEARNING ARRAY ARCHITECTURE

In our previous work, the entropy-based neural network learning algorithm was simulated on standard benchmarks and proved to be advantageous [4][5] over many existing neural networks and machine learning algorithms. Based on this algorithm, SOLAR architecture was developed. It uses self-organization for learning and hardware-reconfiguration. Its strength includes relatively sparse interconnections between neurons. SOLAR differs from classical artificial neural networks (ANNs) in the way it is organized and how it learns. Its most important advantage over ANNs is that it scales well in hardware. While classical ANNs are wire dominated (wiring area grows as a cube of the number of neurons), SOLAR's interconnection area grows almost linearly with the number of neurons. This linear relationship benefits hardware implementation of SOLAR.

In SOLAR, we adopt a feed forward network structure for its stability and fast learning. SOLAR architecuture, shown in Fig. 1, is implemented by an array of randomly connected identical processing units (neurons). The hardware fabric for SOLAR evolved from our earlier work on dynamically reconfigurable architecture for mobile communication system [6]. The modular nature of the array is emphasized by full integration of its routing and processing resources. Fig 2 shows a more detailed view of SOLAR basic cell. It contains a neuron, a configurable switching unit (CSU), and a bidirectional routing unit (BRU). Each neuron receives its data inputs (NI) either

from primary inputs or other neurons and sends its outputs (NO) to other neurons via its routing resources.

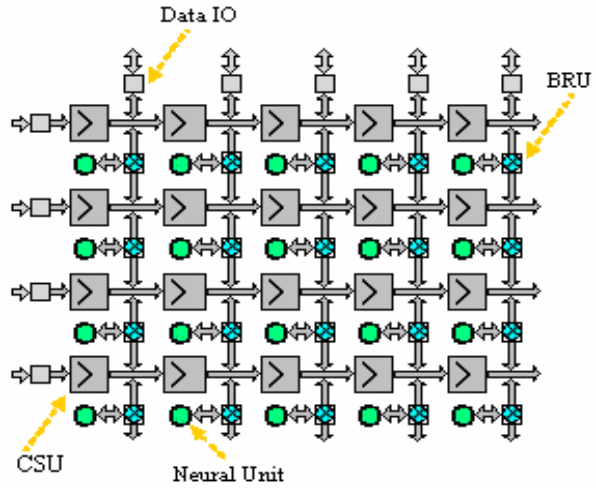

Figure 1. SOLAR architecture

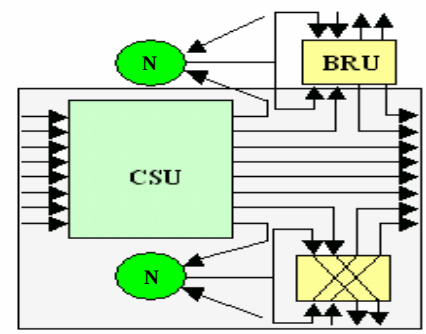

Figure 2. Basic cell of SOLAR

Neuron contains several building blocks: reconfigurable processing unit (RPU) which performs most of its arithmatic and logic operations, control unit (CU) which organizes data flow and operations of the neuron, dynamic probability estimator (DPE), an optional entropy based evaluator(EBE) and dynamic self-reconfiguration memory.

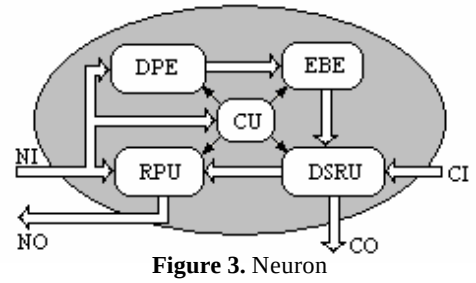

Figure 3. Neuron

Since neurons are data driven, they respond to activities on their input terminals. If several neurons are activated as a group, they can share the same control unit. Dynamic probability estimator uses special algorithm to dynmaically estimate class probabitlities of incoming samples without division. It only uses 3 CLBs per class and all probabilities are simultaneously updated with each new data input. EBE uses the estimated probabilities to estimate class entropy, information index and selects optimum operation of each neuron as described in SOLAR algorithm [4]. It is respossible for setting the neuron's configuration bits which are stored in DSRU. EBE can be shared by neurons which are not activated at the same time (for instance, if learning proceeds from a neuron on one level to the next one). Configuration input (CI) can be used to load DSRU with prestored configuration bits and configuration output (CO) can be used to read the configuration bits stored in DSRU. By connectiong CO of one neuron to CI of another one, a configuration data path can be obtained to scan the configuration information of the entire neural network in or out.

The main interconnection between neurons is achieved by the configurable switching unit (CSU), an example of which is presented in Fig 4.

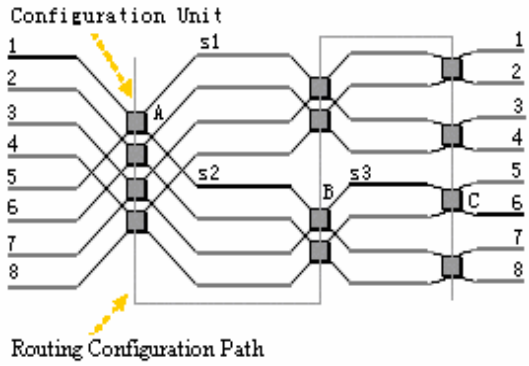

Figure 4. Configurable switching unit (CSU)

CSU contains a number of configuration units embedded into routing configuration path. Using this path, a random sequence is scanned in to establish pseudo-random connections between neurons. For each CSU shown in Figure 4, there are 8 inputs and 8 outputs. The wires connecting the input to the output are separated into three segments. Those segments use 12 configuration units to permute connections between input and output wires. Take for example the configuration unit A. If the bit value fed to the configuration unit A is 1, then input 1 is connected to the signal s1 and input 5 to s2; if bit value fed to the configuration unit A is 0, then input 1 is connected to s2 and input 5 to s1; subsequently, if the bit value fed to the configuration unit B is 1, then s2 is connected to s3, and s3 is connected to the output 6 if configuration unit C is fed with 0. In this way input 1 is connect to the output 6.

The connections between CSU and neurons are graphically presented in Fig 2. Assume that the part within the rectangle of Fig 2 is (m,n)-th element of SOLAR architecture, which means that this element is in row m and column n. The output 1 of the CSU is connected to the

input of neuron in position (m-1, n); the output 2 is connected to the input 8 of CSU in position (m-1, n+1); the outputs 3—6 are directly connected to the corresponding inputs 3—6 of the CSU at the location (m, n+1); the output 7 is connected to the input 1 of the CSU in the position (m+1, n+1), and the output 8 is fed to the input of the neural in the position (m, n). Each neuron has two outputs, one is connected to the input 7 of the CSU located at (m-1, n+1) and the other one is connected to input 2 of the CSU located at (m, n+1). With the assistance from both the configurable switching unit and bidirectional routing unit (BRU), the signal can be passed to either local neuron or a remote neuron in any forward layer. The local connections are asscociated with higher connection probabilities, and the remote connections have smaller probility.

In SOLAR architecture, initial connections are pseudo-randomly defined by configuration bits generated by linear feedback shift register and shifted to the configuration units. Figure 5 shows an example of the initial pseudo-random connections among neuron units in SOLAR that corresponds to a software generated configuration sequence.

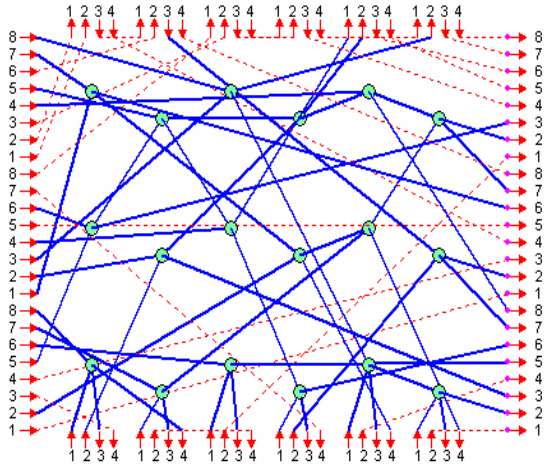

Figure 5. Pseudo-random connections among neurons

This interconnection approach applies to both the neuron's input signals as well as the neuron's control signals, which defines the learning subspace for each neuron. This pseudo-random routing scheme results in 100% allocation of routing resources and full routability between neurons.

3. HARDWARE DEVELOPMENT

Based on the flexible, reconfigurable and expandable SOLAR architecture, we can organize many neurons in hardware to better simulate functions of human brain, utilizing its parallel processing and self-organizing learning ability. To implement 3D SOLAR architecture, we divide the system development into two stages. The first stage is prototyping of a simple SOLAR architecture onto a single VIRTEX FPGA chip using software-hardware codesign. The second stage is system design and contains building a single PCB design with 2x3 array of VIRTEX FPGA chips, and final implementation of a 3D learning machine built by 384 VIRTEX XCV1000 FPGAs. SOLAR architecture has been simulated [4][5] at system level using Matlab and at behavioral level using very large-scale integrated circuit hardware description language. The results gained from performed simulations are satisfactory. The system level simulation using Matlab proves that this self-organizing algorithm is advantageous over many existing artificial neural networks and other classifier algorithms. In addition, the behavioral level simulation using VHDL verifies the hardware architecture and it implementation adaptability.

Prototype

In this section results from the first stage – architecture prototyping are discussed. We adopt hardware-software codesign to prototype a simple SOLAR architecture onto a single VIRTEX FPGA chip using a PCI demo board [7]. This codesign combines both the PC based software Matlab and C++ with FPGA based hardware, interfaced through a PCI bus. This prototyping is used to verify the SOLAR architecture and to develop PCI configuration interface logic used in the following stages of system development. In the hardware-software codesign, software is run on a PC using Matlab and C++ programs, and hardware is configured and run on FPGA chip embedded on PCI board using RTL VHDL description. A snapshot VHDL simulation of EBE module, which can be shared by neurons (see Fig.3), is shown in Fig.6. The learned results, signal "Opt_Threshold" and "ID", represent the searching threshold for the optimal operation "Opt_Func" and the corresponding information index for this particular training neuron in its learning subspace. These two SOLAR prototype design components (software and hardware) communicate dynamically via PCI bus. Since SOLAR is based on a supervised training algorithm, the hardware implementation includes the multiple neurons' training and voting configurations. The trained architecture is dynamically configured to implement the voting. The overall hardware-software codesign system is decomposed into three parts:

- FPGA (hardware): Neuron's architecture is dynamically configured into a single Virtex chip.
- CPU and memory (software): System initialisation, organization and management are implemented by Matlab programming. Some time-critical loop and recursive application are implemented by C++.
- Interface: It contains PCI core and interface logic in the hardware part, and C++ dynamical link libraries called from Matlab console in the software part.

PCB, RACK and 3D SOLAR

The second stage contains design of a single printed circuit board (PCB) design with 6 interconnected VIRTEX XCV1000 FPGA chips. All of the chips have

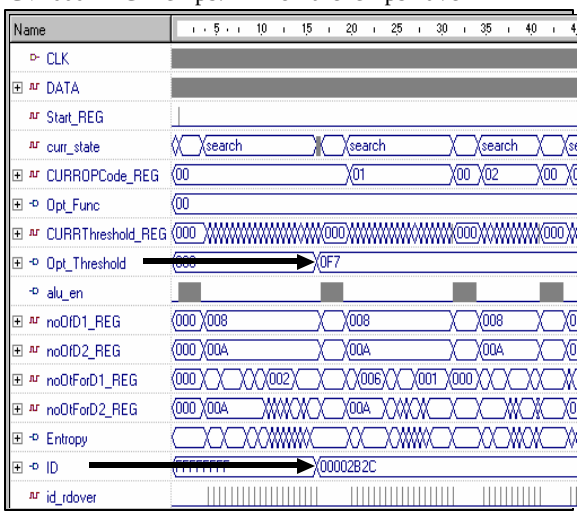

Figure 6. Neuron's EBE module simulation

identical architecture based on Fig. 1 and are interconnected in parallel via several hundred generic I/O pins. The initial configuration information will be transferred into every single chip one by one via JTAG port in a serial way. Then, the layout-identical PCB boards will be expanded to a rack architecture including 4 PCBs with twenty-four VIRTEX XCV1000 chips. These PCB boards will be interconnected vertically and configured in a broadcast fashion – the identical configuration bits are sent to every PCB board, while the FPGA chips are programmed via JTAG port in a daisy chain way. Finally, 3D SOLAR system, containing 4x4 SOLAR racks with close to 400 million gates, as shown on Fig. 7, will be built to implement a network of thousands of interconnected neurons.

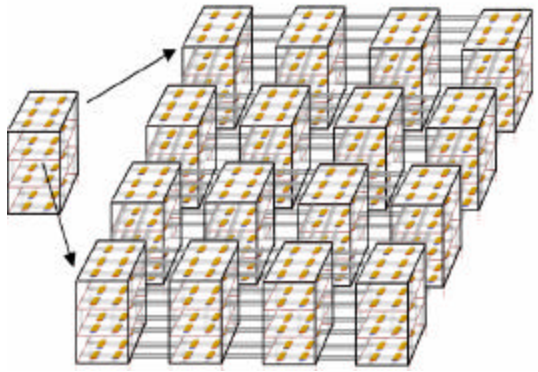

Figure 7. RACK and CUBE SOLAR

Once completed the 3D SOLAR learning machine will have a tremendous self-organizing learning ability based on numerous computing cells (neurons) working in parallel to process the incoming data.

4. SUMMARY

A system level concept of data driven self-organizing learning array (SOLAR) architecture is introduced in this paper. SOLAR is different from traditional neural networks in several important aspects. Firstly, it has a hardware-oriented expandable parallel architecture. Therefore, its speed and learning abilities can be greatly improved compared to traditional neural networks. Secondly, its interconnection number grows linearly with the number of neurons, not like traditional neural networks with a quadratic relationship between the number of wires and the number of neurons; Finally, it has data-driven self-organizing learning hardware. SOLAR architecture is based on simple computing cells (neurons) and flexible routing modules. This architecture is simple, regular, and easily expandable to three dimensions. This simplicity and expandability further favors its hardware implementation.

Based on this architecture, we are developing 3D SOLAR learning machine using 384 high-end VIRTEX XCV1000 FPGA chips. The neurons implemented in SOLAR architecture have self-organizing hardware, which can be dynamically self reconfigurable. With these abilities, we expect that the 3D SOLAR learning machine can perform intelligent tasks such as pattern recognition, prediction and modeling of unknown systems without being programmed beforehand, and find a wide range of applications. To implement this objective, we divide the work into two steps – prototyping using hardware-software codesign, and final design of 3D SOLAR learning machine. SOLAR represents a new idea in hardware design of neural networks. It is thought of as modular and expandable system. It also defines a new breed of dynamically reconfigurable architectures—those that can dynamically reconfigure themselves.

5. REFERENCE

[1] Gert Cauwenberghs, Magdy A. Bayoumi, "Learning on silicon: adaptive VLSI neural systems," Boston, Kluwer Academic, c1999.

[2] Ronald Gary Benson " Analog VLSI Supervised Learning System," PhD Thesis, California Institute of Technology, 1994.

[3] A. Bermak and D. Martinez, "A compact 3D VLSI classifier using threshold network ensembles," IEEE Transaction on Neural Networks, 2002.

[4] J. A. Starzyk and Z. Zhu, "Software simulation of a self-organizing learning array system," The 6th IASTED Int. Conf. Artificial Intelligence & Soft Comp (ASC 2002), Canada.

[5] J. A. Starzyk and T-H.Liu, "Design system of self-organizing learning array," IEEE Int. Symposium on Circuits and Systems (ISCAS), Bangkok, Thailand, May 2003.

[6] J. Becker, A. Alsolaim, M. Glesner, and J. Starzyk, "A parallel dynamically reconfigurable architecture for flexible application-tailored hardware/software systems in future mobile communication," The Journal of Supercomputing, Erratum Vol. 23, 132, 2002.
[7] Nallatech Ltd, "Ballynuey 2 VIRTEX PCI card user guide", 1993-1999.