Available online at www.sciencedirect.com
SCIENCE DIRECT®
Microprocessors and Microsystems 27 (2003) 359–366
MICROPROCESSORS AND MICROSYSTEMS
www.elsevier.com/locate/micpro

Systolic implementation of 2D block-based Hopfield neural network for efficient pattern association

Ming-Jung Seow, Hau Ngo, Vijayan K. Asari*

Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA

Received 23 November 2002; revised 20 February 2003; accepted 25 February 2003

Abstract

A systolic array implementation of block-based Hopfield neural network architecture using completely digital circuits is presented in this paper. The design is based on modelling the energy equation of Hopfield neural network to a systolic (or modular) form. It is shown mathematically that the modified energy equation converges in all circumstances. In addition, it is shown that the architecture provides massive parallelism and can be extended to a larger network by cascading identical chips. The performance of the proposed architecture is evaluated by applying various binary inputs and it is observed that it exhibits the same characteristics of block-based Hopfield neural network in terms of convergence and pattern association.

Keywords: Hopfield network; Pattern association; Modular architecture; Systolic implementation

1. Introduction

Computations in artificial neural network models are developed based on the organizational principles of the functional elements in brain and they employ many simple computational elements that work concurrently to achieve brain-like tasks. Although artificial neural networks have shown great promise, their full potential has yet to be realized as most implementations have been on sequential machines that are unable to exploit the inherent parallelism in these networks. Thus, dedicated chips implementing the neural network in the VLSI form is required to realize the full capability of neural network [1].

It has been observed by many researchers that systolic arrays are very suitable for certain high-speed computations [2]. Systolic architecture introduced by Kung [3,4] is suitable for a large class of regular and symmetric algorithms, such as the multiplication [5] and adaptive filtering [6]. A systolic architecture consists of a large number of processing elements connected together using an interconnection network, which reflects the flow of data among the processing elements. The trick of the architecture is that once a data item has been retrieved from memory, it should be used by all the processing elements, which require it. This helps alleviate the processing–memory communication bandwidth problem experienced even by the fastest von-Neumann machine. Systolic architectures reduce the complexity of the algorithms by exploiting the regularity in the algorithm [7]. The efficiency of an algorithm to be implemented in VLSI is based on the degree of complexity of communication required between the arithmetic elements rather than on the number of interconnection [8]. Thus, structures ideally suited for VLSI are regular and modular with only local short interconnections. Systolic arrays are the simplest of such structures with each element communicating only with its nearest neighbour [9,10]. They have been shown to be capable of implementing powerful computational algorithms using very simple structures made of arrays of identical processing elements. Systolic architecture constitutes a good compromise between time consuming sequential realization and silicon area consuming parallel architecture [11]. For example, in a conventional Hopfield neural network, if N is the number of neurons required and S is the area of an individual physical cell, we need an areas of $O(N^2S)$ for a parallel implementation and we have 1 cycle time for evaluation. On the other hand, a sequential algorithm requires a silicon area of $O(S)$, but the computation time will be $O(N^2)$. A mean solution is

* Corresponding author. Tel.: +1-757-683-3752; fax: +1-757-683-3220.
E-mail address: vasari@odu.edu (V.K. Asari).

0141-933/03/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0141-9331(03)00041-3

provided by the systolic architecture with a silicon area of O(NS) and a computation time of O(N) [12].

In this paper, we present a systolic implementation of 2D block-based Hopfield neural network architecture [13] using fully digital circuits. The proposed architecture requires pre-computed synaptic weights to perform computation for recognition of the input patterns. A block-based architecture for the Hopfield neural network is presented in Section 2. The systolic array implementation of the block-based architecture is described in Section 3. The simulation results and performance evaluation of the new architecture is provided in Section 4 and the conclusions in Section 5.

2. Block-based Hopfield neural network

In this section, we present the development of a block-based architecture for association of $N \times M$ images. To recognize an image containing $N \times M$ pixels using a conventional Hopfield neural network, it is necessary to use a 2D array of fully interconnected neurons. Consequently, it needs N^2M^2 synaptic weights for the implementation of $N \times M$ network and it performs N^2M^2 additions and multiplications in one computational cycle. Hence, training and recognition require considerable amount of computational power if $N \times M$ is large. In the approach by Seow and Asari [13] for image recognition using Hopfield neural networks, an $N \times M$ image is partitioned into N^2/M^2 number of sub-blocks of size $n \times m$, as shown in Fig. 1. Although the number of neurons in this arrangement remains $N \times M$, the number of synaptic weights is reduced to $NMnm$ since each sub-block requires n^2m^2 weights and we have NM/nm such sub-blocks. Thus, each $n \times m$ Hopfield neural network acts independently.

In a 2D Hopfield neural network with $N \times M$ neurons, let the weight, w_{ijkl}, be the interconnection strength from the neuron in the kth row and lth column to the neuron in the ith row and jth column, and the encoding equation is given as [14]

$$w_{ijkl} = \sum_{s=1}^{P} x_{ij}^{s} x_{kl}^{s} \tag{1}$$

for $0 \le i,k \le N-1$ and $0 \le j,l \le M-1$

where $x_{mn}^{s} \in [-1,+1]$ is the input at mth row and nth column of the network from the sth training pattern and P is the number of patterns used for training. The recall of the Hopfield neural network is a combination of calculating the net output and threshold function from each neuron. The net output of the network is computed as

$$\mathrm{Net}_{ij} = \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} w_{ijkl} x_{kl} \tag{2}$$

for $0 \le i \le N-1$ and $0 \le j \le M-1$

and the output is thresholded using a hard limiter

$$y_{ij} = f(\mathrm{Net}_{ij}) = \begin{cases} +1 & \text{if } \mathrm{Net}_{ij} \ge 0 \\ -1 & \text{if } \mathrm{Net}_{ij} < 0 \end{cases} \tag{3}$$

for $0 \le i \le N-1$ and $0 \le j \le M-1$

The recalling process will continue until the network reaches equilibrium. That is, the network said to be converged when the current output is equal to the previous output for all neurons.

In Eq. (1), each neuron is connected to every other neuron through the corresponding synaptic weight. In a neural network for an image processing application, which is presented by Seow and Asari [15], each neuron represents an image pixel and it can be observed that the influence of the neighbouring pixels on a particular pixel is much larger than that from the farther ones. Thus the energy function of the neural network could be modified by incorporating a distance based weighting factor D to reduce the influence of farther neurons on a particular neuron. This reduces

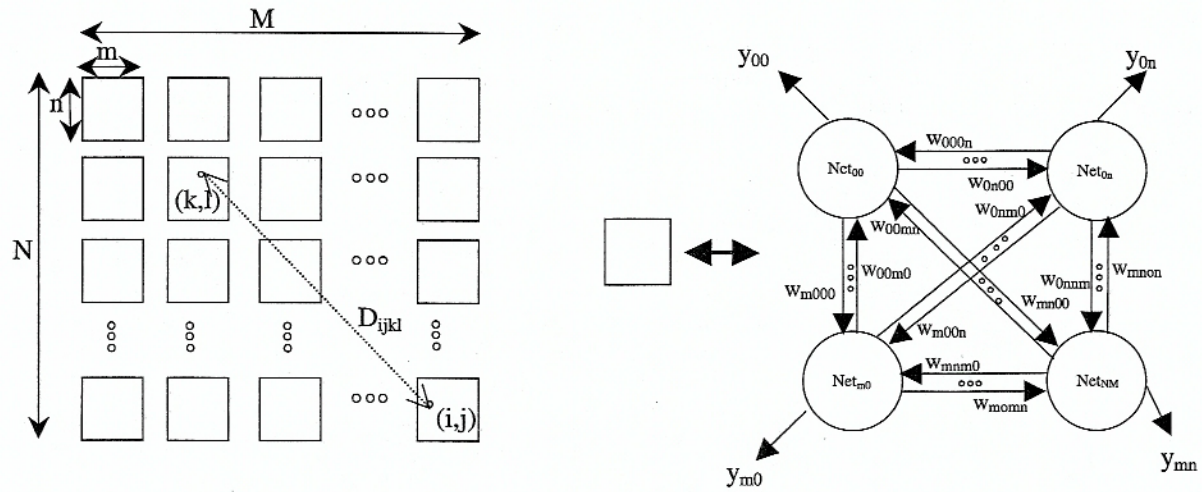

Fig. 1. Block-based neural network architecture.

the value of the synaptic weights from a farther neuron to the neuron under consideration. That is, the weights could be modified by incorporating the D factor as [15]

$$w_{ijkl} = \sum_{s=1}^{P} x_{ij}^{s} x_{kl}^{s} D_{ijkl} \tag{4}$$

for $0 \leq i,k \leq N-1$ and $0 \leq j,l \leq M-1$

where D_{ijkl} is the distance from the (k,l)th neuron to the (i,j)th neuron. The D factor is a controlling parameter in which the relationship between each group of neurons can be distinguished. By specifying the point of reference of each neuron with respect to every other neuron, the D factor can be used for modelling the architecture. That is, the D factor could be used to redistribute the energy function to create a new modular architecture.

2.1. Network architecture based on distance factor

Let the input image of size $N \times M$ be divided into sub-blocks of size $n \times m$ and let each sub-image applied to one of the NM/nm sub-networks with each module working independently. It can be noticed that an $n \times m$ sub-image area and its neighbours in a larger image of size $N \times M$ have very similar intensity values. Let the value of the distance factor D for a cellular structure with cell size of $n \times m$ be expressed as [13]

$$D_{ijkl} = \begin{cases} 1 & \text{if } \eta = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where η is defined by

$$\eta = \max\left\{\left|\left\lfloor \frac{i-1}{n} \right\rfloor - \left\lfloor \frac{k-1}{n} \right\rfloor\right|, \left|\left\lfloor \frac{j-1}{m} \right\rfloor - \left\lfloor \frac{l-1}{m} \right\rfloor\right|\right\}$$

and α is a parameter that governs the effect of neurons from the neighbouring modules and $\alpha \geq 1$. It can be observed that the influence of the neuron in the second cell-neighbourhood onwards on a particular neuron under consideration will be reduced significantly. The network architecture can be modelled as shown in Fig. 1. The selection of the smaller spatial sub-blocks in a larger image can be justified by the fact that there exists a possibility of the presence of similar patterns in the sub-block neighbourhood area. That is, the effective number of patterns to be trained into a sub-network would be smaller when compared with the situation in a total image perspective. One of the major advantages of the D based training algorithm is that all the weights located outside the boundary D are not necessary to be trained. This considerably reduces the number of weights to be trained and hence the training time.

2.2. Energy function of systolic structure

By considering the $n \times m$ sub-block of 2D neural network shown in Fig. 1 into the form of a 1D network, the total energy of an individual block of neurons can be expressed as:

$$E = -\frac{1}{2} \sum_{i=0}^{n\times m-1} \sum_{j=0}^{n\times m-1} w_{ij} x_j y_i \tag{6}$$

Rewriting and redistributing the energy of the network, we get

$$E = -\frac{1}{2} \sum_{i=0}^{b-1} \sum_{j=0}^{(n\times m/b)-1} \sum_{k=0}^{b-1} \sum_{l=0}^{(n\times m/b)-1} \times w_{(j\times b+i)(l\times b+k)} x_{(j\times b+i)} y_{(l\times b+k)} \tag{7}$$

and

$$E = -\frac{1}{2} \sum_{j=0}^{(n\times m/b)-1} \sum_{l=0}^{(n\times m/b)-1} \sum_{i=0}^{b-1} \sum_{k=0}^{b-1} \times w_{(j\times b+i)(l\times b+k)} x_{(j\times b+i)} y_{(l\times b+k)} \tag{8}$$

where b is the number of neurons in each module in the block of neurons. The energy due to a single neuron at $(l \times b + k)$ becomes:

$$E_{(l\times b+k)} = \sum_{j=0}^{(n\times m/b)-1} \sum_{i=0}^{b-1} w_{(l\times b+k)(j\times b+i)} x_{(j\times b+i)} y_{(l\times b+k)} \tag{9}$$

The output function becomes:

$$\mathrm{Net}_{(l\times b+k)} = \sum_{j=0}^{(n\times m/b)-1} \sum_{i=0}^{b-1} w_{(l\times b+k)(j\times b+i)} x_{(j\times b+i)} \tag{10}$$

In Eq. (10), it can be seen that the organization of the output function is in a systolic form. That is, b neurons are grouped together to form $(n \times m)/b$ modules.

2.3. Convergence of the energy function

From Eq. (9), the change in energy in the $(l \times b + k)$th neuron due to its update from $y_{(l\times b+k)}^{\mathrm{old}}$ to $y_{(l\times b+k)}^{\mathrm{new}}$ can be derived as

$$\Delta E_{(l\times b+k)} = -\frac{1}{2}\left(\sum_{j=0}^{(n\times m/b)-1} \sum_{i=0}^{b-1} w_{(l\times b+k)(j\times b+i)} x_{(j\times b+i)} y_{(l\times b+k)}^{\mathrm{new}}\right) - \left(-\frac{1}{2} \sum_{j=0}^{(n\times m/b)-1} \sum_{i=0}^{b-1} w_{(l\times b+k)(j\times b+i)} x_{(j\times b+i)} y_{(l\times b+k)}^{\mathrm{old}}\right) \tag{11}$$

which can be simplified as:

$$\Delta E_{(l\times l+k)} = (y_{(l\times b+k)}^{\mathrm{new}} - y_{(l\times b+k)}^{\mathrm{old}}) \times \left(\sum_{j=0}^{(n\times m/b)-1} \sum_{i=0}^{b-1} w_{(l\times b+k)(j\times b+i)} x_{(j\times b+i)}\right) \tag{12}$$

Now the following three specific cases can be considered:

- If the $(l\times b+k)$th neuron does not change state, i.e. if $y^{old}_{(l\times b+k)}=y^{new}_{(l\times b+k)}$ then by Eq. (12), $\Delta E_{(l\times b+k)}$ will become 0
- If the $(l\times b+k)$th neuron is changed from -1 to $+1$, then $y^{old}_{(l\times b+k)}-y^{new}_{(l\times b+k)}=2$ and according to Eq. (3), Net_{ij} will be greater than or equal to 0 so that the change in energy will be less than or equal to 0, and
- If the $(l\times b+k)$th neuron changes from $+1$ to -1, then $y^{old}_{(l\times b+k)}-y^{new}_{(l\times b+k)}=-2$ and according to Eq. (3) Net_{ij} will be less than 0 so that the change in energy will be less than 0.

Thus, it can be concluded that in all state transitions of the neural network, the energy of the network either remains the same or it decreases. Furthermore, it can be observed that a stable energy minimum state would be achieved when all neurons stop changing their states.

3. Systolic array implementation of block-based Hopfield neural network

A detailed architecture for each module with two neurons is shown in Fig. 2 where x_j represents the jth bit of the pattern and w_{ij} is a weight obtained during training process. Input weights are latched into registers (*reg*) before feeding them to the 2's complement units (*2's compl.*), which has the following functionalities: (1) pass the data from input to output when controlling signal $x_j=0$ or (2) negate input and output the result when controlling signal $x_j=1$, for synchronizing the data. Registers are introduced between 2's-complement units and adders to reduce output delay at each adder. Fig. 3 shows the architecture for an 8-bit neural network that is designed and implemented for the purpose of demonstrating our technique. The network consists of eight neurons, and each neuron is responsible for a single bit in the pattern.

The complexities of modular structures (or systolic implementation) with respect to the non-modular structures (or conventional implementation) are well studied. For instance, three possible complexity discriminations of modular structures with respect to non-modular structures can be the following: (1) neuron counting approach in which the complexity of the network is proportional to the number of neurons in the network, (2) the weight counting approach in which the neurons of the network are all labelled a certain weight and networks are compared based on their total weights, and (3) the dimensional approach in which all neurons of the network take up some space and networks are compared based on the minimum possible volume of the space containing the network.

The neuron counting approach is usually a standard method of complexity analysis because in essence, the amount of memory needed to store the network increases with number of neurons. Modular models usually require extra modules to control the modular learning process or to ordered results of multiple modules. In a non-modular network there is a higher degree of connectivity between all the neurons while in a modular network there are few connections between neurons in different modules. This suggests that modular networks have fewer connections overall, especially when very large problems are undertaken. As a result, a modular network is less structurally complex than a non-modular one as the complexity of the problem increases.

The weight counting approach of complexity is similar to the neurons counting approach measurement. For

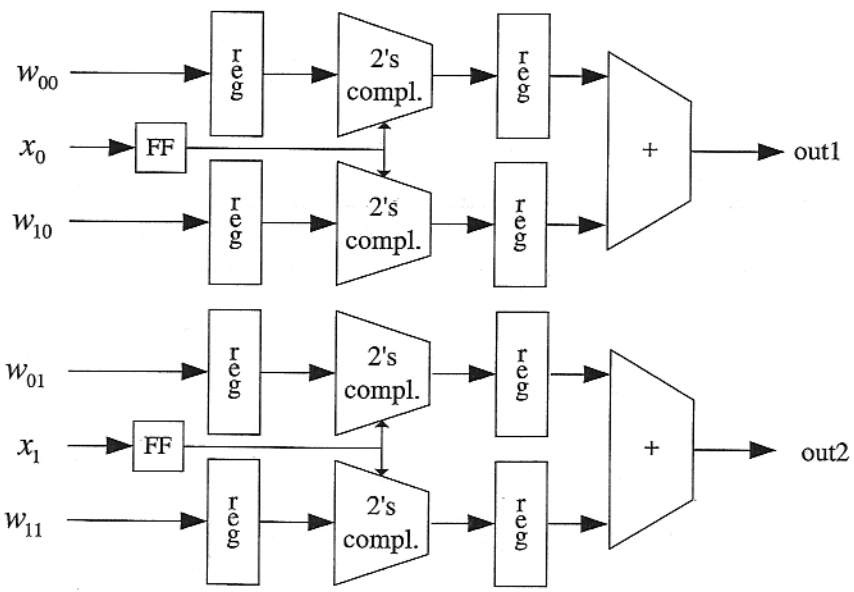

Fig. 2. Each module of neuron architecture.

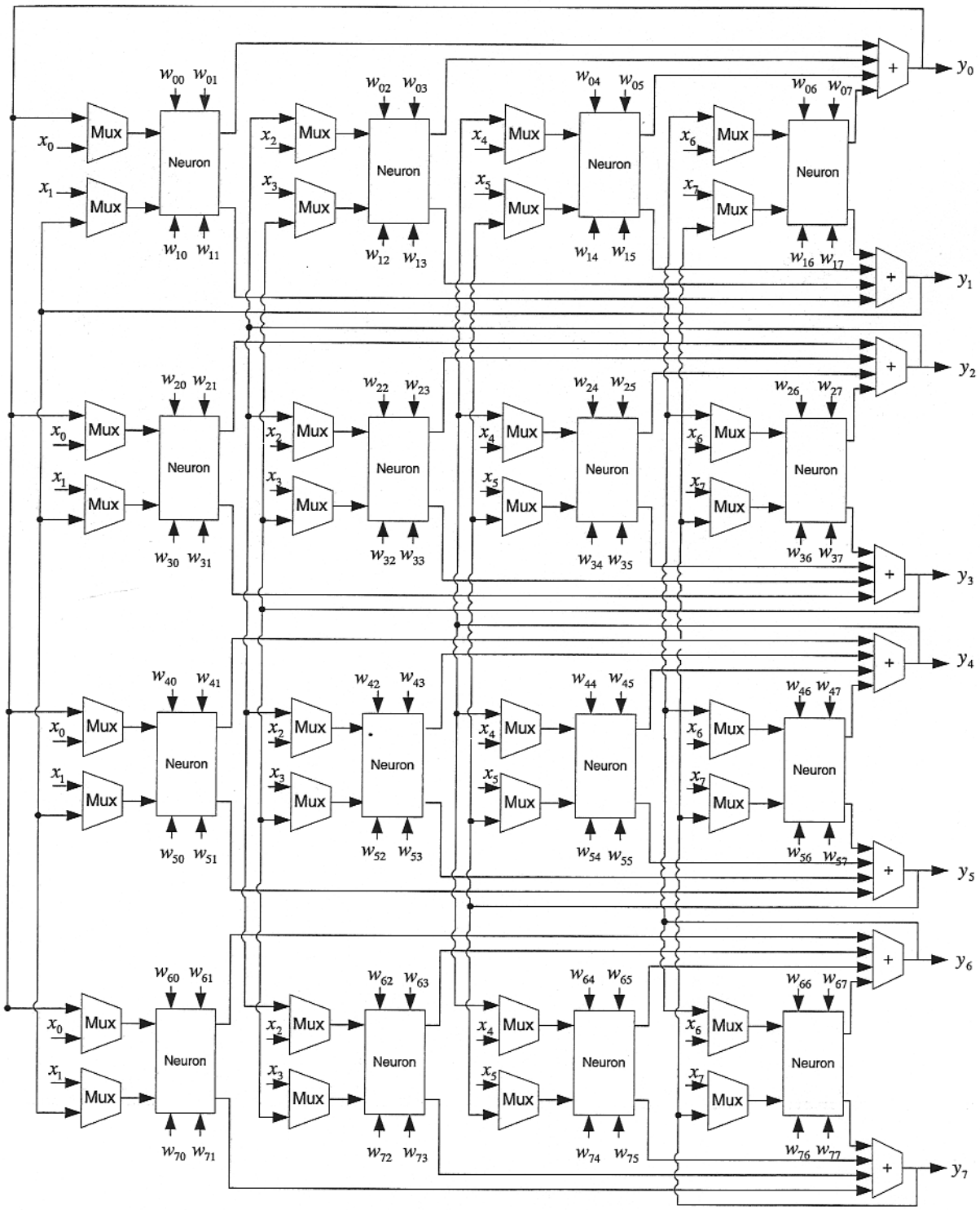

Fig. 3. 8-bit Hopfield neural network architecture.

example, as the number of neurons increases in the network, the number of weights usually increases. However, the weight counting approach provides a more practical comparison of the amount of hardware needed for the network and usually provides a good estimate of the speed of the network. For instance, a typical weight counting approach measurement may have connections proportional to their length. Such a measure would be useful in estimating the amount of hardware needed to hard-wire the network. Since modular networks tend to have fewer connections than an equivalent non-modular network, they will generally

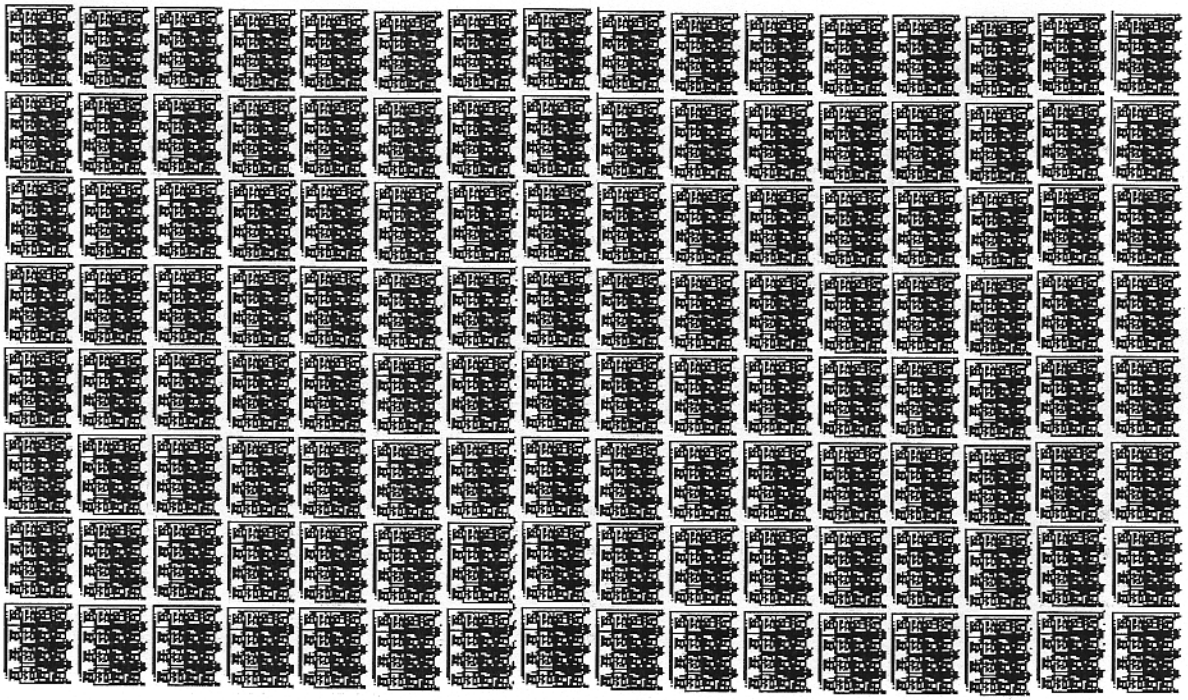

Fig. 4. Internal structure of the VLSI chips of 16 × 8 cells of 4 × 2 neurons.

have a lower number of weights. As a result, the speed of the network is faster.

The dimensional measurement is a very useful indication of the complexity of a network and as in the case with the weight counting measurement, has implications concerning the hardware requirement of the network. For example, the dimension of a neuron is larger than that of the weight connection. So, neurons may take a larger space compared to the weight connections. In addition, weight connections will be required to maintain a certain minimum separation to prevent signal conflicts. With such an analysis, the dimension in a growingly connected network will be primarily determined by the number of neurons in the net as suggested, while the dimension in a highly connected network will be determined by the number of connections. It is likely that a modular network will show a lower overall dimension on most large complex problems, given the lower neuron to connection weight ratio. As a result, the systolic array implementation of modular Hopfield neural network shows less complexity compared to the conventional way of implementation.

4. Simulation results and performance evaluation

For the performance evaluation of the designed Hopfield neural network, the 8-bit architecture for the neural network is implemented and simulated with Altera's Quartus II version 1.1 design tool. The entire architecture is fitted into a FPGA from Altera's APEX family. Specifically, the EP20K60EFC324-1X FPGA is selected for compilation and simulation purposes. Based on simulation results, the maximum clock frequency that can be applied to the network is 83 MHz. The total number of logic cells that the entire architecture resides on is 1941, and the total

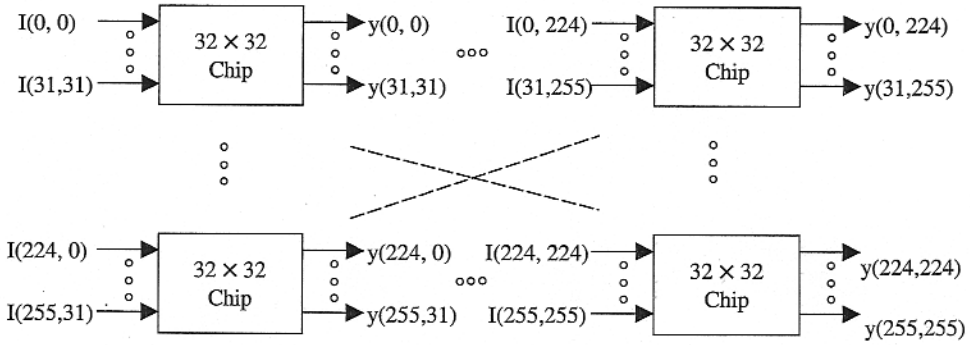

Fig. 5. 256 × 256 neuron expanded architecture with 32 × 32 neuron chips.

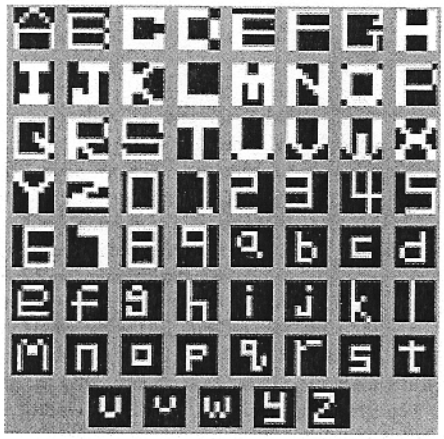

Fig. 6. 32 × 32 simulated characters.

Fig. 7. Reconstruction of the corrupted image of size 32 × 32.

number of flip-flops used for this design is 1696. The latency of each neuron is 2 clock cycles for each input; the results of these neurons are then summed and produce the Net_j as described in Eq. (1).

The block-based architecture used in Fig. 3 for a 4 × 2 neuron network is expanded to construct a VLSI chip for a 32 × 32 neuron architecture as shown in Fig. 4. A larger network can be built with 64 such chips for the reconstruction of an image of size 256 × 256 with the cascaded arrangement as shown in Fig. 5. A set of 32 × 32 simulated characters are used for training the block-based neural network as shown in Fig. 6. The training was done off line and it takes less than a second (0.89 s) to train using MATLAB version 6.1 in a dual processor PC environment with Xeon 1.5 GHz processors. A set of test patterns was generated by introducing various amount of salt and pepper error. To evaluate the performance of the architecture, the 32 × 32 network was initially trained using 61 characters. After adding 30% uniformly distributed random noise to the original images, they were fed to the network for reconstruction. Reconstruction process of the character 'A' is shown in Fig. 7 where the output patterns in successive iterations are shown from left to right. It starts from the noisy image and regenerates the original image from the distorted image in two iterations for a total of 48×10^{-9} s. It can be observed that the quality of the regenerated image is as good as the original image used for training the network.

5. Conclusion

A systolic array implementation scheme for the block-based Hopfield neural network using completely digital circuit has been described. The design is based on a modified energy equation of Hopfield neural network to facilitate the development of a systolic form. It has been shown mathematically that the modified energy equation converges in all circumstances. In addition, it has been shown that the architecture provides massive parallelism and the core architecture can be extended to a larger network by cascading identical chips for training larger images.

References

[1] M. Sindhwani, T. Srikanthan, K.V. Asari, VLSI efficient discrete time cellular neural network processor, IEE Proc. Circuits Syst. 149 (3) (2002) 167–171.

[2] J.I. Guo, C.C. Li, A generalized architecture for the one-dimensional discrete cosine and sine transforms, IEEE Trans. Circuits Syst. Video Tech. 11 (7) (2001) 874–881.

[3] H.T. Kung, Let's design algorithms for VLSI systems, Proc. Caltech Conf. VLSI (1979) 65–90.

[4] H.T. Kung, Why systolic architectures?, IEEE Comput. (1982) 37–46.

[5] C.D. Walter, Systolic modular multiplication, IEEE Trans. Comput. 42 (3) (1993) 376–378.

[6] L.D. Van, W.S. Feng, An efficient systolic architecture for the DLMS adaptive filter and its applications, IEEE Trans. Circuits Syst., Part II 48 (4) (2001) 359–366.

[7] S. Mohiddin, M. Atiquzzaman, T. Dillon, Possibilistic evidential reasoning systems on systolic arrays, Third IEEE Int. Conf. Fuzzy Syst. (1994) 112–117.

[8] M.J. Foster, H.T. Kung, The design of special purpose VLSI chips, IEEE Comput. (1980) 26–40.

[9] S.Y. Kung, VLSI array processors, IEEE ASSP Mag. (1985) 4–22.

[10] S.Y. Kung, An algorithm basis for systolic/wavefront array software, Proc. IEEE Int. Conf. ASSP 25 (1984) A 2.1–A 2.4.

[11] H.T. Kung, W.T. Lin, An algebra for systolic computation, Elliptic Problem Solvers II, 1984, pp. 141–160.

[12] K.V. Asari, C. Eswaran, Systolic array implementation of artificial neural networks, Jl, Microprocessors Microsyst. 18 (8) (1994) 481–488.

[13] M. Seow, K.V. Asari, High storage capacity architecture for pattern recognition using an array of Hopfield neural networks, IEEE Comput. Soc. Proc. 30th Int. Workshop Appl. Imagery Pattern Recognit. (2001) 169–174.

[14] D.W. Tank, J.J. Hopfield, Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit, IEEE Trans. Circuits Syst. CAS-33 (5) (1986) 533–541.

[15] M. Seow, K.V. Asari, Modular architecture for Hopfield network and distance based training algorithm for pattern association, Jl, Neural, Parallel Sci. Comput. 10 (2002) 371–386.

Ming Jung Seow received the BS and MS degrees in computer engineering from the Electrical and Computer Engineering Department at Old Dominion University, VA, USA, in 2001 and 2002, respectively. Currently, he is pursuing the PhD degree in computer engineering at Old Dominion University. His research interests include machine learning and pattern recognition.

Hau Trung Ngo received the BS degree in computer engineering from the Electrical and Computer Engineering Department at Old Dominion University, VA, USA in 2001. Currently, he is pursuing the MS degree in Computer Engineering at Old Dominion University. His research interests include distortion correction in wide-angle camera images and digital design techniques.

Vijayan K. Asari received the BSc (Engng) degree in Electronics and Communication Engineering from the University of Kerala, M.Tech. and PhD degrees in Electrical Engineering from the Indian Institute of Technology, Madras, India. He has been working as an Assistant Professor in T.K.M. College of Engineering, University of Kerala, India. He was with the National University of Singapore from October 1996 to June 1998 where he performed research on biomedical image processing and neural networks. He was with the School of Computer Engineering at Nanyang Technological University, Singapore from July 1998 to July 2000. He joined the Department of Electrical and Computer Engineering, Old Dominion University, Virginia as an Associate Professor in August 2000 and he is currently the director of the VLSI Systems Laboratory at ODU. His research activities include design and development of digital system architectures for real-time embedded systems in the areas of image processing and neural networks. Dr Asari is a Senior Member of IEEE.