

IETE JOURNAL OF RESEARCH

The Institution of Electronics and Telecommunication Engineers

A circular library stamp with the text "LINDA HALL" at the top, "DEC 28 2007" in the center, and "LIBRARY" at the bottom.



President	S Narayana
Vice-Presidents	A K Agarwal P N Chopra Anita G Dandekar

Publications Committee

Chairman	M L Gupta
Co-Chairman	M C Chandra Mouly
Members	H O Agrawal S S Agrawal Smriti Dagur M Jagadesh Kumar Surendra Pal Giridhar R Joshi T K De T S Rathore
Co-opted	S K Kshirsagar KM Paul
Special Invitee	S C Dutta Roy P Banerjee

EDITORIAL BOARD

<u>Chairman</u> S K Koul	<i>Electronic Devices & Components</i> M Jagadesh Kumar V Ramgopal Rao
<u>Honorary Editors</u> <i>Computers and Computing</i> Jayadeva	<i>Instrumentation & Measurements</i> Pawan Kapoor
<i>Communications</i> P V Ananda Mohan Ranjan K Mallik	<i>Circuits and Systems</i> AR Saha Raj Senani
<i>Controls</i> S Mukhopadhyay B Bandopadhyay	<i>Medical Electronics</i> Sneh Anand Vinod Kumar
<i>Electromagnetics</i> S K Koul V M Pandharipande	<i>Opto-electronics</i> K Thyagaraja Subrat Kar
	<i>Power Electronics</i> SP Gupta

Secretary General	V K Panday
Dy Managing Editor	A P Sharma

IETE Journal of Research is published bimonthly by the Institution of Electronics and Telecommunication Engineers. All rights of publication are reserved by the IETE.

Copyright and Reprint permission: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of readers. *Annual Subscription:* Subscription and Advertising rates are available on request and also on IETE website.

IETE Journal of Research welcomes for consideration scientific and technical papers describing original research work on topics of long range interest in all areas of electronics, telecommunications, computer science and engineering and information technology. In addition, indepth tutorial reviews on new and emerging topics in these areas are also welcome. While all the papers submitted go through the same detailed review process, short papers and Letters-to-the-Editor will receive special attention to enable early publication.

Manuscripts may please be submitted in triplicate to the Managing Editor along with a soft copy on CD or through e-mail. Detailed guidelines to authors may be seen in the January issue of the Journal (copy available on request) and also on IETE web site: <http://www.iete.org> under the heading publications.

Address for correspondence:

Managing Editor
IETE
2, Institutional Area,
Lodi Road,
New Delhi 110 003

Telephone : + 91(011) 43538800-899 extn 43

Telefax: + 91 (011) - 24649429

e-mail: ietend@gmail.com
mgeditor@iete.org;

Website: <http://www.iete.org>
<http://www.iete.info>

FREE TO IETE CORPORATE MEMBERS

(Cost of production: Rs 18.00)

COPYRIGHT

It is the IETE policy to own the *copyright* of the scientific and technical papers it publishes on behalf of the authors and their employers, and to facilitate the appropriate reuse of this material by others. Authors are required to sign an IETE copyright transfer form before publication. This form, a copy of which is available in the last January issue of IETE publications, online on IETE Website and also can be sent by post on request. IETE retains the authors and their employers right to reuse their material for their own purposes.

IETE JOURNAL OF RESEARCH

Published bimonthly by the Institution of Electronics and Telecommunication Engineers

SEPTEMBER-OCTOBER, 2007

Vol 53, No 5

CONTENTS

Scanning the Issue	SHIBAN K KOUL	385
Pilot Symbol Assisted Channel Estimation Techniques for OFDM Systems: A Review	REKHA GUPTA AND ADITYA TRIVEDI	387
Real-Time Implementation of Modified Fuzzy Logic Controller for a Non-Linear Quasi-Resonant DC-DC Converter	A ARULSELVI AND UMA GOVINDARAJAN	401
ICA Based Blind Detection for DS-CDMA Systems	JYOTI SAXENA, C S RAI AND P K BANSAL	417
An Efficient Digital Architecture for Principal Component Neural Network and its FPGA Implementation	N SUDHA, C H SIVA SAI PRASANNA AND V KAMAKOTI	425
FPGA Implementation of Gamma Filter for Extracting Fetal Electrocardiogram	K HELENPRABHA AND A M NATARAJAN	433
Time Frequency Analysis and Non-Stationary Signal Classification using PSO based Fuzzy C-Means Algorithm	B BISWAL, P K DASH AND B K PANIGRAHI	441
An Energy Efficient Programmable Hardware Implementation of the Secure Hash 384 and 512 Algorithms	SYED MOHSIN REZA ZAIDI AND M HASAN	451
Detecting Properties of Four-Segment Twin Tandem-Stripe Monolithically Integrated Devices	ABID KARIM	459
Analysis of Fuzzy Techniques and Neural Networks (RBF & MLP) in Classification of Epilepsy Risk Levels from EEG Signals	R SUKANESH AND R HARIKUMAR	465
Parallel Multi-objective Genetic Algorithm for Classification Rule Mining	SATCHIDANANDA DEHURI, ASHISH GHOSH AND RAJIB MALL	475
Precise Measurement of Concentration of Blood Glucose from the Tristimulus Values Obtained from the Color of the Assay in the Course of Clinical Diagnosis	A SIVANANTHA RAJA AND K SANKARANARAYANAN	485
Solutions for Group Domain of Interpretation	MYTHREYI S AND VAIDEHI V	495

Note: The Institution of Electronics and Telecommunication Engineers assumes no responsibility for the statements and opinions expressed by individual authors.

An Efficient Digital Architecture for Principal Component Neural Network and its FPGA Implementation

N SUDHA*

School of Computer Engineering, Nanyang Technological University, Singapore 639 798.
email: sudha@ntu.edu.sg

CH SIVA SAI PRASANNA AND V KAMAKOTI

Department of Computer Science & Engineering, Indian Institute of Technology Madras,
Chennai 600 036, India.

sivapriya535@yahoo.com; kama@iitm.ernet.in

Principal Component Analysis (PCA) finds wide applications in machine vision. The neural network that performs PCA is called Principal Component Neural Network (PCNN). This paper presents a digital hardware design for principal component neural network. The design is efficient in the sense that the learning rule is implemented with a reusable circuit. Results of FPGA implementation of the design show that as many as 500 input vectors can be processed during training phase and 700 input vectors during retrieval phase in a second. Such results are valuable for high-speed applications.

Indexing terms: Principal component analysis, Principal component neural network, FPGA.

1. INTRODUCTION

PRICIPAL Component Analysis (PCA) [1] is a method of representing a data set in more compact form. Principal components are the directions along which the data points in the data set have maximum variance. In mathematical terms, these components are the eigenvectors of the data covariance matrix arranged in the descending order of eigenvalues. PCA has wide applications in machine vision. Some of them are face recognition [2], object recognition [3] and visual robotics [4].

PCA can be realized by a neural network. The neural network that extracts principal components from the input data set is called Principal Component Neural Network (PCNN) [5]. The neural network based approach is advantageous over classical eigenvector techniques due to the following reasons. (1) PCNN extracts the components directly from the input data without the computation of covariance matrix. Hence, memory requirements is less. (2) The number of components can be restricted. (3) The neural networks are adaptive and hence the previously extracted components can be updated online when a

new data arrives (4). Neural networks are generally feasible for hardware implementation due to its regular topology, inherent parallelism and simple arithmetic operations in their learning rules. PCNN has added advantage of its linearity.

PCA is a computationally intensive process. The need for real-time recognition and analysis of images in machine vision calls for the development of high-speed custom hardware. As far as hardware design and implementation on chip for PCNN is concerned, not much work is available in literature. Pulsed circuit architectures based on pulse density modulation and delta-sigma modulation are given in [6] and [7] respectively. The input lines carry analog signals. They are modulated to pulse streams whose frequencies are proportional to the signal voltages. The weights are represented in digital form and stored in registers. The manipulations are done on the frequencies of pulse streams. The resulting outputs are also pulse streams. The idea of modulation is introduced to reduce the wiring in the circuits. However, these circuits are not suitable for processing digital pixel values of images grabbed from digital camera. A systolic array mapping of Adaptive Principal Component Extraction (APEX) is discussed in [5]. An n -input m -output PCNN can be mapped to an array of m processing elements (PEs). Separate

* N Sudha is currently with School of Computer Engineering, Nanyang Technological University, Singapore

systolic arrays are designed for training and retrieval phases. Each PE has memory to store synaptic weights, arithmetic and local control units. It takes $n + 2m - 1$ time units to complete the updating of weights for one input pattern during the training phase and m time units are required between successive input patterns during retrieval phase. The detailed hardware design of a processing element is not given and no implementation result is provided.

In this paper, an efficient hardware design has been proposed for PCNN. The main features of the design are: (1) It is a fully digital one. (2) It has a reusable arithmetic unit that updates the weight vectors corresponding to different principal components one after another. The paper is organized as follows. The learning algorithm of PCNN is briefly given next. Section 3 presents the proposed design and its operation. The FPGA implementation of design and its results are presented in Section 4. The last section concludes the paper.

2. PCNN -ARCHITECTURE AND LEARNING

It is a feedforward neural network with single layer of linear neurons as shown in Fig 1. Each neuron receives the elements of the input vector $X = [x_1, x_2, \dots, x_N]^T$. An N -input M -output network has M neurons and performs principal component analysis of N -dimensional input vectors to extract M principal components. w_{ji} is the weight associated with the i th input to the j th neuron. By Generalized Hebbian

Learning (GHA) algorithm [8], the weight vector of the j th neuron, $W_j = [w_{j1}, w_{j2}, \dots, w_{jN}]^T$, converges to j th principal component of input data vectors. The weights are updated incrementally by the algorithm as below.

$$y_j = \sum_{k=1, N} w_{ki} y_k \quad (1)$$

$$\Delta w_{ji} = \eta y_j [x_i - \sum_{k=1, j} w_{ki} y_k] \quad (2)$$

where η is the learning rate parameter which takes the value in the range (0,1).

2.1. Reformulation of GHA for Efficient Hardware Implementation

The original equation for Δw_{ji} in Eqn (2) requires j multiply and accumulate operations given by $\sum_{k=1, j} w_{ki} y_k$. This can be reduced to one multiply and accumulate operation by reformulating the GHA and hence a time and space efficient hardware implementation of it is achieved. The GHA is reformulated as follows.

$$y_j = W_j^T X_j \quad (3)$$

$$\Delta W_j = \eta y_j [X - \sum_{k=1, j} W_{ki} y_k] = \eta y_j [X - S_j] = \eta y_j X' \quad (4)$$

where

$$S_j = \sum_{k=1, j} W_{ki} y_k = S_{j-1} + W_j y_j \quad (5)$$

$$X' = X - S_j \quad (6)$$

Here, in the computation of each element of S_j for a particular j , only one multiply and accumulate operation is performed. The updated weight vectors for the input X are given by

$$W_j^{new} = W_j^{old} + \eta y_j X' \text{ for } j = 1 \text{ to } M \quad (7)$$

The network has two phases: (1) training and (2) retrieval. During training, the weights are updated for a given input and during retrieval, the outputs are computed for a given input to the trained network.

3. HARDWARE DESIGN

The design of PCNN hardware is shown in Fig 2. The weights storage unit REGSETS W stores the weight vectors, W_1 to W_M , associated to M neurons. The storage elements are arranged in a pipelined fashion as shown in the figure. The arithmetic unit has logic circuits for computing y_j , S_j , X' and W_j^{new} (as in Equations (3)-(6)) and storage for X , η and S_j , namely REGSET X , REG η and REGSET S_j . Due to

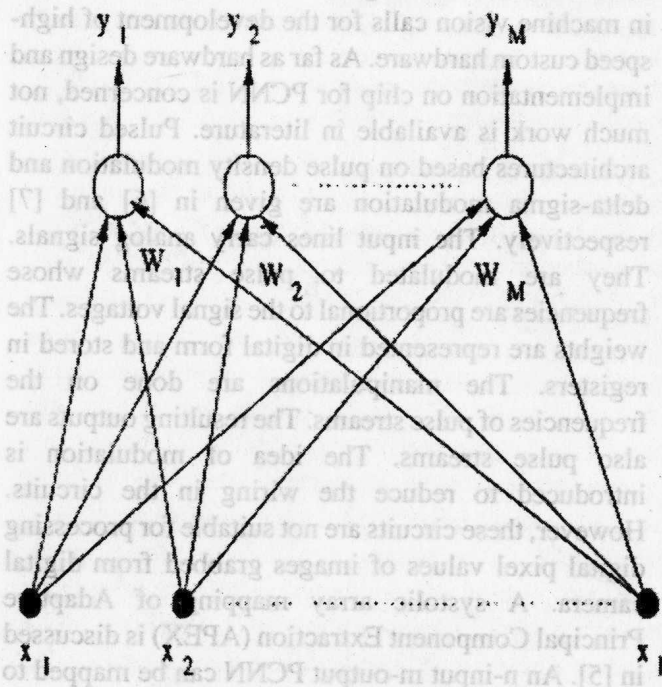
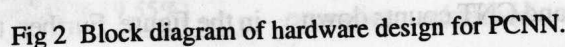


Fig 1. Structure of PCNN

The flow of data in the design is as follows. During the training phase, at j th iteration, y_j is computed first by finding the inner product of input data vector X and the contents of storage unit REGSET W_1 that stores the elements of vector W_j to be updated. S_j is then computed using S_{j-1} , y_j and W_j . S_j is stored in REGSET S_j for the computation of S_{j+1} . Once S_j is known, X' is computed with S_j and X . W_j^{new} is finally updated using W_j^{old} , X' , η and y_j . It is pushed into storage unit REGSET W_M . Simultaneously, the contents of all other storage units in REGSETS W are pushed forward. After M iterations, the individual storage unit in REGSETS W stores the corresponding updated weight vectors for a given input X .

The details of each computational unit in the arithmetic unit that implements an N -input M -output PCNN are given below. Each data element is represented as a floating point number with p -bit mantissa and p -bit exponent in two's-complement form.

The multiplication $x_i w_{ji}$ of two floating point numbers in Equation (3) requires a p -bit Booth multiplier for the multiplication of mantissas and a p -bit adder for the addition of exponents. p clock cycles are required for Booth multiplication and N multipliers and N adders. The summation in the equation is performed by an adder N times and hence



n cycles are required.

Compute S_j :

The multiplications W_{jv} in Equation (4) are implemented using N Booth multipliers and N adders in parallel. It takes p clock cycles to perform these multiplications. The results of these multiplications are then added to the elements of S_{j-1} using N adders in parallel.

Compute X' :

The subtraction of two vectors in Equation (5) is implemented using N subtractors.

Compute W_j :

In order to implement Equation (6), the multiplication ηy_j is first performed using one Booth multiplier and one adder. p clock cycles are required for this multiplication. The result of this multiplication is then multiplied to each element of X' , which is realized using N multipliers and N adders in parallel. This takes p clock cycles. The resultant vector is then added to W_j^{old} using N adders.

The entire arithmetic unit requires about $N + 4p$ clock pulses which constitute one iteration. A global clock GCLK is used for the iterative computation. Its period should be greater than the time taken for one complete iteration. The ASM charts giving the hardware algorithms for training and retrieval phases are shown in Figs 3 and 4.

Training phase

First, initialization of some registers are done. REG η is initialized to a value between 0 and 1. REGSETS W is initialized with random values. The state machine representing the hardware design for training (T-SM) is in initial state T_0 . Then the data vectors from the input data set are presented one after another and the weights are updated for each data vector X . REGSET X is loaded with the elements of X during the first N clock pulses of GCLK. A counter CNT is used for this purpose. It is initialized to N . At each clock pulse i of GCLK, the i th location of REGSET X , say $[\text{REGSET } X]_i$, is stored with the input x_i and CNT is decremented. T-SM is in State T_2 until CNT becomes 0. Once CNT becomes 0, REGSET S_j is initialized to all zeros and a counter CNT is now initialized to M . The T-SM now reaches State T_1 . In the successive M clock pulses of GCLK, the weights get updated starting from W_1 to W_M and CNT counts down

to 0. In any clock pulse j of GCLK, y_j is computed first followed by the computation of S_j , X' and W_j . The values of S_j and W_j are stored in REGSET S_j and REGSET W_j at the clock trigger. Simultaneously, all other register contents in REGSETS W are shifted. The T-SM reaches State T_2 and it continues to be in this state until CNT becomes 0. Once CNT becomes 0, the next input can be placed. The weights converge to principal components of input data set when they are updated repeatedly for the given set several times. The training phase for an input vector takes $(N + M)(N + 4p)$ clock pulses.

Retrieval phase

Initially, REGSETS W stores the principal components that have been computed during training phase. The state machine for retrieval (R-SM) is in the initial state R_0 . The counter CNT is initialized to N and the input data vector X is received and stored in N clock pulses of GCLK as in the training phase. The R-SM continues to be in State R_1 while receiving the inputs. Once X is received, CNT is set to M . In each successive clock pulse j of GCLK, y_j is computed. REGSETS W is circularly shifted and CNT is decremented at clock trigger. The R-SM reaches State R_3 and it continues to be in this state until CNT becomes 0. Once CNT becomes 0, the next input can be placed. The retrieval phase for an input vector takes $(N + M)(N + p)$ clock pulses.

4. FPGA IMPLEMENTATION

The hardware design of a 64-input 16-output PCNN was coded in Verilog hardware description language. Each parameter in the design is represented as a floating point number with an 8-bit mantissa and an 8-bit exponent. The functional behaviour of the design was tested in ModelSim (a package for functional simulation of VLSI design). The principal components were extracted from a given data set by the GHA algorithm implemented in Matlab. The components computed by the hardware algorithm for the same data set were compared with them for correctness.

The design was mapped on to one of the target devices of Xilinx FPGA using Leonardo Spectrum design tool. The specifications of the chosen target device are as follows: family - VIRTEX; device - XCV1000BG560. The floorplan of the device for the design is shown in Fig 5. The portions of the FPGA device used for implementing the design can be seen in the figure. Further, the interconnections between the

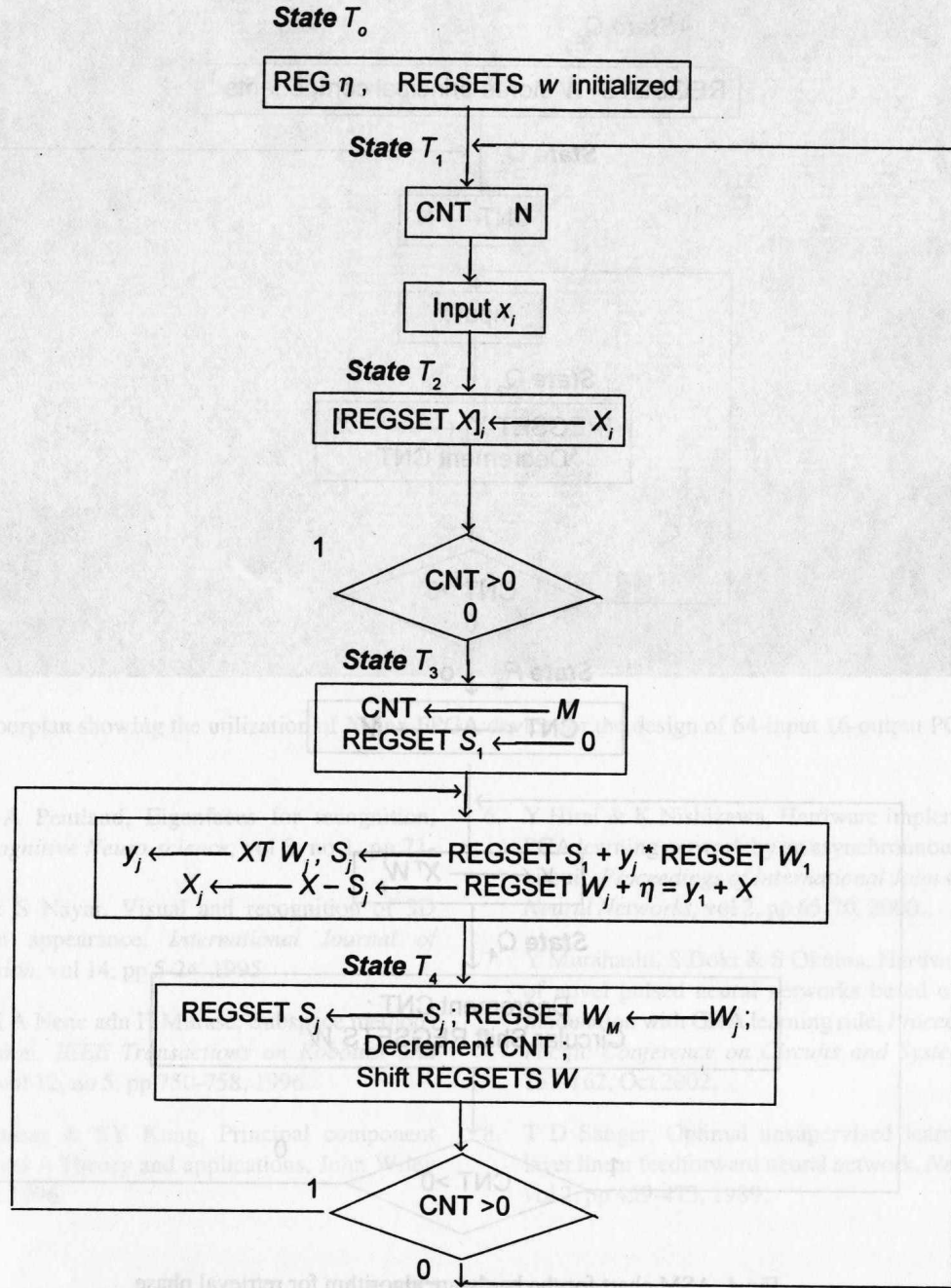


Fig 3 ASM chart for the hardware algorithm for training phase

components of the device utilized for the design can be seen. The device utilization is summarized in Table 1. The slice of the Virtex device contains two logic

TABLE 1: Device utilization for the design of 64-input 16-output PCNN

Component	Usage
IOBs	72 out of 404 (17.82%)
Function Generators	6404 out of 24576 (26.06%)
CLB Slices	3202 out of 12288 (26.06%)
Dffs or Latches	2346 out of 24576 (9.55%)

cells, which realizes the combinational logic of the design. Two such slices comprise one configurable logic block (CLB). The input/output blocks (IOBs) provide the interface between the device pins and the CLBs. The device has 1 million gates.

The time taken for training (T_{tr}) and retrieval (T_r) phases for an input data vector is computed as follows. Let t_c denotes the period of basic clock in seconds. T_{tr} and T_r are given by $(N + M)(N + 4p)t_c$ and $(N + M)(N + p)t_c$ seconds respectively. In the case of 64-input 16-output PCNN design, the maximum frequency of operation obtained through implementation is

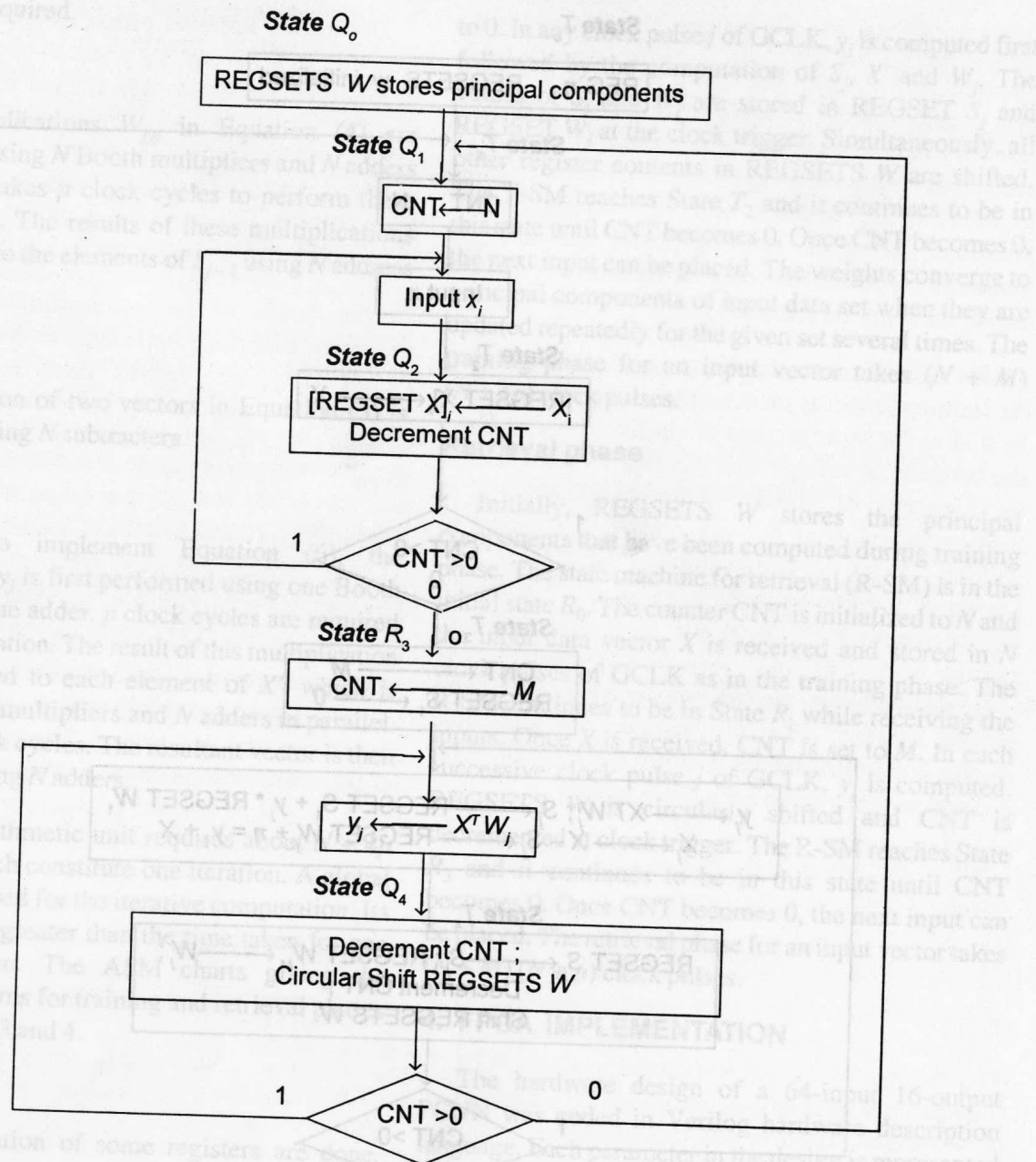


Fig 4 ASM chart for the hardware algorithm for retrieval phase

4.1 MHz. Hence, $T_{tr} \geq 1.873$ ms and $T_r \geq 1.4$ ms. This results in processing upto 535 and 714 data vectors per second during training and retrieval phases of the network respectively.

5. CONCLUSIONS

This paper presents an efficient hardware design for principal component neural network. It extracts M principal components from N_i data vectors of N -dimension. The design has a common arithmetic Unit that is reused M times to implement the learning rule of the network. The HDL code of the design is easily scalable for larger input size and for extracting

more components. The design of a 64-input 16-output PCNN has been implemented in a Xilinx FPGA device and it runs at a clock speed of 4.1 MHz. It takes 1.8 ms and 1.4 ms to process an input during training and retrieval phases respectively. This results in processing more than 500 input data vectors per second during each of the phases and hence the design is suitable for applications like video surveillance and high-speed image communication.

REFERENCES

1. I Jolliffe, Principal component analysis, Springer-Verlag, New York, 1986.

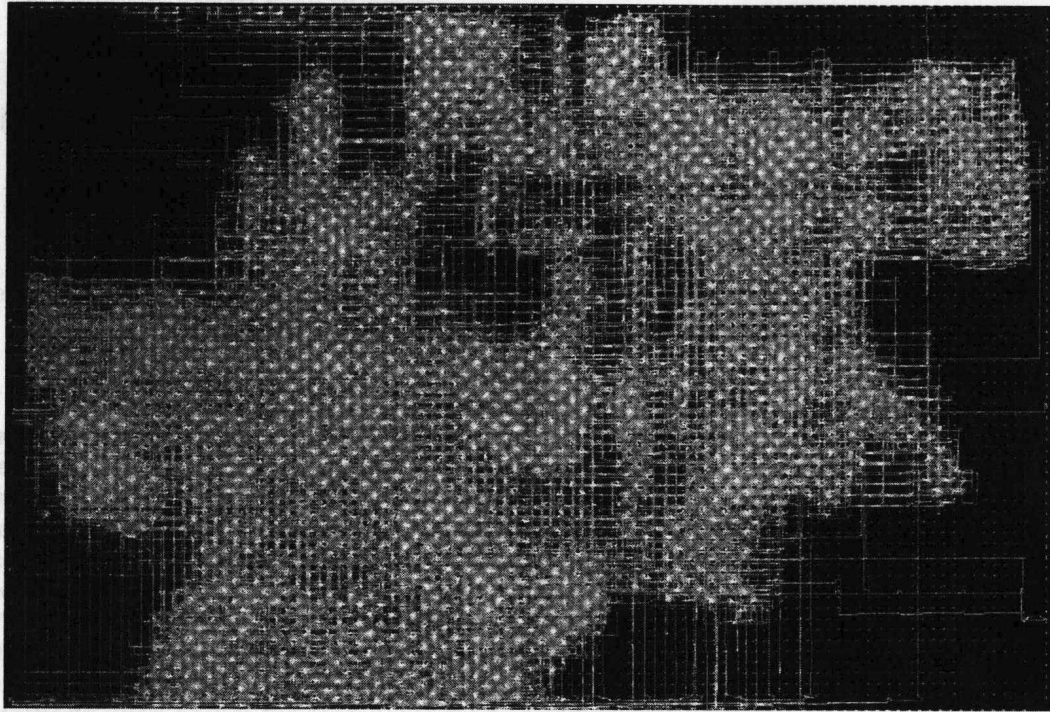
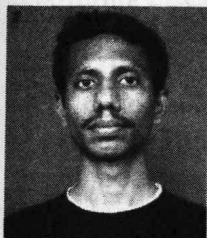


Fig 5 Floorplan showing the utilization of Xilinx FPGA device for the design of 64-input 16-output PCNN

2. M Turk & A Pentland, Eigenfaces for recognition, *Journal of Cognitive Neuro-science*, vol 3, no 1, pp 71-86, 1991.
3. H Murase & S Nayar, Visual and recognition of 3D objects from appearance, *International Journal of Computer Vision*, vol 14, pp 5-24, 1995.
4. S K Nayar, N A Nene and H Murase, Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, vol 12, no 5, pp 750-758, 1996.
5. K I Diamantaras & SY Kung, Principal component neural networks – Theory and applications, John Wiley and Sons, Inc 1996.
6. Y Hirai & K Nishizawa, Hardware implementation of a PCA learning network by an asynchronous PDM digital circuit, *Proceedings of International Joint Conference on Neural Networks*, vol 2, pp 65-70, 2000.
7. Y Murahashi, S Doki & S Okuma, Hardware realization of novel pulsed neural networks based on delta-sigma modulation with GHA learning rule, *Proceedings of Asia-Pacific Conference on Circuits and Systems*, vol 2, pp 157-162, Oct 2002.
8. T D Sanger, Optimal unsupervised learning in single layer linear feedforward neural network, *Neural Network*, vol 2, pp 459-473, 1989.

AUTHORS



Siva Sai Prasanna Ch received the MS degree in Computer Science and Engineering from the Department of Computer Science and Engineering, IIT Madras. Currently, he is a Design Engineer at Atheros India LLC, Chennai. His areas of specialization include computer vision, VLSI and neural networks.



V Kamakoti received the PhD degree in Computer Science and Engineering in 1995 from the IIT Madras. At present, he is an Associate Professor at the Department of Computer Science and Engineering, IIT Madras. His areas of specialization include VLSI design, computer architecture and theoretical computer science.

*

*

*

*

*

*

*

*