3-D VLSI Architecture Implementation for Data Fusion Problems Using Neural Networks

Tuan A. Duong, David Weldon, and Tyson Thomas
Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA 91109

ABSTRACT

This paper gives an overview of hardware implementation techniques employed in solving real-time classification problems using Neural Network, Principle Component Analysis (PCA), and Independent Component Analysis (ICA) techniques. The first part of the paper reviews digital, analog, and hybrid strategies for hardware implementation, outlining their advantages and disadvantages. The second part focuses on dedicated VLSI chips developed at the Jet Propulsion Laboratory (JPL).

A flexible neural network chip with 64 neurons and a 64x64 synaptic weight array with 8-bit resolution is first presented. This chip can be theoretically cascaded to form a larger network, connected in parallel to improve dynamic range or resolution, or connected in a loop to create a feedback neural network. A second neural network chip is presented that was fabricated using Silicon-On-Insulator (SOI) technology. This second chip operates at 1.5V, has neurons with variable transfer functions, and has completely compatible inputs and outputs, allowing simple and direct cascading and feedback. A 64x64 synaptic weight array chip is then introduced that has 8-bit resolution and a time response of less than 250ns. This chip was stacked to obtain a cube of 64 chips with an estimated data processing speed of 10^{12} operations per second.

A data input chip called the Column Loading Input Chip (CLIC) was designed, fabricated in 1.0μm CMOS technology, and tested. The chip can take 64x64 digital bytes and convert them into 64x64 analog inputs to a 3-D parallel processing cube. The CLIC was designed to raster through a large image window, taking a new 64-byte column or row of data from the main image every 250ns. The cube processes this data using PCA or ICA techniques and passes its output to a neural network classifier.

In the cube architecture, power consumption is one of the most important concerns and has, so far, inhibited designs of larger arrays. However, recent SOI technology seems capable of improving major aspects of performance by providing power consumption reduction, latch-up avoidance, and mixed signal noise reduction. A new 3-D architecture is proposed which is similar to the original cube but is more robust for stacking and easier to test, and its application to a hyperspectral sub-pixel classification problem is discussed.

I. INTRODUCTION

At JPL, we have developed a variety of chips that can be used as building blocks for hardware computation of general-purpose algorithms germane to sensor fusion. Our building block chips are cascadable to create larger networks that were necessary for some of our recent applications [1,2]. In addition, many of the chips are stackable in a third dimension to achieve increased parallelism, providing the computational power necessary to solve problems such as real-time spatio-temporal target recognition and Hyperspectral sub-pixel classification. Our latest 3-D chip stacks have been designed to provide computational power on the order of 10^{12} operations per second [3-5].

Section II discusses the hardware implementation strategy used in most of our chips, and explains why our approach is superior to the alternatives. Section III is an overview of the latest building block chips that we are currently using to create powerful prototype 3-D architectures. Section IV will show how the 3-D computational architectures created using our building block chips might be used to solve hyperspectral sub-pixel classification problems. The architecture presented uses Principal Component Analysis (PCA) [6] or Independent Component Analysis (ICA) [7-10] techniques to estimate end members, and then classifies these estimated end members using an artificial neural network.

II. IMPLEMENTATION STRATEGY

In order to accomplish real-time sensor fusion, fundamental operations such as addition, subtraction, and multiplication must be implemented in hardware. If artificial neural networks are to be used, the neuron transfer function must also be realized in hardware to achieve adequate speed. These operations have traditionally been implemented in primarily digital or primarily analog hardware [11,16-18], but we have developed hybrid implementations that retain the advantages of each approach while eliminating or minimizing their weaknesses [1,3].

Fully digital implementations such as the CNAPS board by Adaptive Solutions [11] are attractive for a number of reasons. First of all, digital memory allows for very robust long-term storage of synaptic weights, while digital computation has extremely high noise immunity. In addition, because of the binary nature of digital signals, very fast devices can be used without consideration for their linearity or accuracy. There is also a large amount of flexibility inherent in digital processing, allowing the implementation of nearly any desired architecture with as much precision as is required. This flexibility, however, does not usually include massively parallel implementations, especially those that are scalable. Digital implementations typically occupy a large amount of active die area as well, and have fairly high dynamic power consumption. The architectural limitations coupled with increased power consumption at high clock rates actually limit most digital implementations to relatively slow overall throughput, in spite of the high operational speed of the individual devices.

In contrast to digital implementations, analog techniques can be used to implement fully parallel architectures that are easily scalable. They are also capable of achieving higher throughput with lower power consumption and less die area than digital implementations. Unfortunately, they suffer from low noise immunity and their weight storage mechanism often requires refresh circuitry to maintain accurate values over long periods of time [12]. Alternative

approaches to analog memory, such as floating gate technology [19], eliminate the need for refresh circuitry, but they do not have arbitrary precision and cannot be updated with sufficient speed [13]. After learning, however, neural networks can tolerate relatively poor accuracy [14], so the noise and precision limits of analog computation may not be critical. In general, analog circuitry appears to be much more suitable than digital circuitry for high-density 3-D applications, but the difficulty of realizing refresh circuits across a 3-D chip stack is significant enough to warrant the use of an alternative approach.

In order to capitalize on the suitability of analog circuitry for 3-D architectures while maintaining the stability and accuracy of digital weight storage, JPL has adopted a hybrid approach. Synaptic weights are stored digitally, thereby eliminating the need for refresh circuitry while ensuring adequate time response during learning. Synaptic outputs are represented as analog current signals that can be easily combined with any number of other outputs using only a common wire. This leads to an architecture in which multiplication is performed by Multiplying Digital to Analog Converters (MDACs); addition/subtraction is the result of KCL along the output wire; and neurons are implemented as non-linear I-to-V converters. The overall result is more compact and faster than digital circuitry, but without the noise sensitivity and long-term instability of analog weight storage.

III. JPL HARDWARE

This section outlines the integrated circuit building blocks developed at JPL for hardware artificial neural networks and 3-D parallel data processing architectures. It also outlines some specific 3-D architectures designed to solve real-time spatio-temporal problems.

Neural Network Building Blocks

NN64 Chip:

In our early work, we fabricated a flexible neural network chip in 0.8μm CMOS called the NN64, whose architecture is depicted in Fig. 1. This chip contains:

- 64 voltage inputs ranging from 2.0 V to 3.0 V
- a 64x64 array of 8-bit bipolar synapses (+/-127)
- 64 variable gain neurons
- programmable bypass switches to select the summed current or neuron voltage output

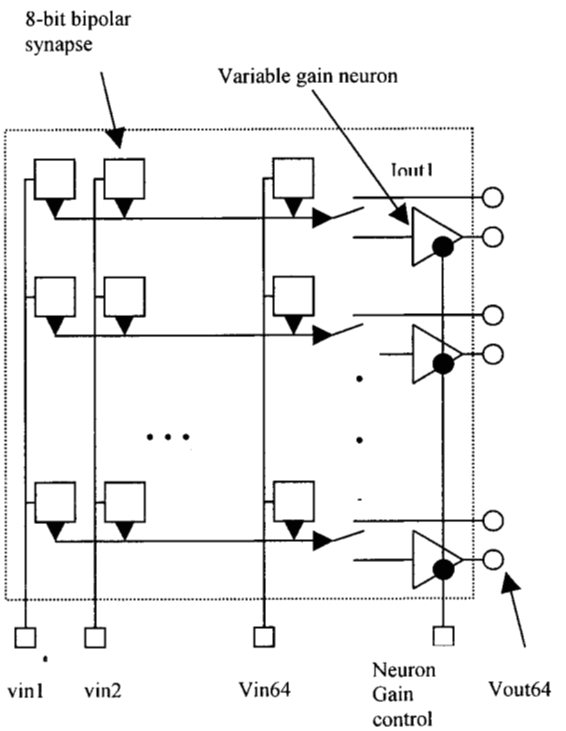

Fig. 1: Block diagram of the NN64 architecture.

The NN64 chip can be used as a basic neural building block in either a feedforward or feedback configuration. It is

potentially expandable horizontally and vertically, allowing for a much larger network to be created if necessary. It can also be connected as if it were stacked in a third dimension, which effectively increases the weight resolution and dynamic range of the network's synapses. Cascading in the third dimension also allows for multiple sub-networks to process the same input data. 3-D architectures are discussed later in the section.

SOICANN Chip:

We recently fabricated a Silicon-on-Insulator Cascadable Artificial Neural Network (SOICANN) using MIT Lincoln Labs' 0.25μm CMOS process, under sponsorship from DARPA's Low Power Electronics Program. Although this chip is not as large as NN64, it was designed to be immediately cascadable without the need for interface circuitry. This allows multiple chips to implement an arbitrarily large feedforward or feedback network. Each chip accepts 8 inputs, has 8 hidden units, has 8 output neurons, and implements a constructive network architecture based on Cascade Error Projection [21-24]. Each hidden unit can be viewed as a single neuron hidden layer with complete connection to all previous hidden layers as well as to all inputs. All neurons are programmable so as to exhibit a logistic transfer function, a gaussian transfer function, or to be bypassed completely. In addition, the output of each neuron can be either voltage or current, making the chip completely cascadable without limitation. SOICANN uses a 1.5V power supply and simulations show an input step response of less than 200 nS through a single chip. As of this writing, the SOICANN die are being shipped back to JPL and have not yet been tested.

3-D Building Blocks

Syn64 Chip:

In [2] and [3], we reported a 64x64 synaptic weight array with 8-bit resolution that was fabricated in 1.0μm AMI CMOS technology. This chip was intended to be a stackable building block for a 3-D architecture. It uses a 5V power supply and requires 64 analog voltage inputs that range from 2.0 to 3.0 volts. These inputs are then multiplied fully in parallel with 64 weight vectors that are stored digitally using an 8-bit bipolar format (+/- 127). The result of each multiplication is a current signal that is summed along one of 64 different lines. The details of this chip can be found in [2].

Column Loading Input Chip:

3-D architectures require large arrays of parallel data as input. To achieve this, the "Column Loading Input Chip" (CLIC) was designed. The CLIC receives a 64x64 array of 8-bit digital data and converts it into a 64x64 analog voltage array in 250ns [5] using a large array of compact digital to analog converters (DACs). The digital input array usually corresponds to an input sub-image of a larger main digital image that is being processed. Inside the CLIC, the sub-image can be shifted up, down, or right one position while a new column or row is loaded from the main image. This allows the sub-window to be moved around inside the main image without having to reload the entire CLIC. The CLIC was fabricated in a 0.8μm HP CMOS process. Its voltage output array is available on 4,096 metal3 pads, each of which measures 66x66μm^2. Each DAC cell in the CLIC array is 101.6x101.6μm^2.

3-D Architectures

Our first cube was created using a vertical stack of sixty-four Syn64 chips

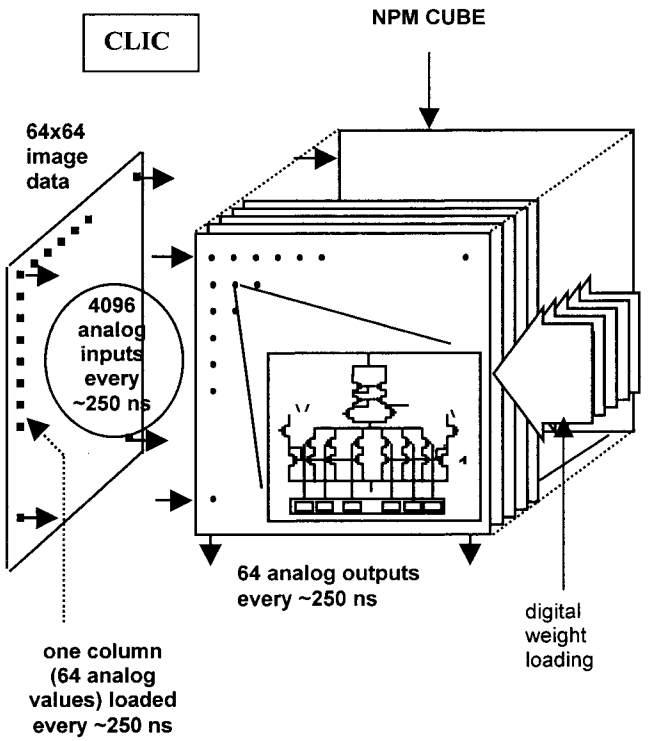

Fig. 2: 3-Dimensional Artificial Neural Network-M (3-DANN-M). In this figure, CLIC provides 64x64 fully parallel analog inputs with a new column (64-bytes) in every 250ns while the NPM performs parallel template matching.

forming a 3-D Neural Processing Module (NPM) intended for massively parallel real-time template matching for spatio-temporal problems[3]. At first an IR focal plane array, which required operation at .77K[4], was mated to the top of the NPM to provide direct parallel analog input. Later the IR focal plane array was replaced with the CLIC in order to exploit the full computational power of the NPM cube with more versatility. Fig. 2 shows a particular implementation called 3-DANN-M where the CLIC obtains a 64x64 sub-window from a 256x256 digital image and sends this sub-image to the NPM cube in a fully parallel fashion. The sub-image is multiplied with sixty-four templates in the cube where each template is a 64x64 array of 8-bit bipolar weights. All multiplications are performed in parallel every 250ns making the cube theoretically capable of 10^{12} operations per second. Fig. 3 shows a photo of the 3-DANN-M.

Current work is focused on combining the Syn64 and the CLIC functionality into a new stackable building block for the next generation 3-DANN-R. This will eliminate the difficult task of bonding the CLIC to the top of the NPM, which greatly simplifies the cube production process while enhancing testability and observability.

Several challenging problems surfaced during the design of the NPM and the CLIC. Specifically, power consumption

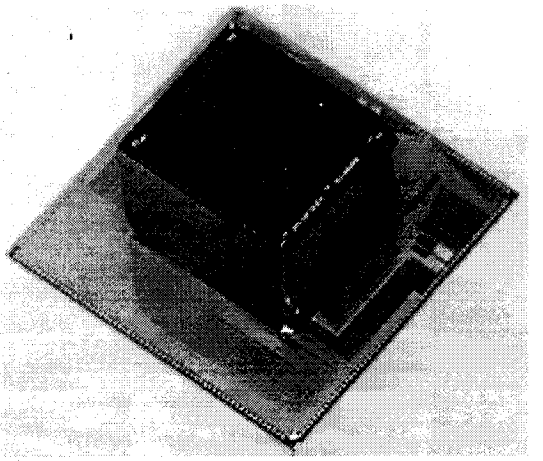

Fig. 3: 3-DANN-M. This photo shows the CLIC on top of the 3-DNPM cube sitting on the motherboard.

and mixed signal noise are so critical that they may prevent us from thinking ahead to larger arrays and bigger chip stacks. Fortunately, Silicon-On-Insulator (SOI) technology is an attractive option that has the potential to neutralize both issues. SOI technology allows us to reduce power consumption drastically by reducing the supply voltage from 5 V to 1.5 V. It also reduces mixed signal noise by eliminating the substrate coupling of digital switching noise to analog components. Since SiO_2 is a good heat conductor it should also

ameliorate thermal management within a 3-D chip structure. The SOICANN chip was designed in SOI in part to evaluate these potential advantages. We have also fabricated Winner-Take-All (WTA) circuits using the same SOI process as SOICANN and the test results are very encouraging [15].

IV. APPLICATION

A lot of interest has recently been generated by research on Hyperspectral Sensor Imaging (HSI), which can be considered as a special case data fusion problem. Real-time classification of hyperspectral data can be extremely useful for certain types of target recognition and terrain or composition identification. In addition, NASA has recently expressed interest in a space-based, low power, miniature system that is capable of classifying hyperspectral data.

The majority of current research on HSI focuses on sub-pixel detection. Unfortunately, the raw sensor data tends to be very noisy and inconsistent which makes the classification problem more difficult. PCA combined with neural networks has already demonstrated some success in sub-pixel detection [20]. Since each pixel contains data from multiple bands, all of which is available in parallel, there is a big advantage to massively parallel processing.

In our application, each pixel contains data from 224 bands of differing wavelengths. In the 3-DANN architecture it takes 4 columns, each containing 64 bands, to process a single pixel. Since neighboring pixels may have relevant information for detecting a particular sub-pixel, a 3x3 window of pixels (see Fig. 4) can be analyzed in parallel, requiring 36 columns of input data. Let the number of desired end members be N, and let $W_1, W_2, \ldots, W_N$ be the orthogonal vectors for PCA or independent vectors for ICA that are to be used for separating the end members. After processing by the 3-DANN cube, the results can be described as follows:

$$Y = \begin{bmatrix} W_1^T \\ \ldots \\ W_N^T \end{bmatrix} X$$

X is an input vector representing one pixel (224x1). This input vector can be physically stored in 4 columns of the CLIC. W_j is a weight vector stored in the columns of 3-DANN. The output vector Y, which is an estimated decoding of the end members, is then sent to the NN64, which can be used as a neural network classifier. This procedure improves detection rates by exploiting the neural network's ability to learn and generalize. Finally a WTA can select the best classification match. Fig. 5 shows the system architecture.

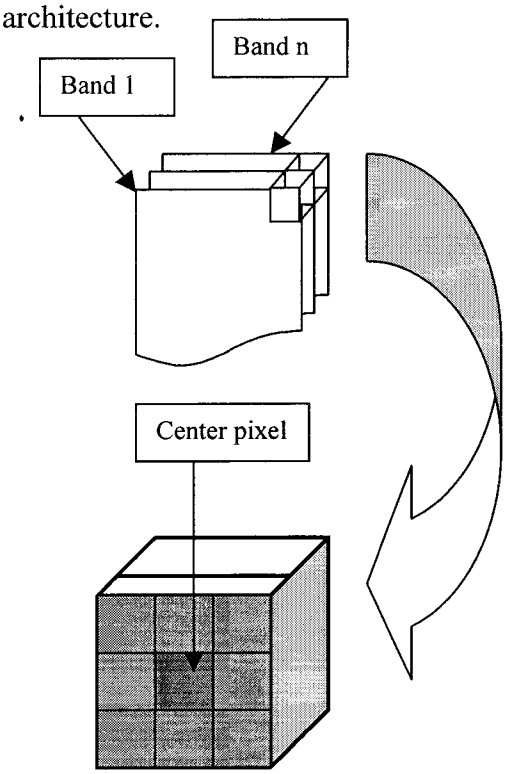

Fig. 4; Structure of hyperspectral data. In this figure, hyperspectral data consists of n=224 bands per pixel. A 3x3 sub-window is analyzed to classify the center pixel.

From hardware designed at JPL, we are able to construct a discrete system for HSI analysis. Even though it is a discrete system, it is still extremely compact and low power in comparison to other state of the art systems capable of performing hyperspectral analysis; *e.g.* banks of Super Harvard Architecture RISC Computer (SHARC) DSP processors [25].

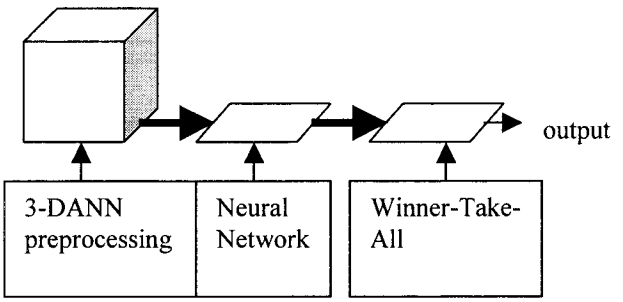

Fig. 5: Full 3-D architecture for real-time HSI sub-pixel classification problem. 3-DANN operates as a linear pre-processor to separate end members, NN64 is the neural network processor to enhance classification, and WTA selects the best match.

V. CONCLUSION

A number of powerful chips developed at JPL for use as building blocks in 3-D systems were presented briefly, along with a description of the 3-D architectures themselves. We also discussed the potential of using SOI technology to overcome two of the most difficult challenges inherent in 3-D chip stacks. Finally, we showed how our 3-D architecture might be applied to solve a hyperspectral sub-pixel classification problem.

Our proposed 3-D architecture is extremely compact and features very high-speed operation with a power consumption of less than 5 Watts. Such a system should satisfy NASA's requirements for high-density, low power, space-based systems capable of synthesizing large amounts of varied sensor data.

Acknowledgments:

The research described herein was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was jointly sponsored by the Ballistic Missile Defense Organization/Innovative Science and Technology Office (BMDO/IST), and the National Aeronautics and Space Administration (NASA). The authors would like to thank Drs T. Daud, A. Thakoor and S. Udomkesmalee for useful discussions.

References:

1. T.A. Duong, et al., "*Learning in neural networks: VLSI implementation strategies*," In: *Fuzzy logic and Neural Network Handbook*, Chap. 27, Ed: C.H. Chen, McGraw-Hill, 1996
2. T.A. Duong, et al., "Cascaded VLSI neural network building-block chips for map classification," *Government Microcircuit Applications conference* 92, Las Vegas, NV, Nov 9-12, 1992, pp 45-46.
3. T.A. Duong, S. Kemeny, T. Daud, A. Thakoor, C. Saunders, and J. Carson, "Analog 3-D Neuroprocessor for Fast Frame Focal Plane Image Processing," *The Industrial Electronics Handbook*, Chap. 73, Ed.-In-Chief J. David Irwin, CRC PRESS, 1997.
4. T.A.Duong, et al. "Room and Low temperature performance of high speed neural network circuits," *Electrochemical Society Proceedings*, pp. 369-377, vol. 97-2, May 2-4, 1997, Montreal, Canada.
5. T.A.Duong, et al., "the results of 64x64 Analog Input Array for 3-Dimensional

Neural Network Processor," *IJCNN'98* in Anchorage, Alaske, pp. 49-53, 98.
6. R.O. Duda, "Pattern Classification and Scene Analysis," John Wiley & Sons Inc., 1973.
7. A.J. Bell and T.J. Sejnowski, 1995, "An Information maximization approach to blind separation and blind deconvolution," Neural Computation, 7, 1129-1159.
8. Pham, D.,T., 1996, "Blind separation of instantaneous mixture of source via an independent component analysis.", IEEE trans. On Signal Processing, vol 44, pp2768-2779.
9. Attias and Schreiner, 1998, "Blind source seperation and deconvolution.", Neural Computing, vol 10, number 6
10. Hyvarinen and Oja, 1997, "A fast fixed point algorithm for . independent component analysis.", Neural Computation, vol 9, pp 1483-1492.
11. D. Hammerstrom, "A Massive Parallel Architecture for Cost-Effective Neural Network Pattern Recognition, Image Processing, and Signal Processing Applications," Digest paper in GOMAC Conf. , pp 281-284, 1992, Las Vegas, Neveda.
12. Eberhardt S.P., T.A. Duong, and A.P. Thakoor, "A VLSI analog synapse 'building-block' chip for hardware neural network implementations," *Proc. 3rd Annual Symposium on Parallel Processing*, Fullerton, CA, March 1989, Vol. 2 pp. 257-267.
13. D.A. Kern, Experiments In Very Large-Scale Analog Computation, Ph.D. Thesis California Institute of Technology, 1992
14. J. Hopfield, "The effectiveness of analogue 'neural network' hardware," *Network* 1;27-40 (1990) (IOP Publ. Ltd., U.K.).
15. T.A.Duong, et al., "Winner/Loser-Take-All On SOI Technology for Neural Networks", SPIE, Orlando Florida, May 1998.
16. W.-C. Fang, B.J. Sheu, O.T.-C. Chen, and J. Choi, "A VLSI Neural Processor for Image Data Compression Using Self-Organizing Netorks," *IEEE Transactions on Neural Networks*, vol. 3, no. 3, pp.506-518, 1992.
17. A. Jayakumar and J. Alspector, "On-Chip Learning in Analog VLSI Using Simulated Anealing," *Digest of Papers in Government Microcircuit Applications Conference*, Las Vegas, NV, 1992, pp.277-280.
18. S. P. Eberhardt, T.A. Duong, R. Tawel, F.J. Pineda, and A.P. Thakoor, "A Robotic Inverse Kinematics Problem Implemented on Neural Network Hardware with Gradient-Descent Learning," *Proceedings of the Second ISTED International Symposium on Expert Systems and Neural Networks*, M.H. Hamza, ed., Hawaii, August 15-17, 1990, pp.70-73.
19. M. Holler, S. Tam, H. Castro, and R. Benson, "An Electrically Trainable Artificial Neural Network (ETANN) with 10,240 "Floating Gate" Synapses," *Proceedings IEEE International Joint Conference on Neural Networks*, vol. 2, June 18-22, Washington, 1989, pp.191-196.
20. A. Howard, C. Padgett, and K. Brown "Intelligent Target Detection in Hyperspectral Imagery," *Thirteenth International Conference on Applied Geologic Remote Sensing*, Vancouver, British Columbia, Canada, 1-3 March 1999.
21. T.A. Duong, A. Stubberud, T. Daud, and A. Thakoor, "Cascade Error Projection-A New Learning Algorithm," Proceedings Int'l IEEE/ICNN in Washington D.C., vol. 1, pp. 229-234, Jun. 3-Jun 7, 1996.

22. T.A. Duong, "Cascade Error Projection- An efficient hardware learning algorithm," Proceeding Int'l IEEE/ICNN in Perth, Western Australia, vol. 1, pp. 175-178, Oct. 27-Dec 1, 1995(Invited Paper).
23. T.A. Duong and T. Daud, "Cascade Error Projection With Low Bit Weight Quantization For High Order Correlation Data", Accepted to IJCNN'99 in washington D.C., 1999.
24. T.A. Duong and T. Daud, "Cascade Error Projection_a Learning algorithm for hardware implementation", Accepted to IWANN'99 in Alicante, Spain.
25. Analog Promotion SHARC2 Overview, http://www/analog.com/new/ads/html/SHARC2/started.html.