6. CONCLUSIONS

Fully parallel, analog neural network hardware systems could indeed meet the processing throughput demands required by various real-time applications in image processing. The results for the map separates data classification problem demonstrate ibis capability. For example, *without* the consideration of the speeds of the AT-VME adapter and the host computer, the VMENA subsystem takes 0.83 seconds to classify all 61K pixel map segment (for an effective computational throughput of 73K pixels/s) whereas the ISAbus based neural network takes about 1.83 seconds to process the same map segment (for an effective computational throughput of 33K pixels/s). However, the host computer and the AT-VME adapter still ingest a great deal of processing time. Though the speed of [he host computer can be easily rectified by replacing the host with a faster computer, the primary bottleneck which limits the data transfer throughput between the host computer and the NNDB still remains at both the AT-VME adapter interface and at the input DACS They arc largely compensated for try the inclusion of three novel features of the VIMB: (1) the local data transfer technique, (2) the. data buffering scheme, and (3) reading the outputs (or winners' indices) during the processing of the input vectors, Contributing to the overall processing speedup of the VMENA individually, each of these techniques is employed in order to utilize the VME data bus to its fullest extent by efficient means of augmenting its data transfer throughput, buffering the data transfers, and time-multiplexing the input/output write/read operations.

Increased computational throughput could be attained by eliminating the data congestion problem via an adoption of a VME64 bus architecture and of a burst mode data transfer between the host computer and the VIMB. The processing speed of the VMENA would be improved by at least a factor of four. Furthermore, another speed-up could be achieved with a simple chip level redesign. The existing NN chips require three consecutive executions to load a single synaptic weight value. A redesigned chip would need only one, hence a three-fold improvement.

7. 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 acknowledge Dr. S. Narathong for his technical expertise and effort in the design, debug, and test of the VME interface. The authors also wish to thank Mr. H. Langenbacher for useful technical discussions and assistance. If rny beauty is perceived, it is the reflection of the professionalism of the co-authors. If any discrepancy is found. it is the sole responsibility y of the principal author.

8. REFERENCES

1. T. X. Brown, M.D. Tran, T.A. Duong, T.Daud, and A.P. Thakoor, "Cascaded VLSI neural network chips: hardware learning for pattern recognition and classification," *Simulation*, 58, 340-347, Special Issue on 'Neural Networks': Model development for applications. 1992.
2. T. J. Graettinger, N, V. Bhat, and J. S. Buck, "Adaptive contr 01 with NeuCOP, the Neural Control and Optimization Package," World Congress on Computational Intelligence, IEEE International Symposium on Involutionary Computation, pp.2389-2393, June 26-July 2, 1994.
3. T. A. Duong, S. P. Ike]-tlardt, T. Daud, and A. Thakoor, "Learning in neural networks: VLSI implementation strategies," In: *Fuzzy Logic and Neural Networks Handbook*, Ed: C. H. Chen, McGraw-Hill (In press).
4. S. P. Eberhardt, A. Moopenn and A.P. Thakoor, "Considerations for hardware implementations of neural networks," in Proc. of the 22nd Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA 1988.
5. R. Tawel, "Learning in analog neural net work hardware., " *Computers Electronic Engineering*, *vol. 19,* No. 6, pp.453-467, 1993.
6. A. P. Thakoor, "Hardware implementations and applications of neural network architectures," ARPA/DoD/NASA Final Report, JPI,1)-12518 (internal document), Pasadena, California, May 1993.

training set's confidence interval, shows that the hardware performs competitively comparable to NN simulation (9 1.2%).3 The original map and the hardware output map after a completion of a hardware-in-the-loop training arc depicted in Figure 5(a) and (b), respectively.

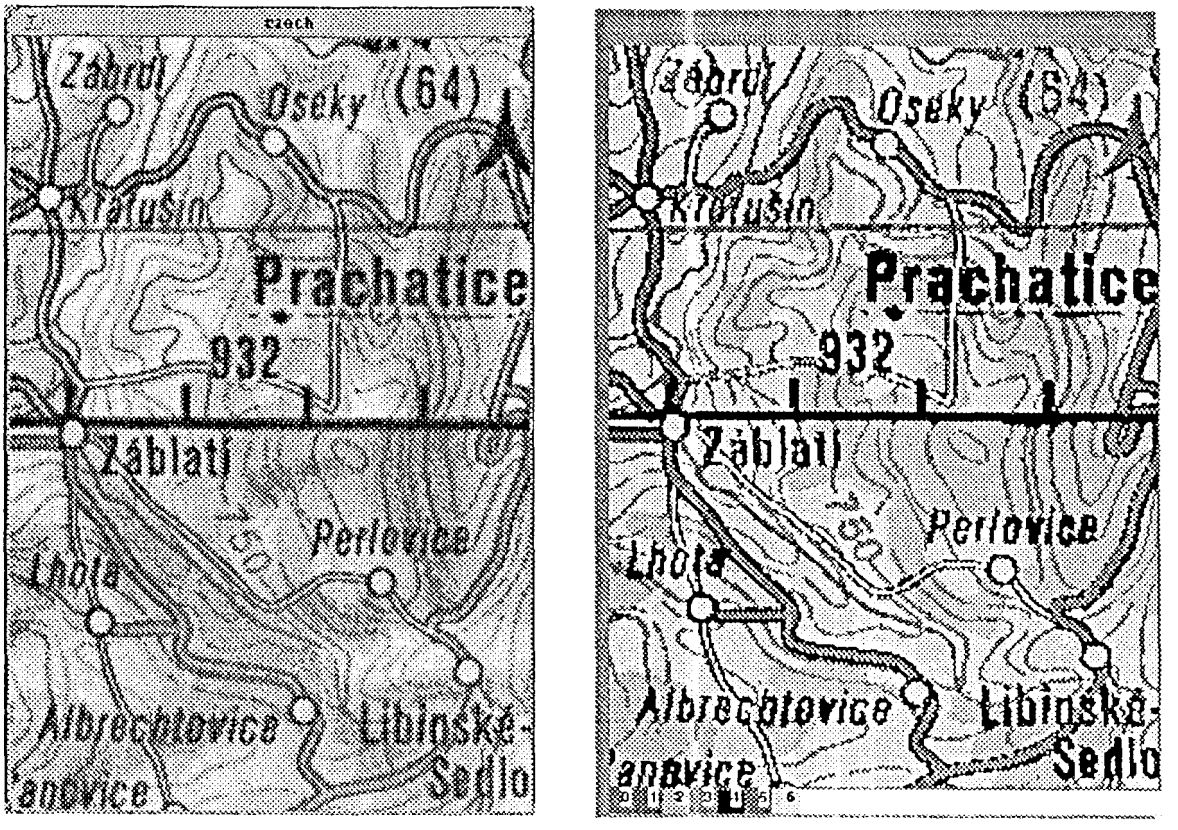


Figure S. The original map image (a) and the hardware neural network output (b),

The discrepancy between the software and the hardware NN results is attributed to several sources. Experimentally, one observable source, as mentioned above, is the impact of the hardware learning rate on the accuracy. If the hardware learning rate deviates by a few percent from its best value. then the accuracy would be substantially affected. Another noticeable source is the computation of the derivative in hardware, which is done by perturbing the biases at the neuron inputs and taking the difference in their neuron outputs. If the perturbed bias value differs from its best value a little, the overall accuracy would be greatly affected. This effect could possibly be due to the limited synaptic resolution. Moreover, not only the training set size but also its contents affect the overall accuracy. For example., two training sets A and B of the same size, i.e. 2300 pixels, but different contents are used to train the network. If set A has more redundant copies of the input/target pairs than set B, then. the result of set A will be less accurate than that of set B. Obviously, this effect is due to the fact that the training set A is the subset of the training set B. Finally, although the learning utilizes all the available bits of hardware precision, the weight updates occasionally err in either magnitude or sign or both due to noises from several layers of wire-wrap connections of and between the VIMB and the NNDB; such a stochastic error can limit the learning capability and alter the overall accuracy.

In: *Applications and Science of Artificial Neural Networks*, Steven K. Rogers and Dennis W. Rock, Eds., SPIE Proc. 2492, pp. 10-22, 1995.

An invited article

A high speed VMEbus based analog neurocomputing architecture for image classification

Mua D. Tran, Tuan A. Duong, Raoul Tawel, Taher Daud, and Anil P. Thakoor

Center for Space Microelectronics Technology
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., Pasadena, CA91109-8099
e-mail: mua@brain.Jpl.Nasa.Gov

ABSTRACT

To fully exploit the real-time computational capabilities of neural networks (NN) - as applied to image processing applications - a high performance VMEbus based analog neurocomputing architecture (VMENA) is developed. The inherent parallelism of an analog VLSI NN embodiment enables a fully parallel and hence high speed and high-throughput hardware implementation of NN architectures. The VMEbus interface is specifically chosen to overcome the limited bandwidth of the PC host computer Industrial Standard Architecture (ISA) bus. The NN board is built around cascadable VLSI NN chips (32x32 synapse chips and 32x32 neuron/synapse composite chips) for a total of 64 neurons and over 8K synapses. Under software control, the system architecture could be flexibly reconfigured from feedback to feedforward and vice versa, and once selected, the NN topology (i.e. the number of neurons per input, hidden, and output layer and the number of layers) could be carved out from the set of neuron and synapse resources. An efficient hardware-in-the-loop Cascade Backpropagation (CBP) learning algorithm is implemented on the hardware. This supervised learning algorithm allows the network architecture to dynamically evolve by adding hidden neurons while modulating their synaptic weights using standard gradient-descent backpropagation. As a demonstration, the NN hardware system is applied to a computationally intensive map-data classification problem. Training sets ranging in size from 50 to 2500 pixels are utilized to train the network, and the best result for the Hardware-in-the-loop learning is found to be comparable to the best result of the software NN simulation. Once trained, the VMENA subsystem is capable of processing at approximately 7S,000 feedforward passes/second, resulting in over twofold computational throughput improvement relative to the ISAbus based neural network architecture.

Keywords: VMEbus, analog neurocomputing, neural network, supervised learning, image classification.

1. INTRODUCTION

Neural networks (NN) have emerged as a powerful new methodology in information processing applications. What distinguishes neural networks from mainstream methodologies is that they are trained from exemplars, and this has made them effective at "learning" and performing arbitrarily complex nonlinear functional estimations, mappings, and even control for a variety of applications.[1,2] Neural networks fall under the general rubric of massively parallel fine-graincd systems. As such, their highly parallel architecture is typically composed of numerous identical and functionally simple computational elements (called neurons) which communicate with one another via the mediation of variable strength pathways (synaptic elements). Although software implementations of these NN models are usually adequate, applications in image processing are computationally intensive and frequently require real-time classification or analysis. These requirements necessitate the mapping of the NN formalism to dedicated hardware solutions. and there now exist a broad range of NN hardware approaches and implementations. In our approach, we maximize the computational throughput by building a fully parallel and reconfigurable neuroprocessor system based On custom analog NN chips developed at JPL.[3-6] An extensive survey of different and currently available NN chips and boards has been compiled by Duong et al.[3]

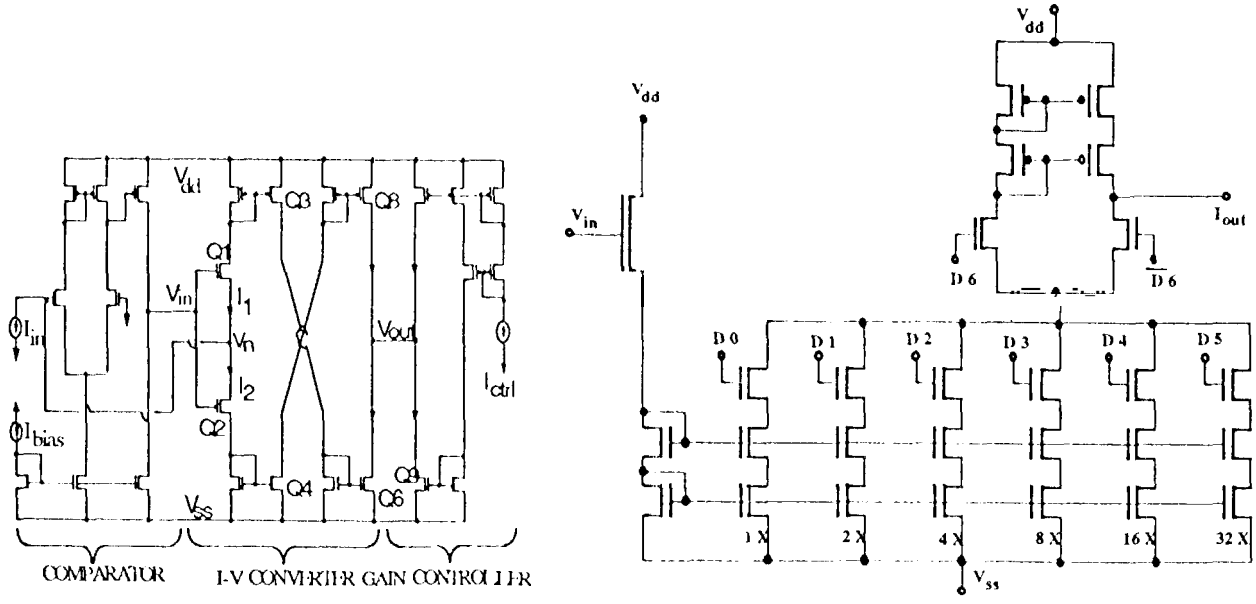
As a demonstration for our NN hardware, we selected the map-data classification problem. This application is motivated by the need 10 effectively analyze, classify, and archive high-resolution digitized pixel map-data renditions of actual ink-printed maps as stored on CD-ROM cartridges. A classification of these dense

and cluttered map images into categorical, discernible and useful features could significantly reduce the storage requirements, and their data arc in the most suitable form, with which further analysis can be carried out.

in the following sections. we begin with a brief background on both the hardware and software pertinent to the implementation of our neuroprocessor. This is followed by a top level description of the VMEbus Neurocomputing Architecture (VMENA). The performance of the VMENA, along with its novel features and technical issues, is discussed in Section 4. Finally, the results of a map data classification problem arc reported in Section 5.

2. CASCADABLE "BUILDING BLOCK" CHIPS

Analog NN hardware systems have emerged as an impel tant class of computing devices. A variety of large scale analog VLSI circuits can now be fabricated. circuits which fully exploit the computational throughput and implement the true asynchronous parallelism of neural network architectures. The real scalability and cascadability power of the synapse chips and the neuron/synapse composite chips in our existing library of VLSI neural network "building block" arc fully utilized and exploited. Furthermore, from this scalability feature, the resolution of the hardware synaptic weights generated from hardware-in-the-loop learning can be increased to augment the high resolution requirements of the software synaptic weights.



(a) (b)

Figure 1. (a) A wide-range, variable-slope sigmoidal neuron circuit. (b) A synaptic cell circuit consisting of cascoded current mirrors and V/I common to all the synapses along a row in the matrix (left), static latches D_{0S} arc used to program the synaptic weights (bottom,right), and current steering block, using latch D_6 as a current polarity regulator (lop).

The VMENA is based on a hybrid of both analog-digital synapse chips and rleurorl/synapse composite chips. This hybrid design utilizes digital memories to store the synaptic weights and multiplying digital-to-analog converters (MDAC) to perform the analog multiplication. Each of the neuron/synapse composite chips consists of a 32x32 crossbar matrix of synapses with the diagonal synapses substituted by 32 neurons, and each of the synapse chips contains a 32x32 crossbar configuration of synapses. Since processing is performed asynchronously and in a

fully parallel fashion and since the circuit settling time is of the order of one microsecond. a chip consisting of approximately 1000 synapses is capable of a computational throughput of nearly a gigs connections per second (GCPS). The neuron design (Figure l(a)) incorporates a wide-rarrgc variable gain and sigmoid-like transfer characteristics. The synapse design (Figure l(b)) is based on a Static Random Access Memory (SRAM) with 7 bits (6-bit + sign bit) of resolution and utilizes two-quadrant current multipliers. Each synapse. therefore, appears as a digital memory so that writing a digital value 10 an appropriate memory location constitutes a change of that synapse's weight. The functional descriptions of these two types of chip sets arc detailed elsewhere.[7,8]

Since these chips are cascadable, the output ports of onc chip can directly be connected to the input perks of another chip without the usc of any intervening glue logic. This means that, in the synapse chips, the input values arc encoded as voltages (for case of propagation) and output values are encoded as currents (for case of summation). Conversely, in the neuron/synapse chips, the input values arc encoded as currents and output values are encoded as voltages. Because of their cascadability, such chips provide a means to scale up a neural network architecture to any arbitrary size. For instance, a fully feedback 64x64 7-bit synapses with 64 neurons neural network architecture can be assembled from two synapse chips and two neuron/synapse chips cascading in a plane. In addition to being able to tile larger neural networks from these cascadable analog NN chips, the synaptic resolutions of such networks can also be scaled up by piggybacking, another synapse chip onto the existing synapse chip in the z-dimension.[9] Moreover, as the "learning" information is contained within the synaptic weights, the synaptic resolution is important for both the training phase and the recall phase. Typically, NN learning algorithm requires high (-12 to 16 bits) synaptic resolution for an effective convergence. In general, high resolution weights generated using an off-line software simulation can not be ported directly to a low precision analog hardware. To compensate for such undesirable discrepancies between the software and hardware synaptic resolutions. hardware-in-the-loop-learning becomes necessarily indispensable.

A supervised hardware-in-the-loop Cascade Backpropagation (CBP) learning algorithm is implemented on the VMENA. The details of the CBP arc presented elsewhere.[3,10] This learning algorithm is similar to the cascade correlation algorithm described by Fahlman and Lebiere[11] with two variations: the methodology used for the weight calculation between the inputs and the outputs, and the training technique of the new hidden units. The weights between the inputs and the outputs arc initially calculated using pseudo-inverse procedure and down-loaded to the hardware while the new hidden neurons arc trained using the standard gradient-descent backpropagation.[12] This algorithm inherits most of the important properties of the "growing" algorithms, including freezing the old weights to minimize the learning computations. I'bus, it is quite adept at learning a variety of difficult tasks, including, as wc will demonstrate in Section S, the map separates data classification problem.

3. VMEbus NEUROCOMPUTING ARCHITECTURE (VMENA)

This section centers around a description of the VMENA and describes its essential components in details. As depicted in Figure 2 and shown in Figure 3, the VMENA system consists of four major inter-dependent functional blocks: (1) An Intel386/16 MHz host computer, (2) a Bit3's AT-VME adapter board, (3) a Neural Network Daughter Board (NNDB), and (4) a VMEbus Interface Mother Board (VIMB). The last three components of the VMENA system arc housed in a 12 6U slots Eurocard chassis. Communication established between the PC host computer and the VIMB is via the Bit3's AT-VME adapter board, and VIMB is communicating with the NNDB via their on-board local buses.

The PC host computer is used to download configuration and control information, weight matrices, and input/output vectors. The commercially available AT-VME adapter serves as an interface between the ISAbus of the host computer and the standard VMEbus of the VIMB. The heart of the VMENA is the NNDB and VIMB, each of which is the subject of discussion in the following two subsections.

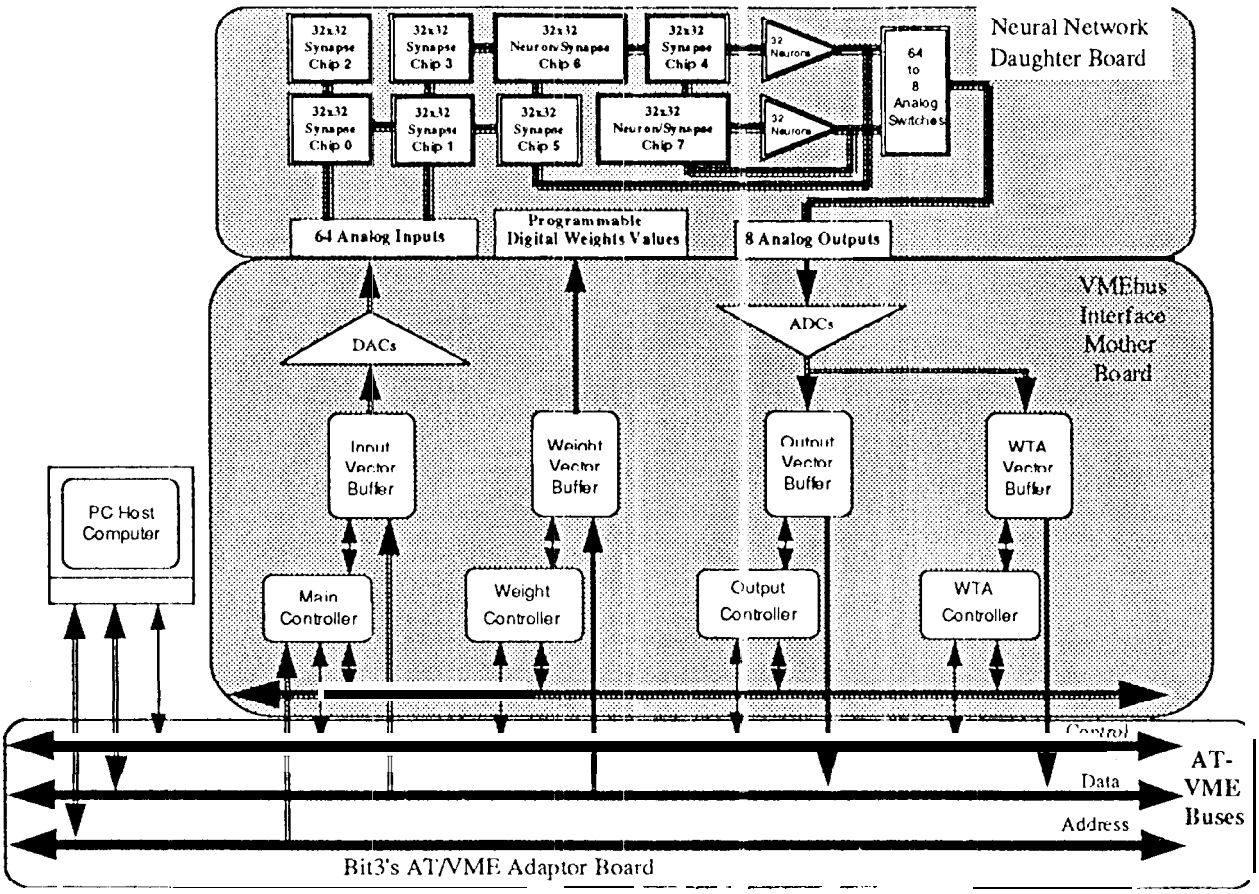


Figure 2. The functional block diagram of the VMEbus based neurocomputing architechture (VMENA).

3.1. **Neural Network Daughter Board (NNDB)**

The NNDB consists of six synapse chips and two synapse/neuron composite chips, for a total of eight neural chips. Since each of [he synapse chips is composed of a 32x32 synaptic array, and each of the synapse/neuron composite chips is composed of a 32x31 synaptic array with 32 neurons along the diagonal, a total of 8128 [=8(32x32)-64] synapses and 64 neurons are available for usc in the NNDB. To optimally utilize all synapses and neurons, the NNDB is hardwired and configured as a feedforward NN architecture with 64 input neurons, from 1 to 55 hidden neurons. and 8 output neurons. The system can be easily reconfigured under software control for a feedback NN architecture, utilizing chips 4, 5, 6, and 7 in the NNDB.

3.2. **VMEBus** Interface Mother **Board** (VIM B)

The VIMB is onc of two major components of the VMENA. It provides a user interface with a high-speed data transfer capability (via the data buffers) between the AT-VME adapter board and the NNDB. It stores both the configuration setup and the weight matrices for the NNDB as well as input vectors for real-time processing. It also collects the output neurons' activations and WTA indices to be read by the host computer.

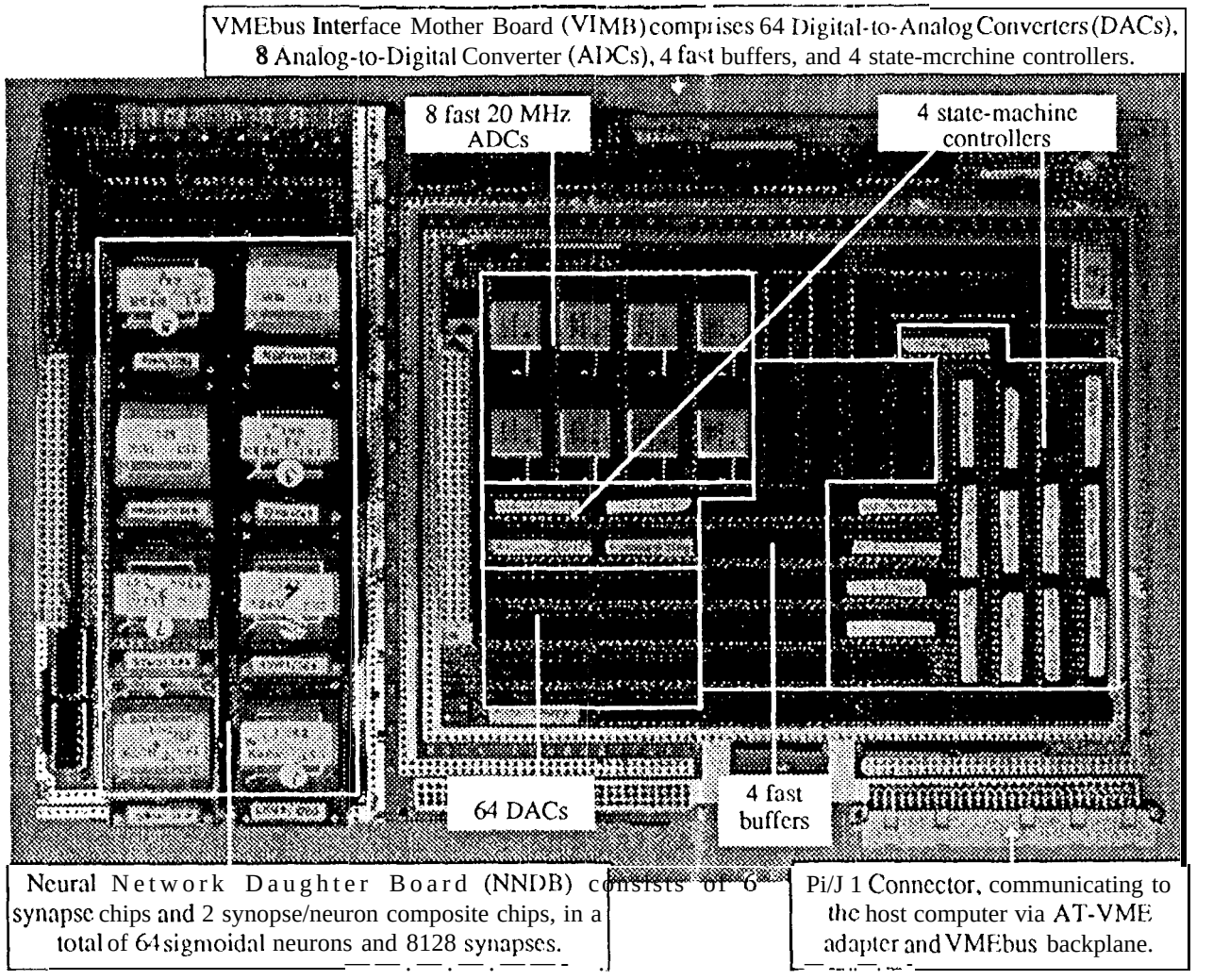


Figure 3. Photograph of VMENA consisting of the Neural Network Daughter Board (NNDB) on the left and the VMEbus Inter face Mother Board (VIMB) on the right.

in order to allow each of the eight neural chips to be trained and to perform a specific function after training, the VIMB must function not only as a digital memory but atso as a digital controller between the NNDB and the host computer. As can be seen from Figures 2 and 3, the VIMB itself consists of four functional units namely,

1. Input Vector Buffer (IVB) and the Main Controller,
2. Weight Vector Buffer (WtsVB) and the Weight Controller,
3. Output Vector Buffer (OVB) and the Output Controller, and
4. WTA Vector Buffer (WtaVB) and the WTA Controller.

Each of these buffers is associated with a state-rnachilw controller whose primary function is to both write/read to/from its buffer and to participate in arbitration with other controllers via the Local Control Bus and the AT-VME Bus. Because VME board space is at a premium, trade-offs between minimizing the number of device counts and maximizing the processing speed receive special attention in the design and implementation of the buffers and their associated controllers. While each of these buffers is implemented using a combination of high density and high-speed (15ns access time) AMD Am7205A-15 First-In-First-Out (FIFO) 8Kx9-bit memory

devices, each of these state-nurchine controllers is implemented using high-speed (15ns access time) Lattice 22V 1OL.P- 15 programmable logic devices (PLD).

Using the combination of five address bits (A 18-22), six address modifier bits (AM0-5), and control signals (AS, WRITE, L> SO, 1X31) of the VMEbus, each of the functional units of the VIMB is decoded by the Main Controller and is memory-nvrppcd into the address space ranging from 0xA00000 to 0xFFFF00 as tabulated in I'able 1.

Table 1. VMENA Memory Map.

Buffers	Memory Space	Functions
WtaVB	0xA00000-0xA7FFFF	PC host reads the winner from the WTA Vector Buffer
OVB	0xA80000-0xBFFFFF	PC host reads 4 words (8 bytes) from the Oututt Vector Buffer
WtsVB	0xC00000-0xD7FFFF	PC host writes synaptic weight values to Weight Vector Buffer
IVB	0xD800000-0xFFFF00	PC host writes input values to Input Vector Buffer

3.2.1. Input Vector Buffer (IVB) and the Main Controller

The Main Controller has two primary functions. When it functions as a global arbitrator/decoder, it coordinates all control signals among other controllers, the AT-VME adapter board, and the NNDB; it arbitrates with other controllers and decodes the address bits (A1 8-22) to determine which of four functional units (buffer and controller) of the VI MB, including itself, has the right to control the VMEbus. When the Main Controller has the right to control the VMEbus, it functions as a local controller for the IVB. It controls the data flow from the host computer to the IVB and from the IVB to the DACS and carries out its normal write/read/transfer operations.

Although inputs are 8-bit wide, they arc reconfigured into 16-bit wide word format so as to fully exploit the total 16-bit data bus bandwidth of the VMEbus. Thus. the Input Vectors Buffer (IVB) is implemented using four FIFOs, which arc connected together to form a 16K by 16-bit word buffer. Because the input vectors stored in the IVB arc in a digital format, they must be converted into analog signals before sending thcm to the NNDB. Since the NNDB is hardwired to have 64 available input neurons, 64 digital to analog converters (DACs) arc necessary. in order to minimize the number of components to save board real-estate, eight small, compact, medium speed DAC chips are selected. Each of these DAC chips consists of eight individual DAC channels each with 8-bit resolution. These eight DAC chips are implemented as four individual banks in order to provide optimal data transfer, using the full 16-bit data bus bandwidth. To maximize the processing speed, a pipeline scheme is adopted. First, the Main Controller reads from the IVB as word-wide inputs, and then writes them to the DACS' buffers; it executes its read/write operations sequentially and continuously until all 64 DACS' buffers arc filled. After waiting for 5µs for the outputs of the DACs to settle, the NNDB proceeds with its processing tasks. As will be discussed later in Section 4, the long settling time of the DACS is onc of the limiting factors and bottlenecks which slows downs the overall system operating speed.

3.2.2. Weight Vector Buffer (WtsVB) and the Weight Controller

The WtsVB unit is used as a temporary storage of the NN configuration/synaptic weights matrices whose elements arc always available to be down loaded to the NNDB. To properly write a weight value into a particular synapse's location, a sequence of operations needs to be performed as illustrated in Figure 4. First, the 5-bit row address along with its control-bits field arc presented to the NNDB. A Chip Enable (CE) control signal is subsequently asserted. Secondly, the 5-bit column address along with its control-bit field are presented to the NNDB; then, the CE is again asserted. Finally, the 7-bit weight value (from -63 to +63) along with its control-bit field arc presented to the NNDB; and the CE is again asserted. Each of these write cycle takes about 225ns for a total of 675ns per synapse or equivalently onc can address about 1.S million synapses per second. For every synaptic weight written to the NNDB, three write cycles arc required; these redundant executions slow down the downloading process and seem to be somewhat cumbersome. However, this computation overhead will be vastly improved in the next generation of NN chip set, where row, column, and weight arc presented to NNDB

simultaneously and only one activated CE strobe is required.[13] This improves the speed in writing synaptic weights to the NN chips by a factor of three. Because of the sequential execution of the row address. the column

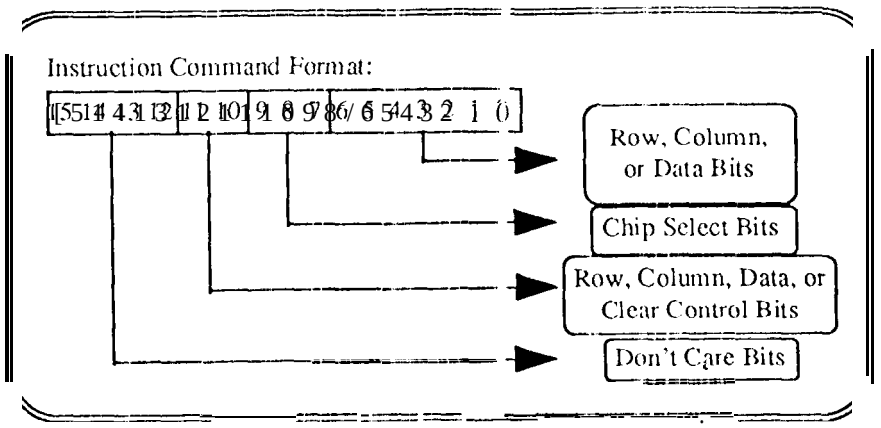


Figure 4. Instruction Command Formal for the Weight Vector Buffer (WtsVB).

address, and the weight data, only onc 8K with 8-bit wide FIFO is needed to store the synaptic weights for the neural network. The flow of data from the host computer to the WtsVB buffer and from WtsVB to NNDB is governed by the Wrtc-machine Weight Controller.

3.2.3 Ouput Vector Buffer (OVB) and the Output Contu oller

The Output Vector Buffer (OVB) is implemented with 2 FIFOs to form an 8K by 16-bit word buffer, and it is used as a temporary neuron outputs storage buffer. Although the NNDB has 64 available neuron outputs, due to NNDB board space limitation, only 8 neuron outputs are read at any given time in the current version. Each of these 8 neuron outputs is connected directly to an analog to digital converter (DATEL ADC-208), operating at a 20 MHz sampling rate. The Output Controller not only does its normal data read/wrilc/Lransfcr operation from the ADCs to OVB and from OVB to the host computer but also handshakes with the Main Controller to indicate the beginning and the end of the neural processing, where the ADC outputs become available upon the completion of neural processing. When the Output Controller recognizes the completion or "done" signal after some preset time-out, it reads the neuron outputs from the ADCS, writes thcm to the OVB, and repeats for 4 times to store 8 neurons' outputs into the OVB. Upon completion of its storage of the output values, the Output Controller sends a signal to the Main Controller to begin another neural processing or to respond to other controllers' requests. Although an 8K FIFO with 16-bit word OVB seems to be small for storage of massive data, the stored outputs arc frequently sent to the host computer during the neural processing of the new input vector. I'bus, the OVB always has some available memory to accommodate the incoming neuron outputs.

3.2.4. Winner-T'olm-All (WTA) Vector Buffer (WtaVB) and the Controller.

Thc principal function of this unit is to find [he largest neuron output among a set of output neurons. in other words, the unit performs the following function: V_{max}=maximum{V_0, Vi, ... V_{No}}, where N_o is the number of output neurons and Vi is the voltage value of the i^{th} output neuron. While all 'less-Lhfin-nlaxilnum' neurons arc considered OFF, the maximum-output neuron is defined to be ON. It is [he ON neuron that is translated into a certain feature or class. For example, if neuron #1 is ON. it corresponds to a red colored road: whereas if neuron #2 is ON, it corresponds to a blue colored river, etc. This process of finding the maximum output neurons among a set of output neurons is termed as a Winner-Take-All (WTA) operation.

As the WTA circuit is receiving the ciglit-neuron outputs, an 8-bit magnitude comparator compares them to determine the maximum output while a 3-bit decoder circuit keeps track of the position of the maximum output.

When the maximum output is found, the 3-bit decoded index is stored into the 8K FIFO Vector Buffer with 9-bit wide - even though only the first 3 bits are meaningful and significant. The decoded index is made available to the host computer whenever it requests. The WTA state-nlachinc controller coordinates all activities during the operations of the comparator, the decoder, and the 8Kx9-bit FIFO.

4. VMENA PERFORMANCE AN]) EVALUATION

Like most system designs, the emphasis in designing the VMENA *has* been placed on optimizing all available resources so as to satisfy system requirements including both board level space and system level computational throughput. As a result, some of the novel features. their impact on the system performance, and other important, open technical issues evolved from this design arc discussed in this section.

The first important feature of the VMENA is that the NN chips of the NNDB have on-chip SRAMs, into which digital synaptic weight values arc written and stored: thus, there is no need to convert the weight values into analog signals externally. A second feature of the VMENA that significantly enhances the overall system performance is that most data transfers in the VIMB are done locally. For instance, a *10CAI* 16-bit data bus is implemented to transfer the input data from the IVB to the DACS. Thus, the local bus frees the VME data bus to be used for other activities such as writing new input vectors into (tic IVB or reading, the NN output vectors. The third salient feature of the VMENA is that it allows the host computer to retrieve the neuron outputs during the training phase as well as the winner indices during the validation phase while the VIMB and NNDB arc otherwise occupied processing ncw input vectors. This "overlap" of processing time in reading the outputs or the winner indices substantially improves the overall performance of the system.

in spite of the above notable features, the first technical issue centered on the weight-loading mechanism of the NNDB. To store a synaptic weight value into a synapse, a sequence of three (row, column, and weight) consecutive execut ions is required. These executions take a major fraction of time, particularly during the hardware-in-lhe-loop learning or the training phase, in which a very large number of synaptic updates are repetitively performed. To compensate for this laggard process, a Weight Vector Buffer (WtsVB) is implemented to buffer the transfer of synaptic weights from the host computer to the NNDB. Although the processing speed is improved somewhat by this buffering scheme, the best solution is to load a weight value into a synapse directly with a single execution instead of three. This would allow the weight and its address to be loaded simultaneously in onc execution, improving the processing speed at least by a factor of three.

The second technical issue is evolved around a judicious selection of a set of 64 input DAC neurons in such a way that the trade-offs between the number of DACs, board real-estate. and computational throughput arc optimized. While faster and fewer DAC channels arc contained in a physically larger DIP package, slower and many DAC channels are compacted in a physically smaller chip. If faster DACs are selected, the whole VME 6U board would consist mainly of 64 DAC chips. For optimal selection, eight medium speed but small and compact DAC chips, each of which contains 8 DAC channels with 8-bit resolution each, arc chosen for implementation. Using these DACS, the synaptic loading throughput from the VIMB into the NNDB is somewhat moderate. The synaptic loading Throughput could have been vastly improved by using faster DACs. For instance, if DACS with 0.1 μs settling time arc used, the synaptic loading throughput would be about 7 times faster than DACS with 5 μs settling lime.

The final technical issue in the design of the VMENA is involved in the process of maximizing the overall system computational throughput, which depends largely on the speeds of both the AT-VME adapter and the host computer. Indeed, both the AT-VME adapter and the host computer turn out to be serious bottlenecks. To illustrate, let us consider our demonstrative map separates data application of a 61 K pixel map segment, which will be discussed in the next section. The VMENA subsystem, which consists of *only* the VIMB and the NNDB and *without* the AT-VME adapter and the host computer. takes about 0.83 second to classify all 61K pixels (for an effective computational throughput of 73K pixels/s). On the other hand, the overall system takes about 30 seconds to classify all the pixels of the same map segment. I'bus, about 2.3% of the computational time is utilized by the subsystem for its classification tasks, and about 97-98% of the time is shared between the host computer for display

of graphics anti the AT-VME adapter for its data transfers between the host computer and the VI MB. Although the communication overhead overwhelms the overall processing. time, the computational throughput of the VMENA subsystem is satisfactory compared it to an ISAbus based of a similar neural network architecture, however. For the same map segment of 61 K pixels, it would take the ISAbus based neural network about 1.83 seconds to process. } lence, the computational throughput of the VMENA subsystem is improved over twofold relative to the ISAbus based neural network architecture.

There arc several approaches to further deal with the above bottlenecks and 10 improve the overall system computational throughput. To rectify the data congestion on the VME data bus. one way is the use of fast buffers. Utilizing the fast (1511s) access time, low cost, and relatively small size 8Kx9 FIFO memory as onc of the basic building devices, all buffers arc implemented as temporary storage buffers of data coming in and going out of the VIMB, so that the 16-bit bandwidth of the VME data bus can fully be utilized. For example, the input vectors arc written to the Input Vector Buffer (IVB) a word instead of a byte at a time. and it is always ready to receive the incoming data. Thus, the VME data bus can be freed immediately to receive ncw data or to participate in other bus related activities. Another way for improving the speed of data transfer between the host computer and the VIMB is the usc of the VME64, in which data could be transferred 64 bits or 4 words simultaneously instead of 16 bits or a word at a time. This usage of VME64 would improve the overal I processing speed fourfold. Moreover, all data transfers between the host computer and the VIMB should be done in the block or burst mode. i.e. a block of data is transferred discretely so that the idle time between write cycles could be substantially minimized, implying another significant speed improvement.

5. MAP SEPARATES DATA CLASSII ICATION PROBLEM

Digitized map-data have found numerous applications for both commercial and industrial, as well as military sectors. Thus, there is a concerted effort to globally digitize printed map-data into high-resolution (24 bits/pixel) map-data files and store the resulting data on CD-ROM storage devices. Furthermore, applications exist where it is no longer adequate to display full color renditions of the digitized map-data - but to prc-process the map-data (either on-line or off-line), trig relevant features. and display the resulting map-data in an uncluttered fashion.

The map separates problem is therefore a feature classification problem where classifications of map pixels arc based on a surrounding window of pixels. This window approach increases the accuracy by classifying pixels within their local context. In this application, each of our digitized map-data sets consists of 61K pixels (arranged as a 305x200 pixel image) (Figure 5(a)), from which seven distinct features corresponding to roads, rivers, forests, contour lines, names/symbols, man-made structure, and open areas, arc to be tagged (classified). For optimal classification accuracy, we selected a 3x3 pixel window for each color. As each pixel is represented in full color (red, green, and blue with onc byte each), a 27 byte digital input vector is generated for each NN input. Each such input vector is to be associated with a 7 byte output (target) vector representing the 7 distinct feature classes discussed. Among these outputs (or feature classes), only one classified output, corresponding to the central pixel of the input window, is ON and the rest are. OFF. Thus, each input vector is paired with a target vector to form an input/target pair, and a training set is manually generated from the validation set by an expert analyst - resulting in 3766 such input/target pairs. Since the map separates data problem is defined to have 7 feature classes, it utilizes only 7 out of 8 VMENA's neuron outputs during the classification phase.

in solving the map-data classification problem in hardware, several unique characteristics of the CBP arc observed. The first characteristic is that the algorithm heavily depends upon the values of the learning rate, particularly during bardwarc-in-tbe-loop-learning. While onc set of learning rates produces stable results, another set of learning rates may drive the algorithm into oscillatory mode or the system into latchup mode. The second characteristic is that as the network grows (as more hidden neurons arc added), the accuracy on the training set increases monotonically; however, it is not automatically implied that the accuracy on the test set will increase monotonically. As a ncw hidden neuron is added, the accuracy on the test data set may get worse over tbc preceding hidden neuron's result, even though the accuracy on the training set generally continues to improve. Consequently, there is a point of diminishing return in the number of added hidden neurons for a given NN architecture, at which the accuracy on the test data set begins to decline. A similar phenomenon is observed in the

NN simulations.' The underlying reason for [his unique behavior is a consequence of the network "over learning" or "over specializing" on the [raining data. Therefore, the criterion for stopping the algorithm is when the percentage of accuracy on the test data set at the current hidden neuron is much lower than that at the 'point of diminishing return' hidden neuron, where the percentage of accuracy is maximal. This maximum percentage of accuracy is the best result for the given learning trail. in this paper, the accuracy or the percentage of accuracy expresses the ability of the neural network to correctly classify an unknown or test data set, and it is generally employed in measuring the algorithmic performance. It is given as follows:

$$\%A_{TestSet} = \frac{(N_{ValidSet}A_{ValidSet} - N_{TrainSet}A_{TrainSet})}{(N_{ValidSet} - N_{TrainSet})} \qquad (1)$$

Where

$N_{ValidSet}$ is the total number of input/target vectors in the validation data set.
$A_{ValidSet}$ is the percentage of accuracy of the validation data set.
$N_{TrainSet}$ is the number of input/target vectors used to train the neural net.
$A_{TrainSet}$ is the percentage of accuracy of the training set. which is the subset of the validation set.

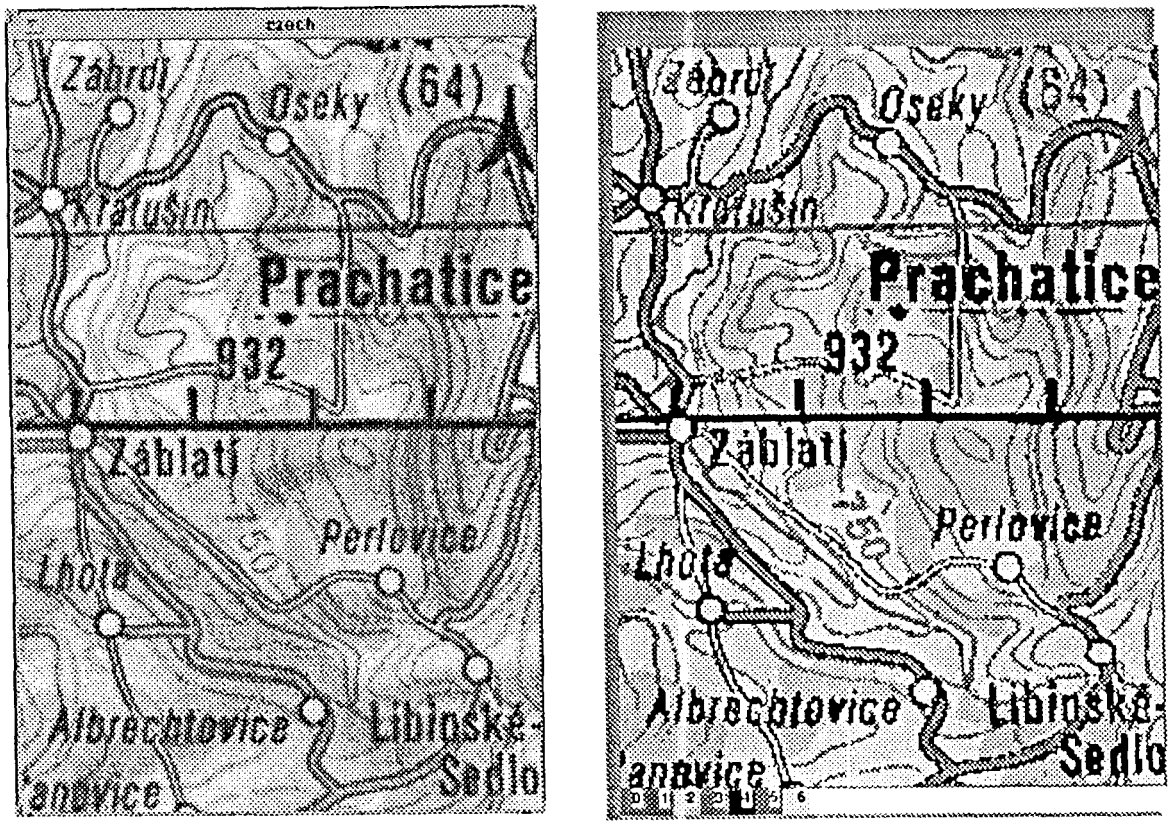
It is interesting to note that the test data set also is the subset of the validation data set. The final distinguishing characteristic is that this percentage of accuracy is a function of the number of the input/target vectors in the training set. So, what size of the training set is necessary to accurately produce a sufficient mapping from the input layer to the output layer for this classification problem in hardware? To properly address this question, series of hardware-in-the-loop learning trials for various size training sets are conducted to determine the spread of the percentage of the accuracy. For each training set size, ten learning trials are performed. The statistical results of each such experiment are tabulated in Table 2.

Table 2. Percentage (%) of accuracy for various sires of training sets for the map classification problem.

Experimental Statistics	% accuracy of different sizes of training set			
	50 pixels	100 pixels	500 pixels	2300 pixels
mean	63.41	65.16	66.61	75.21
standard deviation	2.26	3.29	4.24	7.76
minimum confidence interval	58.89	58.58	58.13	59.69
maximum confidence interval	67.93	71.74	75.09	90.73

The confidence interval of a particular learning trial is defined to be two standard deviations away from the mean; the minimum confidence interval is Iwo standard deviations less than the mean, and the maximum confidence interval is two standard deviations more than the mean This means that a 95% confidence (certainty) of the accuracy of a particular learning trial will fall into this range.[14] From Table 2, we observed that as the size of the training set increases, both the percentage accuracy mean and its standard deviation increase, and the maximum confidence interval limit increases as well. Because the standard deviation of the learning accuracy increases, the accuracy distribution of a given learning trial spreads wider as the size of the (raining set increases. For example, for an arbitrary learning trial consisting of 2300 inllut/target pixel training set, the accuracy can be as low as 6070 and as high as 9170. The best result of the NN hardware (89.3%), which is within the 2300 pixels

training set's confidence interval. shows that the hardware performs competitively comparable to NN simulation (91.2%).3 The original map and the hardware output map after a completion of a hardware-in-(hc-lcrop training are depicted in Figure 5(a) and (b), respectively.



(ir) (b)

Figure 5. The original map image (a) and the hardware neural network output (b).

The discrepancy between the software and the hardware NN results is attributed to several sources. Experimentally, one observable source, as mentioned above. is the impact of the hardware learning rate on the accuracy. If the hardware learning rate deviates by a few percent from its best value, then the accuracy would be substantially affected. Another noticeable source is the computation of the derivative in hardware, which is done by perturbing the biases at the neuron inputs and taking the difference in their neuron outputs. If the perturbed bias value differs from its best value a little, the overall accuracy would be greatly affected. This effect could possibly be duc to the limited synaptic resolution. Moreover, not only the training set size but also its contents affect the overall accuracy. For example, Iwo training sets A and B of the same size, i.e. 2300 pixels, but different contents arc used to train the network, If set A has more redundant copies of the input/target pairs than set B, then, the result of set A will be less accurate than that of set B. Obviously, this effect is duc to the fact that the training set A is the subset of the training set B. Finally, although the learning utilizes all the available bits of hardware precision, the weight updates occasionally err in either magnitude or sign or both due to noises from several layers of wire-wrap connections of and between the VIMB and the NNDB; such a stochastic error can limit the learning capability and alter the overall accur acy.

6. CONCLUSIONS

Fully parallel, analog neural network hardware systems could indeed meet the processing throughput demands required by various real-time application in image processing. The results for the map separates data classification problem demonstrate this capability. For example, *without* the consideration of the speeds of the AT-VME adapter and the host computer, the VMENA subsystem takes 0.83 seconds to classify all 61K pixel map segment (for an effective computational throughput of 73K pixels/s) whereas the ISAbus based neural network takes about 1.83 seconds to process the same map segment (for an effective computational throughput of 33K pixels/s). However, the host computer and the AT-VME adapter still ingest a great deal of processing time, Though the speed of the host computer can be easily rectified by replacing the host with a faster computer, the primary bottleneck which limits the data transfer throughput between the host computer and the NNDB still remains at both the AT-VME adapter interface and at the input DACs. They arc largely compensated for by the incursion of three novel features of the VI MB: (1) the local data transfer technique,(2) the data buffering scheme, and (3) reading the outputs (or winners' indices) during the processing of the input vectors. Contributing to the overall processing speedup of the VMENA individually, each of these techniques is employed in order to utilize the VME data bus to its fullest extent by efficient means of augmenting its data transfer throughput, buffering the data transfers, and time-multiplexing the input/output write/read operations.

Increased computational throughput could be attained by eliminating the data congestion problem via an adoption of a VME64 bus architecture and of a burst mode data transfer between the host computer and the VIMB. The processing speed of the VMENA would be improved by at least a factor of four. Furthermore, another speed-up could be achieved with a simple chip level redesign. The existing NN chips require three consecutive executions to load a single synaptic weight value. A redesigned chip would need only one, hence a three-fold improvement.

7. 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 acknowledge Dr. S. Narathong for his technical expertise and effort in the design, debug, and test of the VME interface, The authors also wish to thank Mr. H, Langenbacher for useful technical discussions and assistance.

8. REFERENCES

1. T. X. Brown, M.D. Tran, T.A. Duong, T. Daud, and A.P. Thakoor, "Cascaded VLSI neural network chips: hardware learning for pattern recognition and class ification," *Simulation*, 58, 340-347, Special Issue on 'Neural Networks': Model development for applications. 1992.
2. T. J. Graettinger, N. V. Bhat, and J. S. Buck, "Adaptive control with NeuCOP, the Neural Control and Optimization Package," World Congress on Computational Intelligence, IEEE International Symposium on Evolutionary Computation, pp.2389-2393, June 26-July 2, 1994.
3. T. A. Duong, S. P. Eberhardt, T. Daud, and A, Thakoor, "Learning in neural networks: VLSI implementation strategies," In: *Fuzzy Logic and Neural Networks Handbook*, Ed: *('. H.* Chen, McGraw-Hill (In press).
4. S. P. Eberhardt, A. Moopenn and A.P. Thakoor, "Considerations for hardware. implementations of neural networks," In Proc. of the 22nd Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA 1988.
5. R. Tawel, "Learning in analog neural network hardware," *Computers Electronic engineering, vol. 19, No. 6,* pp.453-467, 1993.
6. A. P. Thakoor, "Hardware implementations and applications of neural network architectures," ARPA/DoD/NASA Final Report, JPL,D-i2518 (internal document), Pasadena, California, May 1993.
7. T. A. Duong, S.P. Eberthardt, M. D. Tran, T. Daud, and A.P. Thakoor, "Learning and optimization with cascaded VLS1 neural network building-block chips," Proc. IEEE/INNS Int'l Joint Conf. on Neural Networks, vol. I, pp.184-189, June 7-11, Baltimore, MD, 1992..

8. T.A. D uong, T. Brown, M. Tran, H. Langenbacher, and T. Daud, "Analog VLSI neural network building block chips for hardware-in-the-loop learning," Proc. IEEE/INNS Int'l Join Conf. on Neural Networks. vol. 11[, Bejing, China, Nov. 3-6, 1992.
9. T. Daud, T. Duong, M. Tran, H. Langenbacher, and A. Thakoor, "High resolution synaptic weights and t~:lrdw:llc-i[l-[tic-loop learning," Proceedings of IS&T/SPIE Conference (Photonics '95), vol. 2424, San Jose, CA, Feb. 6-10, 1995 (To be published).
10. S.E. Fahlman and C. Lebiere, "The cascade correlation learning architechture," in *Advances in Neural Information Processing Systems II*, ed. D. Touretzky, Morgan Kaufman, San Mateo, CA, 1990, pp.524-532.
11. D. E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," ed. D.E. Rumelhart, J.L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 1: Foundations, MIT Press, Cambridge, MA. 1986, pp. 318-362.
12. T. A. Duong, Allen R. Stubberud, and Taher Daud, "Cascade error projection lemming theory," To be sumitted to *Neural Computation, 1995*.
13. T.A. Duong, S. Kemeny, M. Tran, T. Daud, A. Thakoor, D. Ludwig, C. Saunders, and J. Carson, "Low power analog neurosynapse chips for a 3-D 'sugercube' neuroprocessor," World Congress on Computational Intelligence, IEEE International Symposium on Evolutionary Computation, vol. 111, pp. 1907-1911, June 26-July 2, 1994.
14. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C: The Art of scientific Computing*, Chapter 13 (pp. 473) and Chapter 14 (pp. S48-55 /), 1988.