NESTOR, INC.

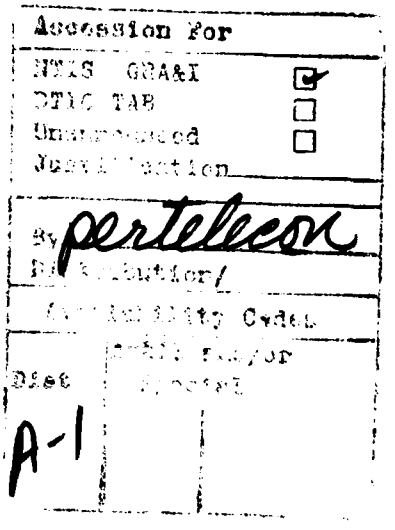
DEVELOPMENT OF GENUINE NEURAL NETWORK PROTOTYPE CHIP
N00014-90-C-0010

QUARTERLY R&D STATUS REPORT
TO
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
AND
OFFICE OF NAVAL RESEARCH

23 JULY TO 22 OCTOBER 1990

28 JANUARY 1991

St #A, Auth: ONR/Code 353
(Dr. Clifford Law - 696-4216)
telecon, 30 Sep 94 CB



R & D STATUS REPORT

ARPA ORDER NO.: 7017 PROGRAM CODE NO.: 8D10

CONTRACTOR: Nestor, Inc.

CONTRACT NO.: N00014-90-C-0010 CONTRACT AMOUNT: $1,199,981.

EFFECTIVE DATE OF CONTRACT: 22 Jan 1990

EXPIRATION DATE OF CONTRACT: 21 Apr 1992

PRINCIPAL INVESTIGATOR: Michael Glier

TELEPHONE NO.: 401-331-9640

SHORT TITLE OF WORK: Development of Genuine Neural Network Chip Prototype

REPORTING PERIOD: 23 July 1990 to 22 October 1990

I. PROGRESS DURING PERIOD

The second revision of the Target specification has been completed and is attached to this report as Appendix A. Two additional working design reviews have been conducted at the subcontractor facility. Control flow for on-chip learning has been defined which supports in-circuit initial device programming and in-circuit incremental learning. Output calculation units have been proposed which use integer arithmetic. Simulations using Sonar and Character recognition data sets have been completed, effects are described for: 5 bit to 8 bit input resolution, input resolution accuracy, requirements for internal circuit computation accuracies and output accuracy. The detailed design work on the input difference calculator, summing elements and output math units has not been completed and delays the availability of the next revision of the chip floor plan. A chip self-test capability has been defined, but the work has not been documented to the extent necessary for inclusion in this report. The chip code name has been changed to the NS1000 to eliminate confusion with other internal projects within Intel that use a standalone "N" prefix designation.

II. SCOPE OF WORK THIS PERIOD

A. Overview of Simulation Work

In response to the request for documentation of the required accuracies of various elements on the NS1000, we evaluated several possible benchmarks for analysis. It was suggested during the quarterly review of July, 1990, that a theoretical analysis could place limits on the required accuracies. However, we noted that such analyses are only practical under limited conditions, such as when one assumes that the data modelled by the NS1000 are described by simple Gaussian distributions. Such circumstances are sufficiently different from the conditions under which the device is expected to perform that it was decided that actual simulation with data sets would be preferable.

Sources of simulation data may be divided into synthetic data and live (real-world) data. For the same reasons that a theoretical analysis did not seem adequate, the generation of synthetic data seemed unlikely to properly test the capabilities of the proposed device. In particular, the generation of a large quantity of synthetic data samples is practical, but the diversity of distributions tends to be limited to simple linear combinations of gaussian distributions. It was felt that a more realistic testbed for device characteristics could be found in live data which displayed idiosyncrasies such as complex nonlinear boundaries, realistic nonlinear density variations, and proper class distributions.

Available live data sets fall into three distinct domains: signal data sets consisting of a small number of classes and a small number of data samples (< 1000), image-based data sets with a larger number of classes, and medium sized quantity of data (< 50,000), and risk assessment data consisting of two classes and very large quantities of data (100,000 - 1,000,000).

In the first category, two military databases are available. The first, a signal data set derived from the Sharem Sea Trials, consists of two classes of active returns from submerged reflectors (Target, Clutter). The total data set size is 366 samples, of which 313 are of class Target and 53 of class Clutter. The second data set is IR target data which contains approximately 5-10 classes and was suggested by the Naval Weapons Center, it has not yet been made available. We expect be able to describe this data set in the next report.

The second category of data is representative of image analysis problems. The data consists of isolated characters derived from the National Institutes of Standards and Technology (NIST) handprint database. A sample of approximately 30,000 isolated handprint numbers has been assembled for analysis. The data set consists of binary pixel images of approximately 60x60 pixels, and ten classes of patterns. This data set is approximately an order of magnitude larger than the military signal data sets.

The third category of data consists of samples of credit card transactions. This data set has been assembled for the detection of credit card fraud. Thus the classes of patterns may be organized into just two categories. A very large quantity of this data is available. The frequency of one class (fraud) however, is very small. Over 300,000 samples of this data are available. Thus this data set is approximately an order of magnitude larger than the preceding character data set.

The generation of comparative performance data to understand the effects of device accuracy limitations requires the use of an *ensemble* of models and performances so that mean performances and standard deviations may be calculated. This is accomplished by dividing the full data set into a number of divisions (say 10). Then a subset of the divisions are used for training a neural network memory, and the remainder are used for testing classification performance. The training and test sets are then rotated among the divisions, and the process is repeated. Results are recorded for each of the orderings of the training and test sets and the ensemble of test results are used to compute the average and standard deviation of performance.

Modelling of these categories of data is limited by the simulation time on a serial computer. The small data sets may be trained and tested in less than 10 minutes on a Sun 3 workstation. Thus an ensemble of runs may be completed in several hours. The medium size data sets require approximately 1 day for full training of a single model. Thus an ensemble would require several weeks to complete. The third category requires substantially longer still for model generation. These training times make the second and third categories of data prohibitive for model parameter analysis. Only simple calculations which could be performed with a static neural network memory were performed with the second category of data. For these reasons, the first category of data was selected for complete analysis of the device characteristics.

All analyses were performed using a *rank graph*. Such a graph is composed by recording the probability of the predicted class for each test pattern in the test set. The test patterns are then ordered from highest to lowest predicted probability for the class of interest and divided into divisions or *tiles*. A rank graph is generated by plotting the cumulative proportion of the class of interest as a function of the percentage of the test data which has been processed. Thus, for example, a point plotted at coordinates (15, 35) indicates that if one plots the number of patterns with class equal to the class of interest in the first 15% of the ranked data, 35% of all patterns of that class will have been identified. The performance of the classifier may be judged by noting the similarity between the ideal performance (illustrated by a line joining the plot origin and a point a the top of the plot which corresponds to the *a priori* distribution of the predicted class), and the line joining the plotted tile points. Alternatively one may note the difference between the plotted tile points and the line joining the plot origin with the upper right hand corner of the plot. This latter line corresponds to random predictions of test set classifications, and thus corresponds to the poorest performance. Good classifier performance will tend to follow the ideal line closely until the tile predicted probability is quite low, at which point the classifier line will begin to fall below the ideal line and may approach the random performance line.

When the distribution of predicted class is high, the difference between the ideal line and the random performance line is quite small. Thus it is difficult to visually judge performance since the classifier performance line will lie between the ideal line and the random line. For a two class problem, the two class probabilities are related through normalization by $P_a = 1 - P_b$. Thus a plot of the class which is represented by the lower *a priori* distribution is equivalent, and more readily visualized with a rank curve. The sonar signal data consisted of approximately 85% class Target and 15% class Clutter. For this reason, the rank curves for the class Clutter were used for device parameter analysis.

B. Proposed Computational Accuracies within the NS1000

The current design plan is to perform all probabilistic calculations using low resolution floating point numbers. All values will be stored using 10 bit mantisas with 6 bit signed exponents. The single exception to this is the sigma value which will use a 4 bit mantissa and a 4 bit signed exponent. The exponential function will be approximated using an algorithm supplied by Intel that has no more than 0.1% error within its valid range, however it is limited to a minimum non-zero output of $1.5 * 10^{-5}$.

The non-normalized probability for the k^{th} class is given by:

$$P_k = \sum_{i=1}^{n_k} e^{-\sigma d_i} c_i. \qquad (1)$$

where $d_i = \sum_{j=1}^{N} |\omega_{ij} - f_j|$, is the city-block distance between the pattern $\underline{f}$ and the i^{th} prototype weight vector, $\underline{\omega}_i$. Then the normalized probability is computed according to:

$$P_k = P_k/Q, \quad \text{where } Q = \sum_{k=1}^{K} P_k. \qquad (2)$$

Each term must be computed with a set dynamic range and number of significant bits. We have estimated the following values for each parameter:

Parameter	Range	Significant bits
c_i	[0, 65,000], (16 bits)	10
d_i	[0, 8192], (13 bits)	13
σ	[0, 10^{-3}], (10 bits)	4
$e^{-\sigma d_i}$	[0.0, 1.0]	10
$e^{-\sigma d_i} c_i$	[0, 65,000], (16 bits)	10
P_k	[0, 65×10^6], (26 bits)	10
Q	[0, 36×10^8], (36 bits)	10
P_k	[0, 999], (10 bits)	10

C. Effects of Reduced Input Data Resolution and Accuracy

The Sonar data set was used to produce the the rank graph in Figure 1. The internal computation resolutions of the NS1000 are those shown in (B) of this section.The original Clutter features in this data set used 8 bit resolution. For this test, the original 8 bit features were scaled into values of 5, 6 and 7 bits, with the unscaled features shown for reference. For this particular problem, the performance of the network is considered to be roughly equivalent for each resolution.

The rank graph in Figure 2 shows the effect of a +/- 1 bit random error in the input feature data. The performance of both are roughly equivalent until 60% of the test data is processed, where the random error tends to help discriminate CLUT better for 3 tiles, where the results are within both error bars for the problem.

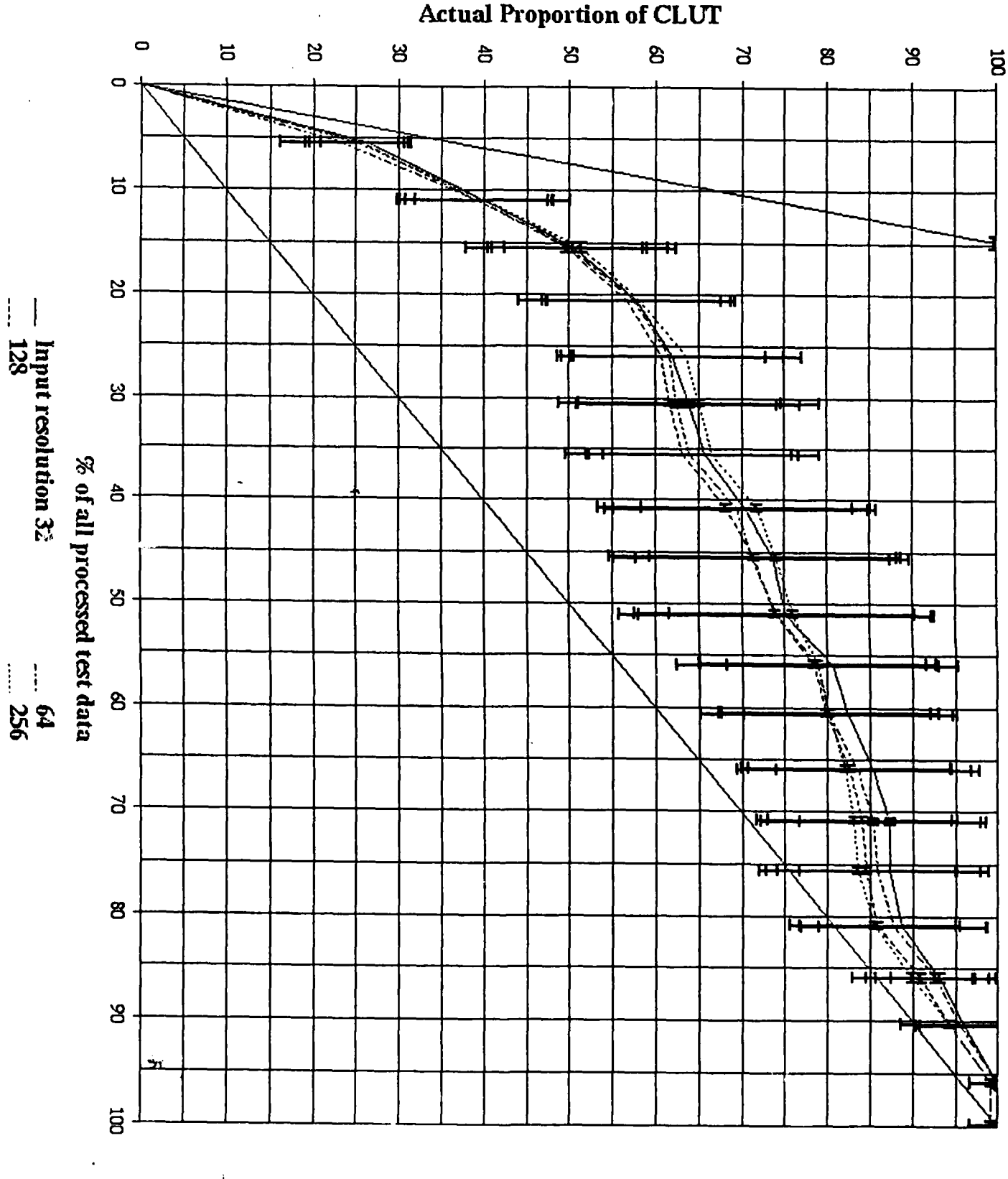

Rank graph: Sonar: Varied input resolutions, Chip npd, decay = 0.001, cutoff = 0, 20 sets 80% for training
Actual Proportion of CLUT
% of all processed test data
0
10
20
30
40
50
60
70
80
90
100
— Input resolution 32
---- 64
---- 128
..... 256

FIGURE 1

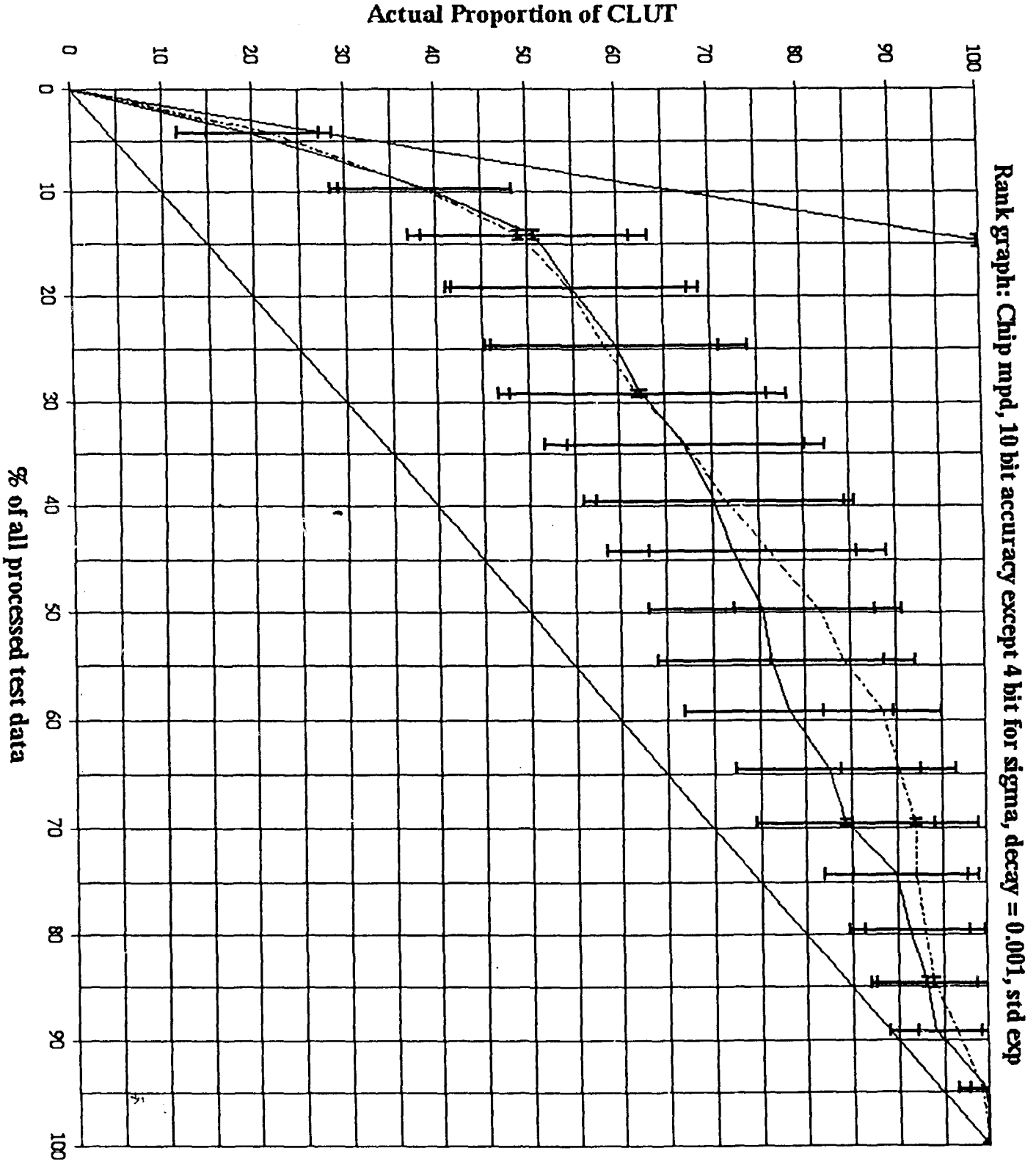

Actual Proportion of CLUT
0
10
20
30
40
50
60
70
80
90
100
Rank graph: Chip mpd, 10 bit accuracy except 4 bit for sigma, decay = 0.001, std exp
% of all processed test data
No input error
Input error +/- 1 bit

FIGURE 2

D. Prototype Distance Resolution Requirement

The recognition memories for the Sonar and Character Recognition data sets were used to construct histograms of prototype distances, and are shown in Figures 3 and 4. The data shows that the requirement for distance resolution for the Sonar and Character recognition problems, measured as a percentage of the total feature space (8192), to be approximately 13.5% and 22%, respectively. This would require a distance resolution of 11 bits minimum for both problems, without regard for the method of overflow detection for input pattern distances which are greater than this value.

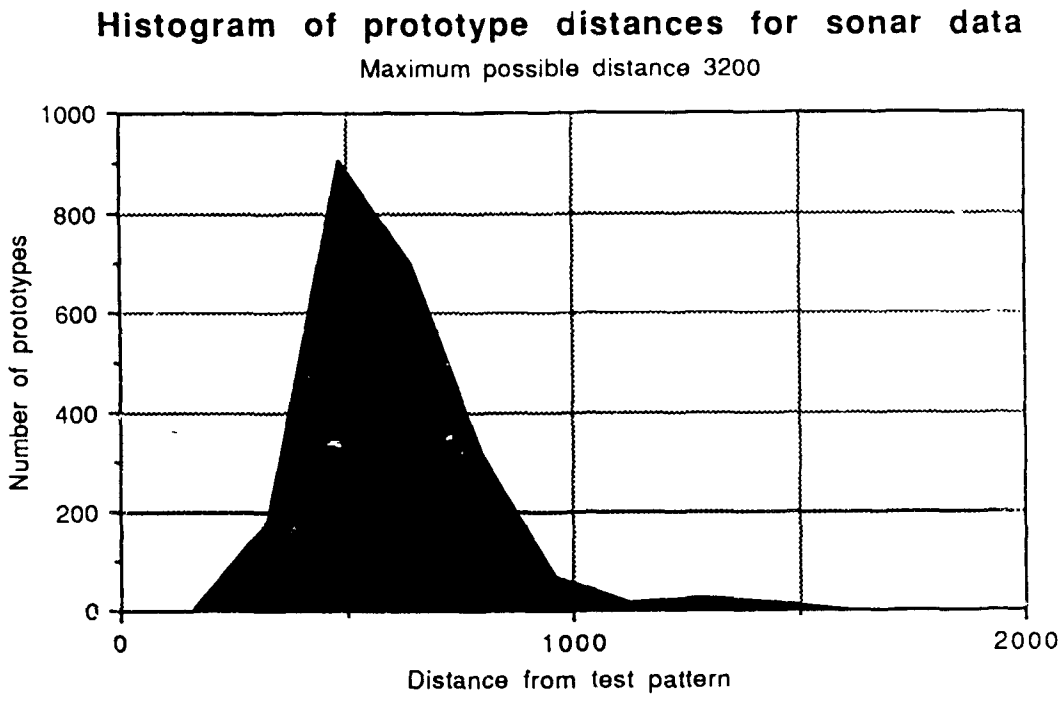

Histogram of prototype distances for sonar data
Maximum possible distance 3200
1000
800
600
400
200
0
Number of prototypes
0
1000
2000
Distance from test pattern

FIGURE 3

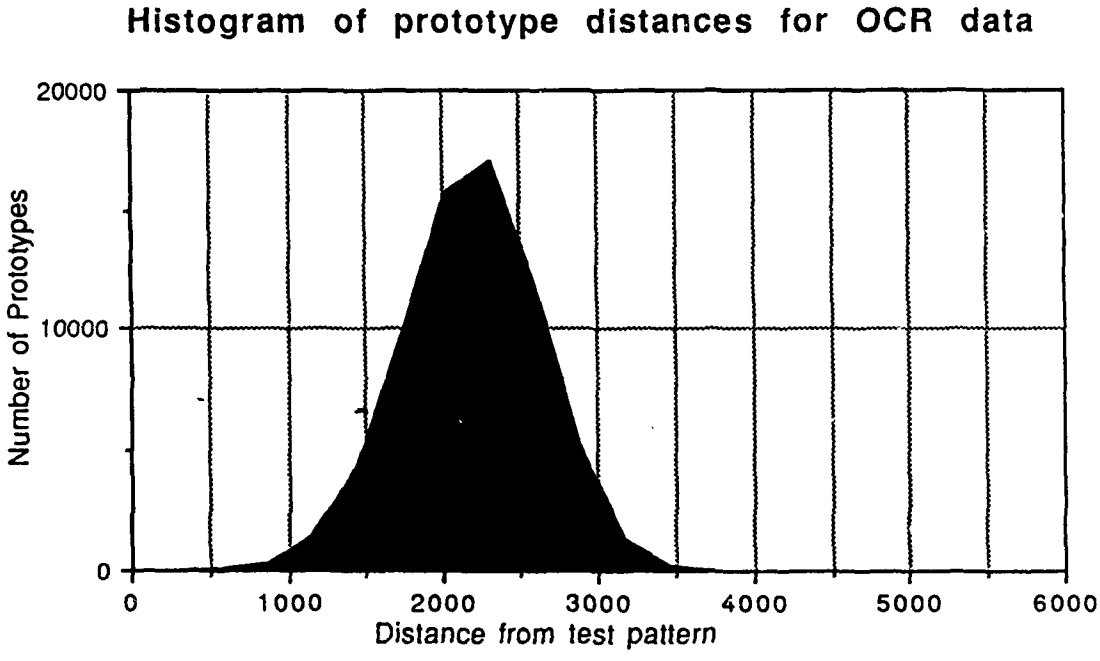

Histogram of prototype distances for OCR data
20000
10000
0
Number of Prototypes
0
1000
2000
3000
4000
5000
6000
Distance from test pattern

FIGURE 4

E. Computation Accuracy, NS1000 vs Software Simulation on Sun 3

The Sonar data set was again used to provide a comparison between computations performed using the standard Sun 3 math libraries which implements the ANSI Floating Point Standard 754-1985 and computations performed on the NS1000 using the proposed resolutions in Section II B above. The graph in Figure 5 shows no significant difference between the computation results on the Sun 3 and the simulated results on the NS1000.

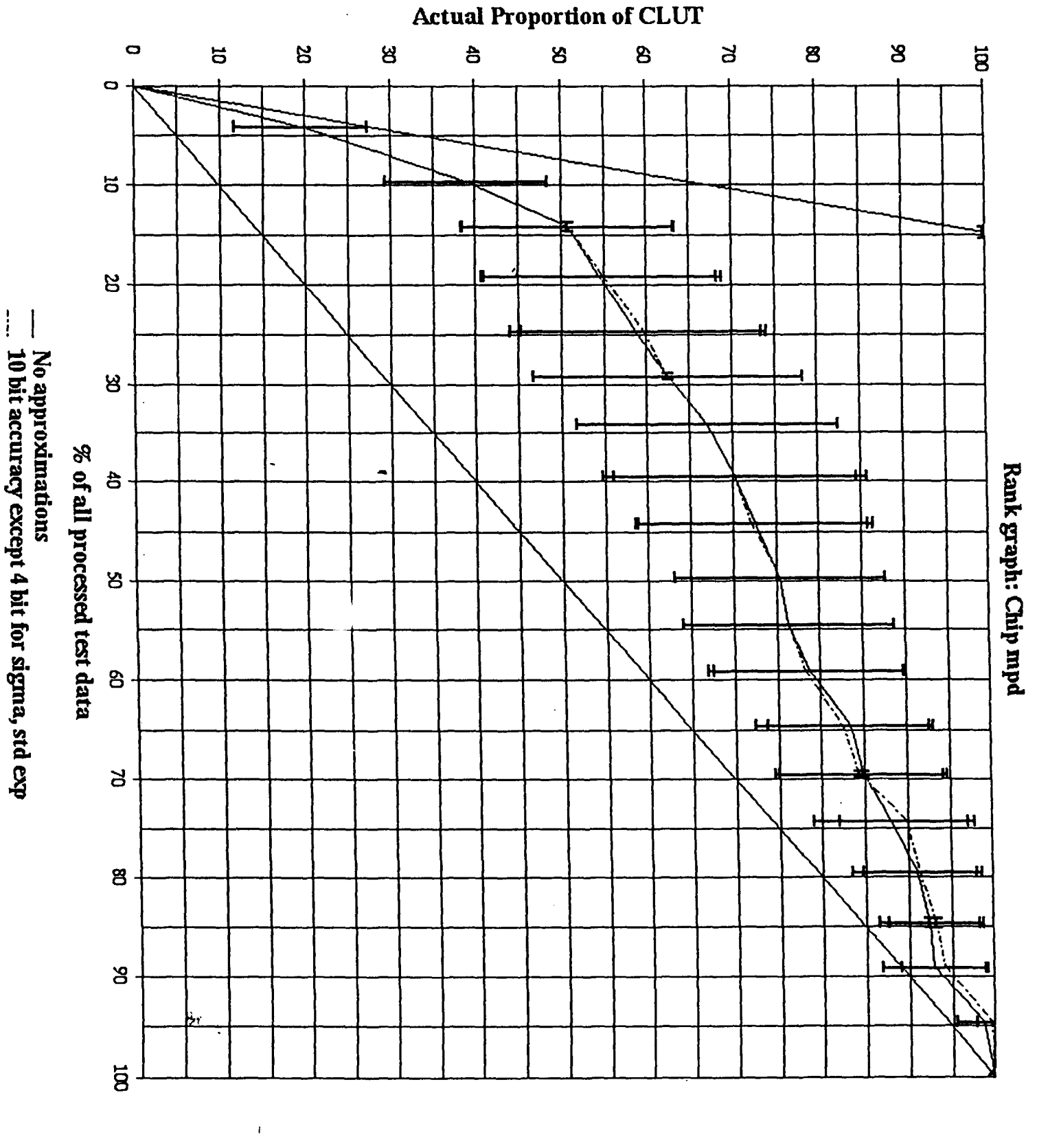

Actual Proportion of CLUT
0
10
20
30
40
50
60
70
80
90
100
% of all processed test data
Rank graph: Chip mpd
— No approximations
---- 10 bit accuracy except 4 bit for sigma, std exp

FIGURE 5

F. Output Probability Resolution

The Sonar data for "Clutter" in Figure 6 was computed using the computation methods described in II E above. The class "Clutter" output was "forced correct", that is, the system regarded any non-zero probability as a correct classification and the results were used to measure the performance of the system. Different probability thresholds were applied, from a high of .7 to a low of .03. The different threshold values show degrading accuracy with progressively lower non-zero values of probability. The standard deviation results indicate that an output probability resolution of 1 part in 10 would be adequate for this problem.

The OCR data set results summary in Figure 7 show the results of 4 different test runs using the NIST handprint data set, again for 10 classes of numeric data. The standard deviation results in run CF4 show that an output probability resolution of 1 part in 10^2 would be adequate for this problem. We strongly believe, however, that a probability resolution of 1 part in 10^3 would be required for a combined alphanumeric class differentiation (64 classes), particularly where the training data set would be very large (>200,000 patterns).

	Total Correct	Incorrect	Forced Correct	Confused	Unidentified	Unforced Throughput	Forced Throughput	Num Pats Encoded	Num Pats Not Encoded
prot_diff_thresh .7									
Mean value	11.73%	1.16%	86.76%	87.11%	0.00%	12.89%	100.00%	100.00%	0.00%
Standard deviation	2.57%	0.91%	1.37%	2.86%	0.00%	2.86%	0.00%	0.00%	0.00%
prot_diff_thresh .09									
Mean value	11.73%	1.16%	86.76%	87.11%	0.00%	12.89%	100.00%	100.00%	0.00%
Standard deviation	2.57%	0.91%	1.37%	2.86%	0.00%	2.86%	0.00%	0.00%	0.00%
prot_diff_thresh .07									
Mean value	14.34%	1.23%	86.76%	84.43%	0.00%	15.57%	100.00%	100.00%	0.00%
Standard deviation	9.21%	0.87%	1.37%	9.28%	0.00%	9.28%	0.00%	0.00%	0.00%
prot_diff_thresh .065									
Mean value	32.69%	3.28%	86.76%	64.03%	0.00%	35.97%	100.00%	100.00%	0.00%
Standard deviation	29.24%	3.43%	1.37%	32.46%	0.00%	32.46%	0.00%	0.00%	0.00%
prot_diff_thresh .06									
Mean value	57.70%	7.64%	86.76%	34.66%	0.00%	65.34%	100.00%	100.00%	0.00%
Standard deviation	34.71%	5.59%	1.37%	39.92%	0.00%	39.92%	0.00%	0.00%	0.00%
prot_diff_thresh .055									
Mean value	71.68%	10.03%	86.76%	18.29%	0.00%	81.71%	100.00%	100.00%	0.00%
Standard deviation	26.24%	5.11%	1.37%	30.94%	0.00%	30.94%	0.00%	0.00%	0.00%
prot_diff_thresh .05									
Mean value	82.97%	12.29%	86.76%	4.74%	0.00%	95.26%	100.00%	100.00%	0.00%
Standard deviation	16.42%	3.13%	1.37%	19.01%	0.00%	19.01%	0.00%	0.00%	0.00%
prot_diff_thresh .04									
Mean value	86.76%	13.24%	86.76%	0.00%	0.00%	100.00%	100.00%	100.00%	0.00%
Standard deviation	1.37%	1.37%	1.37%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
prot_diff_thresh .03									
Mean value	86.76%	13.24%	86.76%	0.00%	0.00%	100.00%	100.00%	100.00%	0.00%
Standard deviation	1.37%	1.37%	1.37%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

FIGURE 6

Results for run cf

Ten different training/testing ensembles. Each training/testing combination had patterns from the same author.

Percent Correct	Mean = 86.641	std_dev = .553643
Percent Incorrect	Mean = 5.685	std_dev = .268463
Percent Forced Correct	Mean = 90.702	std_dev = .410793
Number pats encoded	Mean = 8062.4	std_dev = 1.26491

Results for run cf2

Starting memory from run cf. Tested on eight different test sets. None of the authors were represented in the training data and each of the testing data sets contained examples from different authors. The memory was trained on 12000 patterns.

Percent Correct	Mean = 85.8575	std_dev = 1.11474
Percent Incorrect	Mean = 5.885	std_dev = .759154
Percent Forced Correct	Mean = 90.07	std_dev = .986002
Number pats encoded	Mean = 2345.88	std_dev = 159.071

Results for run cf3

Starting memory from run cf. The memory was trained on some of the data used to create the memory used in run cf2. Tested on eight different test sets. None of the authors were represented in the training data and each of the testing data sets contained examples from different authors.

Percent Correct	Mean = 85.9038	std_dev = 1.06469
Percent Incorrect	Mean = 5.73875	std_dev = .70869
Percent Forced Correct	Mean = 90.1438	std_dev = .77249
Number pats encoded	Mean = 2345.88	std_dev = 159.071

Results for run cf4

Trained on 20000 patterns tested on the same 8 data sets as cf2 and cf3. There was no overlap between the training and testing authors.

Percent Correct	Mean = 87.1937	std_dev = 1.13724
Percent Incorrect	Mean = 5.13125	std_dev = .89141
Percent Forced Correct	Mean = 91.1887	std_dev = .973806
Number pats encoded	Mean = 2313.62	std_dev = 112.605

FIGURE 7

G. Prototype Array & Distance Calculator Unit (PA/DCU)

Use of self-timed circuits for the input distance calculators has been proposed. This would decrease the cycle time for calculating an input pattern distance to 2-3 μsec over the current cycle time estimate of 25 μsec.This would reduce the complexity of clock distribution on-chip but would increase the area/complexity of the PA/DCU by a factor of 2x. Additional work using this type of circuit design will be completed next period.

III. CHANGE IN KEY PERSONNEL

None this reporting period.

IV. SUMMARY OF SUBSTANTIVE INFORMATION DERIVED FROM SPECIAL EVENTS

None this reporting period.

V. PROBLEMS ENCOUNTERED AND/OR ANTICIPATED

Significant delays have been encountered in obtaining a new target/clutter dataset, the availability date for this data is early February, 1991. No other problems were encountered this period. None are anticipated next period.

VI. REVIEW OF FUNDING AND EXPENDITURES

Subcontractor expenditures continue somewhat under budget. A detailed schedule has been established and is shown in Appendix A. This schedule will be updated and included in subsequent Quarterly reporting periods. It should be noted that the schedule reflects activity as of 17 Dec, 1990. The goal for first silicon by Q4, '91, continues on target.

ACTION REQUIRED BY THE GOVERNMENT

A meeting will be scheduled for late March, 1991, to conduct a review of the NS1000 internal design elements and the External Target Specification. It is recommended that interested parties from DARPA and other organizations attend. Final plans regarding the meeting agenda, location and time will be made available by 5 February, 1991.

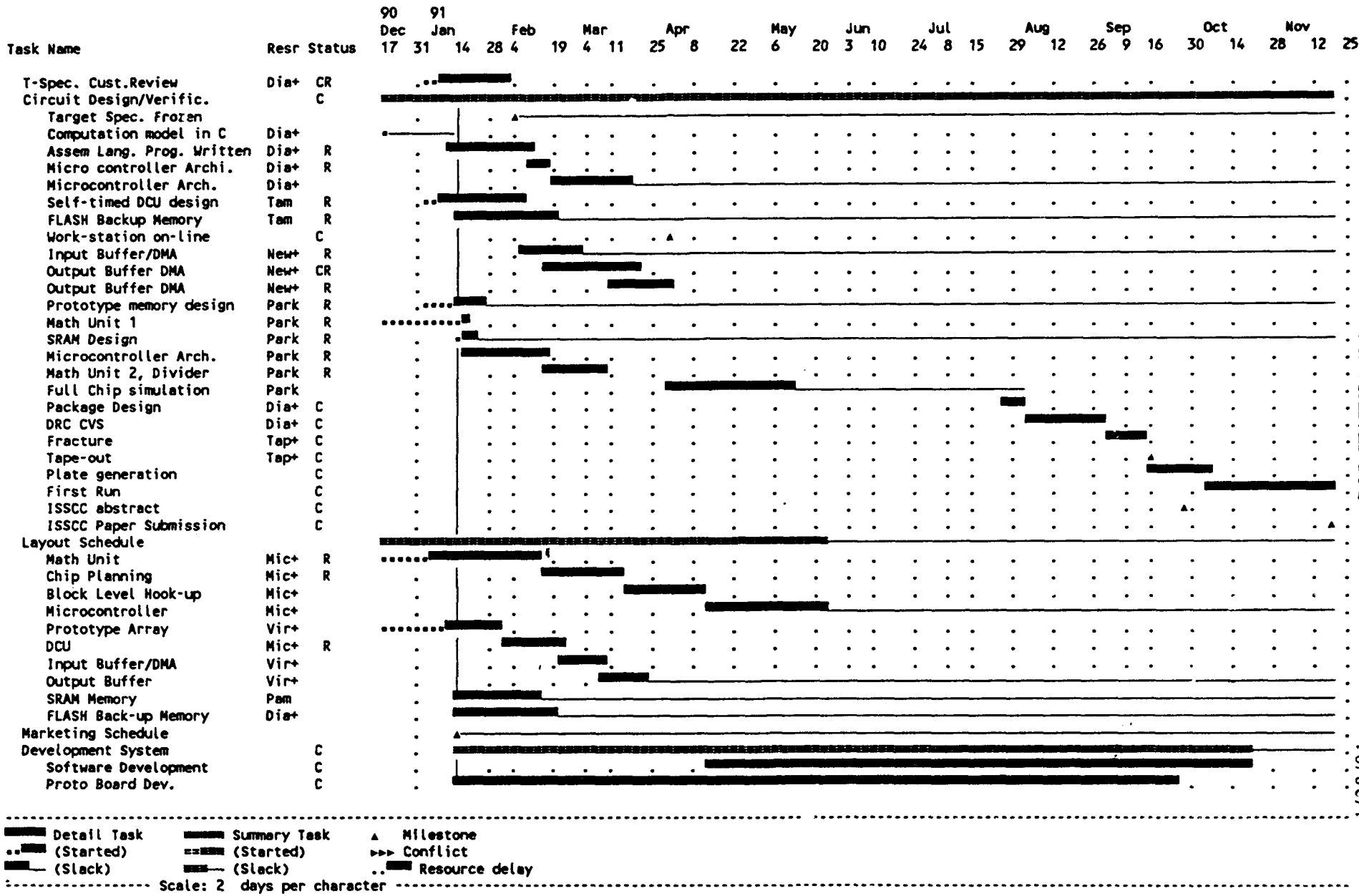
FISCAL STATUS

(1) Amount currently provided on contract: $1,199,981.
(2) Expenditures and commitments to date: $356,944.55
(3) Funds required to complete work: $0.00

PLANNED WORK NEXT PERIOD

The third revision of the external Target specification will be completed and distributed. A design and project review will be held at the subcontractor site. The proposed on-chip learning scheme will be expanded to detail the flow diagram and logic states to reflect controller parallellism and operational logic. Detailed design work will be completed on the input difference calculator, summing network and the output math units, detailed circuit design layout will also begin during this period. Circuit Block diagrams will be provided. Additional refinements to the chip I/O architecture will be completed and updated timing diagrams will be developed. The next revision of the chip floor plan will be generated.

Schedule Name : NS1K Chip Design
Responsible : M.Holler, C.Park, M.Glier
As-of Date : 15-Jan-91 Schedule File : B:\NESTOR2



R&D STATUS REPORT
N00014-90-C-0010
1/28/91

APPENDIX A
NS1000 PROJECT SCHEDULE

Nestor, Inc.

intel

Preliminary Target Specification

NS1000
Adaptive Classifier
(Y1)

Features

- RCE based classifier or probability based P-RCE Bayesian classifier. Capable of high speed pattern classification. Network automatically performs P-RCE calculations when necessary.

- Implements 2 layers of processing. First layer of artificial neurons (prototypes) calculates 1024 *city block* distances in parallel for vectors up to 256 dimensions. The second layer calculates a deterministic RCE classification or a relative classification probabilities for output classes up to 64.

- On-chip learning capable of in-the-field adaptation through the real time allocation of prototypes using the RCE learning algorithm.

- Eight 5-bit digital input channels and four 16-bit digital output channels both doubly buffered. Supports fast I/O speed up to 25 Mbytes/sec.

- 262144 5-bit electrically alterable and non-volatile prototype storage elements.

- 8K bytes of back-up Flash storage for the RCE/PRCE computation constants (λ's, C's, etc).

- Pipelined architecture supports 2.5 billion XPS at 40 Mhz clock.

- Single shot classification processing time of less than 200us and pipelined classification processing time of less than 100us.

- On-chip learning rate at 25ms per new prototype allocation and 500us per prototype threshold adjustment.
- Automatic power on self test and user accessible internal registers for debug.
- TTL compatible operation and fully digital I/O and controls.
- Low power high speed CHMOS-IV Flash memory technology.
- 157 pin PGA package or cost effective 157 pin plastic PLCC package.

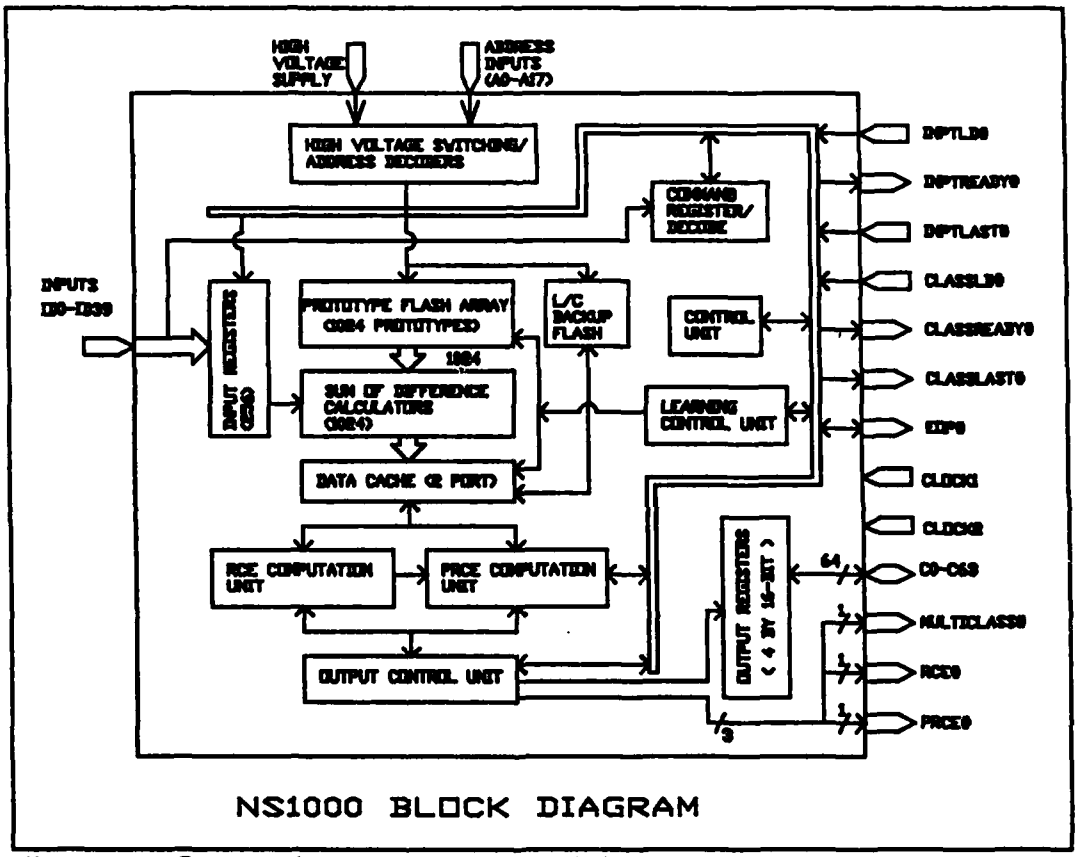


Figure 3 NS1000 Neural Network Block Diagram

1 INTRODUCTION

The *NS1000* VLSI neural network chip offers the highest performance for real-time complex classification tasks by embodying the highest neuron density, unprecedented classification rate and on-chip high speed real-time adaptation.

The NS1000 provides a complete solution to many classification tasks by integrating a three layer feed-forward neural network on a single device. It is a VLSI implementation of the Restricted Coulomb Energy (RCE) neural network architecture. The NS1000 also supports the probability based Bayesian classification when it operates in the Probabilistic RCE (PRCE) mode.

The NS1000 accepts input vectors of 256 in dimension and with granularity of 32 levels. The NS1000 produces a 64 dimensional class output vector. High speed parallel computation of the *city block* distance metric between an input vector and up to 1024 stored feature examples resulted in the unprecedented classification rate. The NS1000's high classification rate is most suitable for applications such as real-time optical character recognition, high speed targeting, and real-time signal detection processes.

The architecture of the NS1000 is shown in figure 1.

2 PRINCIPLES OF OPERATIONS

A block diagram of a typical RCE neural network is shown in figure 2. The neural network has three layers, namely, the input layer, the internal layer (prototype layer), and the output layer (classification layer). In the following, the principles of operations for classification and learning are described.

2.1 Classification Modes

The NS1000 has two basic classification modes, the RCE classification mode and the probability based PRCE classification mode. The block diagrams of the RCE mode and the PRCE mode are similar and is depicted in figure 2.

2.1.1 Internal Layer

The internal (prototype) layer of the NS1000 is common to the RCE mode and the PRCE mode. Its function is to evaluate the similarity between an input vector and the feature examples stored in the prototype layer. The NS1000 uses the *city block* distance metric as the criteria and computes the metric for all the 1024 prototype elements in parallel.

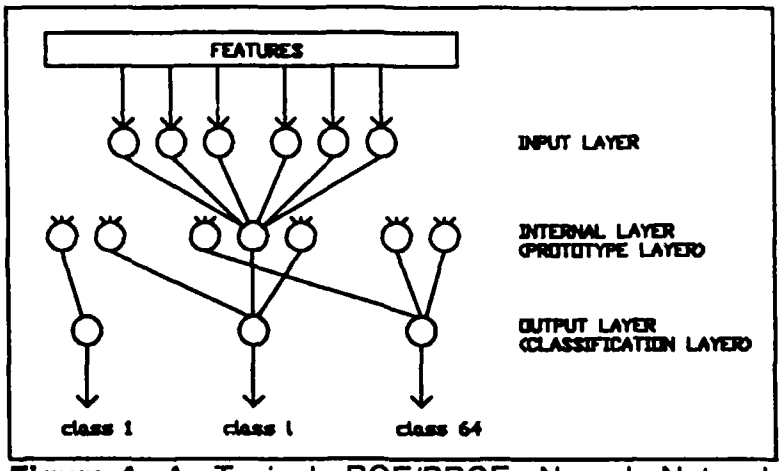


Figure 4 A Typical RCE/PRCE Neural Network consisting of three layers : input, internal (prototype) & output (classifications).

The input layer accepts the input vector and presents it to the prototype layer. Each prototype element in the NS1000's internal layer calculates the *city block* distance between the input vector and the stored feature vector as,

$$d^j = \Sigma_i \mid u_i - p_i^j \mid \qquad (1)$$

where d is the *city block* distance between the input vector ($\underline{u}$) and the stored feature example (prototype vector : $\underline{p}$), j is the index for the prototype element which may range from 1 to 1024 and i is the dimension index which ranges from 1 to 256. When the input vector $\underline{u}$ is similar to the stored prototype vector $\underline{p}^j$, the metric d^j will be small.

2.1.2 RCE Classification

The functional block diagram for the RCE classification mode is shown in figure 3. In the RCE mode, associated with each prototype element j is a threshold value λ^j to which the metric d^j will be compared against. When the metric d^j is less than the threshold λ^j, we have a match between the input vector and the stored feature example. Each prototype is identified with one class and a class output is simply an OR'ed combination of the respective prototypes. Hence, in the RCE mode of operation, when the input matches a feature stored in a prototype, one of the class outputs will give a positive response. The NS1000 may accommodate up to 64 classes.

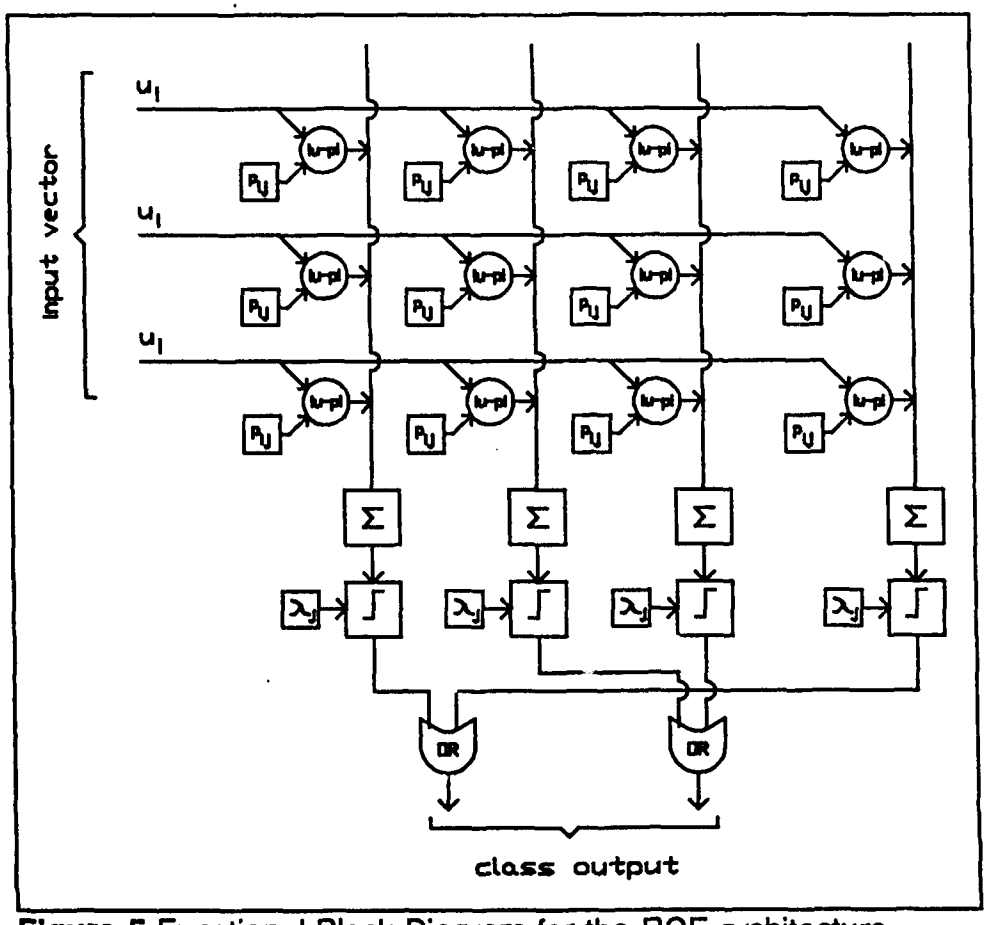


Figure 5 Functional Block Diagram for the RCE architecture.

2.1.3 PRCE Classification

In the situation when the input causes more than one class to give positive response due to ambiguous features, the NS1000 will no longer provide an unequivocal classification based on the RCE strategy. In this case, the NS1000 will enter the PRCE mode. The functional block diagram for the PRCE mode is shown in figure 4. The PRCE classifier is basically a Bayesian classifier.

A Bayesian classifier computes the relative risk between the various classes using the corresponding class probability density functions (PDF's). In a simple two-category situation (class A and class B), classification of an input vector $\underline{u}$ will be :

$\underline{u}$ belongs to class A when $C'_A f_A(\underline{u}) > C'_B f_B(\underline{u})$ (2a)

or

$\underline{u}$ belongs to class B when $C'_A f_A(\underline{u}) < C'_B f_B(\underline{u})$ (2b)

where C'_A and C'_B are the *a priori* probabilities of occurrence of patterns from category A and B respectively. The *a priori* probability C'_A is defined as the ratio of the number of training patterns that belong to class A versus the total number of training patterns and $C'_B = 1 - C'_A$. Functions f_A and f_B are the probability density functions for class A and class B respectively.

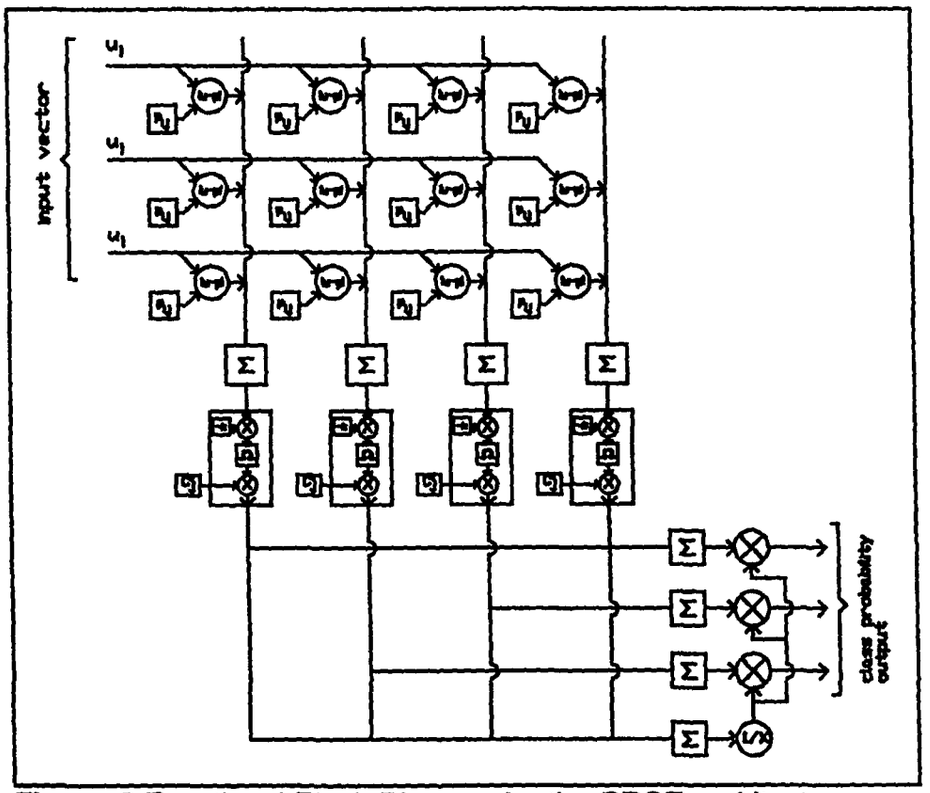


Figure 6 Functional Block Diagram for the PRCE architecture

The construction of decision boundaries requires the knowledge of the underlying PDF's which are usually unknown. A scheme to construct the PDF's was proposed by Parzen. In this scheme, the underlying PDF for class N may be constructed from a linear combination of a family of m distribution functions which are centered around the training feature examples, namely,

$$f_N(\underline{u}) = \{ \Sigma_{j=1}^{m} A_j \Omega[k * (\underline{u} - \underline{p}^j)] \} \qquad (3)$$

where $\Omega[]$ is a known and well behaved distribution function, $\underline{p}^j$ is the center of the j^{th} distribution function corresponding to the j^{th} training feature vector from class N, and A_j and k are constants. Figure 5 illustrates a two-category situation having two training feature examples per class and the function $\Omega[]$ is Gaussian in nature. The quantity ($\underline{u}$ - $\underline{p}^j$) in equation (3) is basically a measure of the closeness between the input vector $\underline{u}$ and the j^{th} training feature example $\underline{p}^j$.

For the NS1000 operating in the PRCE mode, the quantity d^j from equation (1) will be used to represent the quantity ($\underline{u}$ - $\underline{p}^j$) in equation (3). The distribution function $\Omega[]$ takes the form of a decaying exponential. The probability density contribution from the j^{th} prototype to the total probability density for class I will be

$$q(\underline{u})_I^j = C'^j_I \, \Omega[-k * d^j] \qquad (4)$$

where k is a decay constant, and C'^j_I is the *a priori* rate of occurrence based on the training pattern set. Here, C'^j_I is defined as the ratio between the number of training patterns that belonging to class I that fall within the distance λ^j from the training feature example $\underline{p}^j$ to the number of all the training feature examples that fall within that same distance from $\underline{p}^j$. Finally, the NS1000 will perform an accumulation of the q_I^j's pertaining to each class and output a normalized class probability as,

$$P(\underline{u})_I = \{ \Sigma_I (q_I^j) \} / \{ \Sigma_{all} (q_I^j) \} \qquad (5)$$

where the numerator is the accumulation over all prototypes belonging to class I and the denominator is the accumulation over all existing prototypes. The classification decision is simply the comparison of the P's among the various classes.

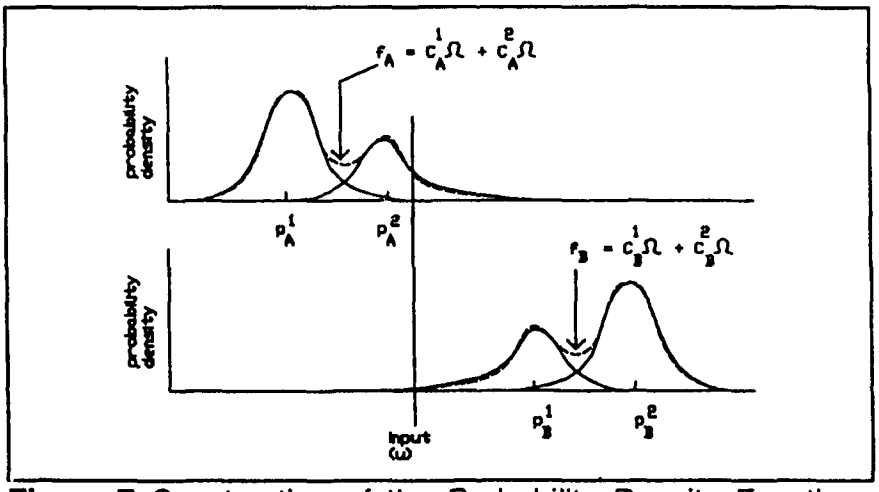


Figure 7 Construction of the Probability Density Functions based on the Parzen scheme for a two-category one-dimensional situation. There are two training feature examples per class and the function Ω is a Gaussian.

2.2 Learning Mode

2.2.1 RCE Learning

Learning in the relatively simple RCE mode is a process of selectively memorizing a set of training feature examples and the adjustment (reduction) of the corresponding matching threshold values λ^j.

2.2.2 PRCE Learning

When the class boundaries overlap and the NS1000 enters the PRCE mode, learning is a process of extracting an estimate of the underlying probabilities functions outlined in equation (3) based on the training feature examples. As described earlier, the NS1000 assumes the distribution function Ω[] taking the form of a decaying exponential[1]. The PRCE learning is a process of memorizing the training exemplars and simultaneously tracking the *a priori* rate of occurrence C'.

[1]The distribution function Ω[] used in NS1000 is an approximation to the true decaying exponential.

Actual learning for the NS1000 follows the scheme for the RCE case with the selective memorization of the training exemplars (engagement of prototype elements) and the adjustment of the matching threshold values λ while simultaneously tracking the *a priori* rate of occurrence C'.

3 MODES DESCRIPTION

3.1 CLASS : Classification

This is the normal processing (classification) mode of the NS1000. The host will send a command word to the NS1000's COMMAND register by writing the instruction on the input bus ID0-ID7. The NS1000 will proceed to clear the necessary data registers and waits for the initiation of the transfer of the input data.

The scheme calls for the host to set a signal INPTLD# to signal the start of the input loading operation. The host can terminate the load operation by setting the INPTLAST# signal. Loading of the input data are performed over the pins ID0-ID39 in a 'burst mode' fashion at the clock rate up to 25 Mhz. The NS1000 acknowledges the successful transfer of 1 bus-width of data (40 bits) by setting the signal INPTREADY#.

After the inputs are loaded, the NS1000 will proceed to calculate the sum of the absolute differences (d's) between the inputs and the prototypes. The individual d's are compared to the lambda's and decision based on the RCE rule will be made. When the RCE class outputs are ready, a signal RCE# will be sent. Up to two RCE classes will be available on the output bus (C0-C6 and C16-C22). When there is more than one RCE class output, the NS1000 will assert the signal MULTICLASS#. This signal is essential in a multi-chip environment. The class identification is a 7-bit binary coded word with the lower 6 bits representing 1 of the 64 classes and the 7th bit is a flag signaling no identification. A confidence level representing the sample density for the two RCE classes will be binary coded at the locations C10-C15 and C26-C31 respectively. Low sample density implies low confidence. The RCE classifications can be read in one clock cycle.

If more than one class are active, the NS1000 will set the signal PRCE# indicating that the PRCE classification rule will be used. The host needs to initiates the transfer of the class probability outputs by setting the signal CLASSLD#. Four 10-bit class probabilities are available on the output bus (C0-C9, C16-C25, C32-C41 and C48-C57) at each clock cycle. A total of 16 clock cycles will be needed to complete the transfer.

In a multi-chip environment, the following cases will arise during classification and the system controller will need to make the correct final determination.

Case 1 : RCE, 1 class, 1 chip --> system accepts answer

Case 2 : RCE, 1 class, >1 chip --> system accepts answer

Case 3 : RCE, >1 class, >1 chip, MULTICLASS signaled, RCE unambiguous NS1000 proceeds to perform PRCE calculations --> system waits for PRCE answers from all chips and perform final normalization

Case 4 : RCE, >1 class, 1 chip, MULTICLASS signaled, NS1000 proceeds to perform PRCE calculations --> system waits for PRCE answers

Case 5 : RCE, >1 class, >=1 chip --> system determines final class by voting and confidence level

Case 6 : PRCE, >1 class, >1 chip --> system accumulated 'raw' probabilities from chips and perform final normalization

Signals Needed : INPTLD#, INPTREADY#, INPTLAST#, RCE#, PRCE#, MULTICLASS# , CLASSLD#, CLASSREADY#, CLASSLAST#, ID0-ID39, C0-C63

NS1000 Registers : COMMAND

3.2 INPTLOAD : Input Load

This mode readies the NS1000 to receive from 1 up to 256 5-bit inputs. The host needs to set the signal INPTLD#. The transfer of the input data is in a 'burst mode' fashion. The NS1000 acknowledge the successful transfer of one bus-width of the input data by setting the signal INPTREADY#. The host may terminate the 'burst mode' transfer by setting INPTLAST# invalid. The NS1000 supports transfer clock rate up to 25 Mhz.

Signals Needed : INPTLD#, INPTREADY#, INPTLAST#, ID0-ID39

NS1000 Registers : COMMAND

3.3 PROTOWRITE : Prototype Write

The PROTOWRITE mode set up the NS1000 for copying the data on the input registers to a prototype. The host will sent the mode word to the COMMAND register and the prototype address (A0 to A9) to the address register. Upon receiving the mode instruction and the prototype address, NS1000 will begin by performing a block erase for the targeted prototype using the Intelligent Erase Algorithm. The programming of the cells will be sequenced internally. Upon completion of the writing of the weights, a signal EOP# will be sent.

Signals Needed : EOP#, A0-A9

NS1000 Registers : COMMAND

3.4 PROTOCLEAR : Prototype Clear

The PROTOCLEAR mode set up the NS1000 for clearing one specified prototype. The host will send the mode word to the COMMAND register and the prototype address to the address register (A0-A9). NS1000 will internally perform the Intelligent Erase Algorithm and send the EOP# signal upon completion.

Signals Needed : EOP#, A0-A9

NS1000 Registers : COMMAND

3.5 WEIGHTREAD : Weight Read

The WEIGHTREAD mode set up the NS1000 for the reading of weights in the prototypes. The host will send the mode word to the NS1000 COMMAND register and the prototype address to the address buffers (A0-A9 & A10-A17). The NS1000 will read the weight of the prototype according to the address specified and load the result to the first output register. (There are four output registers each having a width of 10 bits.)

Signals Needed : A0-A9, A10-A17

NS1000 Registers : COMMAND

3.6 PROTOREAD : Prototype Read

This mode allows the user to access the sum of the absolute differences at a specified prototype (ie the d's). The host first sends the command word to the NS1000 COMMAND register and the prototype address (A0-A9) to the address register. The corresponding 'd' value will be sent to the output pins C0-C12.

Signals Needed : C0-C12

NS1000 Registers : COMMAND

3.7 LAMBDALOAD : Lambda Load

The LAMBDALOAD mode allows the host to load in the starting lambda values for the prototypes. The host will send a command word to the COMMAND register and the prototype address to the address buffer A0-A9. The NS1000 will receive the 13-bit lambda value through the input lines ID0-ID12.

Signals Needed : ID0-ID12, A0-A9

NS1000 Registers : COMMAND

3.8 RESET : Reset

The RESET mode enables the host to clear the input registers of the NS1000 and re-upload the lambda's, C's, and other stored computational constants from the Flash back-up memory. The host initiates the procedure by writing the command word to the NS1000 COMMAND register. Upon the completion of the reset procedures, the NS1000 acknowledges by setting the signal EOP#.

Signals Needed : EOP#

NS1000 Registers : COMMAND

3.9 BACKUP : Backup

This mode essentially is the opposite of the RESET mode in which the computational constants, lambda's, C's, etc, which are stored at the internal RAMs are loaded to the back-up Flash memory. The host initiates this procedure by writing the mode word to the COMMAND register. The NS1000 will then enters the back-up procedure by first clearing the back-up Flash memory followed by copying the RAM based data to the Flash memory. The NS1000 acknowledge the completion of this procedure by sending the signal EOP#.

Signals Needed : EOP#

NS1000 Registers : COMMAND

3.10 RAMREAD : RAM Read

This mode allows the host to read the stored lambda's, C's, etc, that are stored in RAMs. The host initiates this procedure by writing the command word to the NS1000 COMMAND

register followed by the prototype address (A0-A9). The NS1000 will begin to output the RAM data (8 bytes) to the output pins C0-C63.

A 'burst mode' similar to the output of the class probability may be adopted.

Signals Needed: C0-C63

NS1000 Registers : COMMAND

3.11 **LEARN : Learn**

The host initiates the LEARN mode by sending a command word to the NS1000 command register. The host then present the training input vector to the NS1000. The host also needs to set one of the 64 output pins to high to inform the NS1000 the class associated with the input vector. The first phase of the learning mode involves the engagement of new prototypes. Depending on whether there is any activity from existing prototypes, the NS1000 will proceed to engage a new prototype by copying the inputs to the prototype Flash array. In the second phase, the NS1000 will adjustment of the lambda values and update the C values for the prototypes. Upon completion of the above procedures, the EOP# signal will be sent.

Signals Needed : C0-C63, EOP#, INPTLD#, INPTREADY#, INPTLAST#

NS1000 Registers : COMMAND

3.12 **RCE_ONLY : RCE Only Mode**

The RCE_ONLY mode enables the host to suppress the NS1000 from entering the PRCE calculations even under ambiguous RCE classifications. The host initiates this by sending the command word to the NS1000 COMMAND register. When this mode is set, the NS1000 will not enter the PRCE mode and only perform RCE classifications. This is most useful in situations when ambiguous RCE classes are rate and speed is important.

Signals Needed :

NS1000 Registers : COMMAND

3.13 **PROB_DENSITY : Probability Density Mode**

The PROB_DENSITY mode enables the host to suppress the NS1000 from performing the final normalization for probability outputs. Instead, the un-normalized probability

density as defined by the numerator in equation (5) will be made available. This mode is needed in a multi-chip environment where the final class probability must be calculated by the host. The host initiates this by sending the command word to the NS1000 COMMAND register.
Signals Needed : C0-C63

NS1000 Registers : COMMAND

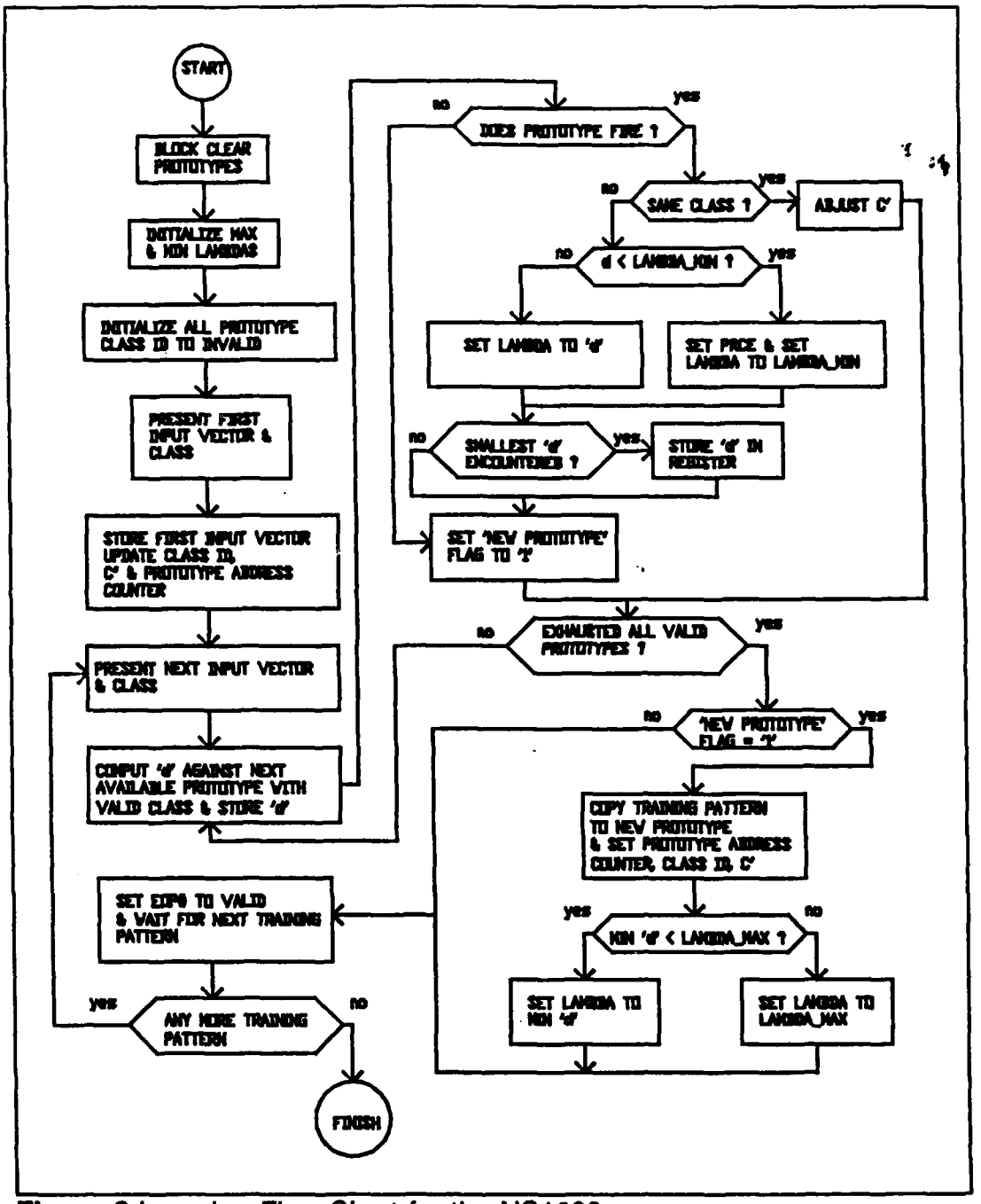


Figure 8 Learning Flow Chart for the NS1000

4 PIN DESCRIPTIONS

4.1 ID0-ID39

ID0 to ID39 are the digital input pins for the (8) input channels. These pins are divided into eight groups each having 5-bit per channel. These pins are directly connected to the external input data bus and are fully TTL/CMOS compatible. Supports up to 25 Mbytes/sec input speed.

4.2 C0-C63

C0 to C63 are digital output pins for the NS1000 and the class input pins during the learning phase. These pins are divided into four groups each having 16 bits per output channel.

The NS1000 will make available up to 2 classes in the RCE mode. These classes are binary coded in locations C0-C6 and C16-C22 in which the lower 6 bits contains the class number and the 7th bit flags the presence of invalid classes. Locations C10-C15 and C26-C31 contains the binary coded level of confidence.

When PRCE mode is needed, four 10-bit class probabilities are available on the output locations C0-C9, C16-C25, C32-C41 and C48-C57 respectively at each clock cycle. NS1000 supports output rate up to 25 Mhz.

During the NS1000 LEARN mode, one of the 64 output pins will need to be set high for class identification.

4.3 INPTLD#

This digital input pin is used by the host to signal the beginning of the input load process. The signal INPTLD# is active LOW.

4.4 INPTREADY#

The INPTREADY# signal is to inform the host that the NS1000 has accepted the data presented at the input bus and will be ready for the next transfer. INPTREADY# is active LOW.

4.5 INPTLAST#

This INPTLAST# signal is used by the host to indicate that the next time INPTREADY# is returned by the NS1000, the burst loading cycle is complete. INPTLAST# is active LOW.

4.6 RCE#

This digital pin is used by the NS1000 to inform the host that the RCE class output are ready. This signal is active LOW.

4.7 PRCE#

This digital pin is used by the NS1000 to inform the host that there is confusion in the RCE classification mode and will proceed to perform PRCE computations. This signal is active LOW.

4.8 MULTICLASS#

This pin signals that the NS1000 has detected more than 1 class in the RCE mode. This signal is active LOW.

4.9 CLASSLD#

This signal is used by the host to inform the NS1000 that the host is ready to begin the transfer of the class outputs. This signal is active LOW.

4.10 CLASSREADY#

This signal is used by the NS1000 to inform the host that valid data is presented on the output bus. This signal is active LOW.

4.11 CLASSLAST#

This signal is used by the NS1000 to inform the host that class output transfer is complete. This signal is active LOW.

4.12 A0-A9 & A10-A17

These are the digital address pins. Address pins A0-A9 are used to specify the prototype address and pins A10-A17 are used to specify the row address.

4.13 **VPP**

VPP is the high voltage input required for the program and erase of the Flash memory. Typical voltage for VPP is 12V. This pin may also be used in the Flash Vt mode.

4.14 **EOP#**

This digital pin is used by the NS1000 to inform the host the completion of certain processes (prototype write, etc.). This signal is active LOW.

4.15 **CE#**

CE# is a digital input which will enable the NS1000 when it is set to low. The NS1000 enters the standby mode when CE# is set to high.

4.16 **OE#**

OE# is digital input which will enable the NS1000's output pins (C0-C63) when it is set to low. The outputs will enter high-impedance state when OE# is set to high.

4.17 **CLK1**

The CLK1 pin is the NS1000's system clock input. This clock drives all of the NS1000's internal synchronous circuity. Clock rate up to 40 Mhz will be acceptable.

4.18 **CLK2**

The CLK2 pin controls the timing of the I/O pins. Maximum clock rate up to 25 Mhz is supported by the NS1000.

4.19 **VCC**

VCC is the +5V input supply pin.

4.20 **VSS**

VSS is the ground pin.

5 ELECTRICAL DATA

5.1 D.C. Specifications

Functional Operating Range V_{CC} = 5V +/- 5%; T_{CASE} = 0°C to +85°C

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input LOW Voltage	-0.3	+0.8	V	
V_{IH}	Input HIGH Voltage	2.0	V_{CC} + 0.3	V	
V_{OL}	Output LOW Voltage		0.45	V	
V_{OH}	Output HIGH Voltage	2.4		V	
I_{CC}	Supply Current CLK=40Mhz		900	mA	V_{CC} @ 5V
I_{LI}	Input Leakage Current		+/-15	uA	
I_{LO}	Output Leakage Current		+/-15	uA	
C_{IN}	Input Capacitance		15	pF	
C_O	Output Capacitance		15	pF	
C_{CLK}	Clock Capacitance		20	pF	
I_{PP1}	Program Current		30	mA	$V_{PP} = V_{PPH}$ Programming in Progress
I_{PP2}	Erase Current		30	mA	$V_{PP} = V_{PPH}$ Erasure in Progress
I_{PPS}	V_{PP} Leakage Current		+/-10	uA	$V_{PP} = V_{PPL}$
V_{PPH}	V_{PP} during P/E Operations	11.4	12.6	V	

5.2 A.C. Specifications

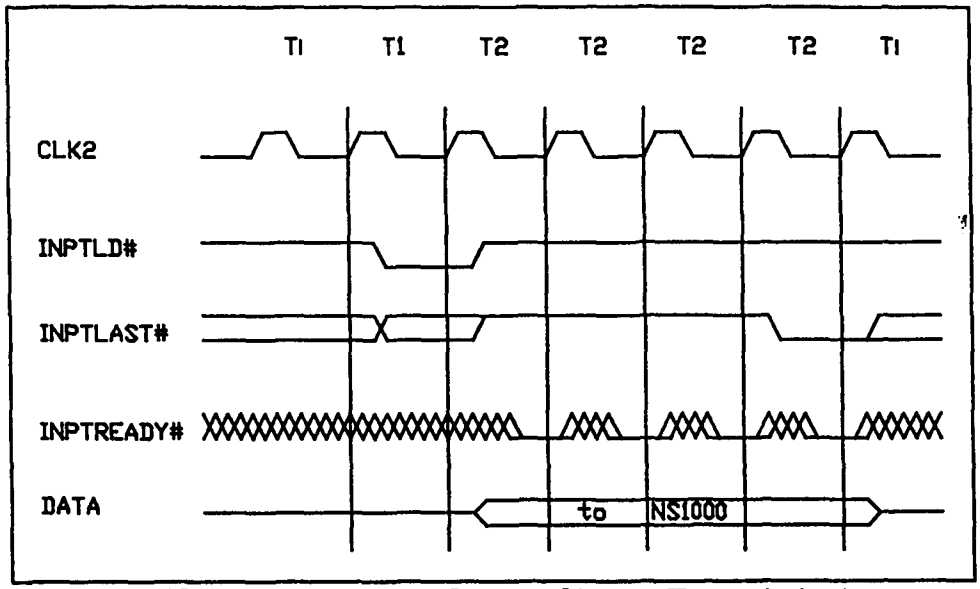


Figure 9 NS1000 Burst Input Cycle. CLK2's T1 cycle is the set up phase and T2's are the transfer phase.

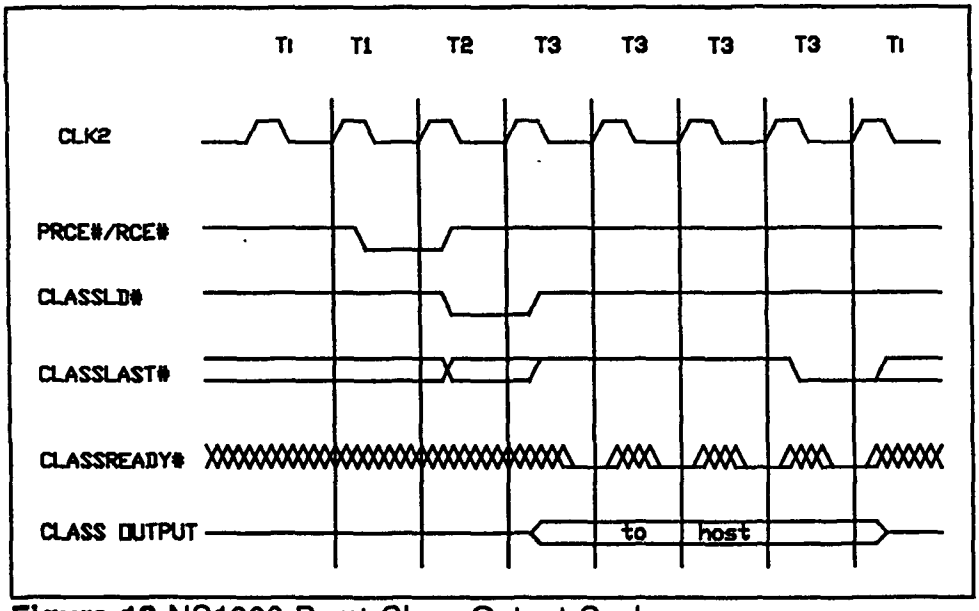


Figure 10 NS1000 Burst Class Output Cycle.