Fifth IFSA World Congress (1993), 793 - 797

High-Performance Pattern Classifier

C. Park, M. Holler, J. Diamond, S. The, U. Santoni, K. Buckmann
Intel Corporation, RN3-17, Santa Clara, CA 95052, U.S.A.

M. Glier, L. Nunez, J. Cole
Nestor Inc., Providence, RI 02906, U.S.A.

Abstract

20 BOPS (Billion Operations Per Second) classification processor has been implemented in Silicon with massive parallel and pipeline architecture. It is suitable for RBF neural networks with Bayes criterion and is applicable to fast fuzzy processing. The resident micro-controller handles learning and other S/W processing. One chip processes up to 1024 prototypes in 256 dimensional feature space. The chip has 3.7 M transistors on 1.55x1.33 cm^2 Si estate with a 0.8μm FEPROM process and is packaged in 168 pin PGA. Functionality has been confirmed at its first stepping.

Introduction

Classifier based on Radial Basis Function (RBF) [2,3] or prototypes (cluster center) with local receptive fields is intuitively sound due to the direct representation of clusters of training samples. RBF neural network inherently fits to statistical approach as Probabilistic Neural Network (PNN) [4,5] and PRCE [6]. It is generally known that learning of RBF type networks is far faster than back propagation with comparable generalization ability [5]. The locally generalization features of RBF's are particularly attractive for VLSI implementation [6,7,8].

By defining a crisp boundary (radius) to the receptive field, the network could be further simplified in both learning and application, such as Restricted Coulomb Energy (RCE)[6]. When regions of classes are overlap in feature space, PNN is suitable with the RBF used as a probability density function (PDF).

The chip is an implementation of RCE and PNN with a massive parallel and pipeline architecture. Resident microcontroller supports variations of learning and post-processing, if required.

The calculates the city block distance (L1 norm) between a given input pattern and all committed prototype vectors as

$$D_i = \sum_{j=0}^{j<DIM} |Xi_j - P_j| \qquad (1)$$

where D_i is distance between the i-th prototype $\mathbf{X}_i = (Xi_0, Xi_1, ..., Xi_{DIM})$, and the input vector, $\mathbf{P} = (P_0, P_1, ..., P_{DIM})$, and DIM is the dimension (≤ 256). City block distance was adopted to simplify the implementation, mainly in the physical design. The difference can be visualized as a polyhedron from L1 norm [6] while a circle or sphere from L2 norm for contour.

After distances are calculated, a fired class list from the RCE method and PDF for each class are generated in the pipeline. Each prototype has an associated class-id attribute, which is stored during the supervised learning process with other attributes as shown in Table 1. The input pattern is recognized as class k if

$$Fired_k = \begin{cases} 1, & \text{if } D_i < L_i \text{ for any prototype i in class k;} \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

where L_i is the threshold parameter (radius of receptive field) of the i-th prototype.

If only one class is fired, primarily classification process is completed for the given input. Otherwise further processing with the PDF would be required with Bayes or nearest neighbor methods. PDF for class k is calculated as

$$PDF_k = \sum_{For\ All PT_i of Class\ K} C_i \cdot exp(K_i \cdot D_i) \qquad (3)$$

where C_i is the amplitude constant representing *a priori* and K_i is the decay constant for the i-th prototype. Examples of single un-normalized PDF and mixture of multiples are shown in Fig 1. The chip calculates un-normalized PDF in 16 bit floating point form; i.e., 6 bit exponent and 10 bit mantissa. This mixture density function can be used as a membership function of the class. For fuzzy application, semantic rules or de-fuzzification should be done by an external host or by the resident μC.

The prototype parameters, defining the radius of receptive field in RCE and the shape of RBF (PDF, here) are initialized when its prototype is created and updated during subsequent learning. Learning is accomplished with software using the final or intermediate result of classification pipeline

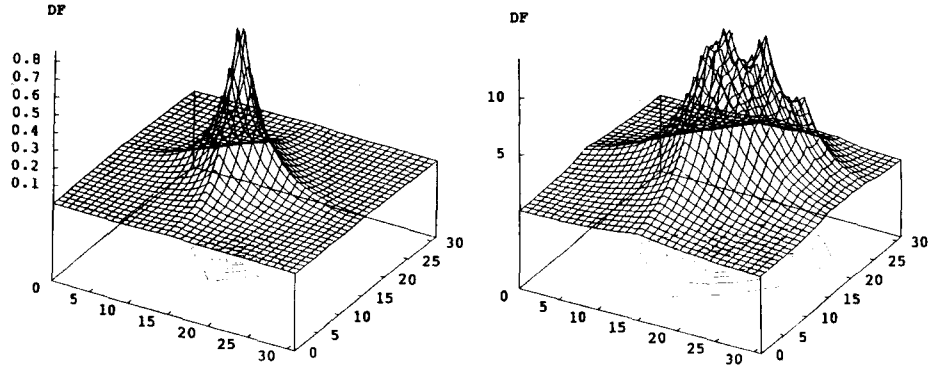

Fig. 1 Un-normalized PDF used in the chip.
a) Class with single PT [center:(15,15), C=1, K=15/32]
b) Class with four PT's [(15,15), C=10, K=1/4], [(17,17), C=8, K=15/32], [(20,20), C=10, K=15/32], and [(23,23), C=5, K=15/32]

(yet the high performance parallel computation blocks are used). RCE learning [6] loads new prototype when the desired class is not fired. If any prototype of incorrect classes are fired, their threshold should be reduced such that they are not fired for the input pattern but the radius should not be reduced less than the prescribed minimum (DMIN); otherwise a large number of prototypes could be generated in the learning process. If the threshold reaches DMIN, the reserved flag representing a low-confidence level, R_i is set. PNN (or PRCE) learning to update parameters for the RBF shape can be merged with RCE learning [6].

Sybmol	Name	Range
N_i	Class ID	0 - 63
L_i	Threshold	0 - 4095
K_i	Decay Constant	2^{-20} – 15/32
C_i	Amplitude Const.	0 - 65535
U_i	Used Flag	0, 1
B_i	Bad Flag	0, 1
R_i	Confidence Flag	0, 1

Table 1 Prototype parameters

Architecture & Implementation

The chip has two major parts: the classification pipeline and the micro-controller. Classification pipeline is composed of prototype array (PA), distance calculation units (DCU), parameter storage RAM (PPR), math unit (MU), IO buffer RAM (IBR, OBR) and classification controller(CMC).

Micro-controller(μC) part is composed of μC core, program storage Flash EPROM(PGF), general purpose RAM (GR), and Timer. μC can access classify pipeline arrays when classify mode is not active. The modules in classification pipeline have their own control and status registers which μC can access any time except the test mode when μC is inactive;i.e., μC controls and monitors the classify operation.

Classify mode data flow (shown in Fig.2) can be described as

- IBR accepts the input vectors of the specified dimension in either 32 bit or 64 bit mode with normal 4 clock access or burst mode. When a complete vector has been received, IBR sends IBFULL signal to CMC. Two RAMs in IBR are used for continuous operation of the classify pipeline.
- CMC activates DCU when IBFULL signal and distance latches are free. Most of the DCU circuitry are used in 2X time sharing manner if there are more than 512 committed prototypes. When distance calculation of the first 512 PT's is completed, CMC activates the MU and transfers distance and prototype parameters to MU. At the same time, DCU calculates distance for the remaining prototypes.
- MU builds the fired class table and calculates PDF.
- When distance calculation for the second half of the prototypes and MU operation for the first half are completed, CMC send IBR the signal indicating completion of input vector access. If a new input vector is already in the second IBR, distance calculations for the new input vector starts immediately.
- When MU operation for all PT's is completed, CMC send MURDY signal to OBR. While OBR gets the data, MU can start processing the next input vector immediately when distance data is available.
- OBR obtains the fired class list and/or PDF from MUR, and outputs to external host in 32 bit or 64 bit wide with normal or burst manner. OBR has a floating point formatter to convert 16 bit to the 32 bit IEEE standard format. The conversion is requested by setting the corresponding control flag in IOC.

The timing diagram of the pipeline is shown in Fig. 3.

Design Strategy

To reduce the complexity of this full custom chip, a modular design with a robust interface scheme had been chosen. The internal bus protocol was designed to allow enough clock skew. With this setting, local clock buffers are used for each module. Selectively non-overlapping clock generators are used also. This scheme reduced loading of the global clock driver to around 50pF. Tapered clock line was also used due to the dominance of clock line parasitics. The classification pipeline had special consideration for timing correctness.

Most of power dissipation occurs at the 512 DCUs and their 2560 sense amps. When classify mode is not active, the current consumed by μC, PGF, and all the RAM's is 30mA (typical) at 5V. In the classify mode, power supply

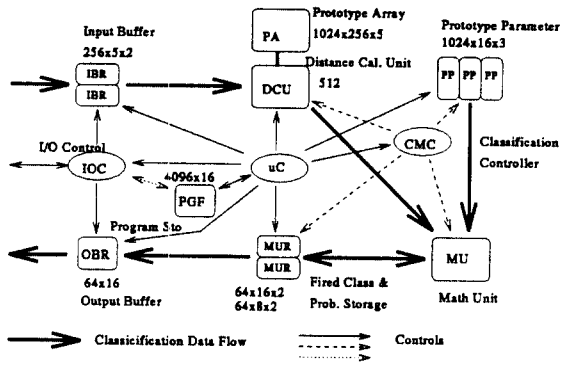

Fig. 2 Controls and Classify Data Flow

current is proportional to the number of prototypes used. When all prototypes are committed, it consumes up to 0.8 A. The circuit and layout for the parallel computation unit was carefully designed with noise considerations.

Prototype Array and Distance Calculation Unit

DCU(distance calculation unit) is the part that reads the prototype array(PA), calculates the City Block Distance between the prototype vectors stored in PA and a given input vector, and stores/drives the calculated data on the local pre-charged bus for MU and μC. The additional access path for programming, erasing, and verifing the PA is also part of DCU. For massive parallel architecture, timing, power dissipation, and physical size are far more significant than those for single process case.

The Parallel Sense Amp (PSA, here the term "parallel" is attached to distinguish the conventional sense amp in the chip to verify PA.) are simple cross-coupled circuitry with bit-line biasing, bitline equalizing, and sense amp output equalizing features as shown in Fig.4. No static current is required for this PSA. During classify mode, the bias voltage (SABIAS in Fig.4) is steady around 2V to generate bitline voltage around 1V. Timing for the two equalization controls and latch load signal are shown in Fig.4. Sense Amp Latches (after PSA's) are for the pipeline operation.

The distance calculation, $|A-B|$, where A is the element of vector read from PA and B is the element of input vector, is implementation of the following scheme,

- Add A and one's complemented B
- If there is no carry, the inverted sum is the difference; else the (sum + 1) is the difference.

The distance accumulator is used to add the difference for all dimensions of a vector up to 256 dimension. It is implemented with 5 bit accumulator [full adders with latches] and 8 bit counter to handle 13 bit summation. Finally the latch and drive part has two distance latches (DL) to accommodate 2x time sharing the most of DCU circuitry with pipeline.

DCU uses two "used" flags (for 2X time multiplexing) to gate the global control signals. When the corresponding "used" flag is not set, effectively the DCU is disabled for zero power dissipation. This control scheme reduces loading at the global control drivers, provides localized operation for

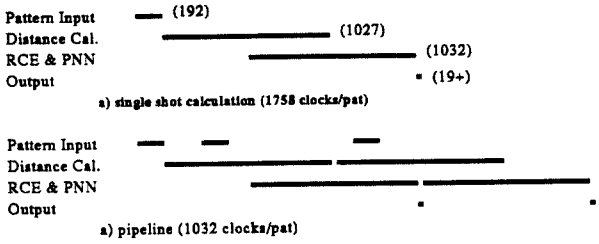

Fig. 3 Classify pipeline timing diagram for 1024 PT's with 256 Dim & 64 Classes

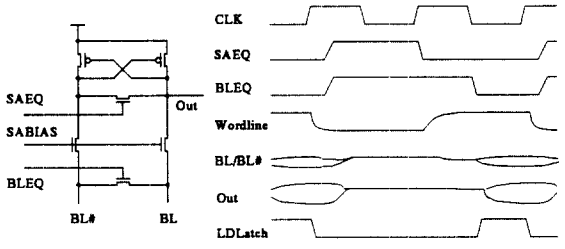

Fig. 4 DCU Sense Amp and Timing Diagram

DCU selection(enable) and decoding pathes are summarized as follows,

- All control signals for PSA, distance calculation, and DL are gated with the corresponding used flag.
- For "used" flag and distance latch access, an NMOS style decoding path is implemented inside DCU. During classify operation, the distance latch access and distance calculation can be done at the same time.
- For PA access from μC, separate access path with decoder is provided which is similar to conventional memory.

Pipeline techniques are used in DCU such as PA read, distance calculation, and its accumulation. FEPROM read takes two clocks as does the distance calculation. In the pipeline, 514 cycles are required to calculate 256 dimensional vector difference (2 x 256 + 2).

DCU and Math Unit operations are also pipelined. For the case of 1024 prototypes with 256 dimension, after the completion of the distance calculation of the first 512 prototypes in 514 clock cycles, MU handles RCE and PDF calculation for those 512 prototypes in 519 clock cycles, while DCU's calculate distances for the remaining 512 prototypes simultaneously. DCU can start distance calculation for the next input vector immediately.

Math Unit, MURAM, and PPRAM

MU is composed of 11 stage pipeline to calculate eq.(2) and (3); i.e., two multiplications, one exponential, one floating point summation, and a comparison in every cycle with 6 clock latency.

At the beginning of the calculation, 64 flags (fired), the fired class counter, and 64 Class ID's and PDF's are initialized. RCE part (fired class list generation) is as follows. When a prototype is fired, MU checks the flag whether its class was already fired or not. For the latter case (new fired class), set the fired flag, store the class to class list (MUR),

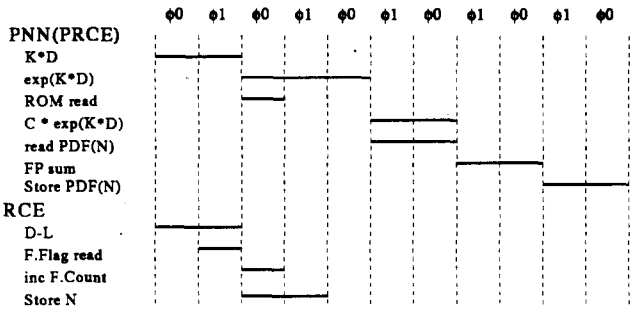

Fig. 5 MU Pipeline Stage

and increment the fired class counter. PDF calculation is done in a more complex way. First exp() calculation is implemented with the following method,

$$\begin{aligned} \exp(-D_i \cdot K_i) &= 2^{-X} \\ &= 2^{-S} \cdot 2^{-(X-S)} \end{aligned} \tag{4}$$

where $X = D_i \cdot K_i log_2 e$ and $S = \lceil X \rceil$. The latter part of eq(4) can be

$$2^{-(X-S)} = 2^{-T} \cdot 2^{-\epsilon} \tag{5}$$

where

$$T = X_M \cdot 2^{-M} \tag{6}$$

where X_M is an integer. In eq(5), $\epsilon < 2^{-M}$. The latter part of eq(5) is

$$2^{-\epsilon} = e^{-\epsilon \cdot \ln 2} = 1 - \epsilon \cdot \ln 2 + R \tag{7}$$

where

$$R \leq (\epsilon \cdot \ln 2)^2 < (2^{M} \cdot \ln 2)^2 \tag{8}$$

In the chip, exp() was implemented in three parts as follows, from eq's (4), (5), and (7),

1. 2^{-S} is implemented with shifter
2. 2^{-T} is from ROM table
3. $1 - \epsilon \cdot ln2$ is from simplified multiplier

The size of the ROM was chosen as 32 X 11 which is enough to guarantee less than 0.1% error in the exp() calculation.

Multipliers are implemented with modified Booth and Wallace tree.

MUR is composed of two sets of 64 word PDF storage and 64 byte fired class storage. Dual port configuration is used for the MU pipeline. Two identical sets are for continuous pipeline operation with flipping. If MUR is not in classify mode, it can be accessed by μC as a general purpose 16 bit and 8 bit RAM.

PPR's are for the prototype parameter storage. Three 1K word RAM's are accessible as general purpose RAM's in normal modes. In classify mode, all three are operating simultaneously to provide 48 bit parameters to MU every cycle.

Micro-controller and Program Storage

```
3 ;
4 ; Sample code to find class-id with the highest PDF
5 ;
6 000000  43 03 10 01 rdi 1001h, r3   ; R3 = (\#class - 1) > 0
7 000002  08 14 68 00 ldi 6800h, ds1  ; Load the base address of MURAM to
8                                     ;   DS1, index register
9 000004  12 31       mov r3, r1      ; R1 holds class\_id for max PDF
10 000005 5A 32       rd1rd r3, r2    ; R2 = Mem(DS1+R3) and decrement R3
11 000006 52 30       loop: rd1r r3, r0 ; R0 = Mem(DS1+R3), next PDF
12 000007 39 20       cmp r0, r2      ; Compare R2 with R0
13 000008 82 03       jnn skip        ; If (R2 >= R0) goto skip
14                                    ;  PDF is 16 bit floating point number but
15                                    ;  integer comparison works for the purpose
16 000009 12 31       mov r3, r1      ; New class\_id in R1
17 00000A 12 02       mov r0, r2      ; New max PDF in R2
18 00000B 28 30       skip: dec r3
19 00000C 82 FA       jnn loop        ; IF (R3 >= 0) goto loop
20 ;
21 ;  Now the class-id in R1
22 ;
```

Fig. 6 μC sample program list

and other operations (decoding, execution, ...). Most of instructions are two CPI except the load/store/jmp. The μC performs peak 25 Mips (average around 15Mips) at 50MHz.

μC has 13 registers (R[0:3], RZ, RO, DS[0:1], DR, SP, PC, IM, and PSW) and 64 dedicated stack locations. PSW has 15 flags for the status of μC, classify pipeline, and interrupt. Entry point of the interrupt service is stored at the first address of program storage.

μC has 61 instructions. Condition code can be referenced to any of 15 flag bits. Assembler list of a short sample program is shown in Fig. 6.

Instruction fetch from the 4 Kword PGF is done by every two cycle; i.e., 40us access at 50MHz. There is a special mode to load the program to PGF from external.

GR is a 256 words RAM for μC. The 32 bit unconditional clock counter is used for the reference of time lapse.

IO Controller and IO Buffer RAM

IO controller provides interface with external host. It has 16 registers accessible by external host, μC, and IOC; i.e., 2 control registers, 4 status registers, 11 data registers. The control registers are for the I/O buffer RAM controls. Four data registers are specific purposes such as dimension, number of classes, mode selection, and chip ID.

IBR is used to receive the input pattern vector in either burst or normal mode with 32 bit of 64 bit data input and to supply the data to PADCU upon request. It has two identical RAMs (256 X 5) for continuous pipeline operation with flipping.

OBR is used to convert floating point format, to pack output data (32 bit or 64 bit), and to facilitate the burst mode. It has 64 words storage and floating point conversion from the internal 16 bit to the 32 bit IEEE standard.

There are five modes of operations as follows,

- NORMAL: External host can access IO registers and μC program takes care all internal operations.
- CLASSIFY: CMC controls classify path covering I/O buffer RAM, PADCU, MU, PPR, and MUR. IOC does IO sequencing for external interface. During this mode,

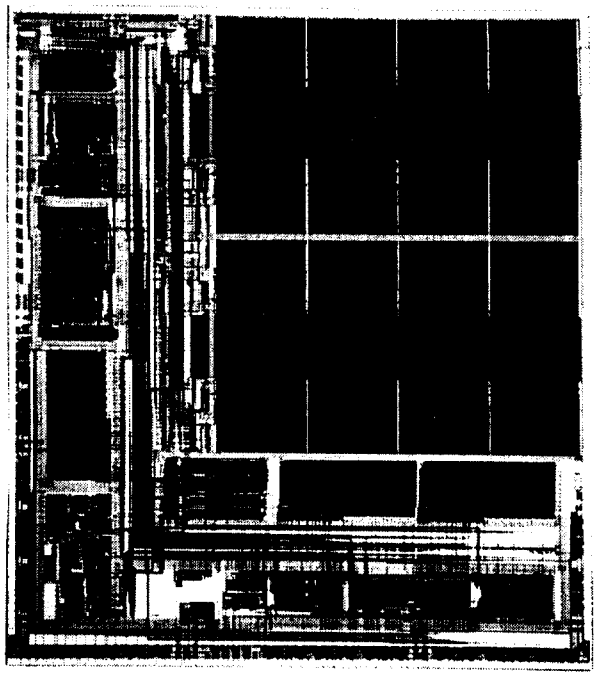

Fig. 7 Microphotograph of the chip

μC can monitor and control the operations of all units.

- MONITOR: IOC passes data on the internal bus to pins D[32:63]. Either instruction and program address buses or internal data and address buses are selected by the flag in IO control register.

- PGF: PGF is accessible to the external host in this mode to update the μC program. Other modules are disabled during this mode.

- TEST: Internal control signals are mapped to data pins and external host performs the function of μC. μC is disabled in this mode. PGF test is done separately. Bitlines of array can directly connected to pins by control register setting for individual array modules.

Experimental Result

For engineering evaluation, a logic analyzer system and PC with an evaluation board have been used. In the logic analyzer environment, 25MHz clock has been used for all tests during functional evaluation. At 50MHz, some parts showed marginal operation, partly due to the crude testing environment. PC evaluation board has 40MHz crystal and programmable clock divider.

All major functions have been verified such as μC blocks, classify pipeline for single shot and continuous inputs, IO operations, and the five mentioned modes. Operating frequency meets the target (40MHz) and power dissipation is around 4 Watt at the peak when all the DCUs are operating and around 150mW in non-classify mode. One bug in the high voltage path to PA array slowed down the programming of Flash EPROM, which will be fixed in the next stepping. Luckily it was the only bug found yet and it does not affect any other operations.

Chip Feature	Value
Operating Frequency	40 MHz
Peak Performance	20 GOPS
Array Bandwidth	55 GB/sec
Number of Transistors	3.7 M
Pattern Vector	256 x 5 bit
Number of Prototypes	1024
Die Size	15.5 x 13.3 mm^2
Power Dissipation	4 Watt (peak)
Process	2 Metal, 2 Poly, Flash ETOX
Package	168 pin PGA

Table 2. Chip features

Testability of the chip was sufficient for the evaluation. Especially the monitor mode and test mode were very effective for both hardware and software debugging. Cells in the arrays can be characterized in the test mode. Selftest program had been made for μC itself, memory modules, and classify pipeline, though it should be improved for more fault coverage. No micro- probing were performed during the whole engineering debug.

Conclusion

A high-density RBF neural network with on-chip software learning and exponential PDF calculation was implemented in Silicon and its functionality has been confirmed at the first stepping. The high-performance and efficiency of dedicated hardware and flexibility of software controls and post-processing make the chip attractive to fast pattern classifiers.

Acknowledgement

This work was supported in part by ARPA and ONR under contract No. N00014-90-C-0010. Authors would like to acknowledge the contributions of S. Tam, H. Castro, D. Seligson, and D. Reilly.

References

1. K. Fukunaga, "Introduction to Statistical Pattern Recognition", 2nd ed. Academic Press, 1990.
2. B.W. Mel and C. Koch, "Sigma-Pi Learning: On Radial Basis Functions and Cortical Associative Learning," pp.474-481, Advances in Neural Information Processing System 2, 1990.
3. D.S.Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," Complex Systems, pp. 321-355, 2, 1988.
4. D.F.Specht, "Probabilistic Neural Networks for Classification, Mapping, or Associative Memory," Proc. IEEE ICNN, vol.1, pp525-532, June 1988.
5. D.F.Specht, "Enhancement to Probabilistic Neural Networks," IJCNN-92, Baltimore, 1992.
6. C.L.Scofield and D.L.Reilly, "Into Silicon: Real Time Learning in a High Density RBF Neural Network," pp.551-556, I, IJCNN-91, 1991.
7. M. Holler, C. Park, K. Buckmann, et. al, "A High Performance Adaptive Classifier using Radial Basis Functions, GOMAC-92, 1992.
8. K. Uchimura, O. Saito, Y. Amemiya, "A High-Speed Digital Neural Chip with Low-power Chain Reaction Architecture," pp1862-1867, IEEE JSSC Vol.27, No.12, Dec. 1992.