A Fast Bipolar-Valued Inner Product Processor Chip for Associative Memory Networks

Chua-Chin Wang, Ya-Hsin Hsueh, and Chenn-Jung Hunng

***Abstract*—In this brief, a novel and high-speed realization of bipolar-valued inner product processor for associative memory networks is presented, wherein the treatment of inner product of two bipolar vectors is given. Besides, a systolic architecture of digital compressors is used to reduce the carry propagation delay in the critical path of the inner product of two bipolar vectors.**

***Index Terms*—Bipolar, digital compressor, valued inner product.**

I. Introduction

Applications of associative memories include pattern recognition [6], code correcting, storage of words and speech data [1], chaotic neural network [9]. In the computation of associative memories [8], the inner product of two vectors might be one of the most frequently used mathematical operations, since the inner product is the core process of recall computations of associative memory [10]. Accordingly, the demand of shortening the delay therewith becomes urgent. Notably, the bipolar-valued data are more commonly used in digital associative memories [3], [5]. Many efforts have been thrown on implementing the associative memories with hardware circuits [2], [4], [7]. However, all of these implementations pay attention to the realization of the binary-valued associative memories but leave the problem of inner product of two bipolar-valued vectors unresolved. In this paper, a bipolar-valued inner product processor for associative memory networks is proposed to compute inner product of two bipolar vectors, wherein a systolic architecture of the digital compressors based on the 3-2 compressor building blocks are included to compute the summation of the individual inner product terms.

II. Associative Memory Networks

A traditional feedforward heteroassociative network stores N-sample pattern pairs, which are $\{(X_1, T_1), (X_2, T_2), \cdots, (X_N, T_N)\}$, where $X_i \in \{-1, +1\}^m$, $T_i \in \{-1, +1\}^p$, and $i = 1, 2, \cdots, N$. The objective is to retrieve pattern $Y_k (k = 1, 2, \cdots, N)$ where $Y_k = T_k$ whenever X_k is the input to the network. We define a correlation matrix [5] $M_i = X_i^T T_i$ for the pair of patterns to be associated, (X_i, T_i). The individual pattern matrices M_i can then be superimposed to store N patterns in the matrix M, where $M = \sum_{i=1}^{N} M_i$. Now assume X_k is the input pattern, which is one of the stored patterns, Y is the output pattern, and X_i's and T_i's are stored pattern pairs. The input is processed and transferred to the output as follows:

$$\begin{aligned} Y &= \mathrm{th}\left(\sum_{i=1}^{N} T_i \cdot \left(X_i^T \cdot X_k\right)\right) \\ &= \mathrm{th}\left(m \cdot T_k + \sum_{i=1, i \neq k}^{N} T_i \cdot \left(X_i^T \cdot X_k\right)\right) \\ &= \mathrm{th}(m \cdot T_k + \eta) \end{aligned} \tag{1}$$

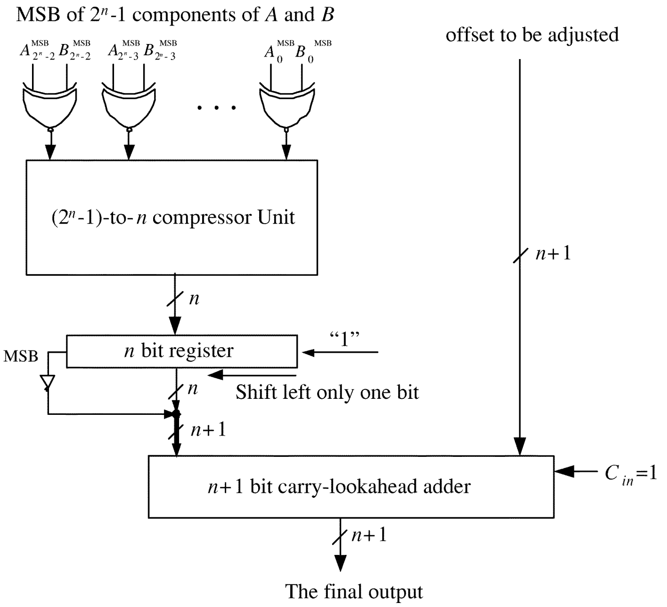


Fig. 1. Architecture of the bipolar-valued inner product processor.

where $\mathrm{th}()$ is a threshold function, and η is the noise term. Hence, if the vector X_k and $X_i (i \neq k)$ are orthogonal, then the noise term η is zero.

Autoassociative memory associates vectors from within only one set, which is $\{X_1, X_2, X_3, \cdots, X_N\}$, where $X_i \in \{-1, +1\}^m$. If some arbitrary pattern X is closer to X_k than to any other X_j, $j = 1, 2, \cdots$, then the network will produce the stored pattern X_k when the key pattern X is presented as an input. Namely, an undistorted prototype vector in response to the distorted prototype key vector can be recovered by the memory. Vector X_k can be regarded in such a case as stored data and the distorted key serves as a search key or argument. Notably, when the cardinality of pattern pairs is large in the above calculation, the carry propagation of the inner product of the bipolar-valued vectors will likely become the critical delay of the entire neural computing.

III. High-Speed Bipolar-Valued Vector Inner Product Processor

The entire design of bipolar-valued inner product processor is divided into four parts, including an individual inner product term generator, a compressor unit, a bipolar-to-binary value converter, and an inner product adjustment unit. The individual inner product term generator produces the individual inner product terms given two bipolar-valued vectors, and passes them to the compressor unit, which computes a summation of inner product terms. Then, a conversion from bipolar-valued digits to binary digits is used to feed a two's complement number into the last processing stage, where the augmented inner product is corrected to the precise value. Fig. 1 shows the architecture of the bipolar-valued inner product processor.

A. Inner Product Term Generator

Since -1 can be represented as $11 \cdots 11$, and $+1$ as $00 \cdots 01$ in two's complement form, we can use the most significant bits (MSB) of the vector components to differentiate between $+1$ and -1 when the inputs to the inner product processor are restricted to the bipolar values. The cardinality of the bipolar vectors may be less than or equal to the number of the XNOR gates, $2^n - 1$, which is exactly the count of the inputs to the $(2^n - 1)$-to-n compressor unit. In case that the length of

Manuscript received July 3, 2001; revised September 24, 2002 and March 7, 2003. This work was supported in part by the National Science Council under Grant NSC 88-2219-E-110-001. This paper was recommended by Associate Editor Y. Ismail.

C.-C. Wang and Y.-H. Hsueh are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan 80424, R.O.C. (e-mail: ccwang@ee.nsysu.edu.tw; andgel@vlsi.nsysu.edu.tw).

C.-J. Hunng is with the Department of Computer Science and Information Education, National Taitung Teachers College, Hualien,Taiwan 970, R.O.C.

Digital Object Identifier 10.1109/TCSI.2003.813972

TABLE I
TRUTH TABLE FOR SUMMATION OF THREE BIPOLAR BINARY INPUTS

Input	q	-1	-1	-1	1	-1	1	1	1
	r	-1	-1	1	-1	1	-1	1	1
	t	-1	1	-1	-1	1	1	-1	1
Output	S_1	-1	-1	-1	-1	1	1	1	1
	S_0	-1	1	1	1	-1	-1	-1	1

the bipolar vectors is less than $2^n - 1$, all the unused inputs are set to zeros, which pad the length of input vectors to $2^n - 1$. Therefore, the inner product needs to be adjusted to the precise value later because it is augmented at this stage by the accumulation of the unused XNOR gates' outputs.

B. $(2^n - 1)$-to-n Compressor Unit

The summation of three bipolar digits q, r, t can be expressed as follows:

$$q + r + t = S_1 \cdot 2^1 + S_0 \tag{2}$$

where S_1 and S_0 are bipolar digits.

All possible combinations of (2) are illustrated in Table I, which is identical to the truth table of a full adder if a logic LOW represents -1 and a logic HIGH represents 1 in this table. Therefore, we tend to construct the compressor unit with full adder building blocks to calculate the summation of the individual inner product terms.

1) Basic 3-2 Compressor Building Block: A 3-2 compressor is basically a full adder. The feature of such a compressor is that the output represents the number of 1's given in inputs. The equations of a full adder are presented as below:

$$\begin{aligned} C_{\text{out}} &= (a \otimes b) \cdot C_{\text{in}}(a \cdot b) \\ \text{Sum} &= (a \otimes b) \otimes C_{\text{in}} \end{aligned} \tag{3}$$

where a and b are inputs, C_{in} is a carry-in from a previous addition, C_{out} and Sum are the carry and the sum outputs, respectively. The output of the compressor unit denotes the number of logic HIGH at the inputs.

2) Framework of $(2^n - 1)$-to-n Compressor: The compressor unit we propose to improve the carry propagation delay of the critical paths consists of parallelized 3-2 compressor building blocks at every processing stage. To illustrate the functionality of the compressor unit, a 63-to-6 compressor that sums the 63 data inputs each with one bit is shown in Fig. 2. At the first processing stage, 21 3-2 compressors are used to generate 21 bits at the second bit and the least-significant-bit position, respectively. Then, 14 3-2 compressors at the second processing stage produce 7 bits at the third bit position, 14 bits at the second bit position, and 7 bits at the least significant bit position. Following the same fashion, a total of 57 3-2 compressors and nine processing stages are needed to produce the sum of 63 bipolar bits. The total delays are also approximately proportional to the count of 3-2 compressors that the critical path resides, as shown as the dashed line in Fig. 2.

The carry propagation delay of $(2^n - 1)$-to-n compressors, where n from 2 to 13, has been computed by a program and can be formalized as follows:

$$D_{\text{cmpr}}(n) = \begin{cases} 2n - 3, & 2 \le n \le 6 \\ 2n - 4, & 7 \le n \le 13 \end{cases} \tag{4}$$

where $D_{\text{cmpr}}(n)$ denotes the total delay of a $(2^n - 1)$-to-n compressor, counted by the number of 3-2 compressor blocks.

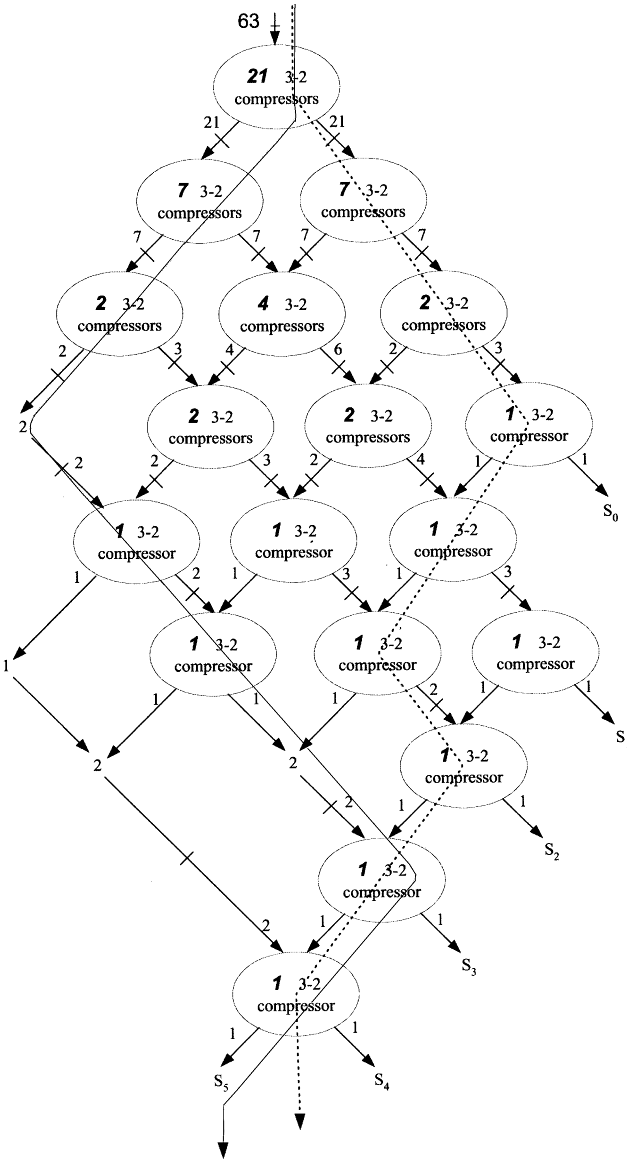


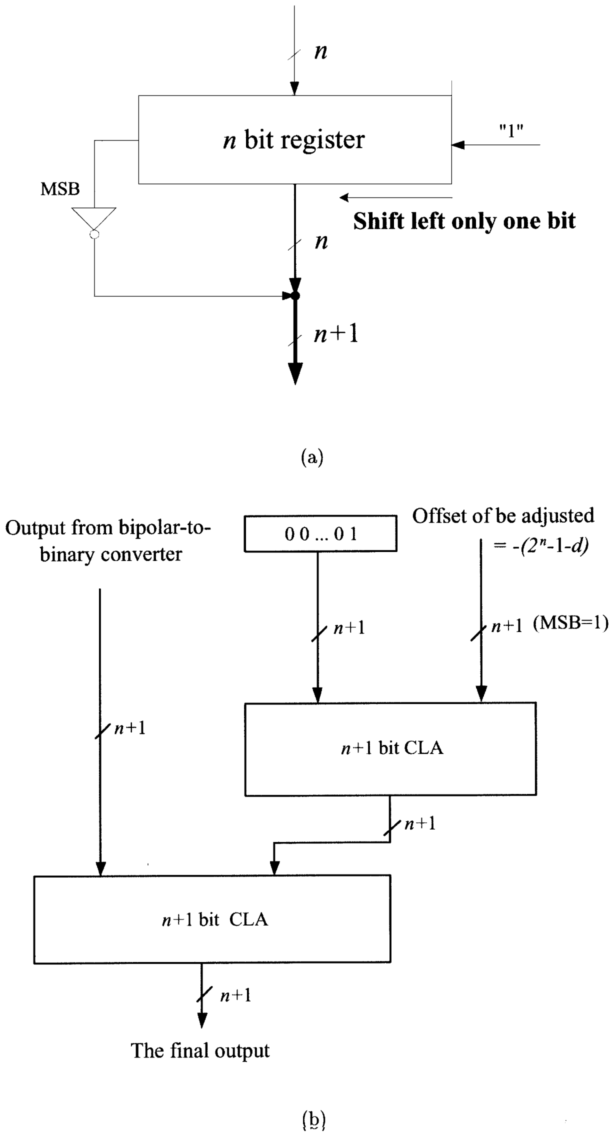
Fig. 2. A 63-to-6 compressor unit.

C. Bipolar-to-Binary Value Converter

The next processing stage is to convert the bipolar-value vector generated from the $(2^n - 1)$-to-n compressor unit into the two's complement representation of signed binary numbers. Note that a logic LOW represents -1, and a logic HIGH denotes 1 at the output of compressor unit. Besides, $(n + 1)$-bit two's complement form is required to represent the n-bit output of the compressor unit since its value falls into the range from $-(2^n - 1)$ to $(2^n - 1)$. We can find the equations for this converter as follows.

We assume the bit representation of the output of the compressor unit is $(Z_{n-1}, \ldots, Z_1, Z_0)$, $Z_i \in \{1, 0\}$. Keep in mind that the actual value of the bit representation {1,0} is, in fact, $\{+1, 1\}$. To avoid any overflow, we add an additional bit to the MSB.

$$\begin{aligned} &\text{Number of the logic HIGH} \\ &\quad = (0, Z_{n-1}, \ldots, Z_1, Z_0). \\ &\text{Number of the logic LOW} \\ &\quad = (0, \text{one's complement of the logic HIGH number}) \\ &\quad = (0, \sim (Z_{n-1}, \ldots, Z_1, Z_0)). \end{aligned}$$



(a)

(b)

Fig. 3. (a) Bipolar-to-binary conversion unit. (b) Inner product adjustment unit.

$$
\begin{aligned}
\text{Sum} &= (\text{the number of the logic HIGH} \\
&\quad - \text{the number of the logic LOW}) \\
&= (\text{the number of the logic HIGH} \\
&\quad + \text{two's complement of the logic LOW}) \\
&= (0, Z_{n-1}, \ldots, Z_1, Z_0) \\
&\quad + (\sim (0, \sim (Z_{n-1}, \ldots, Z_1, Z_0)) + 1) \\
&= (0, Z_{n-1}, \ldots, Z_1, Z_0) + (1, Z_{n-1}, \ldots, Z_1, Z_0) + 1 \\
&= (1, 0, 0, \ldots, 0) + (Z_{n-1}, \ldots, Z_1, Z_0, 0) + 1 \\
&= (\sim (Z_{n-1}), Z_{n-2}, \ldots, Z_1, Z_0, 1) \qquad (5)
\end{aligned}
$$

where $((Z_{n-1}, \ldots, Z_0) + (Z_{n-1}, \ldots, Z_0)) = ((Z_{n-1}, \ldots, Z_0)$ shift left one bit$)$.

We, thus, can implement the bipolar-to-binary value converter as shown in Fig. 3(a) according to (5).

Example 1: The output of compressor unit: $(01100)_2 = (-7)_{10}$, (It is, in fact, $(-1+1+1-1-1)_{\text{bipolar}}$) The result of the conversion: $(001\,100)_2 + (101\,100)_2 + 1 = (111\,001)_2 = (-7)_{10}$

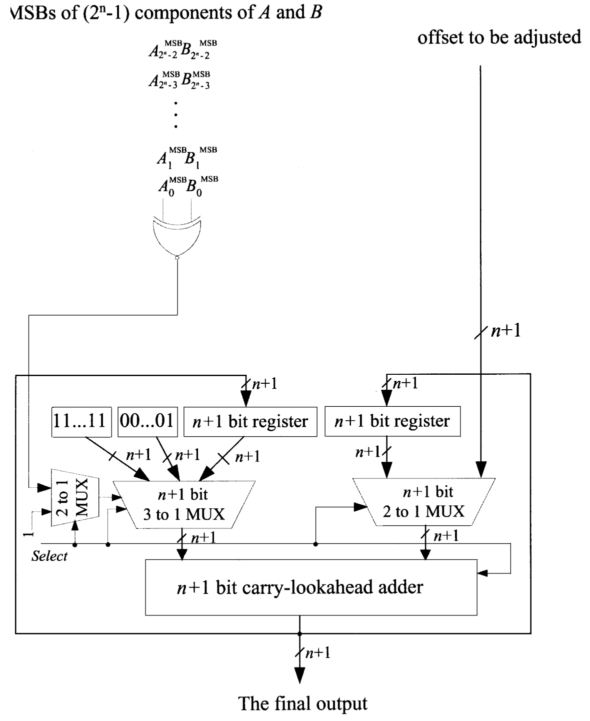


Fig. 4. Primitive architecture of the bipolar-valued inner product processor.

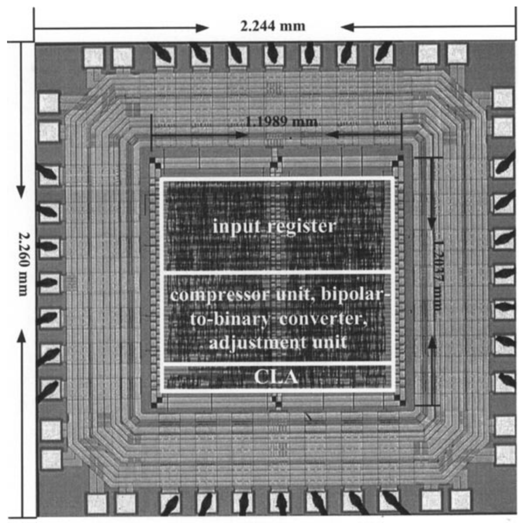

2.244 mm
1.1989 mm
2.260 mm
1.2037 mm
input register
compressor unit, bipolar-to-binary converter, adjustment unit
CLA

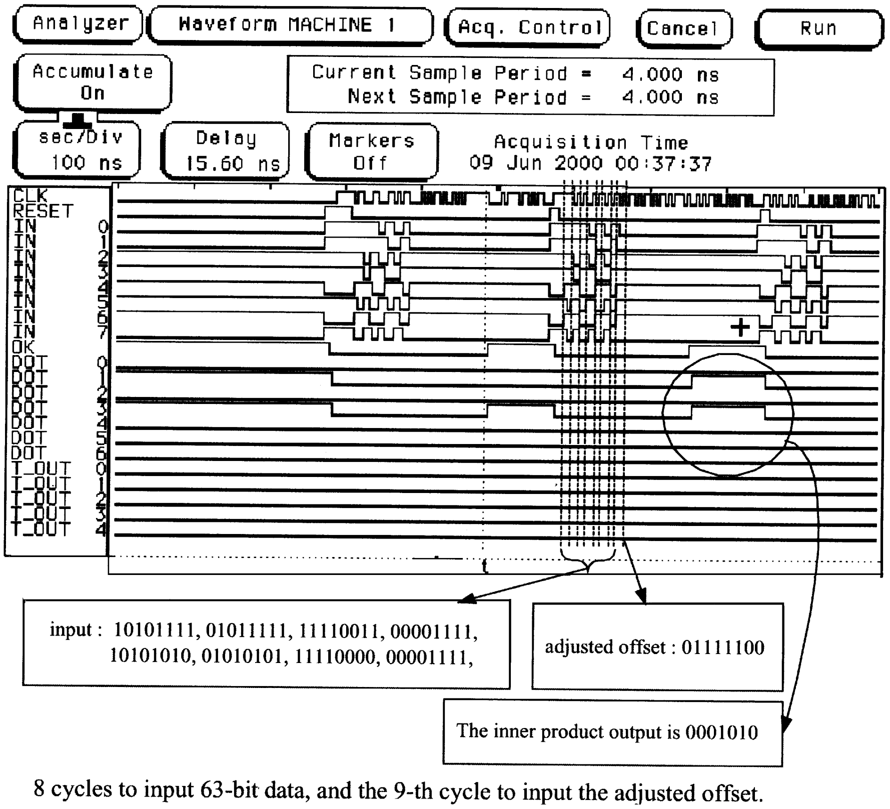
Fig. 5. Die photograph of the bipolar inner product processor.

D. Inner Product Adjustment Unit

The inner product has to be changed to the precise value. Consider the case that the dimension of bipolar vectors, d, is less than $2^n - 1$. Firstly, d, represented as n-bit values, is fed into the addend inputs of the $(n+1)$-bit carry-lookahead adder (CLA) as illustrated in Fig. 3(b), where the MSB of the $(n+1)$ bit inputs is set to 1. We have already attained the one's complement form of $-(2^n - 1 - d)$ at the addend inputs of the adder. Thus, the output of the CLA is the two's complement form of $-(2^n - 1 - d)$ since the augend inputs are set to 1.

E. Performance Analysis

A simple yet slow bipolar-valued inner product processor is presented in Fig. 4 to facilitate the overhead analysis of our design. Although the hardware complexity of the above-mentioned primitive



Accumulate On

Current Sample Period = 4.000 ns
Next Sample Period = 4.000 ns

sec/Div 100 ns

Delay 15.60 ns

Markers Off

Acquisition Time 09 Jun 2000 00:37:37

input : 10101111, 01011111, 11110011, 00001111,
10101010, 01010101, 11110000, 00001111,

adjusted offset : 01111100

The inner product output is 0001010

8 cycles to input 63-bit data, and the 9-th cycle to input the adjusted offset.

Fig. 6. Testing of the real chip (clock $=100$ MHz).

inner product processor is simpler than our proposed scheme, the total delay of the inner product calculation caused by this simple yet slow architecture turns out to be

$$\mathrm{Delay}_{\mathrm{Primitivescheme}} = (2^n-1)(d_{\mathrm{XNOR}}+d_{\mathrm{Reg}}+d_{\mathrm{MUX}}+d_{\mathrm{CLA}}) \quad (6)$$

where d_{XNOR} denotes the delay of the XNOR gate, d_{Reg} represents the delay of the register, d_{MUX} stands for the delay of the multiplexer, while d_{CLA} denotes the delay of the $(n+1)$-bit CLA.

As for the total delay of our proposed scheme, it can be expressed as follows:

$$\mathrm{Delay}_{\mathrm{Ourscheme}} = d_{\mathrm{XNOR}} + n \cdot d_{3-2\mathrm{Comp}} + d_{\mathrm{Reg}} + d_{\mathrm{CLA}} \quad (7)$$

where $d_{3-2\mathrm{Comp}}$ stands for the delay of the 3-2 compressor unit.

From (6) and (7), it is obvious that the total delay of inner product calculation is reduced significantly in our proposed scheme.

IV. Chip Implementation and Measurement

The proposed processor was approved by the Chip Implementation Center (CIC) of the National Science Council (NSC), and then fabricated by the Taiwan Semiconductor Manufacturing Company (TSMC) 0.6—μm 1P3M technology. The total area of this chip is 2260×2244 $\mu\mathrm{m}^2$, core area is 1203.7×1198.9 $\mu\mathrm{m}^2$, and the gate count is 2.3 K.

A. Physical Chip Testing

Fig. 5 is the die photo. Fig. 6 is a snapshot generated by HP 1660CP when the chip is under test. The maximum clock rate is 100 MHz which is the same as predicted by the simulation results, and the power consumption is 300 mW.

V. Conclusion

We have proposed a novel architecture of bipolar-valued inner product processor which can be employed in the implementation of associative memory networks. The systolic architecture of (2^n-1)-to-n compressor can significantly reduce the carry propagation delay in the critical path of the bipolar binary inner product, which is clearly the bottleneck of the whole computation. The physical chip implementation is also presented. The simulation results turn out to be very appealing.

References

[1] J. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, MN: West, 1992.

[2] K. A. Boahen, P. O. Pouliquen, A. G. Anderou, and R. E. Jenkins, "A heteroassociative memory using current-mode MOS analog VLSI circuits," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 747–755, May 1989.

[3] A. Johannet, L. Personnaz, G. Dreyfus, J.-D. Gascuel, and M. Weinfeld, "Specification and implementation of a digital Hopfield-type associative memory with on-chip training," *IEEE Trans. Neural Networks*, vol. 3, pp. 529–539, July 1992.

[4] N. Kazakova, R. Sung, N. Durdle, M. Margala, and J. Lamoureux, "Fast and low-power inner product processor," in *Proc. 2001 Int. Symp. Circuits and Systems*, vol. 4, 2001, pp. 646–649.

[5] T. Kohonen, "Correlation matrix memories," *IEEE Trans. Comput.*, vol. 21, pp. 353–359, Apr. 1972.

[6] J. Austin and S. O'Keefe, "Application of an associative memory to the analysis of document fax images," in *Proc. British Machine Vision Conf.*, 1994, pp. 315–325.

[7] L. Rong, "Reconfigurable parallel inner product processor architectures," *IEEE Trans. VLSI Syst.*, vol. 9, pp. 261–272, Apr. 2001.

[8] K. Nakano, "Associatron-a model of associative memory," *IEEE Trans. Systems, Man, Cybern.*, vol. SMC-2, pp. 380–388, Mar. 1972.

[9] Y. Zhang, Z. He, and C. Wei, "Stability analyzing of chaotic neural network for associative memory," in *Proc. 2002 Int. Joint Conf. Neural Networks*, vol. 1, 2002, pp. 659–664.

[10] B. Kosko, "Bidirectional associative memories," *IEEE Trans. Systems, Man, Cybern.*, vol. 18, pp. 49–60, Jan./Feb. 1988.