Design of an Inner-Product Processor for Hardware Realization of Multi-Valued Exponential Bidirectional Associative Memory

Chua-Chin Wang, Chenn-Jung Huang, and Ying-Pei Chen

Abstract—Inner-product calculations are often required in digital neural computing. The critical path of the inner product of two vectors is the carry propagation delay generated from individual product terms. In this work, a novel and high-speed realization of inner-product processor for the multi-valued exponential bidirectional associative memory (MV-eBAM) is presented in order to reduce the carry propagation delay, wherein the treatment of inner product of two vectors is given. Notably, a systolic-like architecture of digital compressors is used to reduce the carry propagation delay in the critical path of the inner product of two vectors. The architecture we propose here might offer a sub-optimal solution for the digital hardware realization of the inner-product computation.

Index Terms—Bidirectional associative memory (BAM), digital compressor, digital neural computing, exponential BAM, multi-valued.

I. INTRODUCTION

SINCE Kosko [1] proposed the *bidirectional associative memory* (BAM), many researchers have invested efforts on exploring the network's properties and limitations. Due to its intrinsic architecture, the capacity of BAM is unfortunately poor [2]. It is notable that Chiueh and Goodman [3] proposed exponential Hopfield associative memory motivated by the MOS transistor's exponential drain current dependence on the gate voltage in the subthreshold region such that the VLSI implementation of an exponential function is feasible. Although the impressive capacity of an eBAM was found [4], the data representation of BAM or eBAM is still limited to be either bipolar vectors or binary vectors. We consider that the expansion of the data range, i.e., from $\{-1, +1\}^n$ to $\{1, 2, \ldots, L\}^n, L \gg 1$ is also a feasible method to enlarge the capacity. It also enriches the data representation. This observation leads to the multi-valued exponential bidirectional associative memory (MV-eBAM) [5].

Since neural computing used in the networks similar to the MV-eBAM is composed of mass amount of inner-product calculations, the demand of shortening the delay therewith becomes urgent. Otherwise, the hardware realization of any neural network becomes impractical. The inner-product
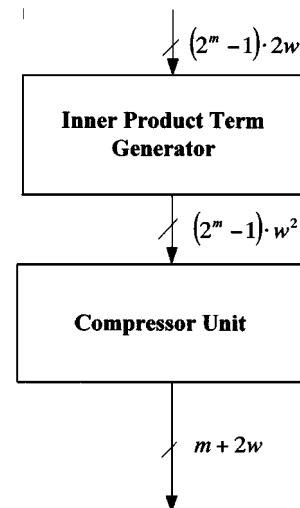
Fig. 1. Data flow of an inner-product calculation for the MV-eBAM.

computation of two multi-valued vectors can be done by a process of successive multiply and accumulate operations conventionally [6], [7]. This method generates the individual inner-product term by using a multiplier and employs a single adder to sum up all the inner-product terms iteratively. The systolic-like architecture of partial product reduction tree for the parallel multiplier introduced by Wallace [8] motivated the implementation of several parallel schemes for the inner-production calculation [6], [9]. Researchers have also proposed a variety of compressors to speed up the process of partial product reduction in multiplication or inner-product operation, such as 4-2, 5-5-4, or 9-2 compressors, etc. [10], [11]. However, Oklobdzija *et al.* [12] pointed out that it is the interconnection of the compressors, rather than the structure of the compressors, that leads to the fastest realization of partial product reduction in multiplication operation. The superiority of the 4-2 compressors and 7-3 compressors built by Zhang *et al.* [13] verified that the conclusion of Oklobdzija *et al.* is correct. Besides, Wang *et al.* [14] compared different compressor architectures and concluded that the systolic-like architecture outperforms the others. Consequently, the novel inner-product processor dedicated to the MV-eBAM presented in this paper will include a systolic-like architecture of compressor unit wherein the arrangement of the 3-2 compressors is tuned in order such that the carry propagation delay in the critical path is
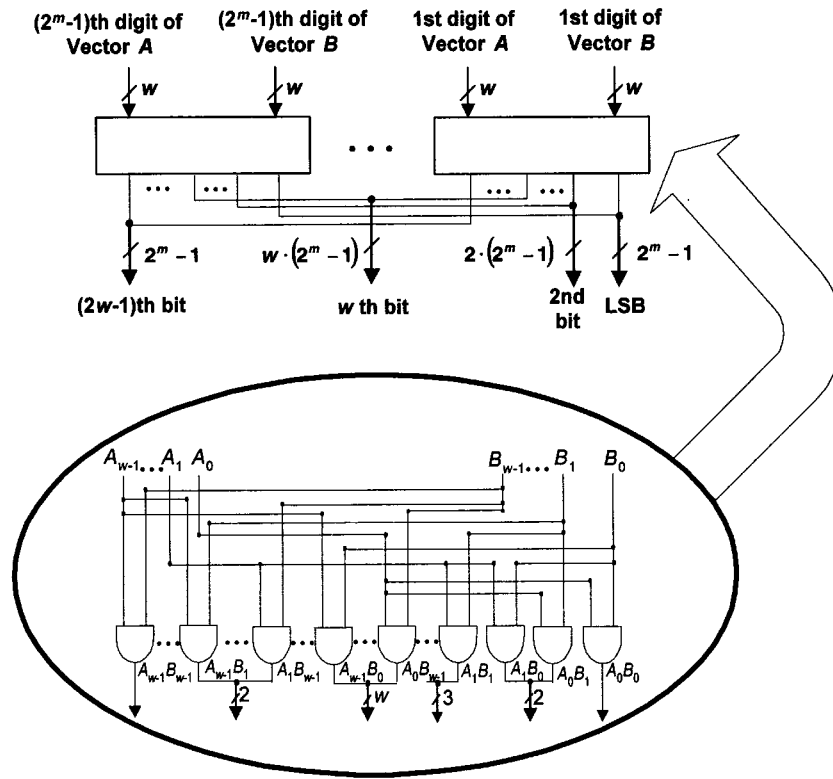
Fig. 2.   Inner-product term generator.

reduced. In addition to the compressor unit, an inner-product term generator is also proposed to produce the individual inner-product terms as the inputs to the compressor unit.

II. THEORY OF MV-eBAM

Before the introduction of the inner-product processor for the MV-eBAM, it is necessary to show how the MV-eBAM operates theoretically. Suppose we are given M pattern pairs, which are

$$\{(X_1, Y_1), (X_2, Y_2), \ldots (X_M, Y_M)\} \qquad (1)$$

where

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{in}), \quad Y_i = (y_{i1}, y_{i2}, \ldots, y_{ip})$$

where n is assumed to be smaller than or equal to p without any loss of generality. Hence, the evolution equations of the MV-eBAM are shown as

$$y_k = H\left(\frac{\sum_{i=1}^{M} y_{ik} b^{-\|X - X_i\|^2}}{\sum_{i=1}^{M} b^{-\|X - X_i\|^2}}\right)$$

$$x_k = H\left(\frac{\sum_{i=1}^{M} x_{ik} b^{-\|Y - Y_i\|^2}}{\sum_{i=1}^{M} b^{-\|Y - Y_i\|^2}}\right) \qquad (2)$$

where

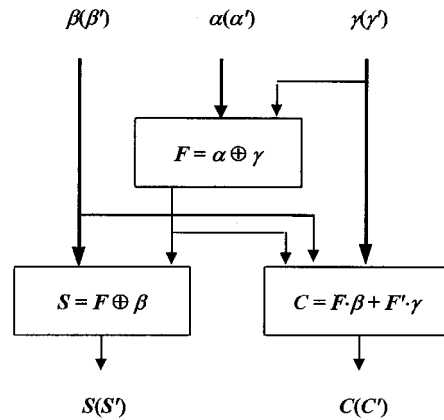|	
X and Y	key patterns;
b	a positive number, called the radix $b > 1$;
x_k, x_{ik}	kth digits of X and X_i with y_k and y_{ik} for Y and Y_i, respectively;
$H(\cdot)$	a staircase function shown as

Fig. 3.   A 3-2 compressor building block.

$$H(x) = \begin{cases} l, & (l - 0.5) \cdot \frac{D}{L} \le x < (l + 0.5) \cdot \frac{D}{L} \\ 1, & x < 1.5 \cdot \frac{D}{L} \\ L, & x \ge D \end{cases} \qquad (3)$$

where $l = 1, 2, \ldots, L, L$ is the number of finite levels, and D is the finite interval of the staircase function. Note that if $D \to \infty$ and $L \to \infty$, then $H(x) \approx x$, for $x > 0$. The reason why the staircase function is used is the argument in $H(\cdot)$ in (3) is not necessarily a positive integer. We, hence, have to assign this argument to a nearest integer.

The reasons for using an exponential scheme in (2) are to enlarge the attraction radius of every stored pattern pair and to augment the desired pattern in the recall reverberation process. In the evolution equations (2), if the given input pattern is close
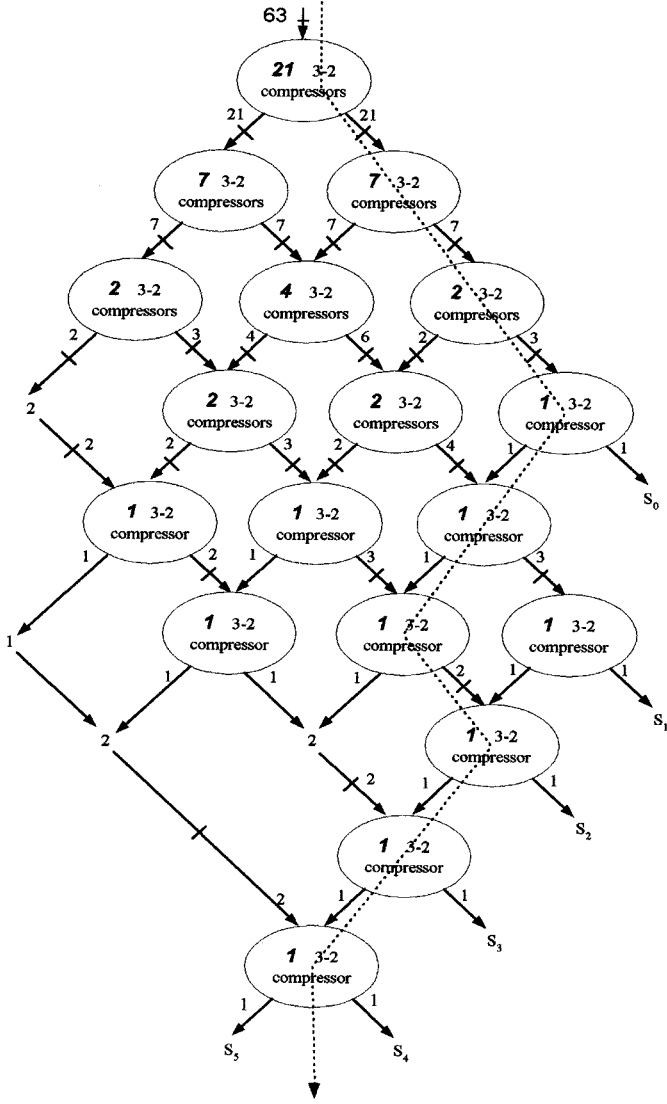
Fig. 4. Systolic-like architecture of $(2^q - 1)$-to-q compressor for $q = 6$.

Fig. 5. An alternative architecture of $(2^q - 1)$-to-q compressor.

to the desired pattern, the weighting coefficient $b^{-\|X - X_i\|^2}$ will be close to the maximum, 1, while if the input pattern is far from the desired one, it will approach zero. As for the purpose of the denominator, it makes the y_k and x_k to be the centroids of all of the y_{ik}'s and x_{ik}'s, respectively.

The capacity of the MV-eBAM can be shown to be very close to the maximum number of combinations of the input vector, i.e., $M_{\max} \approx L^n$ when b is large enough (Wang and Hwang, 1996). Hence, the MV-eBAM indeed possesses a high capacity.

However, one serious problem occurs when it comes to the physical implementation of such a high-capacity associative memory by digital VLSI circuits. In the computation of the MV-eBAM, the inner product of two vectors $\|X - X_i\|^2$ or $\|Y - Y_i\|^2$ might be one of the most frequently used mathematical operations. Notably, if n or p is large in the above calculation, then the carry propagation of the inner product of the vectors will likely become the critical delay of the entire neural computing. This side effect undoubtedly devalues the hardware realization of the MV-eBAM.
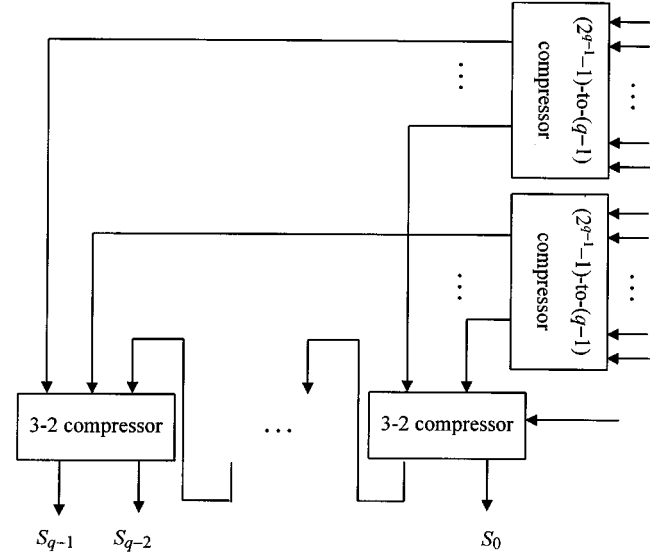
III. HIGH-SPEED INNER PRODUCT PROCESSOR FOR THE MV-eBAM

In order to reduce the carry propagation delay produced in the implementation of the MV-eBAM, it is demanding to develop a special-purpose processor for the inner product of two multi-valued operands. The entire design of multi-valued inner-product processor is divided into two parts, which are an individual inner-product term generator, and a compressor unit. The inner-product term generator produces the individual inner-product terms given two multi-valued vectors, and passes them to the compressor unit, in which a summation of product terms is computed. Fig. 1 shows the data flow of a multi-valued inner-product calculation.

A. Inner-Product Term Generator

Considering the compatibility with the binary digital system, the number of finite levels L in (3) is set to $2^w - 1$ in the implementation of the inner-product processor for the MV-eBAM. Besides, the computation of each individual inner-product term in $\|X - X_i\|^2$ or $\|Y - Y_i\|^2$ turns out to be an unsigned integer operation because each term is always positive. Thus, each product term in (2), *product*, can be evaluated by

$$\begin{aligned} \text{product} = \vec{A} \cdot \vec{B} &= \left(\sum_{i=0}^{w-1} A_i \cdot 2^i \right) \cdot \left(\sum_{i=0}^{w-1} B_i \cdot 2^i \right) \\ &= A_{w-1} B_{w-1} \cdot 2^{2w-2} \\ &\quad + (A_{w-2} B_{w-1} + A_{w-1} B_{w-2}) \cdot 2^{2w-3} + \cdots \\ &\quad + (A_0 B_2 + A_1 B_1 + A_2 B_0) \cdot 2^2 \\ &\quad + (A_0 B_1 + A_1 B_0) \cdot 2^1 + A_0 B_0 \cdot 2^0 \end{aligned} \tag{4}$$

where A_i, B_i is 0 or 1.

Notably, the design of the inner-production term generation becomes simple because only w^2 AND gates are required to produce the w^2 partial products in (4). Fig. 2 shows the configuration of the inner-product term generation unit. Note that the dimension of the stored patterns is set to the count of the inputs
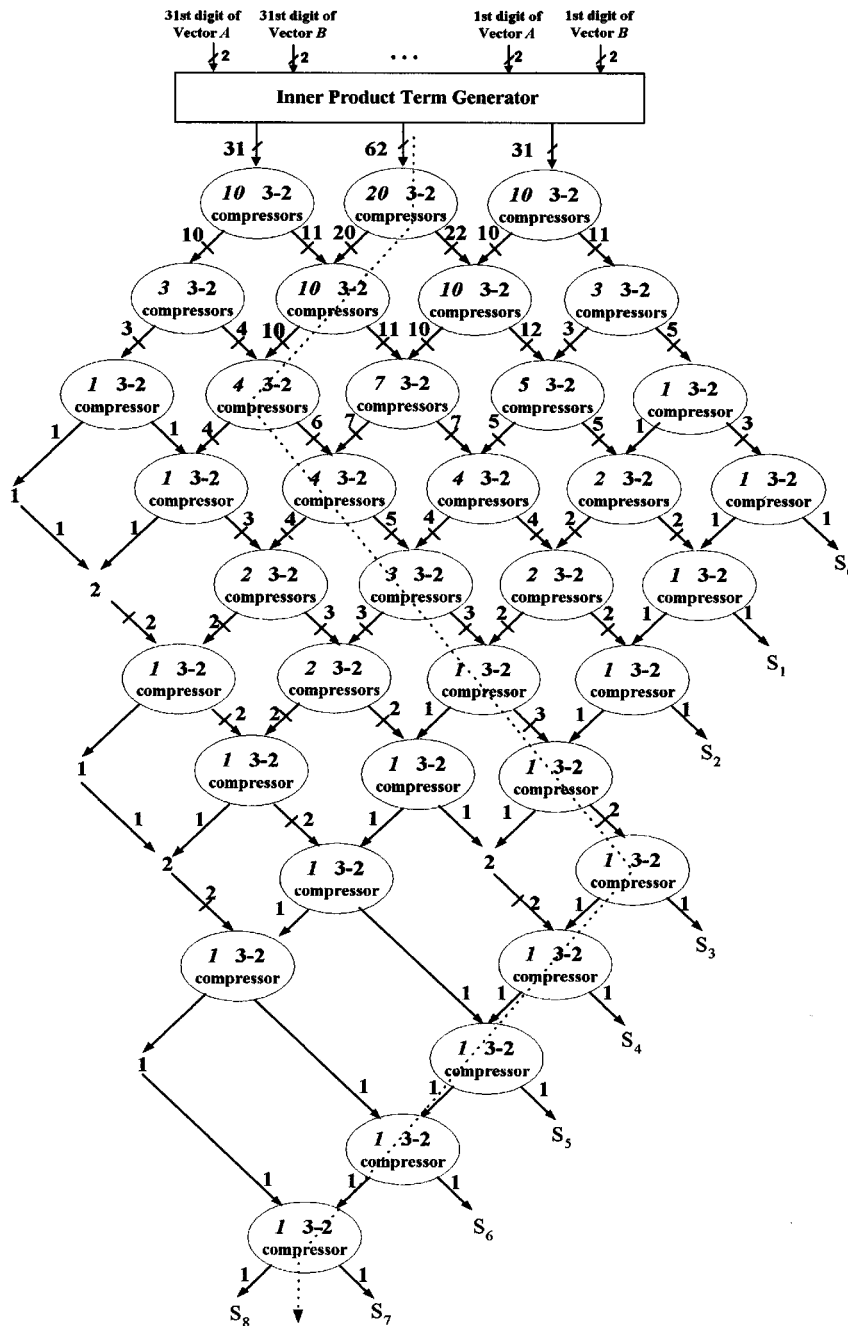
Fig. 6. Systolic-like architecture of a MV-eBAM compressor for $m = 5$ and $w = 2$.

to a $(2^m - 1)$-to-m compressor, which will be introduced in the next section. Therefore, the length of the inputs to the inner-production term generator is $(2^m - 1) \cdot 2w$, and the length of the outputs is $(2^m - 1) \cdot w^2$ according to (4). In case that the dimension of the stored patterns is less than $2^m - 1$, all the unused inputs to the inner-production term generator are padded with zeros.

B. Framework of the Compressor Unit

1) Systolic-Like $(2^q - 1)$-to-q 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 1s given in inputs. The equations of a full adder are shown as follows:

$$S = (\alpha \oplus \gamma) \cdot \beta' + (\alpha \oplus \gamma)' \cdot \beta = F \cdot \beta' + F' \cdot \beta$$
$$C = (\alpha \oplus \gamma) \cdot \beta + (\alpha \oplus \gamma)' \cdot \gamma = F \cdot \beta + F' \cdot \gamma \quad (5)$$

where F denotes $(\alpha \oplus \gamma)$.

As shown in Fig. 3, the logic structure of a typical 3-2 compressor can be split up into two logic layers. One of the three inputs, $\beta(\beta')$, is not required in the first logic layer.

A $(2^q - 1)$-to-q compressor building block can be constructed by cascading 3-2 compressors, as shown in Fig. 4. This architecture, inspired by the design methodology of systolic arrays, consists of parallelized 3-2 compressor building blocks only at
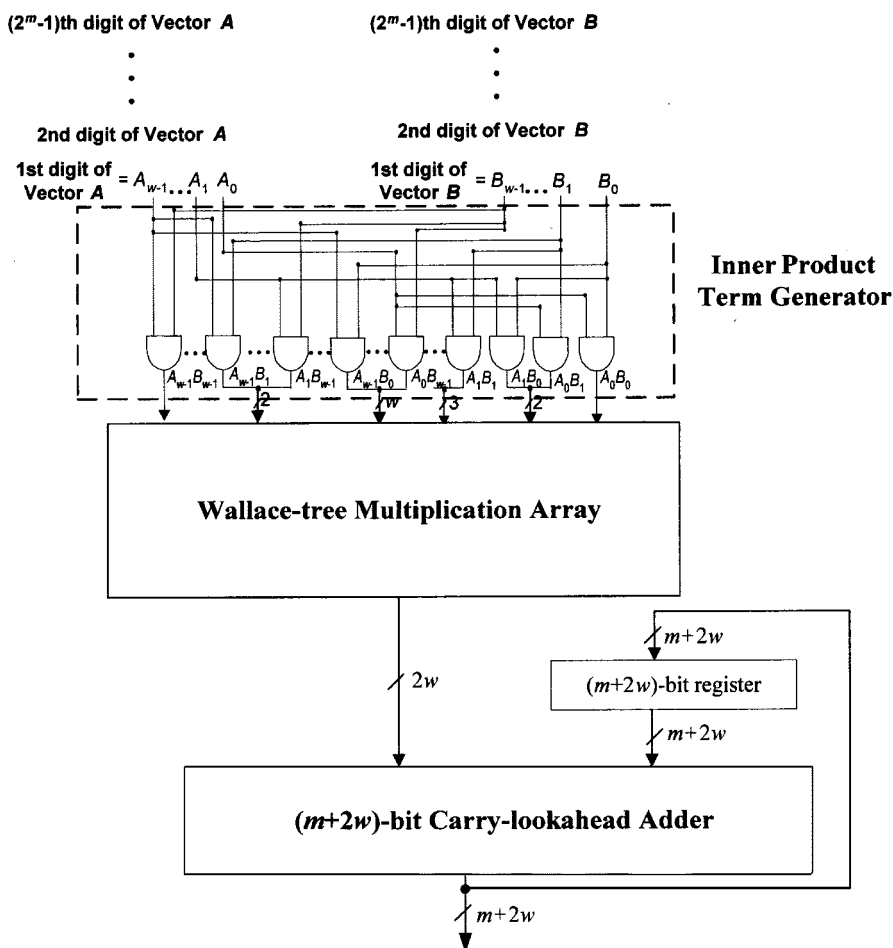
Fig. 7.   Conventional multi-valued inner-product processor.

every processing stage. Note that the number beside the arrow pointing toward each circle represents the count of the inputs to the 3-2 compressors at each processing stage, while the number beside the two outward arrows indicates the count of the outputs of each 3-2 compressor building block. The number inside the circles denotes the count of the 3-2 compressors which process the inputs at some specific bit positions.

To compute the total count of 3-2 compressors used in a $(2^q - 1)$-to-q compressor, we consider an alternative architecture of the (2^q-1)-to-q compressor, which is composed of two $(2^{q-1} - 1)$-to-$(q-1)$ compressors and $(q-1)$ 3-2 compressors, as shown in Fig. 5. Based on the configuration of this compressor, we can derive the count of the 3-2 compressors used in this architecture as follows:

$$N_2 = 1$$
$$N_3 = 4$$
$$N_q = 2 \cdot N_{q-1} + q - 1, \quad q > 2 \tag{6}$$

where N_q denotes the number of the 3-2 compressors used in a $(2^q - 1)$-to-q compressor.

By solving the above recurrence relation, we obtain

$$N_q = 2^q - q - 1. \tag{7}$$

The number of 3-2 compressors used in these two architectures we present above is identical because no unused inputs to the 3-2 compressors appear in both $(2^q - 1)$-to-q compressor structures. Thus, we can conclude that the count of 3-2 compressors used in the systolic-like architecture of the (2^q-1)-to-q compressor is also $2^q - q - 1$.

2) Framework of Digital Compressor Design: According to (4), the summation of the partial product terms is not computed in the inner-product term generator. This implies that the outputs of the w^2 AND gates are fed into the compressor unit at the required bit positions. Besides, $2^m - 1$ individual inner-product terms need to be accumulated at each bit position. Thus, there will be $2^m - 1$ partial product terms at LSB, $2 \cdot (2^m - 1)$ partial product terms at the second bit position, $w \cdot (2^m - 1)$ partial product terms at the wth bit position, and $2^m - 1$ partial product terms at the $(2w - 1)$th bit position (MSB), and so forth, as shown in Fig. 2. Since many accumulation operations must be performed to obtain the final result, the improvement of the carry propagation delay of the critical paths is the major consideration for the architecture of the compressor unit. The entire architecture we propose to achieve this goal is shown in Fig. 6. Since this compressor unit is composed of one or several $(2^m - 1)$-to-m compressors at each bit position, we tend to set the dimension of the stored patterns to $2^m - 1$ to reduce the number of the unused inputs to the basic 3-2 compressor building blocks.

in0: 31 inputs to the compressor unit at LSB position.
in1: 62 inputs to the compressor unit at 2nd bit position.
in2: 31 inputs to the compressor unit at 3rd bit position.
out: output generated by the compressor unit.
test: expected output.
c: measured delay in current test.
d: maximum delay measured so far.
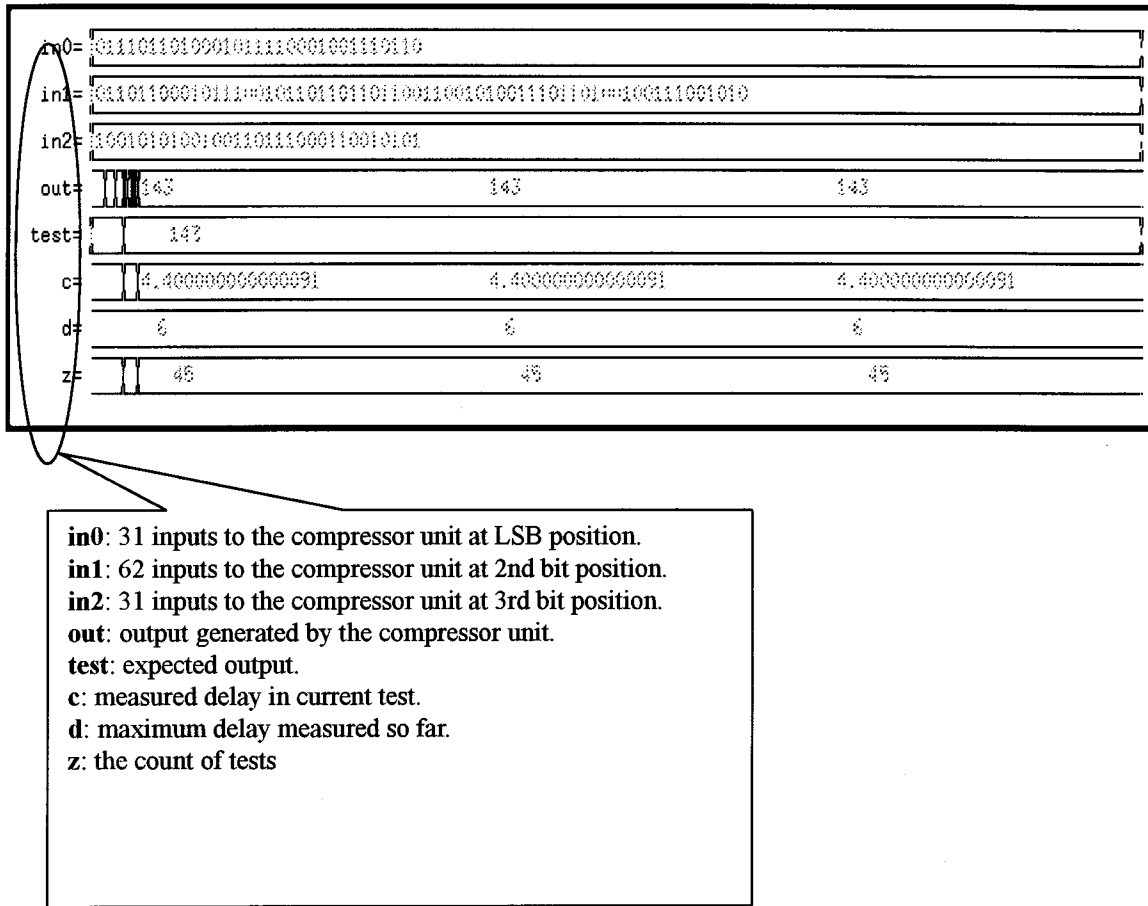**z**: the count of tests

Fig. 8. A waveform diagram sample of the inner-product calculation.

Although it is difficult to derive a general form of the critical delay for the compressor unit due to its irregular structure, the estimated delay can be derived by attaching fictitious 3-2 compressors of $(2w - 2)$ stages on the top of the compressor unit to form a single $(2^q - 1)$-to-q compressor tree. Since there are $(2^m - 1) \cdot w^2$ inputs to the compressor unit, the length of the inputs to the made-up $(2^q - 1)$-to-q compressor tree now becomes $(2^m - 1) \cdot w^2 \cdot (3/2)^{2w-2}$ after tracing back to the top level of the $(2^q - 1)$-to-q compressor. We assume $D_{m,w}$ denotes the count of 3-2 compressors in the critical path of the compressor unit, then the delay can be derived as follows:

$$
D_{m,w} \approx \left\lceil \frac{\log \frac{(2^m-1) \cdot w^2 \cdot \left(\frac{3}{2}\right)^{2w-2}}{2}}{\log \frac{3}{2}} \right\rceil - (2w - 2)
$$

$$
< \left\lceil (m-1) \cdot \frac{\log 2}{\log \frac{3}{2}} + \frac{2 \log w}{\log \frac{3}{2}} + (2w-2) \right\rceil - (2w-2)
$$

$$
< 2m + 11.36 \log w - 1.71. \tag{8}
$$

The number of 3-2 compressors used in the compressor unit can be estimated based on (7). Let N_q denotes the number of the 3-2 compressors used in the made-up $(2^q - 1)$-to-q compressor tree. First, we get

$$
2^q - 1 = (2^m - 1) \cdot w^2 \cdot \left(\frac{3}{2}\right)^{2w-2} \tag{9}
$$

$$
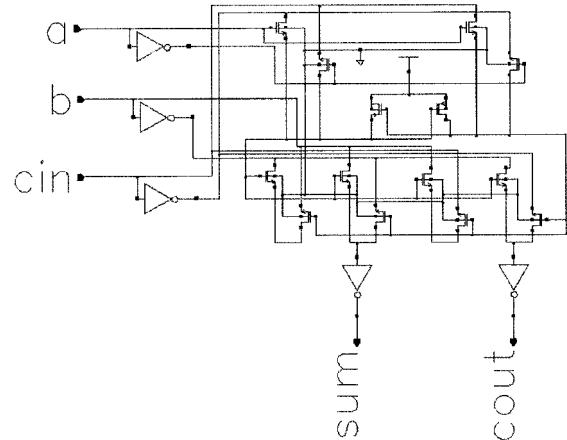q \approx m + 2 \log_2 w + 1.17w - 1.17 \tag{10}
$$

Fig. 9. Schematic diagram of the 3-2 compressor building block.

$$
\zeta \approx (2^m - 1) \cdot w^2 \cdot \left(\frac{3}{2}\right)^{2w-2} \cdot (1 - (2/3)^{2w-2})
$$

$$
= (2^m - 1) \cdot w^2 \cdot \left(\left(\frac{3}{2}\right)^{2w-2} - 1\right) \tag{11}
$$

where ζ denotes the count of the 3-2 compressors at the fictitious $(2w - 2)$ levels. Then, N_q can be computed as follows:

$$
N_q \approx 2^q - q - 1 - \zeta
$$

$$
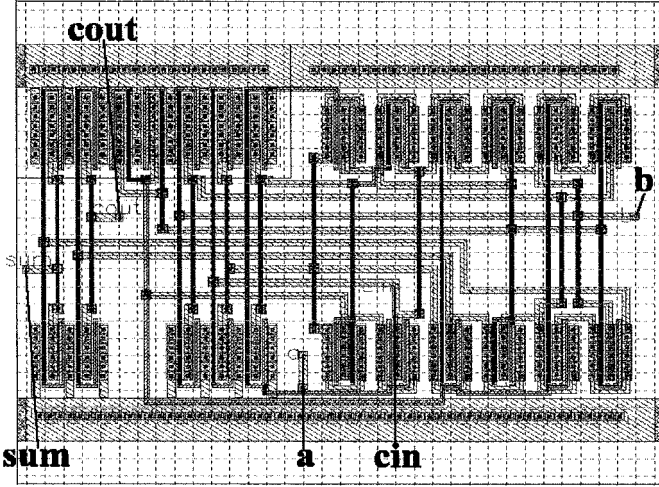= (2^m - 1) \cdot w^2 - m - 2 \log_2 w - 1.17w + 1.17. \tag{12}
$$

Fig. 10.   Circuit layout of the 3-2 compressor building block.

IV. SIMULATION AND ANALYSIS

A. Performance Analysis

A conventional multi-valued inner-product processor is presented in Fig. 7 to facilitate the overhead analysis of our design. As Fig. 7 shows, the $2^m - 1$ components of the two input vectors are fed into the individual inner-product term generator serially, where the number of finite levels, L, in (3) is set to $2^w - 1$. During each cycle, the w^2 outputs of the individual inner-product term generator is streamed into a Wallace-tree multiplication array to obtain the $2w$-bit product. Notably, a fast adder such as carry-lookahead adder (CLA) is required at the final stage of the multiplication. Then the product is fed into another CLA to get the accumulated partial sum of the inner product. Since the maximum value of the inner product, $P_{\max}$, can be derived by

$$
\begin{aligned}
P_{\max} &= (2^m - 1) \cdot (2^w - 1) \cdot (2^w - 1) \cdot \\
&= 2^{m+2w} - 2^{2w} - 2^{m+w+1} + 2^{w+1} \\
&\quad + 2^m - 1 > 2^{m+2w+1}
\end{aligned}
\tag{13}
$$

for $m > 1$ and $w > 2$, the output bit length of the CLA is required to be at least $m + 2w$, which is also the output bit length of the compressor unit as given in Fig. 1.

Similar to the approach taken for the derivation of (8) and (12), the propagation delay of the Wallace-tree multiplication array counted by the number of 3-2 compressors can be estimated as follows:

$$
\begin{aligned}
D_{\text{Wallace tree}} &\approx \frac{\log \frac{w^2 \cdot \left(\frac{3}{2}\right)^{2w-2}}{2}}{\log \frac{3}{2}} - (2w - 2) \\
&< 11.36 \log w - 1.71
\end{aligned}
\tag{14}
$$

and the approximate number of 3-2 compressors used in the multiplication array becomes

$$
N_{\text{Wallace tree}} \approx w^2 - 2\log_2 w - 1.17w + 1.17.
\tag{15}
$$

Next, we need to evaluate the critical delay and the hardware complexity of the two CLAs. Based on the tree-like architecture of the CLA proposed by Dozza *et al.* [15], it can be shown that
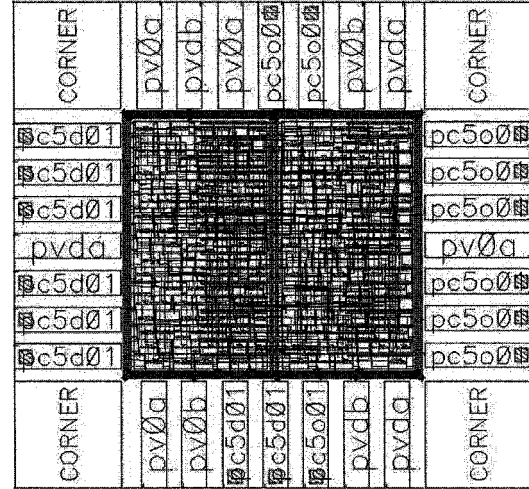
Fig. 11.   Circuit layout of the MV-eBAM inner-product processor for $m = 5$ and $w = 2$.

the delay of an r-bit CLA counted by the number of 2-input logic gates is

$$
D_{r\text{-bit CLA}} = \log_2 r + 3.
\tag{16}
$$

Meanwhile, the number of 2-input logic gates used in this tree-like CLA can be shown as

$$
N_{r\text{-bit CLA}} = 3r \cdot \log_2 r - 3.
\tag{17}
$$

In summary, the conventional scheme requires an extra $2w$-bit CLA, an $(m + 2w)$-bit CLA, and an $(m + 2w)$-bit register. However, the compressor unit as shown in Fig. 6 is replaced by the simpler Wallace tree, and only one set of individual inner-product term generator is needed in this scheme. The extra hardware cost for our proposed scheme can be estimated as follows:

1) $w^2 \cdots (2^m - 1)$ AND gates for the inner-product term generator;
2) $(2^m - 2) \cdot w^2 - m$ 3-2 compressors used in the compressor unit.

Although the hardware complexity of the above-mentioned conventional inner-product processor is simpler than our scheme, the total delay of the inner-product calculation caused by this simple yet slow architecture turns out to be

$$
\begin{aligned}
\text{Delay}_{\text{Conventional scheme}} \\
= (2^m - 1) \cdot \Big(D_{\text{AND}} + D_{\text{Wallace tree}} \\
+ D_{2w\text{-bit CLA}} + D_{(m+2w)\text{-bit CLA}} \Big)
\end{aligned}
\tag{18}
$$

where

D_{AND}	delay of the AND gate;
$D_{\text{Wallace tree}}$	delay of the Wallace tree multiplication array;
$D_{2w\text{-bit CLA}}$	delay of the CLA;
$D_{(m+2w)\text{-bit CLA}}$	delay of the CLA.

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

$$
\text{Delay}_{\text{Our scheme}} = D_{\text{AND}} + D_{\text{Compressor unit}}
\tag{19}
$$

where $D_{\text{Compressor unit}}$ stands for the delay of the compressor unit.

From (8), (14), (16), (18), and (19), it can be shown that the total delay of multi-valued inner-product calculation counted by the number of 2-input logic gates is reduced by

$$\text{diff}_{\text{Delay}} \approx (2^m - 2) \cdot (22.72 \log w - 2.42) + (2^m - 1)$$
$$\cdot (\log_2(m + 2w) + \log_2 w + 7) - 4m. \quad (20)$$

As seen from (20), the delay of inner product is improved significantly in our proposed scheme.

B. Verilog Simulations

In order to verify the correctness and the performance of the implementation of the inner-product processor for the MV-eBAM, Verilog HDL is used to conduct a series of simulations with over 20 000 random testing vectors to explore the critical delays of the proposed architecture. The dimension of the input vectors to the inner-product processor is 31, and each digit is 2 bits wide. The simulation results indicate a delay of about 6.4 ns for the critical paths. Fig. 8 shows a waveform diagram sample of inner-product computation of two testing vectors. Notice that the testing vectors given in Fig. 8 are the inputs to the compressor unit.

C. Chip Implementation

We use the Taiwan Semiconductor Manufacturing Company (TSMC) 0.6-μm 1P3M technology to design the chip. The 3-2 compressor building block is designed as shown in Figs. 9 and 10. Then we use Cadence Silicon Ensemble automatic place and route tools to generate the abstract view and the layout of the chip. At last, the DRACULA and TimeMill are utilized to execute the full-chip-scale post-layout simulation. The circuit layout of the inner-product processor is given in Fig. 11.

V. CONCLUSION

In this paper, we have proposed a novel architecture of the inner-product processor which can be employed in the implementation of multi-valued exponential bidirectional associative memory. The inner-product processor consists of two major components: the individual product term generators and compressor units. The design of the individual product term generator is simplified because only the partial product terms are generated at this stage. The summation of the partial terms and the individual inner-product terms are accumulated at the next stage, i.e., the compressor unit. The systolic-like architecture of the compressor units can significantly reduce the carry propagation delay in the critical path of the inner product, which is clearly the bottleneck of the whole computation.

REFERENCES

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

[2] K. Haines and R. Hecht-Nielsen, "A BAM with increased information storage capacity," in *Proc. IJCNN*, vol. I, 1988, pp. 181–190.

[3] T. D. Chiueh and R. M. Goodman, "High-capacity exponential associative memory," in *Proc. IJCNN*, vol. I, 1988, pp. 153–160.

[4] C.-C. Wang and H.-S. Don, "An analysis of high-capacity discrete exponential BAM," *IEEE Trans. Neural Networks*, vol. 6, pp. 492–496, Mar. 1995.

[5] C.-C. Wang, S.-M. Hwang, and J.-P. Lee, "Capacity analysis of the asymptotically stable multi-valued exponential bidirectional associative memory," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 733–743, Oct. 1996.

[6] S. Y. Kung, *VLSI Array Processor*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[7] L. Breveglieri and L. Dadda, "A VLSI inner product macrocell," *IEEE Trans. VLSI Syst.*, vol. 6, pp. 292–298, June 1998.

[8] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Comput.*, vol. 13, no. 2, pp. 14–17, Feb. 1964.

[9] D. J. Soudris, V. Paliouras, T. Stouraitis, and C. E. Goutis, "A VLSI design methodology for RNS full adder-based inner product architecture," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 315–318, Apr. 1997.

[10] P. J. Song and G. De Micheli, "Circuit and architecture trade-offs for high-speed multiplication," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1184–1198, Sept. 1991.

[11] W. J. Stenzel, "A compact high speed parallel multiplication scheme," *IEEE Trans. Comput.*, vol. 26, pp. 948–957, Feb. 1977.

[12] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Comput.*, vol. 45, pp. 294–305, Mar. 1996.

[13] D. Zhang and M. I. Elmasry, "VLSI compressor design with applications to digital neural networks," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 230–233, June 1997.

[14] C.-C. Wang, C.-J. Huang, and P.-M. Lee, "A comparison of two alternative architectures of digital ratioed compressor design for inner product processing," *Proc. IEEE Int. Symp. Circuits Syst.*, vol. I, pp. 161–164, June 1999.

[15] D. Dozza, M. Gaddoni, and G. Baccarani, "A 3.5 ns, 64 bit, carry-lookahead adder," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. II, June 1996, pp. 297–300.

Chua-Chin Wang was born in Taiwan in 1962. He received the B.S. degree from National Taiwan University, Taiwan, R.O.C., in 1984, and the M.S. and Ph.D. degrees from the State University of New York at Stony Brook in 1988 and 1992, all in electrical engineering.

Currently, he is a Professor in the Department of Electrical Engineering, National Sun Yat-Sen University, Taiwan, R.O.C. His research interests include low-power logic and circuit design, VLSI design, and neural networks and implementations.

C.-J. Huang was born in Hualien, Taiwan, R.O.C., in 1961. He received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C., in 1984, the M.S. degree in computer science from the University of Southern California at Los Angeles in 1987, and the Ph.D. degree in electrical engineering from National Sun Yat-Sen University, Taiwan, in 2000.

He is currently an Assistant Professor in the Department of Information Management, Fortune Institute of Technology, Taiwan. His research interest include computer arithmetic, computer communication networks, and neural networks.

Ying-Pei Chen was born in Taiwan, R.O.C., in 1974. She received the B.S. degree in C.S.I.E. from TKU, Taipei, Taiwan, R.O.C., in 1998. She is currently working toward the M.S. degree in electrical engineering at National Sun Yat-Sen University, Taiwan, R.O.C.

Her research interests are in the area of networking.