Analysis of radix searching of exponential bidirectional associative memory

C.-C. Wang
C.-F. Tsai
J.-P. Lee

Indexing terms: Bidirectional associative memory (BAM), Signal-to-noise ratio (SNR), Exponential BAM, Searching algorithms, Stability

Abstract: The exponential bidirectional associative memory (eBAM) is a high-capacity associative memory. However, in the hardware realisation of eBAM, increasing efforts have been made to obtain an optimally small radix of exponential circuit for the fixed dynamic range of the VLSI circuit transistor, thereby allowing the dimension of the stored patterns to reach maximum. In this paper, the authors prove the stability of eBAM. The absolute lower bound of the radix of the eBAM is also obtained. In addition, an algorithm is presented to compute the optimal radix of an exponential circuit. To preserve the optimality of the radix, an algorithm capable of updating the radix when new pattern pairs are to be installed is proposed. Moreover, a deterministic method is presented to train and install pattern pairs with a predetermined fault tolerance ability.

1 Introduction

After Kosko [1, 2] proposed the bidirectional associative memory (BAM), many investigators attempted to enhance its intrinsic poor capacity and implement the BAM with hardware circuits. Among those efforts, Wang *et al.* [3] presented two alternatives, multiple training and dummy augmentation, to enhance BAM's ability to find the global minimum. Simpson proposed an intraconnected BAM and a high-order autocorrelator [4], and Tai *et al.* [5] presented a high-order BAM. In addition, Wang *et al.* [6] developed a weighted learning algorithm for BAM. Our previous work [7] pointed out that, despite the merits of the above efforts, they increase the complexity of the network and only slightly enhance the capacity. Chiueh and Goodman [8] presented an 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. In that same work, Chiueh also proposed an exponential correlation associative memory (ECAM), which is an autocorrelator utilising the above-mentioned exponential function of VLSI circuits to enlarge the correlation between stored pattern pairs. According to Chiueh's exponential Hopfield associative memory, Jeng *et al.* proposed one kind of exponential BAM [9]. However, the energy function proposed in [9] cannot guarantee that every stored pattern pair will have a local minimum on the energy surface. Moreover, that investigation did not perform capacity analysis. Regarding the stability of the exponential BAM, several researchers have employed different approaches to verify the systematic stability, among which include Jeng [9] and Zhang [10]. However, using a single energy function cannot completely verify its stability.

Although our previous work has estimated the impressive capacity of an eBAM [7], exploring the hardware realisation of such a neural network is a worthwhile task. Chiueh [8], Glasser [11], and Mead [12] confirmed that the dynamic range of the VLSI exponential circuits operating in the subthreshold region is approximately fixed, indicating that this property leads to an interesting limitation. It is that the minimal radix of the exponential circuit must be estimated to obtain the maximum dimension of stored patterns. Thus, finding an optimally small radix to store a group of given pattern pairs is a critical task. Consequently, updating the radix without loss of the optimality when more pairs are to be installed is another problem to be resolved. Moreover, this work also presents a deterministic approach to encode pattern pairs in an eBAM such that every encoded pair must be recollected in a predetermined fault tolerance range.

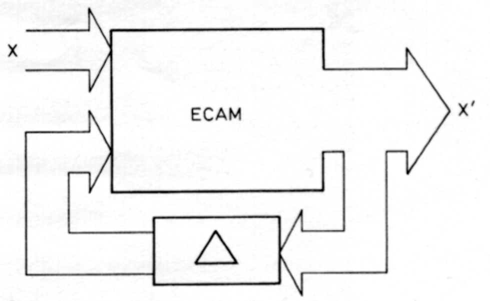

Fig. 1 *ECAM configuration*

2 Radix searching of eBAM

Although Chiueh and Goodman [13] proposed an

IEE Proceedings online no. 19982017

Paper received 25th September 1997

The authors are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan 80424

exponential Hopfield associative memory which also utilised an exponential scheme, they did not demonstrate the stability of the ECAM, whose configuration is illustrated in Fig. 1. In addition, although Jeng *et al.* [9] and Zhang *et al.* [10] have also, respectively, proved the stability of the exponential BAM structure, their proofs are not completely correct. For instance, eqn. 5 of Jeng's work cannot ensure $\Delta E_x \leq 0$, because $x_k^i \cdot (x'_k - x_k)$ is not necessarily non-negative. The proof of Theorem 4 in Zhang's work is erroneous. (Note that in Zhang's work, it is called an MBAM, Modified BAM.) According to Zhang's eqn. 18, (i.e. the definition of energy function, the difference of the energy between the current state and the next state in the first direction) should be $E(X', Y) - E(X, Y)$. Thus,

$$\begin{aligned}\Delta E_X &= E(X',Y) - E(X,Y) \\ &= -\sum_{i=1}^{M} \exp\left(\gamma Y^{(i)} Y^T\right) X^{(i)} X'^T \\ &\quad - \sum_{i=1}^{M} \exp\left(\gamma X^{(i)} X'^T\right) Y^{(i)} Y^T \\ &\quad - \left(-\sum_{i=1}^{M} \exp\left(\gamma Y^{(i)} Y^T\right) X^{(i)} X^T \right. \\ &\qquad \left. - \sum_{i=1}^{M} \exp\left(\gamma X^{(i)} X^T\right) Y^{(i)} Y^T\right) \\ &\neq -\sum_{i=1}^{M} \exp\left(\gamma Y^{(i)} Y^T\right) X^{(i)} (X' - X)^T \\ &= E_{X'} - E_X\end{aligned}$$

Hence, by this argument, we cannot conclude that $\Delta E_x \leq 0$ (i.e. Zhang's eqn. 19 and eqn. 20). This confirms that Zhang's proof of the stability is faulty. Jeng *et al.*'s and Zhang *et al.*'s proofs are erroneous primarily in that they focused on using a single energy function to represent the convergence process of the eBAM.

2.1 Stability of eBAM

Therefore, in this study, we present a novel two-phase method to verify the stability of eBAM. Assume that we are given M bipolar pattern pairs, which are:

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

where

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

$X_i \neq X_j$, $i \neq j$, and $Y_i \neq Y_j$, $i \neq j$. Instead of using Kosko's approach [1], we use the following evolution equations in the recall process of the eBAM:

$$\begin{aligned} y_k &= \begin{cases} 1, & \text{if } \sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} \geq 0 \\ -1, & \text{if } \sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} < 0 \end{cases} \\ x_k &= \begin{cases} 1, & \text{if } \sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} \geq 0 \\ -1, & \text{if } \sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} < 0 \end{cases} \end{aligned} \tag{2}$$

where b denotes a positive number, $b > 1$, '·' represents the inner product operator, x_k and x_{ik} are the kth bits of X and the X_i, respectively, and y_k and y_{ik} are for Y and Y_i, respectively. Herein, an exponential scheme is used to enlarge the attraction radius of every stored pattern pair and to augment the desired pattern in the recall reverberation process.

Theorem 1: The eBAM modelled by eqn. 2 is a stable system.

Proof: We discuss the stability by observing the behaviour of the Lyapunov functions [14] of two directions, X → Y and Y → X, respectively.

Phase 1: X → Y. Define an energy function:

$$E_1(X,Y) = -\sum_{i=1}^{M} (X_i \cdot X) b^{Y_i \cdot Y} \tag{3}$$

Thus, the $\nabla_{x_k} E_1(X, Y)$ can be computed as follows,

$$\nabla_{x_k} E_1(X,Y) = -\sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} \tag{4}$$

The difference of E_1 due to a bit's change, can therefore, be derived as:

$$\begin{aligned}\Delta_{x_k} E_1(X,Y) &= \nabla_{x_k} E_1(X,Y) \cdot \Delta_{x_k} \\ &= -\sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} \cdot (x'_k - x_k)\end{aligned} \tag{5}$$

Case I: $x_k = -x'_k$, then

$$\begin{aligned}\Delta_{x_k} E_1(X,Y) &= -2x'_k \cdot \left(\sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y}\right) \\ &= -2x'_k \cdot (x'_k) \\ &= -2(x'_k)^2 = -2(-x_k)^2 < 0\end{aligned}$$

Case II: $x_k = x'_k$, then

$$\Delta_{x_k} E_1(X,Y) = 0$$

In conclusion, the X → Y phase causes E_1 to decrease,

$$\Delta_{x_k} E_1(X,Y) \leq 0 \tag{6}$$

Phase 2 : Y → X. Define another energy function:

$$E_2(X,Y) = -\sum_{i=1}^{M} (Y_i \cdot Y) b^{X_i \cdot X} \tag{7}$$

Thus, the $\nabla_{y_k} E_2(X, Y)$ can be computed as follows,

$$\nabla_{y_k} E_2(X,Y) = -\sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} \tag{8}$$

The difference of E due to a bit's change, can therefore, be derived as:

$$\begin{aligned}\Delta_{y_k} E_2(X,Y) &= \nabla_{y_k} E_2(X,Y) \cdot \Delta_{y_k} \\ &= -\sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} \cdot (y'_k - y_k)\end{aligned} \tag{9}$$

(i) *Case I:* $y_k = -y'_k$, then

$$\begin{aligned}\Delta_{y_k} E_2(X,Y) &= -2y'_k \cdot \left(\sum_{i=1}^{M} y_{ik} b^{X_i \cdot X}\right) \\ &= -2y'_k \cdot (y'_k) \\ &= -2(y'_k)^2 = -2(-y_k)^2 < 0\end{aligned}$$

(ii) *Case II:* $y_k = y'_k$, then

$$\Delta_{y_k} E_2(X,Y) = 0$$

In conclusion, the Y → X phase also causes E_2 to decrease,

$$\Delta_{y_k} E_2(X,Y) \leq 0 \tag{10}$$

According to the results in eqns. 6 and 10, we can infer that, if the radix is sufficiently large, these two energy functions evolve to their individual local minima, which are assumed to contain the stored pattern pairs. Hence, eBAM is a stable system. Restated, the necessary condition for a stored pattern pair to be recalled correctly is that it must reside at local minima on each of the two energy planes, E_1 and E_2, respectively.

2.2 Absolute lower bound of the radix

As mentioned earlier, the radix, b, must be larger than one to make the systems converge and work. In addition, the larger the b, the larger the SNR (signal-to-noise ratio) and the capacity [7]. Theoretically, a sufficiently large radix can be used to encode any numbers of the pattern pairs without any difficulty in recalling a single pair. However, an infinitely large radix cannot be implemented in the network's hardware realisation. Besides, the fixed dynamic range of the subthreshold region of the MOS transistor limits the magnitude of the pattern's radix and the dimension [8]. Then, what exactly is the smallest radix which can adequately recall every stored unique pattern pair? We call this smallest b the absolute lower bound. The relationship between the radix and the number of pattern pairs to be stored thus becomes a critical issue. Zhang *et al.* noticed a similar problem, finding an inequality, eqn. 23 in [10]. However, they did not offer a deterministic method to compute the absolute lower bound of eBAM.

To further explore the relationship between the radix and the number of the pattern pairs, we can start from the SNR approach to obtain the maximum noise. Assume that (X_h, Y_h) is one of the patterns stored in the eBAM. Thus, eqn. 2 can be rewritten as:

$$y_k = \operatorname{sgn}\left(y_{hk} \cdot b^n + \sum_{i \neq h} y_{ik} b^{X_h \cdot X_i}\right) \qquad (11)$$

where n is assumed to be $\min(n, p)$ without any loss of robustness, where n, p denote the dimensions of X_i and Y_i, respectively. Since we expect Y_h to be recalled when X_h is presented to the network, the first term on the right side of eqn. 11 is the signal; the second term is the noise. If the y_k has the same sign as y_{hk}, the following criterion must be satisfied.

2.3 Absolute stability criterion

Assume that (X_h, Y_h) is one of the patterns stored in the eBAM, while all of the stored pattern pairs are (X_i, Y_i), $i = 1, \ldots, M$. For any stored pattern, its signal strength exceeds that of the noise:

$$b^n > \sum_{i \neq h} b^{X_h \cdot X_i} \qquad (12)$$

is called an absolute stability criterion. This criterion states the sufficient condition to recall any stored pattern.

Proof: The upper bound of the noise is as follows,

$$\left|\sum_{i \neq h} y_{ik} b^{X_h \cdot X_i}\right| \leq \sum_{i \neq h} |y_{ik}|\, b^{X_h \cdot X_i}$$

Hence, the signal must exceed the noise to obtain a correct recall,

$$|y_{hk} \cdot b^n| = b^n > \sum_{i \neq h} |y_{ik}|\, b^{X_h \cdot X_i} = \sum_{i \neq h} b^{X_h \cdot X_i}$$

The only unsolved term is the right-hand side of eqn. 12, (i.e. $\Sigma_{i \neq h}\, b^{X_h \cdot X_i}$). To ensure that every pattern pair is recalled when it is the desired pattern pair, the maximum of the noise term must be computed. Then, the derived radix herein is adequately large to cause the signal to overpower the noise. The stored patterns are assumed to be individually unique. We consider the worst case of pattern pairs distribution to the signal. Thus, the number of patterns X_i's, which are 1-bit Hamming distance away from the desired pattern X_h is at most C_1^n. Similarly, the number of patterns of X_i's which are 2-bit away from the desired X_h is at most C_2^n. Following the same observation, we can have the following formulation.

Assume that the number of noise terms, $M - 1$, possesses the following largest noise condition:

$$\sum_{k=1}^{r-1} C_k^n \leq M - 1 < \sum_{k=1}^{r} C_k^n$$

$$\sum_{k=0}^{r-1} C_k^n \leq M < \sum_{k=0}^{r} C_k^n \qquad (13)$$

where r denotes the largest Hamming distance between the noise patterns and the desired pattern when the noise patterns which are orderly and, respectively, 1-bit, 2-bit, ..., $(r - 1)$-bit away from the desired pattern are stored in the network. We refer to this pattern distribution as the worst case distribution. Restated, we have the following tabulations for $X_i \cdot X_h$ in Table 1. Notably, r in eqn. 13 is unique and can be found by computer programs or other skills if M is given (i.e. the r is deterministic as long as the number of pattern pairs is known).

Table 1: Noise terms with largest noise power

Xi · Xh	Number of terms	HD to the signal
$n - 2 \cdot 1$	C_1^n	1-bit away
$n - 2 \cdot 2$	C_2^n	2-bit away
⋮	⋮	⋮
$n - 2 \cdot (r - 1)$	C_{r-1}^n	$(r - 1)$-bit away

HD = Hamming distance

Substituting eqn. 13 and the results listed in Table 1 into eqn. 12, yields the following:

$$\begin{aligned} b^n &> \sum_{i \neq h} b^{X_h \cdot X_i} \\ &= \sum_{k=1}^{r-1} C_k^n \cdot b^{n-2k} + \left(M - 1 - \sum_{k=1}^{r-1} C_k^n\right) \cdot b^{n-2r} \\ &= b^n \left[\sum_{k=1}^{r-1} C_k^n \cdot b^{-2k} + \left(M - 1 - \sum_{k=1}^{r-1} C_k^n\right) \cdot b^{-2r}\right] \\ 1 &> \left(\sum_{k=0}^{r-1} C_k^n \cdot b^{-2k} - 1\right) + \left(M - \sum_{k=0}^{r-1} C_k^n\right) \cdot b^{-2r} \end{aligned} \qquad (14)$$

Therefore, we can conclude that the following theorem is valid.

2.3.1 Theorem of absolute lower bound of the radix:

Assume that (X_h, Y_h) is one of the patterns stored in the eBAM, while all of the stored pattern pairs are (X_i, Y_i), $i = 1, ..., M$. For any stored pattern, its signal strength exceeds that of the noise. We consider the worst case of pattern pairs distributions to the signal, the number of noise terms, $M - 1$, possesses the largest noise condition, and r denotes the largest Hamming distance between the noise patterns and the desired pattern. Any b which satisfies the condition below can recall all of the M different pattern pairs stored in the eBAM.

$$2 > \sum_{k=0}^{r-1} C_k^n \cdot b^{-2k} + \left(M - \sum_{k=0}^{r-1} C_k^n\right) \cdot b^{-2r} \quad (15)$$

is called the theorem of absolute lower bound of the radix. Restated, the absolute stability criterion is a prerequisite for the theorem of absolute lower bound of the radix.

2.4 Radix searching

In a typical CMOS VLSI process, a transistor operating in the subthreshold region and working as an exponential circuit has a dynamic range of approximately 10^5 to 10^7 [8, 11, 12]. Thus we need to study how the storage capacity of the eBAM changes if the dynamic range of its exponential circuits is limited. Assumed that the dynamic range (D) of the exponential circuits is fixed and:

$$D \equiv b^n$$

$b > 1$, where n denotes the number of bits in the memory patterns. Then, as n increases, b decreases and the capacity no longer scales exponentially with n. If n is extremely large, b is close to 1 because D is fixed. Then, the validity of recalling stored patterns deteriorates according to the result of [7]. Chiueh and Goodman proposed the ECAM; their work can help us to understand that the asymptotic storage capacity of the ECAM is proportional to the dynamic range (D) when the required attraction radius is 0.

While considering the dynamic range limitation [8], an approach must be developed to obtain the minimal radix, b, when a certain number of pattern pairs are given to be stored.

2.4.1 Deterministic radix searching algorithm: According to eqn. 2, the following two equalities are sufficient conditions for a pattern pair (X_j, Y_j) to reside in a stable state.

$$F_{x_{jk}} = x_{jk} \sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y_j} > 0$$

$$F_{y_{jk}} = y_{jk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X_j} > 0 \quad (16)$$

To find an optimal radix for the eBAM, a cost function is defined, which is the function of the radix, b.

$$J(b) = -\sum_{j=1}^{M}\sum_{k=1}^{n} F_{x_{jk}} \cdot H(F_{x_{jk}}) - \sum_{j=1}^{M}\sum_{k=1}^{p} F_{y_{jk}} \cdot H(F_{y_{jk}}) \quad (17)$$

where $H(t) = 0$ if $t > 0$ and $H(t) = 1$ if $t \leq 0$. This cost function has several properties to be a good measure of the radix.

(i) If all of the F_{xjk}s and F_{yjk}s surpass 0, $J(b) = 0$. That is, the optimal b is found.

(ii) If any of F_{xjk} or F_{yjk} is smaller than 0, $J(b)$ must surpass 0.

The above-mentioned properties imply that a gradient descent approach can be aptly used to reach $J(b) = 0$. Hence, the algorithm of searching the optimal radix is summarised as:

$$b(0) = 1.0$$

$$b(t+1) = b(t) - q \cdot \frac{\partial J(b)}{\partial b}, \qquad q > 0 \quad (18)$$

$$\begin{aligned}\frac{\partial J(b)}{\partial b} &= -\sum_{j=1}^{M}\sum_{k=1}^{n}\sum_{i=1}^{M} x_{jk}x_{ik} \cdot \left(\sum_{s=1}^{p} y_{is}y_{js}\right) \cdot b^{Y_i \cdot Y_j - 1} \cdot H(F_{x_{jk}}) \\ &\quad - \sum_{j=1}^{M}\sum_{k=1}^{p}\sum_{i=1}^{M} y_{jk}y_{ik} \cdot \left(\sum_{s=1}^{n} x_{is}x_{js}\right) \cdot b^{X_i \cdot X_j - 1} \cdot H(F_{y_{jk}}) \\ &= -\sum_{j=1}^{M}\sum_{k=1}^{n}\sum_{i=1}^{M} x_{jk}x_{ik} \cdot (Y_i \cdot Y_j) \cdot b^{Y_i \cdot Y_j - 1} \cdot H(F_{x_{jk}}) \\ &\quad - \sum_{j=1}^{M}\sum_{k=1}^{p}\sum_{i=1}^{M} y_{jk}y_{ik} \cdot (X_i \cdot X_j) \cdot b^{X_i \cdot X_j - 1} \cdot H(F_{y_{jk}}) \quad (19)\end{aligned}$$

where q in eqn. 18 is a constant to determine the step size of the gradient descent approach. Hence, this algorithm can deterministically compute the absolute lower bound of the radix.

2.4.2 Optimal radix updating algorithm: In Section 2.4.1, we merely discussed how to find an optimal b right before pattern pairs being stored in the eBAM. If more pattern pairs are to be stored after some pattern pairs have been installed, the original radix is not an optimal radix any more. Hence, two methods can solve this problem. First, all of the pattern pairs, including stored pairs and pairs to be stored, are taken into the algorithm in Section 2.4.1 to compute a new b. This method has a large overhead. That is, all of the stored pattern pairs must be extracted from the eBAM before searching for another optimal b. Hence, we propose a method to update the radix based only upon a given additional pattern pair.

Assume that M pattern pairs are already stored in an eBAM with a radix b_0, thereby causing all of the stored pairs to remain at their respective stable states. Then, we also have the following equation according to eqn. 17,

$$J_0(b) = 0, \qquad \forall b \geq b_0 \quad (20)$$

where $J_0(b)$ is the cost function of the eBAM with only the original M pattern pairs of patterns. If more pairs are to be encoded in this eBAM and every pair is still in its own stable state, a sufficient condition to satisly this demand is that the new radix must be larger than the original radix [7]. Assume that (X_h, Y_h) is the $(M + 1)$ the pair to be stored in the eBAM which has already stored M pattern pairs. Therefore, the overall cost function of the eBAM after the $(M + 1)$th pair is stored is formulated as:

$$J'(b) = -\sum_{j=1}^{M+1}\sum_{k=1}^{n} F'_{x_{jk}} \cdot H(F'_{x_{jk}}) - \sum_{j=1}^{M+1}\sum_{k=1}^{p} F'_{y_{jk}} \cdot H(F'_{y_{jk}}) \quad (21)$$

where:

$$F'_{x_{jk}} = x_{jk} \sum_{i=1}^{M+1} x_{ik} \cdot b^{Y_i \cdot Y_j}$$

$$F'_{y_{jk}} = y_{jk} \sum_{i=1}^{M+1} y_{ik} \cdot b^{X_i \cdot X_j}$$

If the $H(\cdot)$ terms are 0 given $b = b_0$, the original b_0 can sufficiently store and recall the additional pair. Thus, the algorithm stops. On the other hand, the $H(\cdot)$ terms are 1, which can be taken out from the new cost function for the sake of clarity. Hence, the new cost function can be rewritten as:

$$\begin{aligned}
J'(b) &= -\sum_{j=1}^{M+1}\sum_{k=1}^{n} F'_{x_{jk}} - \sum_{j=1}^{M+1}\sum_{k=1}^{p} F'_{y_{jk}} \\
&= -\sum_{k=1}^{n}\left(\sum_{j=1}^{M} F'_{x_{jk}} + F'_{x_{hk}}\right) - \sum_{k=1}^{p}\left(\sum_{j=1}^{M} F'_{y_{jk}} + F'_{y_{hk}}\right) \\
&= -\sum_{k=1}^{n}\left(\sum_{j=1}^{M}\left[x_{jk}\sum_{i=1}^{M+1} x_{ik} \cdot b^{Y_i \cdot Y_j}\right] + x_{hk}\sum_{i=1}^{M+1} x_{ik} \cdot b^{Y_i \cdot Y_j}\right) \\
&\quad - \sum_{k=1}^{p}\left(\sum_{j=1}^{M}\left[y_{jk}\sum_{j=1}^{M} y_{ik} \cdot b^{X_i \cdot X_j}\right] + y_{hk}\sum_{i=1}^{M+1} y_{ik} \cdot b^{X_i \cdot X_j}\right) \\
&= -\sum_{k=1}^{n}\left(\sum_{j=1}^{M}\left[F_{x_{jk}} + x_{jk}x_{hk} \cdot b^{Y_h \cdot Y_j}\right] + b^p + x_{hk}\sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y_h}\right) \\
&\quad - \sum_{k=1}^{p}\left(\sum_{j=1}^{M}\left[F_{y_{jk}} + y_{jk}y_{hk} \cdot b^{X_h \cdot X_j}\right] + b^n + y_{hk}\sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X_h}\right) \\
&= -\sum_{k=1}^{n}\left(\sum_{j=1}^{M} F_{x_{jk}} + b^p + 2 \cdot \sum_{i=1}^{M} x_{ik}x_{hk} \cdot b^{Y_h \cdot Y_i}\right) \\
&\quad - \sum_{k=1}^{p}\left(\sum_{j=1}^{M} F_{y_{jk}} + b^n + 2 \cdot \sum_{i=1}^{M} y_{ik}y_{hk} \cdot b^{X_h \cdot X_i}\right) \\
&= \left\{-\sum_{k=1}^{n}\sum_{j=1}^{M} F_{x_{jk}} - \sum_{k=1}^{p}\sum_{j=1}^{M} F_{y_{jk}}\right\} \\
&\quad + \left\{-n \cdot b^p - 2\sum_{k=1}^{n}\sum_{i=1}^{M} x_{hk} \cdot x_{ik} \cdot b^{Y_h \cdot Y_i} - p \cdot b^n - 2\sum_{k=1}^{p}\sum_{i=1}^{M} y_{hk} \cdot y_{ik} \cdot b^{X_h \cdot X_i}\right\} \\
&= J_0(b) + J_\Delta(b)
\end{aligned} \quad (22)$$

Since we have assumed all of the pattern pairs in the eBAM before the installation of the extra pair are recallable, then $J_0(b)$ is 0. Also, any new radix b must satisfy the condition given in eqn. 20 to ensure that all of the $M + 1$ pairs remain in stable states. Thus, $J_\Delta(b)$ can be deemed to be the legitimate new cost function of eBAM.

$$J_\Delta(b) = -n \cdot b^p - 2\sum_{k=1}^{n}\sum_{i=1}^{M} x_{hk} \cdot x_{ik} \cdot b^{Y_h \cdot Y_i} - p \cdot b^n - 2\sum_{k=1}^{p}\sum_{i=1}^{M} y_{hk} \cdot y_{ik} \cdot b^{X_h \cdot X_i} \quad (23)$$

Again, a gradient descent approach is used to find the optimal b for eBAM.

$$\begin{aligned}
b(0) &= b_0 \\
b(t+1) &= b(t) - q \cdot \frac{\partial J_\Delta(b)}{\partial b}, \qquad q > 0 \\
\frac{\partial J_\Delta(b)}{\partial b} &= -np \cdot b^{p-1} - 2\sum_{k=1}^{n}\sum_{i=1}^{M} x_{hk} \cdot x_{ik} \cdot (Y_h \cdot Y_i) \cdot b^{Y_h \cdot Y_i - 1} \\
&\quad - np \cdot b^{n-1} - 2\sum_{k=1}^{p}\sum_{i=1}^{M} y_{hk} \cdot y_{ik} \cdot (X_h \cdot X_i) \cdot b^{X_h \cdot X_i - 1}
\end{aligned} \quad (24)$$

Notably, the above algorithm stops when the overall cost given in eqn. 21 is 0 (i.e. $J'(b) = 0$). The final b is the updated radix for all of the $M + 1$ pattern pairs. Comparing eqn. 19 with eqn. 24 provides a significant meaning of the updating algorithm. Although eqn. 19 has the $O(n^3)$ computation complexity, eqn. 24 reduces the complexity to only $O(n^2)$ and still maintains the optimality of the radix.

2.4.3 Guarantee recall training algorithm: In the previous Sections, we considered that the retrieval key is always correct. Thus, the retrieval key cannot provide the error correction ability or so called fault tolerance ability. In the following, we present a deterministic algorithm which will select and 'train' the eBAM with pattern pairs to ensure that all of the stored pairs are guaranteed to be recalled in their own basins, respectively, with a certain basin radius. Such an initiative attempts to ensure that every stored pattern pair is guaranteed to be recalled if a given input pattern located in its basin.

Assume that an eBAM is given M pattern pairs, (X_1, Y_1), ..., (X_M, Y_M). Then, a training algorithm of the eBAM can be tabulated as follows.

Step 0: Allow $b(0)$ to be the initial value of the radix. Where $b(0)$ is computed by the searching algorithm of Section 2.4.1.

Step 1: Repeat Step 1 to 3 for (X_1, Y_1), ..., (X_M, Y_M).

Step 2: Reverse the sign of σ bits of X_g and Y_g, respectively, to be X and Y, where σ denotes a predetermined fault tolerance range (i.e. the radius of the basin for a given pair to reside). If $X_g \cdot X > X_i \cdot X$, and $Y_g \cdot Y > Y_i \cdot Y$, $\forall i$, $i = 1, ..., M$, $i \neq g$, then take (X, Y) as the input training pair. Otherwise, go to Step 1 and try another pair.

Step 3: Take (X, Y) as the training pair of the following two equations until $J(b) = 0$.

$$J(b) = -\sum_{k=1}^{n} x_{gk} \sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y} \cdot H\left(x_{gk} \sum_{i=1}^{M} x_{ik} \cdot b^{X_i \cdot X}\right) - \sum_{k=1}^{p} y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X} \cdot H\left(y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X}\right) \quad (25)$$

$$b(t+1) = b(t) - q \cdot \frac{\partial J(b)}{\partial b} \quad (26)$$

$$\frac{\partial J(b)}{\partial b} = -\sum_{k=1}^{n}\sum_{i=1}^{M} x_{gk} x_{ik} \cdot (Y_i \cdot Y) \cdot b^{Y_i \cdot Y_j - 1} \cdot H\left(x_{gk} \sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y}\right) - \sum_{k=1}^{p}\sum_{i=1}^{M} y_{gk} y_{ik} \cdot (X_i \cdot X) \cdot b^{X_i \cdot X_j - 1} \cdot H\left(y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X}\right) \quad (27)$$

The final b is assigned to be $b(0)$ for the training of the next input training pair. Go to Step 1 until all of the pattern pairs are trained.

Step 4: After all of the M pairs either deleted or trained, the final b is used to be the radix of the eBAM.

Notably, in the above algorithm, Step 2 aims to avert any confusion between the two identical pairs, and to increase the radius of the (X_g, Y_g) pair, because the reversion of the signs of σ bits in the basin radius of the training pair. Thus, those pattern pairs trained and stored in the eBAM are guaranteed to be recalled when the input pair is located in its basin of which the radius is σ.

The cost function in Step 3 is different from eqn. 17 because in this algorithm we train one pattern pair, say (X_g, Y_g), each time. If the (X, Y), which is σ bits away from (X_g, Y_g), can recall (X_g, Y_g), it must satisfy the following two conditions according to eqns. 2 and 16,

$$x_{gk} \sum_{i=1}^{M} x_{ik} \cdot b^{Y_i \cdot Y} > 0$$

$$y_{gk} \sum_{i=1}^{M} y_{ik} \cdot b^{X_i \cdot X} > 0$$

Hence, eqn. 25 is defined as the cost function to train (X_g, Y_g).

Table 2: Simulation of searching an optimal radix for the 8 × 8 eBAM

No.	b	No.	b
1	2.790619	6	2.855878
2	2.790124	7	2.832530
3	2.757570	8	2.790527
4	2.733255	9	2.807437
5	2.738821	10	2.821274

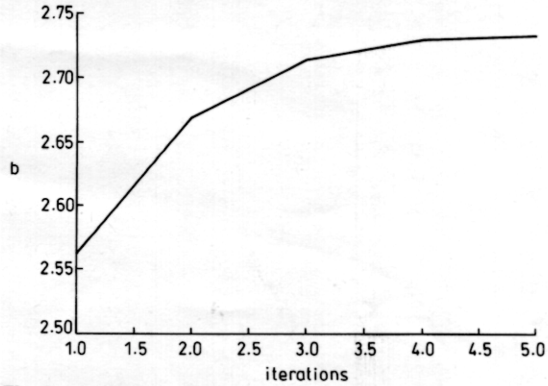

Fig.2 *Radix in searching process*

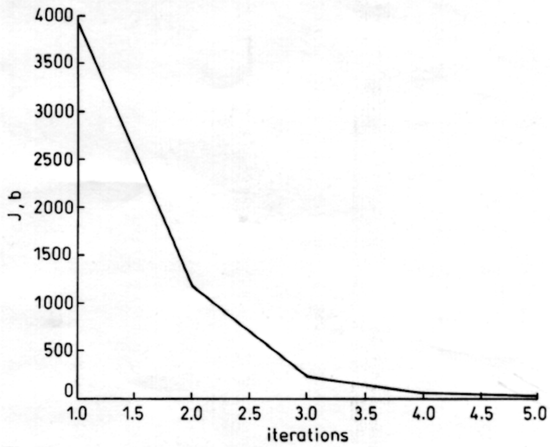

Fig.3 *Cost function in searching process*

3 Simulation analysis

Example 1: The radix searching algorithm is aimed at looking for a minimal radix which can still recall all of the pairs to be stored. Herein, we apply the algorithm to store 200 pattern pairs in an 8 × 8 eBAM, and repeat the algorithm for ten times to obtain Table 2. Note that the reason why the bs are different in Table 2 is that the 200 pairs are randomly generated in each simulation. However, the radix derived from the searching algorithm is close to the natural base, $e \approx 2.71828$. This finding thus confirms the feasibility of using the natural e while implementing the eBAM by VLSI circuits. Figs. 2 and 3 illustrate examples of the evolution of b and $J(b)$, respectively, during the searching procedure.

Example 2: An 8 × 8 eBAM is stored with 70 different pattern pairs. We use the radix searching algorithm in Section 2.4.1 to obtain the radix which makes the eBAM stable and pairs recallable, and then use the training algorithm in Section 2.4.3 to derive the radix which ensures that every pair is recalled within its basin, $\sigma = 1$. Table 3 summarises those results.

Table 3: Simulation of training the 8 × 8 eBAM

No.	*b* (searching)	*b* (training)	No.	*b* (searching)	*b* (training)
1	2.251508	2.270246	6	2.349620	2.532752
2	2.277386	2.277387	7	2.134500	2.779011
3	2.213780	2.664818	8	2.317027	2.491975
4	2.201022	2.830397	9	2.263478	2.509213
5	2.438400	2.575281	10	2.304018	2.326273

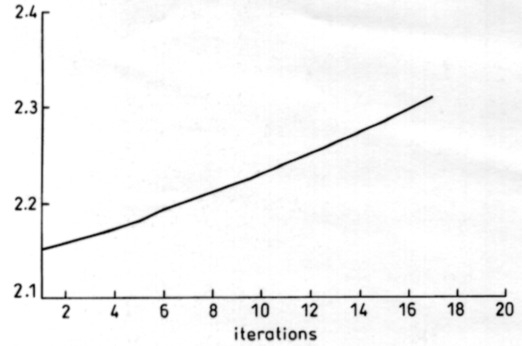

Fig. 4 *Radix in training process*

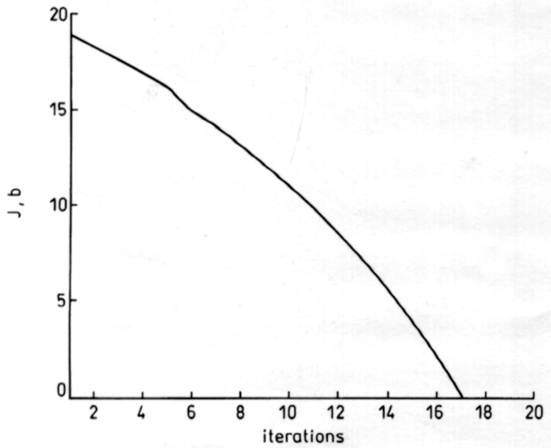

Fig. 5 *Cost function in training process*

Notably, as expected, the trained b is always larger than searched b. The reason for this discrepancy is that we want every trained pair to be recalled with $\sigma = 1$. This condition is more restrictive than merely being statistically recallable. After the training procedure, all of the seventy pairs are correctly recalled with one bit error in the above ten simulations. An example of pairs is correctly recalled with one bit error in the above ten simulations. Figs. 4 and 5 present examples of the evolution of b and $J(b)$, respectively, during the training procedure.

4 Conclusion

This work employs a two-phase method to demonstrate the stability of eBAM. A deterministic method is also proposed to calculate the absolute lower bound of the radix which is the smallest radix capable of recalling every stored pattern pair. Furthermore, to preserve the optimality of the radix due to the limitation of dynamic range, we also present another algorithm to update the optimal radix for the eBAM when additional pattern pairs are to be stored. Finally, a deterministic algorithm to train the encoding of pattern pairs is also derived, capable of ensuring that every trained pair in the eBAM is accurately recalled within a predetermined basin radius. Furthermore, simulation results verify all of the theories presented herein.

5 Acknowledgment

This research was partially supported by the National Science Council under grant NSC 83-0404-E-110-014.

6 References

1 KOSKO, B.: 'Adaptive bidirectional associative memory', *Appl. Opt.*, 1987, **26**, (23), pp. 4947–4960
2 KOSKO, B.: 'Bidirectional associative memory', *IEEE Trans.*, 1988, **SMC-18**, (1), pp. 49–60
3 WANG, Y.-F., CRUZ, J.B., and MULLIGAN, J.H.: 'Two coding strategies for bidirectional associative memory', *IEEE Trans.*, 1990, **NN-1**, (1), pp. 81–92
4 SIMPSON, P.K.: 'Higher-ordered and intraconnected bidirectional assocaitive memory', *IEEE Trans.*, 1990, **SMC-20**, (3), pp. 637–653
5 TAI, H.M., WU, C.H., and JONG, T.L.: 'High-order bidirectional associative memory', *Electron. Lett.*, 1989, **25**, (21), pp. 1424–1425
6 WANG, T., ZHUANG, X., and XING, X.: 'Weighted learning of bidirectional associative memories by global minimization', *IEEE Trans.*, 1992, **NN-3**, (6), pp. 1010–1018
7 WANG, C.-C., and DON, H.-S.: 'An analysis of high-capacity discrete exponential BAM', *IEEE Trans.*, 1995, **NN-6**, (2), pp. 492–496
8 CHIUEH, T.D., and GOODMAN, R.M.: 'Recurrent correlation associative memories', *IEEE Trans.*, 1991, **NN-2**, (2), pp. 275–284
9 JENG, Y.-J., YEH, C.-C., and CHIUEH, T.-D.: 'Exponential bidirectional associative memories', *Electron. Lett.*, 1990, **26**, (11), pp. 717–718
10 ZHANG, B.-L., XU, B.-Z., and KWANG, C.-P.: 'Performance analysis of the bidirectional associative memory and an improved model from the match-filtering viewpoint', *IEEE Trans.*, 1993, **NN-4**, (5), pp. 864–872
11 GLASSER, L.A., and DOPPERPUHL, D.W.: 'The design and analysis of VLSI circuits' (Addison-Wesley, Reading, MA, 1985)
12 MEAD, C.A.: 'Analog VLSI and neural systems' (Addison-Wesley, Reading, MA, 1989)
13 CHIUEH, T.D., and GOODMAN, R.M.: 'High-capacity exponential associative memory', *Proc. IJCNN*, 1988, **1**, pp. 153–160
14 HECHT-NIELSEN, R.: 'Neurocomputing' (Addison-Wesley, Reading, MA, 1990)