Efficient Pattern Recalling using Parallel Alpha-Beta Associative Memories

Mario Aldape-Pérez, Cornelio Yáñez-Márquez, and Oscar Camacho-Nieto

Center for Computing Research, CIC
National Polytechnic Institute, IPN
Mexico City, Mexico
Contact: mario@aldape.org.mx, cyanez@cic.ipn.mx
oscarc@cic.ipn.mx
http://www.aldape.org.mx

(Paper received on February 29, 2008, accepted on April 15, 2008)

Abstract. Associative memories have a number of properties, including a rapid, compute efficient best-match and intrinsic noise tolerance that make them ideal for many applications [1-4]. However, a significant bottleneck to the use of associative memories in real-time systems is the amount of data that requires processing. The aim of this paper is to present the work that produced a dedicated hardware design, implemented on a field programmable gate array (FPGA) that applies the Alpha-Beta Associative Memories model for pattern recognition tasks. Along the experimental phase, performance of the proposed associative memory architecture is measured by learning large sequences of symbols and recalling them successfully.

1 Introduction

An associative memory $\mathbf{M}$ is a system that relates input patterns and output patterns as follows: $\mathbf{x} \rightarrow \mathbf{M} \rightarrow \mathbf{y}$ with $\mathbf{x}$ and $\mathbf{y}$, respectively, the input and output pattern vectors. Each input vector forms an association with its corresponding output vector. For each k integer and positive, the corresponding association will be denoted as: (x^k, y^k). An Associative memory $\mathbf{M}$ is represented by a matrix whose ij-th component is m_{ij} [5]. Memory $\mathbf{M}$ is generated from an a priori finite set of known associations, called the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\{ (x^\mu, y^\mu) \mid \mu = 1,2,...,p \}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu \; \forall \mu \in \{1,2,...,p\}$ $\mathbf{M}$ is auto-associative, otherwise it is heteroassociative; in this case, it is possible to establish that $\exists \mu \in \{1,2,...,p\}$ for which $x^\mu \neq y^\mu$. If we consider the fundamental set of patterns $\{ (x^\mu, y^\mu) \mid \mu = 1,2,...,p \}$ where n and m are the dimensions of the input patterns and output patterns, respectively, it is said that $x^\mu \in A^n$, $A = \{0,1\}$ and $y^\mu \in A^m$. Then the j-th component of an input pattern is $x_j^\mu \in A$. Analogously, the j-th component of an output pattern is represented

Special Issue in Electronics and Biomedical Informatics,
Computer Science and Informatics
Research in Computing Science 35, 2008, pp. 147-156

as $y_j^{\mu} \in A$. A distorted version of a pattern x^k to be recuperated will be denoted as $\tilde{x}^k$. If when feeding an unknown input pattern x^{ω} with $\omega \in \{1,2,...,k,...,p\}$ to an associative memory $\mathbf{M}$, it happens that the output corresponds exactly to the associated pattern y^{ω}, it is said that recuperation is correct.

2 Alpha-Beta Associative Memories

Alpha-Beta Associative Memories mathematical foundations are based on two binary operators: α and β. Alpha operator is used during the learning phase while Beta operator is used during the recalling phase. The mathematical properties within these operators, allow the $\alpha\beta$ associative memories to exhibit similar characteristics to the binary version of the morphological associative memories, in the sense of: learning capacity, type and amount of noise against which the memory is robust, and the sufficient conditions for perfect recall [6]. First, we define set $A = \{0,1\}$ and set $B = \{00,01,10\}$ so α and β operators can be defined as in Table 1.

Table 1. Alpha and Beta operators.

$\alpha : A \text{ x } A \rightarrow B$

x	y	$\alpha(x,y)$
0	0	01
0	1	00
1	0	10
1	1	01

$\beta : B \text{ x } A \rightarrow A$

x	y	$\beta(x,y)$
00	0	0
00	1	0
01	0	0
01	1	1
10	0	1
10	1	1

These two binary operators along with maximum ($\vee$) and minimum ($\wedge$) operators establish the mathematical tools around the Alpha-Beta model. According to the type of operator that is used during the learning phase, two kinds of Alpha-Beta Associative Memories are obtained. If maximum operator ($\vee$) is used, Alpha-Beta Associative Memory of type MAX will be obtained, denoted as $\mathbf{M}$; analogously, if minimum operator ($\wedge$) is used, Alpha-Beta Associative Memory of type min will be obtained, denoted as $\mathbf{W}$ [7].

In order to understand how the learning and recalling phases are carried out, some matrix operations definitions are required.

- α max Operation: $P_{mxr} \nabla_{\alpha} Q_{rxn} = [f_{ij}^{\alpha}]_{mxn}$, where $f_{ij}^{\alpha} = \vee_{k=1}^{r} \alpha(p_{ik}, q_{kj})$
- α min Operation: $P_{mxr} \Delta_{\alpha} Q_{rxn} = [f_{ij}^{\alpha}]_{mxn}$, where $f_{ij}^{\alpha} = \wedge_{k=1}^{r} \alpha(p_{ik}, q_{kj})$
- β max Operation: $P_{mxr} \nabla_{\beta} Q_{rxn} = [f_{ij}^{\beta}]_{mxn}$, where $f_{ij}^{\beta} = \vee_{k=1}^{r} \beta(p_{ik}, q_{kj})$
- β min Operation: $P_{mxr} \Delta_{\beta} Q_{rxn} = [f_{ij}^{\beta}]_{mxn}$, where $f_{ij}^{\beta} = \wedge_{k=1}^{r} \beta(p_{ik}, q_{kj})$

Whenever a column vector of dimension m is operated with a row vector of dimension n, both operations ∇_α and Δ_α, are represented by $\oplus$; consequently, the following expression is valid:

$$y\nabla_\alpha x^t = y \oplus x^t = y\Delta_\alpha x^t \tag{1}$$

If we consider the fundamental set of patterns $\{(x^\mu, y^\mu) \mid \mu = 1,2,...,p\}$ then the ij-th entry of the matrix $y^\mu \oplus (x^\mu)^t$ is expressed as follows:

$$\left[y^\mu \oplus \left(x^\mu\right)^t\right]_{ij} = \alpha(y_i^\mu, x_j^\mu) \tag{2}$$

2.1 Learning Phase

Find the adequate operators and a way to generate a matrix $\mathbf{M}$ that will store the p associations of the fundamental set $\{(x^1,y^1),(x^2,y^2),(x^3,y^3),...,(x^p,y^p)\}$, where $x^\mu \in A^n$ and $y^\mu \in A^m$ $\forall \mu \in \{1,2,...,p\}$.

Step 1. For each fundamental pattern association $\{(x^\mu, y^\mu) \mid \mu = 1,2,...,p\}$, generate p matrices according to the following rule:

$$\left[y^\mu \oplus \left(x^\mu\right)^t\right]_{m \times n} \tag{3}$$

Step 2. In order to obtain an Alpha-Beta Associative Memory of type MAX, apply the binary MAX operator ($\vee$) according to the following rule:

$$\mathbf{M} = \vee_{\mu=1}^{p}\left[y^\mu \oplus \left(x^\mu\right)^t\right] \tag{4}$$

Consequently, the ij-th entry of an Alpha-Beta Associative Memory of type MAX is given by the following expression:

$$v_{ij} = \vee_{\mu=1}^{p} = \alpha(y_i^\mu, x_j^\mu) \tag{5}$$

Step 3. In order to obtain an Alpha-Beta Associative Memory of type min, apply the binary min operator ($\wedge$) according to the following rule:

$$\mathbf{W} = \wedge_{\mu=1}^{p}\left[y^\mu \oplus \left(x^\mu\right)^t\right] \tag{6}$$

Analogously, the ij-th entry of an Alpha-Beta Associative Memory of type min is given by the following expression:

$$\psi_{ij} = \wedge_{\mu=1}^{p} = \alpha(y_i^\mu, x_j^\mu) \tag{7}$$

2.2 Recalling Phase

Find the adequate operators and sufficient conditions to obtain the fundamental output pattern y^μ, when either the memory $\mathbf{M}$ or the memory $\mathbf{W}$ is operated with the fundamental input pattern x^μ.

Step 1. An unknown input pattern x^ω with $\omega \in \{1,2,...,p\}$ is presented to the Alpha-Beta Associative Memory.

Step 2. In order to obtain an unknown output pattern y^ω with $\omega \in \{1,2,...,p\}$, an Alpha-Beta Associative Memory of type MAX will be used according to the following rule:

$$\mathbf{M}\Delta_\beta x^\omega = \wedge_{j=1}^{n} \beta(v_{ij}, x_j^\omega) = \wedge_{j=1}^{n} \{[\vee_{\mu=1}^{p} \alpha(y_i^\mu, x_j^\mu)], x_j^\omega\} \tag{8}$$

Step 3. In order to obtain an unknown output pattern y^ω with $\{ \omega = 1,2,...,p \}$, an Alpha-Beta Associative Memory of type min will be used according to the following rule:

$$\mathbf{W}\nabla_\beta x^\omega = \vee_{j=1}^{n} \beta(\psi_{ij}, x_j^\omega) = \vee_{j=1}^{n} \{[\wedge_{\mu=1}^{p} \alpha(y_i^\mu, x_j^\mu)], x_j^\omega\} \tag{9}$$

Without dependence on the Alpha-Beta Associative Memory type used throughout the recalling phase, a column vector of dimension m will be obtained.

3 Numerical Results

Let p=5, n=4, m=4. Given the fundamental patterns $\{ (x^\mu, y^\mu) \mid \mu = 1,2,...,p \}$, obtain an Alpha-Beta Associative Memory. The fundamental associations will be denoted as: $\{ (x^1, y^1), (x^2, y^2), ..., (x^5, y^5) \}$.

$$x^1 = \begin{bmatrix}1\\1\\0\\1\end{bmatrix} \quad x^2 = \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \quad x^3 = \begin{bmatrix}0\\1\\1\\1\end{bmatrix} \quad x^4 = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \quad x^5 = \begin{bmatrix}1\\0\\1\\1\end{bmatrix}$$

$$y^1 = \begin{bmatrix}1\\1\\0\\1\end{bmatrix} \quad y^2 = \begin{bmatrix}1\\0\\0\\1\end{bmatrix} \quad y^3 = \begin{bmatrix}1\\1\\1\\1\end{bmatrix} \quad y^4 = \begin{bmatrix}1\\1\\0\\0\end{bmatrix} \quad y^5 = \begin{bmatrix}1\\0\\1\\1\end{bmatrix}$$

3.1 Learning Phase

Obtain the corresponding matrices $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_5$, according to step 1, indicated in section 2.1.

$$y^1 \oplus (x^1)^t = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 01 & 01 & 10 & 01 \\ 01 & 01 & 10 & 01 \\ 00 & 00 & 01 & 00 \\ 01 & 01 & 10 & 01 \end{bmatrix}$$

$$y^2 \oplus (x^2)^t = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 10 & 10 & 01 \\ 01 & 01 & 01 & 00 \\ 01 & 01 & 01 & 00 \\ 10 & 10 & 10 & 01 \end{bmatrix}$$

$$\vdots$$

$$y^5 \oplus (x^5)^t = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 01 & 10 & 01 & 01 \\ 00 & 01 & 00 & 00 \\ 01 & 10 & 01 & 01 \\ 01 & 10 & 01 & 01 \end{bmatrix}$$

According to step 2 in section 2.1, an Alpha-Beta Associative Memory of type MAX denoted by $\mathbf{M}$, is obtained. Analogously, according to step 3 in section 2.1, an Alpha-Beta Associative Memory of type min denoted by $\mathbf{W}$, is obtained.

$$\mathbf{M} = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 \\ 10 & 10 & 01 & 01 \\ 10 & 10 & 10 & 01 \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} 01 & 01 & 01 & 01 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \\ 01 & 00 & 01 & 01 \end{bmatrix}$$

3.2 Recalling Phase

Obtain the corresponding output patterns, by performing the operations $\mathbf{M}\Delta_\beta x^\omega$, $\forall \mu \in \{1,2,\ldots,p\}$ as stated in section 2.2. Due to paper space limitations, only the Alpha-Beta MAX type recalling phase results are shown.

$$\mathbf{M}\Delta_\beta x^1 = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 \\ 10 & 10 & 01 & 01 \\ 10 & 10 & 10 & 01 \end{bmatrix} \Delta_\beta \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = y^1$$

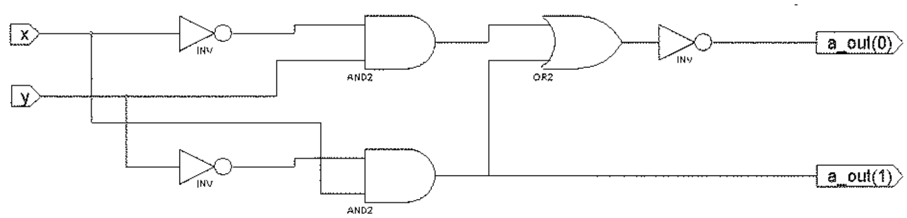

Fig. 1. Alpha unit.

$$\mathbf{M}\Delta_\beta x^2 = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 \\ 10 & 10 & 01 & 01 \\ 10 & 10 & 10 & 01 \end{bmatrix} \Delta_\beta \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = y^2$$

$$\vdots$$

$$\mathbf{M}\Delta_\beta x^5 = \begin{bmatrix} 10 & 10 & 10 & 10 \\ 10 & 01 & 10 & 10 \\ 10 & 10 & 01 & 01 \\ 10 & 10 & 10 & 01 \end{bmatrix} \Delta_\beta \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = y^5$$

The reader can easily verify that the Alpha-Beta min type recalling phase also recalls the whole fundamental set of patterns perfectly.

4 Implementation Details

As previously mentioned, the main goal of this paper is to derive an efficient implementation of the Alpha-Beta Associative Memories which exploits the inherent parallelism of this mathematical model, targeted towards FPGAs. The Alpha operator implementation is shown in Figure 1, while the Beta operator implementation is shown in Figure 2.

The proposed architecture works with a 50 MHz master clock, which implies a 20ns period. As is it shown in Figure 3, the learning phase is implemented with 5 registers, 1 MAX/min block and 2 external 10ns SRAM chips (mounted on the same board), that allow 1MB of data storage, 8 Alpha blocks that allow byte processing instead of bit processing, resulting in an eight times faster learning phase compared against [8].

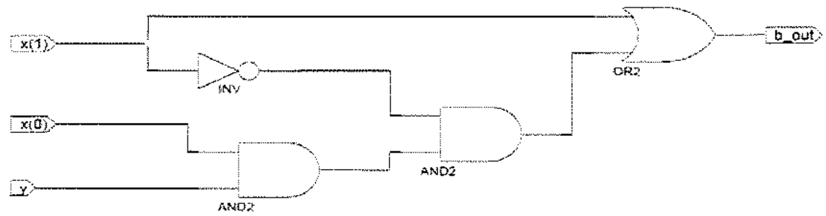

Fig. 2. Beta unit.

There are two remarkable topics to be taken into consideration. The former concerns about the amount of logic resources that are needed to implement the two binary operators (Alpha and Beta). The latter results from the fact that most of the components that constitute the learning phase are combinatorial circuits. Hence, it is possible to read data from the external SRAM memory at the same time that a new bit is shifted to the Alpha blocks.

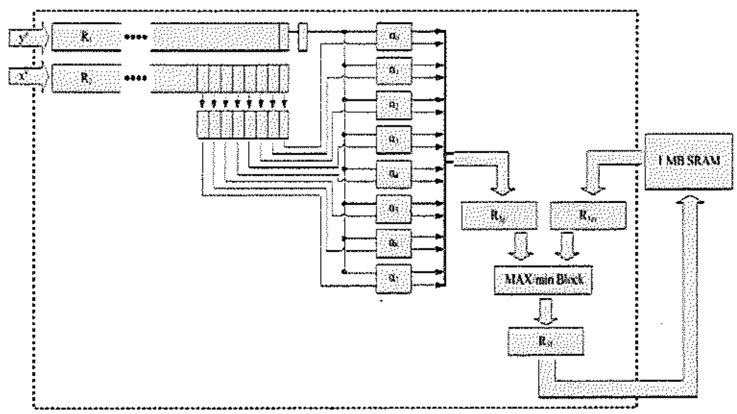

Fig. 3. Learning Phase Architecture.

As it is shown in Figure 4, the recalling phase is implemented with 4 registers, 8 Beta block, 1 min/MAX block and the same 2 external 10ns SRAM chips that were used to store the fundamental associations during the learning phase. The recalling phase is executed as follows. Firstly, R_{3_m} receives one data word from the Alpha-Beta Associative Memory (stored in the 2 external 10ns SRAM chips). Then, R_1 receives the unknown input pattern. Finally, R_2 stores the recalled output pattern.

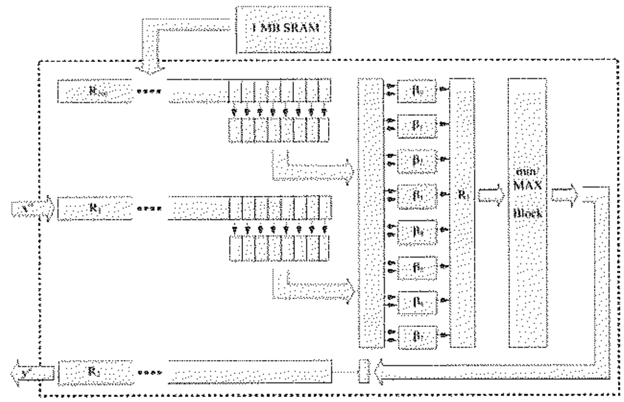

Fig. 4. Recalling Phase Architecture.

5 Experimental Results

The experimental phase was carried out in two stages. In the first one, the same fundamental set of patterns that was presented in section 3, was downloaded to the proposed architecture. The performance results are shown in Figure 5 and Figure 6. As expected, the entire fundamental set of patterns was perfectly recalled. In order to estimate how the Alpha-Beta Associative Memory model performs with high dimensional data, 20 binary images obtained from the Third International Fingerprint Verification Competition (*FVC2004*) were used as fundamental patterns (Figure 7). Originally, each one of these images is 160 by 120 pixels.

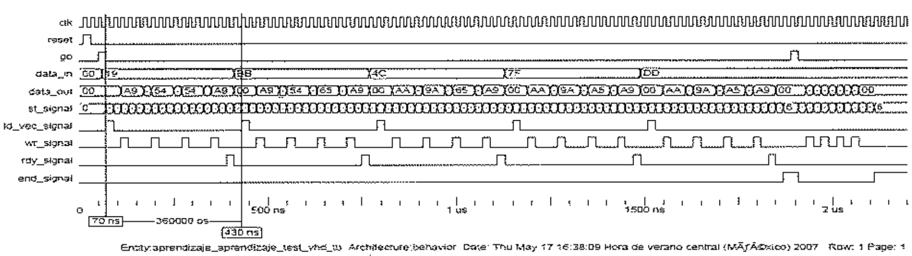

Fig. 5. Learning Phase Performance.

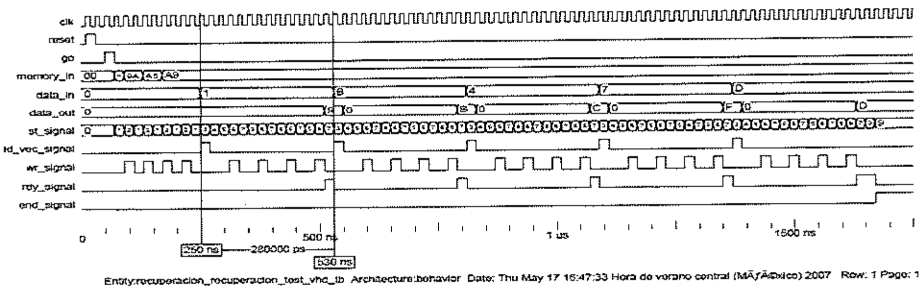

Fig. 6. Recalling Phase Performance.

The Advanced Batch Converter image editor was used to modify the fingerprints dimensions, such that it was possible to keep the pattern associations over the previously mentioned SRAM chips. The experimental phase was carried out as follows: after the register initialization process was concluded, the first association was learned and recalled. Subsequently, the first and second associations were learned and recalled; after that, the same procedure continued in a consecutive manner until the fundamental set of patterns was completely learned and recalled. The above mentioned procedure was executed 10 times, each time changing the fundamental patterns order randomly. A relevant thing to mention about the recalling criterion that was used along the experimental phase is that, in this case, perfect recall means that all of the 1600 bits were exactly recovered. Particularly, outstanding results were achieved by both types of Alpha-Beta Associative Memories (the whole fundamental set of patterns was perfectly recalled).

6 Conclusions and Ongoing Research

In this paper, we introduced a simple but efficient implementation of the Alpha-Beta Associative Memories which exploits the inherent parallelism of this mathematical model targeted towards FPGAs that overcomes a serious challenge in pattern recognition tasks (bottle-neck problems due to high dimensional data). A relevant thing to mention is that after a fundamental pattern is downloaded to the proposed architecture, each bit is learned in 90 ns, which fulfils one of the main purposes of this paper. Moreover, if the learning rate is known, it is possible to estimate the learning phase duration even with high dimensional fundamental patterns. Usually, this situation takes place when the fundamental patterns are RGB images. It is worth to mention that the proposed architecture can be easily adapted to work as an Alpha-Beta bi-directional associative memory [6-7].

Fig. 7. Fundamental Patterns.

Acknowledgments

The authors of the present paper would like to thank the following institutions for their economical support to develop this work: National Polytechnic Institute, Mexico (CIC, SIP, PIFI, COFAA) , CONACyT and SNI.

References

1. Yáñez-Márquez, C., Felipe-Riverón, E. M., López-Yáñez, I., and Flores-Carapia, R. (2006). A Novel Approach to Automatic Color Matching, LNCS 4225: 529-538.
2. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., and López-Yáñez, I. (2006). Alpha-Beta Bidirectional Associative Memories Based Translator, IJCSNS 6: 190-194.
3. Yáñez-Márquez, C., Sánchez-Fernández, L. P., and López-Yáñez, I. (2006). Alpha-Beta Associative Memories for Gray Level Patterns, LNCS 3971: 818-823.
4. Sossa, H., Barrón, R., Cuevas, F., and Aguilar, C. (2005). Associative Gray Level Pattern Processing using Binary Decomposition and α β Memories, Neural Processing Letters, 22: 85-111.
5. Kohonen, T. (1972). Correlation Matrix Memories, IEEE Transactions on Computers, 21: 353-359.
6. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., and López-Yáñez, I. (2006). A New Model of BAM: Alpha-Beta Bidirectional Associative Memories, LNCS 4263: 286-295.
7. Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., and López-Yáñez, I. (2007). Alpha-Beta bidirectional associative memories: theory and applications, Neural Processing Letters, 26: 1-40.
8. Yáñez-Márquez C. (2002). Associative memories based on order relations and binary operators (in Spanish) PhD Thesis, Center for Computing Research, México.