Chip-Architecture for Automatic Learning Based on Associative Memory and Short/Long Term Storage Concept

Masahiro Mizokami, Yoshinori Shirakawa, Tetsushi Koide, and Hans Jürgen Mattausch

Research Center for Nanodevices and Systems, Hiroshima University, 1-4-2 Kagamiyama, Higashi-Hiroshima, 739-8527, Japan

Phone: +81-824-24-6265   Fax: +81-824-22-7185   E-mail:{mizokami, shirakawa, koide, hjm}@sxsys.hiroshima-u.ac.jp

1. Introduction

Pattern recognition and learning are basic functions, which are needed to build artificial systems with capabilities similar to the human brain. Their effective implementation in integrated circuits is therefore of great technical importance. Among the methods for achieving the pattern recognition and the learning functions, that have been proposed so far, the neural-network approach is most widely used. However, the performance progress of hardware that uses neural-networks is much slower than expected initially. Because of this difficult situation a new method that includes the memory element, missing up to now, in a power-efficient LSI hardware is urgently needed.

Presently, we are developing a new associative-memory architecture which achieves small area and high nearest-match speed [1-3]. The proposed architecture can search the reference pattern of minimum distance to the input pattern at high speed for different distance measures. Therefore, it is expected that this small-area and high-speed associative memory becomes the basis for a new method to construct systems with recognition and the learning capability. Moreover, there is the advantage that it can be easily integrated by the use of present CMOS technology.

In this paper, we propose a new associative-memory-based automatic-learning architecture for artificial intelligence systems that have recognition and learning capability. In this architecture the associative memory that we have developed is used to imitate the long-term and the short-term memory of the human brain. (Fig. 1) The automatic learning algorithm and its CMOS implementation are in detail described in the following.

2. Short/Long Term Storage Concept

A memory-based learning system can achieve higher learning efficiency than a neural-network, for which a complicated training is necessary at the beginning to enable the recognition of new data. For a memory-based method, the training which we call "supervised learning" corresponds only to writing of the new data into the memory. The proposed learning algorithm, explained in the following can furthermore automatically learn input data according to the frequency of their appearance, a learning mode, which we call "unsupervised learning".

The basic underlying concept of our proposal tries to model the short/long term storage of the human brain. Therefore, the reference patterns of the associative memory are classified into two areas. One is a short-term storage area where new information is temporarily memorized. The other is a long-term storage area where a reference pattern can be memorized for a longer time without receiving the influence of the constantly changing input patterns.

Fig.2 shows the flow chart of the associative-memory-based recognition and the learning algorithm. The proposed algorithm uses a "rank" for each reference pattern of the associative memory as an index. The reference patterns are classified into the long-term and the short-term memory according to the order of rank. Important specifying features of the algorithm are the process for changing the rank of a pattern and the method for pattern transition between short and long-term memory.

3 Automatic Learning Algorithm

Details of the proposed algorithm are as follows.

Step1. The associative memory searches for the pattern, which is the nearest-match (winner) to the input pattern among the reference patterns.

Step2. The distance "D" between input pattern and winner-pattern is calculated.

Step3. Input pattern and winner-pattern are considered to be the same in case of "D<threshold". In this case, the rank of the reference pattern that became the winner is raised. The rank advancement is decided based on the previous rank of the winner. When the winner belongs to the long-term memory, the advancement becomes J_L. And, if the winner is in the short-term memory, the advancement becomes J_S ($J_L > J_S$). Each of the patterns of rank between the old and the new winner rank are reduced in rank by one. The transition between short and long-term memory occurs by these changes in rank.

Step4. In the case of "D threshold", the distance between input pattern and winner still is minimum. However, because the distance between the two patterns is large, the system considers these two patterns to be different, and inserts the winner pattern at the top tank of the short-term memory. The rank of each of the other reference patterns that exist in the short-term memory is moved down by one, and the reference pattern with the lowest rank is erased and forgotten.

Step5. Whenever input data is given to the system, processing from steps 1 to step 4 are repeated.

The user of the proposed algorithm has to properly decide the number of pattern entries in the short-term and long-term memory (N_S and N_L) according to the application.

4. Simulation and Test-Chip Design

A simulator was written in C programming language to verify the effectiveness of the proposed learning algorithm. The size of the associative memory for the verification experiments was chosen to allow holding of 30 patterns with 256bit each. The Hamming Distance was selected as the distance measure. The relative sizes of long-term and short-term memory were set at 2:1, which means $N_L=20$ and $N_S=10$. The remaining parameters of the learning algorithm were chosen as threshold=10, $J_L=6$ and $J_S=3$.

The automatic learning of 20 new character-bit patterns (each 256 bit) was investigated as the test problem. These 20 new patterns were presented to the system as inputs randomly. The learning task was additionally complicated by also presenting noise patterns as inputs, where each of the 256 bits was set at random to 1 or 0. These noise patterns were arbitrarily intermixed with the 20 new character-bit patterns at the same rate (50% noise patterns, 50% new character-bit patterns).

Fig.3 depicts the simulation result for the number of learned patterns among the 20 new character-bit patterns as a function of the total number of presented input patterns. The blue line shows the result without short/long-term storage concept, where an input pattern, which is identified as new, is stored at the topmost rank of the associative memory. Due to the intermixed noise patterns, the new character-bit patterns cannot be learned efficiently. The number of learned patterns oscillates around 10 due to the noise intermixture rate of 50%. The red line shows the result with the short/long term storage concept, which completes the learning of all 20 new character-bit patterns after about 1800 input cycles, even under the presence of noise-input patterns. The short/long-term memory concept and the transition mechanism from short-term to long-term memory have the effect that noise patterns are unable to advance from the

short-term to the long-term memory. These concepts are thus the key to efficient memory-based hardware for automatic learning.

Fig.4 shows the architecture of the test chip, which is divided roughly into the associative memory block, the rank-processing circuit and the automatic learning control circuit. A test chip of the described architecture was designed in 0.35um CMOS technology. (Fig.5) The automatic learning circuit of the test chip receives the result of the nearest-match search from the associative memory, including the input-winner distance (D), and generates the signals for the rank-processing circuit and the learning signals for the associative memory within one clock cycle. Table.1 shows the parameter table of the designed test chip. The associative memory finishes the nearest-match search in 250nsec or less, and the automatic learning circuit operates at a maximum operation frequency of 166MHz (gate level simulation).

5. Conclusion

We proposed a new associative-memory-based automatic-learning architecture and verified its effectiveness by simulation and CMOS test-chip design. This architecture is expected to enable an automatic learning function in integrated intelligent systems, which is not possible with the conventional neural-network concept.

References

[1] H. J. Mattausch et al. , "Compact associative-memory architecture with fully-parallel search capability for the minimum hamming distance", IEEE Journal. of Solid-State Circuits, Vol.37, pp.218-227, 2002.
[2] H. J. Mattausch et al. , "Fully-parallel pattern-matching engine with dynamic adaptability to Hamming or Manhattan distance", 2002 Symposium on VLSI Circuit Dig. of Tech. Papers, pp.252-255, 2002.
[3] H. J. Mattausch et al. , "An architecture for Compact Associative Memories with Deca-ns Nearest-Match Capability up to Large Distance", ISSCC Dig. of Tech. Papers, pp.170-171, 2001.
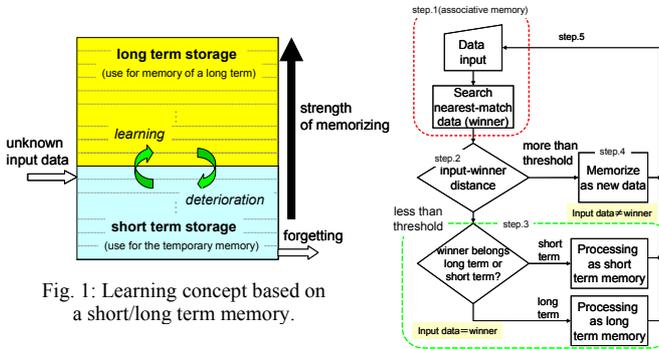
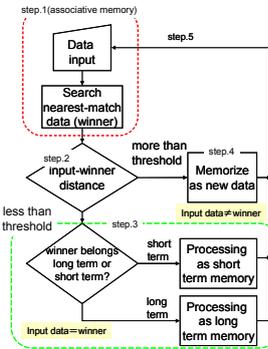Fig. 1: Learning concept based on a short/long term memory.
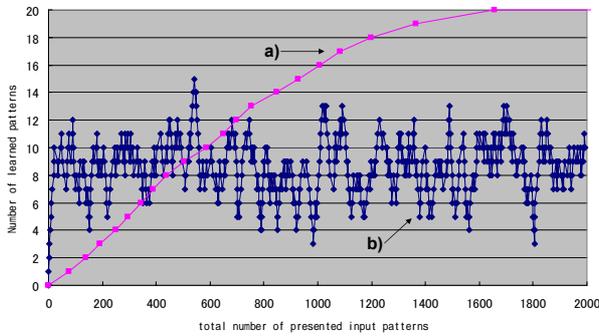

Fig.2: Flow chart of proposed algorithm.

Fig. 3: Simulation result.
a) Learning algorithm based on short/long term storage concept.
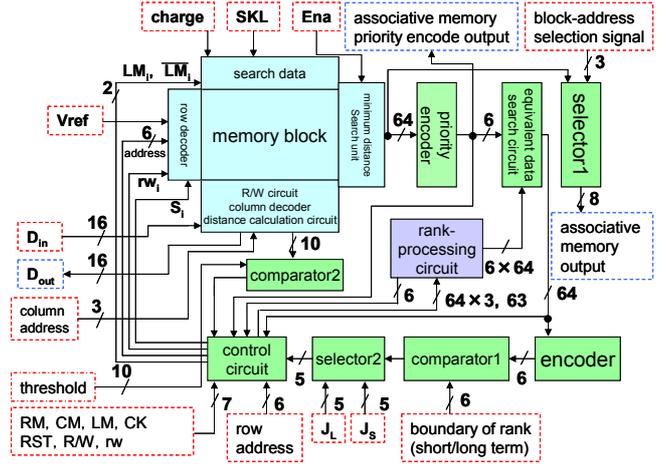b) Without short/long term storage (unknown data is inserted at the top rank).

Fig.4: Associative-memory-based recognition and automatic learning architecture, which corresponds to the associative memory that can process 64 patterns. Parameters J_L, J_S and the threshold in the algorithm can set externally.
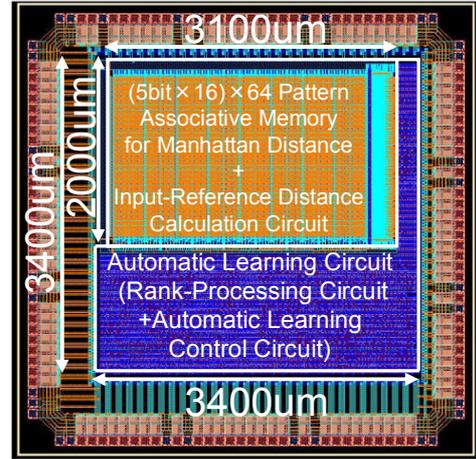

Fig.5: Layout of test chip.

Table.1: Characteristics of the designed test chip.

Distance Measure	Manhattan Distance (5bit × 16		
Reference Patterns	64		
Short Term Storage	24	Default	Variable
Long Term Storage	40	Default	Variable
Nearest-Match Range	0 496		
Technology	0.35um 2-poly 3-metal CMOS		
Supply Voltage	3.3V		
Number of Transistors	402,768		
Design Area	11.04mm²		
Associative Memory	6.2mm²		
Automatic Learning Circuit	4.84mm²		
Automatic Learning Algorithm Processing Time	< 290nsec search time 250nsec		
Automatic Learning Circuit Max Operation Frequency	166MHz gate level simulation		