A Word-Parallel Digital Associative Engine with Wide Search Range Based on Manhattan Distance

Yusuke Oike[†], Makoto Ikeda[†‡], and Kunihiro Asada[†‡]

[†]Dept. of Electronic Engineering, University of Tokyo
[‡]VLSI Design and Education Center (VDEC), University of Tokyo
7–3–1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
Phone: +81-3-5841-6719, Fax: +81-3-5841-8912
E-mail: {y-oike, ikeda, asada}@silicon.u-tokyo.ac.jp

Abstract

A word-parallel digital associative engine with accurate and wide-range Manhattan-distance computation is presented. It performs a continuous search operation to detect not only the nearest-match data but also all data in the sorted order of the exact Manhattan distance. The word-parallel digital implementation using a hierarchical search path provides a high-speed search operation with faultless precision, a low-voltage operation mode, and a potential capability of unlimited data capacity. Word-parallel distance calculation circuits autonomously count the Manhattan distance using a weighted search clock to detect the nearest-match data. An associative engine, with 64 words of 8 bit × 32 element, has been fabricated using a 0.18 μm CMOS process and successfully tested. The worst-case search time of all data sorting takes 5.85 μs at a supply voltage of 1.8 V.

Introduction

Associative processors based on content addressable memories (CAMs) have been proposed for various applications such as pattern recognition, data compression and intelligent processing to reduce the considerable memory access and processing time [1]–[6]. Some fully-parallel processors [1]–[4] employ *Hamming distance* for associative processing since *Hamming distance* estimation is realized by less computational effort than *Manhattan distance*. On the other hand, associative processing based on *Manhattan distance* is capable of many practical applications such as vector-quantization recognition [5], code-book-based image compression [6] and so on as shown in Fig.1. Although associative processors based on *Hamming distance* are capable of *Manhattan distance* estimation using thermometer encoding as reported in [2], they require 2^i bit length for i-bit data elements. Therefore, associative processing with a compact bit length requires the natural binary coding for *Manhattan distance* such as [6]–[9].

In this paper, we present a word-parallel associative engine with accurate and wide-range Manhattan-distance computation. The word-parallel digital implementation using a hierarchical search path enables a high-speed search operation with faultless precision, a low-voltage operation mode, and a potential capability of unlimited data capacity. These features are important for a system-on-a-chip application in future process technologies, which is difficult to attain using the conventional mixed-signal approaches [7]–[9]. Furthermore, it performs a continuous search operation to detect not only the nearest-match data but also all data in the sorted order of the exact Manhattan distance. It requires considerable search operations in case of the conventional architectures [6]–[9]. Word-parallel distance calculation circuits autonomously count the Manhattan distance using a weighted search clock to detect the nearest-match data. The unique associative processing with accurate and wide-range Manhattan-distance computation efficiently realizes various new applications such as human-like learning and high-speed data sorting in addition to the conventional use. An associative engine, with 64 words of 8 bit × 32 element, has been fabricated in a 0.18 μm CMOS process and successfully tested.

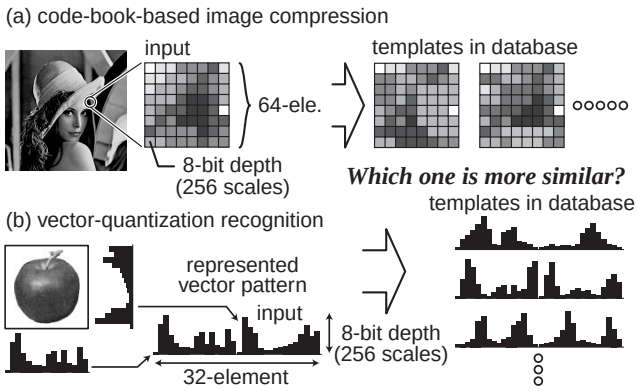

Fig. 1 Application examples of Manhattan-distance search.

Word-Parallel Manhattan Distance Computation

A. Element Circuit Structure and Computation Flow

Associative processing based on Manhattan distance generally handles i-bit × j-element data as shown in Fig.1. Manhattan distance compuation requires SAD (summation of absolute difference) between an input and all stored data. Fig.2 (a) shows an 8-bit element structure. The stored data are divided into blocks and hierarchically connected by a bypass line to reduce the search signal propagation path as shown in Fig.2 (b). The 8-bit element consists of 8 SRAM cells, a bit selector, a subtractor based on a half adder (HA) with an absolute function (ABS), a flag register (FR) with a bit comparison function, and a chained search circuit as shown in Fig.3.

The present algorithm and circuit implementation for Manhattan distance computation are shown in Fig.4 through Fig.7. First, absolute flags are generated in element parallel. Then, a distance counting operation is executed by a chained search signal propagation in word parallel. It is processed by weighted search clocks which are autonomously provided by word-parallel distance calculation circuits. Finally, the nearest-match data is detected in *Candidates* which are activated by the word-parallel calculation circuits at the same time. All the data can be detected by a continuous search operation in the sorted order of Manhattan distance.

B. Absolute Flag Generation

Fig.4 (a) shows the element-parallel absolute flag generation. First, an input data A_{ij} is compared with a stored data B_{ij} from MSB to LSB in element parallel. It determines '$A_{ij} > B_{ij}$' or '$A_{ij} < B_{ij}$' using an input $\overline{A_{ij}}$ and a sum result S_{ij} of HA. The

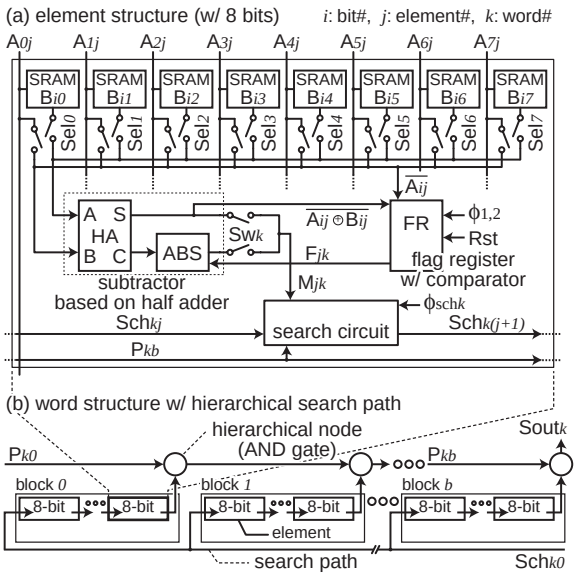

Fig. 2 Block diagram: (a) an 8-bit element structure, (b) a word structure with hierarchical search path.

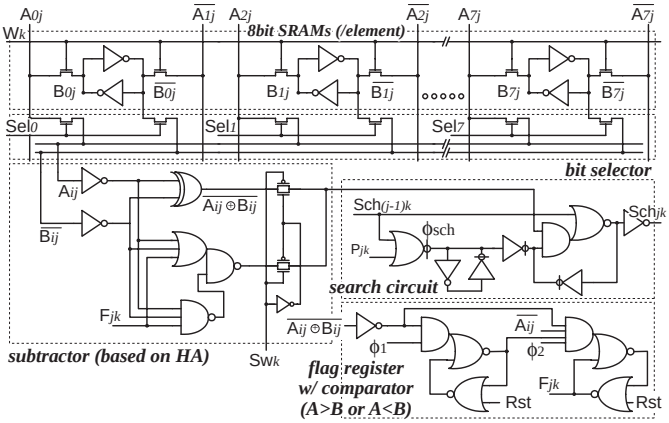

Fig. 3 Circuit configuration of an 8-bit element cell.

comparison result F_{jk} is stored in a flag resister and used for an absolute function by switching a carry result C_{ij} of HA between $A_{ij} \cdot \overline{B_{ij}}$ and $\overline{A_{ij}} \cdot B_{ij}$. The absolute difference is calculated in element parallel during the word-parallel summation.

C. Distance Counting Operation

The distance counting operation is executed from LSBs to MSBs of elements in word parallel as shown in Fig.4 (b). A sum result S_{0j} of A_{0j} and B_{0j} is set to M_{jk} as a control signal of a chained search circuit. A search signal detects the first-encountered mismatch bit with $M_{jk} = 1$ in each block. The search clock period is limited by the search signal propagation path via chained search circuits. Therefore, a hierarchical search path based on [3] is implemented as shown in Fig.2 (b). A bypass search signal P_{kb} is also used for a mask permission signal to the next block, which makes only one mismatch bit maskable in each word for the next clock period. The interrupted search signal starts again from the masked bit, and finally a search signal can be detected as $Sout_k$ when all the mismatch bits have been masked. Therefore, the operation clocks represent the number of mismatch bits. After that, a distance counting operation is executed again for a carry result C_{0j} in a similar manner to the counting operation for a sum result S_{0j}. These counting operations are repeated from A_{0j} to A_{7j}.

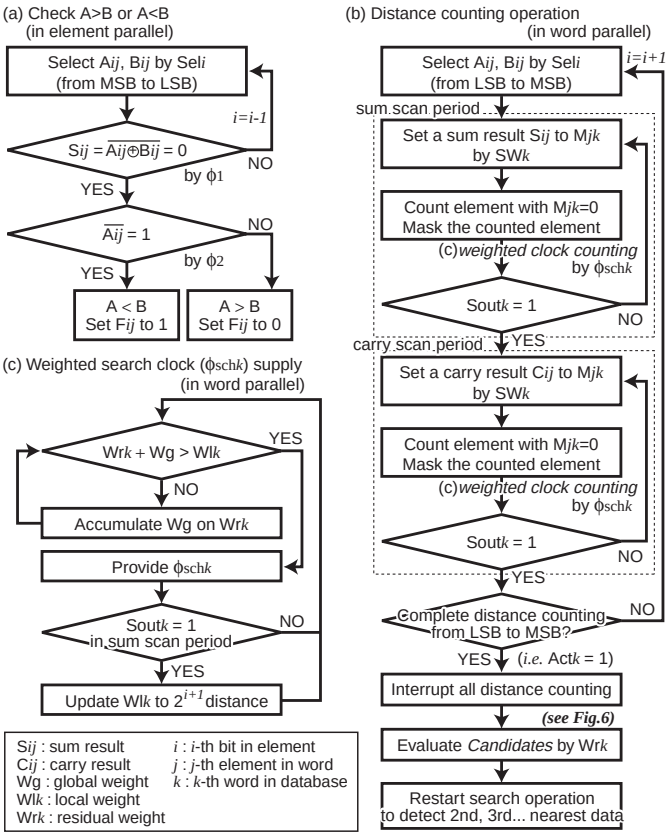

Fig. 4 Search operation flow: (a) absolute flag generation, (b) distance counting operation, (c) weighted search clock supply.

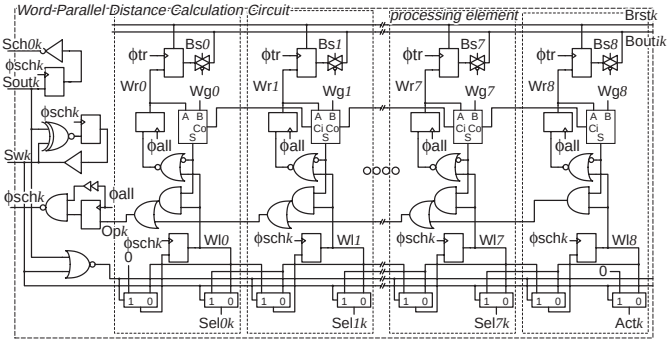

Fig. 5 Word-parallel distance calculation circuits using autonomous weighted search clocks.

D. Weighted Search Clock Technique

Fig.5 shows a word-parallel distance calculation circuit using autonomous weighted search clocks. The word-parallel circuit receives the search output signal $Sout_k$, and it counts the Manhattan distance based on a weight of a search clock ϕsch. A search clock has different weights according to the bit number i that is currently evaluated in elements. For example, it has a weight of 2^i- and 2^{i+1}-bit Manhattan distance during a counting operation for i-th sum and carry outputs, respectively. A word-parallel circuit autonomously provides ϕsch_k to count all the mismatch bits faster. Therefore, it has a local weight Wl_k as a current weight of ϕsch_k, and accumulates a global weight Wg on a residual weight Wr_k as shown in Fig.4 (c). A search clock ϕsch_k is provided and the local weight Wl_k is subtracted from Wr_k when the sum total of Wr_r and Wg exceeds Wl_k. The local weight Wl_k always precedes the global weight Wg in every word since the global weight Wg is commonly updated according to the worst case. Some fractional weights caused

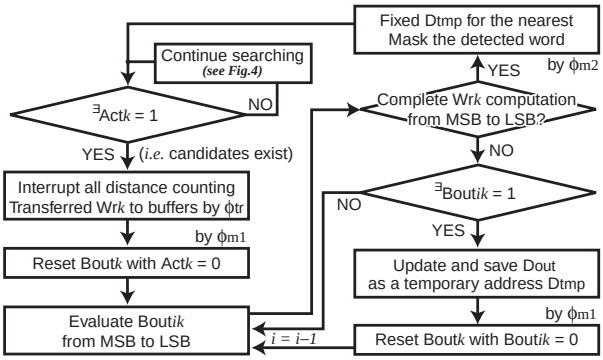

Fig. 6 Nearest-match detection flow in candidates.

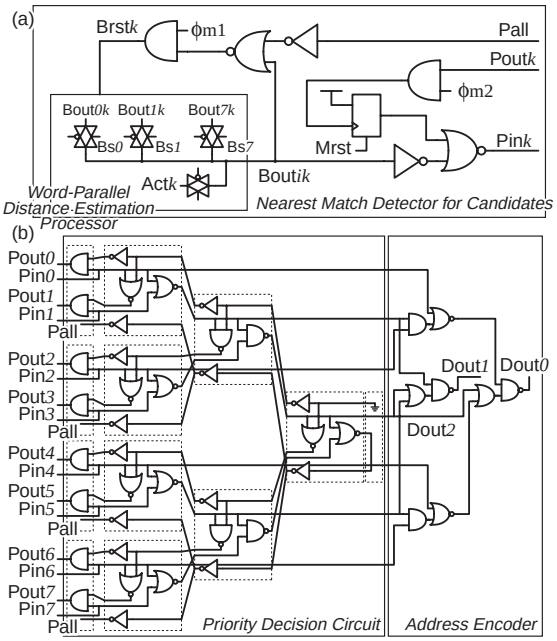

Fig. 7 Circuit configuration: (a) a nearest-match detector for candidates, (b) a binary-tree priority encoder simplified with 8 inputs.

by the precedence are stored as a residual weight Wr_k. In the present counting technique, the number of processing elements per word is determined by just the bit length N per element as shown in Fig.5. A word-parallel circuit also controls bit select signals Sel_{ik} according to Wl_k, and finally provides Act_k to a priority address encoder as a *Candidate*.

E. Nearest-Match Detection in Candidates

The distance counting operation is interrupted at the detection timing of Act_k, and then the process moves to nearest-match detection for *Candidates* as shown in Fig.6. *Candidates* are all the words activated by Act_k at the same time. They have different residual weight according to their Manhattan distance from the input since the distance is given by $\Sigma Wg - Wr_k$. ΣWg is the total distance weight operated before the detection timing of Act_k. Note that *Candidates* are closer to the input than all the other undetected words in the present search algorithm, hence they include the nearest match data. This feature contributes to detect the nearest-match data, and also enables a continuous search operation for data sorting in order of the exact Manhattan distance. The nearest-match detection in *Candidates* is carried out by a nearest-match detector and a priority address encoder. It evaluates each residual weight Wr_k from MSB to LSB as shown in Fig.6. The process maintains consistency with each other word. It keeps all residual weights other than

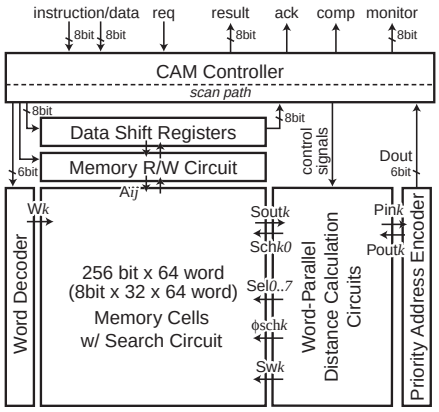

Fig. 8 Block diagram of Manhattan-distance associative engine.

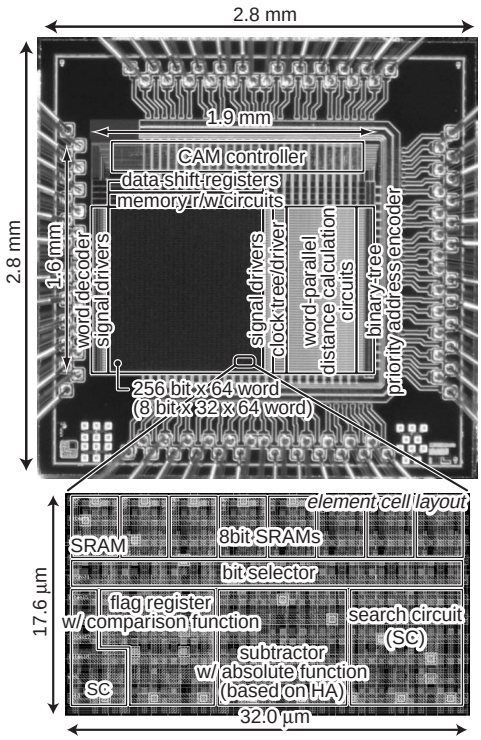

Fig. 9 Chip microphotograph and layout of an element cell.

the nearest data in *Candidates*, and then the detected nearest data is masked to continue a search operation for the next nearest data. The circuit configuration is shown in Fig.7.

Chip Implementation

We have designed and fabricated an associative engine using the present search architecture in a 1P5M 0.18 μm CMOS process[1]. Fig.8 illustrates a block diagram of the search engine. It consists of a search memory array with 64 words of 8 bit $\times$ 32 element, a memory read/write circuit with data shift registers, a word decoder, word-parallel distance calculation circuits, a priority address encoder for nearest-match detection in candidates, and a CAM controller. These components are implemented in a die size of 2.8×2.8 mm^2. Fig.9 shows a chip microphotograph and an 8-bit element cell layout. A 32-element word is divided into four blocks to reduce the critical path.

[1]The chip in this study has been fabricated through VLSI Design and Education Center(VDEC), University of Tokyo in collaboration with Hitachi Ltd. and Dai Nippon Printing Co.

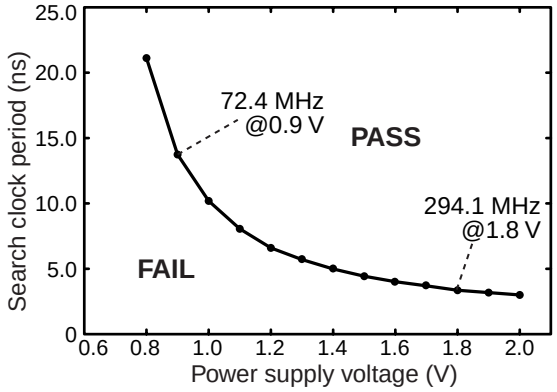

Fig. 10 Power supply voltage vs search clock period.

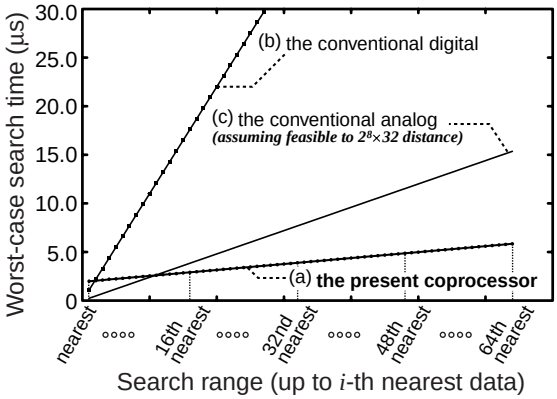

Fig. 11 Characteristics of the present continuous search operation for wide-range associative processing.

Measurement Results and Discussions

The measurement results show that the operation speed attains 294.1 MHz and the power dissipation is 320.7 mW at a supply voltage of 1.8 V. The total search time for nearest-match detection is 2.00 μs in the worst case. Fig.10 shows the operation speed as a function of the supply voltage from 0.8 V to 2.0 V. The fully digital implementation enables a low-voltage operation mode up to 0.8 V. It attains an operation frequency of 72.4 MHz and a power dissipation of 15.1 mW at 0.9 V. The associative processing ensures Manhattan distance computation with faultless precision.

Fig.11 shows the worst-case search time for wide-range Manhattan distance computation. The present search engine is capable of a continuous search operation to detect all data in the sorted order of the exact Manhattan distance in addition to the nearest-match data. It efficiently realizes a wide-range search operation as shown by (a) in Fig.11. On the other hand, the conventional architectures require considerable search operations. Fig.11 (b) is estimated based on [6] as a conventional digital technique. Fig.11 (c) is estimated based on [9] as a conventional mixed-signal technique assuming that it is scalable to the same capacity as the present coprocessor since there was no report on such a long distance search by mixed-signal techniques so far. A capacity scalability is also one of advantages of the present digital implementation.

Table I shows the core area and SRAM ratio of various data capacities. The integration ratio of SRAMs is almost equivalent to the ratio of 19 % of the conventional digital processor [6]. Furthermore, the present architecture has the possibility of a large database capacity in a practical die size since it makes device scaling easier than the conventional mixed-signal techniques. Table II summarizes the chip specifications.

TABLE I Core area and SRAM ratio.

Data size	Core area	SRAM ratio
8-bit 32-ele. 64-word (16K)	2.37 mm^2	17.2 %
8-bit 64-ele. 128-word (64K)	6.70 mm^2	21.9 %
8-bit 128-ele. 256-word (256K)	22.25 mm^2	25.3 %
8-bit 256-ele. 512-word (1M)	81.04 mm^2	27.5 %

TABLE II Specifications of the associative coprocessor.

Process	1P5M 0.18 μm CMOS process
Chip size	2.8 mm $\times$ 2.8 mm
Power voltage supply	0.8 V – 1.8 V
Database capacity	8-bit 32-element 64-word templates
Distance measure	Manhattan distance
Functions	Nearest detection / All data sorting
Nearest detection time	1.65 μs ~ 2.00 μs
All data sorting time	5.85 μs
Operation speed	294.1 MHz @ 1.8 V 72.4 MHz @ 0.9 V
Power dissipation	320.7 mW @ 1.8 V, 294.1 MHz 15.1 mW @ 0.9 V, 72.4 MHz

Conclusions

We have proposed a new word-parallel digital architecture and circuit implementation for accurate and wide-range Manhattan distance computation employing a hierarchical search path and a weighted search clock technique. It is capable of the detection of all data in the sorted order of the exact Manhattan distance in addition to the nearest-match data. The weighted search clock technique performs the wide-range associative processing with fewer additional cycles. Furthermore, the digital implementation enables a low-voltage operation for SoC applications in future process technologies. It also makes device scaling easier and provides the possibility of a large data capacity with unlimited search distance. An associative engine, with 64 words of 8 bit $\times$ 32 element, has successfully performed the Manhattan distance computation. The worst-case search time of all data sorting takes 5.85 μs at a supply voltage of 1.8 V.

References

[1] M. Ikeda *et al.*, "Time-Domain Minimum-Distance Detector and Its Application to Low-Power Coding Scheme on Chip-Interface," in *Proc. of ESSCIRC*, pp. 464 – 467, Sep. 1998.

[2] H. J. Mattausch *et al.*, "Fully-Parallel Pattern-Matching Engine with Dynamic Adaptability to Hamming or Manhattan Distance," *Symp. on VLSI Circuits Dig. Tech. Papers*, pp. 252 – 255, Jun. 2002.

[3] Y. Oike *et al.*, "A High-Speed and Low-Voltage Associative Co-Processor With Hamming Distance Ordering Using Word-Parallel and Hierarchical Search Architecture," in *Proc. of IEEE CICC*, pp. 643 – 646, Sep. 2003.

[4] S. Nakahara *et al.*, "A Digital Circuit for a Minimum Distance Search Using an Asynchronous Bubble Shift Memory," *IEEE ISSCC Dig. Tech. Papers*, pp. 504 – 505, Feb. 2004.

[5] M. Yagi *et al.*, "An Image Representation Algorithm Compatible With Neural-Associative-Processor-Based Hardware Recognition Systems," *IEEE Tran. Neural Networks*, Vol. 14, No. 5, pp. 1144 – 1161, Sep. 2003.

[6] A. Nakada *et al.*, "A Fully Parallel Vector-Quantization Processor for Real-Time Motion-Picture Compression," *IEEE J. Solid-State Circuits*, Vol. 24, No. 6, pp. 822 – 830, Jun. 1999.

[7] T. Yamashita *et al.*, "Neuron MOS Winner-Take-All Circuit and Its Application to Associative Memory," *IEEE ISSCC Dig. Tech. Papers*, pp. 236 – 237, Feb. 1993.

[8] M. Nagata *et al.*, "A Minimum-Distance Search Circuit using Dual-Line PWM Signal Processing and Charge-Packet Counting Techniques," *IEEE ISSCC Dig. Tech. Papers*, pp. 42 – 43, Feb. 1997.

[9] Y. Yano *et al.*, "Associative Memory with Fully Parallel Nearest-Manhattan-Distance Search for Low-Power Real-Time Single-Chip Applications," in *Proc. of IEEE ASP-DAC*, pp. 543 – 544, Jan. 2004.