A 256-Element Associative Parallel Processor

Frederick P. Herrmann, *Member, IEEE,* and Charles G. Sodini, *Fellow, IEEE*

***Abstract*— A 256-element associative processing chip is designed for pixel-parallel image processing and machine vision applications. A five-transistor three-state dynamic memory cell is used, and each processing element has 64 trits of memory. Other processing element components include a function generator, an activity register, and connections to a reconfigurable mesh network and a response resolution subsystem. These are implemented with compact circuits designed within memory pitch constraints. The chip was fabricated in a double-poly CCD-CMOS process and characterized as fully functional. A sample image processing application is demonstrated on a four-chip prototype system.**

I. Introduction

IMAGE processing problems tend to be rich in parallelism, with thousands or millions of pixels receiving identical processing steps. Massively parallel machines are ideally suited to such tasks, but million-dollar supercomputers are overkill for all but a few applications. More economical solutions are available with pipelined digital chips [1] or arrays of simple analog circuits [2], [3], but these highly specialized designs are typically optimized for one particular application. Associative processors occupy a middle ground between these extremes. As smart memory systems in which every word functions as a small processing element, associative processors retain the flexibility of programmable digital systems, but they can achieve a high level of integration when implemented in a memory design style. The associative processor described in this work is well suited to problems in machine vision and image processing which map naturally onto large processor-per-pixel arrays.

The block diagram in Fig. 1 shows how an associative processor can serve as an image processing coprocessor controlled by a host computer. A two-dimensional network connects the processing elements and provides an interface for image input and output. A format converter reorders the image bits before loading them into the associative processor. A second format converter may be necessary at the output, depending on the destination of the processed images.

Fig. 2 shows a processing element with 64 general-purpose associative memory cells, organized as two 32-cell half-words. Data stored in the memory is compared with patterns presented on the column lines, and the sense amplifier (SA) detects the match result. Both stored and presented patterns may include *don't cares* ($\times$'s), which match both 0's and 1's. The match result passes from the sense amplifier to the word logic, comprising a function generator (FG) and a single-bit activity register (AR). The function generator takes a second input from the AR, for which it computes a new value during every match operation. The function generator is also used to control the two write drivers, so that write operations can be enabled or disabled by the results of previous matches. This match-write feedback path is one of two features that distinguish associative processors like this one from associative memories [4]. The other is the ability to mask individual cells while writing, so that some cells in the word are preserved while others are modified. These match and conditional write operations turn out to be surprisingly powerful primitives, which can be combined to perform more complex tasks, including bit-serial arithmetic.

In addition to the 64 general-purpose trit cells, each PE also has six specialized cell pairs. The carry cell C holds intermediate values for bit-serial arithmetic and other algorithms, while the home (H) and directional (NEWS) cell pairs are used for inter-PE communication and response resolution. Each special cell pair has one match-only A cell and one match/write B cell. (e.g., cells HA and HB make up the H cell pair.)

The next two sections describe the circuit design of the memory and the network. Section IV presents some experi-

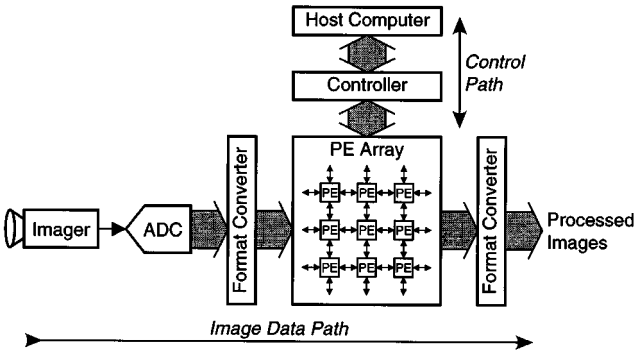

Fig. 1. Image processing system using array of associative processing elements.

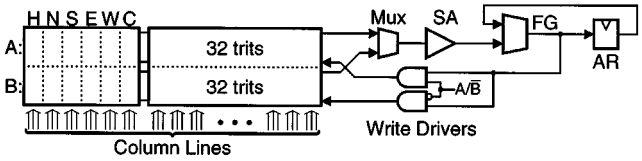

Fig. 2. Associative processing element.

Manuscript received September 20, 1994; revised January 24, 1995. This work was supported by NSF and ARPA, Contract MIP-9117724, and by IBM.

F. P. Herrmann was with the Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, MA 02173 USA.

C. G. Sodini is with the Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

IEEE Log Number 9409886.

0018–9200/95$04.00 © 1995 IEEE

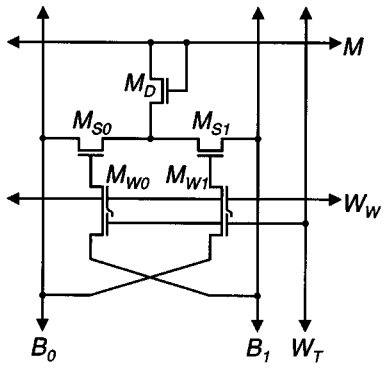

Fig. 3. Dynamic associative processor cell with dual-gate write devices.

mental results, and the following section describes a demonstration application.

II. Memory Design

This associative processor is *fully parallel*, meaning that it supports the simultaneous examination of multiple cells in multiple memory words. Other organizations are also possible [5], such as bit-parallel/word-serial or bit-serial/word-parallel. The first approach allows several memory words to share the same word logic and sense amplifier [6]. The second and more common organization requires that match and write operations examine or modify only one bit of each memory word at a time [7]–[10]. Multibit operations take several cycles, and combining results of the single-bit matches requires relatively complex word logic. Both of these organizations give up some of the available parallelism, but they have the advantage of requiring only conventional RAM cells in their implementations. In contrast, fully-parallel systems must use more specialized associative memory cells, which historically have been relatively expensive compared to standard RAM. However, this design uses a dense dynamic cell with only five n-channel transistors [11]. This technology should make fully-parallel systems competitive, especially in applications requiring many relatively simple PE's.

A. Dynamic Memory Cell

Fig. 3 shows the dynamic cell used in this design. The gates of M_{S0} and M_{S1} act as the storage nodes, and the states in which exactly one of these nodes are charged are called '0' and '1,' respectively. When both nodes are discharged, the cell is in the $\times$ state. Thus the cell stores a single ternary digit, or *trit*. The hypothetical state in which both nodes would be charged is not used.

The match operation begins with the match line M precharged high. If the trit presented on the bit lines $B_{0,1}$ does not match the stored trit, then the match line will discharge through one of the storage devices $M_{S0,1}$. The diode-connected transistor M_D prevents the bit lines of adjacent cells from being shorted through the match line.

The write operation is performed by raising the write-word W_W and write-trit W_T lines and driving the bit lines appropriately. If the W_T line is held low, then the write is masked and the state of the cell will remain unchanged. As in the content addressable memory cell of [12], the bit lines are crossed to provide adequate signal voltage across the storage devices when writing a 0 or 1.

The use of two write enable signals logically requires two switches in series, so that charge is transferred between the bit line and the storage node only when both switches are closed. One might imagine implementing these switches with conventional MOS transistors, but this configuration suffers from a charge sharing problem involving the parasitic junction capacitance at the intermediate node. Electrons could move from the bit line to the intermediate node any time W_T was asserted. If W_W were asserted some time later, these electrons could move onto the gate of storage device. Only a few such *spooning* cycles are needed to completely discharge the storage node and change the state of the cell. The process can run in reverse as well, charging the storage node by spooning electrons to the bit line.

Dual-gate CCD transistors greatly reduce the charge spooning problem by largely eliminating the parasitic junction capacitance. Some nonideal charge transfer is still possible, however, because electrons left behind in interface traps may eventually make their way across the dual-gate device, or else they may recombine with substrate holes to produce a net charge pumping current [13]. Similar mechanisms are responsible for charge transfer inefficiencies in charge coupled devices [14]. However, earlier experiments with a single-cell test chip indicated that these nonideal currents are small enough to allow the cell to maintain its state through one refresh cycle [11].

B. Match Sense Amplifier

Virtually all RAM sense amplifiers operate differentially, comparing signals from true and complementary bit lines. In contrast, the CAM sense amplifier has only a single-ended match line signal available. If the match is successful, then the match line will remain at its high precharged potential. Unsuccessful matches produce a mismatch current dependent on the number of mismatching trit positions. A complete mismatch will discharge the match line quite rapidly, but only a small current will flow if only one or two cells mismatch. The match sense amplifier must distinguish the successful matches from near mismatches, and must do so in the presence of undesirable noise signals.

Fig. 4 presents simulation results of successful and unsuccessful match operations, with the mismatch case demonstrating a moderate discharge rate. Note that even when the match is successful, the match line can be pulled down by capacitive coupling to the bit lines. This is the dominant noise signal which the sense amplifier must tolerate. A second source of noise is the transient switching pulse generated when the bit line drivers are enabled. The current surge produces an $L(dI/dt)$ drop on the supply rails, which in turn couples to the match line. The pulse dies away as the bit lines settle.

The sense amplifier design of Fig. 5 incorporates a multiplexer to support the associative processor's duplex match lines. The control signal $SASET$ is asserted during the precharge phase to pull the internal node $\overline{M}$ low. Either

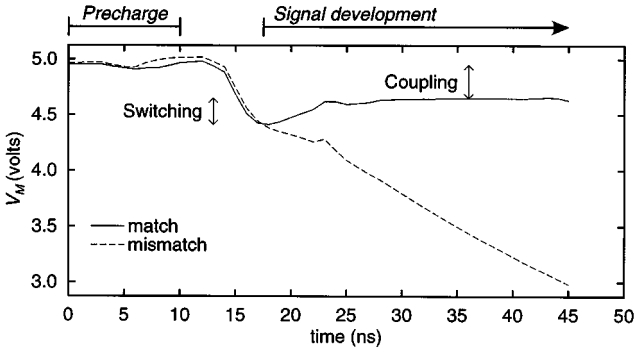

Fig. 4. Simulated match line potential with switching and coupling noise.

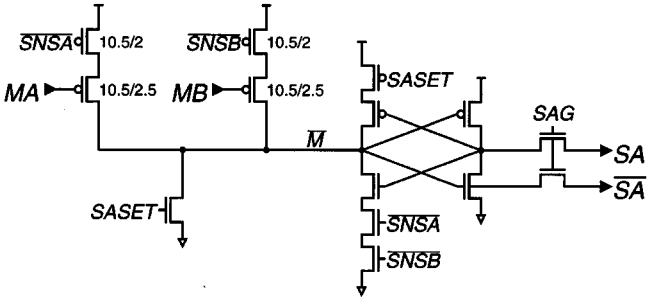

Fig. 5. Sense amplifier.

the $\overline{SNSA}$ or the $\overline{SNSB}$ signal will be driven low during the signal development phase, so that $\overline{M}$ will be pulled high if the corresponding match line falls more than a p-channel threshold voltage below the positive supply ($V_{DD} - |V_{TP}|$). The magnitude of V_{TP} is significantly greater than the expected coupling noise, and switching noise tolerance is achieved by delaying the $\overline{SNSA}$ and $\overline{SNSB}$ signals until after the noise pulse has had time to settle out.

The five transistor feedback gate allows the sense amplifier to double as a static latch. It is disabled during the precharge and sensing phases, but it provides active drive for the node $\overline{M}$ at other times. Finally, the function generator requires that the sense amplifier's true and complementary outputs remain stable during the precharge phase of the match. Two pass transistors provide the necessary isolation; they are enabled by the signal SAG.

C. Array Driver Circuits

While the sense amplifier is designed to tolerate switching noise, the driver circuits address the other half of the problem by minimizing noise generation. The write-word circuit of Fig. 6 needs the large n-channel device M_{N2} to provide sufficient output conductance to prevent coupling problems in the array. If all 256 PE's switched their W_W lines simultaneously, however, a noise pulse of a few 100 mV could result. To prevent this, the small transistor M_{N1} is turned on first, while the feedback transistor M_{P2} inhibits M_{N2} until the W_W line has fallen more than two threshold drops below V_{DD}.

A similar noise reduction strategy is used for the bit line driver circuit, with small and large drivers connected

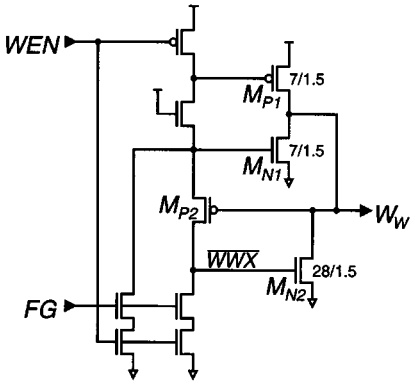

Fig. 6. Write-word driver.

in parallel. Instead of a feedback circuit, however, separate control signals are used to enable the large driver. This is done so that different timing can be used for the match and write operations. The large driver can be turned on earlier during the match, since the worst-case capacitive load is smaller.

III. Mesh Network

Most image processing algorithms have modest communication requirements. A two-dimensional rectangular mesh topology provides a natural representation for image data and maps readily into silicon. It works best for local communication within small neighborhoods of processing elements, but even relatively low-level algorithms such as computing histograms or labeling connected components can require some nonlocal communication. Several authors have proposed augmenting the basic mesh-connected computer with additional communication paths, such as broadcast busses [15]–[17] and tree or pyramid structures [18], [19]. The associative processor chip of this work retains the simple mesh topology but supports an asynchronous propagation mode to speed near-local communication. It also includes a response resolver to report global information on the state of the array [20].

Five of the processing element's specialized cells are used in the nearest-neighbor communication mode. The HB cells controls the net driver, so that when a PE writes to its HB cell, the bit is transmitted to the PE's four nearest neighbors. The value is then matchable in the neighbors' NEWS A cells. For example, each PE's NA cell behaves as if it is storing the same value as the PE's north neighbor's HB cell. A PE can examine all four nearest neighbors' HB cells in a single match, but communicating over greater distances requires copying the direction cell to the HB cell at each PE along the way.

In contrast, the reconfigurable mode allows asynchronous propagation through several PE's in a single clock period. In this mode, the NEWS B cells are used to establish communication links from neighboring PE's. For example, a PE writes a 1 to its SB cell to indicate that it will receive signals from its south neighbor. If the south neighbor writes a 1 to its HB cell, then the PE's own HB cell will also be set to 1, and so on to other connected PE's. This mechanism allows PE's to group themselves into connected regions, or *coteries* [8],

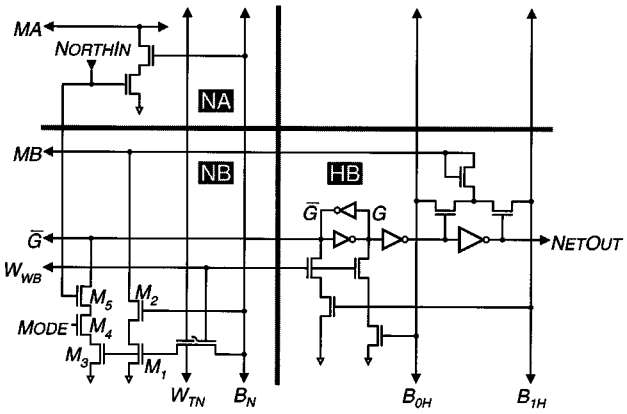

Fig. 7. Network cell circuits.

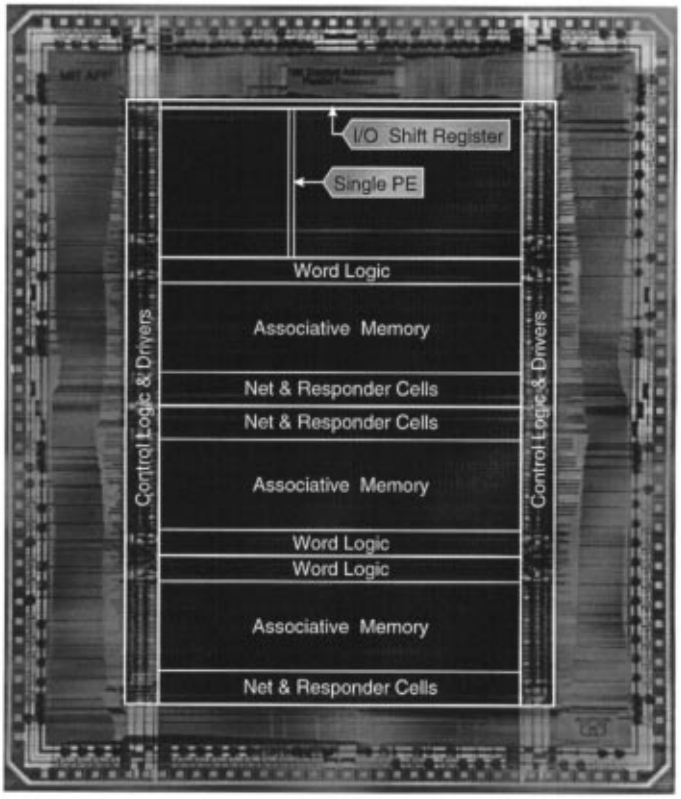

Fig. 8. Photomicrograph of integrated associative processor.

of arbitrary shape. Algorithms have been published using a coterie network to perform a variety of image processing and low-level vision tasks, such as labeling connected components and computing means, medians, and convex hulls [21].

Fig. 7 shows the circuit implementation of the HB cell and one of the four NEWS cell pairs (NA,NB). The NA cell is the simplest, using a two-transistor NAND stack to discharge the match line MA when both the north input and the bit line B_N are true. Because it has only one discharge path, the cell cannot perform the complete ternary match operation—there is no way to mismatch a presented 1. One can usually work around this limitation, however, and the slight inconvenience to the programmer wins a significant advantage for the circuit designer, in that the incoming signal need not be inverted locally. This permits a dense n-channel implementation and eliminates the need for n-wells. The NEWS A and B cells are in fact slightly smaller than the associative trit cells.

The HB cell uses a static register and a three-transistor comparator. It supports ternary matches, but can not store the $\times$ state. The flip-flop inverters are designed with weak p-channel devices, so that the cell can be written by pulling one side of the flip-flop down to V_{SS}. Masked writes are performed by holding both bit lines low.

The NB cell uses a three-transistor NAND stack to gate the network input, so that the left side of the HB flip-flop $(\overline{G})$ is pulled low when all three transistors conduct. That is, the HB cell is set when (M_3) the NB cell contains a 1 to enable north inputs, and (M_4) the network is in asynchronous reconfigurable mode, and (M_5) the network input is true. This NAND design was chosen over alternatives based on pass-transistor switches. While the pass transistor designs may provide faster network propagation, they have the disadvantage of capacitively coupling the signal being switched to the control signal at the gate. One must either drive the gate with an active circuit, such as a static register, or else ensure that the network is precharged to a known state before writing the configuration state. The first option consumes too much area, and the second unduly complicates the timing of the write operation. In contrast, the NAND implementation allows the configuration state to be stored dynamically on the gates of M_3 and M_1. Since both transistors' sources are connected to the supply, coupling through C_{gs} is eliminated. As in the associative trit cell, dual-gate transistors are used to support both horizontal and vertical write enables. Transistors M_1 and M_2 are used to examine the cell with the match operation. Like the NA cell, the NB cell can not mismatch a presented 1 trit.

IV. Results

Fig. 8 is a photomicrograph of the associative processor chip. One can immediately observe that it looks much more like a memory than a processor, with large arrays of repeated structures rather than numerous unique functional units. The chip integrates four arrays of 64 processing elements, and each PE has 64 trits of associative memory. The network is folded to connect the PE's in a 16 $\times$ 16 array, and the south edge of the logical network incorporates a shift register for interfacing with the format converter.

The chip was fabricated through the MOSIS service, using a 1.2 μm CCD-CMOS process. Conservative design rules $(L = 1.5\ \mu\text{m})$ were used to ensure functionality in this experimental process. Table I presents a summary of the chip's specifications. A 6 V supply is used to compensate for the process's high (2 V) second-poly threshold voltage. The chip performs match and write instructions in 175 ns. Power dissipation is highly dependent on the activity of the processing elements and the instructions being executed. An idle chip requires 48 mW to execute Nop (no operation) instructions, while a pathological program designed to move as much charge as possible burns a maximum of 690 mW. More

TABLE I
ASSOCIATIVE PROCESSOR CHIP SPECIFICATIONS

Associative memory	16 k	trits
Processing elements	256	
Transistors	134 k	
Minimum drawn channel length	1.5	μm
Package (PGA)	144	pins
Area		
Die	7.9×9.2	mm^2
Array	5.9×7.2	mm^2
Cell	32×33.5	μm^2
Minimum clock period	175	ns
Power		
Idle	48	mW
Typical	220	mW
Maximum	690	mW

typical image processing applications dissipate about 220 mW, with the network circuits accounting for 25% of that figure.

Expressed in terms of energy per computation, performing a 3×3 Laplacian convolution requires 76 nJ for each pixel processed. Although comparisons to very different architectures are naturally problematic, this result is of the same order as the energy used by other image processing chips implemented in similar technology. LSI Logic's convolution chips, the L64243 and L64240, require 25 nJ and 125 nJ, respectively [22]. The four-PE Image Signal Multiprocessor from Matsushita requires 283 nJ [23]. However, the associative processor's pixel-parallel organization makes it considerably more general. It does not, for example, impose any hardware limit on kernel size.

A 1024-PE demonstration system was constructed using four associative processing chips, the minimum number necessary to fully exercise two-dimensional interchip communication. Instruction vectors are delivered by a specialized controller connected to a host workstation.

V. DEMONSTRATION APPLICATION

A simplified image processing application was demonstrated on the associative processor to establish its utility in performing a pixel-parallel task.

Fig. 9(a) shows an unprocessed image of a child's toy block. A higher level algorithm might require a preprocessing step to remove unnecessary detail, such as the texture in the background. One could try a low-pass spatial filter, but this operation destroys not only the distracting details but also the useful edge information, so that the resulting image is merely blurred. A nonlinear filtering operation can do a better job of preserving segment boundaries while smoothing connected regions, as shown in Fig. 9(b). The algorithm is described in [3], but it can be thought of as a convolution in which each pixel computes its own kernel, so that no smoothing occurs where neighboring pixels differ by more than a given threshold.

The four-chip system was fed the 32×32 detail (Fig. 9(c)) of the larger image. The associative processor performed 100 iterations with a threshold of 16/128. Including the necessary refresh cycles, the operation was completed in 9.3 ms. Larger images can be readily accommodated by adding more chips to the system. Given sufficient hardware, the execution time is independent of the size of the image. This demonstration establishes the feasibility of performing real-time image processing with an associative processor.

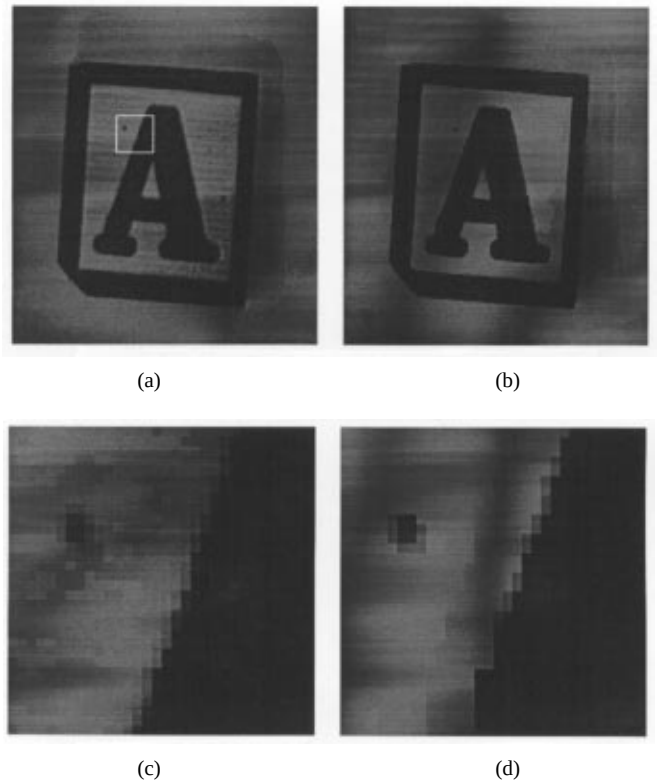

Fig. 9. Original and processed images of toy block. (a) Unprocessed image, with box indicating area of detail. (b) Simulator output after 100 iterations with threshold of 16/128. (c) Detail of unprocessed image. (d) Result of processing (c) on associative processor hardware, using the same procedure as in (b).

VI. CONCLUSION

An associative processor chip with 256 PE's and 16 kilotrits of memory has been described. The chip features a five-transistor dynamic memory cell using dual-gate devices. Substantial density gains should be achievable with improved technology, as cells similar to this one (although without the masked write feature) have proved to be quite dense when implemented in a more aggressive process [24].

Compact circuits for a reconfigurable mesh network and a response resolver were designed within memory pitch constraints. These should be readily applicable to other fine-grained parallel systems. An image smoothing and segmentation algorithm was presented as a sample pixel-parallel application, and the associative processing hardware was shown to perform the operation in under than 10 ms, less than a standard video frame period. This demonstration proves that real-time performance is attainable with an associative processor. The chip's power needs are comparable to those of more specialized systems implemented in similar technology, but the associative processor benefits from the flexibility inherent in its pixel-parallel organization.

ACKNOWLEDGMENT

The fabrication was provided by MOSIS.

REFERENCES

[1] P. A. Ruetz, "The architectures and design of a 20 MHz real-time DSP chip set," *IEEE J. Solid-State Circuits*, vol. 24, no. 2, pp. 338–348, Apr. 1989.
[2] P. C. Yu, S. J. Decker, H. Lee, C. G. Sodini, and J. L. Wyatt, Jr., "CMOS resistive fuses for image smoothing and segmentation," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 545–553, Apr. 1992.
[3] C. L. Keast and C. G. Sodini, "A CCD/CMOS-based imager with integrated focal plane signal processing," *IEEE J. Solid-State Circuits*, vol. 28, no. 4, pp. 431–437, Apr. 1993.
[4] C. C. Foster, *Content Addressable Parallel Processors*, Computer Science Series. New York: Van Nostrand Reinhold, 1976.
[5] R. M. Lea and I. P. Jalowiecki, "Associative massively parallel computers," *Proc. IEEE*, vol. 79, no. 4, pp. 469–479, Apr. 1991.
[6] I. N. Robinson, "Pattern-addressable memory," *IEEE Micro*, vol. 12, no. 3, pp. 20–30, June 1992.
[7] K. E. Batcher, "Bit-serieal parallel processing systems," *IEEE Trans. Computers*, vol. C-31, no. 5, pp. 377–384, May 1982.
[8] C. C. Weems, Jr., "The content addressable array parallel processor: Architectural evaluation and enhancement," in *IEEE Int. Conf. Computer Design*, 1985, pp. 500–503.
[9] R. Heaton, D. Blevins, and E. Davis, "A bit-serial VLSI array processing chip for image processing," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 364–368, Apr. 1990.
[10] M. Motomura, J. Toyoura, K. Hirata, H. Ooka, H. Yamada, and T. Enomoto, "A 1.2 million transistor, 33 MHz, 20 b dictionary search processor (DISP) ULSI with a 160 kb CAM," *IEEE J. Solid-State Circuits*, vol. 25, no. 5, pp. 1158–1165, Oct. 1990.
[11] F. P. Herrmann, C. L. Keast, K. Ishio, J. P. Wade, and C. G. Sodini, "A dynamic three-state memory cell for high-density associative processors," *IEEE J. Solid-State Circuits*, vol. 26, no. 4, pp. 537–541, Apr. 1991.
[12] J. P. Wade and C. G. Sodini, "Dynamic cross-coupled bit-line content addressable memory cell," *IEEE J. Solid-State Circuits*, vol. SC-21, no. 1, pp. 119–121, Feb. 1987.
[13] G. Groeseneken, H. E. Maes, N. Beltrán, and R. F. De Keersmaecker, "A reliable approach to charge-pumping measurements in MOS transistors," *IEEE Trans. Electron Devices*, vol. ED-31, no. 1, pp. 42–53, Feb. 1984.
[14] C. H. Séquin and M. F. Tompsett, *Charge Transfer Devices*. New York: Academic, 1975.
[15] Q. F. Stout, "Mesh-connected computers with broadcasting," *IEEE Trans. Computers*, vol. C-32, no. 11, pp. 826–830, Sept. 1983.
[16] V. K. P. Kumar and C. S. Raghavendra, "Array pocessor with multiple broadcasting," in *Proc. 12th Annual Symp. Comput. Architecture*, June 1985, pp. 2–10.
[17] Y. Chen, W. Chen, G. Chen, and J. Sheu, "Designing efficient parallel algorithms on mesh-connected computers with multiple broadcasting," *IEEE Trans. Parallel Dist. Syst.*, vol. 1, no. 2, pp. 241–246, Apr. 1990.
[18] S. L. Tanimoto, "A pyramidal approach to parallel processing," in *Proc. 10th Annual Symp. Comput. Architecture*, 1983, pp. 372–378.
[19] R. Cypher and J. L. C. Sanz, "SIMD architectures and algorithms for image processing and computer vision," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 37, no. 12, pp. 2158–2174, Dec. 1989.
[20] F. P. Herrmann and C. G. Sodini, "A dynamic associative processor for machine vision applications," *IEEE Micro*, vol. 12, no. 3, pp. 31–41, June 1992.
[21] M. C. Herbordt, C. C. Weems, and M. J. Scudder, "Nonuniform region processing on SIMD arrays using the coterie network," *Machine Vision Applicat.*, vol. 5, no. 2, pp. 105–125, Spring 1992.
[22] LSI Logic Corporation, L64240 and L64243 product descriptions, 1991.
[23] M. Maruyama, H. Nakahira, T. Araki, S. Sakiyama, Y. Kitao, K. Aono, and H. Yamada, "An image signal multiprocessor on a single chip," *IEEE J. Solid-State Circuits*, vol. 25, no. 6, pp. 1476–1483, Dec. 1990.
[24] T. Yamagata, M. Mihara, T. Hamamoto, Y. Murai, T. Kobayashi, M. Yamada, and H. Ozaki, "A 288 kb parallel content-addressable memory using a stacked-capacitor cell structure," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1927–1933, Dec. 1992.

Frederick P. Herrmann (S'87–M'95) was born in Chicago, IL, in 1965. He attended the Massachusetts Institute of Technology, Cambridge, MA, from 1983 to 1994 and received the S.B., S.M., and Ph.D. degrees in electrical engineering.

In 1994, he joined the technical staff at MIT Lincoln Laboratory. His research interests include associative processing and circuit design for specialized memories and liquid crystal displays.

Charles G. Sodini (S'80–M'82–SM'90–F'95) was born in Pittsburgh, PA, in 1952. He received the B.S.E.E. from Purdue University, Lafayette, IN, in 1974, and the M.S.E.E. and Ph.D. degrees from the University of California, Berkeley, in 1981 and 1982, respectively.

He was a Member of Technical Staff at Hewlett–Packard Laboratories from 1974 to 1982, where he worked on the design of MOS memory and later, on the development of MOS devices with very thin gate dielectrics. He joined the faculty of the Massachusetts Institute of Technology, Cambridge, MA, in 1983 where he is currently a Professor in the Department of Electrical Engineering and Computer Science. His research interests are focused on IC fabrication, device modeling, and device level circuit design, with emphasis on analog and memory circuits and systems.

Dr. Sodini held the Analog Devices Career Development Professorship of Massachusetts Institute of Technology's Department of Electrical Engineering and Computer Science and was awarded the IBM Faculty Development Award from 1985 to 1987. He has served on a variety of IEEE Conference Committees, including the International Electron Device Meeting where he was the 1989 General Chairman. He was the Technical Program Co-Chairman for the 1992 Symposium on VLSI Circuits and the 1993–1994 Co-Chairman of the Symposium. He has served on the Electron Device Society Administrative Committee from 1988–1994 and is currently a member of the Solid-State Circuits Council.