A Mixed-Signal Array Processor with Early Vision Applications

David A. Martin, Hae-Seung Lee, and Ichiro Masaki

***Abstract*— Many early vision tasks require only 6 to 8 b of precision. For these applications, a special-purpose analog circuit is often a smaller, faster, and lower power solution than a general-purpose digital processor, but the analog chips lack the programmability of digital image processors. This paper presents a programmable mixed-signal array processor which combines the programmability of a digital processor with the small area and low power of an analog circuit. Each processor cell in the array utilizes a digitally programmable analog arithmetic unit with an accuracy of 1.3%. The analog arithmetic unit utilizes a unique circuit that combines a cyclic switched-capacitor analog-to-digital converter (ADC) and digital-to-analog converter (DAC) to perform addition, subtraction, multiplication, and division. Each processor cell, fabricated in a 0.8-μm triple-metal CMOS process, operates at a speed of 0.8 MIPS, consumes 1.8 mW of power at 5 V, and uses 700 μm by 270 μm of silicon area. An array of these processor cells performed an edge detection algorithm and a subpixel resolution algorithm.**

***Index Terms*— Analog processing circuits, stereo vision, switched capacitor circuits.**

I. Introduction

As digital processors continue to increase in speed, many electronic systems have become almost entirely digital, having only an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC) at the edges of the system. For many applications, this is the optimal approach. However, there are some applications for which an analog or mixed-signal system offers superior performance and a more efficient use of power and silicon area than a digital system.

One such application is the initial processing performed on an image in a real-time vision system, usually referred to as early vision. Many early vision algorithms require a processor with only 6 to 8 b of accuracy, so the limited accuracy of an analog circuit is acceptable. Several analog vision chips which are faster, smaller, and lower in power than digital image processors have been designed [1]. They perform functions such as finding the focus of expansion [2], image smoothing [3], position determination [4], and stereo functions [5]. These analog implementations achieve higher performance with less area and power by exploiting characteristics of the vision algorithm which naturally map to a circuit architecture and by integrating the imaging and processing circuits on the same chip. The primary limitation of these analog implementations is that they can only perform one specific function. Digital image processors, although they usually use more power and area, can execute a variety of vision algorithms [6], [7].

This paper describes a mixed-signal array processor, fabricated in a digital CMOS process, that combines the high performance, low area, and low power of a special-purpose analog vision chip with enough programmability and flexibility to perform the operations needed for most early vision tasks. Also, although most commercial imagers are now fabricated in a charge coupled device (CCD) process, CMOS imagers are starting to gain in popularity [8], [9]. Since the mixed-signal array processor is designed in a digital CMOS process, a small array of processor cells can be included on a CMOS imager itself to perform simple early vision tasks. The mixed-signal array processor's key enabling circuit is a programmable analog arithmetic circuit which serves as the arithmetic logic unit (ALU) for each processor cell.

Section II provides an overview of the high-level processor architecture and functionality. The programmable analog ALU is described in Section III. Test results for the mixed-signal array processor and its ALU are presented in Section IV. Section V compares the mixed-signal array processor to several digital image processors.

II. Processor Architecture and Operation

A. Array

The chip-level architecture of the mixed-signal array processor is shown in Fig. 1. Although the figure shows the structure of a 5 $\times$ 5 array, the array is completely scalable; any size array from one processor cell to the maximum that will fit into the available chip area is possible. The processor is a rectangular array of processor cells, each of which can communicate with its four nearest neighbors via analog data lines. The N, S, W, and E signals are analog I/O lines that go off-chip. The $D0$–$D4$ signals are digital outputs.

The mixed-signal array processor works as follows. First, the processor is programmed through the programming bit stream, which is represented in Fig. 1 by the dotted line that begins with $D_{\rm IN}$ and snakes throughout the array. Every processor in the array can perform a different function; it is thus a multiple-instruction, multiple-data (MIMD) processor. However, once the processor cells are programmed, they perform the same function repeatedly; they are not reprogrammed for every new piece of data. This is acceptable for the data-independent algorithms common to most early vision tasks. After the array is programmed, an imager sends pixel values in a discrete-time stream of analog voltages to the array processor through the analog I/O lines. The processor performs the desired operations, and the results of these operations are sent

Manuscript received August 5, 1997; revised November 3, 1997. This work was supported by NSF under contract 9 117 724-MIP.

D. Martin is with Texas Instruments New Jersey Design Center, Edison, NJ 08837 USA.

H.-S. Lee and I. Masaki are with MIT Microsystems Technology Laboratory, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0018-9200(98)01699-0.

0018–9200/98$10.00 © 1998 IEEE

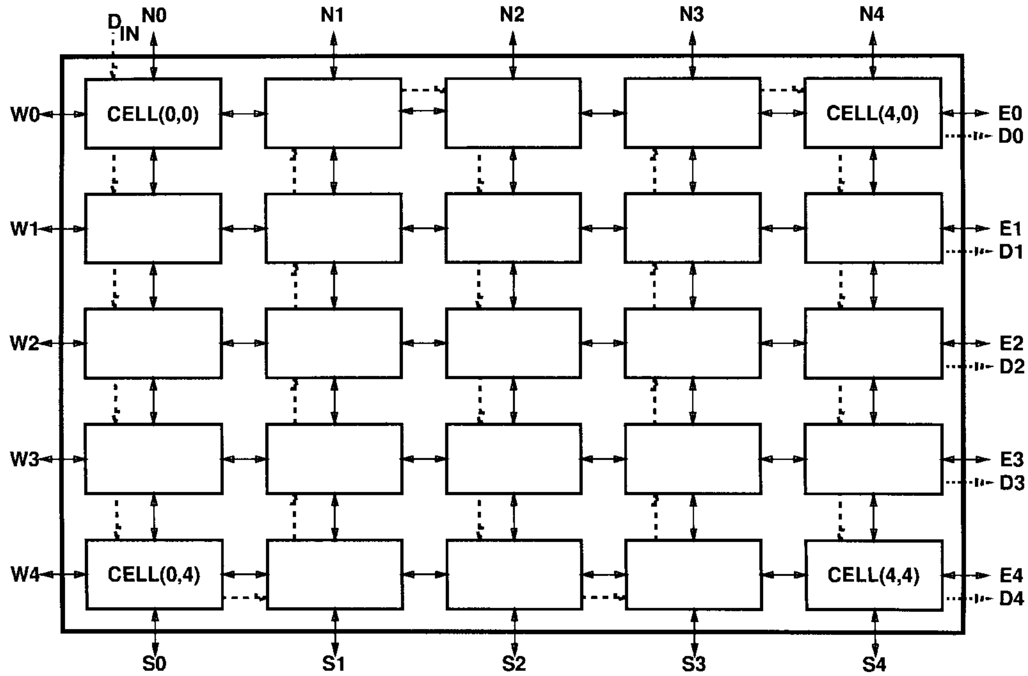


Fig. 1. Array architecture, shown for a 5 × 5 array.

out of the array processor in a digital format for any further processing.

For most applications, it is usually not practical to store an entire image in the processor at once. Instead, the imager sends the pixel values in a stream, of which the processor holds a small part (a pixel and its nearest neighbors), usually between 3 and 25 pixel values. Thus, the mixed-signal processor can easily perform the local operator operations common to most early vision tasks.

B. Processor Cell

The architecture of a single processor cell is shown in Fig. 2. It consists of a digital control register, an analog sample-and-hold to store one piece of data, an ALU, and a switch fabric. The control register controls the functionality of the ALU. The ALU has two analog inputs and one analog output; it can perform addition, subtraction, division, or multiplication. The switch fabric, which consists of NMOS pass gates, routes analog data among the analog storage unit, the ALU, and the four I/O lines (E, W, N, S).

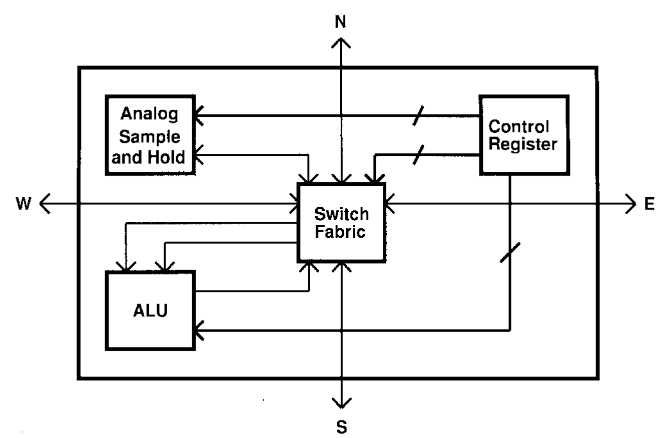


Fig. 2. Processor cell architecture.

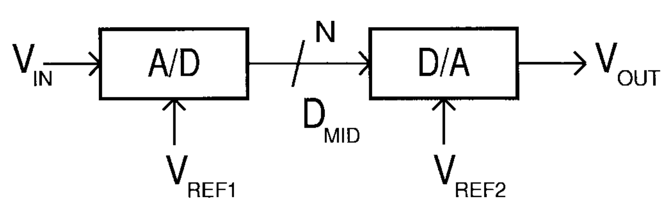


Fig. 3. A/D–D/A processor concept.

III. Design of the ALU

The ALU uses an ADC followed by a DAC to perform a calculation, as shown in Fig. 3. Assume that each subcircuit has N bits of resolution. The output of the ADC, $D_{\rm MID}$, equals $(V_{\rm IN}/V_{\rm REF1})2^N$. The output of the DAC, $V_{\rm OUT}$, is $(D_{\rm MID}/2^N)V_{\rm REF2}$. When the two equations are combined, the 2^N factor drops out, and the result is

$$V_{\rm OUT} = \frac{V_{\rm IN} V_{\rm REF2}}{V_{\rm REF1}}. \tag{1}$$

By using the ADC and DAC reference voltages as additional inputs to the circuit, four different arithmetic operations can be performed by the same circuit. The ALU has two inputs, V_1 and V_2, as shown in Fig. 4. Multiplication is implemented by connecting the input of the ADC to V_1, the reference voltage for the ADC to V_{K1} (representing a constant $K1$), and the reference voltage for the DAC to V_2. Using (1) gives $V_{\rm OUT} = (V_1 V_2)/V_{K1}$. For division, the input of the ADC is V_{K2} (representing a constant $K2$), the reference voltage for the ADC is V_1, and the reference voltage for the DAC is V_2.

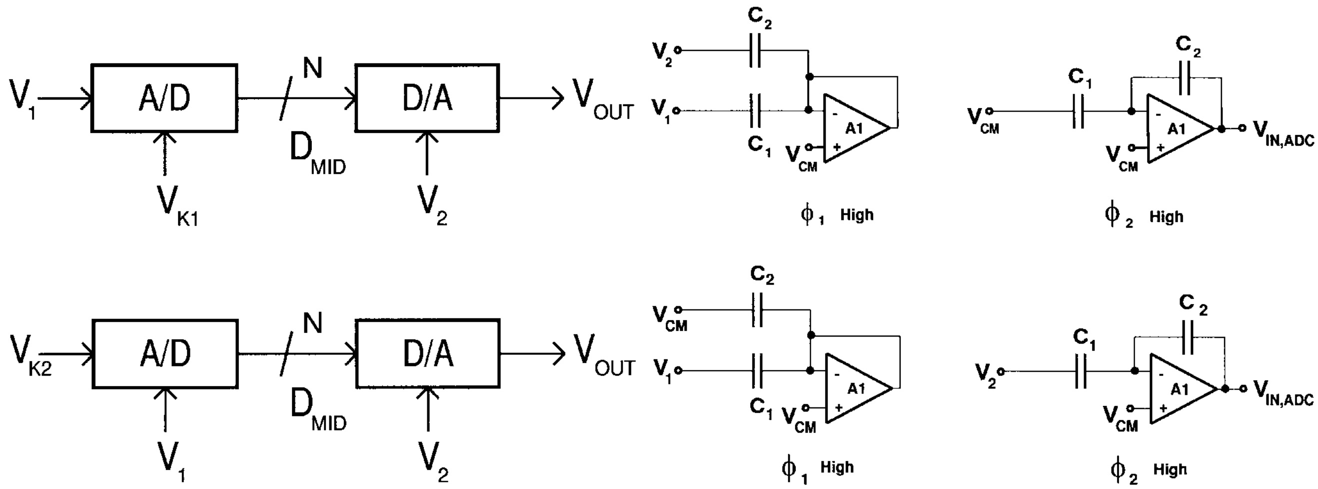


Fig. 4. Clockwise from first quadrant: add, subtract, divide, multiply.

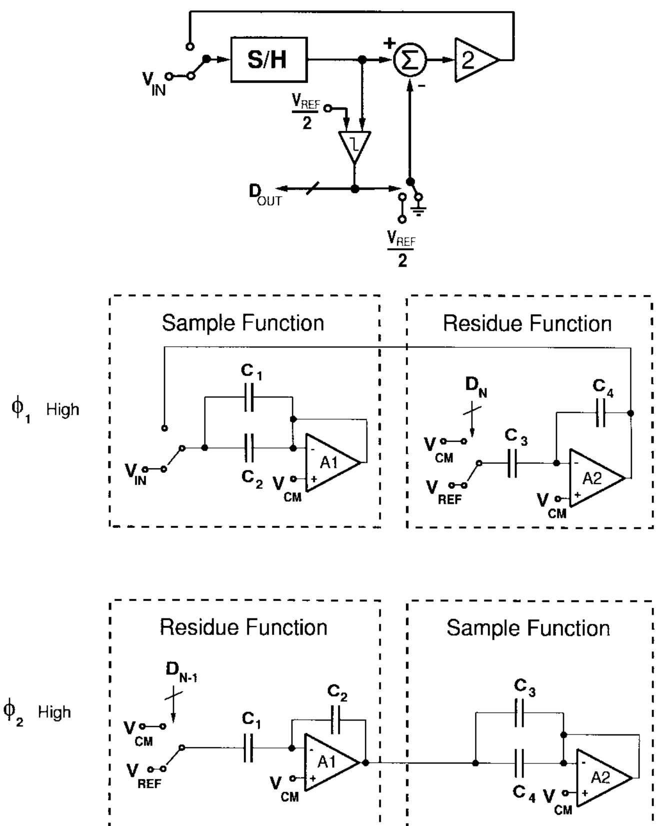


Fig. 5. ADC algorithm and schematic, all capacitors = 53 fF.

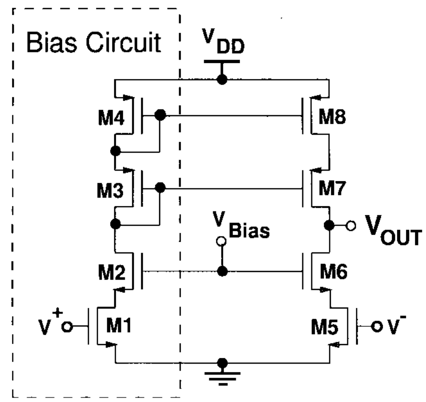


Fig. 6. Op amp schematic.

Using (1) gives $V_{\mathrm{OUT}} = (V_2/V_1)V_{K2}$. $K1$ and $K2$ can easily be changed by changing their respective bias voltages.

Addition and subtraction are implemented using two extra cycles prior to the ADC operation. For addition, C_1 is connected to V_1, and C_2 is connected to V_2 when ϕ_1 is high. When ϕ_2 is high, C_1 is connected to the common-mode voltage V_{CM} so that, at the end of ϕ_2, the output of the op amp is $V_1 + V_2$. For subtraction, C_1 is connected to V_1, and C_2 is connected to V_{CM} when ϕ_1 is high. When ϕ_2 is high, C_1 is connected to V_2 so that, at the end of ϕ_2, the output of the op amp is $V_1 - V_2$.

A. Design of the ADC

Because the array processor uses many copies of the ALU, small area and low power is a key design goal. Therefore, a two-stage pipelined cyclic architecture is used for both the ADC and the DAC [10]. This architecture has a small area requirement since the same circuit is used for each bit. In this implementation, both circuits are clocked for four clock cycles (a total of eight phases), yielding eight nominal bits of resolution.

The ADC algorithm is shown in Fig. 5. Its digital output is serial, starting with the most significant bit (MSB). The ADC first samples its input and compares it against a voltage reference $V_{\mathrm{REF}}/2$, which is half of the full-scale voltage V_{REF}. If the input is greater than $V_{\mathrm{REF}}/2$, the MSB is set to one and $V_{\mathrm{REF}}/2$ is subtracted from the input voltage; otherwise, the MSB is set to zero and the input voltage is not modified. The result is then multiplied by two to get the residue voltage, which is sampled for the next cycle.

The circuit schematic of the ADC is also shown in Fig. 5; it is implemented with two op amps, four capacitors, and switches. When ϕ_1 is high, C_1 and C_2 sample the output of op amp A2 while C_3 and C_4 perform the reference subtraction and multiply-by-two operations. When ϕ_2 is high,

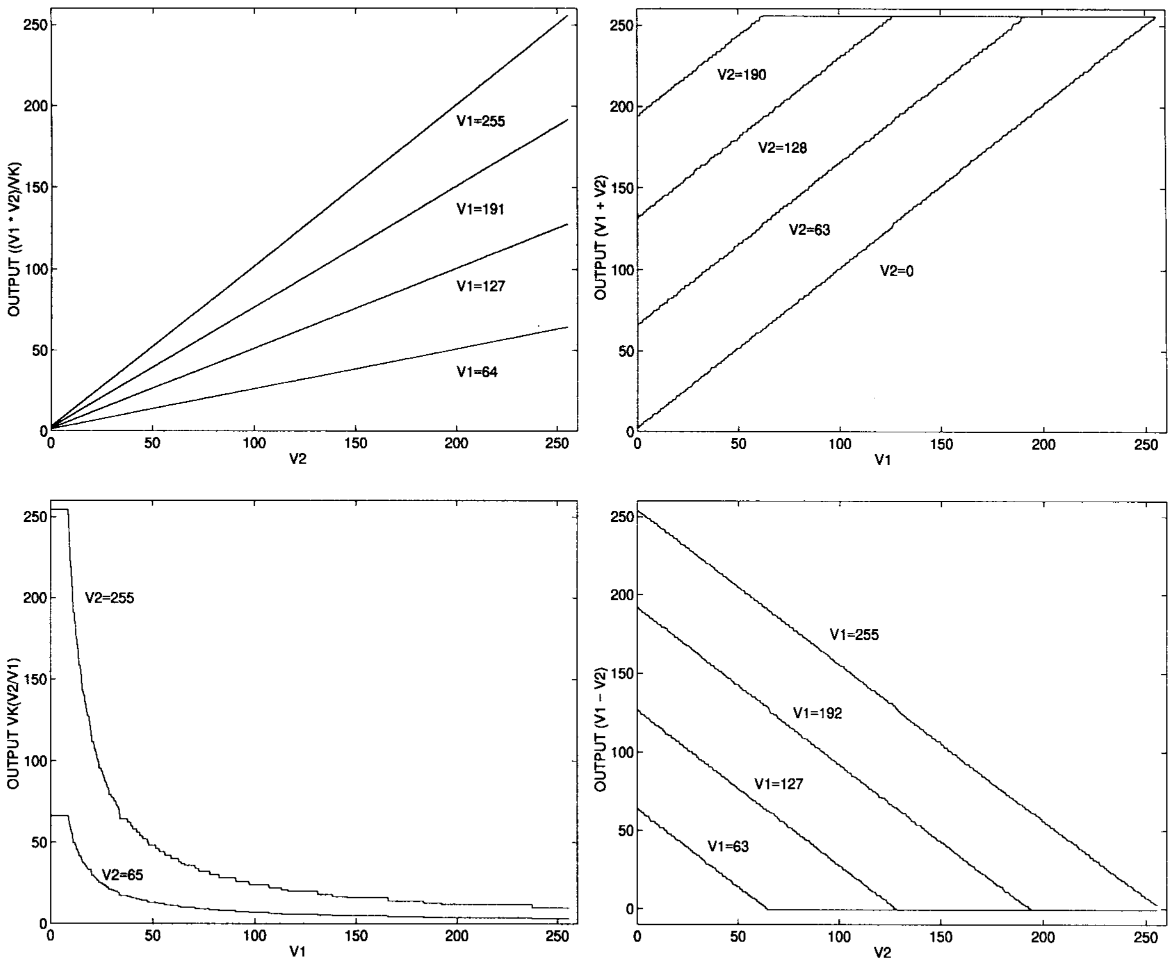


Fig. 7. Arithmetic data. Clockwise from first quadrant: add, subtract, divide, multiply. All axes normalized to a 0 → 255 numerical range.

the capacitors (and their op amps) alternate functions. Note that zero is represented by the common-mode voltage $V_{\rm CM}$.

During the nonoverlapping period between ϕ_1 and ϕ_2, the op amps are reused as comparator preamps by attaching $V_{\rm REF}/2$ to the top plates of the sampling capacitors and by opening the op amp's feedback path, as described in [11].

B. Design of the DAC

The DAC also employs a two-stage pipelined cyclic converter. It uses the same type of algorithm and circuit implementation as the ADC except that it adds a reference voltage and then divides its state voltage by two on every cycle instead of subtracting a reference and multiplying by two on every cycle. Also, the ADC produces its output MSB first while the DAC requires its data least significant bit (LSB) first. Therefore, there is an 8-b bidirectional buffer between the two converters to hold the digital data. The ALU is pipelined so that both the ADC and DAC operate continuously.

C. Op Amp Design

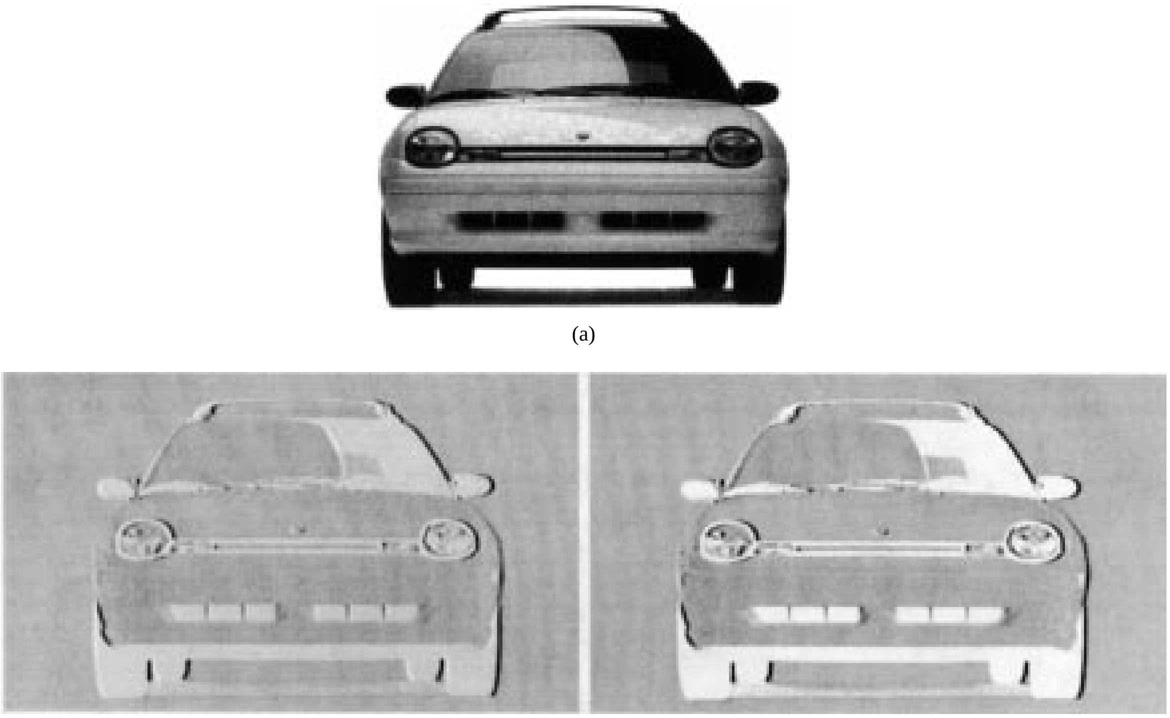
The schematic of the op amp used in the ADC (and the DAC) is shown in Fig. 6. Small area and low power were key requirements in the design. Transistors M1–M4 form a biasing circuit for the op amp. Every two adjacent op amps, A1 and A2, share this biasing circuit so that the two op amps together require only 12 transistors. The positive input is always connected to the common-mode voltage $V_{\rm CM}$ of 0.95 V. In simulations, the op amp has a dc gain of 3000 and a unity gain frequency of 45 MHz with a 5-V supply.

IV. Experimental Results

Two test chips were fabricated in a 5-V, 0.8-μm triple-metal CMOS process, one with a single processor cell for characterizing the ALU and one with a 5 × 5 processor array, shown in Fig. 9, for testing the processor's functionality. Each processor cell uses 700 μm × 270 μm of area and consumes 1.825 mW of power at 0.8 MIPS/cell.

The four arithmetic functions were tested at a speed of 0.8 MIPS/cell. Sample data plots for each function are shown in Fig. 7. In these plots, one input is held constant at several values while the other input is ramped through all possible input values. For simplicity, the 2.048 V range of the inputs has been normalized to a 0-to-255 numerical range.

The addition, multiplication, and division operations do not produce a result exceeding 255, and the subtraction operation does not produce a result below zero. This saturation effect is due to the fact that the DAC output, which is the output of the ALU, cannot go above its full-scale voltage or below $V_{\rm CM}$. Also, note that although the result in (1) is continuous, the output actually contains quantization error due to the ADC.



(a)

(b) (c)

Fig. 8. Edge maps: (a) input image, (b) mixed-signal processor edge map, and (c) software edge map.

TABLE I
SUMMARY OF ARITHMETIC FUNCTIONS

Function	Maximum Absolute Error (%)	Differential Error Range (%)
Addition	1.1	$\pm$ .7
Subtraction	1.2	$\pm$ .9
Multiplication	1.3	$\pm$ .8
Division, numerator $>$ 45	1.3	$\pm$ .7
Division, all inputs	-6.2	$\pm$ 5.8

The accuracy of the arithmetic functions is summarized in Table I. In all cases except one, the output is within 1.3% of the correct output. The division operation has 6.2% of error for some denominator values below 45. This is due to the fact that, in the division operation, the numerator is used as the reference for the ADC. When this reference is low, the dynamic range of the ADC is reduced, but the error sources are not reduced. This reduces the accuracy of the ADC.

One application which requires an early vision system with low cost, high performance, and programmability is the vision system for an automotive intelligent cruise control (ICC) system. Using cameras and a stereo algorithm, an ICC system detects the distance D to the car ahead and adjusts the motor speed to keep D constant [12], [13]. The stereo algorithm calculates the distance to an object from the disparity of the object. The disparity is the difference between the x-coordinates of an object in the right and left images, and it is inversely proportional to the distance to the image [14].

One early vision task required by an ICC stereo algorithm is an edge detection function, which was implemented with a Sobel filter. The equation for this filter is

$$\text{Edge}(x,y) = \frac{K_2 P(x+1,y) + K_2 P(x+2,y)}{K_1 P(x-1,y) + K_1 P(x-2,y)}. \quad (2)$$

$P(x,y)$ is the pixel value at location (x,y). K_1 and K_2 are constants used to adjust for the light levels. The input image is shown in Fig. 8,[1] along with both an edge map generated by the mixed-signal array processor and an edge map generated in software from the same input image. The edge locations are the same in both edge maps (using a simple thresholding metric to select the edges), and the edge values are within 5% of each other.

In addition, the stereo algorithm utilizes a subpixel resolution algorithm to increase the resolution of the disparity measurement and thus the distance measurement. Assigning an edge value to E_2 and the value of the adjoining edges to E_1 and E_3, it can be shown [15] that the fractional part of the actual location of Edge 2 is given by

$$X_{\text{Fractional}} = \frac{E_2 - .75E_1 - .25E_3}{E_2 - .5E_1 - .5E_3}. \quad (3)$$

This equation was implemented in the mixed-signal array processor; it increased the edge position resolution, and thus the distance measurement, by a factor of four. These two algorithms demonstrate the array processor's functionality and show that the ALU's accuracy is sufficient for the early vision tasks required by an ICC system.

V. COMPARISON TO DIGITAL IMAGE PROCESSORS

The mixed-signal array processor was compared to two different digital array processors designed for early vision applications, integrated memory array processor (IMAP) [6] and high-density parallel processor (HDPP) [16]. IMAP was fabricated in a 0.55-μm BiCMOS process and consists of an array of 64 processor cells, each of which has a bank of SRAM

[1] Input image courtesy of Dodge.

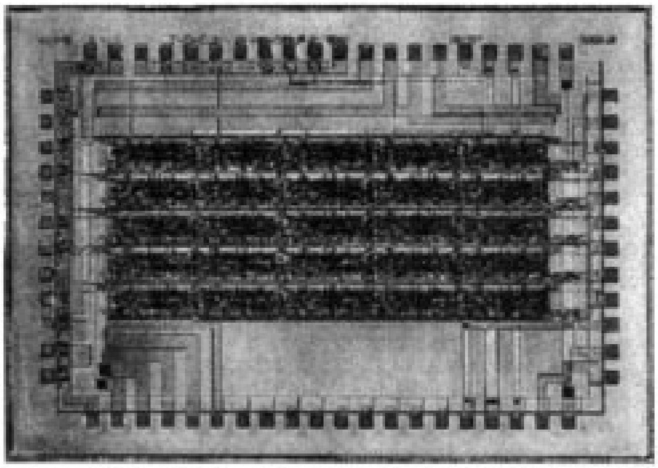


Fig. 9. Array chip die photo.

TABLE II
COMPARISON OF MIXED-SIGNAL AND DIGITAL PERFORMANCE PER CELL

Metric	Mixed-Signal	IMAP	HDPP
MIPS/cell	0.8	3.63	0.06
Power/cell	1.825mW	53mW	.06mW
Cell Area	0.19 mm^2	$3mm^2$	$.015mm^2$
$MIPS/mm^2$	4.2	1.2	4.0
MIPS/mW	0.43	0.068	1.00

memory. HDPP was fabricated in a 0.6-μm CMOS process and consists of an array of 4096 cells, each of which has a bank of DRAM memory.

To compare the two processors, the MIPS of an individual processor cell were divided by the processor cell area and power consumption to get a measure of how efficiently they use power and area. The results are shown in Table II. It is assumed that in an actual application, all three of the array processors could be programmed so that most or all of the processor cells would be used on every cycle.

For a given instruction speed, IMAP uses six times more power and three times more area than the mixed-signal processor. HDPP has approximately the same efficiency as the mixed-signal processor, being slightly more efficient in power and slightly less efficient in area. There are two important factors to note, however. First, the speeds of the digital processors are for a multiplication operation. The literature describing the digital image processors did not specify a speed for a division operation, but it would probably take about four to eight times as many instructions to perform a division operation. The mixed-signal array processor, on the other hand, can perform a division operation as well as a multiplication operation at 0.8 MIPS. Second, both HDPP and IMAP were fabricated in more advanced (0.55 μm and 0.6 μm versus 0.8 μm) processes. Despite these two factors, the mixed-signal processor is still more efficient than IMAP and approximately as efficient as HDPP in its use of power and area.

The digital image processors have certain advantages over the mixed-signal processor, such as the ability to manipulate data at the bit level. One conclusion to draw from this comparison is that an analog processor, preferably a programmable one, should be used to perform the initial low-precision floating point vision operations, such as division and multiplication, in which analog circuits have an advantage over digital circuits. The remaining operations, which usually involve logical, bit, and memory storage operations, are most efficiently performed in a digital processor.

VI. CONCLUSION

The use of a programmable analog ALU has enabled the design of a mixed-signal array processor which operates with an accuracy of 1.3% with less power and area than comparable digital vision processors. It performed the edge detection and subpixel resolution algorithms needed by an ICC system. A 1 cm^2 array of the mixed-signal processor cells would dissipate 1 W at 420 MIPS. Fabricated in a digital CMOS process, a small array of processor cells can be included on a CMOS imager itself to perform simple early vision tasks.

ACKNOWLEDGMENT

The authors would like to acknowledge the advice of C. G. Sodini from MIT, J. Bergendahl from MIT, K. Nagaraj from Texas Instruments, and V. Gopinathan from Lucent. Fabrication of the chips was done through MOSIS.

REFERENCES

[1] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison Wesley, 1989.
[2] I. S. McQuirk, "An analog VLSI chip for estimating the focus of expansion," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1996.
[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, pp. 431–437, Apr. 1993.
[4] D. L. Standley, "An object position and orientation IC with embedded imager," *IEEE J. Solid-State Circuits*, vol. 26, pp. 1853–1858, Dec. 1991.
[5] G. Erten and R. Goodman, "Analog VLSI implementation for stereo correspondence between 2-D images," *IEEE Trans. Neural Networks*, vol. 7, pp. 266–277, Mar. 1996.
[6] N. Yamashita *et al.*, "A 3.84 GIPS integrated memory array processor with 64 processing elements and a 2-MB SRAM," *IEEE J. Solid-State Circuits*, vol. 29, pp. 1336–1343, Nov. 1994.
[7] J. C. Gealow, F. P. Herrmann, L. T. Hsu, and C. G. Sodini, "System design for pixel-parallel image processing," *IEEE Trans. VLSI Syst.*, vol. 4, pp. 32–41, Mar. 1996.
[8] B. Ackland and A. Dickinson, "Camera on a chip," in *ISSCC 1996 Dig. Tech. Papers*, vol. 39, pp. 22–25, 1996.
[9] A. Pollack, "New technology promises 'camera on a chip'," *New York Times*, May 27, 1997.
[10] R. H. McCharles, V. A. Saletore, and D. A. Hodges, "An algorithmic analog-to-digital converter," in *ISSCC 1977 Dig. Tech. Papers*, 1977.
[11] H. Ohara, H. Ngo, M. H. Armstrong, C. F. Rahim, and P. R. Gray, "A CMOS programmable self-calibrating 13-bit eight-channel data acquisition peripheral," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 930–938, Dec. 1987.
[12] Y. Ninomiya *et al.*, "A real-time vision for intelligent vehicles," in *Intelligent Vehicles'95*, IEEE Industrial Electronics Society, pp. 315–320, Sept. 1995.
[13] I. Masaki *et al.*, "Cost-effective vision systems for intelligent vehicles," in *Intelligent Vehicles'94*, IEEE Industrial Electronics Society, pp. 39–43, Oct. 1994.
[14] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
[15] D. A. Martin, "ADAP: A mixed-signal array processor with early vision applications," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1996.
[16] J. C. Gealow, "An integrated computing structure for pixel-parallel image processing," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1997.